INOVANCE

Advancing industrial technology,
for a better world

H5U and Easy Series Programmable Logic Controller

Programming Guide

Industrial Automation | New Energy Vehicle | Intelligent Elevator | Intelligent Robot | Digital Energy | Rail Transit

Data code 19012249A19

Legal and Disclaimer

Copyright

Copyright © Shenzhen Inovance Technology Co., Ltd. 2025 All rights reserved.

This documentation is the exclusive property of Shenzhen Inovance Technology Co., Ltd. No individual or entity may excerpt, reproduce, modify, translate, or distribute any content herein without written consent from Inovance.

Legal action will be taken against infringement.

Trademarks

INOVANCE is a registered trademark of Shenzhen Inovance Technology Co., Ltd. and its affiliates. All other trademarks or registered trademarks mentioned in this documentation are the property of their respective owners. Unauthorized use of these trademarks by third parties for any purpose without written authorization could violate the rights of their owners.

Disclaimer of Liability

Due to continuous updates and improvements of products and technologies, the content of this documentation may not fully match the actual products. In the event of any discrepancies, the actual products shall prevail.

The contents are subject to change without notice due to product upgrade.

Waste Disposal

The storage, use, and disposal of this product (including optional accessories) must comply with local laws and regulations.

Qualifed Personnel

The product/system described in this documentation may be operated only by personnel qualified for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel can identify the risks of the product/system and prevent possible dangers.

Proper Use of the Product

Proper transportation, storage, assembly, installation, commissioning, operation, and maintenance are required to ensure the safe operation of the product without any problems. The required ambient conditions must be met. All operations must follow the guidelines provided in this documentation.

Preface

Introduction

The H5U series PLC, a new generation of small-sized PLC developed by Inovance, supports EtherCAT bus communication and features powerful motion control and distributed I/O control functions. It allows process encapsulation and reuse using the FB/FC function, and multi-layer network communication through the RS485, CAN, Ethernet, and EtherCAT interfaces.

The Easy series small-sized and medium-range PLCs are available in 14 models, covering the demands of automation equipment requiring small footprint, multi-axis motion control, accurate temperature control, and easy networking.

This document describes the basic knowledge of H5U and Easy series PLC programming, quick start guidance, communication, motion control, and the use of high-speed counters.

Related Manuals

Document Name	Data Code	Description
H5U Series Programmable Logic Controller User Guide	19011517	This document describes the product information, mechanical installation, communication connection specifications, features, and usage methods of the PLC.
Easy Series Programmable Logic Controller User Guide	PS00006444	This document describes the product information, mechanical installation, and electrical installation of the product.
Easy321 and Easy511 Programmable Logic Controller User Guide	PS00017976	This document describes the whole scene task information of Easy321 and Easy511 products, including product introduction, installation, wiring, configuration commissioning, special functions, fault diagnosis, and maintenance.
Easy321S Programmable Logic Controller User Guide	PS00021275	This document describes the whole scene task information of Easy321S products, including product introduction, installation, wiring, configuration commissioning, special functions, fault diagnosis, and maintenance.
H5U & Easy Series Programmable Logic Controller Instruction Guide	19012250	This document describes the basic instructions and complex instructions used for programming applications, as well as examples of these instructions.
H5U and Easy Series Programmable Logic Controller Programming Guide (this guide)	19012249	This document introduces the basic knowledge of PLC programming, quick start guidance, communication, motion control, and the use of high-speed counters.

Access to the Guide

This guide is not delivered with the product. You can obtain the PDF version in the following way:

- Do keyword search under Service and Support at http://www.inovance.com.
- Scan the QR code on the product with your smart phone.
- Scan the QR code below to install My Inovance app, where you can search for and download user guides.

Contents

1 New Functions 22

2 Overview 29

2.1 Introduction 29

2.1.1 Product Introduction 29

2.1.2 Software Introduction 29

2.1.3 Networking Solution 29

2.2 Software Acquisition and Installation 32

2.2.1 Acquisition Pathway 32

2.2.2 Requirements on the Installation Environment 32

2.2.3 Installing the Software 33

2.2.4 Uninstalling the Software 35

2.3 Software Interface 36

3 F1 Redirection 38

3.1 Online Help 38

3.2 Precise Redirection 39

4 Quick Start 42

4.1 Overview 42

4.2 Communication Connection 42

4.2.1 Overview 42

4.2.2 Ethernet Connection 42

4.2.3 USB Connection 53

4.3 Programming Flowchart 55

4.4 Programming Example 56

4.4.1 Requirements 56

4.4.2 Creating or Opening a Project 56

4.4.3 Connecting to the Target PLC 58

4.4.4 (Optional) Performing Hardware Configuration 58

4.4.5 Programming and Compiling 59

4.4.6 (Optional) Logging in to PLC 60

4.4.7 Downloading the Program 60

4.4.8 HMI Monitoring 61

4.5 Switching PLC Working Mode 61

4.6 Editing the Program Online 62

4.7 Setting a Program Scan Cycle 63

4.8 Setting an EtherCAT Task Cycle 63

4.9 Packing and Decompressing a Project Archive 64

4.10 Logging in to PLC 66
4.10.1 Overview 66
4.10.2 PLC Login and Logout 66
4.10.3 Login Password Management 67
4.11 Trace Monitor Variables 70
4.11.1 Overview 70
4.11.2 Adding Trace Monitor Variables 70
4.11.3 Importing and Exporting Trace Data 75
4.12 Data Record 75
5 Programming Basics 80
5.1 Overview 80
5.2 Elements 80
5.2.1 Bit Elements 80
5.2.2 Word Elements 81
5.2.3 Special Elements 82
5.3 Variables 83
5.3.1 Introduction 83
5.3.2 Defining Variables 84
5.3.3 Defining an Array 87
5.3.4 Defining a Structure 87
5.3.5 Defining Enumerations 88
5.3.6 Defining an IP Variable 90
5.3.7 Defining a String Variable 90
5.3.8 Defining a Specific Union 90
5.3.9 Using the Variable 91
5.4 Binding Variables to Element Addresses 92
5.4.1 Introduction 92
5.4.2 Variable Properties 93
5.4.3 Binding a Basic Variable to an Element 93
5.4.4 Binding an Array Variable to an Element 93
5.4.5 Binding a Structure Variable to an Element 94
5.4.6 Binding Enumeration Variables to Soft Elements 95
5.4.7 Binding a Specific Union Variable to an Element 96
5.5 Variable Used as the Variable Group Subscript 97
5.5.1 Rules of Usage 97
5.5.2 Programming Example 98
5.6 Pointer Type Variables 101
5.6.1 Introduction 101

5.6.2 Obtaining the Directing Address 102
5.6.3 Pointer Address Operation 103
5.6.4 Indirect Addressing Operation 104
5.6.5 Programming Example 105
5.7 System Variables 105
5.7.1 Introduction 105
5.7.2 _SYS_CAN CAN Interface Information 106
5.7.3 _SYS_COM Serial Port Information 110
5.7.4 _SYS_COM_SAVE Serial Port Parameters 116
5.7.5 _SYS_ECAT_Master Operation Status 117
5.7.6 _SYS_ECAT_Slave Operation Status 120
5.7.7 _SYS_EncAxis Encoder Axis Information 123
5.7.8 _SYS_Ethernet Network Port Information 128
5.7.9 _SYS_INFO PLC Operation Information 134
5.7.10 _SYS_MC_Axis Motion Control Axis Information 142
5.7.11 _sGROUPAXIS_INFO Axis Status 147
5.7.12 _sMCGROUP_INFO Axis Group Status 148
5.7.13 _sGROUPPOS_INFO Axis Target Position 151
5.7.14 _SYS_PN PN Operation Status 152
5.8 Timer 153
5.8.1 Introduction 153
5.8.2 Pulse Timer (TPR) 154
5.8.3 On-Delay Timer (TONR) 154
5.8.4 Off-Delay Timer (TOFR) 155
5.8.5 Accumulation Timer (TACR) 156
5.9 Graphical Block Instructions 157
5.9.1 Instruction Composition 157
5.9.2 Instruction-Level Programming 158
5.9.3 Label Number Function 159
5.10 Subprogram 161
5.10.1 Introduction 161
5.10.1.1 Introduction of Subprogram 161
5.10.1.2 Subprogram Execution Mechanism 161
5.10.2 General Subprogram Application 162
5.10.2.1 Creating a General Subprogram 162
5.10.2.2 Calling the General Subprogram 163
5.10.3 Encrypting Subprogram Application 163
5.10.3.1 Encrypting a General Subprogram 163

5.10.3.2 Calling the Encrypted Subprogram 164
5.10.4 Interrupt Subprogram Application 164
5.10.4.1 External Interrupt Subprogram 164
5.10.4.2 Timed Interrupt Subprogram 165
5.10.4.3 Comparison interrupt Subprogram 167
5.10.4.4 Communication Interrupt Subprogram 167
5.11 Function Blocks and Functions 170
5.11.1 Function Block 171
5.11.2 Function 177
5.11.3 Authorizing a Function Block 180
5.11.4 Setting Initial Values of the Function Block 182
5.11.5 Encrypting the Function Block or Function 185
5.12 Execution Block 186
5.13 Cross-Reference 188
5.13.1 Cross Reference List 188
5.13.2 Browsing Cross-References 189
5.14 Folder 191
5.15 Element Usage Table 194
5.16 Number of Opened Windows 196
5.17 PLC Time Setting 196
5.18 I/O Refresh 199
5.18.1 Introduction 199
5.18.2 Local I/O Refresh 200
5.18.3 Module I/O Refresh 201
5.18.4 Remote I/O Refresh 203
6 Programming Language 206
6.1 Structured Text (LiteST) 206
6.1.1 Overview 206
6.1.2 Expressions 206
6.1.3 Variables 209
6.1.4 Constants 209
6.1.5 Function Blocks, Functions, Subprograms, and Interrupts 210
6.1.6 Syntax Instructions 210
6.1.6.1 Overview 210
6.1.6.2 Assignment Instructions 211
6.1.6.3 Calling a Function Block 212
6.1.6.4 IF Instructions 212
6.1.6.5 CASE Instructions 213

6.1.6.6 WHILE Instructions 215
6.1.6.7 REPEAT Instructions 215
6.1.6.8 FOR Instructions 216
6.1.6.9 EXIT Instructions 216
6.1.6.10 CONTINUE Instructions 217
6.1.6.11 RETURN Instructions 217
6.1.6.12 REGION Instructions 218
6.1.6.13 Comment 220
6.1.7 List of Instructions 220
6.1.7.1 Basic Axis Control Instructions 220
6.1.7.2 Cam and Gear Instructions 222
6.1.7.3 Encoder Instructions 222
6.1.7.4 Communication Instructions 223
6.1.7.5 Timer Instructions 223
6.1.7.6 Interrupt Instructions 224
6.1.7.7 Operation Instructions 224
6.1.7.8 Other Instructions 225
6.1.7.9 Instruction Examples 226
6.1.8 Troubleshooting 227
6.1.8.1 Protection Against Zero Divisor 227
6.1.8.2 Array Out-of-Bounds 228
6.1.8.3 Infinite Loop 228
6.1.8.4 Precautions for Array Subscript Usage 229
6.2 Ladder Diagram (LD) 229
6.2.1 Overview 229
6.2.2 En/Eno Instructions 232
6.3 Sequential Function Chart (SFC) 234
6.3.1 SFC Elements 234
6.3.2 SFC Program Compilation 236
6.3.3 Find, Replace, and Goto 237
6.3.4 SFC Compilation 238
6.4 Tips 238
6.4.1 ST Quick Input 238
6.4.2 ST Prompt Upon Hovering 238
6.4.3 ST Automatic Bracket Completion 239
6.4.4 Unit Conversion 239

7 Extension Modules 242
7.1 H5U Extension Module Configuration 242
7.1.1 Overview 242
7.1.2 Hardware Configuration 243
7.1.3 Configuring an Extension Module 243
7.1.3.1 Configuring a Digital Input (DI) Module 243
7.1.3.2 Configuring a Digital Output (DO) Module 245
7.1.3.3 Configuring an Analog Input (AI) Module 246
7.1.3.4 Configuring an Analog Output (AO) Module 248
7.1.3.5 Configuring a Temperature Detection Module 250
7.2 Easy Extension Module and Expansion Card Configuration 251
7.2.1 Overview 251
7.2.2 Extension Module Configuration 254
7.2.2.1 Hardware Configuration 254
7.2.2.2 Configuring an Extension Module 254
7.2.2.2.1 Configuring a Digital Input (DI) Module 255
7.2.2.2.2 Configuring a Digital Output (DO) Module 256
7.2.2.2.3 Configuring Digital Input and Output Modules 258
7.2.2.2.4 Configuring an Analog Input (AI) Module 258
7.2.2.2.5 Configuring an Analog Output (AO) Module 263
7.2.2.2.6 Configuring a Temperature Detection Module 267
7.2.2.2.7 Configuring a Communication Module 269
7.2.2.2.8 Configuring a 2-Channel Encoder Input Module 273
7.2.2.2.9 Configuring a 2-Channel High-Speed Count Module 277
7.2.2.2.10 Configuring a Power Module 283
7.2.2.2.11 Application Example 284
7.2.3 Expansion Card Configuration 285
7.2.3.1 Hardware Configuration 285
7.2.3.2 Configuring an Expansion Card 286
7.2.4 Application Example 287
7.3 GL20-RTU-ECT Extension Module Configuration 288
7.3.1 Overview 288
7.3.2 Configuring Hardware and Extension Module 290
7.3.3 Fault Diagnosis 293
7.3.3.1 Diagnosis Function 293
7.3.3.2 Fault Code 295
7.3.4 Disabling a Module 296
7.3.4.1 Overview 296
7.3.4.2 Disabling a Module Through the Module List 297

7.3.4.3 Disabling a Module Through System Variable 298
7.4 GR10-EC-6SW Branch Module Configuration 300
7.4.1 Overview 300
7.4.2 Adding a Branch Module and Its Slave Device 301
7.4.3 Adding the Branch Module and Its Slave Device 302
7.5 GS20-ECT-8L Module Configuration 303
7.5.1 Overview 303
7.5.2 Module Configuration 303
7.5.3 Fault Diagnosis 312
7.5.3.1 EtherCAT Fault Diagnosis 312
7.5.3.2 IO-Link Fault Diagnosis 313
7.5.4 Object Lists 314
7.5.4.1 Process Data 314
7.5.4.2 EtherCAT Object Dictionary Data (CoE Objects) 318
7.5.4.3 Configuration Data for Process Data Communication 325
7.5.4.4 IO-Link Slave Configuration Data 332
7.6 Basic Functions 333
7.6.1 Auto Scan (Easy) 334
7.6.2 Disabling (H5U and Easy) 336
7.6.2.1 Disabling Module Through Right-Click Menu 336
7.6.2.2 Disabling Module Through System Variables 337
7.6.3 Enabling an Extension Module (H5U&Easy) 338
8 Serial Communication 340
8.1 Overview 340
8.2 Serial Communication Network 340
8.3 Free Protocol Configuration 342
8.3.1 Configuring the Free Protocol 342
8.3.2 (Optional) Disabling the Free Protocol 342
8.4 Master Configuration 343
8.4.1 Configuring Modbus-RTU and Modbus-ASCII Masters 343
8.4.2 (Optional) Disabling a Modbus-RTU Slave 348
8.5 Slave Configuration 350
8.5.1 Configuring Modbus-RTU and Modbus-ASCII Slaves 350
8.6 Application Example 353
8.7 Serial Parameter Modification 356
8.7.1 Modifying COM Port Parameters 356
8.7.2 Modifying Slave Address Parameters 357
9 Ethernet Communication 358
9.1 Overview 358

9.2 Setting the IP Address 358
9.3 Modbus TCP Communication Configuration 359
9.3.1 Configuring a Modbus-TCP Master 359
9.3.2 (Optional) Disabling a Modbus-TCP Slave 363
9.3.3 Configuring a Modbus-TCP Slave 365
9.3.4 Application Example 367
9.4 FINS TCP&UDP Communication Configuration 372
9.4.1 FINS TCP&UDP Configuration 372
9.4.2 Application Example 374
9.4.3 Error Codes 378
9.5 SFTP Function 379
9.6 (Optional) Network Firewall 381
9.6.1 Network Firewall 381
9.6.2 Configuring a Network Firewall 381
9.6.3 Application Example 386
10 CAN Communication 388
10.1 Overview 388
10.2 Hardware Interface 388
10.3 CAN Network 389
10.3.1 CAN Communication Networking 389
10.3.2 Relationship Between CAN Communication Distance and Baud Rate 389
10.3.3 System Variable of CAN Interface 390
10.4 CANlink Communication 390
10.4.1 Principle of CANlink3.0 Communication 390
10.4.2 System Variable of CANlink Interface 391
10.4.3 CANlink Indicators 392
10.4.4 Configuring a CANlink Network 393
10.4.5 CANlink Application Example 401
10.4.6 Troubleshooting for CANlink Communication 405
10.5 CANopen Communication 405
10.5.1 CANopen Communication Protocol 405
10.5.2 List of CANopen Axis Control Instructions 406
10.5.3 Related Terms 407
10.5.4 CANopen Indicators 407
10.5.5 CANopen Configuration 408
10.5.5.1 Configuring a CANopen Master 408
10.5.5.2 Configuring a CANopen Slave 413

10.5.6 Troubleshooting for CANopen Communication 421
10.5.6.1 Troubleshooting 421
10.5.6.2 Error Codes 422
11 EtherCAT Communication 425
11.1 Overview 425
11.2 Master Configuration 425
11.2.1 Importing Device Description XML 425
11.2.2 Scanning a Device 426
11.2.2.1 Offline Scanning 426
11.2.2.2 Online Scanning 431
11.2.3 Configuring a Master 433
11.2.4 Device Start/Stop, Disable, and Enable 435
11.2.5 Monitoring the Master Status 436
11.2.6 List of System Variables 438
11.3 Slave Configuration 440
11.3.1 General Settings 440
11.3.2 Process Data 444
11.3.3 Startup Parameters 446
11.3.4 IO Functional Mapping 448
11.3.5 Device Start/Stop, Disable, and Enable 452
11.3.6 Disabling a Slave Through Instruction 453
11.3.7 List of System Variables 455
11.4 Fault Diagnosis 457
11.4.1 Fault Diagnosis for IN-OUT Reverse Connection 457
11.4.2 Fault Diagnosis for Continuous Frame Loss 458
11.4.3 Fault Diagnosis for Occasional Frame Loss 459
11.5 Troubleshooting 460
11.5.1 Fault Indication 460
11.5.2 Troubleshooting 460
11.5.3 Fault Codes 460
12 Ethernet-IP Communication 464
12.1 Overview 464
12.2 Technical Specifications 464
12.2.1 Ethernet-IP Transmission Specifications 464
12.2.2 Ethernet-IP Communication Specifications 465
12.2.3 Ethernet-IP Quick Reference Table 466
12.2.4 Ethernet-IP Solution Selection Example 468
12.3 Class 1 Communication 469
12.3.1 Planning IP Addresses for Ethernet-IP Devices 469

12.3.2 Master Configuration 469
12.3.2.1 Adding an Ethernet-IP Slave 469
12.3.2.2 Exporting an EDS File 470
12.3.3 Slave Configuration 474
12.3.3.1 General Settings 474
12.3.3.2 Connection Settings 475
12.3.3.3 Configuring IO Variable Mapping 478
12.3.4 Ethernet-IP Master Application Example 479
12.3.5 Ethernet-IP Slave Application Example 490
12.3.6 Tag Communication 491
12.3.6.1 Configuring Producer Tag Data 491
12.3.6.2 Ethernet-IP Consumer Tag Connection 492
12.3.6.3 Setting Tag Data Set 494
12.3.6.4 Tag Communication Application Example 495
12.4 Service Message Tag Communication 496
12.4.1 Service Message Tag: Server Configuration 497
12.4.2 Service Message Tag: Client Configuration 497
12.4.3 Application Example 499
12.5 Fault Diagnosis 500
12.5.1 Loop Detection 500
12.5.2 Address Conflict Detection 505
13 PROFINET Communication 510
13.1 Overview 510
13.2 Configuration Process 516
13.2.1 TIA Portal Configuration 516
13.2.2 AutoShop Configuration 519
13.3 Enable and Disable 521
14 OPC UA Communication 523
14.1 Overview 523
14.2 OPC UA Configuration 524
14.3 Application Example 533
14.4 Configuring Fault Diagnosis Function 540
15 MC Communication 549
15.1 Overview 549
15.2 Configuration 549
15.3 List of System Variables 550
15.4 Supported Specifications 551
15.5 Fault Codes 553

16 Motion Control 555
16.1 Introduction of Motion Control Axis 555
16.1.1 Overview 555
16.1.2 PLCopen State Machine 559
16.1.3 Axis Units 560
16.1.4 Axis Configuration Parameters 561
16.1.5 Axis System Variables 563
16.1.6 List of Axis Control Instructions 567
16.2 Motion Control Axis Settings 569
16.2.1 Creating a Project 569
16.2.2 Project Configuration 570
16.2.3 Setting Axis Parameters 576
16.2.3.1 Bus Servo Axis 577
16.2.3.2 Local Pulse Axis 577
16.2.4 Programming 578
16.2.5 Downloading the Project 581
16.2.6 Basic Motion 582
16.2.6.1 Preconditions 582
16.2.6.2 PLC Program Control 583
16.2.6.3 Online Commissioning 586
16.3 Motion Control Axis Configuration 589
16.3.1 Comparison of Bus Servo Axis and Local Pulse Axis 589
16.3.2 Basic Settings 589
16.3.3 Mode/Parameter Setting 594
16.3.3.1 Configuration Interface 594
16.3.3.2 Encoder Mode 594
16.3.3.3 Mode Settings 595
16.3.3.4 Software Limits 596
16.3.3.5 Deceleration Upon Axis Fault 596
16.3.3.6 Following Error 596
16.3.3.7 Axis Speed Setting 597
16.3.3.8 Torque Setting 597
16.3.3.9 Probe Setting 597
16.3.3.10 Output Setting 597
16.3.3.11 Options 598
16.3.3.12 Hardware Limit Logic 598
16.3.4 Homing 598
16.4 Online Monitoring 601

16.5 Axis Control Functions 603
16.5.1 Overview 603
16.5.2 Online Commissioning 604
16.5.3 Rules for Calling Instructions 610
16.5.4 Limit Processing 610
16.5.5 Positioning Curve 611
16.6 Dragging a Motion Control Axis 613
16.7 Modifying Axis Configuration Parameters 614
16.8 Fault Type 618
17 High-Speed Counter 620
17.1 About High-Speed Counter 620
17.2 Creating a Counter Axis 620
17.3 User Units and Unit Conversion of Counter Axis 620
17.4 Work Mode Setting 624
17.4.1 Linear Mode 624
17.4.2 Rotary Mode 625
17.5 Counter Parameter Setting 625
17.5.1 Overview 626
17.5.2 Counting Modes 626
17.5.2.1 Overview 626
17.5.2.2 Phase A/B Mode 627
17.5.2.3 CW/CCW Mode 628
17.5.2.4 Pulse & Direction Mode 628
17.5.2.5 Single Phase Counting 628
17.5.3 Reset Setting 629
17.5.4 Probe Terminal Setting 629
17.5.5 Preset Terminal Setting 629
17.5.6 Compare Output Terminal Setting 630
17.6 Counter Axis Instruction Application (H5U) 630
17.6.1 Overview 630
17.6.2 Instruction for Axis Position Counting and Velocity Measurement 630
17.6.3 Axis Position Preset Instruction 631
17.6.4 Probe Instruction 632
17.6.5 Comparison Instructions 634
17.6.6 High-Speed Hardware Comparison Output 635
17.6.7 Comparison Interrupt 636
17.7 Counter Axis Instruction Application (Easy) 638
17.7.1 Overview 638

17.7.2 Instruction for Axis Position Counting and Velocity Measurement 638
17.7.3 Axis Position Preset Instruction 639
17.7.4 Probe Instruction 640
17.7.5 Comparison Instructions 642
17.7.6 High-Speed Hardware Comparison Output 644
17.7.7 Comparison Interrupt 645
17.7.8 Axis Gear Ratio Setting 646
17.7.9 Axis Linear/Rotary Mode Setting 647
18 Interpolation Functions 648
18.1 Introduction 648
18.1.1 Overview 648
18.1.2 List of Axis Group Instructions 649
18.1.3 Configuration Interface 649
18.2 Interpolation Operation 652
18.2.1 Overview 652
18.2.2 Creating an Axis Group 652
18.2.3 Enabling the Axis Group 653
18.2.4 Linear Interpolation 653
18.2.5 Circular Interpolation 654
18.2.6 Elliptical Interpolation 655
18.2.7 Axis Group Stop 656
18.2.8 Axis Group Pause 657
18.2.9 Single-Axis Motion 657
18.2.10 Setting the Current Position 658
18.2.11 Reading the Current Status 659
18.2.12 Resetting an Axis Group Fault 660
18.2.13 Homing 660
18.3 Buffer and Transition 661
18.3.1 Overview 661
18.3.2 Interrupt+No Transition 662
18.3.3 Buffer+No Transition 662
18.3.4 Previous Velocity+No Transition 663
18.3.5 Additional Angle Mode 664
18.4 Handling Methods of Single-Axis Configuration Parameters in Interpolation 664
18.5 System Variables 665
18.5.1 Axis Status of Axis Group _sGROUPAXIS_INFO 665
18.5.2 Status of Axis Group _sMCGROUP_INFO 666

18.5.3 Axis Target Position within Axis Group_sGROUPPOS_INFO 669
18.6 Fault Codes 670
19 Bus Encoder Axis 675
19.1 Introduction 675
19.2 Software Configuration 675
19.2.1 Basic Settings 675
19.2.2 Unit Conversion 677
19.2.3 Mode/Parameter Setting 679
19.2.3.1 Configuration Interface 679
19.2.3.2 Linear/Rotary Mode Selection 679
19.2.3.3 Counter Mode Selection 680
19.2.3.4 Frequency Sampling Period 682
19.2.3.5 Input Filter Time 682
19.2.3.6 Input Terminal Function Selection 682
19.2.3.7 Output Terminal Function Selection 683
19.3 System Variables 684
19.4 Function Examples 687
19.4.1 Configuration Setup 687
19.4.2 Counter Enable 688
19.4.3 Counter Preset 689
19.4.4 Probe Functions 690
19.4.5 One-Dimensional Comparison Output 692
19.4.6 Two-Dimensional Comparison Output 694
19.4.7 Digital Output (DO) Terminal Control 695
19.4.8 Reading Axis Status 695
20 Electronic Cam 696
20.1 Introduction 696
20.2 Software Configuration 696
20.2.1 Overview 696
20.2.2 Cam Node Setting 697
20.2.3 Cam Curve Setting 698
20.2.4 Importing/Exporting an Electronic Cam 698
20.2.5 Uploading a Cam Table 699
20.2.6 Calling System Variables and Instructions 699
20.3 System Variables 699
20.3.1 Cam Node 700
20.3.2 Cam Table 700
20.3.3 Cam Contact Node 701

20.4 State Machine 702
20.5 Electronic Cam Operation 704
20.5.1 Gear Operation 704
20.5.2 Cam Operation 706
20.5.3 Cam Table 709
20.5.3.1 Introduction 709
20.5.3.2 Cam Table Specifications 709
20.5.3.3 Data Flow of Cam Table 710
20.5.3.4 Creating a Cam Table 710
20.5.3.5 Switching the Cam Table 711
20.5.3.6 Modifying Cam Data 712
20.5.3.7 Saving the Cam Table 713
20.5.4 Master Axis Phase Shift 714
20.5.5 Motion Superimposition 714
20.5.6 Handling Methods for Single-Axis Configuration Parameters in Cam or Gear 715
21 Offline Commissioning 717
21.1 Overview 717
21.2 Starting Offline Commissioning 718
21.3 Description of Offline Commissioning for Motion Control Axis 721
21.4 Simulation and Commissioning 721
21.4.1 Overview 721
21.4.2 PLC Configuration 721
21.4.3 HMI Configuration 722
21.4.4 Starting Commissioning 723
21.5 Ethernet/IP Offline Commissioning 723
21.5.1 Overview 723
21.5.2 Slave Offline Configuration 723
21.5.2.1 Overview 723
21.5.2.2 Configuring a Slave (Simulation) 724
21.5.2.3 Configuring a Master (Physical) 724
21.5.2.4 Commissioning 725
21.5.3 Master Offline Commissioning 726
21.5.3.1 Overview 726
21.5.3.2 Configuring a Slave (Physical) 726
21.5.3.3 Configuring a Master (Simulation) 727
21.5.3.4 Commissioning 728
21.6 OPC UA Offline Commissioning 728

22 Memory Management 730
22.1 Overview 730
22.2 Memory Management of Customized Variable Table 730
22.2.1 Expanding and Collapsing Complex Type Variables 730
22.2.2 Monitoring Variables 730
22.2.3 Reading and Writing Memory Data 731
22.2.4 Synchronizing and Clearing Data 731
22.2.5 Saving and Loading Data 732
22.2.6 Editing Initial Values and Comments 733
22.2.7 Changing and Displaying Base 733
22.3 Memory Management of Soft Elements 733
22.3.1 Operation Interfaces 733
22.3.2 Data Operation 734
22.3.3 Bit Comments 735
22.3.4 Rules for Editing Different Data Types 736
22.4 Function Example 736
23 Configuring the PID Function 738
24 Project Management 749
24.1 Project Comparison 749
24.2 Project Encryption 756
24.3 Batch Download 757
25 Fault Diagnosis 762
25.1 Fault Diagnosis Through Panel 762
25.1.1 Indicators 762
25.1.2 Description of MFK Keys 763
25.1.2.1 Overview 763
25.1.2.2 Restoring Default IP Address 763
25.1.2.3 Downloading User Program in the SD Card 764
25.1.2.4 LED Display of the CPU Module 764
25.2 Fault Diagnosis Through Software 764
25.2.1 Accessing Basic PLC Information 764
25.2.2 Viewing Operation Logs 765
25.3 Error Codes 766
26 FAQ 839
26.1 FAQ for AutoShop Installation 839
26.2 FAQ for AutoShop Usage 840
27 Appendix 843
27.1 Modbus Protocol 843
27.1.1 Description of Modbus Message 843

27.1.2 Modbus RTU Message Frame 843
27.1.3 Modbus ASCII Message Frame 843
27.1.4 Modbus TCP Message Frame 844
27.1.5 Parameter Definition 845
27.1.5.1 Modbus Data Model 845
27.1.5.2 Parameter List 845
27.1.5.3 Parameter Description 846
27.1.6 List of Fault Codes 851
27.2 Firmware Programming and Upgrade 852
27.2.1 USB Firmware Programming 852
27.2.2 SD Card Firmware Programming 853
27.3 Firmware Upgrade 855
27.4 Function Application of Download File Generation 856
27.4.1 Generating Down Files 856
27.4.1.1 Overview 856
27.4.1.2 Generating a Down File 856
27.4.1.3 Upgrading the Down File through an SD Card 857
27.4.1.4 Downloading the Down File Through AutoShop 857
27.4.1.5 Downloading and Uploading the Down File Through HMI 858
27.4.1.6 Compatibility Issues 859
27.4.2 Generating Updown Files 859
27.4.2.1 Overview 859
27.4.2.2 Generating an Updown File 860
27.4.2.3 Opening the Updown File 861
27.4.2.4 Downloading and Uploading the Updown File through HMI 861
27.4.2.5 Downloading and Uploading the Updown File Through AutoShop 862
27.4.2.6 Downloading the Updown File through an SD Card 863
27.5 Communication Application of Customized Variables 863
27.5.1 Overview 863
27.5.2 Example Project Requirements 863
27.5.3 PLC Programming 863
27.5.3.1 Accessing Customized Variables Through HMI Tag Communication Function 864
27.5.3.2 Accessing Customized Variables Through Address Mapping 865
27.5.4 HMI Configuration 865
27.5.4.1 Accessing Customized Variables Through HMI Tag Communication Function 865
27.5.4.2 Accessing Customized Variables Through Address Mapping 868
27.5.5 Example Running Result 870

1 New Functions

V4.11.0.5

Revision Date	Software and Firmware Version	Release Version	Description
January 2026	• Software version: AutoShop V4.11.0.5 • Firmware version: V6.11.0.1	A19	Modified Configuring a Communication Module, and added the GL20-2CAN series expansion module.

V4.11.0.0

Revision Date	Software and Firmware Version	Release Version	Description
September 2025	• Software version: AutoShop V4.11.0.0 • Firmware version: V6.5.0.0	A19	• Addition ◦ 3.2 Precise Redirection ◦ 5.3.5 Defining Enumerations and Binding Enumeration Variables to Soft Elements ◦ REGION ◦ En/Eno ◦ 5.10.4.4 Communication Interrupt Subprogram ◦ 9.5 SFTP Function ◦ PN Operation Status System Variable • Modification ◦ Added online help in 3 F1 Redirection. ◦ Updated dual-IP settings and automatic IP address acquisition in 4.2.2 Ethernet Connection. ◦ Added a pop-up window for modifying variable name in 5.3.2 Defining Variables. ◦ Added description about using two variables as array subscript/two-dimensional array in Rules of Use. ◦ In _SYS_CAN for CAN Interface Running Information, added description about the _CANOpen.RestartMaster, _CANSet.SetFlag, and _CANSet.BaudRate variables. ◦ Added description about _MbMst.CompatMode in _SYS_COM for Serial Port Running Information.

Revision Date	Software and Firmware Version	Release Version	Description
			◦ Added description about SFTP server diagnosis in _SYS_ETHERNET for Ethernet Information. ◦ Added LD program cross reference and ST program cross reference in Cross-References. ◦ Added expressions and restrictions in FB, FC, Subprogram, and Interrupt and Overview of Ladder Diagram (LD). ◦ Added Numerical Conversion in the Program Area and ST Automatic Bracket Completion in Tips. ◦ Modified the interface pictures for configuring AI and AO modules in Configuring Extension Modules. ◦ Added the description that Modbus (serial port) supports functions 05 and 06 and added support to low baud rates in Configuring Modbus-RTU or Modbus-ASCII Master. ◦ Added support to low baud rates in8.5.1 Configuring Modbus-RTU and Modbus-ASCII Slaves. ◦ Added the Manufacturer Error code bus servo axis loop variable in Basic Settings. ◦ Modified Disabling (H5U and Easy) and added 7.6.2.2 Disabling Module Through System Variables. ◦ Added description about status parameters in EtherNet/IP Master Application Example. ◦ Added description about two CSV files generated during exporting and comment pop-up window during variable editing in Application FAQ.

V4.10.2.0

Revision Date	Software and Firmware Version	Release Version	Description
December 2024	• Software version: AutoShop V4.10.2.0 and later • Firmware version: V6.4.0.0 and later	A18	• Addition ◦ F1 Online Help ◦ Managing Login Password ◦ 4.12 Data Record

Revision Date	Software and Firmware Version	Release Version	Description
			◦ Setting the PLC Time ◦ 5.18 I/O Refresh ◦ 5.16 Number of Opened Windows ◦ Configuring 2-Channel High-Speed Count Modules ◦ Configuring the Power Module ◦ Disabling Modules ◦ 21.6 OPC UA Offline Commissioning ◦ Project Encryption Management ◦ Batch Program Download ◦ 23 Configuring the PID Function ◦ Scanning the Device Online ◦ 11.4 Fault Diagnosis ◦ Configuring the Fault Diagnosis Function • Modification ◦ Added element comment modification in Ladder Diagram (LD). ◦ Added GL20-0032ETN, GL20(GL20S)-0004ER, and GL20(GL20S)-0004ETP-2A modules in DI Modules. ◦ Added GL20-4AD, GL20-8ADV, and GL20-8ADI extension modules in AO Modules. ◦ Modified system variables in System Variable List. ◦ Set watchdog parameters in Basic Settings.

V4.10.0.0

Revision Date	Software and Firmware Version	Version	Description
October 2024	• Software version: AutoShop V4.10.0.0 and later • Firmware version: V6.3.0.3 and later	A17	Corrected minor errors.
July 2024	• Software version: AutoShop V4.10.0.0 and later	A16	• Addition ◦ Execution Blocks ◦ 5.13.1 Cross Reference List

Revision Date	Software and Firmware Version	Version	Description
	• Firmware version: V6.3.0.3 and later		◦ 5.15 Element Usage Table ◦ 9.6.1 Network Firewall ◦ 14 OPC UA Communication ◦ 26 FAQ • Modification ◦ Modified the procedure and description in 4.4.2 Creating or Opening a Project. ◦ Modified comparison of execution blocks in 24.1 Project Comparison. ◦ Modified the procedure and parameter description of the batch alias modification function in Scanning the Device and Configuring the Master. ◦ Modified parameter information (the Modbus retry count can be set to 0) in Configuring Modbus-RTU/Modbus-ASCII Master and Configuring Modbus-TCP Master. ◦ Added description about PC IP addresses in multiple network segments in 4.2.2 Ethernet Connection. ◦ Modified descriptions in Diagnosis Fault, 7.4.1 Overview, and 11.5.2 Troubleshooting. ◦ Added error codes 8005, 8006, and 8403 in Error Codes. ◦ Updated system variables in System Variable List. ◦ Updated the slave running status information in _SYS_ECAT_Slave Running Status. ◦ Modified user unit information in 19.2.2 Unit Conversion, Axis Position Count and Speed Measurement Instructions, Axis Position Count and Speed Measurement Instructions, and Axis Unit.

Other historical versions

Revision Date	Version	Description
March 2024	A15	• Addition ◦ 18.2.6 Elliptical Interpolation ◦ Uploading and Downloading Down Files Through HMI • Modification ◦ Modified description of elliptical interpolation in 18.1.1 Overview.

Revision Date	Version	Description
		◦ Modified the elliptical interpolation operation procedure in Configuration Page.
February 2024	A14	• Modified description in (Optional) Disabling Modbus-RTU Slave. • Modified description in 10.1 Overview. • Modified instruction description in 13.1 Overview. • Modified cam description in 20.2.1 Overview. • Modified the cam curve in Cam Curve Settings. • Modified the procedure in Firmware Programming Through SD Cards.
December 2023	A13	• Addition ◦ Configuring 2-Channel Encoder Input Modules ◦ Fault Diagnosis ◦ 24.1 Project Comparison ◦ 21.6 OPC UA Offline Commissioning • Modification ◦ Added description for AI modules GL20-8ADV and GL20-8ADI in AO Modules. ◦ Added GL20-2SSI, GL20-0404ETP-5V, GL20-8ADV, and GL20-8ADI modules in 7.3.1 Overview. ◦ Optimized description about Modbus-TCP communication and added description about FINS TCP/UDP communication in the Ethernet Communication section. ◦ Added description about Easy devices in the PROFINET Communication section. ◦ Added "Contour Mode" in Creating Project Configuration.
August 2023	A12	Corrected minor errors.
May 2023	A11	• Added the BYTE variable and specific union data type. • Added the section "Encrypting Function Blocks (FB) or Functions (FC)". • Added the section "Folder" for program blocks, function blocks (FBs), and functions (FCs). • Added local extension modules GL20-2SCOM and GL20-2S485. • Added the GR10-EC-6SW branch module and GS20-ECT-8L module. • Added the H5U series PLC PROFINET communication function. • Corrected details of the earlier version.
	A10	• Added local modules.

Revision Date	Version	Description
January 2023		• Corrected details of the earlier version.
November 2022	A09	• Added the LiteST programming language. • Added the section"EtherNet/IP Slave Application Example" and the section "Exporting EDS Files". • Added the section "Free Protocol Cancellation (SerialSR Instruction)." • Added the section "Modifying Serial Port Parameters." • Added models of extension modules supported by Easy series host and running cases of all modules. • Added extension modules and use methods of GL20-RTU-ECT coupler. • Corrected details of the earlier version.
September 2022	A08	• Added the EIP function. • Added use methods of IP variables. • Added Easy series functions. • Corrected details of the earlier version.
August 2021	A07	• Added the method of setting the FB default. • Added graphic block functions of quickly increasing or decreasing label numbers and implementing incremental paste. • Added the function of packing and decompressing project archives. • Added the function of uploading and downloading Updown files. • Added the function of setting axis settings parameters by instructions. • Added the function of dragging the motion control axis. • Added the function of disabling the EtherCAT slave station by instructions. • Corrected details of the earlier version.
May 2021	A06	Kept the material versions consistent.
March 2021	A03	• Added the chapter "Electronic Cam." • Added the chapter "Bus Encoder Axes." • Added the chapter "Offline Commissioning." • Added the chapter "Memory Management." • Corrected details of the earlier version.
August 2020	A02	• Optimized the function of binding structure variables to soft elements. • Added the function of setting IP addresses using system variables. • Added the function of binding array variables to soft elements.

Revision Date	Version	Description
		• Corrected details of the earlier version.
May 2020	A01	• Added hard limit processing and the function of automatically starting the EtherCAT slave station, and updated the software UI. • Added the interpolation function and axis group configuration. • Added the Down file download and login functions. • Updated motion control axis fault codes.
February 2020	A00	First release

NOTE

This guide is applicable to AutoShop V4.4.0.0 and firmware V5.0.0.0 and later version.

2 Overview

2.1 Introduction

2.1.1 Product Introduction

The H5U series PLC, a new generation of small-sized PLC developed by Inovance, supports EtherCAT bus communication and features powerful motion control and distributed I/O control functions. It allows process encapsulation and reuse using the FB/FC function, and multi-layer network communication through the RS485, CAN, Ethernet, and EtherCAT interfaces.

Easy series small- and medium-sized PLCs are available in eight models, covering the demands of automation equipment requiring small footprint, multi-axis motion control, accurate temperature control, and easy networking.

2.1.2 Software Introduction

AutoShop is programming configuration software provided by Inovance for small-sized PLCs. It provides a friendly programming and commissioning environment and supports various and powerful communication and control functions. AutoShop supports various programming languages, such as the ladder diagram (LD), sequential function chart (SFC), and structured text (LiteST).

AutoShop has the following features:

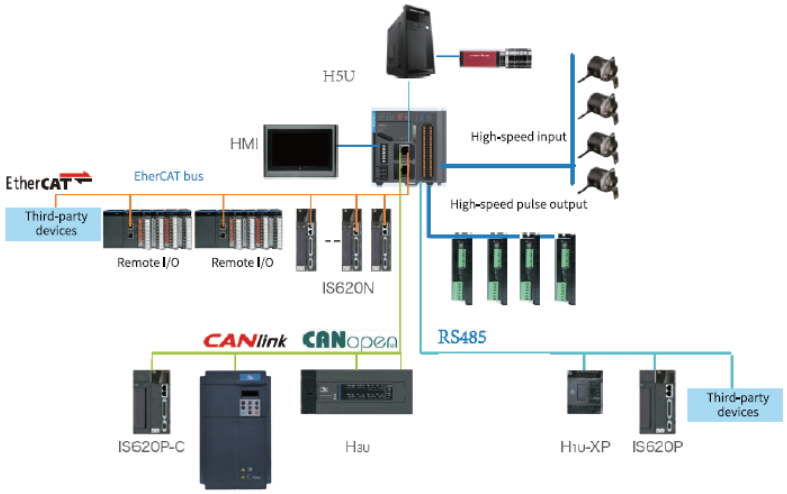
- Flexible communication mode: This feature allows AutoShop to communicate with PLC through COM, USB, and Ethernet. It enables the remote operation and remote collaborative commissioning functions for users, greatly facilitating users.
- Powerful network support: This feature enables the Modbus standard communication function based on configuration and supports CANopen communication configuration and Inovance CANlink communication configuration, greatly reducing difficulties and improving the working efficiency.
- Powerful motion control function: AutoShop has abundant motion control instructions and supports many functions such as the G codes, axis positioning, electronic cam, and flying shear/chasing shear.
- Convenient and diverse commissioning methods: AutoShop supports various functions such as motion profile graphics, monitoring, online modification, oscilloscope, and fault diagnosis, facilitating user commissioning and localization.
- Powerful intellectual property protection function: Functions such as upload password, download password, identifier, and upload forbidding help to effectively protect the intellectual property.

2.1.3 Networking Solution

H5U series networking scheme

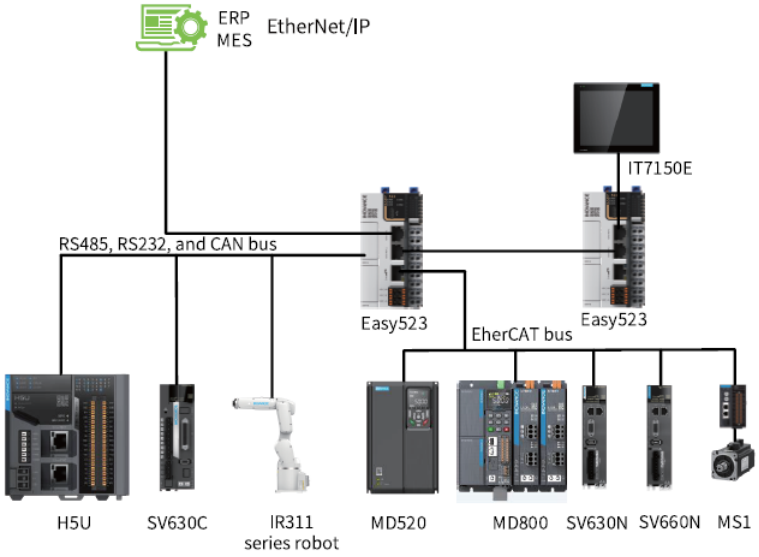
H5U provides the EtherCAT, CAN, Ethernet, and RS485 interfaces to implement multi-level network communication and meet requirements of multiple scenarios. It is equipped with four high-speed inputs, four medium-speed inputs, and four high-speed outputs, which can realize 4-axis pulse output and 4-axis encoder counting.

The following figure shows a typical application topology.

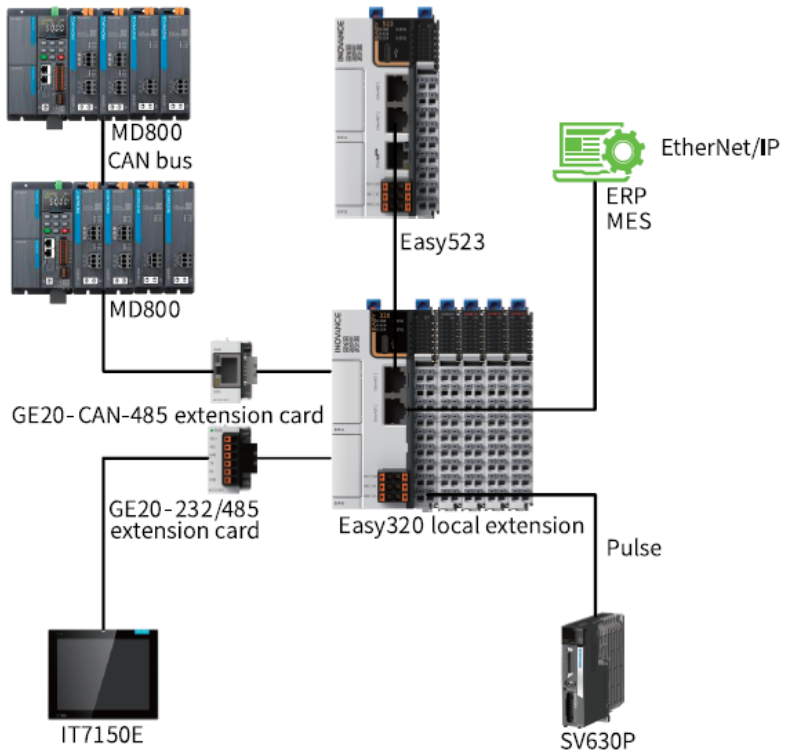


Easy series networking scheme

Two Easy series models are used as an example to introduce the typical application topology, as shown in the following figures.



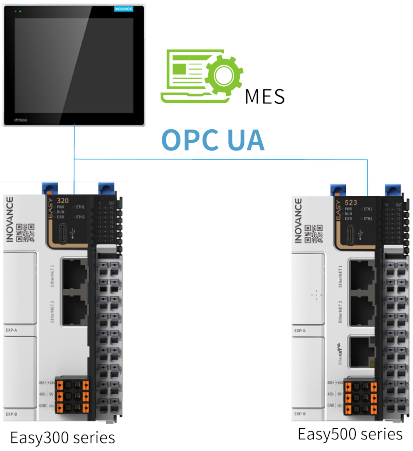
The following figure shows the Easy523 series application topology.



The following figure shows the Easy320 series application topology.

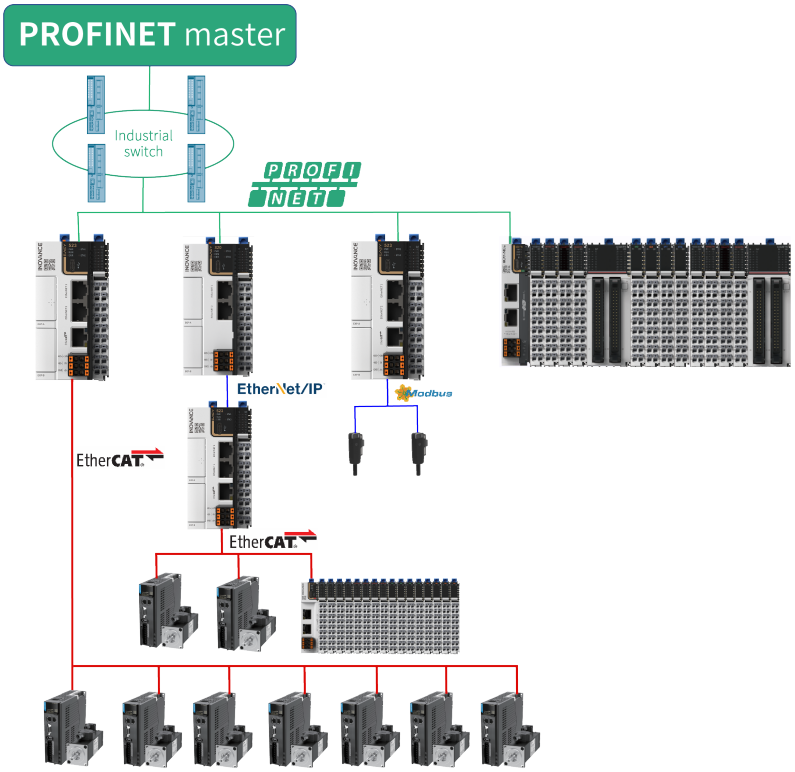
OPC UA topology

The MES system connects to the Easy320 and Easy52X PLCs through the OPC UA server, as shown in the following figure.



PROFINET topology

The PROFINET master connects to the Easy523 and Easy320 PLCs and the GL20 modules through PROFINER. It connects to other slaves, such as SV630N, SV660N, and other servos, as well as other EtherCAT-enabled modules through the EtherCAT port, as shown in the following figure.



2.2 Software Acquisition and Installation

2.2.1 Acquisition Pathway

AutoShop programming software is provided for free. To obtain it, visit www.inovance.com, search for the keyword, and download the guide.

> **NOTE**
>
> As Inovance constantly improves its products and documentation, it is recommended that you update your software versions and consult the latest reference materials when needed to facilitate your application design.

2.2.2 Requirements on the Installation Environment

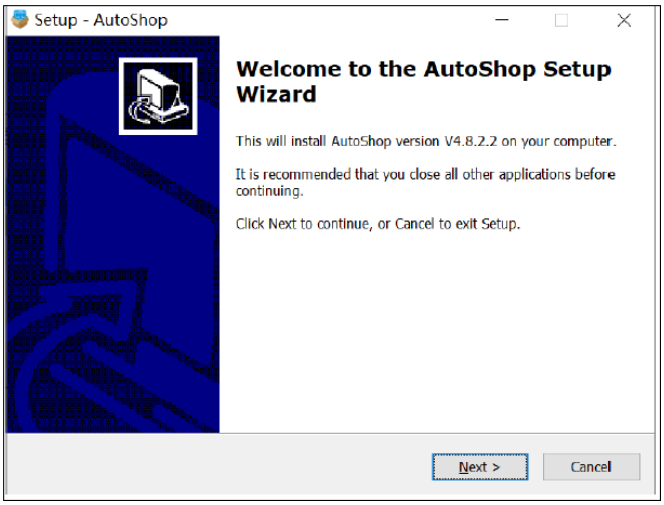
The following table lists the items required for a PC where AutoShop programming software is installed.

Item	Requirement
Operating system	Windows 7 or 10, 64-bit is recommended
Primary frequency of CPU	4 GHz or above
Memory	4 GB or above
Hard drive	More than 5 GB

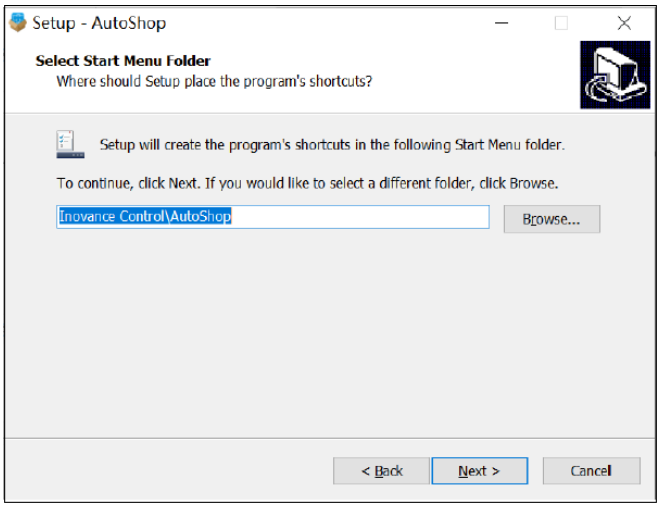
2.2.3 Installing the Software

To install AutoShop, you can download the software installation package of the latest version at www.inovance.com, and the installation UI of the latest version shall prevail. AutoShop V4.8.1.0 is installed on Windows 10 as an example.

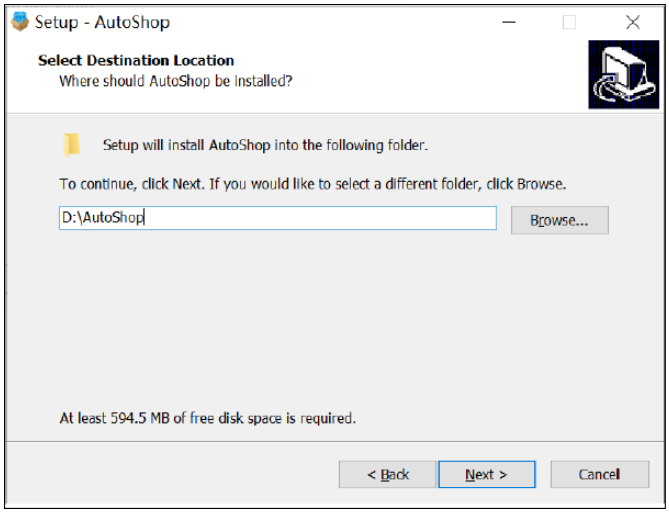
1. Decompress the “AutoShop V4.8.1.0 Setup.zip” package.
2. Double-click “AutoShop V4.8.1.0 Setup.exe”. In the “Select language” dialog box, select a language and click “OK”.
3. Click “Next”.



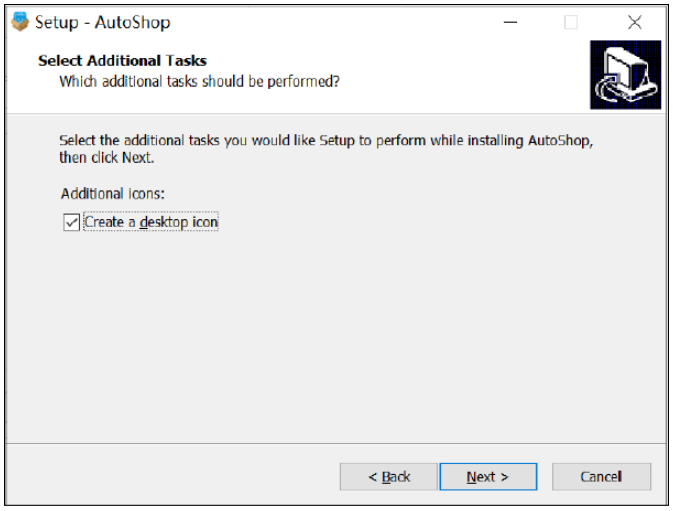
4. Click “Browse”. In the dialog box, select an installation path and click “Next”. In general, the default path is used.



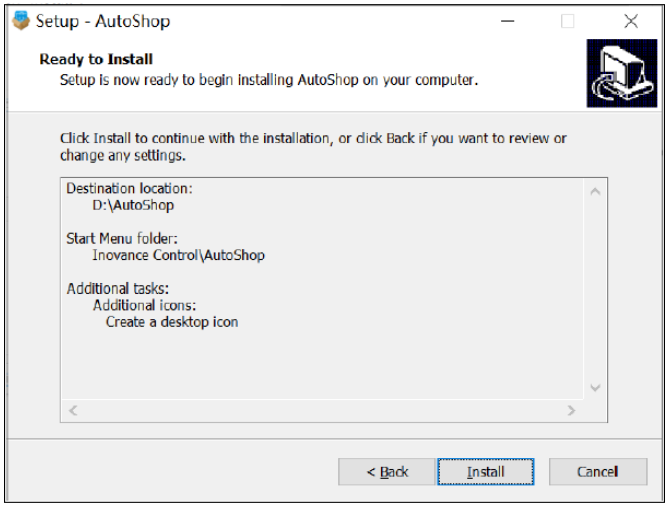
5. Click “Browse”. In the dialog box, select an installation path for the program shortcut and click “Next”. In general, the default path is used.



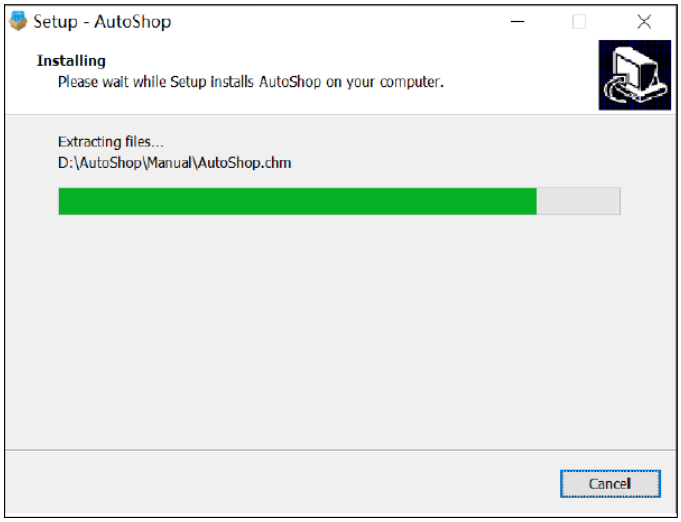
6. Click “Next”.



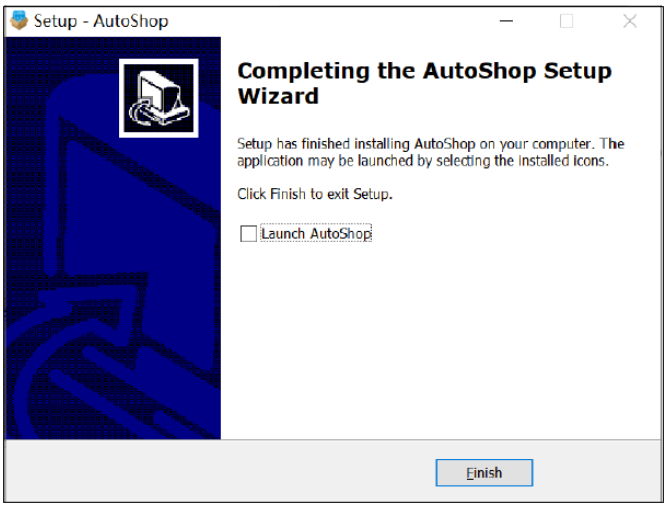
7. Click “Install”.



The following figure shows the installation progress.



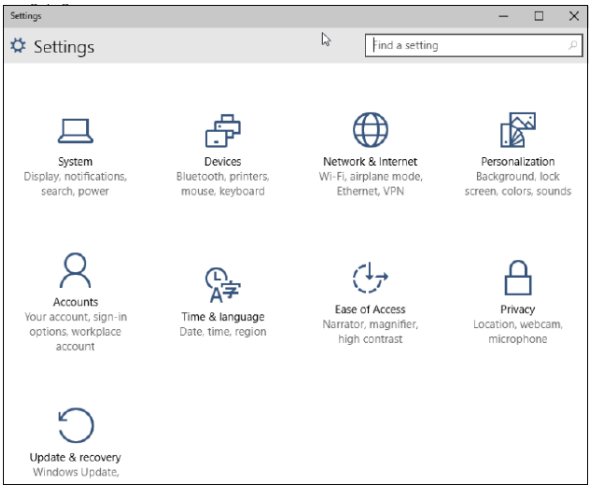
8. Click “OK”.



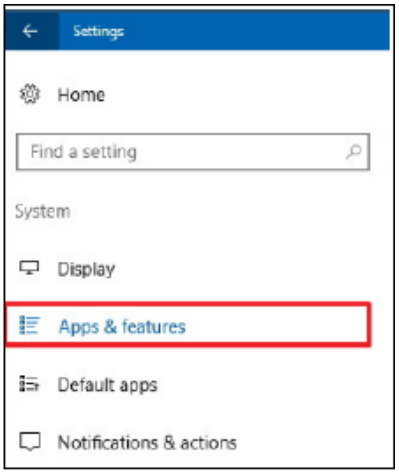
2.2.4 Uninstalling the Software

AutoShop V4.6.5.0 is uninstalled on Windows 10 as an example.

1. Click on the desktop. In the start menu displayed, click to access the "Settings" page.



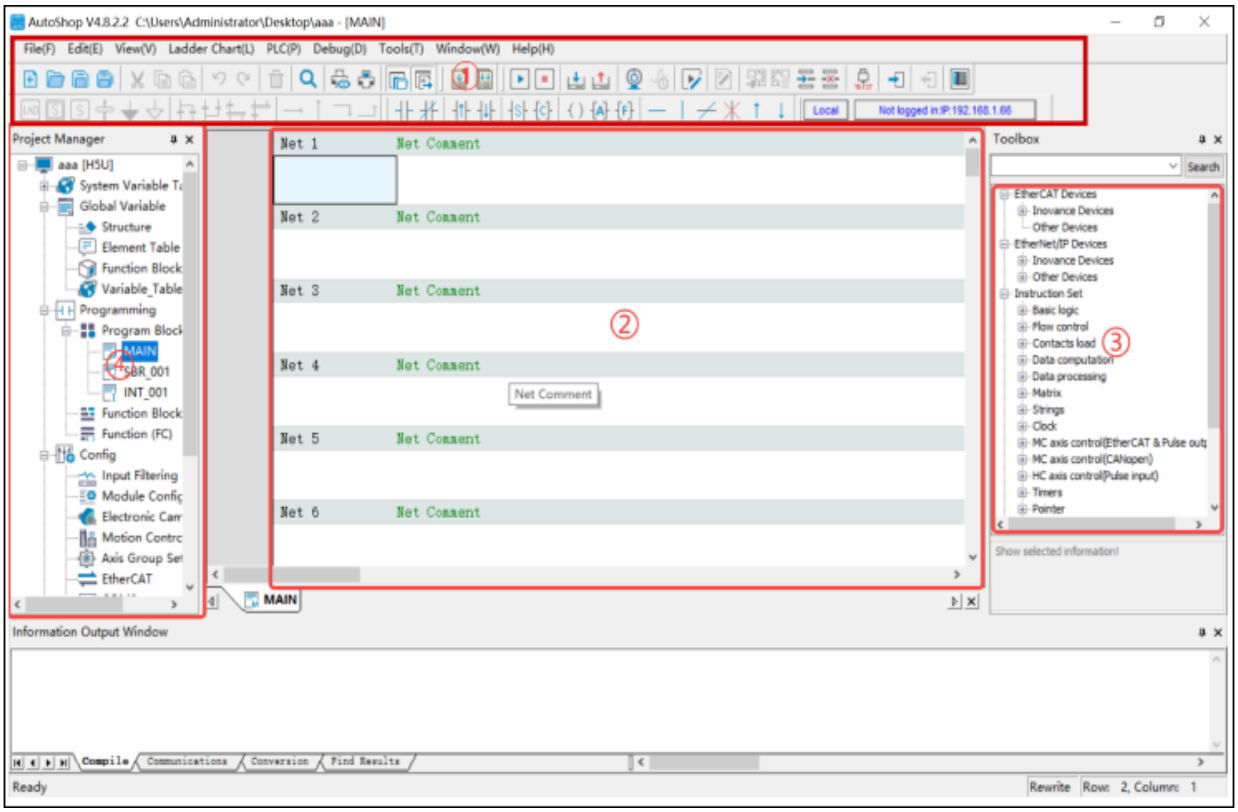
2. Click "Apply". The "Settings" dialog box is displayed.



3. On the "Apps & features" page, click "AutoShop Version Vx.x.x.x" and then "Uninstall". A prompt box is displayed, as shown in the following figure.
4. Click "Uninstall". The "AutoShop Uninstall Wizard" dialog box is displayed, as shown in the following figure.
5. Click "Yes". After the software is uninstalled, click OK.

2.3 Software Interface

The main screen of AutoShop programming software consists of the menu bar, toolbar, engineering management section, program editing section, toolbox, and status bar, as shown in the following figure.



No.	UI	Description
①	Menu bar	Including settings of programming, commissioning, and communication.

No.	UI	Description
②	Toolbar	Including shortcut modes of file management and programming commissioning tools.
③	Project managing area	Including parameter management, variable management, program management, and configuration management of the PLC project.
④	Program editing section	Compiling application programs for users.
⑤	Tool kit	Set of instructions supported by the slave station and select PLC in a project. Tool kits are classified into ladder diagram ones and LiteST ones and can be switched to each other. The two types only differ in instructions supported in the instruction sets.
⑥	Status bar	Including information such as the PLC status, fault status, PLC firmware version, scan cycle, CPU utilization, and memory utilization.

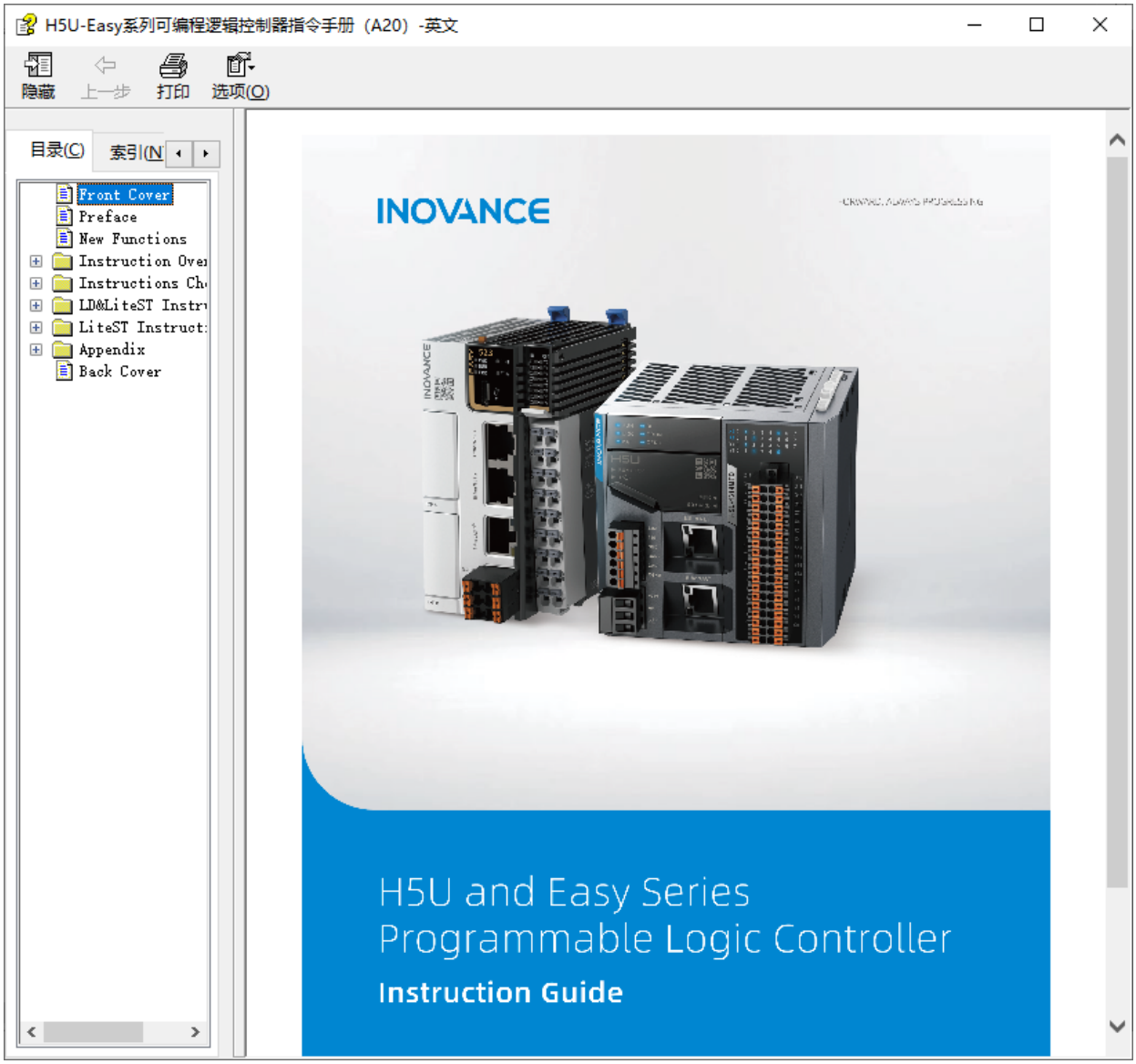
3 F1 Redirection

On the AutoShop interface, run an H5U or Easy project, and press F1. The H5U and Easy Series Programmable Logic Controller Instruction Guide is displayed, as shown in the following figure.

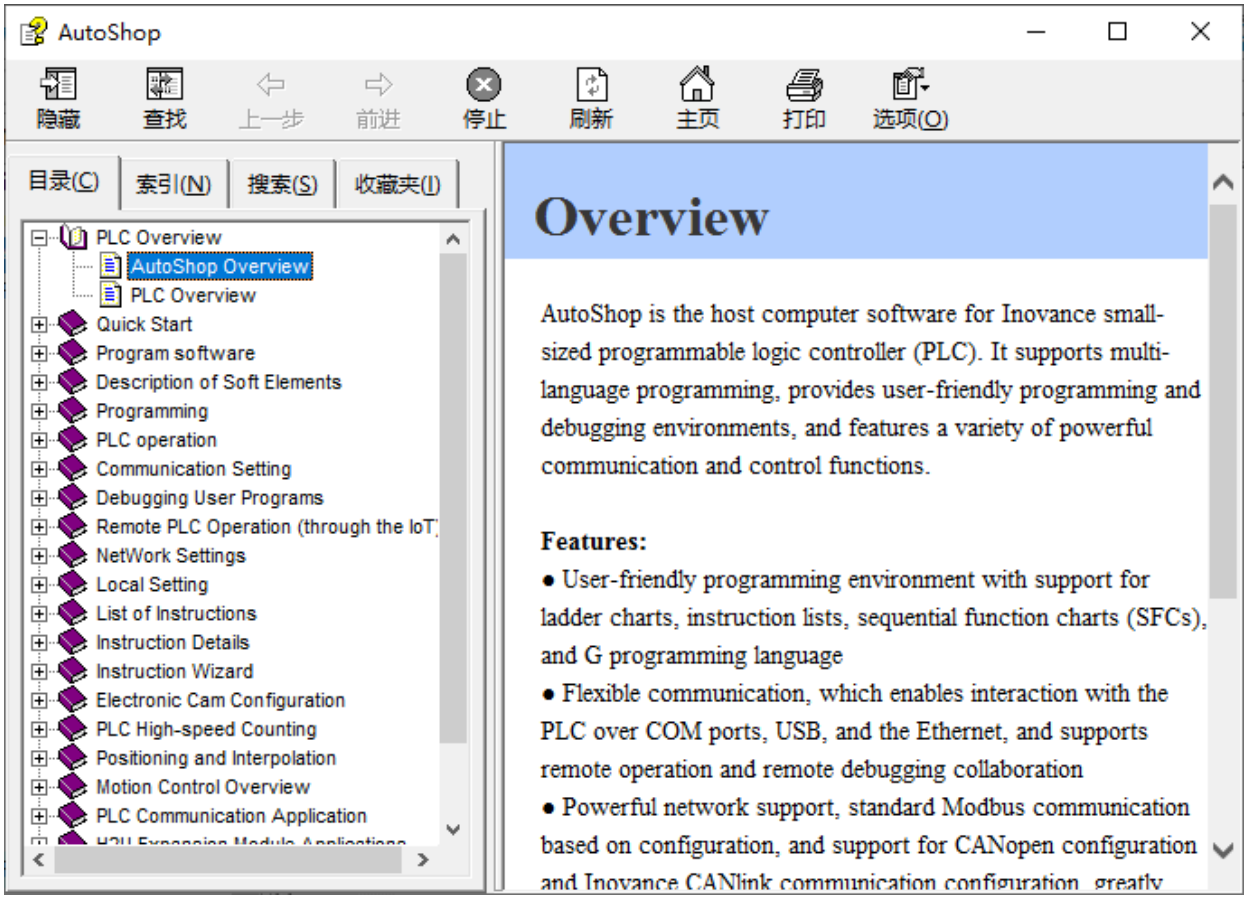
Run a project of the H3U or earlier series, and press F1. The AutoShop Help Guide is displayed, as shown in the following figure.

3.1 Online Help

On the AutoShop interface, run an H5U or Easy project, and press F1. The H5U and Easy Series Programmable Logic Controller Instruction Guide is displayed, as shown in the following figure.



Run a project of the H3U or earlier series, and press F1. The AutoShop Help Guide is displayed, as shown in the following figure.



3.2 Precise Redirection

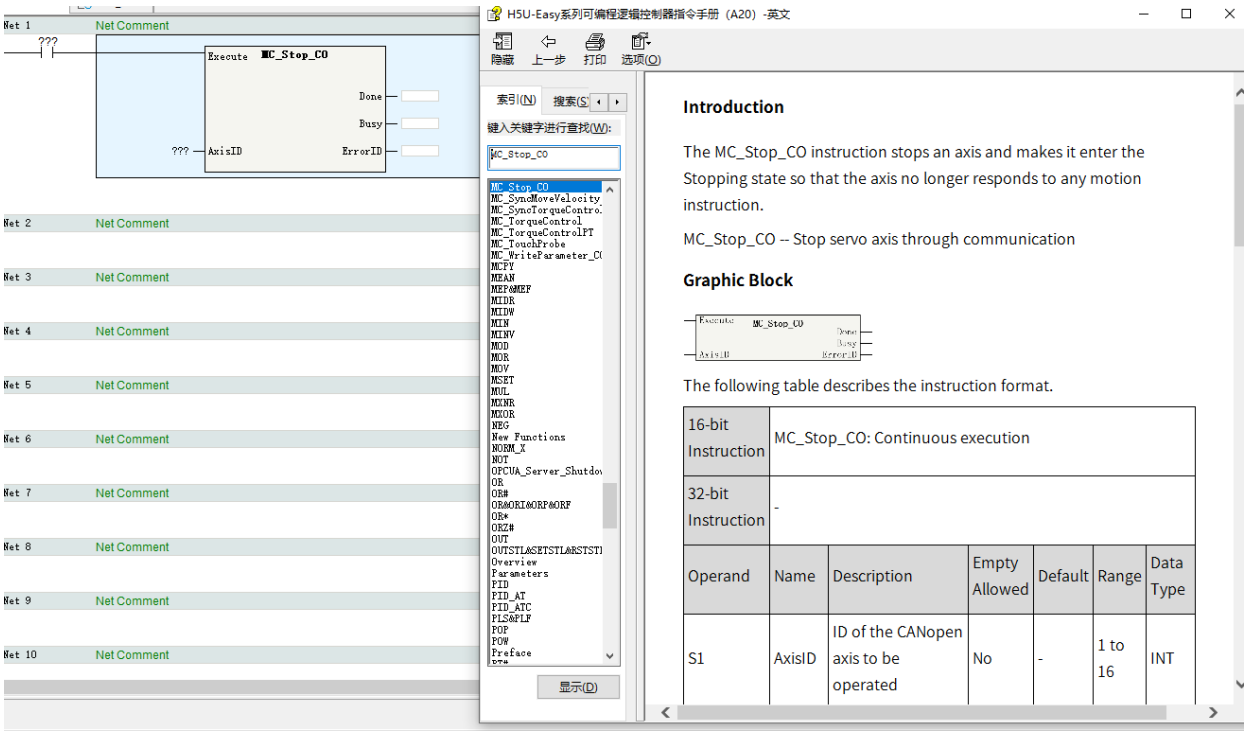
Preconditions

- PLC: Easy500 series, Easy300 series, H5U series, and H3U series
- Firmware version: V6.5.0.0 and later
- Software version: AutoShop V4.11.0.0 and later

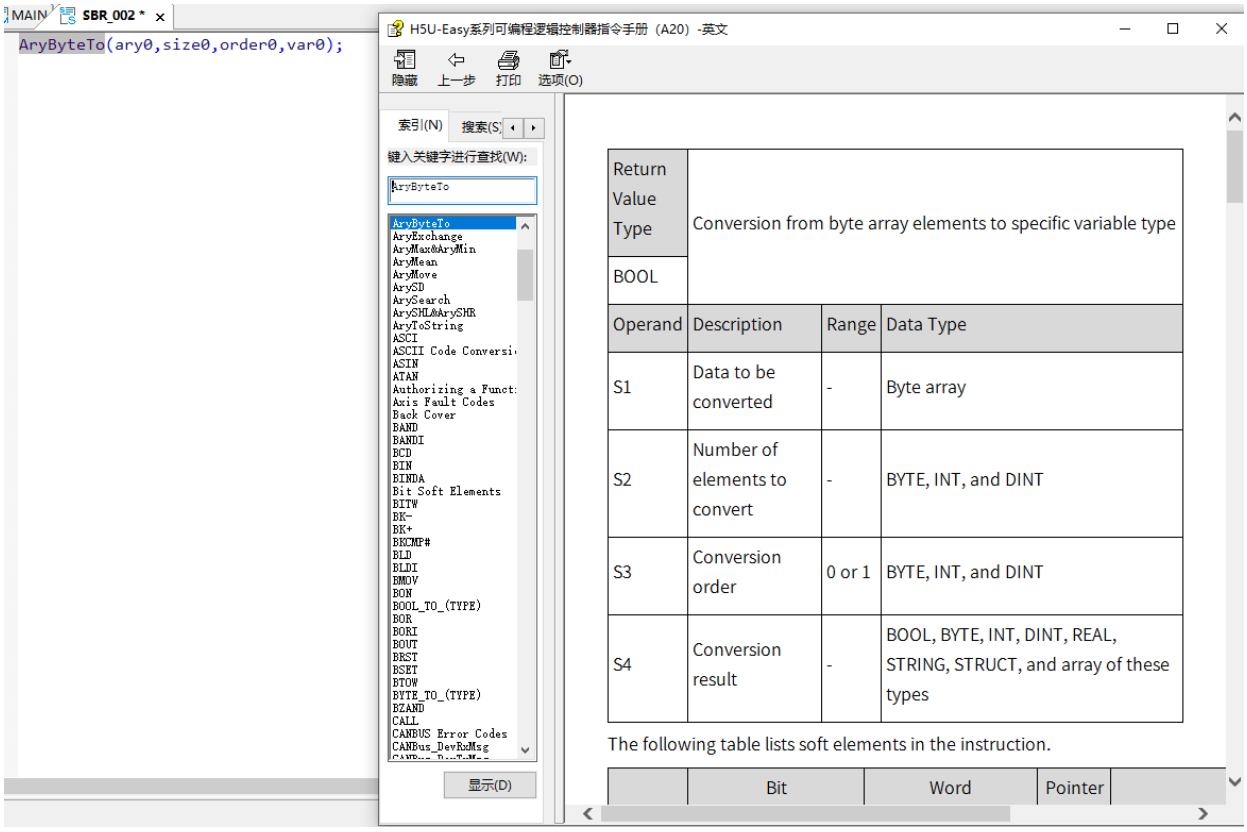
Introduction

In the AutoShop interface, open an H5U or Easy series project. In the LD or ST programming interface, or when an instruction is selected in the toolbox, press the F1 key to jump directly to the corresponding instruction description in the Help guide.

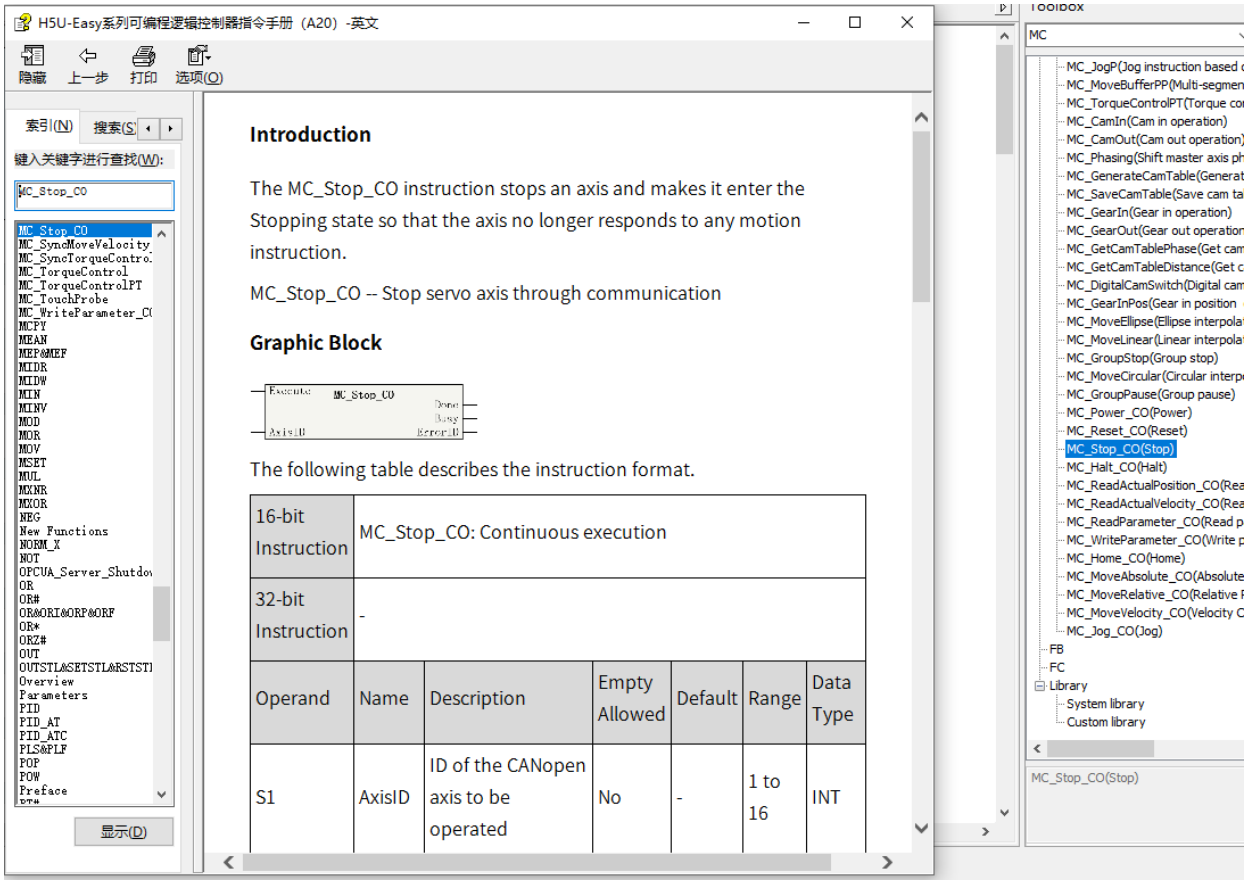
- In the LD programming interface, after entering an instruction and selecting it, press the F1 key to open the instruction guide and jump to the corresponding section.



- In the ST programming interface, after entering an instruction and selecting it, press the F1 key to open the instruction guide and jump to the corresponding section.



- In the Toolbox, after selecting an instruction, press the F1 key to open the instruction guide and jump to the corresponding section.



This precise redirection function allows you to directly navigate from an instruction name in the toolbox or within the LD/ST programming interfaces to the detailed introduction of that instruction in the guide.

4 Quick Start

4.1 Overview

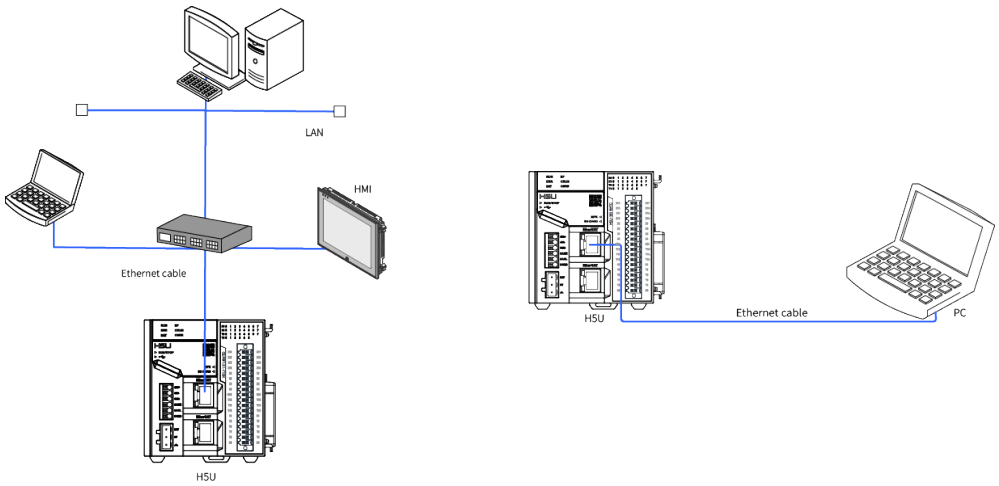
This section provides a simple programming example and general features used in the programming commissioning process for you to quickly master programming and commissioning. It is quite helpful to beginners.

4.2 Communication Connection

4.2.1 Overview

A PLC communicates with the PC where AutoShop is located through USB or Ethernet connection to upload, download, monitor, and commission the program.

For example, H5U communicates with the PC through Ethernet, achieving multi-point connection or point-to-point connection in hub-based, switch-based, or direction connection mode, as shown in the following figure.



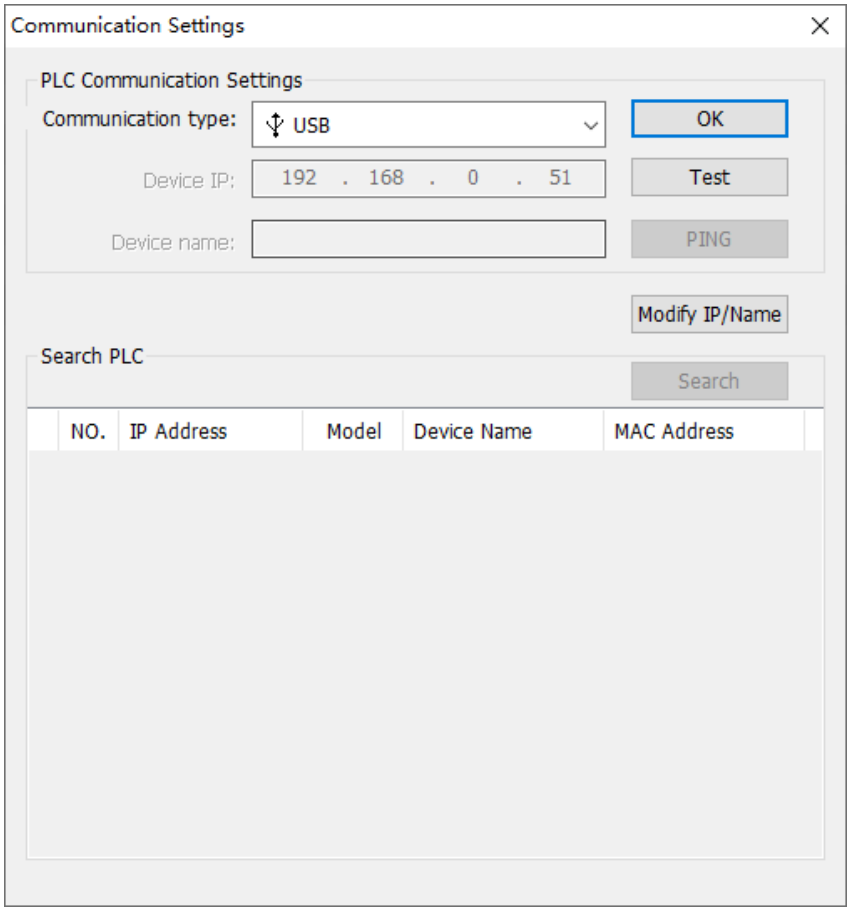
4.2.2 Ethernet Connection

When connecting a PC to an PLC through Ethernet connection, you may need to connect to the target PLC and change the PLC IP address and device name.

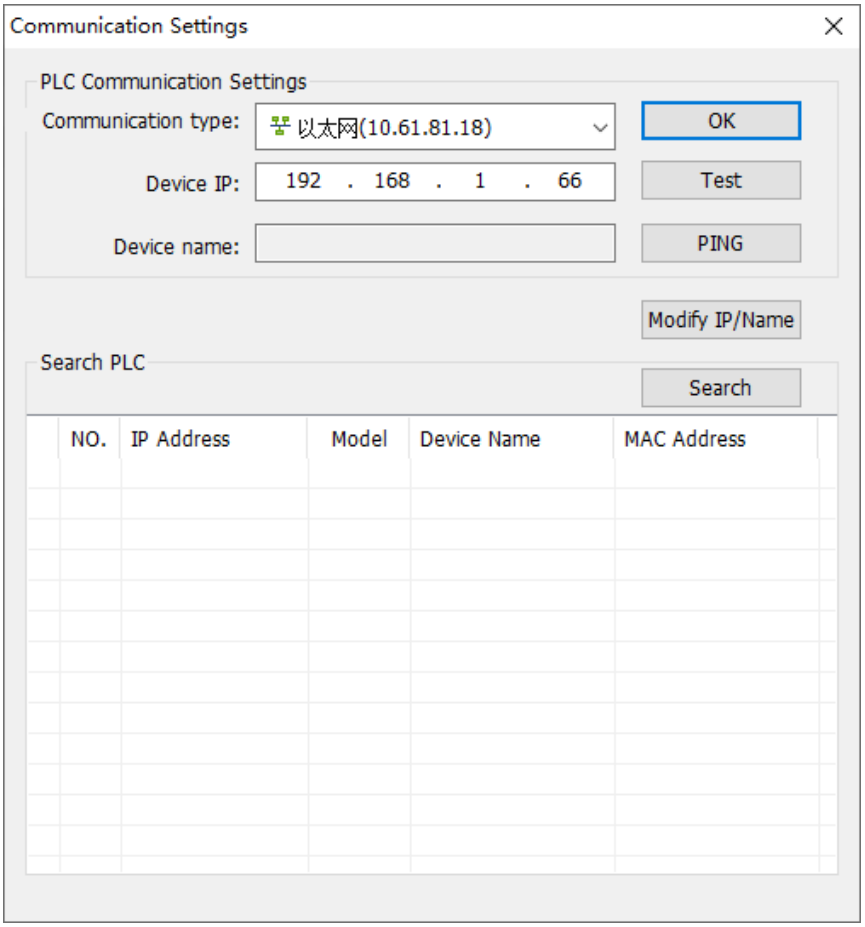
Connecting to the target PLC

To connect a PC to an PLC through Ethernet connection, you need to select the PLC with the specified IP address.

- If the IP address of the PLC is provided, configure the communication type and IP address of the PLC and then connect to the target PLC.
  1. Connect the PC to the PLC using a network cable.
  2. Double-click on the desktop of the PC to start the AutoShop programming software.
  3. Choose "Tools" > "Communication Settings" in the menu bar or click in the toolbar. The "Communication Settings" dialog box is displayed.



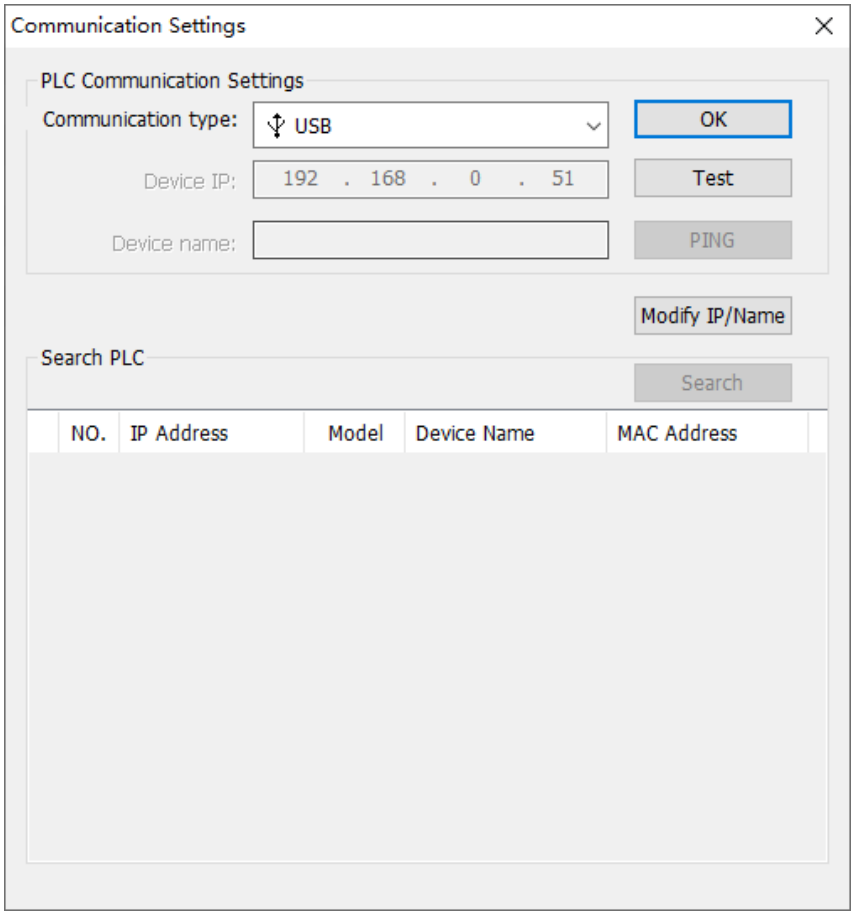
4. Set parameters in the "Communication Settings" dialog box.
   - Set "Communication type" to "Ethernet".
   - Set "Device IP" to the actual IP address of the PLC.



> **NOTE**
>
> To test the network connection between the PC and the PLC, click "PING".

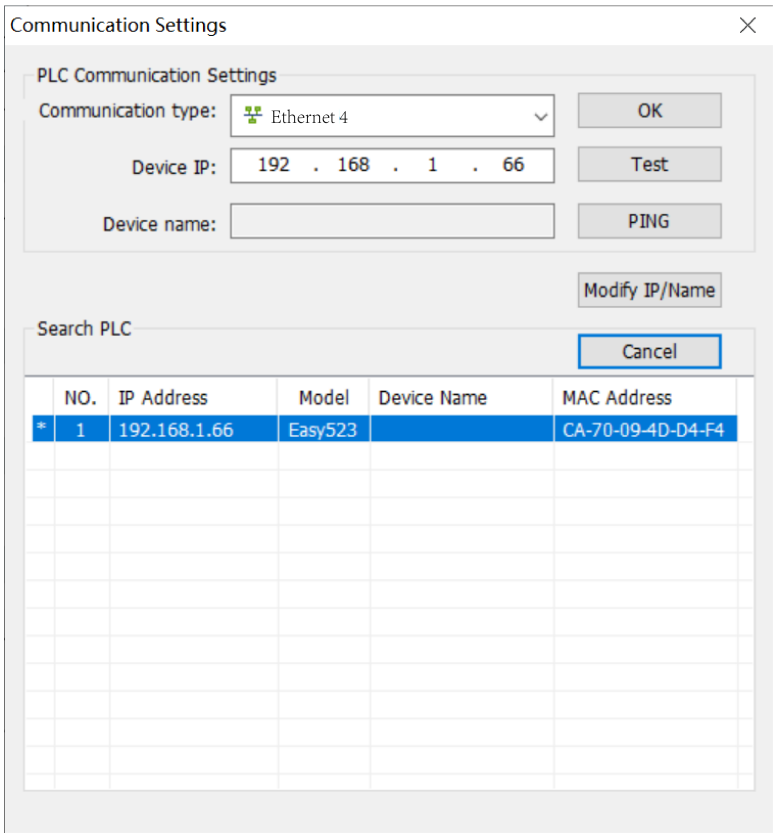
5. Click "Test" to check whether the target PLC is connected successfully.
   - H5U series: When LEDs of the connected H5U display 0 alternatively, the H5U is connected successfully.
   - Easy series: When the RUN indicator of the connected Easy flashes, the Easy is connected successfully.
6. Click "OK" to connect to the target PLC.
   Then, the IP address of the PLC is displayed in the toolbar and you can download, upload, monitor, or modify the PLC program online.

- You can search for the target PLC.
  1. Connect the PC to the PLC using a network cable.
  2. Double-click on the desktop of the PC to start the AutoShop programming software.
  3. Choose "Tools" > "Communication Settings" in the menu bar or click in the toolbar. The "Communication Settings" dialog box is displayed.



  4. Set "Communication type" to "Ethernet", and click "Search" to search for the PLC connected in the LAN.
     - In switch-based connection mode, you can only search for the PLC in the same segment of the PC.

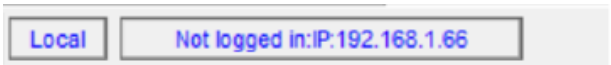
- In direct connection mode, you can search for the PLC in the same segment or a different segment of the PC.



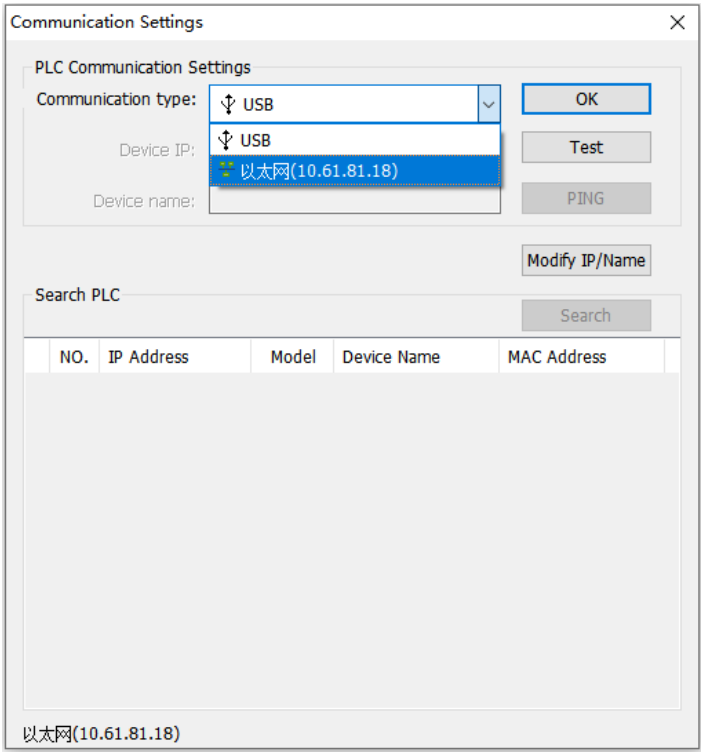
5. Select the IP address row in the search result and click "Test" to check whether the PLC is connected successfully.
   - H5U: When LEDs of the connected H5U display 0 alternatively, the H5U is connected successfully.
   - Easy: When the RUN indicator of the connected Easy flashes, the Easy is connected successfully.

> **NOTE**
>
> To test the network connection between the PC and the PLC, click "PING".

6. Click "OK" to connect to the target PLC.
   Then, the IP address of the PLC is displayed in the toolbar and you can download, upload, monitor, or modify the PLC program online.



- In case of multi-segment IP addresses, you can manually switch the Ethernet IP address of the current client accessing the PLC.
  1. The local PC has set with multiple IP addresses on the "Advanced TCP/IP" setting page of the Ethernet.
  2. Connect the PC to the PLC using a network cable.
  3. Double-click on the desktop of the PC to start the AutoShop programming software.
  4. Choose "Tools" > "Communication Settings" in the menu bar or click in the toolbar. The "Communication Settings" dialog box is displayed.



5. In the "Communication Settings" dialog box, select the Ethernet IP address to be connected to the PLC.
6. Connect to the target PLC by referring to steps 4 to 6 of "if the IP address of the PLC is provided" or "if the IP address of the PLC is not provided".

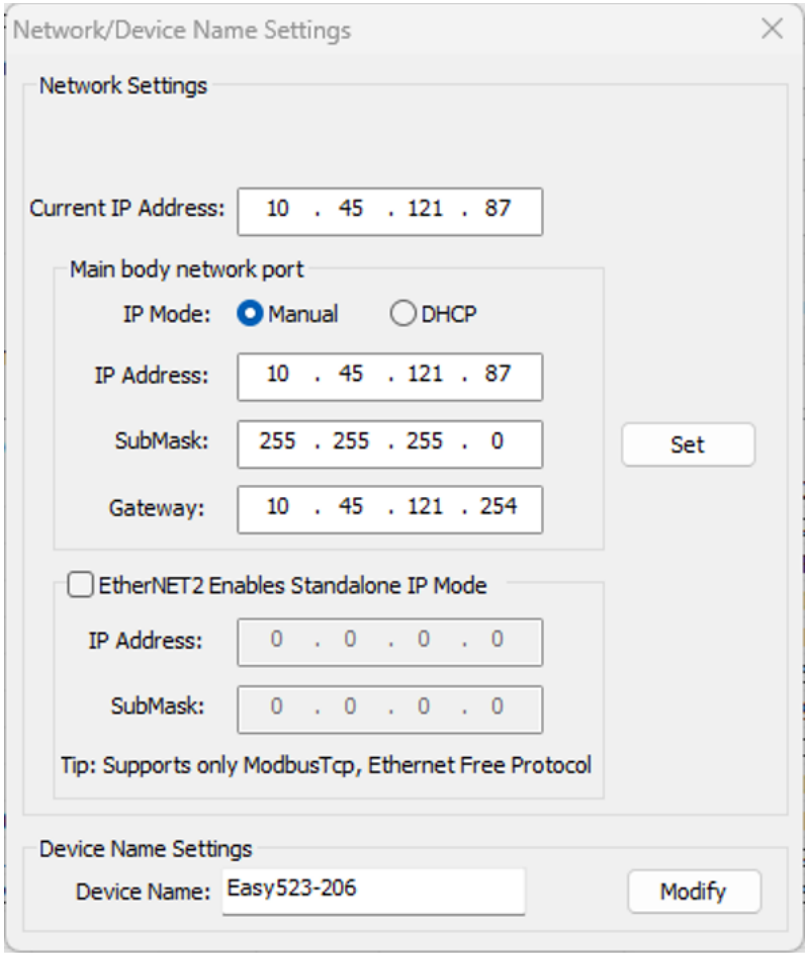
Changing the PLC IP address and device name

NOTE

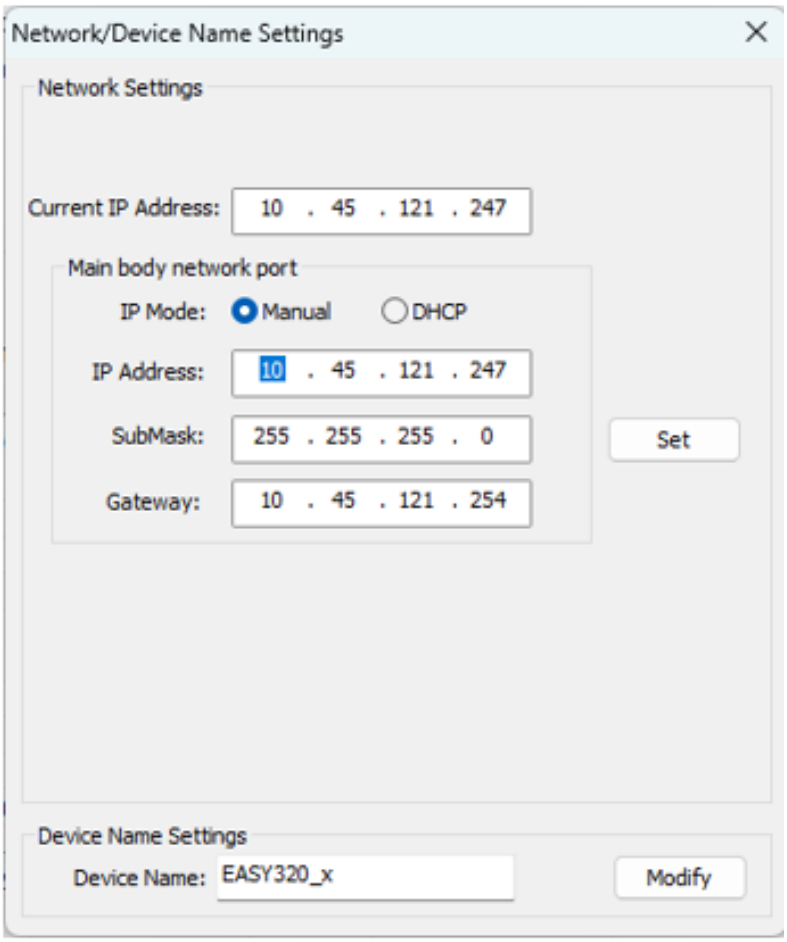
- The Easy52X and Easy320 models, software of AutoShop V4.11.0.0 and later and firmware of V6.5.0.0 and later support the DHCP function.
- The Easy52X model, software of AutoShop V4.11.0.0 and later, and firmware of V6.5.0.0 and later support the dual-IP function.

You can change the IP address or the PLC as needed or change the device name of the PLC for differentiation.

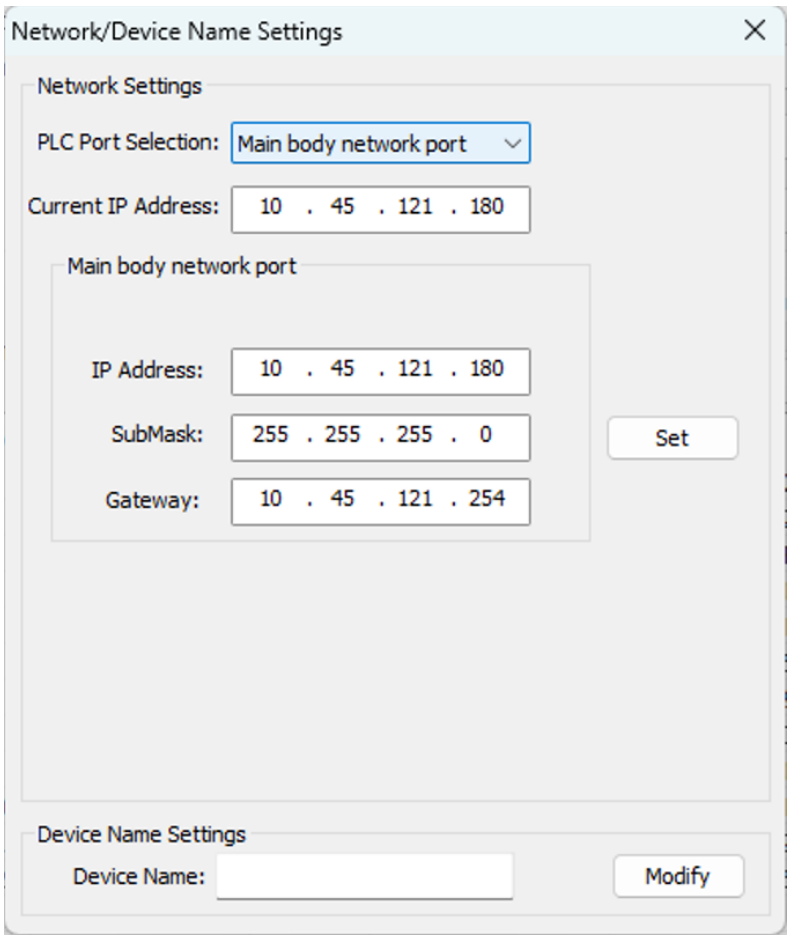
1. After connecting to the target PLC, click "Modify IP" or "Modify device name" in the "Communication Settings" dialog box. The "Modify IP" or "Modify device name" dialog box is displayed. The following figures show the interfaces of different models.
   - The interface of the Easy52X series is shown in the following figure.



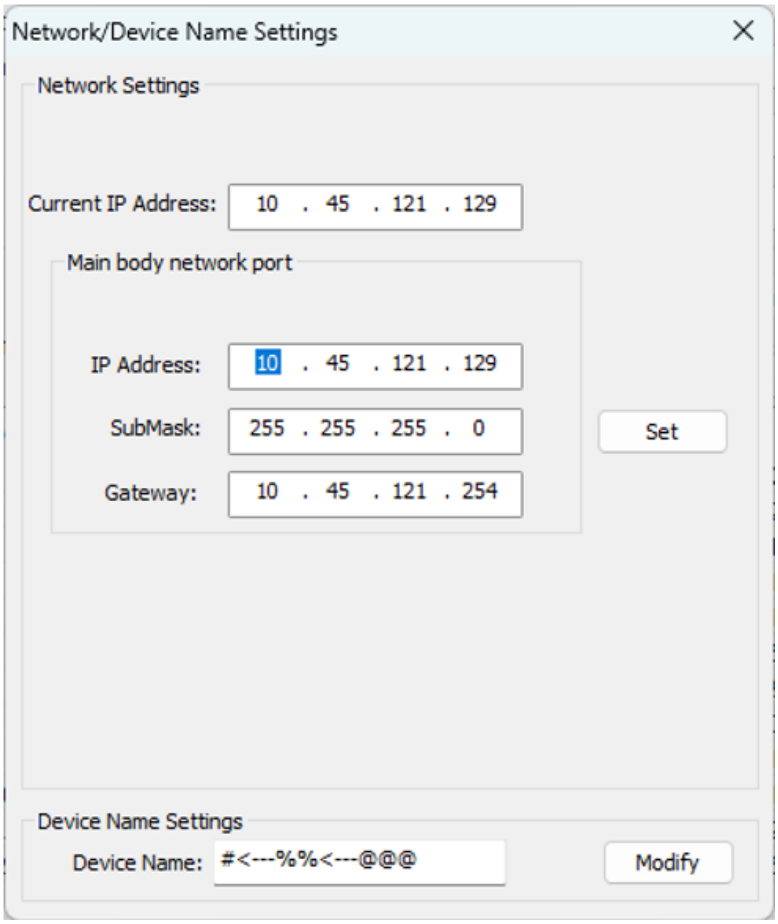
- The interface of the Easy320 model is shown in the following figure.



- The interface of the Easy511 and Easy321 series is shown in the following figure.

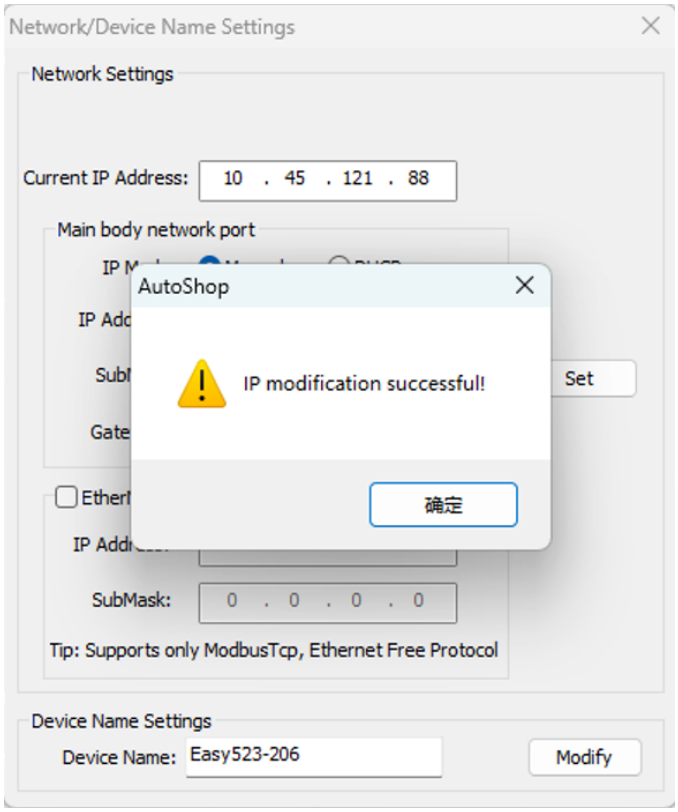


- The interface of other series is shown in the following figure.

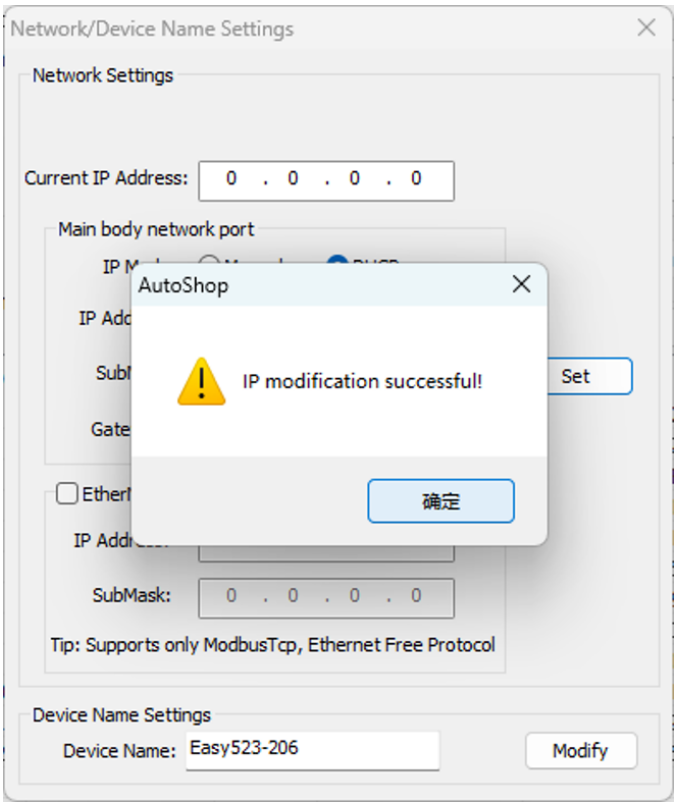


2. Set the main network port, that is, EtherNET1. In the "Main body network port" section, select "Manual" or "DHCP" for "IP Mode".
   - Manual

     Fill in a new IP address, subnet mask, and default gateway, and then click "Set". After the modification, the system prompts "IP modification successful!". Click "OK".



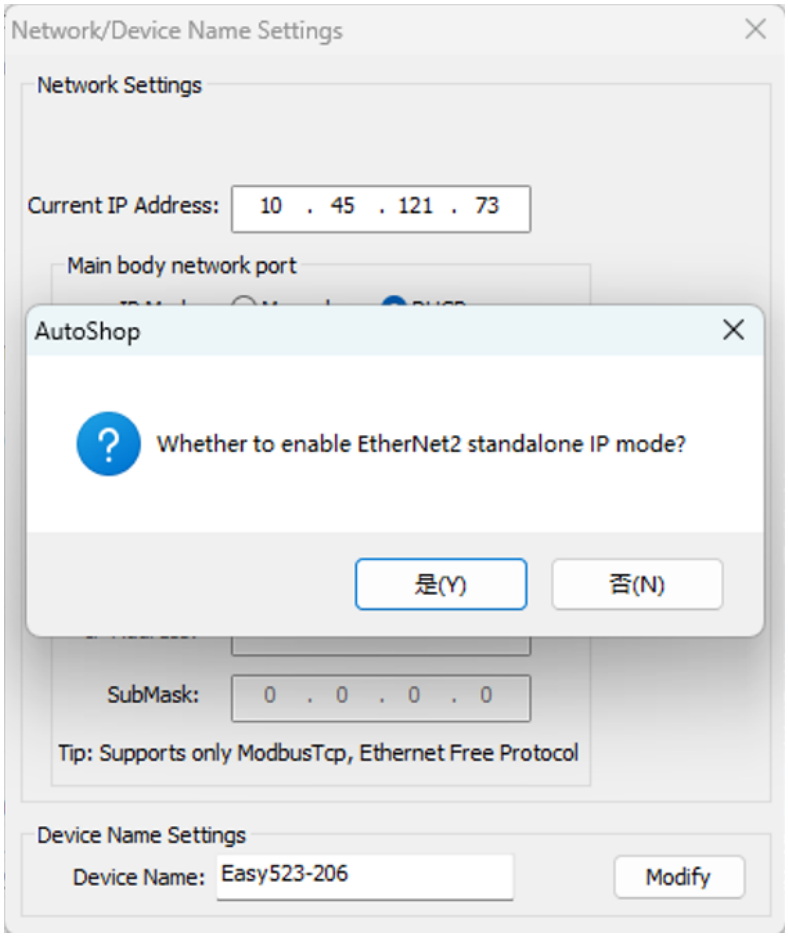
- DHCP
  Click "Set" and the setting takes effect immediately. After the setting, the system prompts "IP modification successful!". Click "OK".



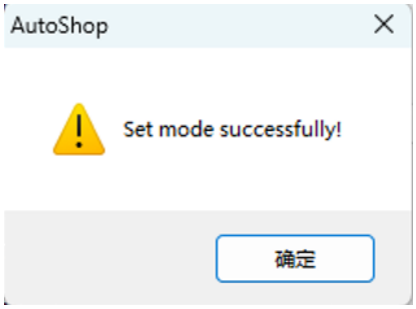
NOTE

- When the new PCB software (V3.0.0.0) is used with an old AutoShop background (V4.0.0.0 or earlier version), the following situation will occur: When you modify the IP in the communication settings, even if the software prompts success, the IP may not be successfully modified.
- When PLC does not support automatic IP address acquisition, it prompts "PLC does not support this function of dynamically acquiring IP!".
- The lease time of DHCP server should be configured as 86,400 seconds (corresponding to the 24-hour standard cycle defined in RFC 2131).

3. Set the dual IP mode of PLC.
   - Enable the PLC dual IP mode. Select "Enable EtherNet2 standalone IP mode". In the "Whether to enable EtherNet2 standalone IP mode?" dialog box displayed, click "Yes" to enable the EtherNet2 standalone IP mode.



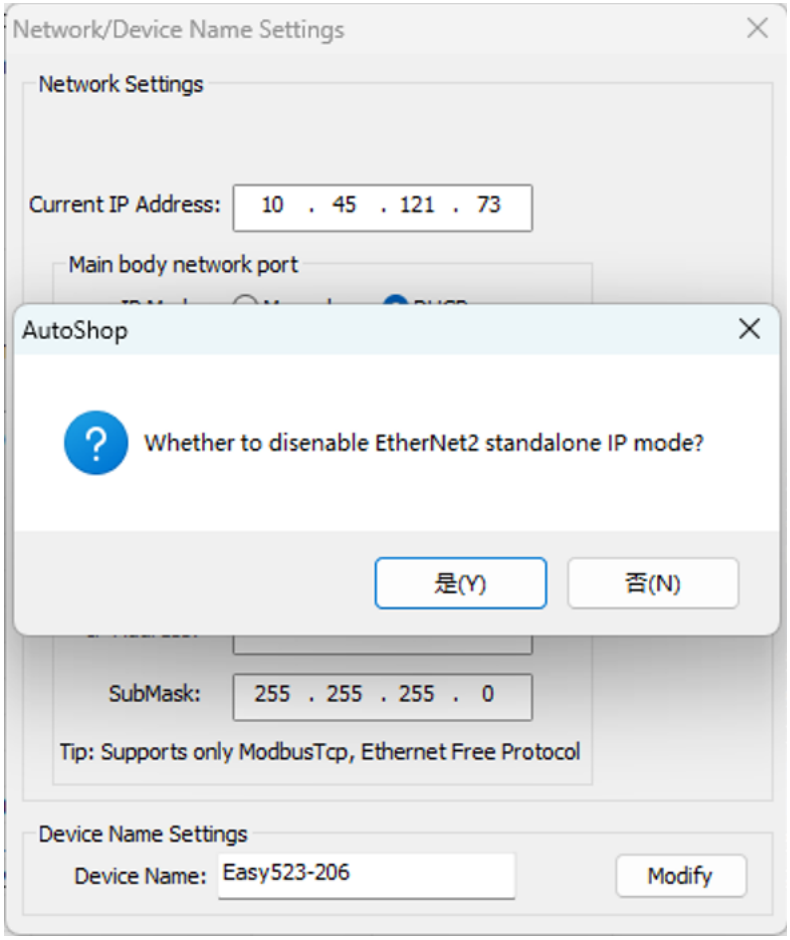
Then, "Set mode successfully!" is displayed.



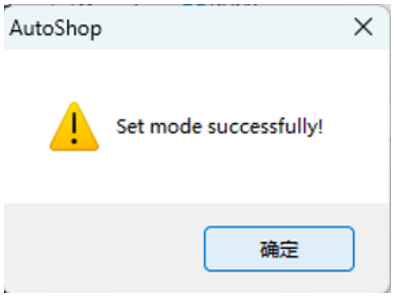
NOTE

- After the dual IP mode is enabled, the EtherNET2 port only supports Modbus-TCP and Ethernet free protocol, but temporarily does not support search, scanning, firmware upgrade, EtherNet/IP, OPC UA and other protocols.
- EtherNET2 IP cannot be in the same network segment as EtherNET1.

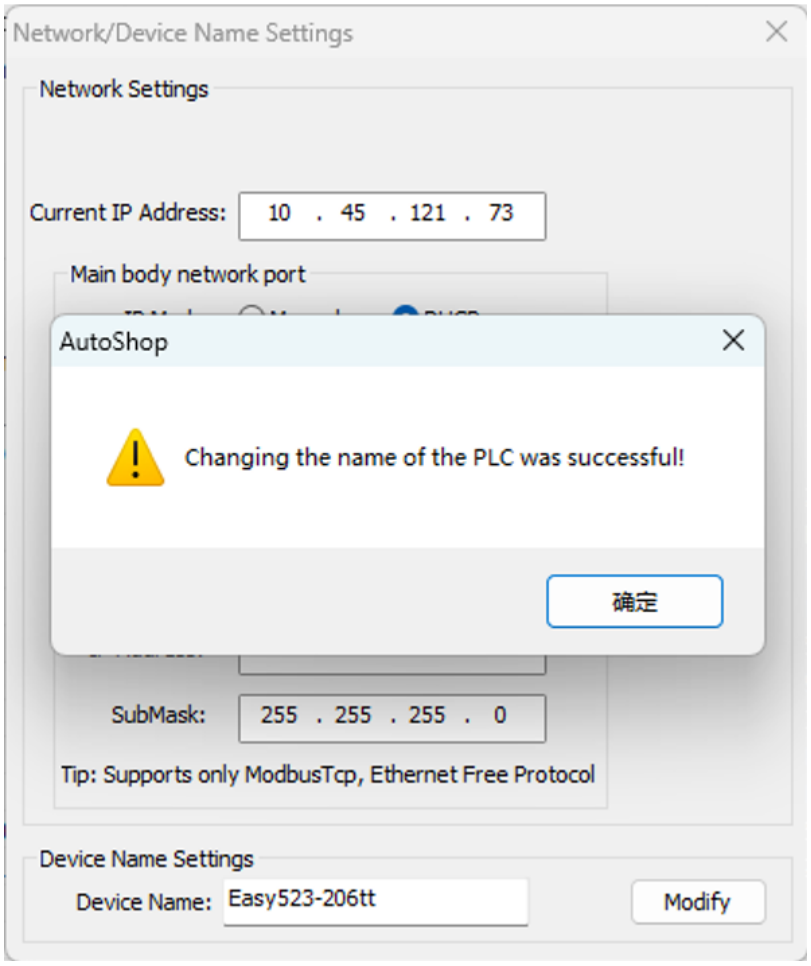
- Disable the PLC dual IP mode. Deselect "Enable EtherNet2 standalone IP mode". In the "Whether to disable EtherNet2 standalone IP mode?" dialog box displayed, click "Yes" to enable the EtherNet2 standalone IP mode.



Then, "Set mode successfully!" is displayed.



4. In the "Device Name" section, set "Device Name" as needed and click "Modify device name". In the dialog box displayed, click "OK".

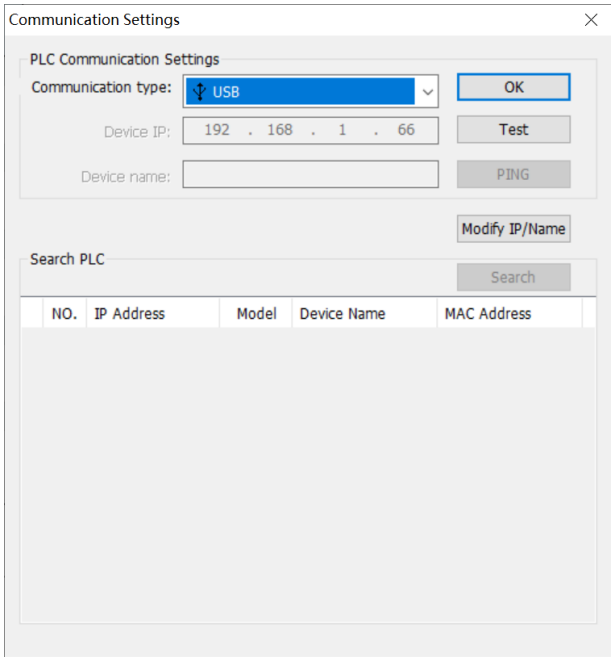


4.2.3 USB Connection

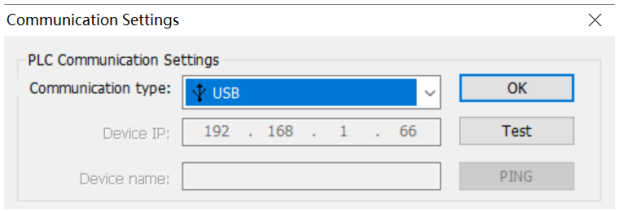
When connecting a PC to an PLC through USB connection, you may need to connect to the target PLC and change the PLC IP address and device name.

Connecting to the target PLC

1. Connect the PC to the PLC using a USB cable.
2. Double-click on the desktop of the PC to start the AutoShop programming software.
3. Choose "Tools" > "Communication Settings" in the menu bar or click in the toolbar. The "Communication Settings" dialog box is displayed.



4. Set "Communication type" to "USB" and click "Test" to check whether the target PLC is connected successfully.
   - H5U: When LEDs of the connected H5U display 0 alternatively, the H5U is connected successfully.
   - Easy: When the RUN indicator of the connected Easy flashes, the Easy is connected successfully.



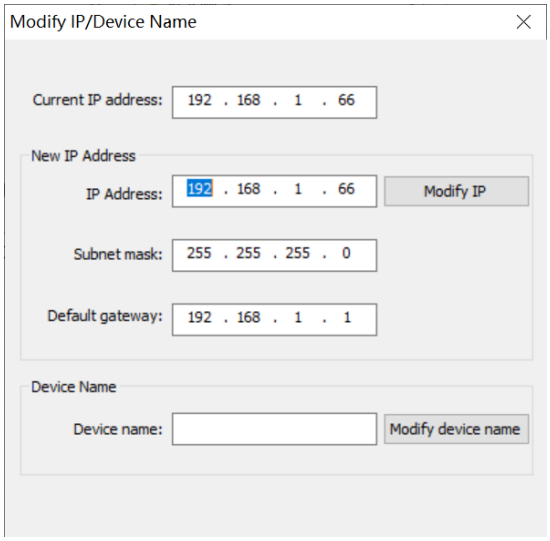
> **NOTE**
>
> If AutoShop V4.0.0.0 is used with an old PCB software (V3.0.0.0 or earlier versions), clicking "Test" will not initiate connection to the H5U device. In this case, upgrade the PLC software.

5. Click "OK" to connect to the target PLC.
   Then, you can download, upload, monitor, or modify the PLC program online.

Changing the PLC IP address and device name

You can change the IP address or the PLC as needed or change the device name of the PLC for differentiation.

1. After connecting to the target PLC, click "Modify IP" or "Modify device name" in the "Communication Settings" dialog box. The "Modify IP" or "Modify device name" dialog box is displayed.



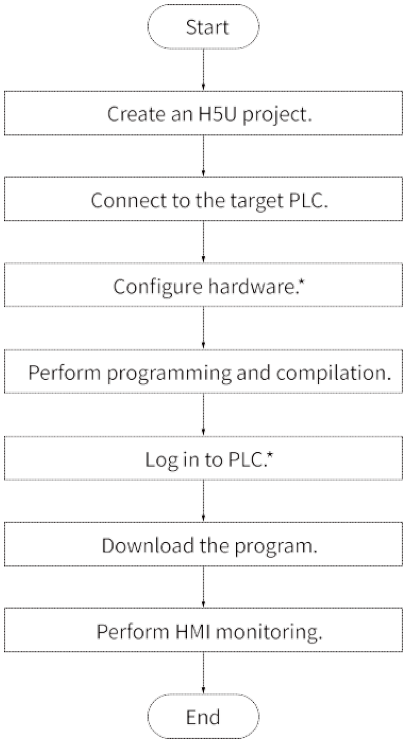
2. Set "IP Address", "Subnet mask", and "Default gateway" in the "New IP Address" section, and click "Modify IP". In the dialog box displayed, click "OK". The new IP address takes effect after the PLC is powered on again.

> **NOTE**
>
> When the new PCB software (V3.0.0.0) is used with an old AutoShop background (V4.0.0.0 or earlier version), the following situation will occur:
>
> When you modify the IP in the communication settings, even if the software prompts success, the IP may not be successfully modified.

3. In the "Device Name" section, set "Device Name" as needed and click "Modify device name". In the dialog box displayed, click "OK".

4.3 Programming Flowchart

A typical user program flowchart of the H5U series PLC is compiled and commissioned as an example, as shown in the following figure.



NOTE

* If only the H5U host is used, skip the hardware configuration step. If it is connected to a PLC without a login password, skip the “Login PLC” step.

4.4 Programming Example

4.4.1 Requirements

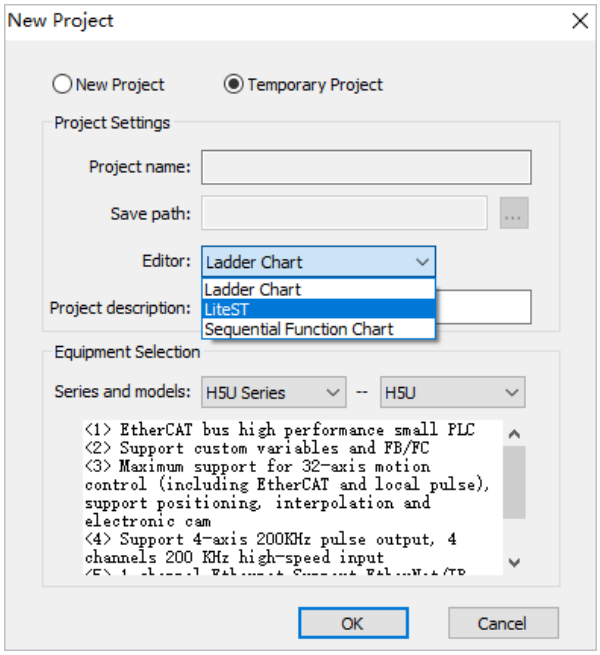
Compile and commission the next marquee program based on the following requirements:

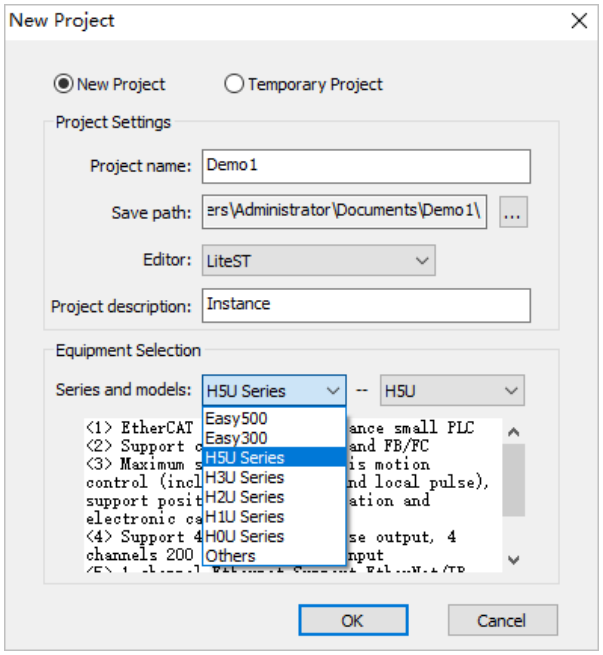
Using a 16-output module, PLC program shifts one bit rightwards from Y20 every one second. When the output reaches Y27, the PLC program returns the output to Y20 for a new cycle, which is monitored through HMI.

4.4.2 Creating or Opening a Project

When you create or open a project, the project main screen is displayed.

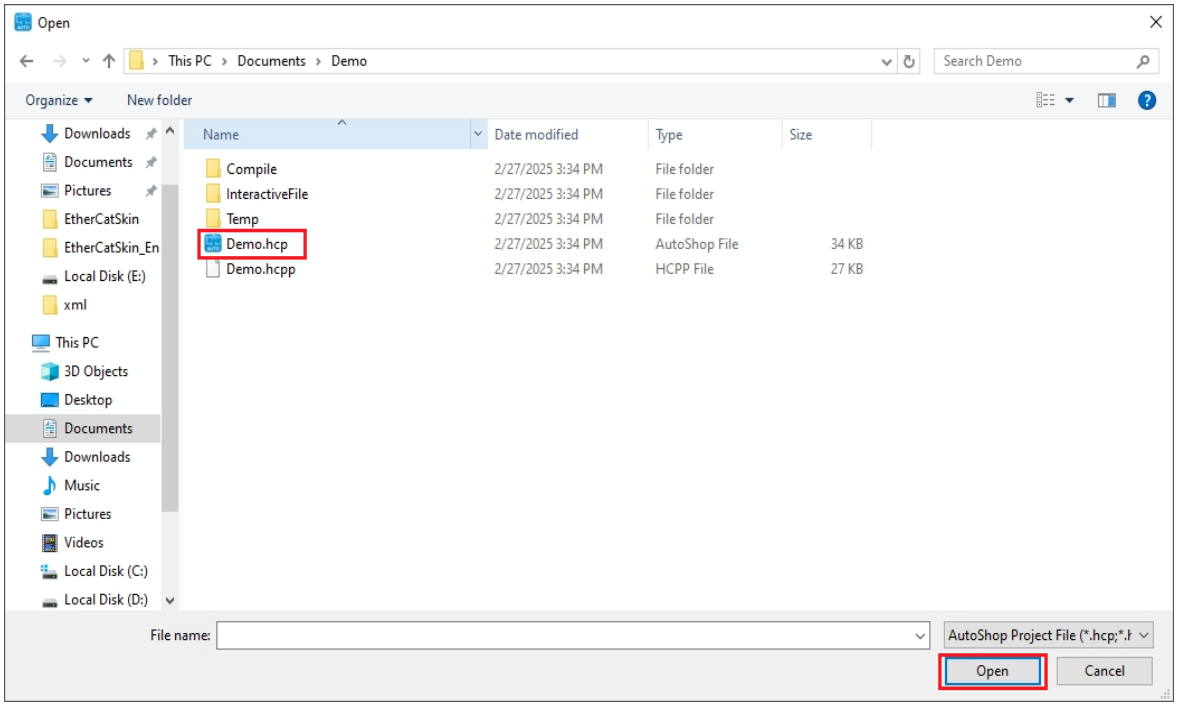
- Creating a project
  1. Double-click the AutoShop shortcut icon on the desktop to run AutoShop.
  2. Choose "File" > "New Project" in the menu bar or click in the toolbar. In the dialog box displayed, choose an editor type from the "Editor" drop-down list, and set "Series and models" to "H5U".





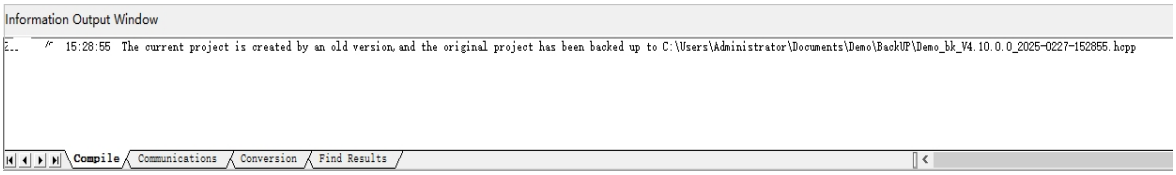
3. Set "Project name" and "Save path", and click "OK" to create a project. Then, the main screen of the project is displayed.

- Opening a project
  1. Double-click the AutoShop shortcut icon on the desktop to run AutoShop.
  2. Choose "File" > "Open Project" in the menu bar or click in the toolbar. In the dialog box displayed, select a saved project file and click "Open". Then, the main screen of the project is displayed.



See the following tips when creating and saving a project in different versions of AutoShop.

- You cannot open a project that has been created and saved in AutoShop of a later version in an earlier version.
- When you open a project that has been created and saved in AutoShop of an earlier version in a later version, the current project is backed up first and then opened. The project file is backed up to the project directory and named in the format of "original project file name_bk_version of AutoShop of the project.hcpp". The backup details are displayed in the information output window, as shown in the following figure.



When a project is opened in AutoShop of a later version, its project version will be updated to the version of AutoShop in which it is opened. AutoShop of an earlier version cannot open this project, but can open the backup project and obtain the project information. AutoShop of 4.10.1.0 and later versions support project backup.

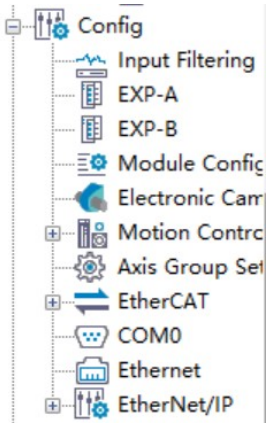
4.4.3 Connecting to the Target PLC

This section takes the Ethernet connection between the PC and PLC as an example. For details about how to connect to the target PLC, see 4.2.2 Ethernet Connection.

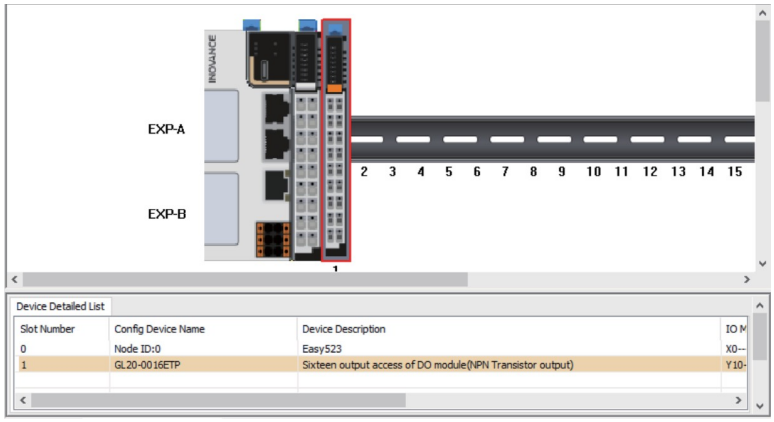
4.4.4 (Optional) Performing Hardware Configuration

If the local expansion module is installed for the H5U host, perform the steps in this section. Otherwise, skip this section.

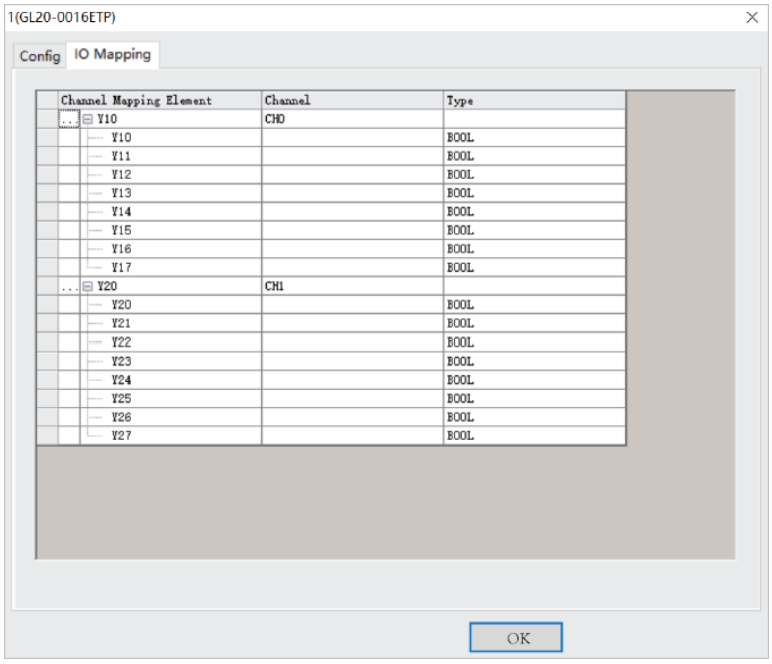
1. In the “Project Manager” tree, unfold “Config” and double-click “Module Config”. The “Module Config” page is displayed.



2. Double-click the modules under “Module” on the right of the page based on the installation sequence to add the corresponding modules.



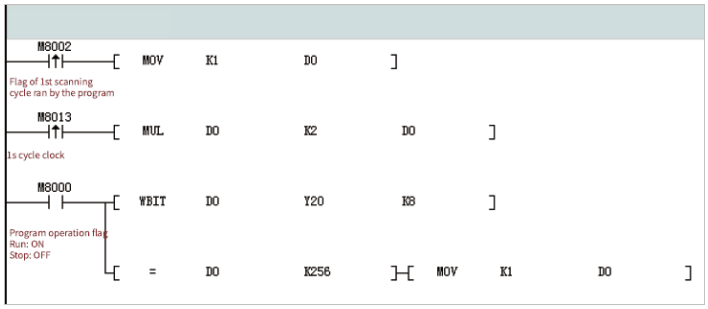
3. Double-click an added module and configure the channel mapping element corresponding to the module channel.



4. Click “OK”.

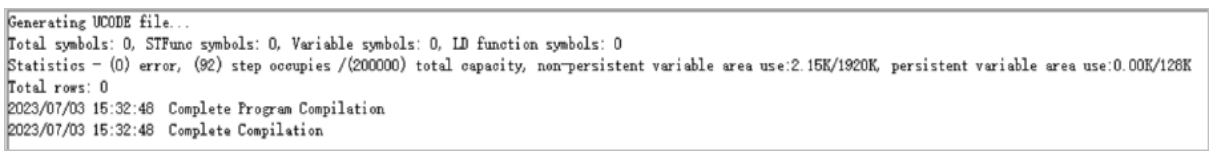
4.4.5 Programming and Compiling

1. In the "Project Manager" tree, unfold "Programming" and double-click "MAIN" to compile the marquee program.



2. (Optional) Save the file.
   - Save all files: In the toolbox, click to save opened files (modified files) whose name contain "*" of the current project.
   - Save the project: In the toolbox, click to save all opened and unopened files of the current project.
3. In the toolbar, click or for compiling.
   - : Compile the opened program.
   - : Compile the overall project.

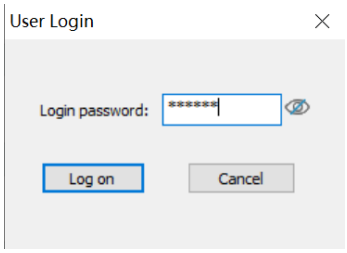
   After compiling is completed, the compilation information is displayed at the bottom of the main screen.



4.4.6 (Optional) Logging in to PLC

In case of Ethernet connection between a PC and a PLC with the login password set, to modify, upload, download, or verify the program online, perform the steps in this section. Otherwise, skip this section.

1. In the toolbar, click . The "User Login" page is displayed.



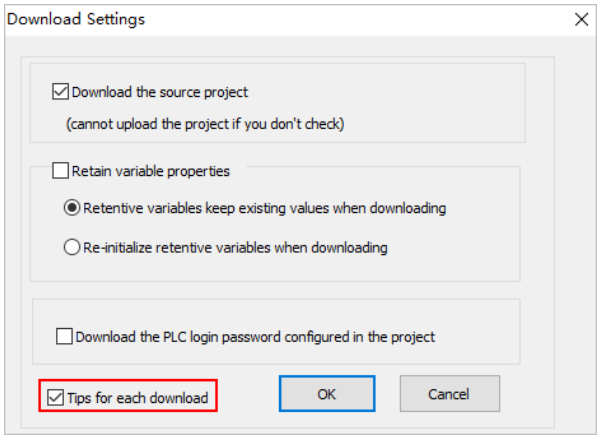
2. Enter the PLC login password in the "Login password" text box and click "Log on".

> **NOTE**
>
> For more details about logging in to the PLC, see 4.10 Logging in to PLC.

4.4.7 Downloading the Program

1. Choose "PLC" > "Download" in the menu bar or click in the toolbar to download the program.

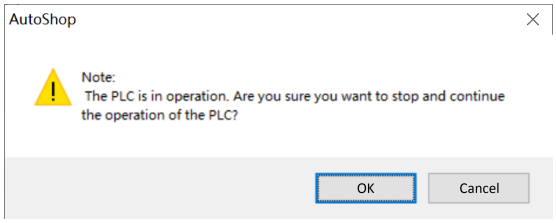
- If "Tips for each download" is selected, the "Download Settings" dialog box is displayed. Click "OK", as shown in the following figure.



Name	Description
"Download the source project (cannot upload the project if you don't check)"	Select this option to enable project uploading, or deselect this option to disable project uploading.
"Retain variable properties"	Retain existing values of retentive variables when downloading; or Re-initialize retentive variables when downloading.

> **NOTE**
>
> If "Tips for each download" is deselected, the "Download Settings" dialog box is not displayed.

- If the PLC is in the running state, perform operations to access the prompt dialog box and click "OK", as shown in the following figure: Proceed to Step 2.



- If the PLC is in the stopped state: Proceed to Step 3.

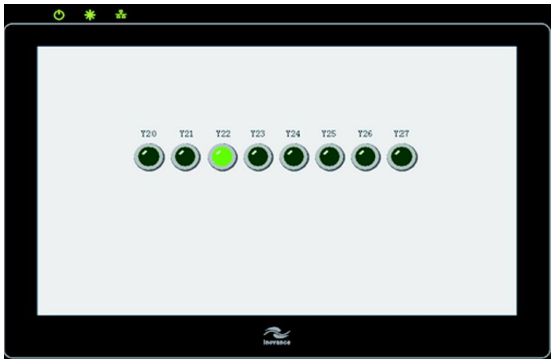
3. Upon download completion, click "OK" in the dialog box to switch the PLC to the running state.
4. Upon download completion, click in the toolbar to switch the PLC to the running state.

> **NOTE**
>
> When the new PCB software (V3.0.0.0) is used with an old AutoShop background (V4.0.0.0 or earlier version), the following situation will occur:
>
> If the PLC is not configured with a login password, projects downloaded through the new background cannot be uploaded through the old background.

4.4.8 HMI Monitoring

You can monitor the execution of the marquee program through the HMI.

1. Create a project, connection, or variable and view the configuration screen on the HMI side. For details, see HMI Quick Start.
2. When running the PLC program, you can monitor the marquee status change on the HMI configuration screen.



4.5 Switching PLC Working Mode

A PLC can work in RUN or STOP mode.

- RUN: The PLC detects input at the X point, scans and calculates the user program, refreshes elements, and enables output and communication at the Y point.
- STOP: The PLC stops scanning of the program and output at the Y point, and disables the communication function.

PLC working modes can be switched in either of the following two ways:

- Toggle the "RUN/STOP" switch on the PLC.
- In the toolbar, click to run the PLC or click to stop the PLC.

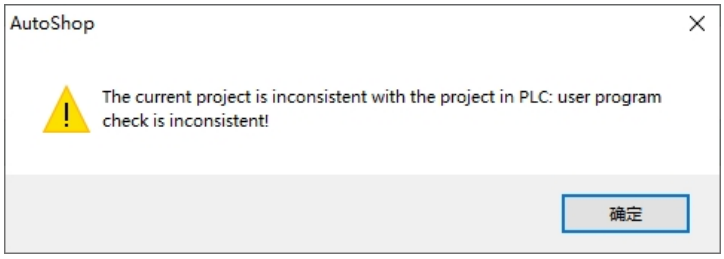
4.6 Editing the Program Online

After the mode is enabled, online editing of the program does not affect the running PLC. Therefore, the program can be edited without the need of stopping the PLC, facilitating program commissioning.

CAUTION

- Program modification does not support:
  - Adding or deleting program files
  - Renaming program files
  - Modifying program file properties
  - Encrypting or decrypting subprograms
- Modifying configuration is not supported.
- Global variables:
  - Adding or deleting variables is supported.
  - Modifying names or comments of variables is supported, but modifying other features is not supported.
- FB/FC local variables:
  - Adding, modifying, or deleting IN/INOUT/OUT type variables is not supported.
  - Adding or deleting Var type variables is supported.
  - Var variables added before the current online modification cycle: Modifying names or comments of Var type variables is supported, but modifying other features is not supported.
  - Var variables added during the current online modification cycle: Modifying any features of Var variables is supported.
- Do not power off the system within 5s to 10s after successful online modification and download of the program. Otherwise, program errors may occur.

1. Choose "PLC" > "Online Edit Mode" in the menu bar or click in the toolbar to enable the online modification mode. At the same time, the software enters the monitoring mode.
   In online modification mode, if the running program is different from the program in the PLC, the "Failed to enter online modification mode!" prompt dialog box is displayed. In this case, you need to check whether the program is started correctly.



2. Click to compile the program online as required.
3. Click to download the modified program to the running PLC. The PLC does not stop running during the operation.
   Information shown in the following figure is displayed in the lower part of the main screen. Then, the PLC runs with the modified program.

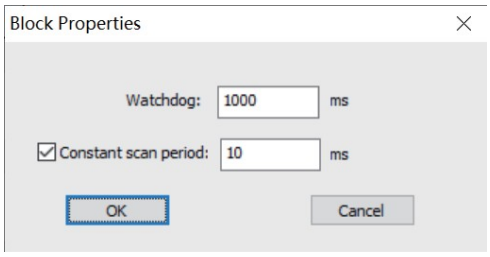
```
Information(2023-07-03 15:45:42)  Downloadmodbus.foid Success
Information(2023-07-03 15:45:42)  Downloadcanlink.foid Success
Information(2023-07-03 15:45:42)  Downloadmodcfg.foid Success
Information(2023-07-03 15:45:42)  Downloadstprog.o Success
Information(2023-07-03 15:45:42)  DownloadEcam.foid Success
Information(2023-07-03 15:45:42)  Download successful!
```

4.7 Setting a Program Scan Cycle

You can select the constant scan cycle or the non-constant scan cycle.

- When the constant scan cycle is selected, the program runs by the specified scan cycle. If the actual running time is less than or equal to the set scan cycle, the PLC scans the program based on the set value. If the actual running time is greater than the set scan cycle, the PLC scans the program based on the actual running time.
- When the non-constant scan cycle is selected, the PLC automatically adjusts the scan cycle based on the running time of the program.

1. In the “Project Manager” tree, right-click “Program Block” and select “Block Properties”. The “Block Properties” dialog box is displayed.



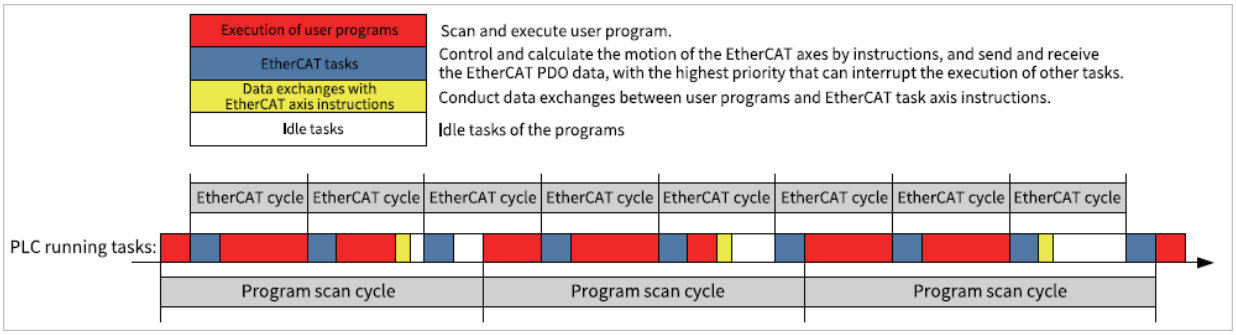
2. Set “Watchdog” and “Constant scan period”.
   - ① Watchdog: indicates the maximum scan cycle of the program. When the running time of the program is greater than the value, the PLC reports an error and stops running. Setting range: 100 ms to 30000 ms
   - ② Constant scan period: When the check box is selected, the constant scan cycle mode is selected and the scan cycle equals to the value. Otherwise, the scan cycle equals to the actual running time.
3. Click “OK” and download the program to the PLC. The settings take effect immediately.

4.8 Setting an EtherCAT Task Cycle

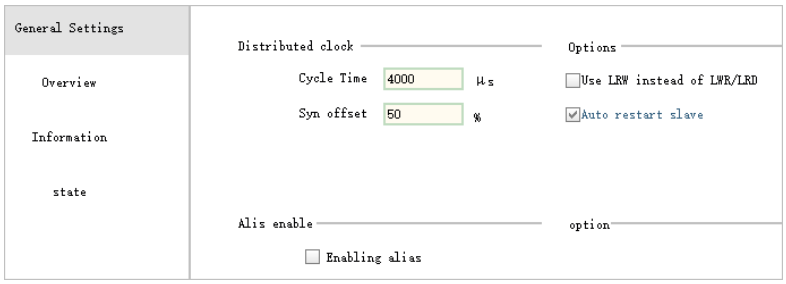
You can set EtherCAT task cycles as needed.

One program scan cycle contains the program execution, EtherCAT task, and EtherCAT axis instruction data exchange. When program scanning is idle, the scan cycle also contains the idle task. The priority of the EtherCAT task is the highest so that the EtherCAT task can interrupt execution of other tasks.

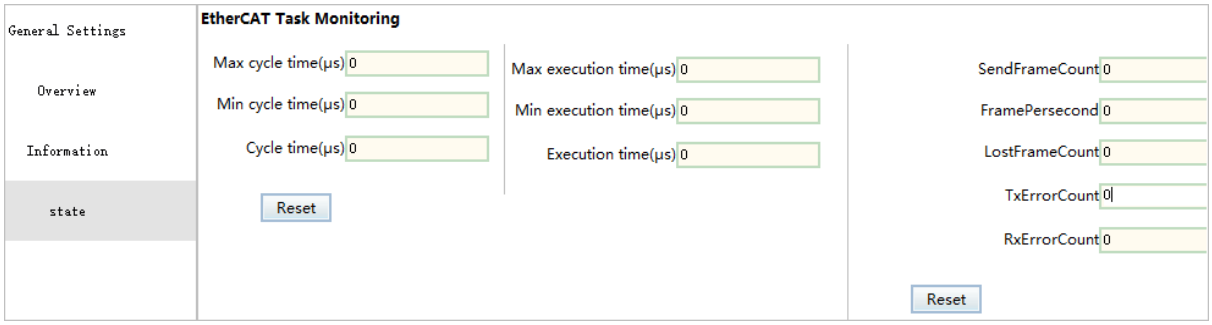
The following figure shows the relationship between the EtherCAT task cycle and the program scan cycle.



1. In the "Project Manager" tree, unfold "Config" and double-click "EtherCAT". The "General Settings" page is displayed, as shown in the following figure.



2. Set the cycle time of the EtherCAT task to a value ranging from 1000 μs to 9000 μs.
3. (Optional) In monitoring mode, click the "state" tab. On the "EtherCAT Task Monitoring" page displayed, obtain communication information, as shown in the following figure.



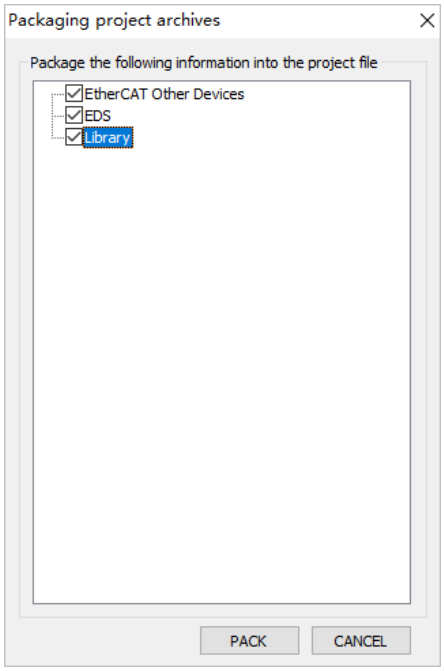
4. Click "OK" and download the program to the PLC. The settings take effect immediately.

4.9 Packing and Decompressing a Project Archive

A project archive is a .hclib file consisting of the current project, EDS file, library file, and third-party EtherCAT device XML file. The current project cannot be a temporary project, and the library file used in the program is packed by default. To provide your project for other users, you only need to pack the project archive into a .hclib file and send it to the target user. The user then decompresses the file and directly compiles and downloads the project archive, without the need of sending the EDS file, library file, or third-party XML file.

Packing a project archive

1. In the menu bar, choose "File" > "Pack Project Archives". The "Packaging project archives" dialog box is displayed.



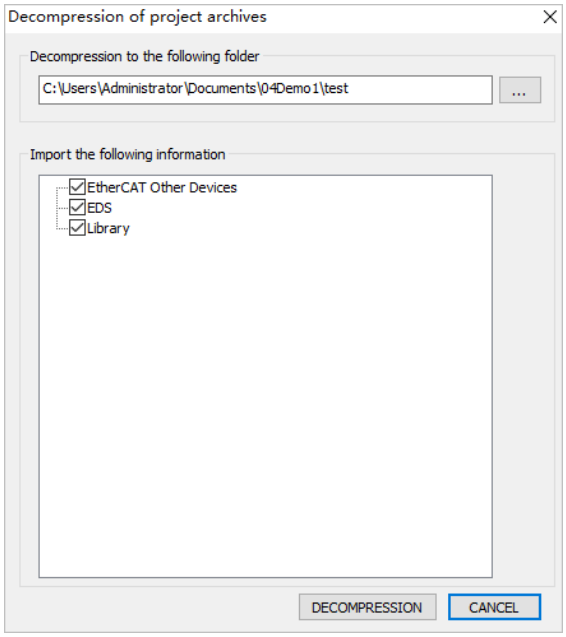
2. Select the files to be packed (library files used in the program are automatically selected), and click "PACK" to save the .hclib file.

Decompressing a project archive

Preconditions: A .hclib file is prepared.

Procedure

1. In the menu bar, choose "File" > "Decompress Project Archives". The "Open" dialog box is displayed.
2. Select and open the .hclib file to be decompressed. The "Decompression of project archives" dialog box is displayed.



3. Set "Decompression to the following folder".
   By default, the project archive is decompressed into the folder with the project archive name in the folder where the .hclib file is located. If the storage folder does not exist, it is automatically created. You can also click "..." to select a target folder for decompression.
4. Select required files (all selected by default), click "DECOMPRESSION", and import the third-party EtherCAT device XML file, EDS file, and library file.

4.10 Logging in to PLC

4.10.1 Overview

> **CAUTION**
>
> Only PLCs with a login password set need to use this function.

The login function protects the intellectual property of customers and prevents the PLC program from unauthorized modification.

The following table lists the number of login users, login password, and login permissions.

、Item		Description
Number of login users		Only one user can log in to the PLC at a time. Other users can log in only after the user logs out. Otherwise, a login failure is prompted.
Login password		A login password contains up to eight characters in ANSI C mode.
Login permissions	Upload and download[1]	A permission password is required.
	Modifying the program online	A permission password is required.
	RTC	A permission password is required.
	Changing the IP address	A permission password is required.
	Clearing the PLC program storage space	A permission password is required.
	Verifying the program	A permission password is required.

> **NOTE**
>
> [1]: Upload and download refer to the upload and download of user programs and configuration data through the AutoShop programming software.

4.10.2 PLC Login and Logout

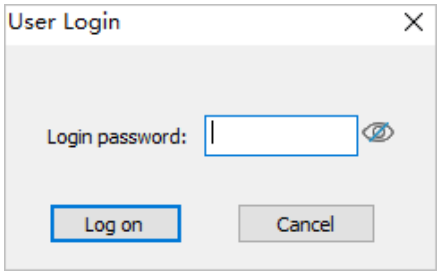
Logging in to PLC

- **Preconditions**

  The PC is properly connected to the PLC, and the PLC is online.

- **Procedure**

1. In the toolbar, click . The "User Login" page is displayed, as shown in the following figure.



2. Enter the PLC login password in the "Login password" text box and click "Log on" to log in to the PLC. After successful login, the login button turns gray and the logout button turns red.

> **NOTE**
>
> After a user logs in to the PLC, other users attempting to log in to the PLC will receive a pop-up prompt saying that the PLC is being used and rejects the login request. Only after the current user logs out can other users log in to the PLC.

Logging out of PLC

- **Preconditions**
  The PC is properly connected to the PLC, and the PLC is online.
- **Procedure**

1. In the toolbar, click to log out of the PLC.
2. After successful logout, the login button turns blue and the logout button turns gray, as shown in the following figure.

4.10.3 Login Password Management

Introduction

The password used to log in to the PLC can be set, changed, or deleted.

NOTE

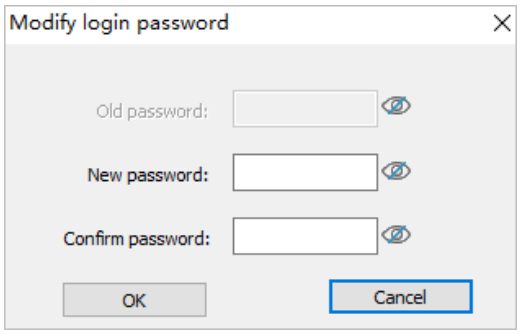
1. The password is saved in memory. When you re-open the project and re-connect to the PLC, you need to log in again.
2. When generating a Down file, you can opt for the login password. If no password is set, you do not need to enter a password.
3. A password can only contain some characters in the ANSI character set, including uppercase and lowercase letters, numbers, and some special characters such as parenthesis ((), comma (,), exclamatory mark (!), and atmark (@).
4. When you opt for the login password, some functions, such as IP address modification, PLC scan, and login password modification, will exit the login status. After opting for the login password, you need to log in again.

Preconditions

- Software version: AutoShop V4.10.2.0 and later
- The PC is properly connected to the PLC, and the PLC is online.

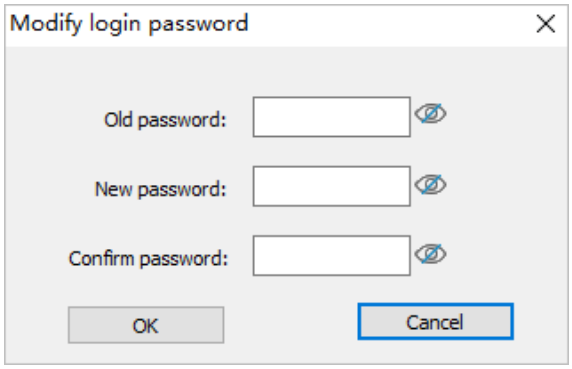
Procedure

1. Set a PLC login password.
   a. In the menu bar, choose "PLC" > "Set/Modify Login PLC Password". The "Modify login password" dialog box is displayed.



   b. Enter the new password in the "New password" and "Confirm password" text boxes, and click "OK" to set the PLC login password.

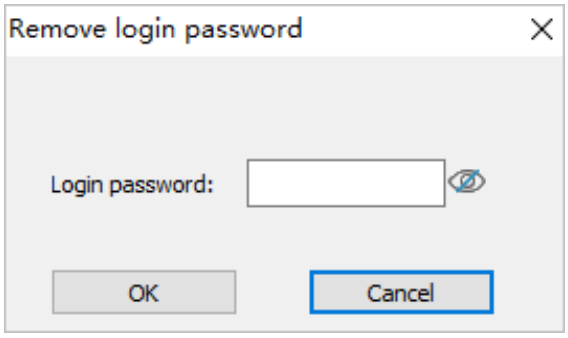
2. Modify the PLC login password.
   a. In the menu bar, choose "PLC" > "Set/Modify Login PLC Password". The "Modify login password" dialog box is displayed.



b. Enter the old password in the "Old password" text box and the new password in the "New password" and "Confirm password" text boxes, and click "OK" to change the PLC login password.

3. Delete the PLC login password.

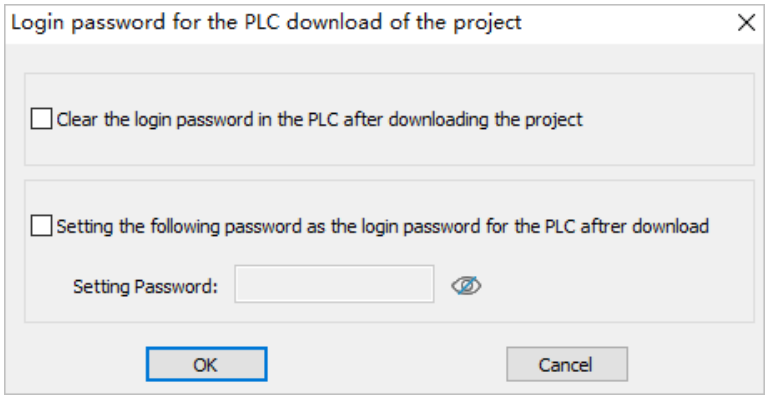
a. In the menu bar, choose "PLC" > "Delete Login PLC Password". The "Remove login password" dialog box is displayed, as shown in the following figure.



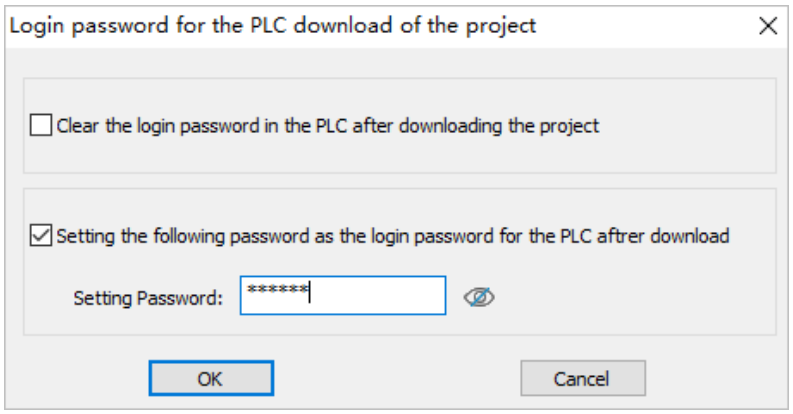
b. Enter the PLC login password in the "Login password" text box and click "OK" to delete the PLC login password.

Setting a PLC login password for a project that is still valid when the project is downloaded

1. In the menu bar, choose "File" > "Set/modify PLC login password in project". The password setting dialog box is displayed, as shown in the following figure.

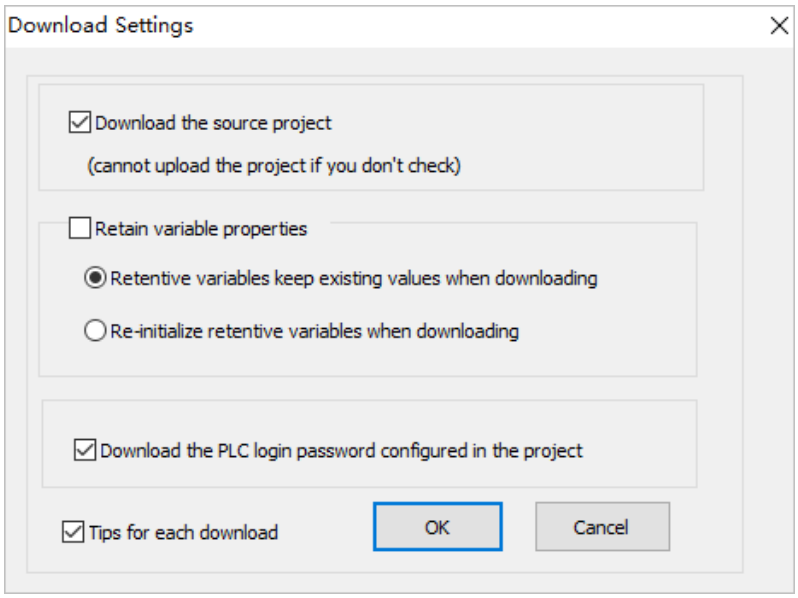


2. Select "Setting the following password as the login password tor the PLC after download". The "Setting Password" field is available and then you can set a password. Set a password and click "OK". The PLC login password is saved to the project.

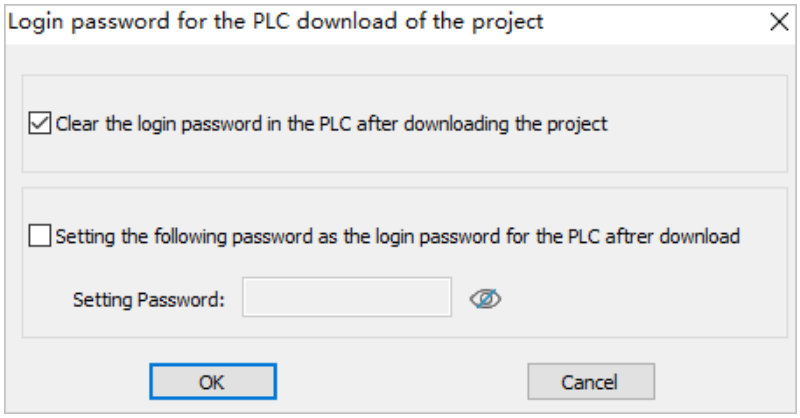


3. After a PLC login password in a project and "Download the PLC login password configured in the project" is selected in the "Download Settings" window, the PLC login password saved to the project will be

replaced by the current login password of the PLC during project download.



4. Similarly, if "Clear the login password in the PLC after downloading the project" is selected and "OK" is clicked, the PLC login password saved in the project will be cleared during project download. In this case, if "Download the PLC login password configured in the project" is also selected, when the project is downloaded, the PLC login password is cleared and no login is required for subsequent PLC use.



4.11 Trace Monitor Variables

4.11.1 Overview

Like a digital sampling oscilloscope, the Trace feature can record historical values of variables. When the Trace feature is enabled, AutoShop starts to save the data records containing time, and you can continuously monitor variable changes on the Trace page to facilitate program commissioning.

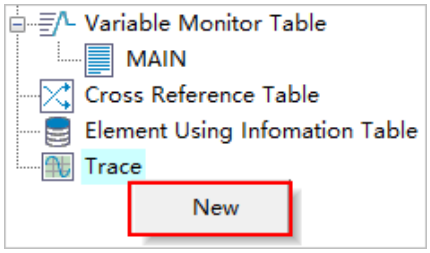
4.11.2 Adding Trace Monitor Variables

Introduction

This section introduces how to add Trace monitor variables by capturing changes of the D100, D200, and TEST variables.

Procedure

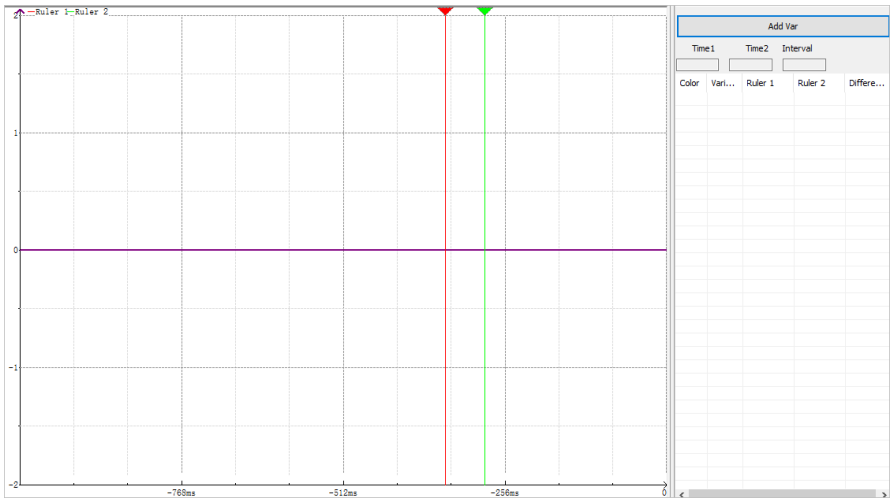
1. In the "Project Manager" tree, right-click "Trace" and select "New" to create a Trace monitor view, as shown in the following figure.



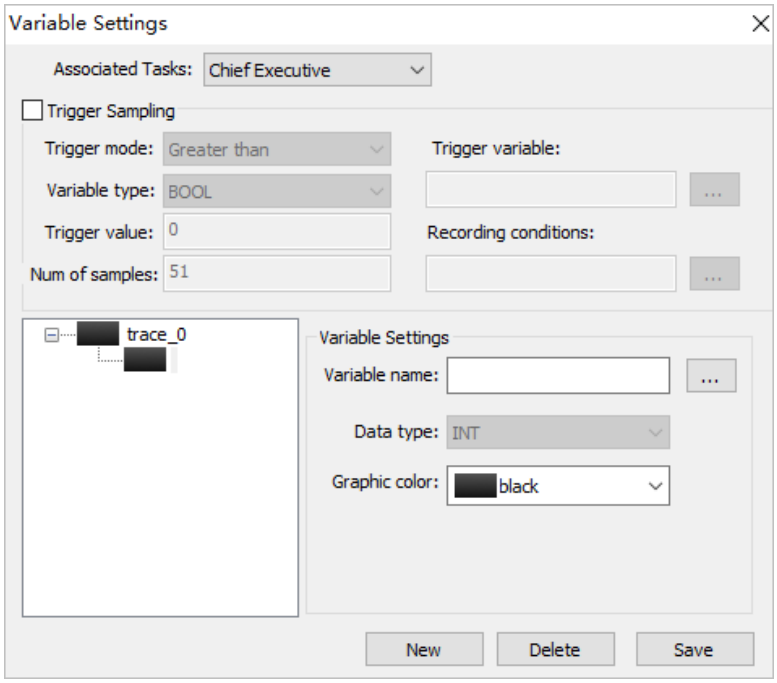
NOTE

Right-click the newly created monitor chart and select "Delete" to delete the monitor chart, or select "Rename" to rename the monitor chart. The name can contain up to 128 characters and cannot contain Chinese and English punctuation marks.

2. Double-click the new Trace monitor chart to access the Trace page, as shown in the following figure.



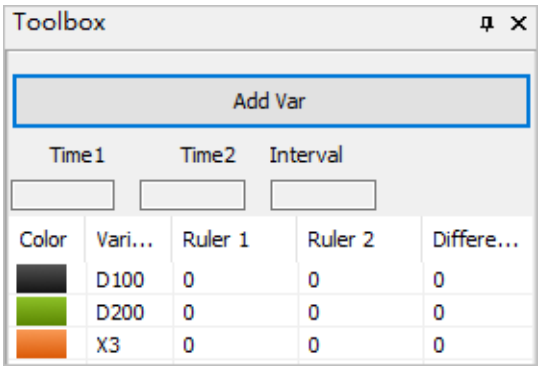
3. Click "Add Var". The "Add Var" dialog box is displayed, as shown in the following figure.



Parameter Name	Description
Associated Tasks	Indicates a variable used to select a monitor for sampling in main tasks or EtherCAT tasks.

Parameter Name	Description
	Select "Chief Executive" or "Ethercat Task" from the drop-down list.
Trigger Sampling Enables	Indicates the function switch to trigger sampling enabling, which is valid upon checked.
Trigger mode	Indicates the mode to trigger sampling. Select "Greater than", "Less than", "Not equal to", "Equal to", "Greater than or equal to", or "Less than or equal to" from the drop-down list.
Trigger value	Indicates that triggering stops when the trigger variable meets the trigger mode and trigger value conditions.
Variable type	Indicates the data type of a trigger variable. Select "BOOL", "INT", "DINT", "REAL", or "BYTE" from the drop-down list.
Number of samples	Indicates the number of sampling points before trigger stops when trigger conditions are met.
Trigger variable	Indicates a trigger object.
Recording conditions	Indicates a BOOL variable of recording conditions to start tracking. This parameter can be left empty.
Variable Settings	Indicates the variable name of the monitor, corresponding variable data type, and graphic color on the Trace monitor view. Note: Up to ten variables are supported.

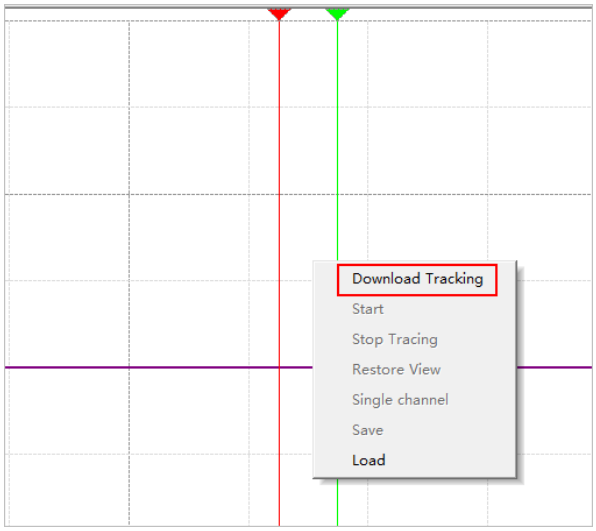
4. Set "Associated Tasks" to "Chief Executive" and set the D100, D200, and TEST variables to be captured as an example. Set other parameters according to the preceding table, and click "Save" to save the variable settings, as shown in the following figure. View variable information in the toolbox on the right.



Parameter Name	Description
Color	Indicates the legend color of a variable on the Trace view.
Variable Name	Indicates the name of a variable.
Time1	Indicates the value and time of the point corresponding to ruler 1.

Parameter Name	Description
Time2	Indicates the value and time of the point corresponding to ruler 2.
Difference/time	Indicates the value and time difference between the points corresponding to the two rulers.

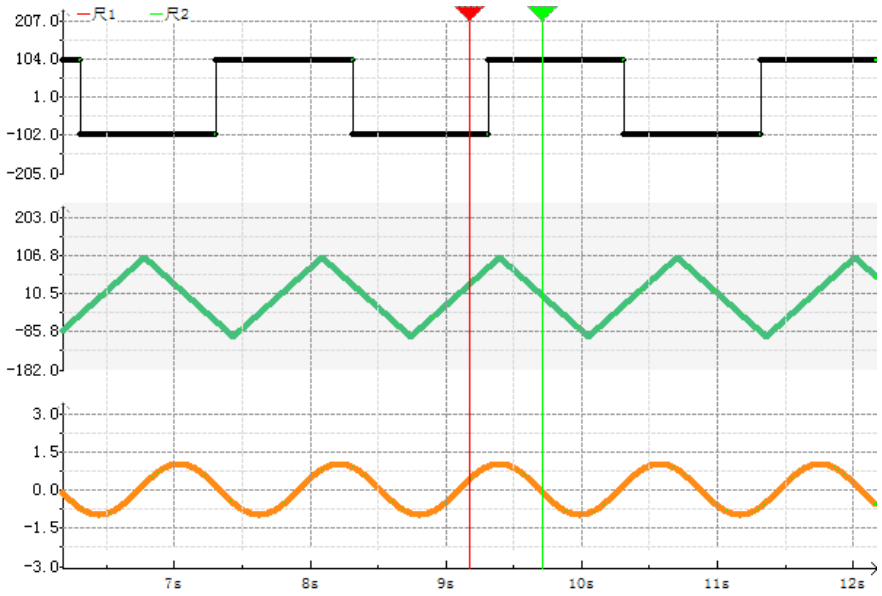
5. After configuring the variables, right-click on the Trace view and select "Download Tracking", as shown in the following figure.



The following table lists the shortcut menus.

Parameter Name	Description
Download Tracking	Indicates to download the set tracking variables and start tracking.
Start	Indicates to start the original variables to continue to track data acquisition after tracking variables are stopped.
Stop Tracing	Indicates to stop data acquisition after tracking starts.
Restore View	Indicates to restore the initial acquisition speed and data interval after you use the mouse roller to zoom in or out and translate the view.
Single channel/Multichannel	Indicates variable collection for a single coordinate system or multiple coordinate systems.
Save	Indicates to save all collected data in the tracking stopped state.
Load	Indicates to load the saved file of collected data.

6. After tracking is downloaded, you can start to monitor variables, as shown in the following figure.



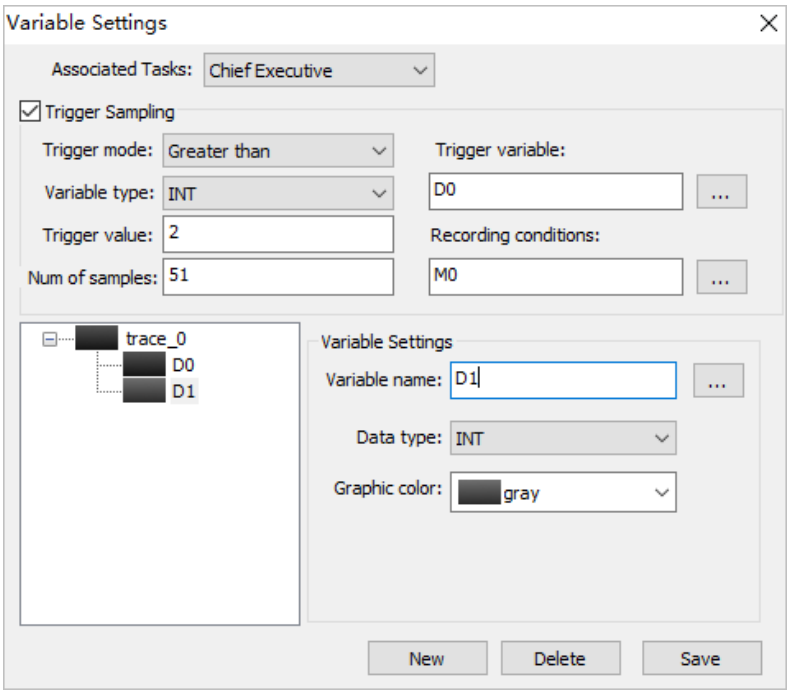
NOTE

In Trace mode, the horizontal and vertical coordinates of the chart automatically adapt to variable values. You can adjust the coordinates by scrolling the mouse wheel. Specifically, directly scroll the mouse wheel to zoom in or out the horizontal coordinate, and scroll the mouse wheel while holding down the Ctrl key to zoom in or out the vertical coordinate.

In Monitor mode, you can drag the rulers to view the data of specific points, or drag the curve back and forth to view the data of key points. During this process, the Trace monitor chart stops scrolling, but data is still being collected. To restore to automatic scrolling mode, right-click the Trace monitor chart and select "Restore View".

For other operations to the Trace monitor chart, see the preceding table of shortcut menus.

Trace monitor example



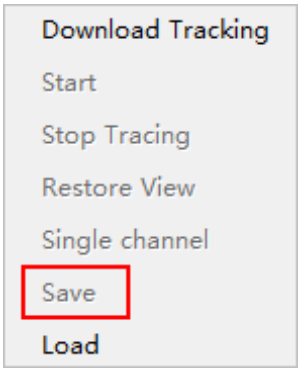
When M0 is set to TRUE, Trace sampling starts. When D0 is greater than value 2 of "Trigger value", sampling stops after 51 data entries are sampled. D0 and D1 are tracking variables.

4.11.3 Importing and Exporting Trace Data

Exporting Trace data

- Preconditions
  Data acquisition is stopped, that is, tracking is stopped.
- Procedure

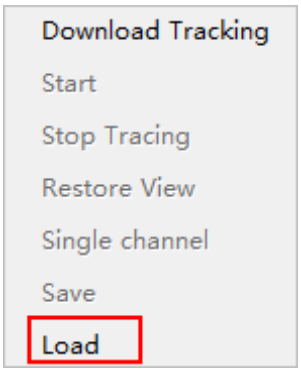
Right-click on the Trace monitor view and select "Save" to save data to the corresponding folder.



Importing Trace data

- Preconditions
  - A Trace data file is prepared.
  - Data acquisition is stopped, that is, tracking is stopped.
- Procedure

Right-click on the Trace monitor view and select "Load". In the dialog box that is displayed, select a Trace data file to load data to the Trace monitor view.



4.12 Data Record

Introduction

The data record (offline Trace) function records the value of the specified variable or register according to certain conditions and save the record data in .CSV format to the SD card mounted on the PLC.

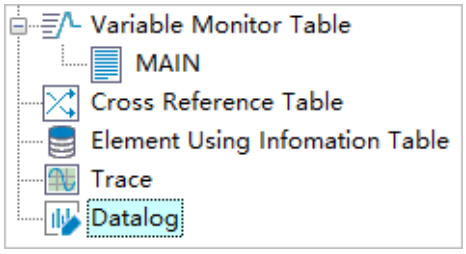
Preconditions

- Applicable models: Easy523, Easy522, Easy521, Easy502, Easy501, Easy320, and Easy302
- Firmware version: V6.4.0.0 and later

- Software version: AutoShop V4.10.2.0 and later
- The GE20-TF-RTC extension card is obtained.
- The SD capacity is greater than 256 MB and the remaining capacity is greater than 128 MB.

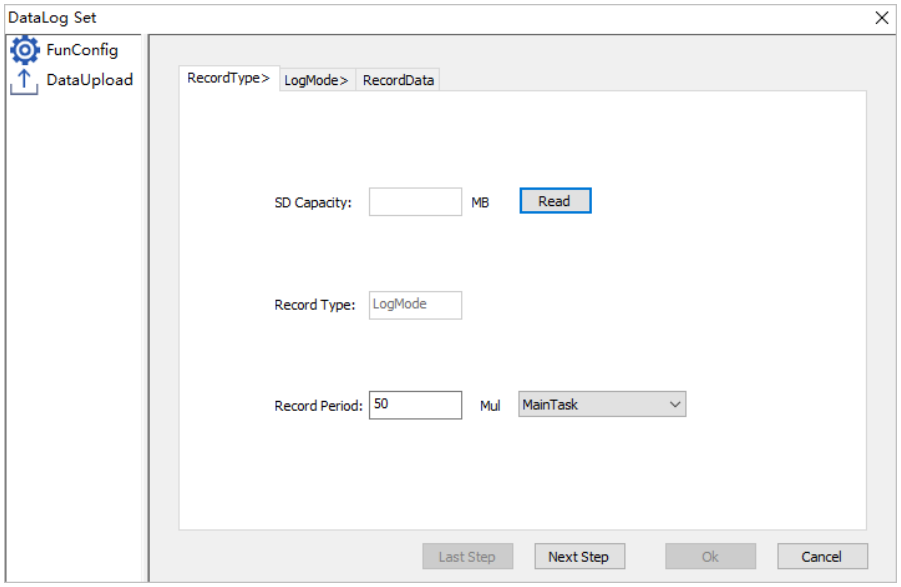
Procedure

1. In the "Project Manager" tree, double-click "Datalog". The "DataLog Set" window is displayed, as shown in the following figure.



2. The data record function is configured by a wizard, which is RecordType > LogMode > RecordData. You can click "Last Step" or "Next Step" to switch between tabs. You can click "OK" only after you have made configurations on the tab page.

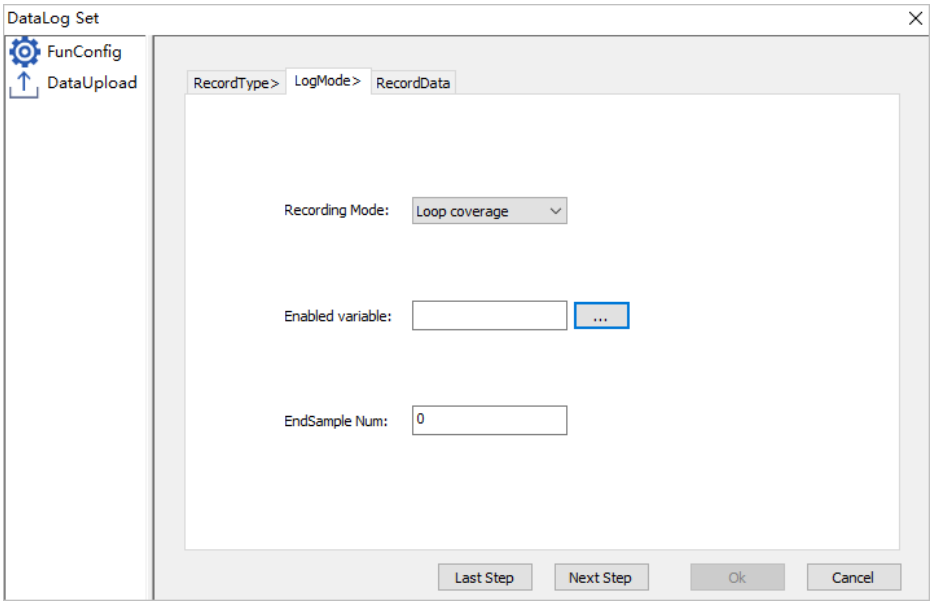
   The following figure shows configurations on the "RecordType" tab page.



Name	Description
SD Capacity	It is read-only and blank by default, showing the available capacity of the current SD card in MB.
Read	Read the remaining capacity of the SD card. • If the PLC has no SD card, "No SD card" is displayed. • If the remaining capacity of the SD card is less than 128 MB, the system prompts "The remaining capacity of the SD card is less than 128 MB. Please replace the SD card in time".
Eject	This button is hidden by default. If the SD capacity is displayed after you click "Read", this button is displayed. Click this button to eject the SD card in a safe manner.
Record Type	Only the log mode is supported.

Name	Description
Record Period	Set the data recording interval. The value range is 1 to 65535 in the unit of main task period or EtherCAT task period. The default value is 50.

The following figure shows configurations on the "LogMode" tab page.

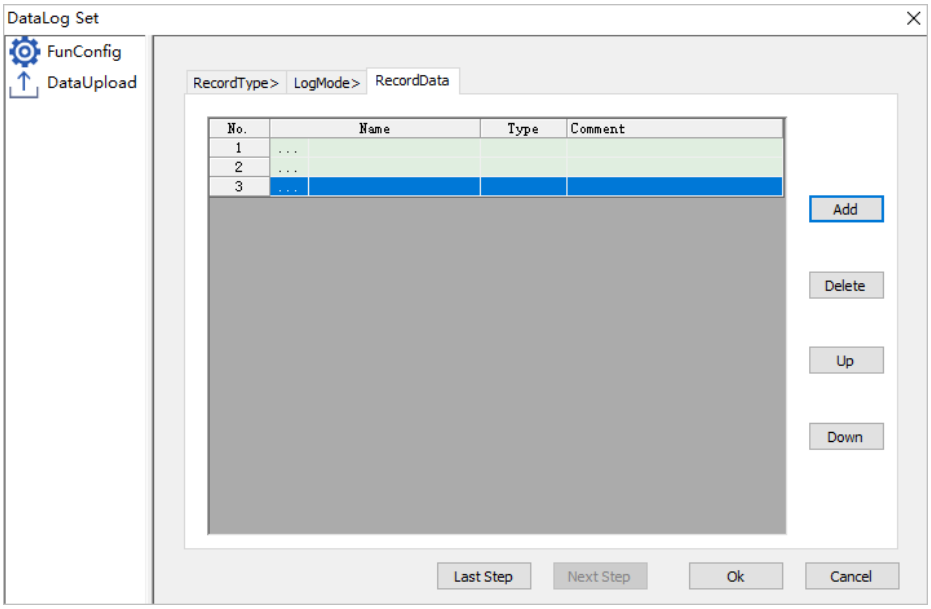


Name	Description
Recording Mode	Options are "Loop coverage" and "Stop when used up". "Loop coverage" is selected by default. • Loop coverage: When the remaining capacity of the SD card is 128 MB, new data overwrites the earliest data. • Stop when used up: When the remaining capacity of the SD card is 128 MB, data writing stops.
Enabled variable	Only defined BOOL-type variables can be input. Click "..." to display the variable input assistant.
EndSample Num	It indicates the number of the sampled records continued after "Enabled variable" turns from ON to OFF. The default value is 0, with value ranging from 0 to 65535.

NOTE

The data record function can save up to 1 GB data and can keep the record for at least 100 minutes.

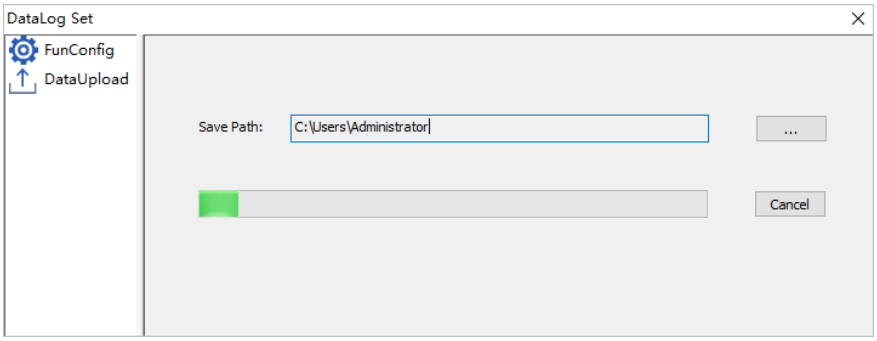
The following figure shows configurations on the "RecordData" tab page.



Name	Description
No.	Up to 32 variables can be recorded at a time.
Variable Name	You can only select or manually input variables defined in the variable table. When a not-defined variable is manually input, the row of the variable turns red.
Data Type	Five data types are supported: BOOL, INT, DINT, REAL, and BYTE.
Comment	When an added variable has comments, its comments are automatically obtained and displayed.
Add	Add a row.
Delete	Delete the row where the cursor is located.
Up	Move the variable from the row where the cursor is located up by one row.
Down	Move the variable from the row where the cursor is located down by one row.

3. After configuration, click "OK". The configuration is saved and the "DataLog Set" window is closed. If the current configuration has an error, a window is displayed, asking you to make correction. The configuration can be downloaded together with the project to the PLC and uploaded together with the project to AutoShop.

4. Data Upload: Upload the CSV file in the SD card mounted on the PLC to the PC on which AutoShop runs, as shown in the following figure.



Name	Description
Save Path	Click "...". The "Save Path" field is displayed and the selected path is displayed in this field.
Export/Cancel	After you click this button, the system starts to upload the file from the PLC and displays the transmission result upon upload completion. • Export: This button is hidden by default, and is displayed after a save path is selected. After this button is clicked, the system starts to export the file and this button is hidden until the export is completed. • Cancel: This button is hidden by default. It is displayed when the system starts to export the file and displays a prompt window, and is hidden in this progress.
Progress bar	The progress bar is hidden by default. When file transmission starts, the progress is displayed in real time. When transmission ends, the progress bar is hidden again.

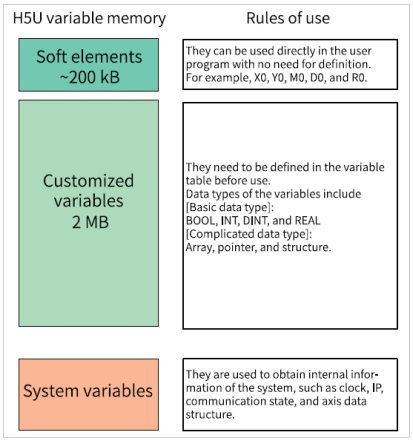
- Before uploading data, ensure that AutoShop is connected to the PLC properly. During export, do not switch the function configuration view; otherwise, the upload will be terminated.
- Never remove the SD card during data recording; otherwise, the SD card may be damaged.

5 Programming Basics

5.1 Overview

The variable memory structure for programming contains soft elements, customized variables, and system variables.

The following figure shows the corresponding rules of use.



5.2 Elements

5.2.1 Bit Elements

The PLC supports bit elements. The following table describes the specific type, range, number of points, and description of bit elements.

Type	Range	Number of Points	Data Type	Description
X	X0 to X1777	1024 points, octal	BOOL	Input
Y	Y0 to Y1777	1024 points, octal	BOOL	Output
M	M0 to M7999	8000 points	BOOL	M0 to M999 not retained upon power failure, M1000 to M7999 retained upon power failure
S	S0 to S4095	4096 points	BOOL	S0 to S999 not retained upon power failure, S1000 to S4095 retained upon power failure
B	B0 to B32767	32,768 points	BOOL	B0 to B999 not retained upon power failure, B1000 to B32767 retained upon power failure

NOTE

- Bit elements of the BOOL type can be used in input or output parameters of the BOOL-type instructions.
- X and Y are used as input and output elements respectively.
- M elements used for ordinary programming only support up to M7999. Some soft elements after M8000 can be directly programmed in the program. M8000 special features are detailed in the section about special soft elements.

5.2.2 Word Elements

The PLC supports word elements. The following table describes the specific type, range, number of points, and description of word elements.

Type	Range	Number of Points	Data Type	Description
D	D0 to D7999	8000 points	BOOL/INT/DINT/ REAL	D0 to D999 not retained upon power failure, D1000 to D7999 retained upon power failure
R	R0 to R32767	32,768 points	BOOL/INT/DINT/ REAL	R0 to R999 not retained upon power failure, R1000 to R32767 retained upon power failure
W	W0 to W32767	32,768 points	BOOL/INT/DINT/ REAL	W0 to W999 not retained upon power failure, W1000 to W32767 retained upon power failure

- The range of words that are retained upon power failure cannot be changed.
- A word element can be used as an integer or floating-point number. As an element does not have the data type attribute, whether an element is used as an integer or floating-point number is determined by a parameter in the corresponding instruction.
- When a word element is used as an integer, it is a 16-bit or 32-bit integer based on the instruction parameter. When used as a 16-bit integer, a word element occupies one element. When used as a 32-bit integer, a word element occupies two elements.
- When used as a floating-point number, a word element occupies two elements.

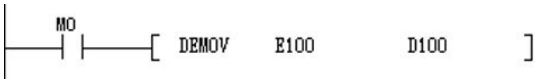
Example

- Word element used as a 16-bit integer
  Use the 16-bit assignment instruction to assign the value 100 to the word element D100, which occupies D100.

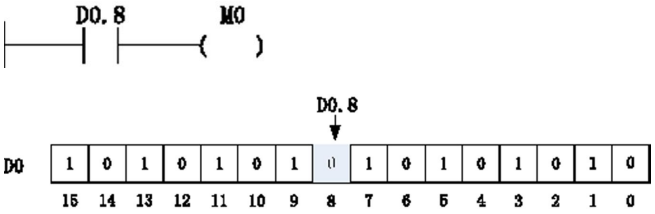
- Word element used as a 32-bit integer

Use the 32-bit assignment instruction to assign the value 100 to the word element D100, which occupies D100 (low-order) and D101 (high-order).

- Word element used as a floating-point number
  Use the floating-point instruction to assign the value 100 to the word element D100, which occupies D100 and D101.



- Bit-based operation on word elements
  Bit-based operations on word elements can be implemented by using a dot (.). For example, writing D0.8 during programming indicates an operation on the 8th bit of the word element D0.



The bits of the word element are counted from the 0th bit. When the 8th bit of D0 is 0, the output M0 is OFF. When the 8th bit of D0 is 1, the output M0 is ON.

5.2.3 Special Elements

The PLC supports special elements. The following table describes the type, range, function, and description of the special elements supported by the PLC.

Type	Function	Range	Number of Points	Description
SBR	Subprogram label	SBR0 to SBR1023	1024	Used by the CALL instruction. Subprograms can be set as common subprograms or encrypted subprograms, which share the capacity of the system program area.
L	Jump label	L0 to L1023	1024 points	Used in combination with the CJ and LBL instructions
I	External interrupt	-	4	Interrupt label, which is on the rising edge, falling edge, or rising and falling edges of X port
	Timer interrupt	-	4	Timing duration, in ms
	Comparison interrupt	-	16	Limited by the number of internal encoder axes (high-speed counters)

Type	Function	Range	Number of Points	Description
K	Decimal	K-32768 to K32767 (16-bit), K-2,147,483,648 to K2,147,483,647 (32-bit)	-	-
H	Hexadecimal	H0000 to HFFFF (16-bit), H00000000 to HFFFFFFFF (32-bit)	-	-
E	Floating-point number, real number	-3.402823e+38 to -1.175495e-38, 0, +1.175495e-38 to +3.402823e+38	-	Up to 7 decimal significant digits for a single-precision floating-point number (The excess part is automatically rounded off.)
Character	Character, string	-	-	Used as instruction parameters

A single-precision floating-point number has a maximum of 7 significant decimal digits. If the 9-bit decimal floating-point number 1234567.89 is transmitted to the destination location D0, the actual value of D0 is 1234567.9. The precision is reduced, as shown in the following figure.

5.3 Variables

5.3.1 Introduction

In a PLC programming system, in addition to using direct addresses, such as the X, Y, M, D, R and other elements, for programming, you can also use variables without specific storage addresses for programming to implement the required control logic, or the complete control process of the application object, so as to facilitate code compiling and improve the readability of the code.

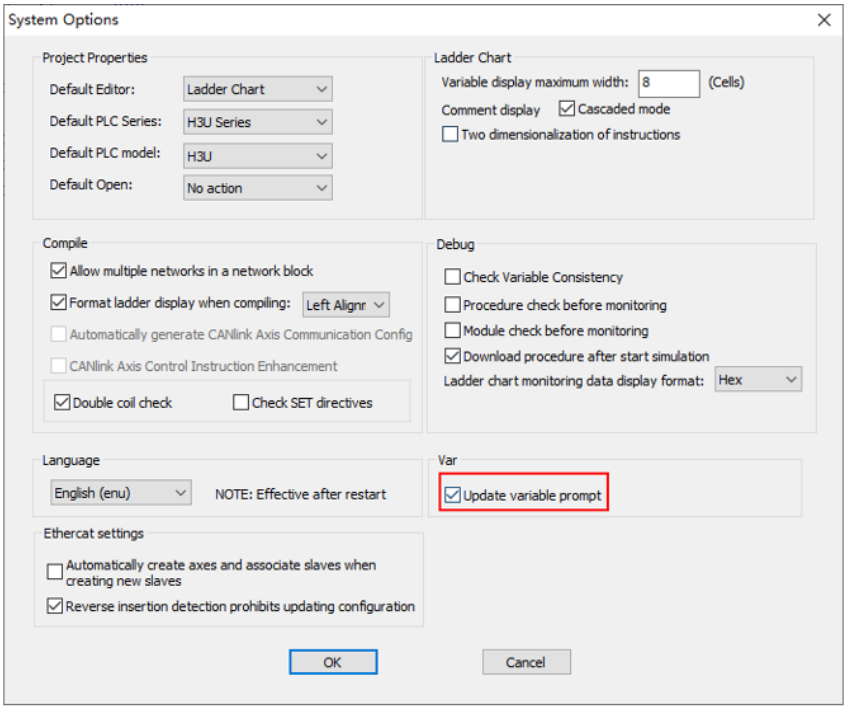
Type	Capacity	Data type	Description
Pointer	4096 points (32-bit)	BOOL/INT/DINT/REAL	Pointer variable Not retained upon power failure
BOOL	2 MB (8-bit)	BOOL/INT/DINT/REAL/IP/STRING/BYTE variable BOOL/INT/DINT/REAL/IP/STRING/BYTE array BOOL/INT/DINT/REAL/IP/STRING/BYTE compound structure	256 KB data retained upon power failure Other data not retained power failure
INT			
DINT			
REAL			

Type	Capacity	Data type	Description
IP			
STRING			
BYTE			

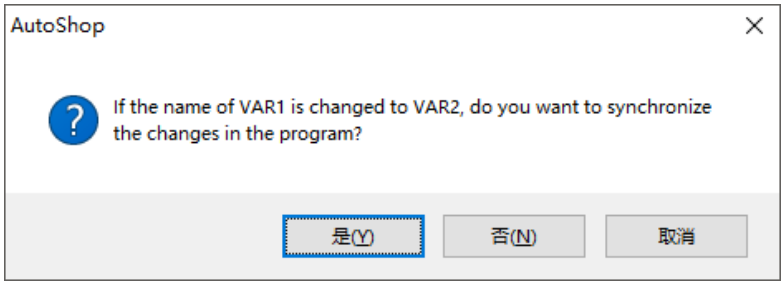
5.3.2 Defining Variables

Introduction

- The PLC supports customized variables. You can define a global variable and directly use the variable name during programming.
  Abide by the following rules when naming a global variable:
  - It contains only letters, digits, and underscores (_) and does not start with a digit or underscore (_).
  - It is not the same as the name of an element, constant, standard data type, instruction, subprogram, or interrupt subprogram.
  - It cannot be keywords such as ARRAY, TRUE, FALSE, ON, OFF, and NULL.
- Update variable prompt
  - The variable name is modified directly by default.
  - Variable definition prompt.
    Choose "Tools" > "System Options" in the menu bar. In the "System Options" window displayed, select "Update variable prompt".



When you modify the variable name, a window will pop up to prompt synchronous changes. Click "Yes" to modify the variable name.



NOTE

This function requires firmware of V6.5.0.0 or later and software of AutoShop V4.11.0.0 or later.

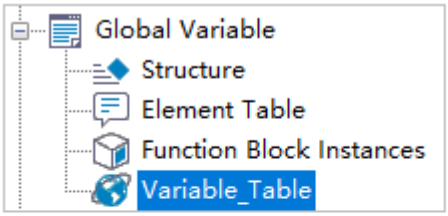
Variable Data Types

Structures and arrays are supported. The following table lists the supported data types.

Data Type	Description
BOOL	Boolean
INT	Single-word integer
DINT	Double word integer
REAL	Real number
STRING	String type
IP	IP
BYTE	Byte

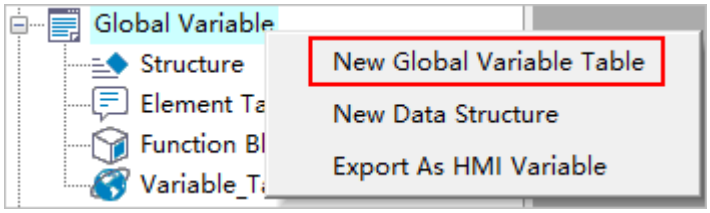
Defining Global Variables

"Global Variable" in the project management window is used for variable management, allowing you to add, delete, and edit variables, as shown in the following figure.



NO.	Variable Name	Data Type	Initial Value	Power Down Hold	Network Pubilc
1	var_1	BOOL	OFF	Non Retained	Private
2	var_2	BOOL	OFF	Non Retained	Private
3	var_3	INT	0	Non Retained	Private
4	var_4	INT	0	Non Retained	Private

1. Add a variable table and variables. Right-click "Global Variable" and choose "New Global Variable Table" to create a global variable table, as shown in the following figure.



2. Double-click the variable table to go to the variable editing page, as shown in the following figure.
   - Edit a variable: Double-click the text box to edit or click the drop-down box to select.
   - Add a variable: Right-click and choose "Insert Row(&I)".
   - Delete a variable: Right-click the row to be deleted and choose "Delete Row(&L)".

NO.	Variable Name	Data Type	Initial Value	Power Down Hold
1	Axis0	Axis		Non Retained
2	Axis1	Axis		Non Retained
3	bMovVel	BOOL	OFF	Non Retained
4	bMovStart	BOOL	OFF	Non Retained
5	bMovStop	BOOL	OFF	Retained
6	bMcReset	BOOL	OFF	Non Retained
7	bMovAbsDone	BOOL	OFF	Non Retained
8	bAbsCmdAbort	BOOL	OFF	Non Retained
9	bAbsBusy	BOOL	OFF	Non Retained
10	bAbsErr	BOOL	OFF	Non Retained
11	iAbsErrID	INT	0	Non Retained
12	bMovRelDone	BOOL	OFF	Non Retained
13	bRelCmdAbort	BOOL	OFF	Non Retained
14	bRelBusy	BOOL	OFF	Non Retained
15	bRelErr	BOOL	OFF	Non Retained
16	iRelErrID	INT	0	Non Retained
17	bAutoRun	BOOL	OFF	Non Retained
18	var_1	BOOL[10]		Non Retained
19	var_2	BOOL	OFF	Non Retained

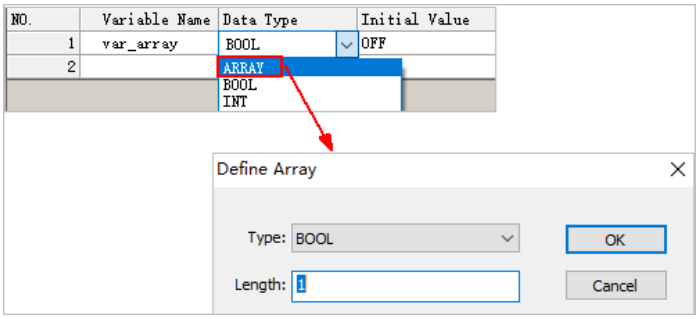
Parameter Name	Description
Variable Name	Custom variable name. You can directly use the variable name for programming.
Data Type	The data types include BOOL, INT, DINT, REAL, IP, STRING, and BYTE variables, BOOL, INT, DINT, REAL, IP, STRING, and BYTE arrays, and BOOL, INT, DINT, REAL, IP, STRING, and BYTE structures. If the data type is an array, you can set the type and length of the array variable in the displayed dialog box. If the data type is a pre-defined structure, you can define a structure variable.
Initial Value	You can assign an initial value to a variable. For arrays and structures, the initial value of each element can be specified individually.
Power Down Hold	"Power Down Hold" can be set to "Non Retained" or "Retained". The specified initial value is valid only when this parameter is set to "Non Retained".
Network Public	This parameter can be set to "Private", "Public", or "In/Out". For structure, specific union, structure array, and specific union array variables, this parameter must be set to "Private". When this parameter is set to "Public", a label configuration file named "LabelConfig.xml" will be generated in the "InteractiveFile" folder under the

Parameter Name	Description
	project directory after project compiling. Importing this configuration file into third-party software enables label communication.

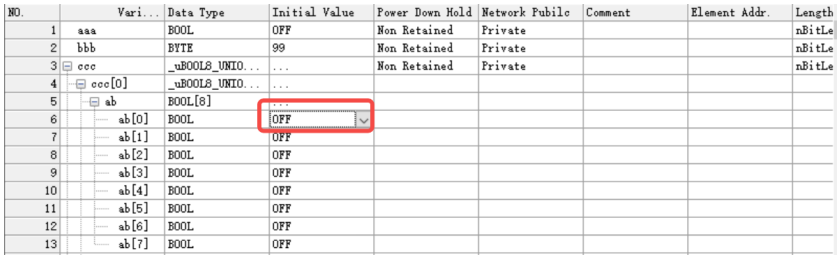
5.3.3 Defining an Array

During user programming, if the data type is set to "ARRAY", an array can be defined.

1. Select the type and length of the array variable in the displayed dialog box and click "OK" to define an array.



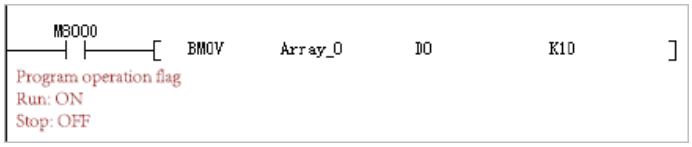
2. Click "+" next to the array variable to edit the initial values and comments of member variables.



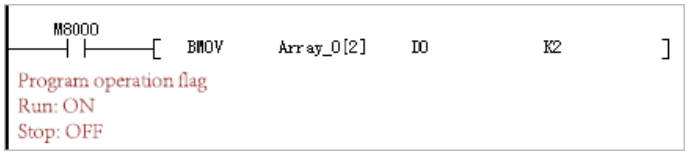
When an array is used in an instruction, if the array subscript is not specified, the access starts from the first element of the array. If the array subscript is specified, the access starts from the element specified by the subscript.

The following are two examples.

- Assign Array_0[0]-Array_0[9] to D0-D9.



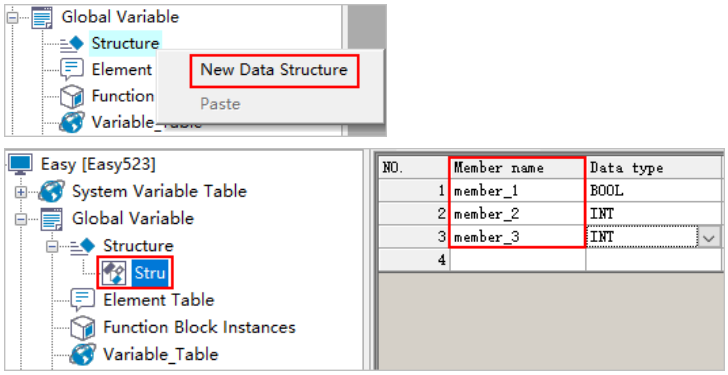
- Assign Array_0[2]-Array_0[3] to D0-D1.



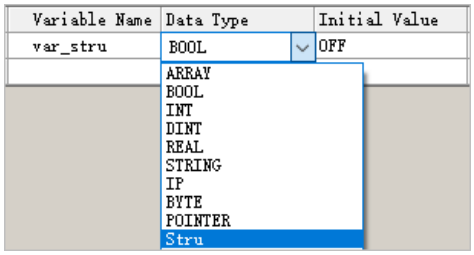
5.3.4 Defining a Structure

To define a structure variable, you need to define the data structure of the structure in advance.

1. Right-click "Structure" under "Global Variable", choose "New Data Structure", and enter a structure name. The structure is defined. When defining a variable in the variable table, you can select this structure as the data type of the variable to define the variable as a structure variable.



2. After the structure and member variables are created, you can select "Stru" in the "Data Type" column to define a structure variable.



3. Click the "Initial Value" column of the structure variable to set the initial values of structure variable members.

5.3.5 Defining Enumerations

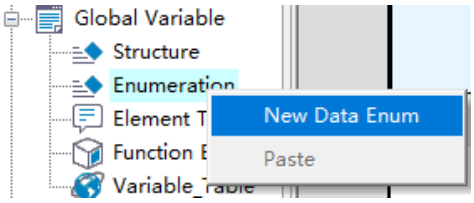
Prerequisites

- Firmware version: V6.5.0.0 and later
- Software version: AutoShop V4.11.0.0 and later

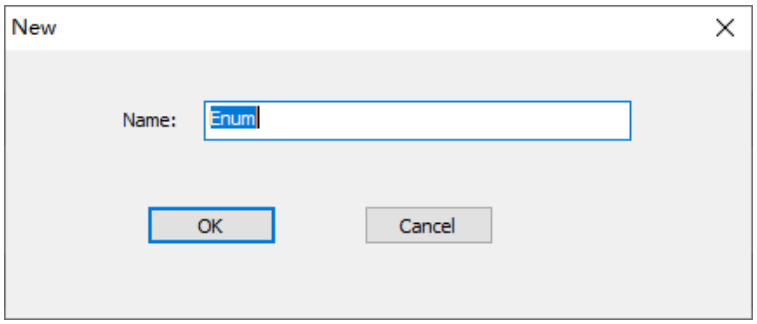
Introduction

To define an enumerated variable, you need to define the data structure of the enumerated variable in advance.

1. Choose "Global Variable" > "Enumeration" in the left-side project management tree and select "New Data Enum". When defining a variable in the variable table, you can select a variable with the data type Enum to define the variable as an enumerated variable.



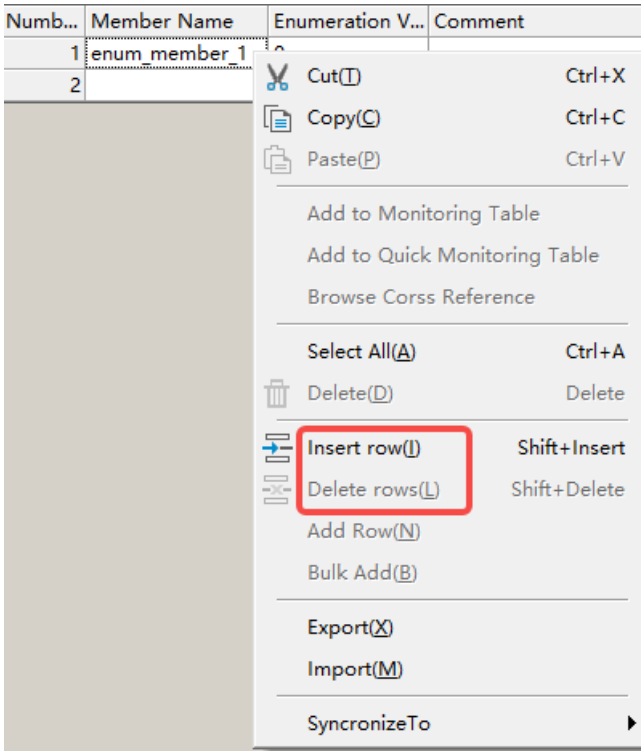
2. In the "New" dialog box, enter the enumeration name and click "OK".



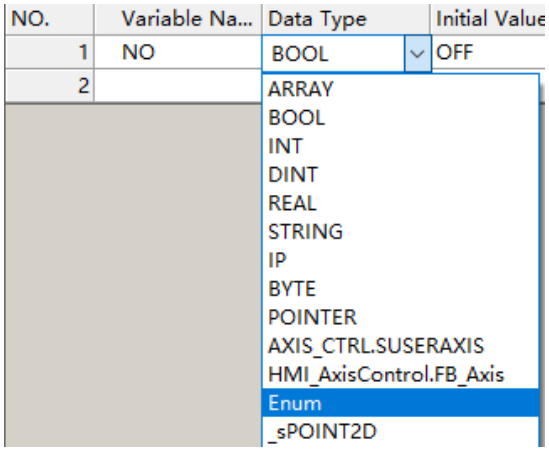
3. The "Enum" sheet displays enumerated variable members of the Enum structure. By default, the member "enum_member_1" and its value of 0 exist.

Numb...	Member Name	Enumeration V...	Comment
1	enum_member_1	0	
2			

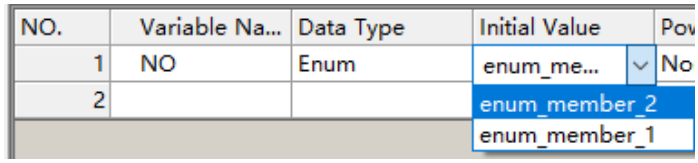
Right-click on the sheet and select "Insert row" in the pop-up menu, or double-click a blank line on the sheet to automatically create an enumeration member. The value of the member increments by 1 automatically. To delete an enumeration member, right-click the target line and select "Delete rows".



4. After an enumeration table is created and members are added to the table, select "Enum" from the "Data Type" column and define enumerated variables.



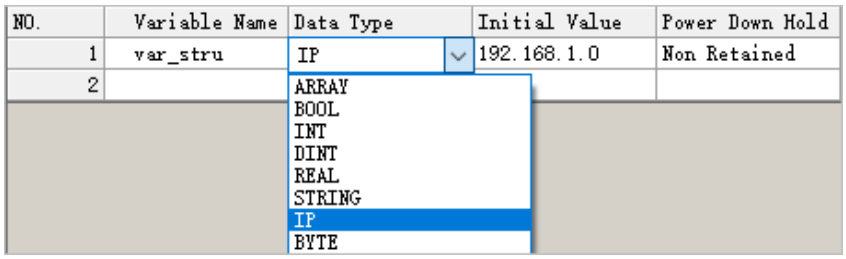
5. You can select a value for an enumerated variable from the "Initial Value" drop-down list. If no value is selected, the first option in the drop-down list is used by default.



5.3.6 Defining an IP Variable

You can define IP variables in the variable table or program. An IP variable occupies 32 bits, and the default value is "192.168.1.0".

- Select "IP" from the "Data Type" drop-down list.



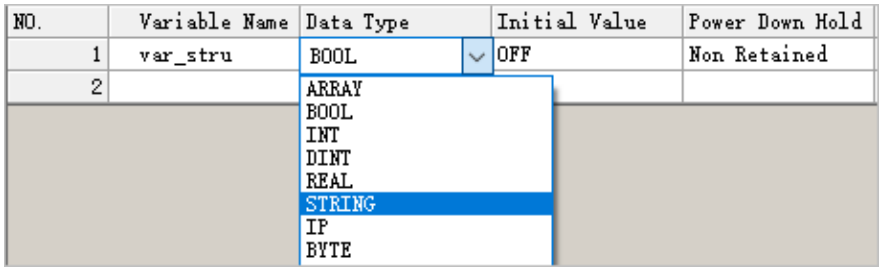
- Use an IP variable in the LiteST program, and assign a value to the IP variable by using single quotation marks.

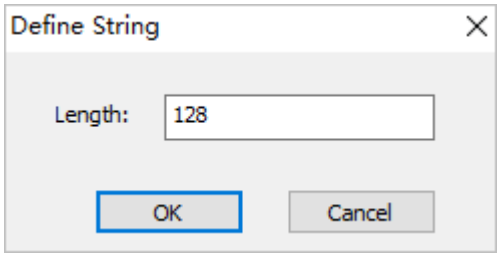
```
1  var_1:='10.45.121.90';
```

5.3.7 Defining a String Variable

You can define string variables in the variable table or program.

- Select "STRING" from the "Type" drop-down list of the variable table, and set the length of the string in the displayed dialog box.
  The default length is 128 bytes and the maximum length is 256 bytes. The last byte is the terminator by default.





- Use a string variable in the LiteST program, and assign a value to the string variable by using single quotation marks.

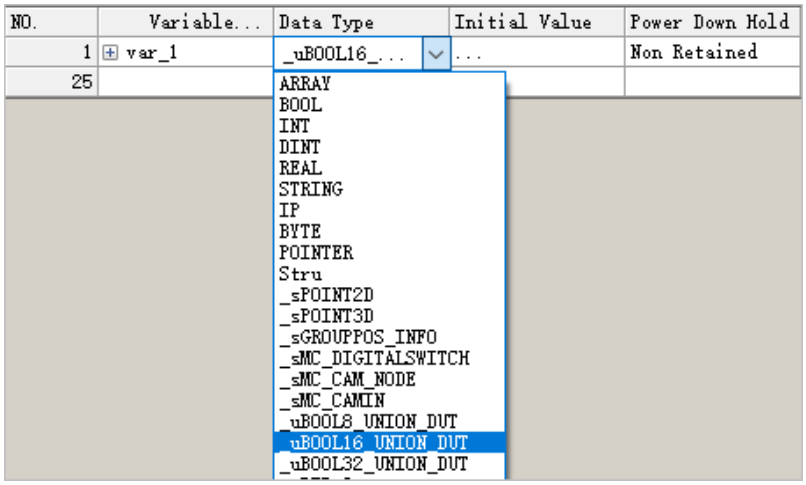
```
1  var_1:='abc';
```

5.3.8 Defining a Specific Union

A specific union is similar to a structure in that they both are collections of different types of elements. The difference lies in the fact that each member of a structure has its own independent storage space, while the members of a specific union share the same memory space (which is why a specific union is called a union). The members are not used simultaneously, but rather one after another.

You can define specific union variables in the variable table or program. There are three types of specific union variables: _uBOOL8_UNION_DUT, _uBOOL16_UNION_DUT, and _uBOOL32_UNION_DUT, corresponding to lengths of 1 byte, 2 bytes, and 4 bytes, respectively.

- Select the required specific union variable type from the “Type” drop-down list of the variable table.



Take _uBOOL32_UNION_DUT as an example. Create a variable in the variable table, and select “_uBOOL32_UNION_DUT” from the “Type” drop-down list.

NO.	Variab...	Data Type	Initial Value	Power Down Hold	Network Public	Comment	Element Addr.	Length
1	var_1	_uBOOL32_UNI...	...	Non Retained	Private		D0	nBitLen:3
2	ab	BOOL[32]	...					
35	ai	INT[2]	...					
36	ai[0]	INT	0				D0	
37	ai[1]	INT	0				D1	
38	abyte	BYTE[4]	...					
39	abyte[0]	BYTE	0				D0	
40	abyte[1]	BYTE	0				D0.8	
41	abyte[2]	BYTE	0				D1	
42	abyte[3]	BYTE	0				D1.8	
43	byte0	BYTE	0				D0	
44	byte1	BYTE	0				D0.8	
45	byte2	BYTE	0				D1	
46	byte3	BYTE	0				D1.8	
47	i0	INT	0				D0	
48	i1	INT	0				D1	
49	f0	REAL	0.000000				D0	
50	di0	DINT	0				D0	
51								

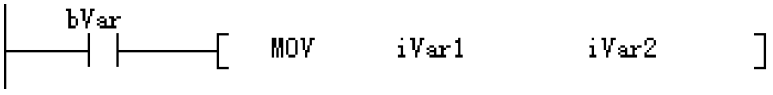
- In a program, you can access different members of a specific union variable by using the dot operator (“.”). This allows you to parse variables in different scenarios.

```
IF eDataType=1 THEN
    var_1.byte0 := 200;
ELSIF eDataType=2 THEN
    var_1.i0 := 2000;
ELSIF eDataType=3 THEN
        var_1.f0 := 2000.22;
END_IF;
```

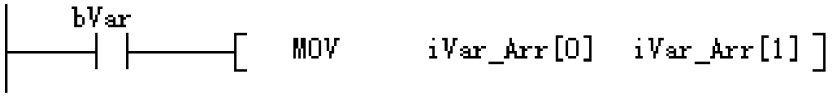
5.3.9 Using the Variable

After a variable is defined, you can directly use the variable name for programming without assigning elements.

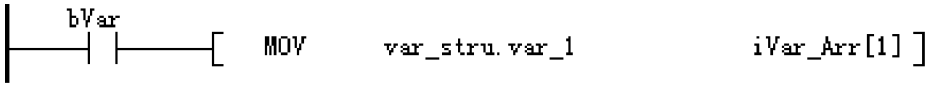
- When a common variable is used, directly use the variable name during programming.



- When an array variable is used, use [Number] to indicate an array element during programming. The number starts from 0.



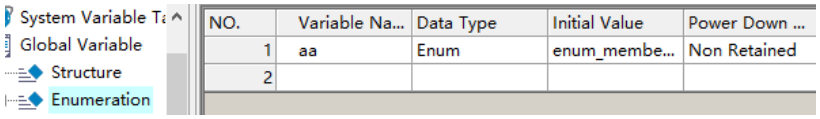
- When a structure variable is used, use Structure variable name.Member variable to indicate a structure member during LiteST programming.



- Pay attention to the following points when using an enumerated variable:
  - Directly access the corresponding enumerated member during programming.

```
1    aa:=enum_member_2;
```

    If aa is a variable of the Enum type, it must be defined in the variable table. Note that if aa is of another type, such as INT or DINT, enumerated value assignment is not supported.



NO.	Variable Na...	Data Type	Initial Value	Power Down ...
1	aa	Enum	enum_membe...	Non Retained
2				

- ◦ For a variable of the Enum type, only the enumerated values can be assigned.

- When an IP or string variable is used, use a value enclosed in a pair of single quotation marks ('Value') to indicate the value of the variable.

```
1  var_1:='10.45.121.90';
```

```
1  var_1:='abc';
```

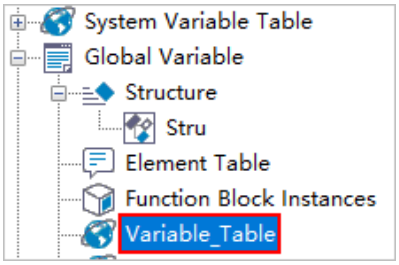
For BYTE, INT, and DINT variables and arrays, you can perform bit operations using the syntax "Variable_name.digit" in programming. For details, see 5.2.2 Word Elements.

5.4 Binding Variables to Element Addresses

5.4.1 Introduction

Customized variables can be bound to soft element addresses so that customized variable addresses are associated with soft element addresses.

You only need to enter the target address in the address column in the variable table and compile the project. Then the software will automatically generate an address for the customized variable.



NO.	Variable Name	Data Type	Initial Value	Power Down Hold	Network Public	Comment	Element Addr.	Length
1	Test_1	BOOL	OFF	Retained	Private		M1000	nBitLen:1
2	Test_2	INT	0	Non Retained	Private		D100	nBitLen:16
3	Test_3	BOOL	OFF	Non Retained	Private		Y0	nBitLen:1

5.4.2 Variable Properties

A customized variable turns retentive or non-retentive upon power failure according to the bound soft element.

As shown in the following figure, M1000 is in the area of retention upon power failure, so Test_1 is retentive upon power failure after being bound to it; D100 is in the area of non-retention upon power failure, so Test_2 is non-retentive upon power failure after being bound to it.

The variable automatically turns retentive or non-retentive at power failure after being bound to an element.

NO.	Variable Name	Data Type	Initial Value	Power Down Hold	Network Pubilc	Comment	Element Addr.	Length
1	Test_1	BOOL	OFF	Retained	Private		M1000	nBitLen:1
2	Test_2	INT	0	Non Retained	Private		D100	nBitLen:16
3	Test_3	BOOL	OFF	Non Retained	Private		Y0	nBitLen:1

5.4.3 Binding a Basic Variable to an Element

- A BOOL variable consists of one bit and can be bound only to a bit element. An INT variable consists of 16 bits and can be bound to one word element. A DINT or REAL variable consists of 32 bits and can be bound to two consecutive word elements.
- A BYTE variable consists of eight bits, and a word element consists of 16 bits. Therefore, a BYTE variable occupies only the low-order 8 bits of a word element upon binding.
- An IP variable consists of 32 bits, occupying two consecutive word elements.
- A STRING variable consists of a customized number of bits, such as two bits. For example, if the variable consists of five bits, it occupies six bytes. After it is bound to soft element D0, it occupies D0, D1, and D2.

5.4.4 Binding an Array Variable to an Element

To bind an array variable to a soft element, enter the address to be mapped in the address column in the variable table.

- A word variable occupies a specified number of word elements based on the variable type. One INT variable occupies one 16-bit element, while a REAL or DINT variable or an IP variable occupies two 16-bit elements.
- Each member of a BYTE array variable occupies half a word element. For example, when a BYTE[5] variable is bound to D0, the first member is bound to the low-order 8 bits of D0 and the second member is bound to the high-order 8 bits of D0. Other elements are bound like this in sequence.
- Each member of a STRING array variable contains two bytes and occupies word elements in sequence.
- A BOOL variable occupies a specified number of bit elements.
- An array variable can be only bound to soft elements of corresponding types. That is, word variables can be only bound to word elements and bit variables can be only bound to bit elements.

For example, if a BOOL array variable is defined as Array_0 to be bound to the M0 element and the length is 10, the variable occupies elements M0 to M0-M9. If an INT array variable is defined as Array_1 to be bound to the D0 element and the length is 10, the variable occupies elements D0 to D0-D9.

5.4.5 Binding a Structure Variable to an Element

AutoShop earlier than V4.0.0.0 does not allow to bind the structure to soft elements. This section describes how to bind a project customized by AutoShop earlier than V4.0.0.0 to a variable after AutoShop is upgraded to V4.0.0.0, and how to bind a project customized by AutoShop 4.0.0.0 or later to a soft element.

When binding a structure variable to a soft element, enter a word element address to be mapped in the address column in the variable table, and click "Compile". AutoShop will automatically generate the address of the structure member. The address assignment rules are as follows:

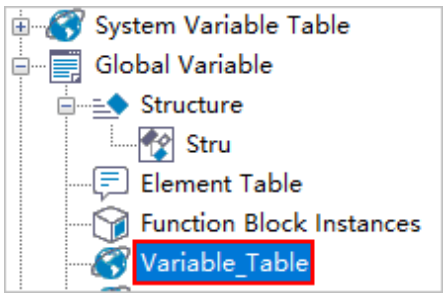
1. An INT variable occupies one 16-bit element, while a REAL or DINT variable occupies two 16-bit elements.
2. Multiple consecutive BOOL variables are considered to occupy one 16-bit element as a whole, and are assigned with an address in sequence from bit 0 of the 16-bit element. Multiple non-consecutive BOOL variables are considered to occupy one 16-bit element separately.
3. Arrays and structure variables occupy one 16-bit element as a whole.

For example, the Stru_0 variable of the Stru type is defined and the D1000 element is bound.

No.	Member Variable Name	Data Type
1	member_1	BOOL
2	member_2	INT[2]
3	member_3	BOOL[2]
4	member_4	DINT
5	member_5	REAL
6	member_6	INT
7	member_7	BOOL

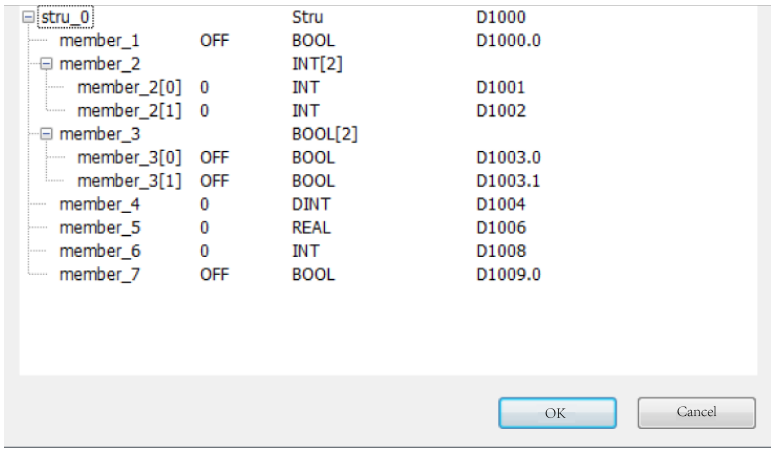
- Stru_0.member_1//The type is BOOL, so D1000.0 is bound.
- Stru_0.member_2//The type is INT array, so D1001 and D1002 are bound.
- Stru_0.member_3//The type is BOOL array, so D1003.0 and D1003.1 are bound.
- Stru_0.member_4//The type is DINT, so D1004 is bound.
- Stru_0.member_5//The type is REAL, so D1006 is bound.
- Stru_0.member_6//The type is INT, so D1008 is bound.
- Stru_0.member_7//The type is BOOL, so D1009.0 is bound.

1. To bind a project customized by AutoShop 4.0.0.0 and later, enter the address to be mapped in the address column in the variable table.



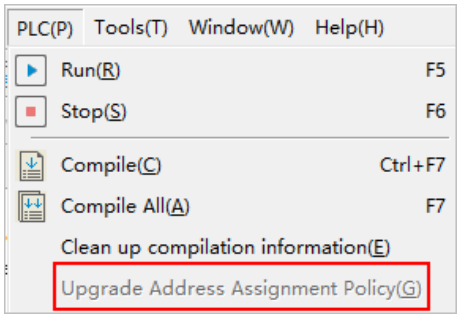
NO.	Variab...	Data Type	Initial Value	Power Down Hold	Network Public	Comment	Element Addr.	Length
1	Test_1	BOOL	OFF	Retained	Private		M1000	nBitLen:1
2	Test_2	INT	0	Non Retained	Private		D100	nBitLen:16
3	Test_3	BOOL	OFF	Non Retained	Private		Y0	nBitLen:1
4	⊞ Stru_0	Stru_0	...	Retained	Private		D1000	nBitLen:160

Compile the project. An address will be automatically generated. Then, double-click the initial value of the corresponding structure variable in the variable table to view the mapping address of each member in the structure and set the variable input value.



2. The address assignment policy for projects customized by AutoShop earlier than V4.0.0.0 is different. To bind a structure variable for such a project, upgrade the address assignment policy.
   The procedure is as follows: Open the project, switch to the variable table page, click "PLC" in the menu bar, and select "Upgrade Address Assignment Policy", as shown in the following figure.

> **NOTE**
>
> The upgrade can result in change to the original addresses that are automatically assigned. If the monitoring variable table is used for communication with the HMI, the information of the monitoring variable table needs to be updated to the HMI. If the element binding method is used, the addresses will not be affected.



After the upgrade, the method of binding a structure variable to a soft element is the same as that later than AutoShop 4.0.0.0. For details, see the preceding procedure.

5.4.6 Binding Enumeration Variables to Soft Elements

The address assignment for the enumeration type is the same as that for DINT. One type occupies two 16-bit elements. It occupies 32 bits and can be bound to two consecutive word elements.

5.4.7 Binding a Specific Union Variable to an Element

Specific union variables are classified into three types, and specific union members obtain addresses at a fixed offset.

- A uBOOL8_UNION_DUT variable consists of eight bits and occupies the low-order 8 bits of a word element. Members of the variable are bound to soft elements in sequence based on the offset.
- A uBOOL16_UNION_DUT variable consists of 16 bits and occupies one word element. Members of the variable are bound to soft elements in sequence based on the offset.

	Variable ...	Data Type	Initial V...	Power Dow...	Network P...	Comment	Element A...
1	⊟ var_1	_uBOOL8_UNION_DUT	...	Non Retained	Private		D0
2	⊟ ab	BOOL[8]	...				
3	ab[0]	BOOL	OFF				D0.0
4	ab[1]	BOOL	OFF				D0.1
5	ab[2]	BOOL	OFF				D0.2
6	ab[3]	BOOL	OFF				D0.3
7	ab[4]	BOOL	OFF				D0.4
8	ab[5]	BOOL	OFF				D0.5
9	ab[6]	BOOL	OFF				D0.6
0	ab[7]	BOOL	OFF				D0.7
1	byte0	BYTE	0				D0

	Variable ...	Data Type	Initial V...	Power Dow...	Network P...	Comment	Element A...
1	⊟ var_1	_uBOOL16_UNION_DUT	...	Non Retained	Private		D0
2	⊟ ab	BOOL[16]	...				
3	ab[0]	BOOL	OFF				D0.0
4	ab[1]	BOOL	OFF				D0.1
5	ab[2]	BOOL	OFF				D0.2
6	ab[3]	BOOL	OFF				D0.3
7	ab[4]	BOOL	OFF				D0.4
8	ab[5]	BOOL	OFF				D0.5
9	ab[6]	BOOL	OFF				D0.6
10	ab[7]	BOOL	OFF				D0.7
11	ab[8]	BOOL	OFF				D0.8
12	ab[9]	BOOL	OFF				D0.9
13	ab[10]	BOOL	OFF				D0.10
14	ab[11]	BOOL	OFF				D0.11
15	ab[12]	BOOL	OFF				D0.12
16	ab[13]	BOOL	OFF				D0.13
17	ab[14]	BOOL	OFF				D0.14
18	ab[15]	BOOL	OFF				D0.15
19	⊟ abyte	BYTE[2]	...				
20	abyte[0]	BYTE	0				D0
21	abyte[1]	BYTE	0				D0.8
22	i0	INT	0				D0
23	byte0	BYTE	0				D0
24	byte1	BYTE	0				D0.8

- A uBOOL32_UNION_DUT variable consists of 32 bits and occupies two consecutive word elements. Members of the variable are bound to soft elements in sequence based on the offset. For example, the member var_1 is bound to soft element D0. The address offset of other BOOL array members is 0, so the initial address ab[0] is bound to D0.0, ab[15] to D0.15, and ab[31] to D1.15. The address offset of INT array members is 0, so the initial address ai[0] is bound to D0 and ai[1] to D1. The address offset of BYTE array members is 0, so the initial address abyte[0] is bound to D0, abyte[1] to D0.8, and so on.

  The offset is 0 for byte0, 8 bits for byte1, 16 bites for byte2, and 24 bits for byte 3. Therefore, the members are bound to D0, D0.8, D1, and D1.8 in sequence.

  The offset is 0 for i0 and 16 bits for i1. Therefore, the members are bound to D0 and D1 in sequence.

  The offset is 0 for the floating-point number variable f0. Therefore, the variable is bound to the D0 and D1 elements. The offset is 0 for the DINT variable. Therefore, the variable is bound to the D0 and D1 elements.

	Variable ...	Data Type	Initial V...	Power Dow...	Network P...	Comment	Element A...
1	⊟ var_1	_uBOOL32_UNION_DUT	...	Non Retained	Private		D0
2	⊞ ab	BOOL[32]	...				
35	⊞ ai	INT[2]	...				
38	⊞ abyte	BYTE[4]	...				
43	byte0	BYTE	0				D0
44	byte1	BYTE	0				D0.8
45	byte2	BYTE	0				D1
46	byte3	BYTE	0				D1.8
47	i0	INT	0				D0
48	i1	INT	0				D1
49	f0	REAL	0.000000				D0
50	di0	DINT	0				D0

5.5 Variable Used as the Variable Group Subscript

5.5.1 Rules of Usage

NOTE

The software of AutoShop V4.11.0.0 and later and firmware of V6.5.0.0 and above support two variables for array subscript/two-dimensional array function.

Only two variables can be used as the subscript in the variable group.

The format is defined as array[index1][index2] or stru[index1].var[index2]. In the format, array indicates an array or a structure array, index, index2, var, and i are variables, and stru indicates the structure.

Basic combination types

- Array variables only support bit variable arrays, word variable arrays, doubleword variable arrays, and float point variable arrays, but not pointer variables.
- Index variables, as array subscript variables, support single-word INT variable (16-bit) and doubleword DINT variable (32-bit), but not soft elements or other variables such as bit variables, float point variables, or pointer variables. A specified element of an array or a specified member of the structure can be used as an index variable, such as array[index[5]] and array[stru.index]. An array element with a variable sub-index or an array member of the structure cannot be used as index variables, such as array[index[i]] and array[stru[i].index].

Complex combination types

- Array elements can be used as operands of instructions, with the index variable at the end, such as array[index], stru. Array[index], stru1[3].stru2. Array[index], and stru1.stru2.stru3. array[index].
- Members of structure arrays can be used as operands of instructions, with the index variable in the middle, such as stru[index].var, stru1[index].stru2.var, and stru1.stru2[5].stru3[index].array[3].
- Enumeration type is supported.
- One-dimensional and two-dimensional arrays are supported.
- One two-domensional array supports up to 10240 elements.
- The format of two-dimensional array is: array name [subscript, subscript].

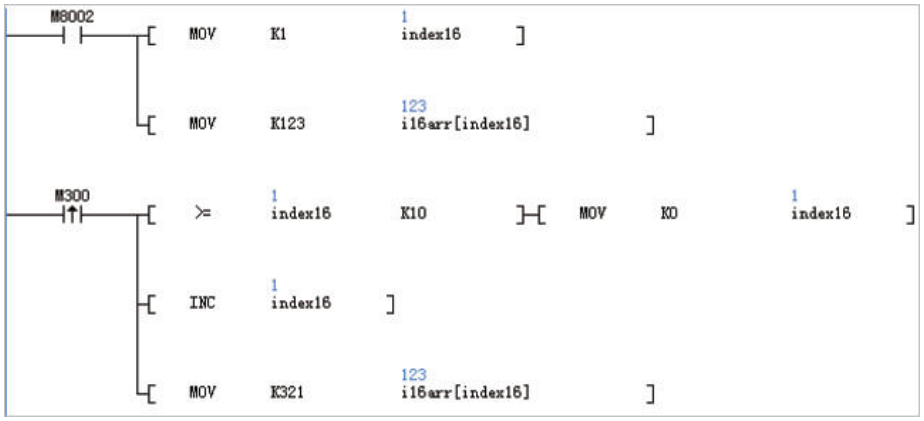
NOTE

- Operands of the ZSET/ZRST instructions do not support arrays that use variables as subscripts.
- Operands of the PTxxx instructions do not support arrays that use variables as subscripts.
- Operands of the SFC instructions do not support arrays that use variables as subscripts.
- For operands (array type operands) of instructions (such as the BMOV instruction for batch assignment) that use multiple consecutive variables, variables in the arr[index] format can be used, but elements of a structure array in the stru[index].var format must not be used (because they are not consecutive). If jump assignment is required, a loop instruction must be used to achieve batch jump assignment.
- This function is mainly used in operands of single-cycle instructions and is not recommended for operands of multi-cycle instructions. If it is necessary to use this function in operands of multi-cycle instructions, logic and timing must be strictly controlled. In case of poor timing control, abnormal execution or conflict may occur upon value switchover (such as the pulse output instruction axis).

5.5.2 Programming Example

Example 1

The program for assigning a value for an element in an array is shown in the following figure.



After startup, value 123 is assigned for i16arr[1]. Then, M300 assigns value 321 for the next array element upon each trigger. The following figure shows the result upon startup.

i16arr	INT[10]		
i16arr[0]	INT	Dec	0
i16arr[1]	INT	Dec	123
i16arr[2]	INT	Dec	0
i16arr[3]	INT	Dec	0
i16arr[4]	INT	Dec	0
i16arr[5]	INT	Dec	0
i16arr[6]	INT	Dec	0
i16arr[7]	INT	Dec	0
i16arr[8]	INT	Dec	0
i16arr[9]	INT	Dec	0

Result upon the first trigger by M300

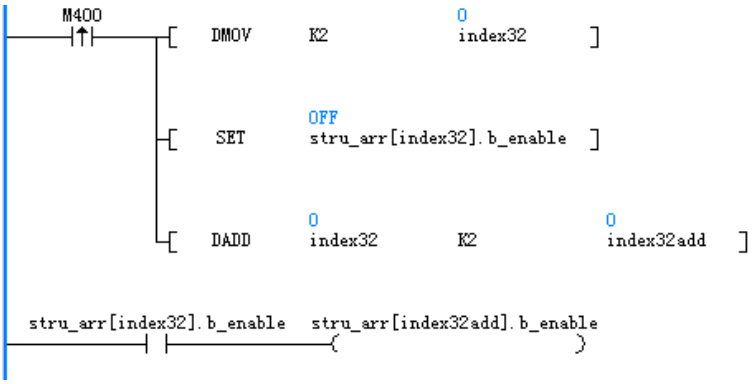
i16arr	INT[10]		
i16arr[0]	INT	Dec	0
i16arr[1]	INT	Dec	123
i16arr[2]	INT	Dec	321
i16arr[3]	INT	Dec	0
i16arr[4]	INT	Dec	0
i16arr[5]	INT	Dec	0
i16arr[6]	INT	Dec	0
i16arr[7]	INT	Dec	0
i16arr[8]	INT	Dec	0
i16arr[9]	INT	Dec	0

Result after multiple triggers by M300

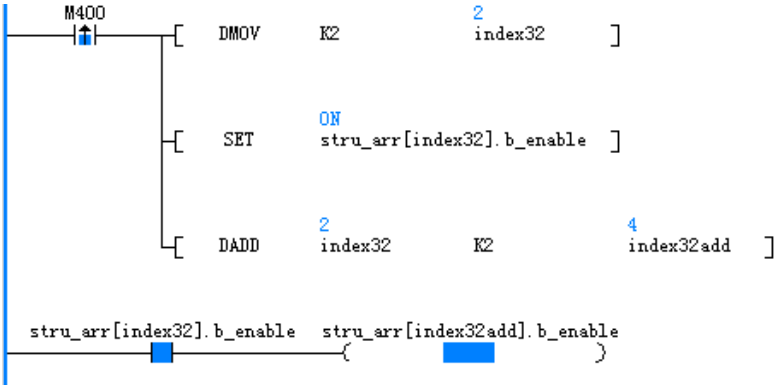
i16arr	INT[10]		
i16arr[0]	INT	Dec	0
i16arr[1]	INT	Dec	123
i16arr[2]	INT	Dec	321
i16arr[3]	INT	Dec	321
i16arr[4]	INT	Dec	321
i16arr[5]	INT	Dec	321
i16arr[6]	INT	Dec	0
i16arr[7]	INT	Dec	0
i16arr[8]	INT	Dec	0
i16arr[9]	INT	Dec	0

Example 2

The program for operating a member variable of a structure array is shown in the following figure.



After a trigger by M400, stru_arr[2].b_enable is set and stru_arr[4].b_enable is controlled based on the status of stru_arr[2].b_enable.



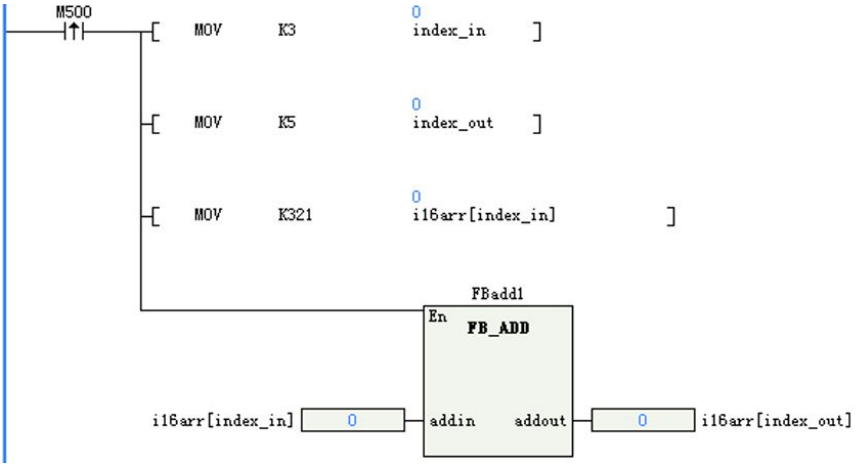
stru_arr	Stru1[5]		
stru_arr[0]	Stru1		
b_enable	BOOL	Bin	OFF
i16_a	INT	Dec	0
i32_b	DINT	Dec	0
stru_arr[1]	Stru1		
b_enable	BOOL	Bin	OFF
i16_a	INT	Dec	0
i32_b	DINT	Dec	0
stru_arr[2]	Stru1		
b_enable	BOOL	Bin	ON
i16_a	INT	Dec	0
i32_b	DINT	Dec	0
stru_arr[3]	Stru1		
b_enable	BOOL	Bin	OFF
i16_a	INT	Dec	0
i32_b	DINT	Dec	0
stru_arr[4]	Stru1		
b_enable	BOOL	Bin	ON
i16_a	INT	Dec	0
i32_b	DINT	Dec	0

The following figure defines the structure.

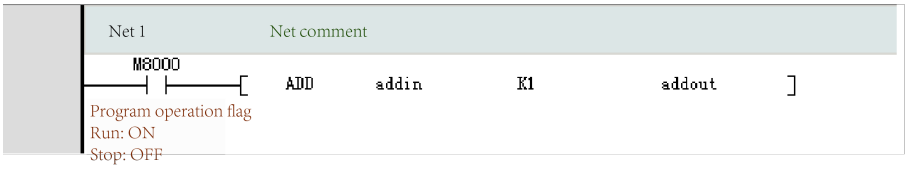
1	b_enable	BOOL	
2	i16_a	INT	
3	i32_b	DINT	

Example 3

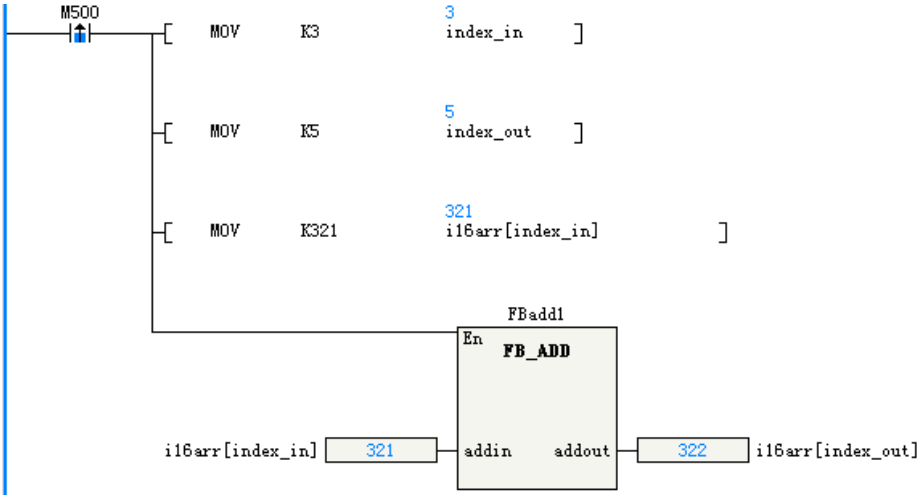
The program for using a variable as the array subscript for FB parameters is shown in the following figure.



FB program



After a trigger by M500, value 321 is assigned for i16arr[3] and then for i16arr[5] after FB calculation. The FB program adds 1 to the input parameter value and then assigns the new value for the output.



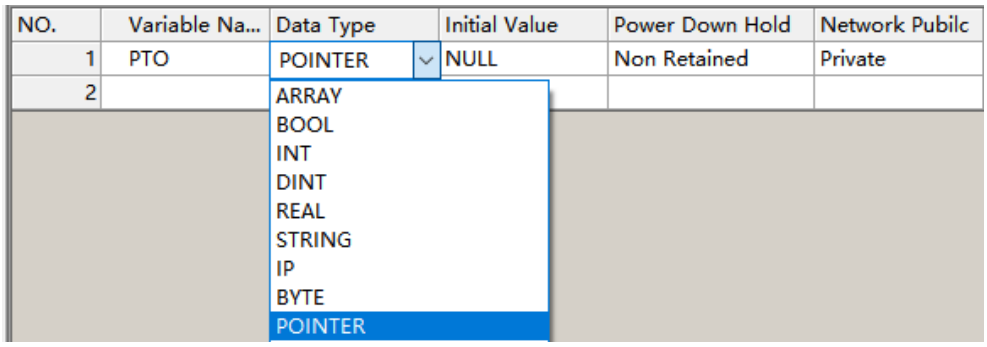
i16arr	INT[10]		
i16arr[0]	INT	Dec	0
i16arr[1]	INT	Dec	0
i16arr[2]	INT	Dec	0
i16arr[3]	INT	Dec	321
i16arr[4]	INT	Dec	0
i16arr[5]	INT	Dec	322
i16arr[6]	INT	Dec	0
i16arr[7]	INT	Dec	0
i16arr[8]	INT	Dec	0
i16arr[9]	INT	Dec	0

5.6 Pointer Type Variables

5.6.1 Introduction

Pointer type variables can be used as addresses of pointer storage soft elements or array variables. During programming, pointer type variables can be used for indirect addressing or indexed addressing. H5U and Easy series PLCs do not support the V and Z soft element functions.

In the variable table, after the variable name is defined and POINTER is selected as the data type, a pointer type variable is defined. The initial value of the pointer variable is NULL, indicating a null pointer, and the pointer variable is non-retentive upon a power failure.



Pointer type variables allow address operations and indirect addressing operations. Instruction of pointer address operations specify the address operations of pointers. The following table lists the instruction of pointer address operations. These instructions can be run to obtain addresses, realize offsets for pointer addresses, and compare pointer addresses.

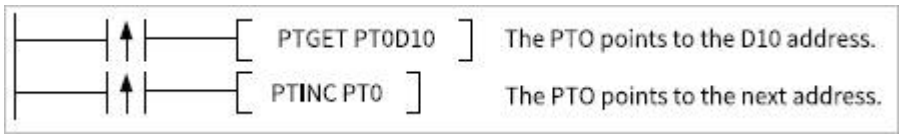
Instruction	Description
PTGET	Obtaining pointer addresses

Instruction	Description
PTINC	Pointer variable address incremented by 1
PTDEC	Pointer variable address decremented by 1
PTADD	Adding offsets for pointer variable addresses
PTSUB	Deducting offsets for pointer variable addresses
PT>, PT>=, PT<, PT<=, PT=, and PT<>	PT variable contact comparison
PTMOV	Pointer variable mutual assignment

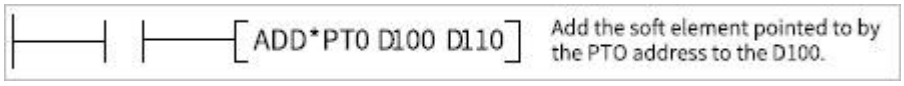
For other instructions, when pointer type variables are used, indirect addressing operations are performed on pointer type variables, indicating the operations on the values of soft elements or array variables directed by the pointer type variables. Indirect addressing operations on pointer type variables are indicated by "*Pointer type variable" during programming.

Example

- Address operations on pointer type variables



- Indirect addressing operations on pointer type variables



NOTE

In programming, in addition to the instructions for pointer address operations in the preceding table, if other instructions use a pointer type variable, the programming software automatically adds an asterisk (*) before the variable. Alternatively, users can enter asterisk (*) manually.

5.6.2 Obtaining the Directing Address

Directing addresses of pointer type variables can be obtained by the pointer variable assignment instruction (PTGET).

Example

When the instruction energy flow is effective, the pointer type variable PT0 directs to D10. That is, PT0 obtains the address of the D10 soft element.

Pointer type variables can direct to bit elements (X, Y, M, S, and B), word elements (D, R, and W), and customized array variables.

5.6.3 Pointer Address Operation

After pointer addresses of pointer type variables are obtained, they can be added or deducted to specify the element offset of pointer type variables.

Example 1

When the instruction energy flow is effective, one soft element offset is added for the PT0 pointer address of the pointer type variable. For example, PT0 originally directed to D10 and then directs to D11 after the PTINC instruction is executed. After PTINC execution, the system automatically adds one offset based on the element or array variable type directed by the pointer type variable.

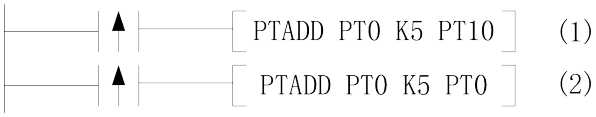
Current PT0 Pointer	PT0 Pointer After PTINC Execution
D10	D11
M200	M201
diVal[3]	diVal[4]

Example 2

When the instruction energy flow is effective, one soft element offset is deducted for the PT0 pointer address of the pointer type variable. For example, PT0 originally directed to D10 and then directs to D9 after the PTINC instruction is executed. After PTDEC execution, the system automatically deducts one offset based on the element or array variable type directed by the pointer type variable.

Current PT0 Pointer	PT0 Pointer After PTDEC Execution
D10	D9
M200	M199
diVal[3]	diVal[2]

Example 3

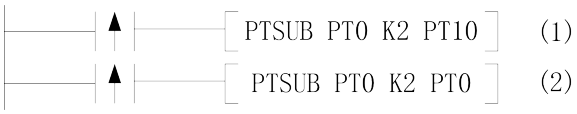


When the energy flow of instruction (1) is effective, five soft element offsets are added for the PT0 pointer address of the pointer type variable and assigned to PT10. For example, PT0 originally directed to D10 and then PT10 directs to D15 after execution of instruction (1). When the energy flow of instruction (2) is effective, five soft element offsets are added for the PT0 pointer address and assigned to PT0. For example, PT0 originally directed to D10 and then directs to D15 after execution of instruction (2).

Current PT0 Pointer	Pointer After Execution of Instruction (1)	

	PT10 Pointer	PT0 Pointer	PT0 Pointer After Execution of Instruction (2)
D10	D15	D10	D15
M200	M205	M200	M205
diVal[3]	diVal[8]	diVal[3]	diVal[8]

Example 4



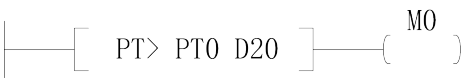
When the energy flow of instruction (1) is effective, two soft element offsets are deducted for the PT0 pointer address of the pointer type variable and assigned to PT10. For example, PT0 originally directed to D10 and then PT10 directs to D8 after execution of instruction (1). When the energy flow of instruction (2) is effective, two soft element offsets are deducted for the PT0 pointer address and assigned to PT0. For example, PT0 originally directed to D10 and then directs to D8 after execution of instruction (2).

Current PT0 Pointer	Pointer After Execution of Instruction (1)	PT0 Pointer	PT0 Pointer After Execution of Instruction (2)
	PT10 Pointer		
D10	D8	D10	D8
M200	M198	M200	M198
diVal[3]	diVal[1]	diVal[3]	diVal[1]

NOTE

In all the preceding examples of pointer address operations, the PTGET instruction must be used to obtain the pointer address first. When PT points to an array variable, pay attention to boundary checking when executing address operation instructions.

Example 5



The PT> instruction determines whether the PT0 pointer address of the pointer type variable is greater than D20. If the PT0 pointer directs to D21, the output M0 is ON. Similar inspections such as PT>=, PT<, PT<=, PT=, and PT<> can determine the directions of PT pointers.

5.6.4 Indirect Addressing Operation

After you obtain an address for a pointer type variable by running address operation instructions, the address can be used in instructions, indicating the indirect addressing operation on the soft element or array variable directed by the pointer variable.

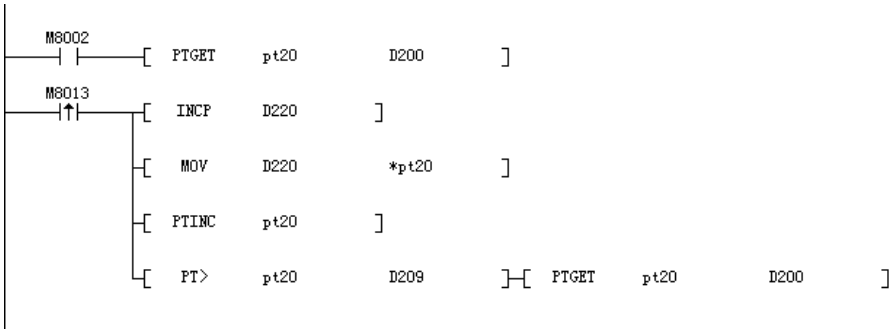
Example

When the instruction energy flow is effective, the soft element directed by the PT0 pointer of the pointer type variable is added to D100. For example, if PT0 directs to D10, the instruction execution result is D200 = D10 + D100.

To indirectly represent a specified soft element using a pointer type variable, an effective pointer address must be obtained by using a pointer address operation instruction first.

5.6.5 Programming Example

This section uses a pointer type soft component as an example. The component cycles the value of D220 to the first 10 elements starting from D200 every 1 second.



NOTE

After a value is assigned to the address pointed to by the pointer, the pointer address is incremented by 1. Therefore, the value for the pointer address monitored by AutoShop in this program is inconsistent with the displayed data.

5.7 System Variables

5.7.1 Introduction

This section describes the PLC operation status using system variables, such as the device model, version number, serial port information, and Ethernet and CAN communication status.

The system variables are described as follows:

SysVar (System Variable Category)	Description
_SYS_CAN	CAN communication information, such as the station No., baud rate, and online state of the slave station

SysVar (System Variable Category)	Description
_SYS_COM	Serial communication information, such as the station No., baud rate, and online state of the slave station
_SYS_ECAT_MASTER	EtherCAT master station status
_SYS_ECAT_SLAVE	EtherCAT slave station status
_EthIPScanner	EtherNet/IP system variable information
_SYS_ENCODER_AXIS	Data structure of the external encoder axis
_SYS_ETHERNET	Ethernet communication information, such as the IP address, MAC address, online state, and error diagnosis.
_SYS_INFO	PLC system information, such as the firmware version, real-time clock (RTC), module diagnosis, and system logs.
_SYS_MC_AXIS	Data structure of the motion control axis
_sGROUPAXIS_INFO	Status of a coordinate axis in the axis group
_sMCGROUP_INFO	Axis group status
_sGROUPPOS_INFO	Target position of a coordinate axis in the axis group
_SYS_PN	PN slave station status

5.7.2 _SYS_CAN CAN Interface Information

Interface Information

- CAN interface information

The software of AutoShop V4.11.0.0 and later and firmware of V6.5.0.0 and later support _CANSet.SetFlag and _CANSet.BaudRate variables.

Name	Data Type	Description	R/W	Comparison with H3U
_CAN.BaudRate	INT	Baudrate (kbps)	R	D8285
_CAN.LoadRate	INT	Load rate (%)	R	D8240
_CAN.RxPerSec	INT	Received frames per second (FPS)	R	D8290

Name	Data Type	Description	R/W	Comparison with H3U
_CAN.TxPerSec	INT	Sent FPS	R	D8291-D8290
_CAN.RxErrCnt	INT	Receive error counter	R	High-order 8 bits of D8989
_CAN.TxErrCnt	INT	Send error counter	R	Low-order 8 bits of D8989
_CAN.Protocol	INT	Communication protocol. 0: CANlink; 1: CANopen	R	D8280
_CANSet.SetFlag	INT	Parameter setting enable flag	R/W	-
_CANSet.BaudRate	INT	Baudrate (kbps)	R/W	-

NOTE

The usage of _CANSet.SetFlag and _CANSet.BaudRate is as follows:

- When the _CANSet.SetFlag is not 0, the baud rate configured by the system variable _CANSet.BaudRate is used. Otherwise, the baud rate configured in the project is used. After the baud rate is changed, the _CAN.BaudRate system variable is updated simultaneously and takes effect.
- Configuration changes take effect upon power-on or during switching from STOP to RUN. Therefore, changing the configuration enable or baud rate during operation will not take effect. After the baud rate is changed, a RUN/STOP cycle is required for the change to become effective.
- The valid input range for the baud rate system variable is: 1000, 800, 500, 250, 125, 100, 50, 20, and 10.
- The configuration via system variables applies to all CAN protocols, including CANopen Master/Slave, CANLink Master/Slave, and CAN Freeform protocol.

- CANLink interface information
  The following table lists CANlink ports.

Name	Data Type	Description	R/W	Comparison with H3U
_CANLink.Address	INT	Station No. or address	R	D8284
_CANLink.Heartbeat	INT	Heartbeat time (ms)	R	D8282
_CANLink.Network Start	BOOL	Network startup	R	M8290
_CANLink.SyncTrigger	BOOL	Sync triggering	R	M8291
_CANLink.SyncWrErr	INT	Synchronous write error code	R	D8307
_CANLink.ConfigErr	INT	Configuration error code	R	D8308

Name	Data Type	Description	R/W	Comparison with H3U
_CANLink.NodeState	INT[64]	Station status (=2: online; ≠2: offline)	R	-
_CANLink.NodeState[0]	INT	Local station status (=2: online; ≠2: offline)	R	D7800
_CANLink.NodeState[1]	INT	1# station status (=2: online; ≠2: offline)	R	D7801
_CANLink.NodeState[…]	INT	-	R	-
_CANLink.NodeState[63]	INT	63# station status (=2: online; ≠2: offline)	R	D7863
_CANLink.Online	DINT[2]	Station status on the CANlink configuration monitoring page of AutoShop, with one bit indicating a station	R	-
_CANLink.Online[0]	DINT	-	R	D8241
_CANLink.Online[1]	DINT	-	R	D8242
_CANLink.Disable	BOOL[64]	Enables (set to ON) or disables (set to OFF) a slave. The array elements of 0 and the master No. are meaningless.	R/W	M8290
_CANLink.Disable[0]	BOOL	-	R/W	-
_CANLink.Disable[1]	BOOL	-	R/W	-
_CANLink.Disable[…]	BOOL	-	R/W	-
_CANLink.Disable[63]	BOOL	-	R/W	-

- CANOpen interface information

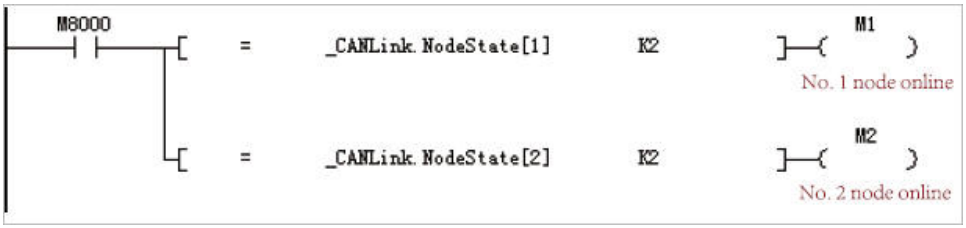
The software of AutoShop V4.11.0.0 and later and firmware of V6.5.0.0 and later support _CANOpen.RestartMaster.

Name	Data Type	Description	R/W	Comparison with H3U
_CANOpen.NodeID	INT	Node ID	R	D8284
_CANOpen.NodeState[0]	INT	Local station status (=5: online; ≠5: offline)	R	D7800
_CANOpen.NodeState[1]	INT	1# station status (=5: online; ≠5: offline)	R	D7801
_CANOpen.EMCY.NodeID	INT	Emergency event node ID	R	-
_CANOpen.EMCY.ErrorCode	INT	Emergency event error code	R	-
_CANOpen.Debug	_sCOP_DEBUG	Commissioning Information	R	-
_CANOpen.Debug.NodeID	INT	Commissioning Information	R	-
_CANOpen.Debug.State	INT	Commissioning Information	R	-
_CANOpen.Debug.Index	INT	Commissioning Information	R	-
_CANOpen.Debug.SubIndexAndSize	INT	Commissioning Information	R	-
_CANOpen.Debug.Data	INT[4]	Commissioning Information	R	-
_CANOpen.Debug.Data[0]	INT	Commissioning Information	R	-
_CANOpen.Debug.Data[1]	INT	Commissioning Information	R	-
_CANOpen.Debug.Data[2]	INT	Commissioning Information	R	-
_CANOpen.Debug.Data[3]	INT	Commissioning Information	R	-
_CANOpen.ConfigError.NodeID	INT	Configuration error node ID	R	D8287
_CANOpen.ConfigError.ConfigIndex	INT	Configuration No.	R	D8288

Name	Data Type	Description	R/W	Comparison with H3U
_CANOpen.ConfigError.ErrorCode	DINT	Error code	R	D8254 and D8255
_CANOpen.RestartMaster	INT	Restart the CANopen master	W	-

Program example

Determination on the online status of CANlink slave stations



5.7.3 _SYS_COM Serial Port Information

Serial port information

- COM serial port information
  The following table lists configuration information of the COM port. Each serial port corresponds to a separate system variable. _COM corresponds to COM0, and _COM1 to _COM15 correspond to COM1 to COM15 respectively.

Name	Data Type	Description	R/W	Comparison with H3U
_COM.BaudRate	DINT	Baudrate (bps)	R	Bit4 to bit7 of D8120
_COM.DataBits	INT	Data bit	R	Bit0 of D8120
_COM.Parity	INT	Parity bit	R	Bit1 and bit2 of D8120
_COM.StopBits	INT	Stop bit	R	Bit3 of D8120
_COM.Interface	INT	Physical Interfaces	R	-

- MbMst Modbus master (serial port) information
  The following table lists Modbus-based information about serial ports of the master station. Each serial port corresponds to a separate system variable. _MbMst corresponds to COM0, and _MbMst1 to _MbMst15 correspond to COM1 to COM15 respectively.

Parameter	Data Type	Description	R/W	Comparison with H3U
_MbMst.Port	INT	Serial port number	R	-

Parameter	Data Type	Description	R/W	Comparison with H3U
_MbMst.Timeout	INT	Timeout interval (ms)	R	D8129*10
_MbMst.Enable	BOOL	Enabled	R	-
_MbMst.Activate	BOOL	Active state	R	-
_MbMst.Busy	BOOL	Busy	R	-
_MbMst.Error	BOOL	Error state	R	M8129
_MbMst.ResponseTime	INT	Response time (ms)	R	-
_MbMst.ErrorSlvAddr	INT	ID of the slave with communication error	R	-
_MbMst.CompatMode	INT	Compatibility mode. 0: Standard 1: Compatible with H0U	R/W	-

- MbMstEx Modbus master (serial port) extension information
  The following table lists MbMstEx Modbus master (serial port) extension information. Each serial port corresponds to a separate system variable. _MbMstEx corresponds to COM0, and _MbMstEx1 to _MbMstEx15 correspond to COM1 to COM15 respectively.

Parameter	Data Type	Description	R/W	Comparison with H3U
_MbMstEx.SlvDisableSetFlag	DINT	Whether to disable slave stations _MbMstEx.SlvDisableSetFlag=0: Not start _MbMstEx.SlvDisableSetFlag≠0: Start	R/W	-
_MbMstEx.SlvDisable	BOOL[256]	Slave station disabled The array number corresponds to the slave station number	R/W	-

Parameter	Data Type	Description	R/W	Comparison with H3U
		_MbMstEx.SlvDisable=ON: Disabled _MbMstEx.SlvDisable=OFF: Not disabled		
_MbMstEx.SlvDisable[0]	BOOL	-	R/W	-
_MbMstEx.SlvDisable[1]	BOOL	-	R/W	-
_MbMstEx.SlvDisable[…]	BOOL	-	R/W	-
_MbMstEx.SlvDisable[255]	BOOL	-	R/W	-
_MbMstEx.RetryTimes	INT	Number of retries (Modbus instructions)	R/W	-

- MbSlv Modbus slave (serial port) information
  The following table lists Modbus-based information about serial ports of the slave station. Each serial port corresponds to a separate system variable. _MbSlv corresponds to COM0, and _MbSlv1 to _MbSlv15 correspond to COM1 to COM15 respectively.

Parameter	Data Type	Description	R/W	Comparison with H3U
_MbSlv.Port	INT	Serial port num	R	-
_MbSlv.SlaveAddress	INT	Slave address	R	D8121
_MbSlv.Connected	BOOL	Connection state	R	-

- Free protocol-based SerialSR serial port information

Parameter	Data Type	Description	R/W	Comparison with H3U
_SerialSR.port	INT	Serial port/Port number	R	-
_SerialSR.states	INT	Operation status	R	-

Parameter	Data Type	Description	R/W	Comparison with H3U
_SerialSR.sent	INT	Number of sent bytes	R	-
_SerialSR.received	INT	Number of received bytes	R	-
_SerialSR.mutexF	DINT	Interlock flag	R	-
_SerialSR.trigger	DINT	Trigger flag	R	-
_SerialSR.errorid	DINT	Error message	R	-
_SerialSR.timeout	DINT	Timeout interval (ms)	R	D8129*10

- Sendbuf to send buffer data

Parameter	Data Type	Description	R/W	Comparison with H3U
_SerialSR.sendlen	DINT	Number of bytes to transmit	R	-
_SerialSR.sendbuf	INT[256]	Transmission buffer	R	-
_SerialSR.sendbuf[0]	INT	-	R	-
_SerialSR.sendbuf[1]	INT	-	R	-
_SerialSR.sendbuf[2]	INT	-	R	-
_SerialSR.sendbuf[…]	INT	-	R	-
_SerialSR.sendbuf[254]	INT	-	R	-
_SerialSR.sendbuf[255]	INT	-	R	-

- Recvbuf to receive buffer data

Parameter	Data Type	Description	R/W	Comparison with H3U
_SerialSR.recvlen	DINT	Number of bytes to receive	R	-
_SerialSR.recvbuf	INT[256]	RX buffer	R	-
_SerialSR.recvbuf[0]	INT	-	R	-
_SerialSR.recvbuf[1]	INT	-	R	-
_SerialSR.recvbuf[2]	INT	-	R	-
_SerialSR.recvbuf[…]	INT	-	R	-
_SerialSR.recvbuf[254]	INT	-	R	-
_SerialSR.recvbuf[255]	INT	-	R	-

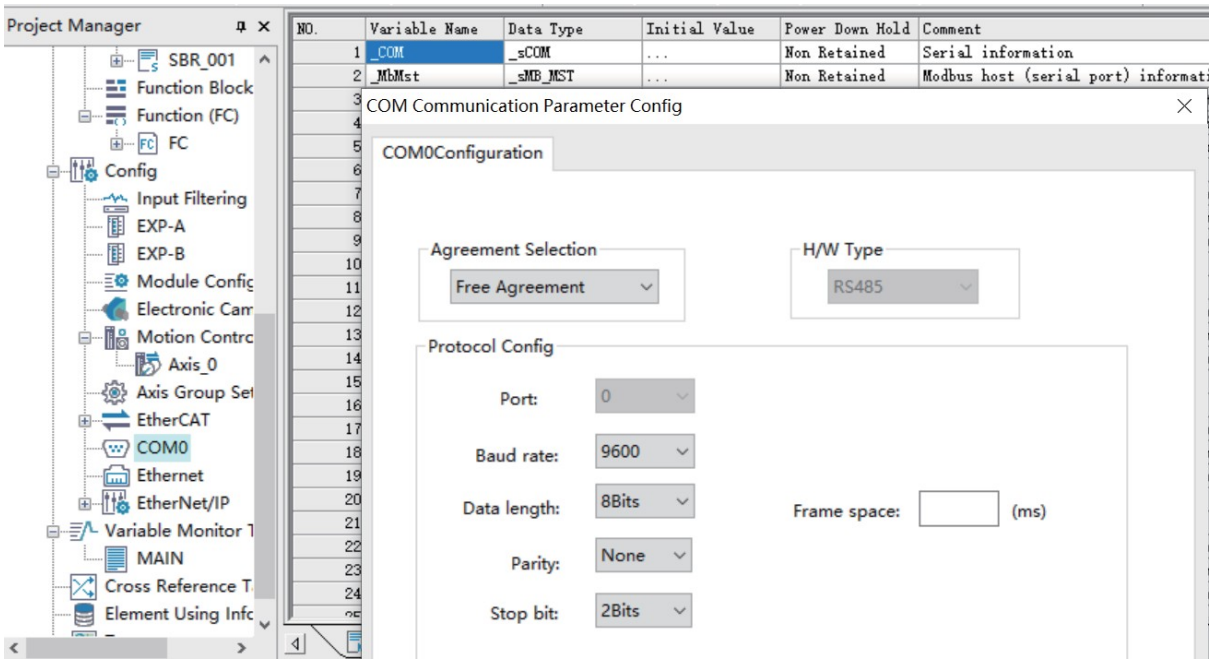
- Configuration data of serial port free protocol-based instruction
  The preceding table lists free protocol-based instruction configuration for a serial port. Each serial port corresponds to an independent system variable: _ SerialSR corresponds to COM0, and _SerialSR1 to _SerialSR15 correspond to COM1 to COM15.

Parameter	Data Type	Description	R/W	Comparison with H3U
_SerialSR.abort	INT	Serial port free protocol canceled	R/W	-
_SerialSR.startchar_en	INT	Reception start character enable (0 to 4)	R/W	-
_SerialSR.startchar	BYTE[4]	Reception start character	R/W	-
_SerialSR.startchar[0]	BYTE	-	R/W	-
_SerialSR.startchar[1]	BYTE	-	R/W	-
_SerialSR.startchar[2]	BYTE	-	R/W	-
	BYTE	-	R/W	-

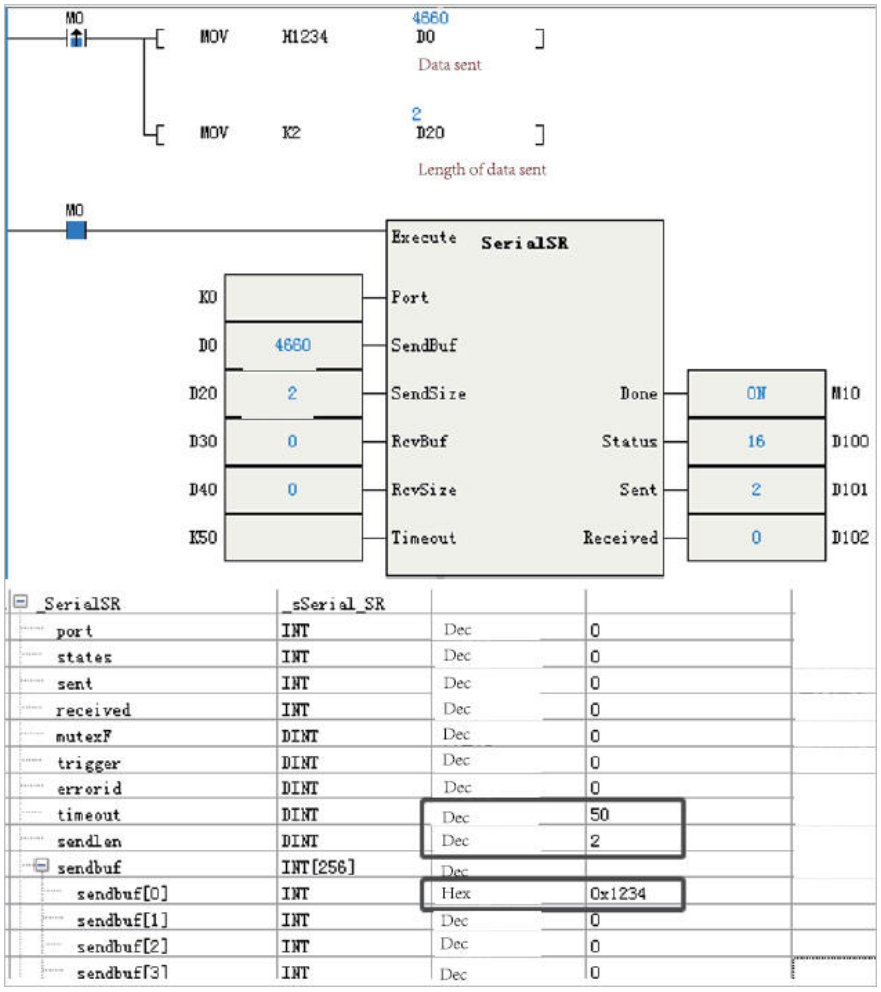
Parameter	Data Type	Description	R/W	Comparison with H3U
_SerialSR.startchar[3]				
_SerialSR.endchar_en	INT	Reception end character enable (0 to 4)	R/W	-
_SerialSR.endchar	BYTE[4]	Reception end character	R/W	-
_SerialSR.endchar[0]	BYTE	-	R/W	-
_SerialSR.endchar[1]	BYTE	-	R/W	-
_SerialSR.endchar[2]	BYTE	-	R/W	-
_SerialSR.endchar[3]	BYTE	-	R/W	-
_SerialSR.Bytetimeout_en	BOOL	Idle interrupted frame reception enable	R/W	-
_SerialSR.Bytetimeout	INT	Duration for judging idle interrupted frame reception (ms)	R/W	-

Program example

1. COM configuration information, as shown in the following figure.



2. SerialSR instruction sending, as shown in the following figure.



5.7.4 _SYS_COM_SAVE Serial Port Parameters

- Setting COMSet serial port parameters

  The following table lists parameter settings of the COM port. Each serial port corresponds to a separate system variable. _COMSet corresponds to COM0, and _COM1Set to _COM15Set correspond to COM1 to COM15 respectively. The parameter settings are valid by switching PLC start/stop status or powering off and on the device.

Name	Data type	Description	R/W	Comparison with H3U
_COMSet.SetFlag	DINT	Parameter setting enable flag 0: Disable Non-zero: Enable	R/W	-
_COMSet.BaudRate	DINT	Baud rate (bps)	R/W	-
_COMSet.DataBits	INT	Data bit	R/W	-
_COMSet.Parity	INT	Parity bit	R/W	-
_COMSet.StopBits	INT	Stop bit	R/W	-
_COMSet.Interface	INT	Physical interface	R/W	-
_COMSet.Protocol	INT	Communication protocol	R/W	-

- Setting _COMProtocolSet serial port parameters

The following table lists protocol-based parameter settings of the COM port. Each serial port corresponds to a separate system variable. _COMProtocolSet corresponds to COM0, and _COM1ProtocolSet to _COM15ProtocolSet correspond to COM1 to COM15 respectively. The parameter settings are valid by switching PLC start/stop status or powering off and on the device.

Name	Data type	Description	R/W	Comparison with H3U
_COMProtocolSet.port	INT	Serial port number	R	-
_COMProtocolSet.AddressSetFlag	INT	Whether to set the slave station number or address	R/W	-
_COMProtocolSet.Address	INT	Slave station number or address	R/W	-

5.7.5 _SYS_ECAT_Master Operation Status

The following table lists information about the EtherCAT master station, such as the master station operation status and maximum cycle time.

Name	Data type	Description	R/W
	BOOL		R

Name	Data type	Description	R/W
_ECATMaster.bMasterRunState		Operation status of the master station (ON: running; OFF: stopped)	
_ECATMaster.bLinkState	BOOL	Monitors the connection status of the master station. ON: Normal OFF: Network cable disconnected	R
_ECATMaster.bHeartBeat	BOOL	EtherCAT real-time task heartbeat	R
_ECATMaster.bBlockHeartBeat	BOOL	EtherCAT non-real-time task heartbeat	R
_ECATMaster.dMaxCycleTime	DINT	Max. cycle time (μs)	R
_ECATMaster.dMinCycleTime	DINT	Min. cycle time (μs)	R
_ECATMaster.dCycleTime	DINT	Cycle time (μs)	R
_ECATMaster.dMaxExeTime	DINT	Max. execution time (μs)	R
_ECATMaster.dMinExeTime	DINT	Min. execution time (μs)	R
_ECATMaster.dExeTime	DINT	Execution time (μs)	R
_ECATMaster.dtx_frames	DINT	Total number of sent frames (frame)	R
_ECATMaster.drx_frames	DINT	Total number of received frames (frame)	R
_ECATMaster.dtx_frame_rates	DINT	Frame sending rate (frame/s)	R
_ECATMaster.drx_frame_rates	DINT	Frame receiving rate (frame/s)	R
_ECATMaster.dtx_bytes_rate	DINT	Byte sending rate (byte/s)	R
_ECATMaster.drx_bytes_rate	DINT	Byte receiving rate (byte/s)	R
_ECATMaster.dloss_frames	DINT	Number of lost EtherCAT frames (frame)	R
_ECATMaster.bResetTime	BOOL		R/W

Name	Data type	Description	R/W
		Execution time and cycle time for resetting	
_ECATMaster.bStartMaster	BOOL	Master station startup (When the system variable is set to ON, the EtherCAT master station is restarted and then the value automatically turns to OFF)	R/W
_ECATMaster.bStopMaster	BOOL	Master station stop (When the system variable is set to ON, the EtherCAT master station is stopped and then the value automatically turns to OFF)	R/W
_ECATMaster.bClearFrameCounter	BOOL	Frame sending and receiving counter upon resetting	R/W
_ECATMaster.bDisableMaster	BOOL	Disable a master station	R/W
_ECATMaster.iSlavesState	INT	Online status of all slave stations (1: All slave stations are online; 0: Some slave stations are offline.)	R
_ECATMaster.iFirstErrorSlave	INT	The first faulty slave station	R
_ECATMaster.dLibVersion	DINT	EtherCAT library version	R
_ECATMaster.dMstVersion	DINT	EtherCAT master station version	R
_ECATMaster.dDriveVersion	DINT	Version of the EtherCAT network adapter driver	R
_ECATMaster.dtx_error_cnt	DINT	Number of EtherCAT sending errors	R
_ECATMaster.drx_timeout_cnt	DINT	Number of frame receiving timeout events by EtherCAT	R
_ECATMaster.drx_corrupt_cnt	DINT	Number of invalid frame receiving events by EtherCAT	R
_ECATMaster.drx_unmach_cnt	DINT	Number of mismatched frame receiving events by EtherCAT	R
_ECATMaster.dRxPDOLength	DINT	Total number of PDOs received by EtherCAT	R

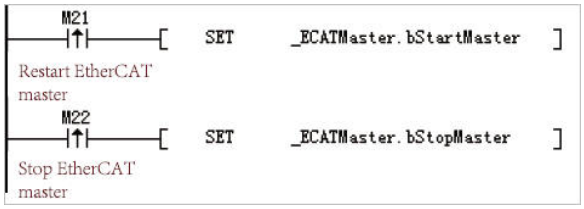
Name	Data type	Description	R/W
_ECATMaster.dTxPDOLength	DINT	Total number of PDOs sent by EtherCAT	R
_ECATMaster.dConfigureState	DINT	EtherCAT configuration status	R
_ECATMaster.dDelay	DINT	EtherCAT synchronization regulator	R
_ECATMaster.iSlaveLinkState	INT	Connection status of all slave stations 0: A slave station non-operational (non-OP) or faulty 1: All slave stations operational (OP) and normal	R
_ECATMaster.iUnµsed1	INT	Reserved 1	R
_ECATMaster.iUnµsed2	INT	Reserved 2	R
_ECATMaster.iDisableState	INT	Master station disabled	R

The preceding table lists information about the EtherCAT master station, such as the master station operation status and maximum cycle time.

Program example

1. The following figure shows the monitoring of EtherCAT master station operation status.

2. The following figure shows the restart of EtherCAT master stations.



> **NOTE**
>
> The instruction enable is edge-triggered.

5.7.6 _SYS_ECAT_Slave Operation Status

The following table lists parameters of GL20-RTU-ECAT.

Name	Data Type	Description	R/W
_RTU_ECT_INFO	_sRTU_ECT_INFO[72]	GL20ECAT coupler parameter information	-
_RTU_ECT_INFO[0]	_sRTU_ECT_INFO	-	-
_RTU_ECT_INFO[0].Runstate	BYTE	Operation status 0: Communication port module not active 1: Communication port module in the INIT state 2: Communication port module in the PREOP state 4: Communication port module in the SFAEOP state 8: Communication port module in the OP state	R
_RTU_ECT_INFO[0].LinkState	BYTE	Master connection state 0: Module abnormal or communication port module in a non-OP state 1: Communication port module in the OP state and all modules normal	R
_RTU_ECT_INFO[0].FaultID	INT	SDO 0XF100 FaultID	R
_RTU_ECT_INFO[0].SoftwareVersion	DINT	SDO 0X100A software version	R
_RTU_ECT_INFO[0].ErrorSlot1	DINT	SDO 0XF110 0x01 ErrorSlot1, slot0 to slot31	R
_RTU_ECT_INFO[0].ErrorSlot2	DINT	SDO 0XF110 0x02 ErrorSlot2, slot32 to slot63	R

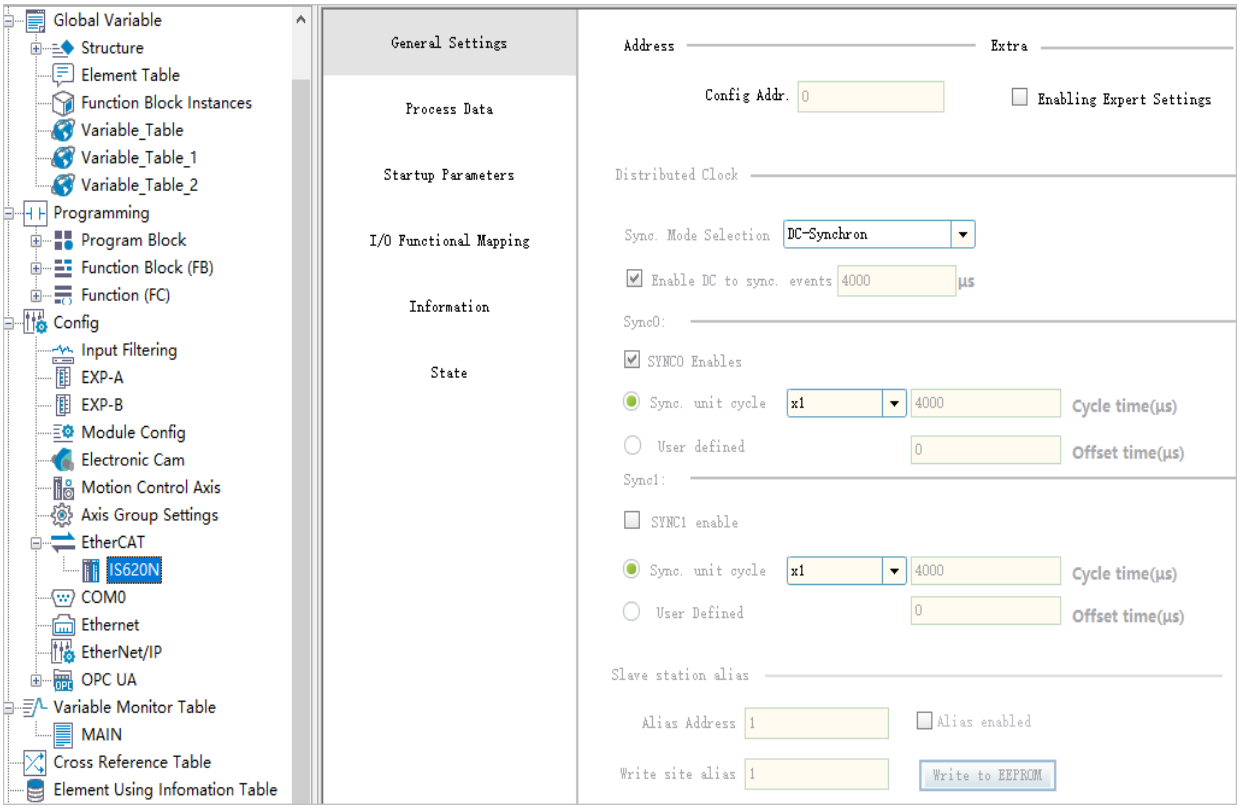
The following table lists the EtherCAT slave running status information.

Name	Data Type	Description	R/W
_ECATSlave	_sECTSLV_INFO[125]	Operation status of the EtherCAT slave	-
_ECATSlave[0]	_sECTSLV_INFO	-	-
_ECATSlave[0].bSlaveRunState	BOOL	Slave running status (ON: The slave is in the OP mode; OFF: The slave is not in the OP mode)	R
	BOOL	Status of slave alias writing (ON: busy)	R

Name	Data Type	Description	R/W
_ECATSlave[0].bSetAliasState			
_ECATSlave[0].bSetAliasError	BOOL	Failure of slave alias writing	R
_ECATSlave[0].bMatchState	BOOL	Slave type mismatch	R
_ECATSlave[0].bConfigError	BOOL	Slave configuration error	R
_ECATSlave[0].bSetAlias	BOOL	Setting of the slave alias, valid on the rising edge (used for commissioning on the AutoShop configuration page)	R/W
_ECATSlave[0].bDisableEnable	BOOL	Enabling of slave disabling OFF: Slave not disabled ON: Slave disabled	R/W
_ECATSlave[0].wALState	INT	Status of the EtherCAT state machine (1/2/4/8)	R
_ECATSlave[0].wAlCode	INT	Fault Codes	R
_ECATSlave[0].wActAlias	INT	Actual station alias	R
_ECATSlave[0].wTarAlias	INT	Station alias to be written (used for commissioning on the AutoShop configuration page)	R/W
_ECATSlave[0].wStationAddress	INT	Actual station name	R
_ECATSlave[0].wSlaveRingPos	INT	Configuration address	R
_ECATSlave[0].wSDOErrCnt	INT	Number of startup parameter configuration errors	R
_ECATSlave[0].dCfgErrorCode	DINT	Configuration error code	R
_ECATSlave[0].wDisableState	INT	Configuration status	R

NOTE

_ECATSlave is an array. For example, [0] represents configuration address 0, as shown in the following figure.



Program example

The following figure shows monitoring on the EtherCAT slave running status.

5.7.7 _SYS_EncAxis Encoder Axis Information

For EncAxis encoder axis information, see the following table.

Name	Data type	Description	R/W
_EncAxis[0]	_sENC_AXIS	-	R
_EncAxis[0].Axis	INT	Axis number	R
_EncAxis[0].Reserced0	INT	Reserved	R
_EncAxis[0].Unit	REAL	Unit	R
_EncAxis[0].UpperPosition	REAL	Upper limit position	R
_EncAxis[0].LowerPosition	REAL	Lower limit position	R
_EncAxis[0].Position	REAL	Position	R
_EncAxis[0].Velocity	REAL	Speed	R
_EncAxis[0].Reserced1	DINT[2]	Reserved	R

Name	Data type	Description	R/W
_EncAxis[0].Reserced1[0]	DINT	-	R
_EncAxis[0].Reserced1[1]	DINT	-	R

For EncAxis[0].Counter counter information, see the following table.

Name	Data type	Description	R/W
_EncAxis[0].Counter	_sENC_CNT	Counter	R
_EncAxis[0].Counter.ID	INT	Counter ID	R
_EncAxis[0].Counter.DecodeMode	INT	Decoding mode 1: Phase A/B single frequency 2: Phase A/B double frequency 3: Phase A/B quadruple frequency 4: CW/CCW 5: Pulse+Direction 6: Single-phase counter	R
_EncAxis[0].Counter.Source	INT	Signal source 1: X0X1 2: X2X3 3: X0 4: X1 5: X2 6: X3 7: Internal 1 ms 8: Internal 1 us 9: X4X5 10: X6X7 11: X4 12: X5 13: X6 14: X7	R
_EncAxis[0].Counter.CountMode	INT	Count mode 0: Linear mode 1: Rotary mode	R
	INT	Control word	R

Name	Data type	Description	R/W
_EncAxis[0].Counter.ControlWord			
_EncAxis[0].Counter.StatusWord	INT	Status word	R
_EncAxis[0].Counter.CountValue	DINT	Counting value	R
_EncAxis[0].Counter.Frequency	REAL	Frequency	R

For EncAxis[0].Reset reset information, see the following table.

Name	Data type	Description	R/W
_EncAxis[0].Reset	_sENC_RST	Reset	R
_EncAxis[0].Reset.Pin	INT	Reset pin 0: No assignment 1: X0 2: X1 3: X2 4: X3 5: X4 6: X5 7: X6 8: X7	R
_EncAxis[0].Reset.Edge	INT	Reset edge 0: Both rising edge and falling edge invalid 1: Valid at the rising edge 2: Valid at the falling edge 3: Valid at both rising edge and falling edge	R

For EncAxis[0].Preset preset information, see the following table.

Name	Data type	Description	R/W
_EncAxis[0].Preset	_sENC_PRESET	Preset pin 0: No assignment 1: X0	R

Name	Data type	Description	R/W
		2: X1 3: X2 4: X3 5: X4 6: X5 7: X6 8: X7	
_EncAxis[0].Preset.Pin	INT	Preset pin	R
_EncAxis[0].Preset.Edge	INT	Preset edge (Pulse unit position)	R
_EncAxis[0].Preset.Value	DINT	Preset value 0: Both rising edge and falling edge invalid 1: Valid at the rising edge 2: Valid at the falling edge 3: Valid at both rising edge and falling edge	R
_EncAxis[0].Preset.Position	REAL	Preset position	R

For EncAxis[0].Probe probe information, see the following table.

Name	Data type	Description	R/W
_EncAxis[0].Probe	_sENC_PROBE[2]	Probe	R
_EncAxis[0].Probe[0]	_sENC_PROBE		R
_EncAxis[0].Probe[0].Pin	INT	Pin	R
_EncAxis[0].Probe[0].Edge	INT	Probe edge	R
_EncAxis[0].Probe[0].PositiveValue	DINT	Probe rising edge	R
_EncAxis[0].Probe[0].NegativeValue	DINT	Probe falling edge	R
_EncAxis[0].Probe[0].PositivePosition	REAL	Probe rising edge position	R
_EncAxis[0].Probe[0].NegativePosition	REAL	Probe falling edge position	R

Name	Data type	Description	R/W
_EncAxis[0].Probe[1]	_sENC_PROBE		R
_EncAxis[0].Probe[1].Pin	INT	Pin	R
_EncAxis[0].Probe[1].Edge	INT	Probe edge	R
_EncAxis[0].Probe[1].PositiveValue	DINT	Probe rising edge	R
_EncAxis[0].Probe[1].NegativeValue	DINT	Probe falling edge	R
_EncAxis[0].Probe[1].PositivePosition	REAL	Probe rising edge position	R
_EncAxis[0].Probe[1].NegativePosition	REAL	Probe falling edge position	R

For EncAxis[0].Match comparison interruption, see the following table.

Name	Data type	Description	R/W
_EncAxis[0].Match	_sENC_MATCH[2]	Compare	R
_EncAxis[0].Match[0]	_sENC_MATCH		R
_EncAxis[0].Match[0].Value	DINT	Numerical value	R
_EncAxis[0].Match[0].Position	REAL	Position	R
_EncAxis[0].Match[0].Enable	BOOL	Enable	R
_EncAxis[0].Match[0].InterruptEnable	BOOL	Interruption enable	R
_EncAxis[0].Match[0].OutputEnable	BOOL	Output enable	R
_EncAxis[0].Match[0].InterruptMap	INT	Interruption association	R
_EncAxis[0].Match[0].OutputPin	INT	Output pin	R
_EncAxis[0].Match[0].OutputMode	INT	Output mode	R
_EncAxis[0].Match[0].OutputWidth	REAL	Output width	R
_EncAxis[0].Match[1]	_sENC_MATCH		R

Name	Data type	Description	R/W
_EncAxis[0].Match[1].Value	DINT	Numerical value	R
_EncAxis[0].Match[1].Position	REAL	Position	R
_EncAxis[0].Match[1].Enable	BOOL	Enable	R
_EncAxis[0].Match[1].InterruptEnable	BOOL	Interruption enable	R
_EncAxis[0].Match[1].OutputEnable	BOOL	Output enable	R
_EncAxis[0].Match[1].InterruptMap	INT	Interruption association	R
_EncAxis[0].Match[1].OutputPin	INT	Output pin	R
_EncAxis[0].Match[1].OutputMode	INT	Output mode	R
_EncAxis[0].Match[1].OutputWidth	REAL	Output width	R

5.7.8 _SYS_Ethernet Network Port Information

Ethernet network port information

The preceding variable table lists local information such as the IP address and MAC addresses.

Name	Data Type	Description	R/W
_Ethernet.MACAddress	INT[3]	Physical address	R
_Ethernet.MACAddress[0]	INT	-	R
_Ethernet.MACAddress[1]	INT	-	R
_Ethernet.MACAddress[2]	INT	-	R
_Ethernet.IPAddress	DINT	Local IP address	R/W
_Ethernet.Mask	DINT	Subnet mask	R/W
_Ethernet.Gateway	DINT	Gateway	R/W
IPCommand	INT	IP command	R/W

Modbus-TCP master protocol

Parameter	Data Type	Description	R/W
_MbTcpMst[0]	_sMB_TCP_MST	-	R
_MbTcpMst[0].IPAddress	DINT	IP address of a slave station	R
_MbTcpMst[0].Port	INT	Port number of a slave station	R
_MbTcpMst[0].Timeout	INT	Timeout interval (ms)	R
_MbTcpMst[0].Number	INT	Configuration number	R
_MbTcpMst[0].Enable	BOOL	Enabled	R
_MbTcpMst[0].Connected	BOOL	Connection state	R
_MbTcpMst[0].Busy	BOOL	Busy	R
_MbTcpMst[0].Error	BOOL	Error state	R
_MbTcpMst[0].ResponseTime	INT	Response time (ms)	R
_MbTcpMst[0].ErrorSlvAddr	INT	ID of the TCP slave with communication error	R
_MbTcpMst[…]	_sMB_TCP_MST	-	R

Modbus-TCP slave

Parameter	Data Type	Description	R/W
_MbTcpSlv.Port	INT	Port number	R
_MbTcpSlv.SlaveAddress	INT	Slave address	R
_MbTcpSlv.Connected	BOOL	Connection state	R
_MbTcpSlv.Connections	INT	Number of connections	R
_MbTcpSlv.IPAddress	DINT[32]	List of client IP addresses	R
_MbTcpSlv.IPAddress[0]	DINT	-	R
_MbTcpSlv.IPAddress[1]	DINT	-	R
_MbTcpSlv.IPAddress[2]	DINT	-	R
_MbTcpSlv.IPAddress[…]	DINT	-	R
_MbTcpSlv.IPAddress[30]	DINT	-	R
_MbTcpSlv.IPAddress[31]	DINT	-	R

EthIP scanner information

Parameter	Data Type	Description	R/W
_EthIPScanner[0].Instance	DINT	Label instance ID	R
_EthIPScanner[0].Connected	DINT	Connecting status 0: Initializing 1. Connected 2: Invalid network path 3: No response 4: Response error 5: Timeout 6: Connection closed 100: Request received 101: Request failed. Invalid network path 102: Request timed out 103: Client object request received 104: New Class 3 response	R
_EthIPScanner[0].GeneralStatus	INT	General error state 0: Request successful Others: Request failed	R
_EthIPScanner[0].ExtendedStatus	INT	Extended error state	R
_EthIPScanner [0].Reserved	DINT[6]	Reserved field R	R
...	...	...	...
_EthIPScanner[255].Instance	DINT	Label instance ID	R
_EthIPScanner[255].Connected	DINT	Connecting status 0: Initializing 1. Connected 2: Invalid network path 3: No response 4: Response error 5: Timeout 6: Connection closed	R

Parameter	Data Type	Description	R/W
		100: Request received 101: Request failed. Invalid network path 102: Request timed out 103: Client object request received 104: New Class 3 response	
_EthIPScanner[255].GeneralStatus	INT	General error state	R
_EthIPScanner[255].ExtendedStatus	INT	Extended error state	R
_EthIPScanner [255].Reserved	DINT[6]	Reserved field R	R

NOTE

- _EthIPScanner[n-1] represents the tag data with instance value n, and _EthIPScanner[n-1].Instance has a value of n. The array subscript corresponds one-to-one with Instance, with a difference of 1.
- To determine that "the tag with instance value n has normal communication", the system variable _EthIPScanner[n-1] must meet the following conditions:
  - CLASS1 type tag: Connected==1 && GeneralStatus==0 && ExtendedStatus==0
  - CLASS3 type tag: Connected==104 && GeneralStatus==0 && ExtendedStatus==0
  - UCMM type tag: Connected==100 && GeneralStatus==0 && ExtendedStatus==0

EthIP task statistics

Parameter	Data Type	Description	R/W
_EthIPTask.RunningState	DINT	Operation status 0: Stop 1: Running	R
_EthIPTask.MaxCycleTime	DINT	Maximum cycle time (μs)	R
_EthIPTask.MinCycleTime	DINT	Minimum cycle time (μs)	R
_EthIPTask.CycleTime	DINT	Cycle time (μs)	R
_EthIPTask.MaxExeTime	DINT	Maximum execution time (μs)	R
_EthIPTask.MinExeTime	DINT	Minimum execution time (μs)	R
_EthIPTask.ExeTime	DINT	Execution time (μs)	R
_EthIPTask.Reserved	DINT[6]	Reserved	R

FinsSlv TCP/UDP slave

Parameter	Data Type	Description	R/W
_FinsSlv.Port	INT	Port number	R
_FinsSlv.TcpErrorCode	INT	FINS TCP error code	R
_FinsSlv.UdpErrorCode	INT	FINS UDP error code	R
_FinsSlv.LogEnable	BOOL	FINS log information enable	w

Ethernet diagnosis

Name	Data Type	Description	R/W
_EthernetDiag[0]	_SETHERNET_DIAG	-	R
_EthernetDiag[0].interfaceState	INT	Network port state	R
_EthernetDiag[0].linkStatus	INT	Link connection state	R
_EthernetDiag[0].duplex	INT	Duplex mode	R
_EthernetDiag[0].autonegotiateStatus	INT	Auto negotiation mode	R
_EthernetDiag[0].speed	DINT	Speed, in Mbps	R
_EthernetDiag[0].mediaFcsErrors	DINT	Cyclic redundancy check	R
_EthernetDiag[0].mediaFrameTooLong	DINT	Data frame too long	R
_EthernetDiag[0].interfaceRcvMbps	REAL	Receiving flow, in Mbps	R
_EthernetDiag[0].interfaceSendMbps	REAL	Sending flow, in Mbps	R
_EthernetDiag[0].interfaceDiscards	DINT	Number of discarded received packets	R
_EthernetDiag[0].interfaceErrors	DINT	Number of received error packets	R
_EthernetDiag[0].lLinkDownCounter	DINT	Network disconnection count	R
_EthernetDiag[0].Reserved	DINT[20]	Reserved field	R
_EthernetDiag[…]	_SETHERNET_DIAG	-	R

EthIP scanner diagnosis

Name	Data Type	Description	R/W
_EthIPSysDiag	_sETHIP_SYS_DIAG	-	R
_EthIPSysDiag.iNumFailedSentIOMessages	INT	Number of I/O messages failed to send	R
_EthIPSysDiag.iNumFailedRecvIOMessages	INT	Number of I/O messages failed to receive	R
_EthIPSysDiag.iNumCurrentIOConnections	DINT	Number of Class 1 connections	R
_EthIPSysDiag.iNumCurrentExplicitConnections	DINT	Number of Class 3 connections	R
_EthIPSysDiag.iAllLostPacket	DINT	Number of all lost I/O packets	R
_EthIPSysDiag.lRcvdPPS	DINT	Number of received PPS (packet per second)	R
_EthIPSysDiag.lSendPPS	DINT	Number of sent PPS (packet per second)	R
_EthIPSysDiag.fRecvSpeed	REAL	Receiving speed, in Mbps	R
_EthIPSysDiag.fSendSpeed	REAL	Sending speed, in Mbps	R

EthIP connection diagnosis

Name	Data Type	Description	R/W
_EthIPConnDiag[0]	_sETHIP_CONN_DIAG	-	-
_EthIPConnDiag[0].nConnInstance	DINT	Connection instance	-
_EthIPConnDiag[0].lMaxSendDelay	DINT	Maximum delay of any message sent continuously, in ms	-
_EthIPConnDiag[0].lMaxRcvdDelay	DINT	Maximum delay of any message received continuously, in ms	-
_EthIPConnDiag[0].lLostPackets	DINT	Number of lost packets	-
_EthIPConnDiag[0].lAverSendinterval	DINT	Average frame sending interval, in ms	-
_EthIPConnDiag[0].lAverRcvdinterval	DINT	Average frame receiving interval, in ms	-
_EthIPConnDiag[0].Reserved	DINT[6]	Reserved field	-
_EthIPConnDiag[…]	_sETHIP_CONN_DIAG	-	R

SFTP server diagnosis

Name	Data Type	Description	R/W
_SftpSeverDiag	_sSFTP_SERVER_DIAG	SFTP server diagnosis information	R
_SftpSeverDiag.iState	INT	Server status	R
_SftpSeverDiag.iErrorCode	INT	Error code	R
_SftpSeverDiag.diSDCard TotalSize	DINT	Total storage space of SD card	R
_SftpSeverDiag.diSDCardFreeSize	DINT	Available storage space of SD card	R

Firmware of V6.5.0.0 or later and software of AutoShop V4.11.0.0 or later support system variables corresponding to SFTP server diagnosis.

5.7.9 _SYS_INFO PLC Operation Information

_DevInfo: Device information, as shown in the following table

Name	Data type	Description	R/W	Comparison with H3U
_DevInfo.Device	INT	Device model ID	R	-
_DevInfo.Vender	INT	Manufacturer ID	R	-
_DevInfo.HWVersion	DINT	Hardware version	R	-
_DevInfo.SWVersion	DINT	Software version	R	D8100 and D8101
_DevInfo.FPGAVersion	DINT	FPGA version	R	D8104, D8105
_DevInfo.NSTDVersion	DINT	Customized version	R	-

The preceding table lists the PLC device information.

_OSM: System monitor, as shown in the following table

Name	Data type	Description	R/W
_OSM.CPU	INT	CPU usage	R
_OSM.Memory	INT	Memory usage	R

The preceding table lists the CPU and memory usage for CPU performance diagnosis.

_Program: User program information, as shown in the following table

Name	Data type	Description	R/W	Comparison with H3U
_Program.TotalSize	DINT	Total program capacity	R	-
_Program.μsedSize	DINT	Program capacity used	R	-
_Program.Interval	DINT	Program task cycle (μs)	R	-
_Program.CurPeriod	DINT	Current program task cycle (μs)	R	D8010
_Program.MinPeriod	DINT	Minimum program task cycle (μs)	R	D8011
_Program.MaxPeriod	DINT	Maximum program task cycle (μs)	R	D8012
_Program.CurRunTime	DINT	Current operating time (μs)	R	-
_Program.MinRunTime	DINT	Minimum program operating time (μs)	R	-
_Program.MaxRunTime	DINT	Maximum program operating time (μs)	R	-
_Program.AveRunTime	DINT	Average program operating time (μs)	R	-
_Program.Reset	BOOL	Reset cycle time	R/W	-

The preceding table lists the execution cycle of the program and task, which can be used to judge the program execution logic complexity.

_CurErrLst: List of current errors, as shown in the following table

Name	Data type	Description	R/W
_CurErrLst.Quantity	INT	Number of current errors	R
_CurErrLst.ErrorInfo	_sERR_INFO[42]	List of current errors	R
_CurErrLst.ErrorInfo[0]	_sERR_INFO	-	R
_CurErrLst.ErrorInfo[0].ErrorCode	INT	Error code	R
_CurErrLst.ErrorInfo[0].ComponentID	INT	Component ID	R
_CurErrLst.ErrorInfo[0].Location	DINT	Error position	R

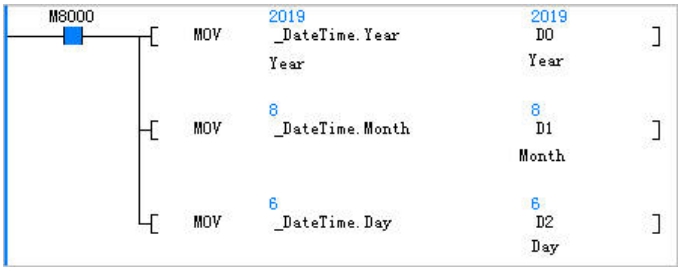
Name	Data type	Description	R/W
_CurErrLst.ErrorInfo[0].Timestamp	DINT	Timestamp, indicating the number of seconds from 00:00:00 on January 1, 1970 to the error generation time	R
_CurErrLst.ErrorInfo[1]	_sERR_INFO	-	R
_CurErrLst.ErrorInfo[1].ErrorCode	INT	Error code	R
_CurErrLst.ErrorInfo[1].ComponentID	INT	Component ID	R
_CurErrLst.ErrorInfo[1].Location	DINT	Error position	R
_CurErrLst.ErrorInfo[1].Timestamp	DINT	Timestamp, indicating the number of seconds from 00:00:00 on January 1, 1970 to the error generation time	R
CurErrLst.ErrorInfo[…]			
_CurErrLst.ErrorInfo[30]	_sERR_INFO	-	R
_CurErrLst.ErrorInfo[30].ErrorCode	INT	Error code	R
_CurErrLst.ErrorInfo[30].ComponentID	INT	Component ID	R
_CurErrLst.ErrorInfo[30].Location	DINT	Error position	R
_CurErrLst.ErrorInfo[30].Timestamp	DINT	Timestamp, indicating the number of seconds from 00:00:00 on January 1, 1970 to the error generation time	R
_CurErrLst.ErrorInfo[31]	_sERR_INFO	-	R
_CurErrLst.ErrorInfo[31].ErrorCode	INT	Error code	R
_CurErrLst.ErrorInfo[31].ComponentID	INT	Component ID	R
_CurErrLst.ErrorInfo[31].Location	DINT	Error position	R
_CurErrLst.ErrorInfo[31].Timestamp	DINT	Timestamp, indicating the number of seconds from 00:00:00 on January 1, 1970 to the error generation time	R

The preceding table lists PLC error logs, up to 32 records. You can check details in the AutoShop fault record. For details about error codes, see Error Code.

_DateTime: RTC clock, as shown in the following table

Name	Data type	Description	R/W	Comparison with H3U
_DateTime.Second	INT	Second, ranging from 0 to 60, in which 60 is the leap second	R	D8013
_DateTime.Minute	INT	Minute, ranging from 0 to 59	R	D8014
_DateTime.Hour	INT	Hour, ranging from 0 to 23	R	D8015
_DateTime.Day	INT	Day in a month, ranging from 1 to 31	R	D8016
_DateTime.Month	INT	Month, ranging from 1 to 12	R	D8017
_DateTime.Year	INT	Year	R	D8018
_DateTime.WeekDay	INT	Day of a week, ranging from 0 to 6, in which 0 indicates Sunday and 1 indicates Monday	R	D8019
_DateTime.YearDay	INT	Number of the day counted from January 1 of each year, ranging from 0 to 365, in which 0 indicates January 1	R	-
_DateTime.Timestamp	DINT	Total number of seconds from 00:00:00 on January 1, 1970 to the current time	R	-

The following figure shows the program.



The preceding table lists RTC information.

The following table describes M8000/D8000 elements.

Name	Data type	Description	R/W
M8000	BOOL	ON during operation of the user program	R

Name	Data type	Description	R/W
M8001	BOOL	Inverted M8000 state	R
M8002	BOOL	ON in the first operation cycle of the user program	R
M8003	BOOL	Inverted M8002 state	R
M8011	BOOL	Free-run clock with a cycle of 10 ms	R
M8012	BOOL	Free-run clock with a cycle of 100 ms	R
M8013	BOOL	Free-run clock with a cycle of 1s	R
M8014	BOOL	Free-run clock with a cycle of 1 min	R
M8020	BOOL	Zero flag	R
M8021	BOOL	Borrow flag	R
M8022	BOOL	Carry flag	R
M8029	BOOL	Multi-cycle instruction execution completion flag, applicable to the RAMP, SORT, and SORT2 instructions	R
M8040	BOOL	SFC STL status transition disable	R
M8161	BOOL	OFF-16-bit mode, ON-8-bit mode Bit processing mode for ASCII/HEX/CCD/ LRC/CRC/RS	R
M8163	BOOL	BINDA instruction output switchover flag (retained or switched to 0000h)	R
M8165	BOOL	Flag of descending sort enabling for the SORT2 instruction	R
M8168	BOOL	SMOV instruction data format settings: OFF-BCD mode or ON-HEX mode	R
M8333	BOOL	Flag of all BKCMP instruction matrix comparison results equal to 1	R
D8066	INT	Critical errors in user programs and instructions (triggered, not reset)	R
D8067	INT	Minor errors in user programs and instructions (triggered, not reset)	R

The preceding table lists information about M8000/D8000 components. A small number of such components are reserved.

_LemStation: Local module expansion master station, as shown in the following table

Name	Data type	Description	R/W
_LemStation	_sLEM_STATION	Local module expansion master station	-
_LemStation.State	INT	Operating status; 0: Stop, 1: Operating	R
_LemStation.MC_State	INT	Module control status; 0x01: INIT; 0x02: PreOP; 0x08: OP; 0x80: ERROR	R
_LemStation.ErrorCode	DINT	Error code	R
_LemStation.ConfigNumber	INT	Number of configured modules	R
_LemStation.ModuleNumber	INT	Number of actually mounted modules	R
_LemStation.SlotMonitor	INT	Stops if module slot changes are detected during operation. Monitoring cycle, unit: s; 0: Disable	R/W
_LemStation.MsgMonitor	INT	Module controller log monitoring; 1: Enable the log; 2: Disable the log	R/W

_ExtSlt: GL10 local expansion module information, as shown in the following table

Name	Data type	Description	R/W
_ExtSlt[0]	_sEXT_SLT	-	R
_ExtSlt[0].ConfigModule	INT	Module type of the AutoShop configuration	R
_ExtSlt[0].MountedModule	INT	Type of the installed electrical module	R
_ExtSlt[0].LogicVersion	DINT	Version of the logic device (module version)	R
_ExtSlt[0].SWVersion	DINT	Software version (module version)	R
_ExtSlt[0].Error	BOOL	Error status (ON: failed; OFF: normal)	R
_ExtSlt[1]	_sEXT_SLT	-	R
_ExtSlt[1].ConfigModule	INT	Configuration module type	R
_ExtSlt[1].MountedModule	INT	Installed module type	R
_ExtSlt[1].LogicVersion	DINT	Version of the logic device	R
_ExtSlt[1].SWVersion	DINT	Software version	R
_ExtSlt[1].Error	BOOL	Error state	R
_ExtSlt[...]	_sEXT_SLT	-	-

Name	Data type	Description	R/W
_ExtSlt[15]	_sEXT_SLT	-	R
_ExtSlt[15].ConfigModule	INT	Configuration module type	R
_ExtSlt[15].MountedModule	INT	Installed module type	R
_ExtSlt[15].LogicVersion	DINT	Version of the logic device	R
_ExtSlt[15].SWVersion	DINT	Software version	R
_ExtSlt[15].Error	BOOL	Error state	R

_GL20_ExtSlt: GL20 local expansion module information, as shown in the following table

Name	Data type	Description	R/W
_GL20_ExtSlt[0].ConfigModule	DINT	Type of the configured module	R
_GL20_ExtSlt[0].MountedModule	DINT	Type of the installed module	R
_GL20_ExtSlt[0].LogicVersion	DINT	Version of the logic device	R
_GL20_ExtSlt[0].SWVersion	DINT	Software version	R
_GL20_ExtSlt[0].Error	BOOL	Error state	R
_GL20_ExtSlt[0].bDisableSlot	BOOL	Module disabled	R/W

_ExtCard: GE20 local expansion module information, as shown in the following table

Name	Data type	Description	R/W
_ExtCard[0].ConfigModule	INT	Type of configured expansion card	R
_ExtCard[0].MountedModule	INT	Type of installed expansion card	R
_ExtCard[0].LogicVersion	INT	Version of the logic device	R
_ExtCard[0].SWVersion	INT	Software version	R
_ExtCard[0].Error	BOOL	Error state	R
_ExtCard[0].DI0	BOOL	4DI expansion card DI0	R/W
_ExtCard[0].DI1	BOOL	4DI expansion card DI1	R/W
_ExtCard[0].DI2	BOOL	4DI expansion card DI2	R/W
_ExtCard[0].DI3	BOOL	4DI expansion card DI3	R/W
_ExtCard[0].DO0	BOOL	4DO expansion card DO0	R/W

Name	Data type	Description	R/W
_ExtCard[0].DO1	BOOL	4DO expansion card DO1	R/W
_ExtCard[0].DO2	BOOL	4DO expansion card DO2	R/W
_ExtCard[0].DO3	BOOL	4DO expansion card DO3	R/W
_ExtCard[0].AD0	INT	AD0 channel of analog expansion card	R/W
_ExtCard[0].AD1	INT	AD1 channel of analog expansion card	R/W
_ExtCard[0].DA0	INT	DA0 channel of analog expansion card	R/W
_ExtCard[0].ConfigData	INT[16]	Expansion card configuration data	R

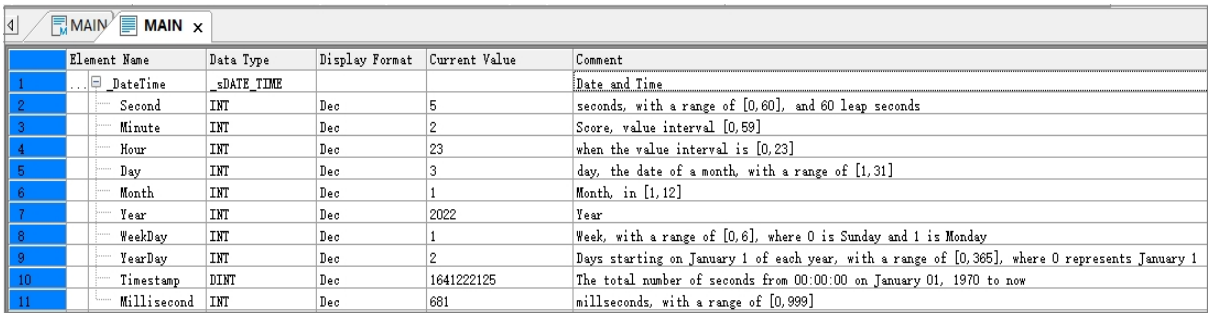
_LemStationLFR: Local module expansion master station, as shown in the following table

Name	Data type	Description	R/W
_LemStationLFR	_sLEMSTATION_LFR	Frame loss rate of local expansion modules	R
_LemStationLFR.ReqErrCnt	DINT	Request error counter	R
_LemStationLFR.ResErrCnt	DINT	ErrorFrameCount	R
_LemStationLFR.LossFramesRate	REAL	Frame loss rate	R
_LemStationLFR.Debug	_sLBμs_DEBUG	Commissioning terminal of local expansion module	R

- The configured expansion card type must be consistent with the actually installed expansion card type.
- 4DI is the default mapping address of the input expansion card. 4DO is the default mapping address of the output expansion card. AD and DA are default mapping addresses of analog expansion cards in the same program control way as modules.

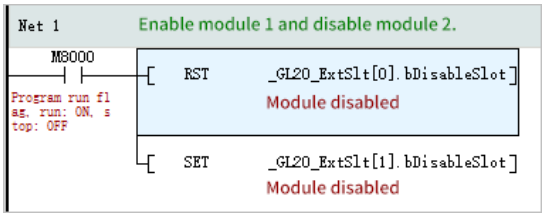
Program example

- RTC clock: Monitored in the monitoring table, as shown in the following figure

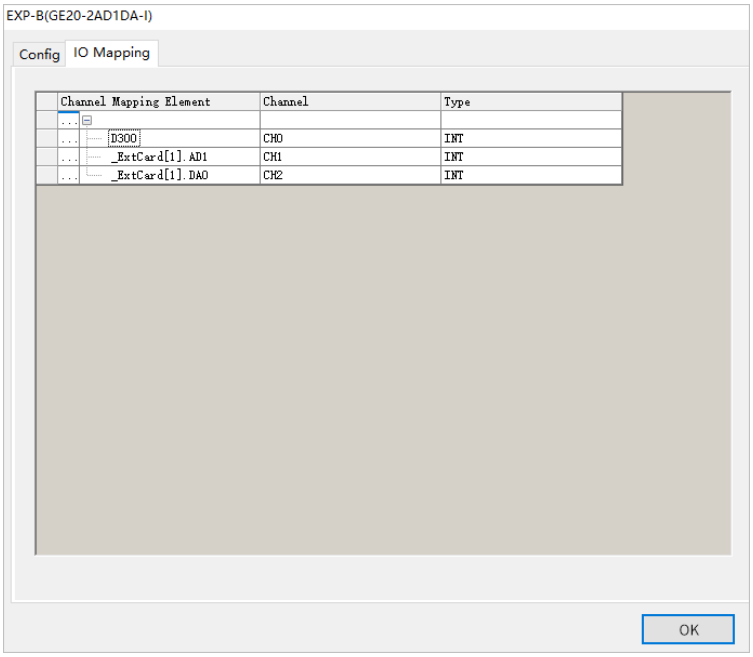


	Element Name	Data Type	Display Format	Current Value	Comment
1	_DateTime	_sDATE_TIME			Date and Time
2	Second	INT	Dec	5	seconds, with a range of [0,60], and 60 leap seconds
3	Minute	INT	Dec	2	Score, value interval [0,59]
4	Hour	INT	Dec	23	when the value interval is [0,23]
5	Day	INT	Dec	3	day, the date of a month, with a range of [1,31]
6	Month	INT	Dec	1	Month, in [1,12]
7	Year	INT	Dec	2022	Year
8	WeekDay	INT	Dec	1	Week, with a range of [0,6], where 0 is Sunday and 1 is Monday
9	YearDay	INT	Dec	2	Days starting on January 1 of each year, with a range of [0,365], where 0 represents January 1
10	Timestamp	DINT	Dec	1641222125	The total number of seconds from 00:00:00 on January 01, 1970 to now
11	Millisecond	INT	Dec	681	millseconds, with a range of [0,999]

- GL20 local modules: The program enable module 1 and disable module 2 to make program compilation and download take effect.

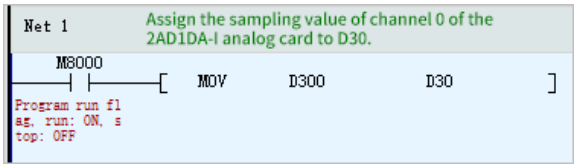


- GE20 local expansion modules: Configure GE20-2AD1DA-I I/O mapping, as shown in the following figure



> **NOTE**
>
> AutoShop V4.8.1.0 and above does not support the modification of GE20-2AD1DA-I/GE20-2AD1DA-V channel mapping elements. To modify the element, do as follows in AutoShop earlier than V4.8.1.0:

Use the LD programming language to program GE20-2AD1DA-I sampling, and assign the voltage sampling value of CH0 from D300 to D30, which can be compiled and downloaded for running, as shown in the following figure.



5.7.10 _SYS_MC_Axis Motion Control Axis Information

The following table describes the McAxis axis operation status.

Name	Data type	Description	R/W
_McAxis[0]	_sMCAXIS_INFO	-	-
_McAxis[0].bPowerState	BOOL	Axis enable state	R
_McAxis[0].bDebugState	BOOL	Axis commissioning state	R

Name	Data type	Description	R/W
_McAxis[0].fSetPosition	REAL	Position reference	R
_McAxis[0].fSetVelocity	REAL	Velocity reference	R
_McAxis[0].fSet_Acc_Dec	REAL	Acceleration/Deceleration rate reference	R
_McAxis[0].fSetTorque	REAL	Torque reference	R
_McAxis[0].fActPosition	REAL	Current position	R
_McAxis[0].fActVelocity	REAL	Current speed	R
_McAxis[0].fAct_Acc_Dec	REAL	Current acceleration/deceleration rate	R
_McAxis[0].fActTorque	REAL	Current torque	R
_McAxis[0].wPLCOpenState	INT	PLCOpen state machine 0: PowerOff 1: ErrorStop 2: Stopping 3: StandStill 4: Discrete motion 5: Continuous motion 7: Homing 8: Synchronized motion	R
_McAxis[0].wConfigState	INT	Configuration status 0: Init (axis in the initialization state) 1: Configure finish (reading of configuration data completed) 2: Sync finish (synchronized with EtherCAT tasks) 3: Wait communication (communication with the servo drive established) 4: Slave ready (initialization completed for the servo drive controlled by axes) 5: Axis ready (communication established)	R
_McAxis[0].wAxisError	INT	Axis fault[1]	R
_McAxis[0].wServoError	INT	Servo drive fault[1]	R
_McAxis[0].bEnterDebug	BOOL	Monitoring mode (online commissioning mode of AutoShop axes)	R/W

Name	Data type	Description	R/W
_McAxis[0].bPowerOn	BOOL	Enabled (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].bStop	BOOL	Stopped (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].bReset	BOOL	Reset (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].bJogP	BOOL	Jog+ (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].bJogN	BOOL	Jog- (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].bHome	BOOL	Homing (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].bSetPos	BOOL	Current position setting (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].bAbsPos	BOOL	Absolute positioning (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].bRevPos	BOOL	Reciprocating (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].bRelPos	BOOL	Relative positioning (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].bVelocity	BOOL	Continuous motion (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].bTorque	BOOL	Torque mode (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].wDebugMotionType	INT	Commissioning motion type (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].fJogVelocity	REAL	Jog speed (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].fPositionOffser	REAL	Homing offset (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].fPresetPosition	REAL	Reset position (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].fTarPosition1	REAL	Target position 1 (online commissioning mode of AutoShop axes)	R/W

Name	Data type	Description	R/W
_McAxis[0].fTarVelocity1	REAL	Target speed 1 (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].fTarAcceleration1	REAL	Target acceleration rate 1 (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].fTarDecelaration1	REAL	Target deceleration rate 1 (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].wCurveType1	INT	Curve type 1 (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].fTarPosition2	REAL	Target position 2 (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].fTarVelocity2	REAL	Target speed 2 (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].fTarAcceleration2	REAL	Target acceleration rate 2 (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].fTarDecelaration2	REAL	Target deceleration rate 2 (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].wCurveType2	INT	Curve type 2 (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].dUnused	DINT	Reserved	R
_McAxis[0].fTarTorque	REAL	Target torque (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].fTarTorqueSlop	REAL	Torque slope (online commissioning mode of AutoShop axes)	R/W
_McAxis[0].wControlWord	INT	Control word	R
_McAxis[0].wStatusword	INT	Status word	R
_McAxis[0].dSetPosition	DINT	Target position	R
_McAxis[0].dActPosition	DINT	Current position	R
_McAxis[0].dSetVelocity	DINT	Velocity reference	R
_McAxis[0].dActVelocity	DINT	Current speed	R
_McAxis[0].dSetTorque	INT	Torque reference	R
_McAxis[0].dActTorque	INT	Current torque	R

Name	Data type	Description	R/W
_McAxis[0].dDO	DINT	Digital output	R
_McAxis[0].dDI	DINT	Digital input	R
_McAxis[0].wModesOfOperation	INT	Control mode 6: Homing mode 8: Synchronous position mode 10: Synchronous torque mode	R
_McAxis[0].wModesOfOperationDisplay	INT	Current control mode 6: Homing mode 8: Synchronous position mode 10: Synchronous torque mode	R
_McAxis[0].wTouchFunction	INT	Touch probe function	R
_McAxis[0].wTouchStatus	INT	Probe status	R
_McAxis[0].dTouch1PPos	DINT	Touch probe 1 positive edge	R
_McAxis[0].dTouch2PPos	DINT	Touch probe 2 positive edge	R
_McAxis[0].dTouch1NPos	DINT	Probe 1 position on the falling edge	R
_McAxis[0].dTouch2NPos	DINT	Probe 2 position on the falling edge	R
_McAxis[0].wErrorCode	INT	Fault type[1]	R
_McAxis[0].wAxisRingPos	INT	Axis configuration position	R
_McAxis[0].wAxisID	INT	Axis ID	R
_McAxis[0].fUnits	REAL	Axis gear ratio	R
_McAxis[0].bMotionState	BOOL	Motion status, indicating whether an axis is in motion	R
_McAxis[0].bphlimit	BOOL	Positive limit input status of hardware	R
_McAxis[0].bnhlimit	BOOL	Negative limit input status of hardware	R
_McAxis[0].bhomestate	BOOL	Home switch input status of hardware	R
_McAxis[0].bpslimit	BOOL	Software positive limit reached or not	R
_McAxis[0].bnslimit	BOOL	Software negative limit reached or not	R
_McAxis[0].dLocialAxisSetPos	DINT	Local pulse axis position settings	R

Name	Data type	Description	R/W
_McAxis[…]	_sMCAXIS_INFO	-	R

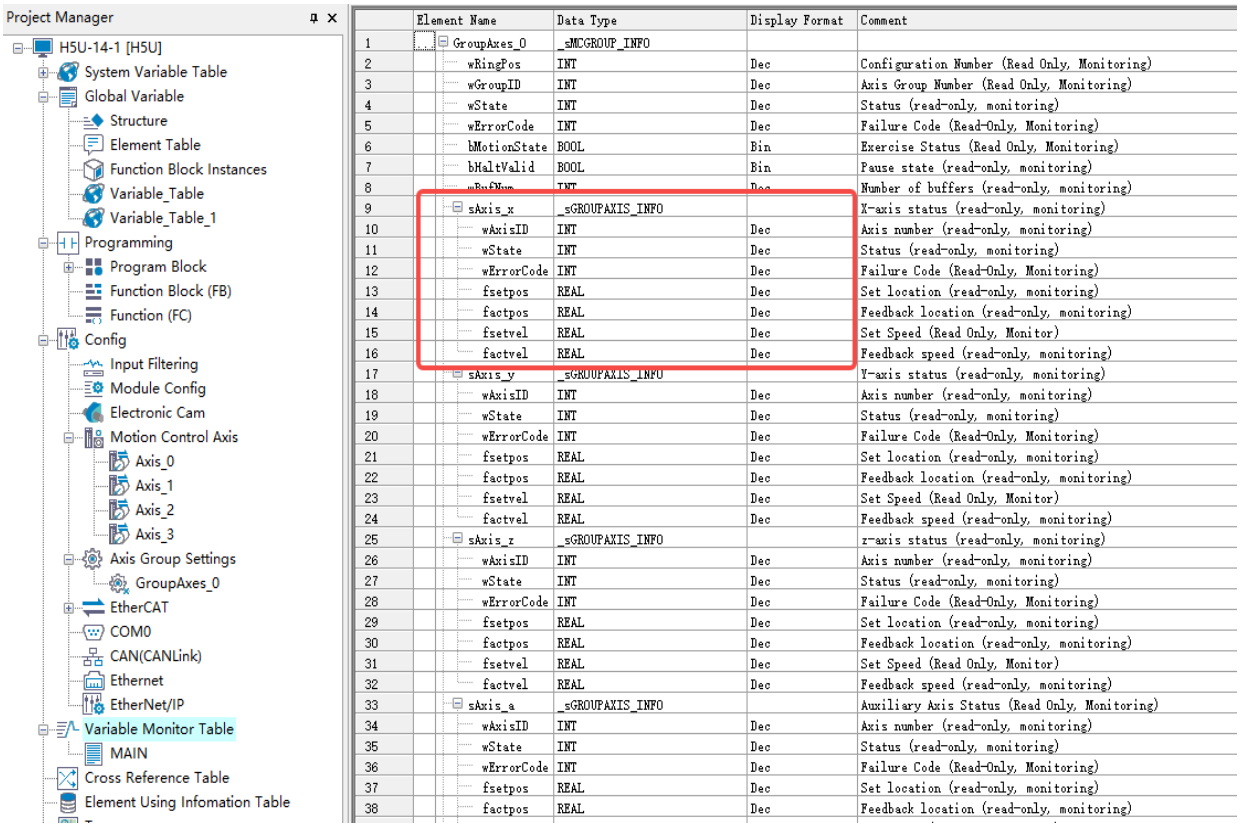
NOTE

[1]: For details about axis faults and drive faults, see Fault Category.

5.7.11 _sGROUPAXIS_INFO Axis Status

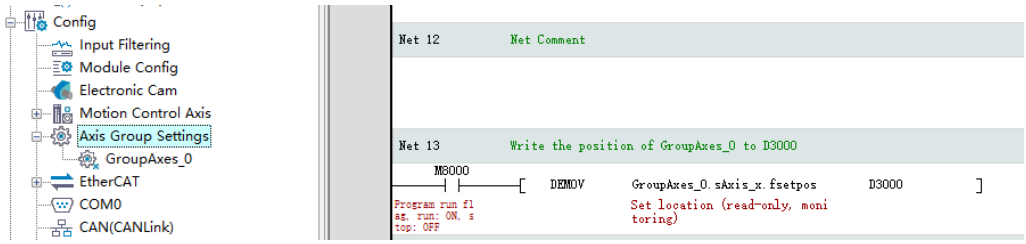
Name	Type	Description
wAxisID	INT16	Axis ID
wState	INT16	The status of an axis' PLCOpen state machine 0: PowerOff 1: ErrorStop 2: Stopping 3: StandStill 4: DiscreteMotion 5: ContinuousMotion 7: Homing 8: SynchronizedMotion
wErrorCode	INT16	The fault code of an axis
fsetpos	REAL	Position reference
factpos	REAL	Feedback position
fsetvel	REAL	Speed reference
factvel	REAL	Feedback speed

This system variable exists in the axis group _sMCGROUP_INFO and is used to represent the state of individual axes within the axis group.



	Element Name	Data Type	Display Format	Comment
1	GroupAxes_0	_sMCGROUP_INFO		
2	wRingPos	INT	Dec	Configuration Number (Read Only, Monitoring)
3	wGroupID	INT	Dec	Axis Group Number (Read Only, Monitoring)
4	wState	INT	Dec	Status (read-only, monitoring)
5	wErrorCode	INT	Dec	Failure Code (Read-Only, Monitoring)
6	bMotionState	BOOL	Bin	Exercise Status (Read Only, Monitoring)
7	bHaltValid	BOOL	Bin	Pause state (read-only, monitoring)
8	wBufNum	INT	Dec	Number of buffers (read-only, monitoring)
9	sAxis_x	_sGROUPAXIS_INFO		X-axis status (read-only, monitoring)
10	wAxisID	INT	Dec	Axis number (read-only, monitoring)
11	wState	INT	Dec	Status (read-only, monitoring)
12	wErrorCode	INT	Dec	Failure Code (Read-Only, Monitoring)
13	fsetpos	REAL	Dec	Set location (read-only, monitoring)
14	factpos	REAL	Dec	Feedback location (read-only, monitoring)
15	fsetvel	REAL	Dec	Set Speed (Read Only, Monitor)
16	factvel	REAL	Dec	Feedback speed (read-only, monitoring)
17	sAxis_y	_sGROUPAXIS_INFO		Y-axis status (read-only, monitoring)
18	wAxisID	INT	Dec	Axis number (read-only, monitoring)
19	wState	INT	Dec	Status (read-only, monitoring)
20	wErrorCode	INT	Dec	Failure Code (Read-Only, Monitoring)
21	fsetpos	REAL	Dec	Set location (read-only, monitoring)
22	factpos	REAL	Dec	Feedback location (read-only, monitoring)
23	fsetvel	REAL	Dec	Set Speed (Read Only, Monitor)
24	factvel	REAL	Dec	Feedback speed (read-only, monitoring)
25	sAxis_z	_sGROUPAXIS_INFO		z-axis status (read-only, monitoring)
26	wAxisID	INT	Dec	Axis number (read-only, monitoring)
27	wState	INT	Dec	Status (read-only, monitoring)
28	wErrorCode	INT	Dec	Failure Code (Read-Only, Monitoring)
29	fsetpos	REAL	Dec	Set location (read-only, monitoring)
30	factpos	REAL	Dec	Feedback location (read-only, monitoring)
31	fsetvel	REAL	Dec	Set Speed (Read Only, Monitor)
32	factvel	REAL	Dec	Feedback speed (read-only, monitoring)
33	sAxis_a	_sGROUPAXIS_INFO		Auxiliary Axis Status (Read Only, Monitoring)
34	wAxisID	INT	Dec	Axis number (read-only, monitoring)
35	wState	INT	Dec	Status (read-only, monitoring)
36	wErrorCode	INT	Dec	Failure Code (Read-Only, Monitoring)
37	fsetpos	REAL	Dec	Set location (read-only, monitoring)
38	factpos	REAL	Dec	Feedback location (read-only, monitoring)

For example, write the position reference of the X-axis into the D3000 in the PLC:



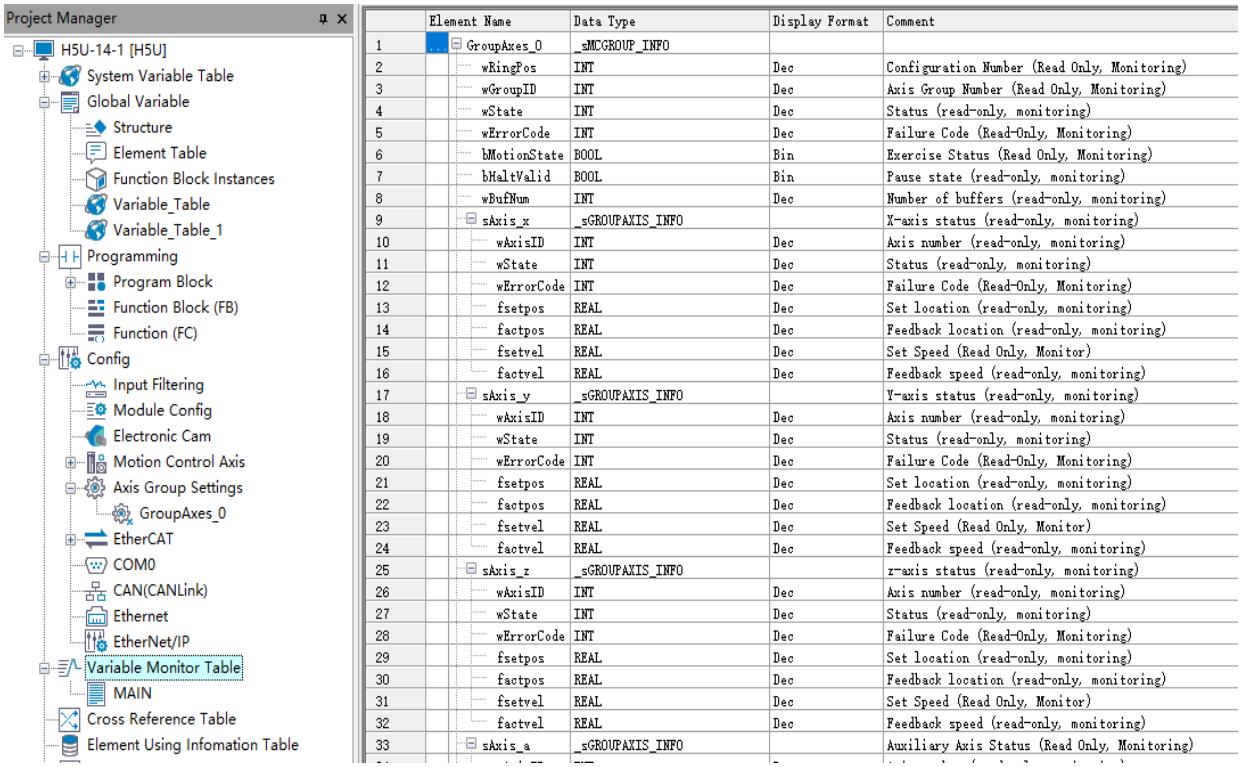
5.7.12 _sMCGROUP_INFO Axis Group Status

Name	Type	Description
wRingPos	INT16	Axis group number
wGroupID	INT16	Axis number
wState	INT16	Axis group status 0: Initialization The axis configuration in the axis group is not completed. 1: Disabled Not all axes in the axis group are enabled. 2: Single stop An axis in the axis group calls the instruction MC_Gtop. 3: Single homing An axis in the axis group calls the instruction MC_Home. 4: Single motion

Name	Type	Description
		An axis in the axis group calls single-axis motion instructions such as MC_MoveAbsolute. 5: Stop upon a fault An axis in the axis group is in a fault state. 6: StandStill All axes in the axis group are in the StandStill state. 7: Stopping The instruction MC_GroupStop is called. 8: Synchronous motion A linear interpolation or circular interpolation instruction is called.
wErrorCode	INT16	Fault code
bMotionState	BOOL	Motion status TRUE: In motion FALSE: Not in motion
bHaltValid	BOOL	Paused status TRUE: Pause valid FALSE: Pause invalid
wBufNum	INT16	Number of buffered curves
sAxis_x	_sGROUPAXIS_INFO	Status of the X-axis
sAxis_y	_sGROUPAXIS_INFO	Status of the Y-axis
sAxis_z	_sGROUPAXIS_INFO	Status of the Z-axis
sAxis_a	_sGROUPAXIS_INFO	Status of the auxiliary axis
fSetvel	REAL	Velocity reference In linear interpolation mode, it indicates the interpolation velocity of a space straight line. In circular interpolation mode, it indicates the linear velocity of a circular arc.
fSetacc_dec	REAL	Acceleration/deceleration reference Indicates the change rate of setvel.

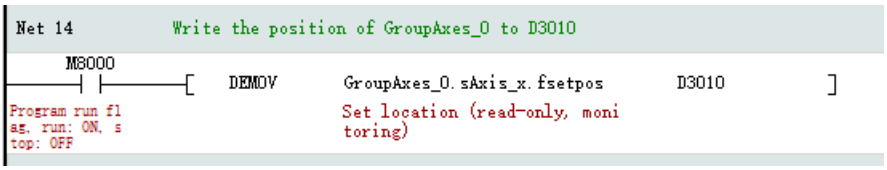
Name	Type	Description
fSetvel_buf	REAL	Velocity reference of a buffered curve In linear interpolation mode, it indicates the interpolation velocity of a space straight line. In circular interpolation mode, it indicates the linear velocity of a circular arc.
fSetacc_dec_buf	REAL	Acceleration/deceleration reference of a buffered curve Indicates the change rate of fSetvel_buf
fSetdis	REAL	Distance reference In linear interpolation mode, it indicates the distance at which a space straight line moves after the instruction is executed. In circular interpolation mode, it indicates the length of a circular arc in which the circular arc moves after the instruction is executed.
fLeftdis	REAL	Remaining distance In linear interpolation mode, it indicates the left distance for this section of a space straight line after the instruction is executed. In circular interpolation mode, it indicates the length of a space circular arc left after the instruction is executed.
fCenter_x	REAL	Coordinates of point X at the center of a circular arc during circular interpolation
fCenter_y	REAL	Coordinates of point Y at the center of a circular arc during circular interpolation
fCenter_z	REAL	Coordinates of point Z at the center of a circular arc during circular interpolation
fRadius	REAL	Radius of a circular arc during circular interpolation
fStartAng	REAL	Start angle during circular interpolation
fSetAng	REAL	Motion angle during circular interpolation

This system variable is used to indicate the status of the entire axis group:



	Element Name	Data Type	Display Format	Comment
1	GroupAxes_0	_sMCGROUP_INFO		
2	wRingPos	INT	Dec	Configuration Number (Read Only, Monitoring)
3	wGroupID	INT	Dec	Axis Group Number (Read Only, Monitoring)
4	wState	INT	Dec	Status (read-only, monitoring)
5	wErrorCode	INT	Dec	Failure Code (Read-Only, Monitoring)
6	bMotionState	BOOL	Bin	Exercise Status (Read Only, Monitoring)
7	bHaltValid	BOOL	Bin	Pause state (read-only, monitoring)
8	wBufNum	INT	Dec	Number of buffers (read-only, monitoring)
9	sAxis_x	_sGROUPAXIS_INFO		X-axis status (read-only, monitoring)
10	wAxisID	INT	Dec	Axis number (read-only, monitoring)
11	wState	INT	Dec	Status (read-only, monitoring)
12	wErrorCode	INT	Dec	Failure Code (Read-Only, Monitoring)
13	fsetpos	REAL	Dec	Set location (read-only, monitoring)
14	factpos	REAL	Dec	Feedback location (read-only, monitoring)
15	fsetvel	REAL	Dec	Set Speed (Read Only, Monitor)
16	factvel	REAL	Dec	Feedback speed (read-only, monitoring)
17	sAxis_y	_sGROUPAXIS_INFO		Y-axis status (read-only, monitoring)
18	wAxisID	INT	Dec	Axis number (read-only, monitoring)
19	wState	INT	Dec	Status (read-only, monitoring)
20	wErrorCode	INT	Dec	Failure Code (Read-Only, Monitoring)
21	fsetpos	REAL	Dec	Set location (read-only, monitoring)
22	factpos	REAL	Dec	Feedback location (read-only, monitoring)
23	fsetvel	REAL	Dec	Set Speed (Read Only, Monitor)
24	factvel	REAL	Dec	Feedback speed (read-only, monitoring)
25	sAxis_z	_sGROUPAXIS_INFO		z-axis status (read-only, monitoring)
26	wAxisID	INT	Dec	Axis number (read-only, monitoring)
27	wState	INT	Dec	Status (read-only, monitoring)
28	wErrorCode	INT	Dec	Failure Code (Read-Only, Monitoring)
29	fsetpos	REAL	Dec	Set location (read-only, monitoring)
30	factpos	REAL	Dec	Feedback location (read-only, monitoring)
31	fsetvel	REAL	Dec	Set Speed (Read Only, Monitor)
32	factvel	REAL	Dec	Feedback speed (read-only, monitoring)
33	sAxis_a	_sGROUPAXIS_INFO		Auxiliary Axis Status (Read Only, Monitoring)

For example, write the X-axis coordinates of the center of an axis group to D3010:



5.7.13 _sGROUPPOS_INFO Axis Target Position

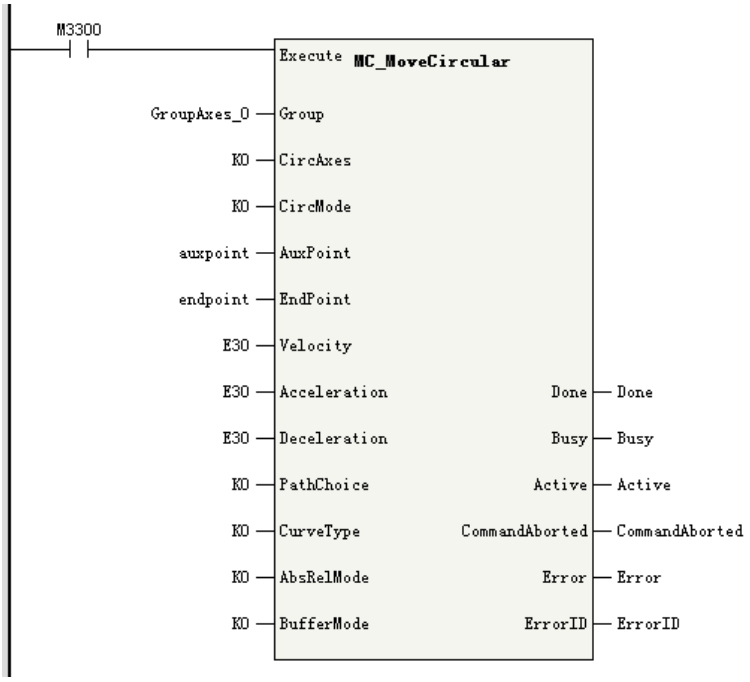
Name	Type	Description
px	REAL	The position of the X-axis
py	REAL	The position of the Y-axis
pz	REAL	The position of the Z-axis
pa	REAL	The position of the auxiliary axis

This structure sets the target position of a circular arc as an input parameter to the MC_MoveCircular.

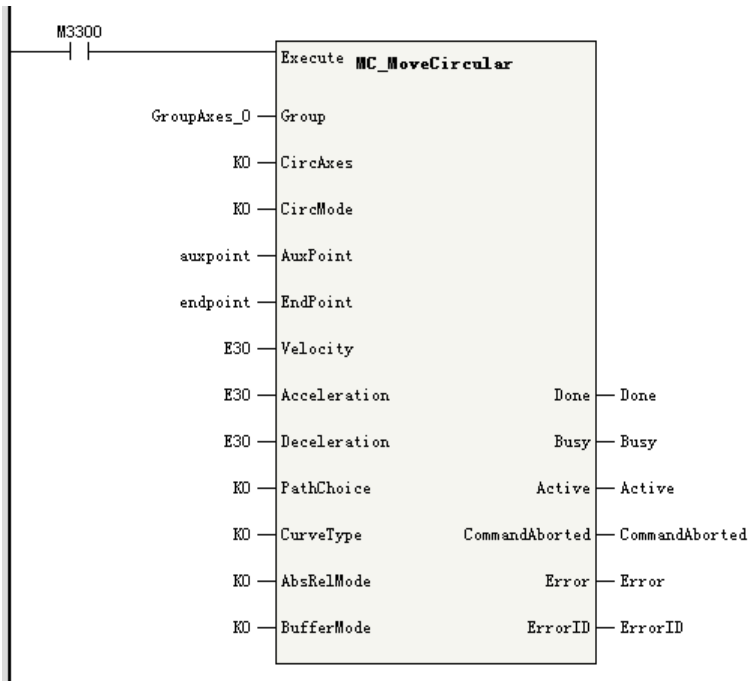
1. Create a global variable

NO.	Variable...	Data Type	Initial Value	Power Down Hold
1	auxpoint	stru1	...	Non Retained
3	endpoint	_sGROUPPOS_INFO	...	Non Retained

2. Assign values to the global variable



3. Call the instruction MC_MoveCircular



5.7.14 _SYS_PN PN Operation Status

Preconditions

- Firmware version: V6.5.0.0 and later
- Software version: AutoShop V4.11.0.0 and later

Introduction

The following table lists information of the _tagPNStack variable.

Name	Data Type	R/W	Description
_tagPNStack	_sPNStack	-	PN Stack operation parameter
_tagPNStack._connected	BOOL	R	Connection state between PN and the master
_tagPNStack._enable	BOOL	R	PN enable state

5.8 Timer

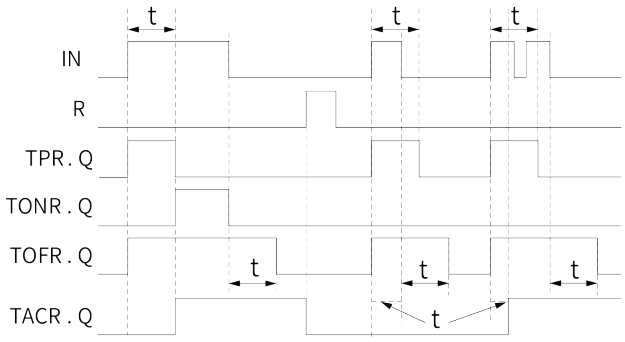
5.8.1 Introduction

H5U supports four types of timers with the reset function, including the pulse timer (TPR), connection delay timer (TONR), off delay timer (TOFR), and accumulation timer. For details, see the IEC61131-3 standard.

The time base of the timers is 1 ms, and the timer count value and state are updated when the timer instruction is executed. The program supports a maximum of 4096 timer instructions. The instruction parameters of these four types of timers are the same, which are listed as follows.

Name	Definition	Data Type	Description
IN	Instruction execution input	-	Start input
PT	Input variable	DINT	Delay time
R	Input variable	BOOL	Reset input
Q	Output variable	BOOL	Timer output
ET	Output variable	DINT	Current timing time

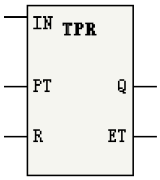
Timer timing



5.8.2 Pulse Timer (TPR)

When the IN input of the timer instruction changes from OFF to ON, the timer starts timing and the output Q turns ON. At this time, no matter how the IN input flow changes, Q remains ON for the time period specified by PT. When the timing duration reaches the time period specified by PT, Q changes to OFF.

During timing of the timer, ET outputs the current timing duration. After the timing duration reaches the value specified by PT, if the IN input flow is ON, the ET value is retained; if the IN input flow is OFF, the ET value becomes 0.



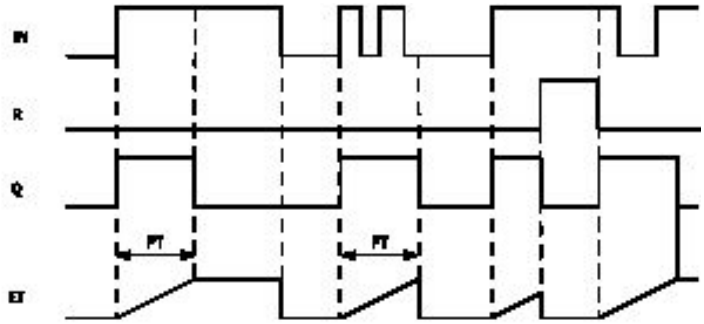
During timing, if the reset input R changes from OFF to ON, the timing duration of the TPR timer is reset to 0, and the output Q turns OFF. After the reset input R turns OFF, if the IN input flow is active, the timer resumes timing.

Description of parameters:

PT ranges from 0 to 2147483647 ms (about 24 days). If the value of PT is less than or equal to 0, it is considered 0.

Timing diagram

The following figure shows the timing diagram of the parameters IN, R, Q, and ET.



NOTE

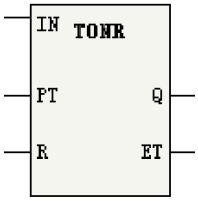
The output parameters "ET" and "Q" are updated when this instruction is executed. Therefore, the change in the state of "Q" is not at the time when the elapsed time after the timer starts equals "PT", but at the time when the instruction is executed for the first time after the elapsed time after the timer starts reaches "PT". That is, the delay of the output parameters can be up to one cycle.

5.8.3 On-Delay Timer (TONR)

When the IN input of the timer instruction changes from OFF to ON, the timer starts timing and the output Q turns ON. During the period when the IN input flow remains ON, the running time of the timer is the time specified by PT. After the timing duration reaches the time period specified by PT, Q turns ON. During the timing process or after timing is completed, when the IN input flow changes to OFF, timing ends and Q turns OFF.

When the IN input flow is ON, ET outputs the current timing duration during timing of the timer, and the ET value is retained after the timing duration reaches the value specified by PT. When the IN input flow is OFF, the ET value becomes 0.

During timing, if the reset input R changes from OFF to ON, the timing duration of the TONR timer is reset to 0, and the output Q turns OFF. After the reset input R turns OFF, to resume timer timing, you need to set the IN input flow to ON again.



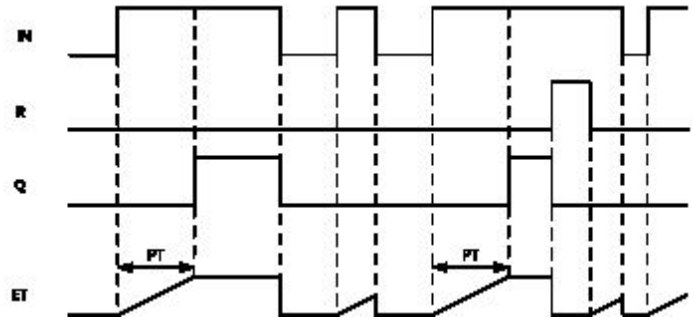
During timing, if the reset input R changes from OFF to ON, the timing duration of the TONR timer is reset to 0, and the output Q turns OFF. After the reset input R turns OFF, if the IN input flow is active, the timer resumes timing.

Description of parameters:

PT ranges from 0 to 2147483647 ms (about 24 days). If the value of PT is less than or equal to 0, it is considered 0.

Timing diagram

The following figure shows the timing diagram of the parameters IN, R, Q, and ET.



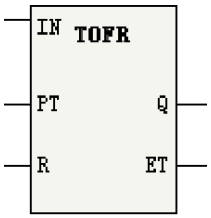
NOTE

The output parameters "ET" and "Q" are updated when this instruction is executed. Therefore, the change in the state of "Q" is not at the time when the elapsed time after the timer starts equals "PT", but at the time when the instruction is executed for the first time after the elapsed time after the timer starts reaches "PT". That is, the delay of the output parameters can be up to one cycle.

5.8.4 Off-Delay Timer (TOFR)

When the IN input of the timer instruction changes from OFF to ON, the timer starts timing and the output Q turns ON. When the IN input flow changes from ON to OFF, during the period when the IN input flow remains ON, the running time of the timer is the time specified by PT. After the timing duration reaches the time period specified by PT, Q turns OFF.

When the IN input flow is ON, the ET output is 0. When the IN input changes from ON to OFF, ET outputs the current timing duration during timing of the timer, and the ET value is retained after the timing duration reaches the value specified by PT.



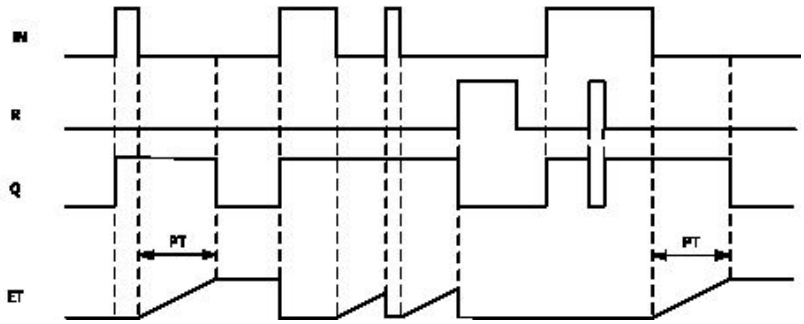
When the IN input flow is ON, if the reset input R changes from OFF to ON, the output Q turns OFF; if R resumes OFF, the output Q resumes ON. When the IN input flow changes from ON to OFF, if the reset input R changes from OFF to ON during the timing process or after timing is completed, the output Q turns OFF, and ET is reset to 0. After the reset input R turns OFF, to resume timer timing, you need to set the IN input flow to OFF again.

Description of parameters:

PT ranges from 0 to 2147483647 ms (about 24 days). If the value of PT is less than or equal to 0, it is considered 0.

Timing diagram

The following figure shows the timing diagram of the parameters IN, R, Q, and ET.



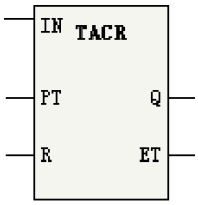
NOTE

The output parameters "ET" and "Q" are updated when this instruction is executed. Therefore, the change in the state of "Q" is not at the time when the elapsed time after the timer starts equals "PT", but at the time when the instruction is executed for the first time after the elapsed time after the timer starts reaches "PT". That is, the delay of the output parameters can be up to one cycle.

5.8.5 Accumulation Timer (TACR)

When the IN input flow of the timer instruction is ON, if the timer value has not reached the time period specified by PT, the timer continues to count, and the output Q is OFF; when the timing duration reaches the time period specified by PT, Q turns ON. During the timing process, if IN changes from ON to OFF, the timing duration is retained. When IN turns ON again, the timer starts counting from the current retained value. After the time specified by PT is reached, Q becomes ON.

When the IN input flow is ON, ET outputs the current timing value. After the timing duration reaches the time period specified by PT, the ET value is retained. When the IN input flow turns OFF, ET remains unchanged.



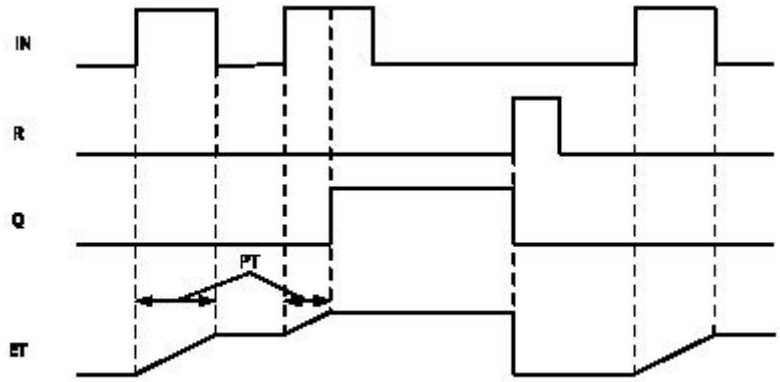
During the timing process or after timing is completed, if the reset input R changes from OFF to ON, the output Q turns OFF, and ET is reset to 0. After the reset input R turns OFF, if the IN input flow is active, the timer resumes timing.

Description of parameters:

PT ranges from 0 to 2147483647 ms (about 24 days). If the value of PT is less than or equal to 0, it is considered 0.

Timing diagram

The following figure shows the timing diagram of the parameters IN, R, Q, and ET.



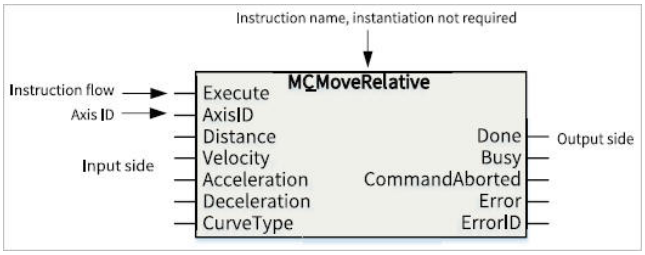
NOTE

The output parameters "ET" and "Q" are updated when this instruction is executed. Therefore, the change in the state of "Q" is not at the time when the elapsed time after the timer starts equals "PT", but at the time when the instruction is executed for the first time after the elapsed time after the timer starts reaches "PT". That is, the delay of the output parameters can be up to one cycle.

5.9 Graphical Block Instructions

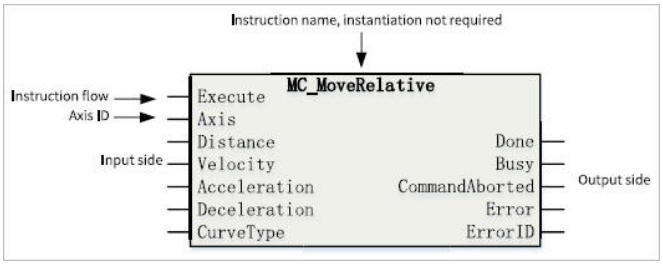
5.9.1 Instruction Composition

Some instructions support graphical block programming. An graphical block instruction is composed of the instruction name, flow signal, input side, and output side. The following figure shows the composition of a graphical block instruction of a motion control axis.



The floating-point numbers such as the target position and target velocity in the instructions are single-precision floating-point data. Therefore, the values in the instructions must meet the requirements of the range and precision of single-precision floating-point data when being processed in the PLC program. That is, a value should fall between -3.4E38 and +3.4E38, with a maximum of 7 significant digits. If a value has more than 7 significant digits, the excess part will be automatically rounded.

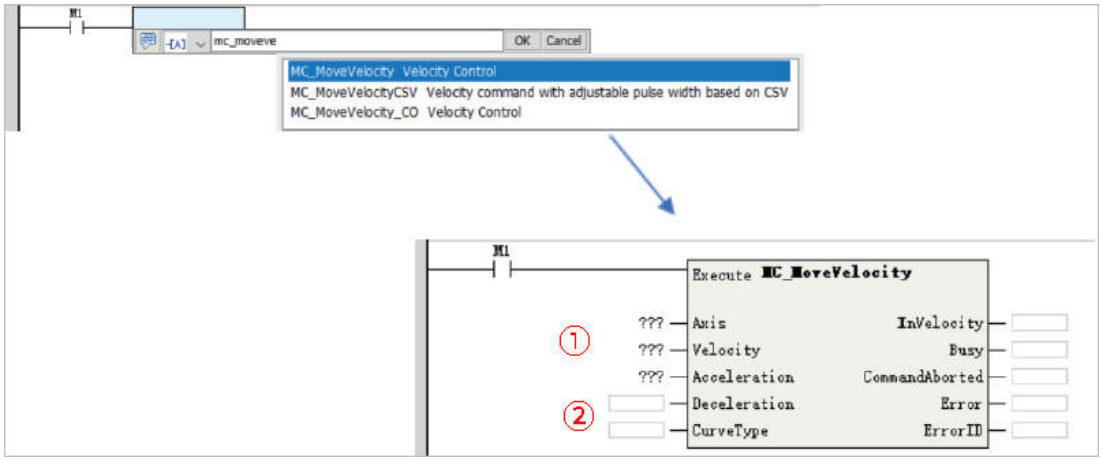
Since AutoShop 4.0.0.0 with PCB software 3.0.0.0, the motion control axis control instructions (EtherCAT/ pulse output, pulse input) of graphical blocks support access by axis name. "AxisID" is changed to "Axis", and access by axis ID is still supported, as shown in the following figure.



5.9.2 Instruction-Level Programming

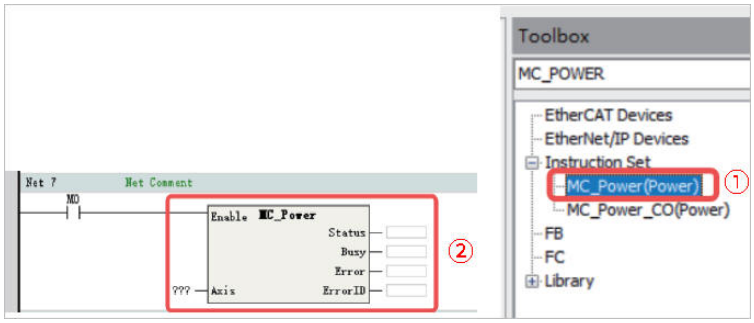
During programming, you only need to enter the name of a graphical block instruction and simply press the "Enter" key to add the graphical block instruction to the program network. You can also directly edit the instruction parameters.

- When editing a ladder diagram, enter an instruction name or select an instruction name according to the instruction prompt and click "OK". The graphical block instruction is added to the ladder diagram network, as shown in the following figure.



- Enter parameters in the graphical block instruction to complete editing of the graphical block instruction. In the instruction, parameters (with "???") next to ① are mandatory, and parameters next to ② are optional. If a parameter is not used, the default parameter value is used automatically in the instruction input, and the state cannot be obtained in the instruction output in the program or during monitoring and commissioning.

- All instructions under "Instruction Set" in the "Toolbox" pane are in graphical block mode. During programming, you can directly double-click an instruction under "Instruction Set" to add the instruction to the current focus position of the ladder diagram, as shown in the following figure.



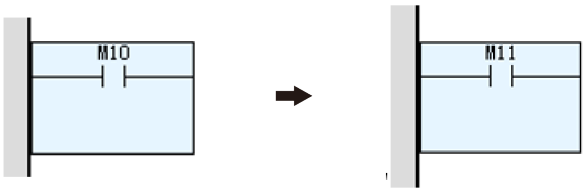
①: Double-click an instruction to add it to the ladder diagram. ②: The instruction is added successfully.

5.9.3 Label Number Function

Graphical blocks can be used to quickly increase or decrease label numbers and implement incremental paste.

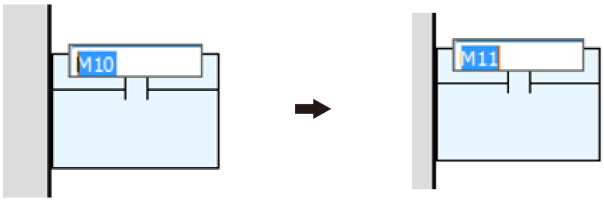
Quickly Increasing/Decreasing Label Numbers

When editing the ladder diagram, you can press "Alt"+↑/↓ to quickly increase or decrease the label number of an element or array subscript, as shown in the following figure.

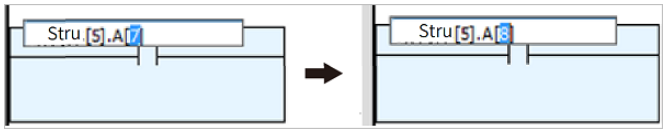


Select M10 and then press Alt+Up to modify M10 to M11.

- This function can be used during command editing, as shown in the following figure.

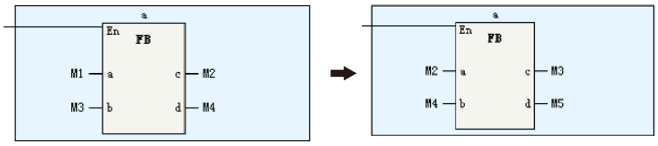


- For complex array variables, you can select the array subscript that needs to be increased or decreased, as shown in the following figure.



Click the subscript of "A" to operate the number "7".

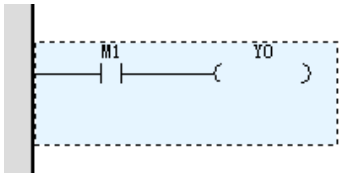
- When a function block is selected, the operation will be performed on all pins, as shown in the following figure.



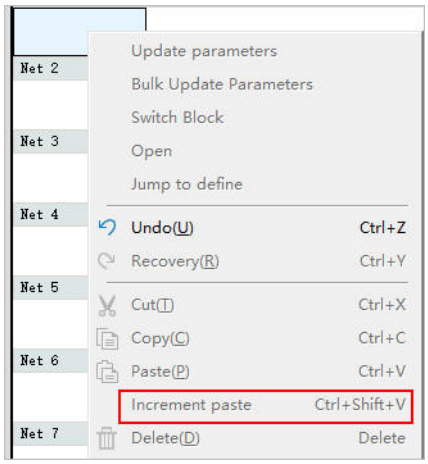
Incremental Paste

When editing the ladder diagram, you can use the incremental paste function to continuously paste the copied elements for multiple times. At the same time, the element number or array subscript can be specified during the process.

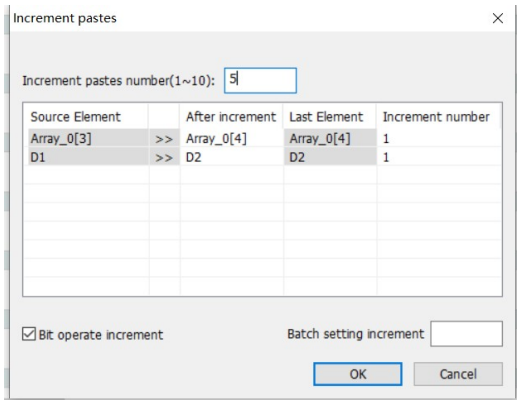
1. Select an element in the ladder diagram and press "Ctrl"+"C", or right-click the element and choose "Copy", as shown in the following figure.



2. Right-click the destination position and choose "Increment paste" from the shortcut menu (or press "Ctrl"+"Shift"+"V"), as shown in the following figure.

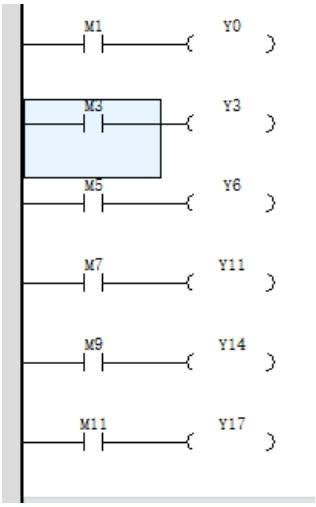


3. Specify the increment value and paste times in the displayed configuration window, as shown in the following figure.



- "Incremental pastes number (1-10)": You can set the paste times.
- "After increment": You can enter the expected value after increment, and "Increment number" is automatically calculated based on this value.
- "Increment number": You can set the increment in the target element each time a paste operation is performed.
- "Bit operate increment": During bit operation of an element, if this option is selected, the increment applies to the bit operation of the target element.
- "Batch setting increment": You can set the increments in batches.

4. Click "OK". The paste operation is performed based on the configuration, as shown in the following figure.



5.10 Subprogram

5.10.1 Introduction

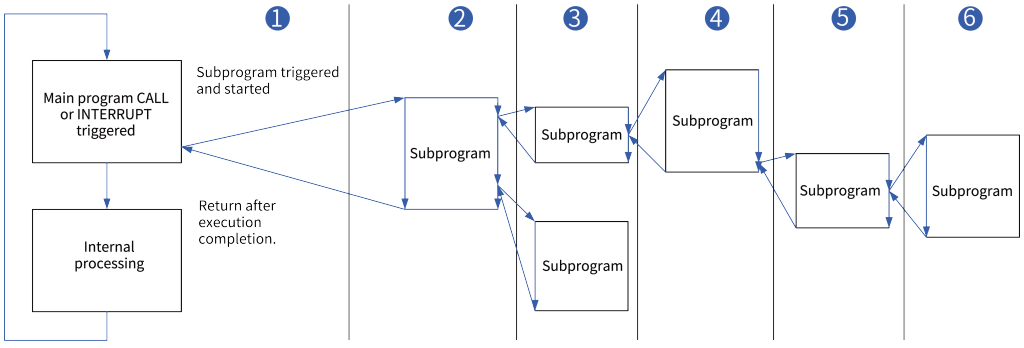
5.10.1.1 Introduction of Subprogram

The following table lists the subprogram categories and corresponding description.

Code	Name	Description
SBR	Subprogram	Up to 1024 subprograms are supported. Subprograms can be set as common subprograms or encrypted subprograms. Common subprograms and encrypted subprograms have infinite capacity and share the system capacity of 200,000 steps.
INT	Interrupt subprogram	External interruption: X000 to X003 input interrupt, including the rising edge, falling edge, and rising and falling edges
		Timed interruption: 4 points (time base = 1 ms)
		Comparison interruption: 16 points ranging from 1 to 16

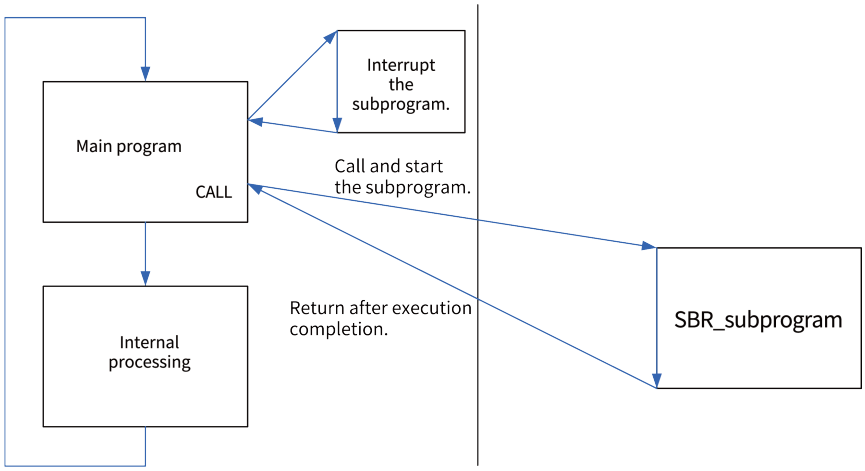
5.10.1.2 Subprogram Execution Mechanism

The following figure shows the execution logic and circular scanning methods of the main program and subprograms.



Layers of nested subprograms

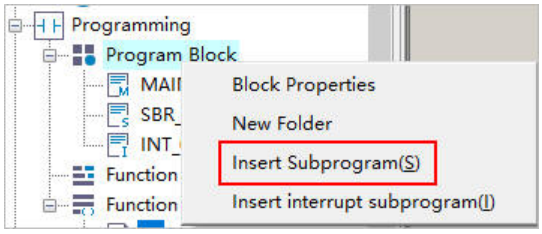
A maximum of six subprograms can be nested. The main program calls a subprogram as the first layer. One layer is added each time a subprogram is called. If the call of the nested layer is returned, the number of nested layers is not increased, as shown in the following figure.



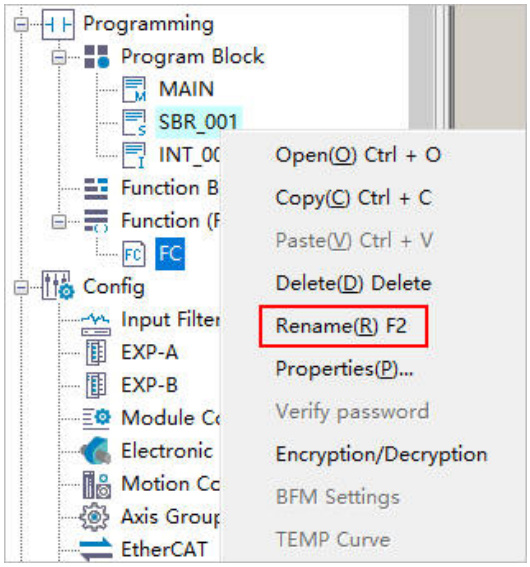
5.10.2 General Subprogram Application

5.10.2.1 Creating a General Subprogram

In the "Project Manager" tree, unfold "Programming", right-click "Program Block" or a folder under "Program Block", and select "Insert Subprogram". The new subprogram is displayed under "Program Block".



Subprogram naming rule: SBR_SN, in which the subprogram SN can be changed during renaming or property modification.



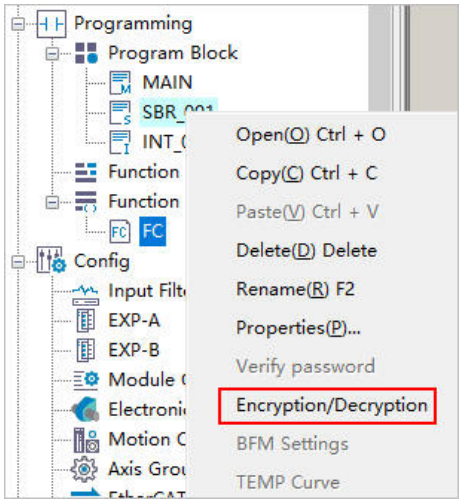
5.10.2.2 Calling the General Subprogram

The following figure shows how to call a general subprogram.

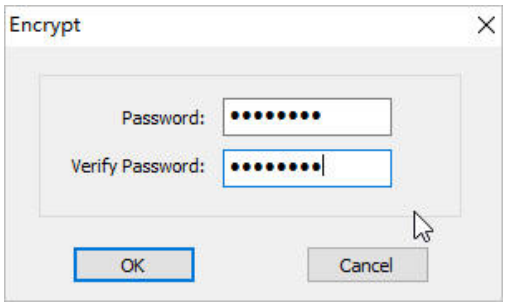
5.10.3 Encrypting Subprogram Application

5.10.3.1 Encrypting a General Subprogram

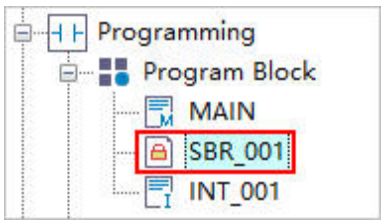
1. Encrypt the SBR_001 general subprogram as an example. Right-click SBR_001 and select "Encryption/Decryption".



2. In the "Encrypt" dialog box displayed, set "Password" and "Verify Password" for the SBR_001 general subprogram.

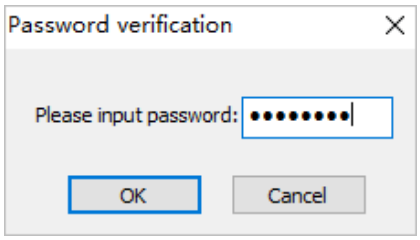


After encryption, the SBR_001 general subprogram is shown in the following figure.



If you repeat the preceding operations on the encrypted general subprogram, the subprogram will be decrypted.

To access an encrypted general subprogram, you can double-click it or right-click it and select "Verify password". In the pop-up dialog box, input the correct password.



5.10.3.2 Calling the Encrypted Subprogram

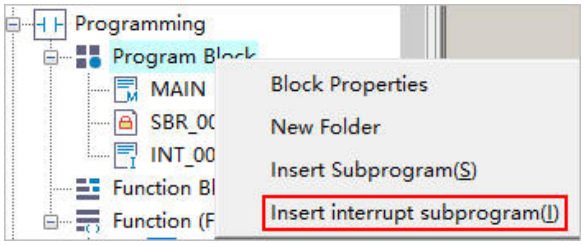
The method of calling an encrypted subprogram is the same as that of calling a general subprogram.

5.10.4 Interrupt Subprogram Application

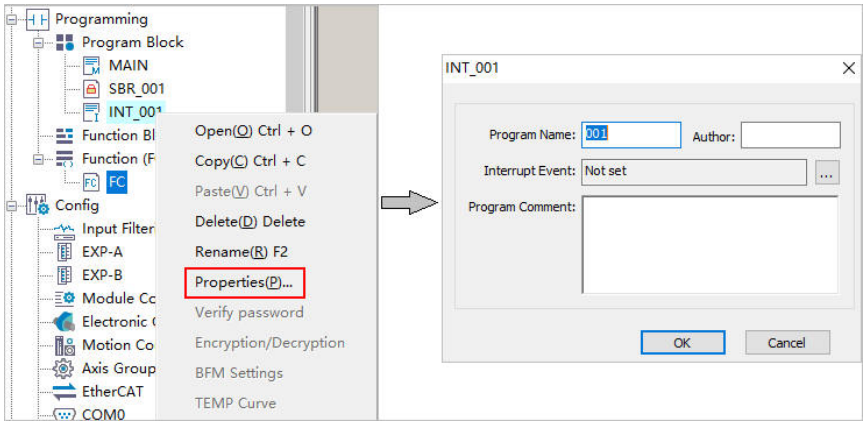
5.10.4.1 External Interrupt Subprogram

External interrupt subprograms must be immediately executed to respond to external input signals. External interrupt subprograms are executed regardless of scan cycles.

1. In the "Project Manager" tree, unfold "Programming", right-click "Program Block" or a folder under "Program Block", and select "Insert interrupt subprogram".

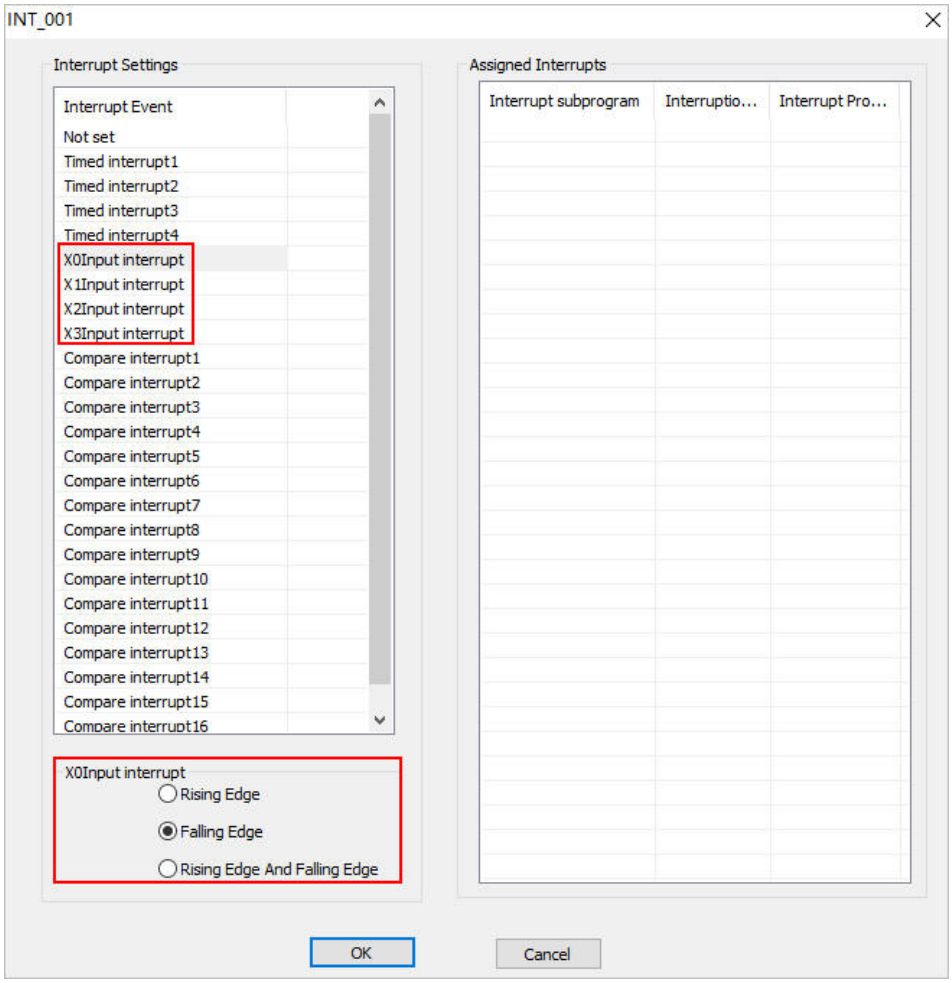


2. Right-click the inserted interrupt subprogram (such as INT_001 in the figure above) and select "Properties" to open the interrupt subprogram settings page as shown in the following figure.

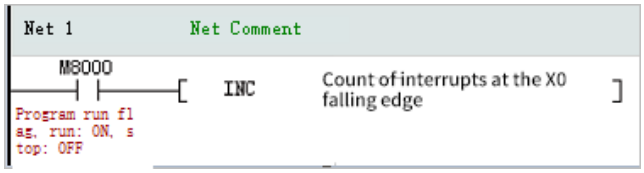


3. Click next to the "Interrupt Event" field to open the interrupt selection page.

4. Select an external interrupt, such as X0 input interrupt, and then select the corresponding property, such as "Rising Edge", "Falling Edge", and "Rising Edge And Falling Edge".



5. Write interrupt subprograms in INT_001.

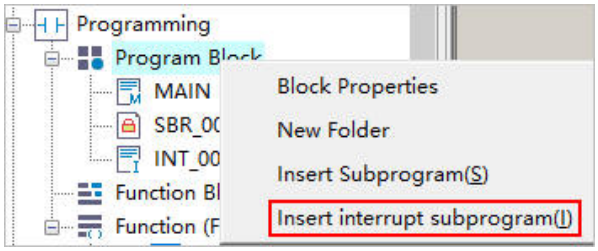


6. Enable EI in the main program. When the external interrupt conditions are met, the corresponding interrupt subprogram will be executed.

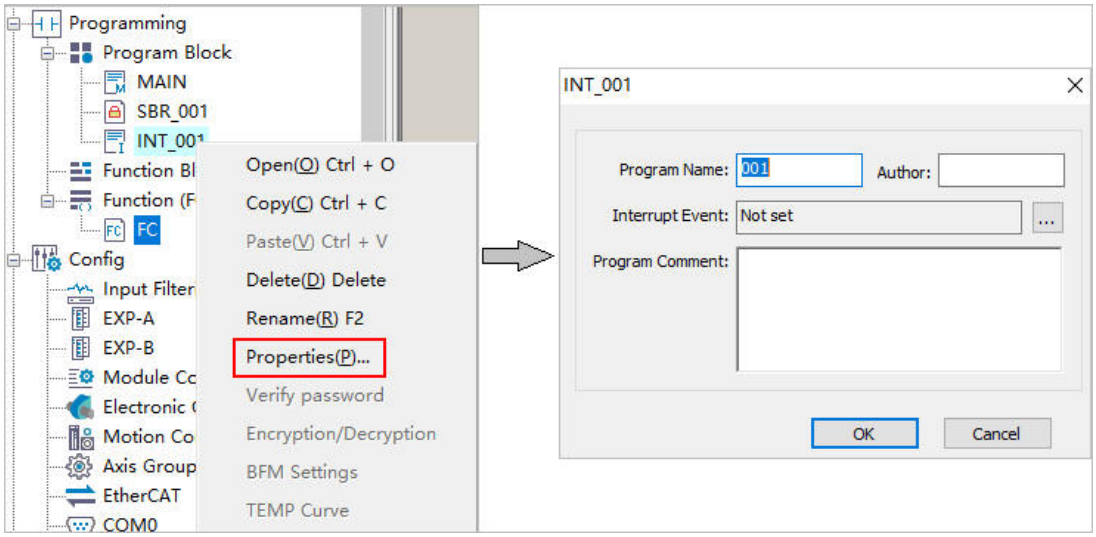
5.10.4.2 Timed Interrupt Subprogram

Timed interrupt subprograms apply to scheduled execution of set program blocks. Timed interrupt subprograms are executed regardless of scan cycles.

1. In the "Project Manager" tree, unfold "Programming", right-click "Program Block" or a folder under "Program Block", and select "Insert interrupt subprogram".

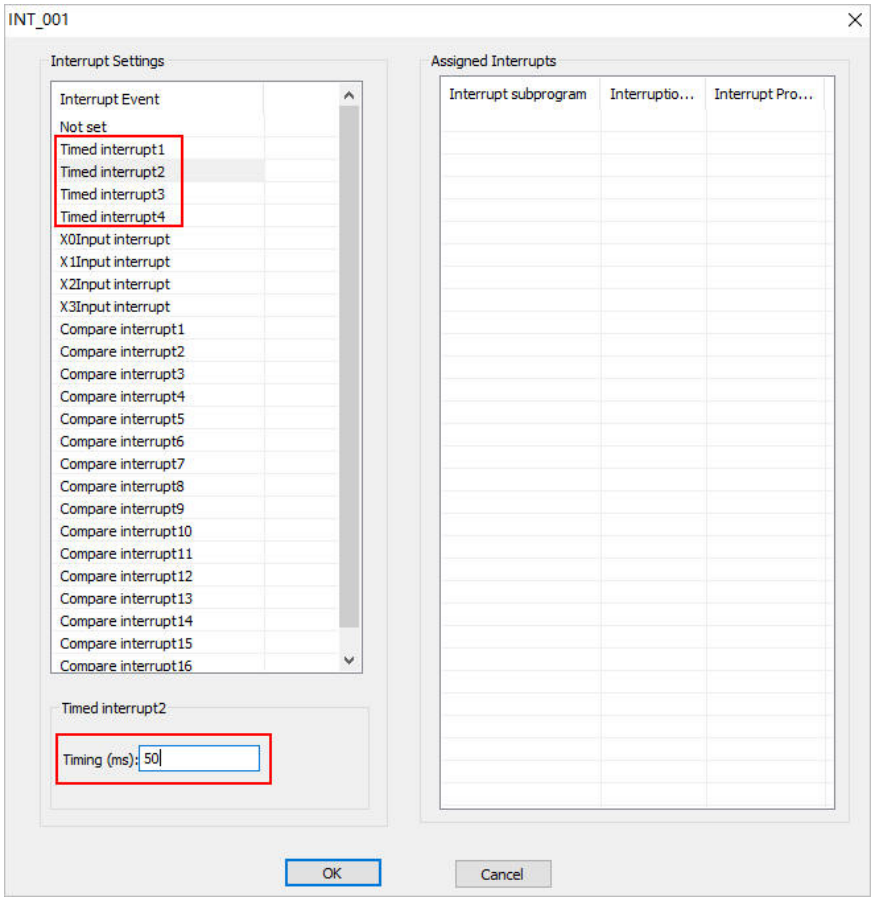


2. Right-click the inserted interrupt subprogram (such as INT_001 in the figure above) and select "Properties" to open the interrupt subprogram settings page as shown in the following figure.



3. Click next to the "Interrupt Event" field to open the interrupt selection page.

4. Select a timed interrupt and set Timing (ms) to a value ranging from 1 ms to 1000 ms.



5. Write interrupt subprograms in INT_001.

6. Enable EI in the main program. When the timed interrupt conditions are met, the corresponding interrupt subprogram will be executed.

5.10.4.3 Comparison interrupt Subprogram

Comparison interrupt subprograms must be immediately executed to respond to the setpoint of the counter axis. Comparison interrupt subprograms are executed regardless of scan cycles.

For details about the procedure, see 17.6.7 Comparison Interrupt.

5.10.4.4 Communication Interrupt Subprogram

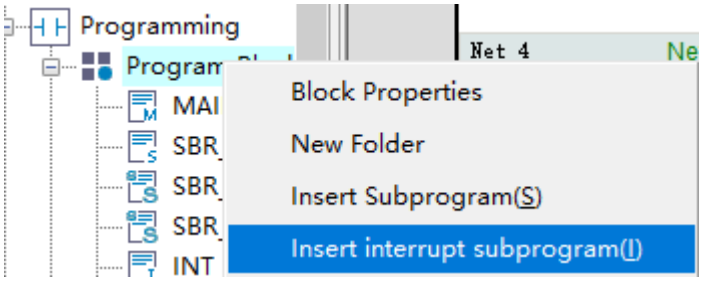
Communication interrupt subprogram is suitable for scenarios where a program needs to be executed immediately in response to the completion of a specified communication cycle. This subprogram is not affected by the PLC scan cycle.

Preconditions

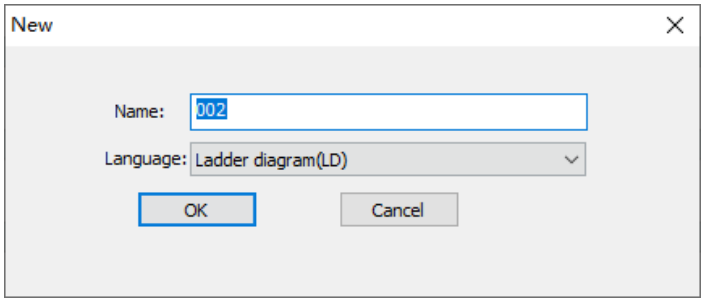
- PLC: Easy5X series
- Firmware version: V6.5.0.0 and later
- Software version: AutoShop V4.11.0.0 and later

Procedure

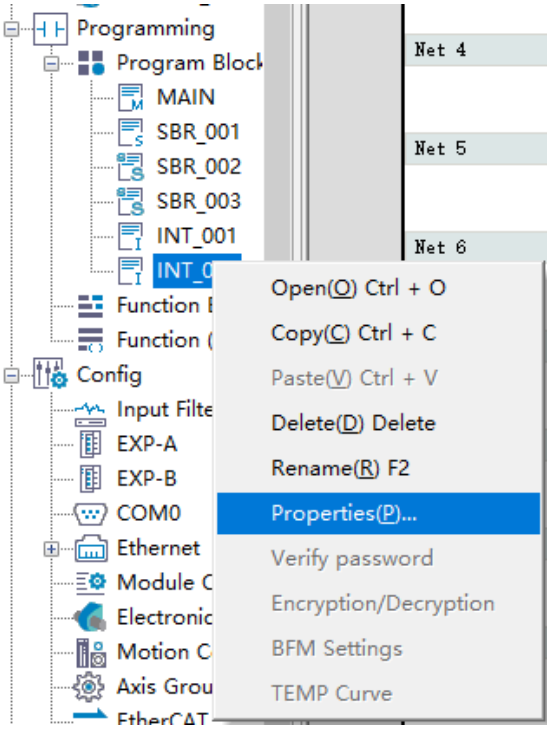
1. Under the item "Programming" in "Project Manager", right-click the "Program Block", and select "Insert interrupt subprogram".



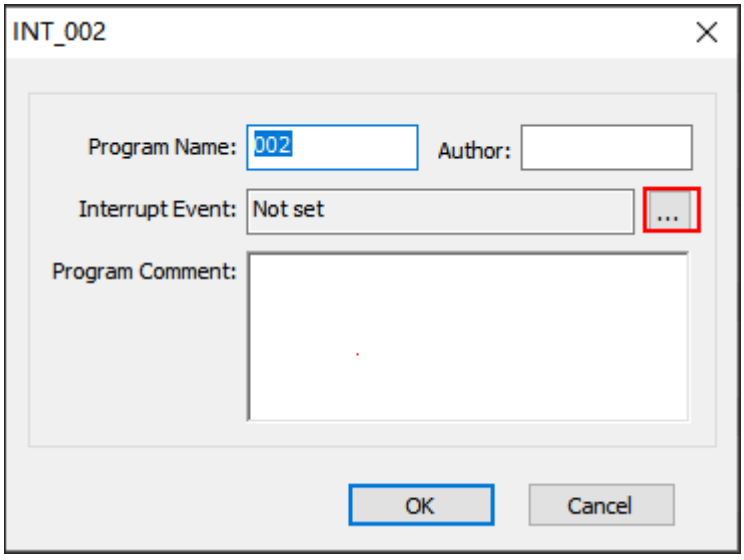
2. In the "New" dialog box displayed, set the name and language, and click "OK".



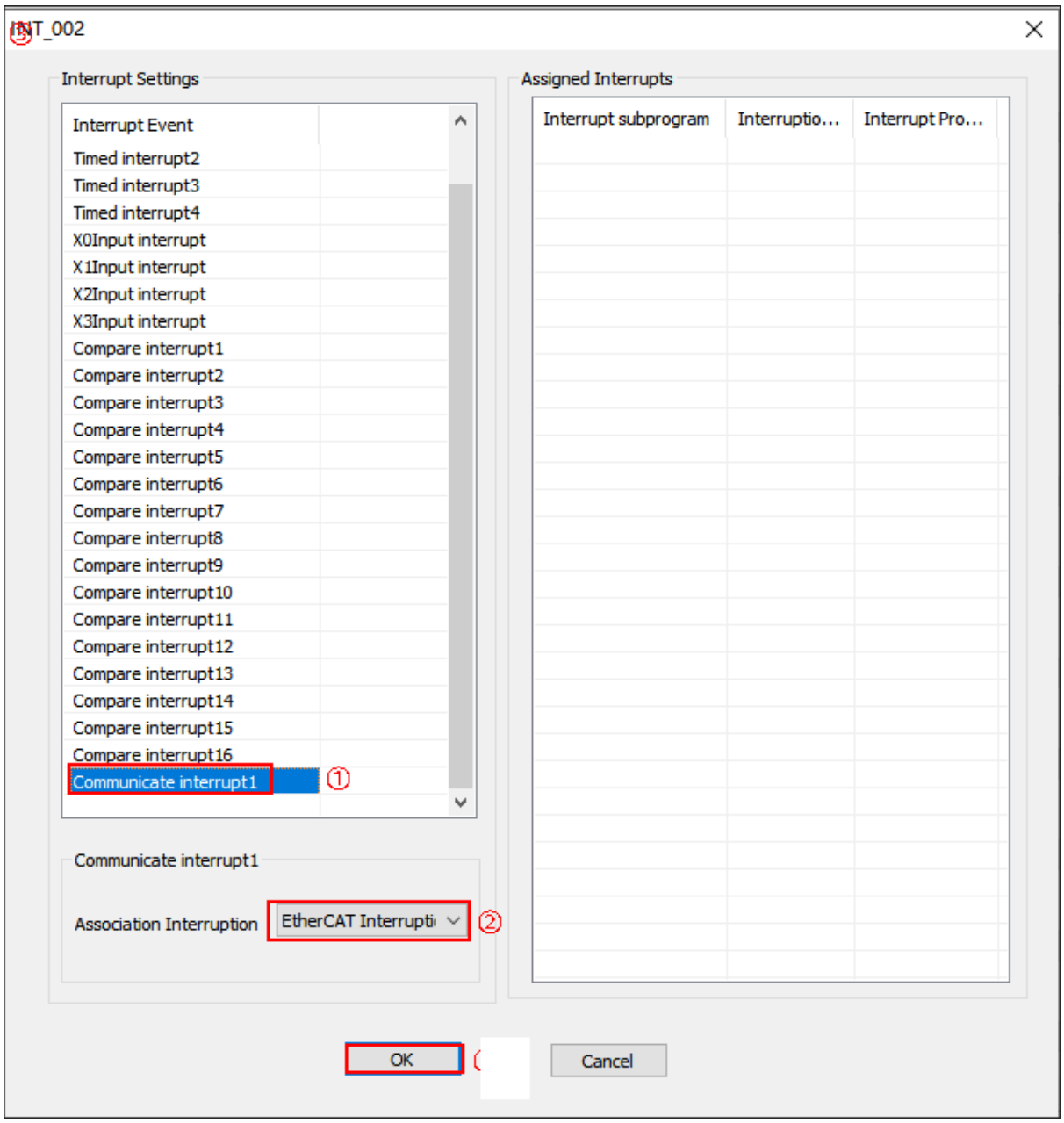
3. An interrupt program (such as INT_002) is added under "Program Block". Right-click the program and select "Insert interrupt subprogram". The interrupt subprogram setting page is displayed.



4. Click next to the "Interrupt Event" field to open the interrupt selection page.



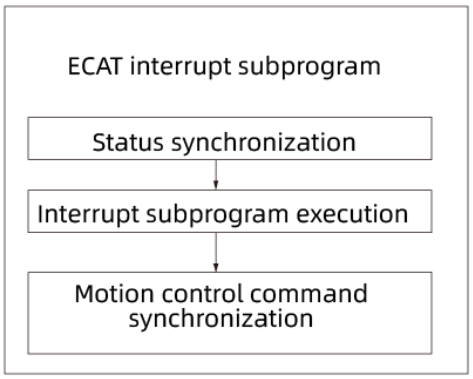
5. Select "Communicate interrupt1", set the "Association Interruption" type (default is EtherCAT Interruption), and click "OK". Then you can compile the interrupt subprogram in INT_002.



6. Enable EI in the main program. When the timed interrupt conditions are met, the associated communication task will send an interrupt signal upon completion of each communication cycle (if associated with EtherCAT interrupt, this is the EtherCAT task cycle), and thus the corresponding interrupt subprogram will be executed.

Impact of associating with EtherCAT interrupt

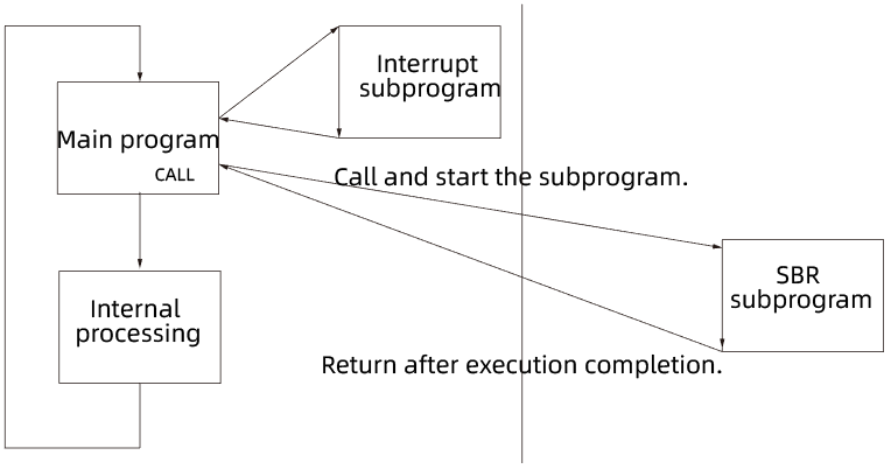
- Besides executing the program logic within the interrupt subprogram, the EtherCAT interrupt subprogram also obtains the status of and sends commands to EtherCAT axis control tasks. Therefore, when the main task cycle is longer than the EtherCAT task cycle, multiple command synchronizations can occur within one scan cycle, improving the timing of motion control instruction execution.



CAUTION

The command parameters synchronized by the EtherCAT interrupt do not include commands for updating the cam table (MC_GenerateCamTable), cam tappet (MC_DigitalCamSwitch), or online commissioning commands.

- Since axis control system variables are updated only once per main task cycle, monitoring these variables in the background will exhibit a delay.
- Enabling the communication interrupt (EtherCAT interrupt) subprogram will increase the execution time of the main task scan cycle.



EtherCAT interrupt applicability

- The EtherCAT interrupt is suitable for situations where the main task scan cycle is significantly larger than the EtherCAT cycle (for example, scan cycle 8 ms, EtherCAT cycle 2 ms). When the main task scan cycle is ≤ the EtherCAT cycle, the EtherCAT interrupt subprogram is not recommended, as the improvement in responsiveness is negligible and it increases the main task execution time.
- It is recommended that the execution time of the EtherCAT interrupt subprogram be less than the EtherCAT task cycle.
  If the amount of code in the EtherCAT interrupt subprogram causes its execution time to exceed the EtherCAT task cycle, EtherCAT interrupt signals may accumulate, leading to delayed command synchronization.

5.11 Function Blocks and Functions

5.11.1 Function Block

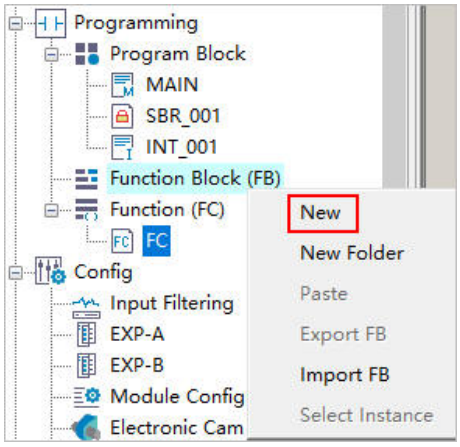
A function block (FB) abstractly encapsulates the part used repeatedly in a program into a general program block that can be called repeatedly within the program. Using encapsulated function blocks in programming can improve program development efficiency, reduce programming errors, and improve program quality.

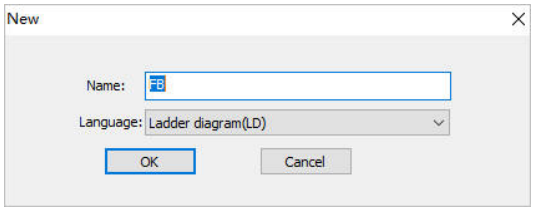
Different instances can be created based on the same function block. These instances can output one or more values during execution. The system allocates memory for internal variables of each instance, and these variables describe the running state of the function block. With the same input parameters, different instances provide different calculation results.

The basic steps of using a function block are as follows: Create a function block > Program the function block > Instantiate the function block > Run the function block > Encapsulate the function block > Import the function block.

Creating a Function Block

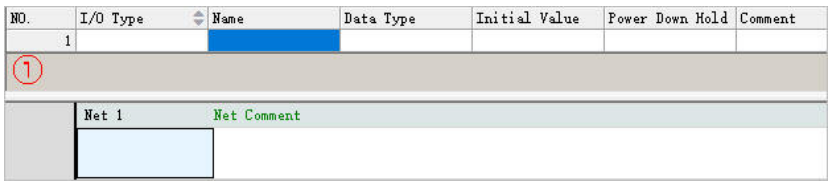
Expand the "Programming" node in the "Project Manager" tree, right-click "Function Block (FB)", or right-click a folder under "Function Block (FB)", choose "New", enter a name and set the function block language in the displayed dialog box, and click "OK". A function block is created successfully, as shown in the following figure.

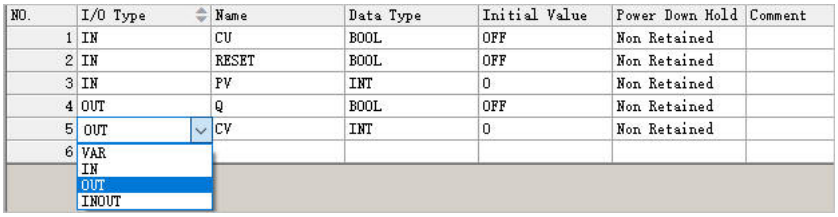




Programming the Function Block

Double-click the created function block under "Function Block (FB)" to go to the function block program editing interface. Compared with ordinary program editing, the function block program editing interface has an additional input/output and local variable definition window, as shown in the following figure.





①: Input/output and local variable definition window

- "I/O Type": attribute of the function block variable

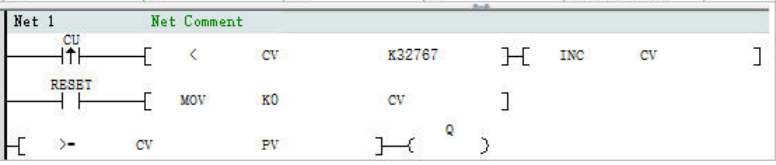
Variable Type	Type Description	Description
IN	Input variable	The parameter is provided by the function block that calls the variable, and the input is transferred to the instruction of the function block.
OUT	Output variable	The parameter is provided to the function block that calls the variable, that is, structure data is output from the function block.
INOUT	Input/Output variable	I/O variables can be transferred to the called function block and modified within the called function block.
VAR	Local variable	A local variable is only valid in the current function block and cannot be accessed externally.

- "Name": name of the variable
- "Data Type"
  The supported data types include BOOL, INT, DINT, REAL, BYTE, IP, and STRING. You can also define array variables and structure variables. To use structure variables, you need to create structure members in the structure of global variables.
- "Initial Value"
  You can set the initial value of a variable when execution starts.
- "Power Down Hold"
  This attribute allows you to choose whether to retain the value of a variable upon power failure.
  - "Non Retained": The variable resumes the specified initial value after power-on.
  - "Retained": If you select "Re-initialize retentive variables when downloading", the variable resumes the specified initial value during program downloading; otherwise, it retains the previous value.

The function block program can call functions (FC) or function blocks (FB) and supports up to 8 levels of nested calls. In addition to variables, the function block program can also use supported elements, such as M8000, as global variables.

Example 1: Counting up with FB encapsulation - ladder diagram (LD)

NO.	I/O Type	Name	Data Type	Initial Value	Power Down Hold	Comment
1	IN	CU	BOOL	OFF	Non Retained	
2	IN	RESET	BOOL	OFF	Non Retained	
3	IN	PV	INT	0	Non Retained	
4	OUT	Q	BOOL	OFF	Non Retained	
5	OUT	CV	INT	0	Non Retained	



Example 2: Counting up with FB encapsulation - structured text (LiteST)

NO.	I/O Type	Name	Data Type	Initial Value	Power Down Hold	Comment
1	IN	In1	INT	0	Non Retained	
2	OUT	Out1	INT	0	Non Retained	
3						

```
1  Out1 :== In1 + 1;
```

CAUTION

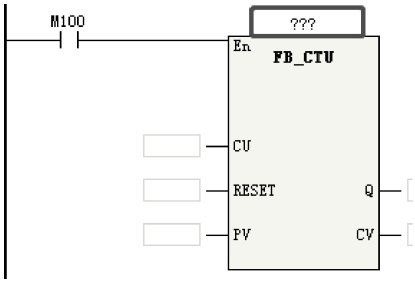
The total size of FB/FC internal definitions is up to 100 K.

Instantiating and Calling the Function Block

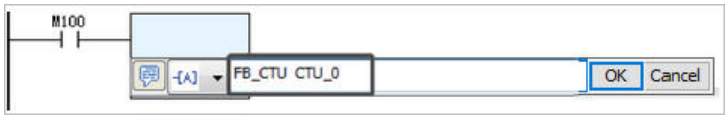
After the FB program is compiled, the function block needs to be instantiated before it can be used in ladder diagram or structured text applications.

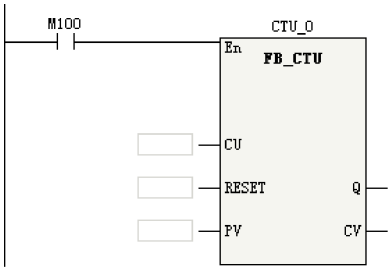
Ladder diagram (LD)

- Method 1: Directly enter the FB name in the ladder diagram application, and then enter the instance name in "???" at the top of the function block instruction and click "OK" to instantiate the function block, as shown in the following figure.



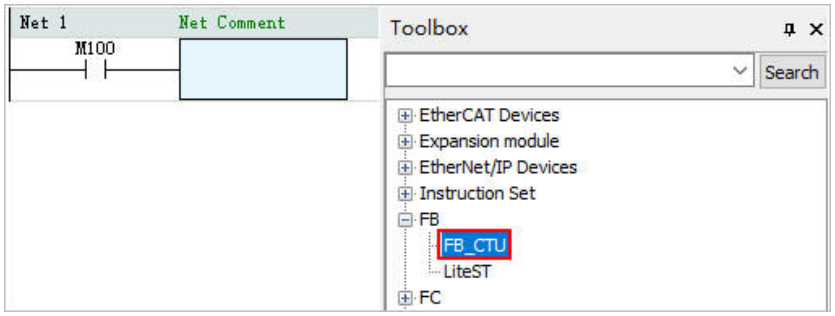
- Method 2: Directly enter the FB name+instance name in the ladder diagram application and click "OK" to instantiate the function block, as shown in the following figure.





After instantiation is completed, edit the instruction parameters in the FB instruction as required by the program to call the instantiated function block.

- Method 3: Double-click the FB instruction under "FB" of the "Toolbox" pane to add the FB instruction to the selected position in the ladder diagram, as shown in the following figure. Then enter the instance name in the graphic block instruction to complete the instantiation definition.



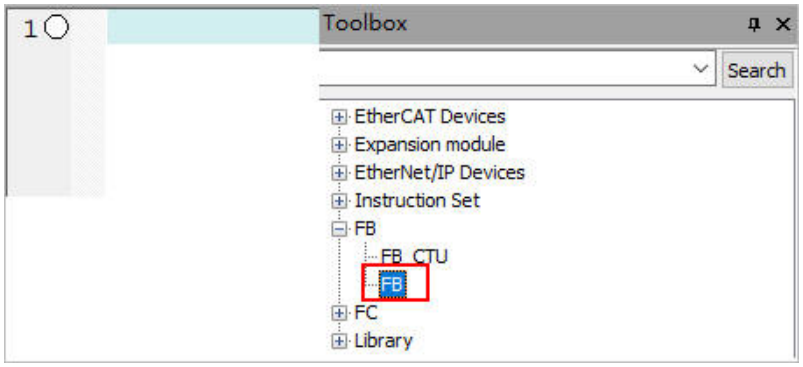
Structured text (LiteST)

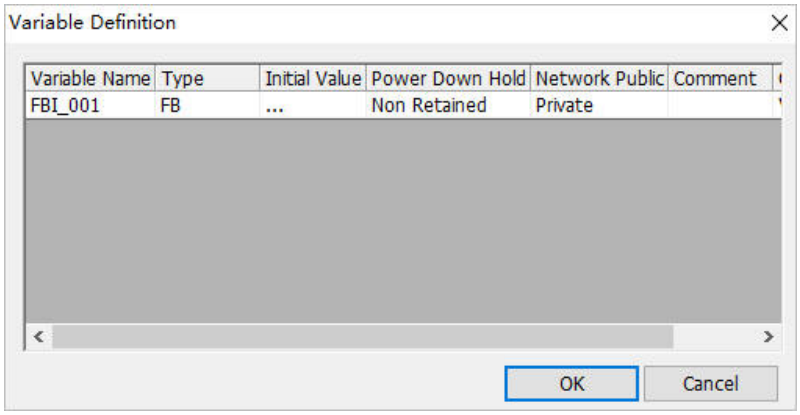
- Method 1: Declare a function block instance in the function block instance table. In an structured text application, enter the FB name directly and then press Tab, as shown in the following figure.

NO.	Variable Name	Data Type	Initial Value	Comment	Length
1	FB_001	FB	...		nBitLen:48
5					

```
1  FB_001(In1 := ,Out1 => );
```

- Method 2: Double-click the FB instruction under "FB" of the "Toolbox" pane. In the dialog box displayed, input the name of the FB instance to be declared, and click OK to add the FC instruction to the selected position in the structured text application, as shown in the following figure.

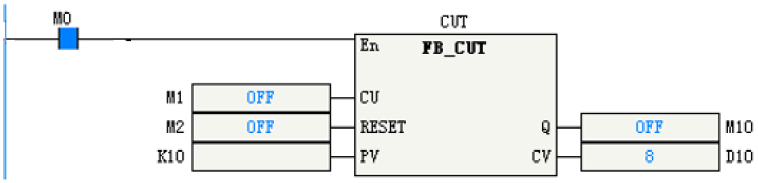




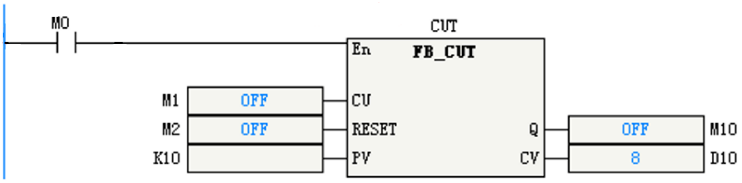
Running the Function Block

Ladder diagram (LD)

After the function block is instantiated, the En of the function block is connected to the ladder network. When the En network flow is ON, the function block program is executed, and the output of the function block changes with the input state and internal variable state. When the En network flow is OFF, the function block program is not executed, and output of the function block is not refreshed, as shown in the following figure.



When the counter function block CUT flow is ON, the function block is executed. The output CV increases by 1 when the input condition CU changes on the rising edge, as shown in the following figure.



When the counter function block CUT flow is OFF, the function block is not executed. The output CV is not refreshed when the input condition CU changes on the rising edge.

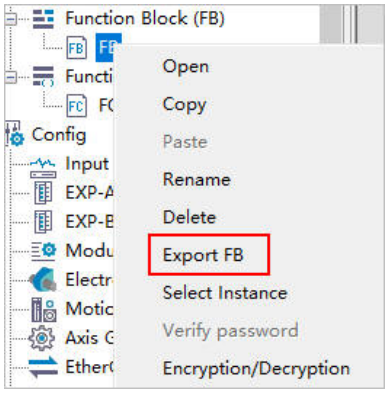
Structured text (LiteST)

When the program is executed to the function block in the structured text application, the content of the function block is executed.

Encapsulating the Function Block

The function block can be encapsulated into a library after editing and commissioning. The function block encapsulated into a library can be multiplexed in different programs through library management of AutoShop.

1. Right-click "Function Block (FB)" under "Programming" and choose "Export FB", as shown in the following figure.



2. In the "Export Library" dialog box displayed, select the function block to be encapsulated, specify the export path and version number, check whether the source is visible, and click "OK". The FB is exported to the specified location, and a function block in .fe format is generated.
   - Select "Source Visible" as required. If the source code is visible, after importing the library in the project, you can debug or modify the function block program.
   - If the source code is invisible, after the library is imported, the function block program can only be called but not viewed or modified in the project.

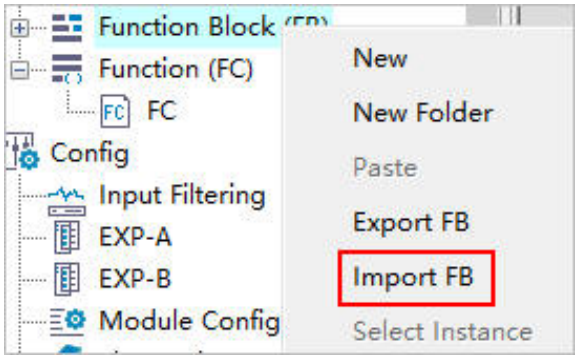
> **CAUTION**
>
> - Encrypted function blocks, function blocks that call encrypted function blocks, and function blocks that call encrypted functions cannot be selected for export.
> - When the selected function block has other function blocks or structures, its function blocks and structures are also exported.-

-

Importing the Function Block

After the function block is exported as a library, it can be called in other programs after being imported. You can import the function block library in either of the following two ways.

- Method 1: Right-click "Function Block (FB)" under "Programming" in the "Project Manager" tree and choose "Import FB" to import the library, as shown in the following figure.



  This method can only be used to import function blocks of which the source code is visible. After importing, you can double-click to open the function block program and edit and debug it. The function block library imported using this method is managed in the project. If you want to call the function block in a new project, you need to re-import the library.

- Method 2: Right-click "Library" > "Custom Library" in the "Toolbox" pane and choose "Import FB" to import the library.

This method can be used to import function blocks of which the source code is visible or invisible. The libraries imported this way are managed as custom libraries, and the function blocks in the libraries can be used directly when a new project is created. You can double-click the function block library imported in the toolbox to directly add it to the ladder diagram program or LiteST program as an instruction. If you need to view or modify a function block program of which the source code is visible, you need to import it in the project management window.

> **NOTE**
>
> When a library imported from the toolbox contains other function blocks or structures, its function blocks and structures can be selected and applied in the variable table and instance table.

5.11.2 Function

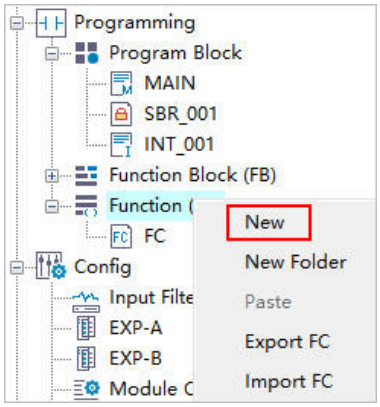
A function (FC) is an independently encapsulated program block. The program block can define input/output parameters and non-static internal variables. That is, when a function is called with the same input parameters, the output results are the same. An important feature of a function is that its internal variables are static, and there is no internal state storage. You will obtain the same output with the same input parameters. This is the main difference between a function and a function block.

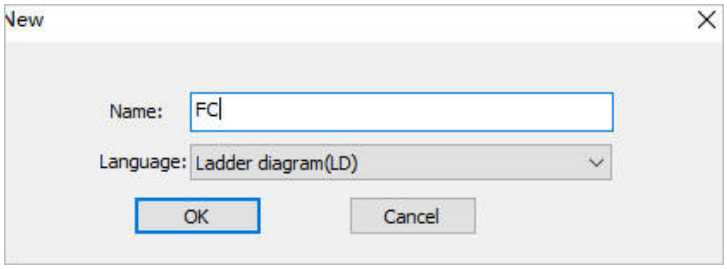
FC, as a basic arithmetic unit, is often used in various mathematical operations. For example, sin(x) and sqrt(x) are typical functions.

The basic steps of using a function are as follows: Create a function -> Program the function -> Call the function -> Run the function -> Encapsulate the function.

Creating a Function

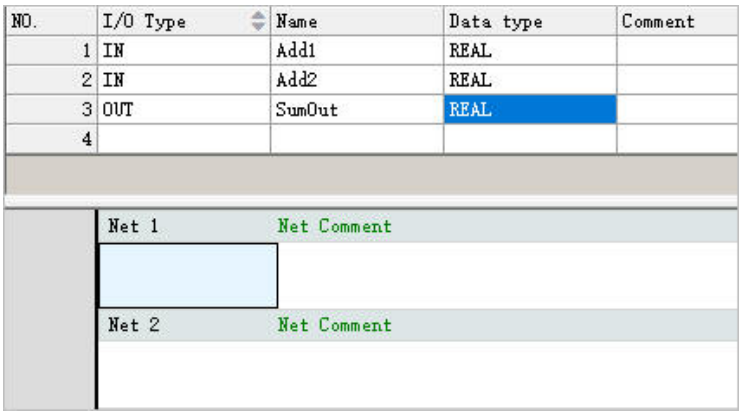
Expand the "Programming" node in the "Project Manager" tree, right-click "Function (FC)", or right-click a folder under "Function (FC)", choose "New", enter a name and set the function language in the displayed dialog box, and click "OK". A function is created successfully.





Programming the Function

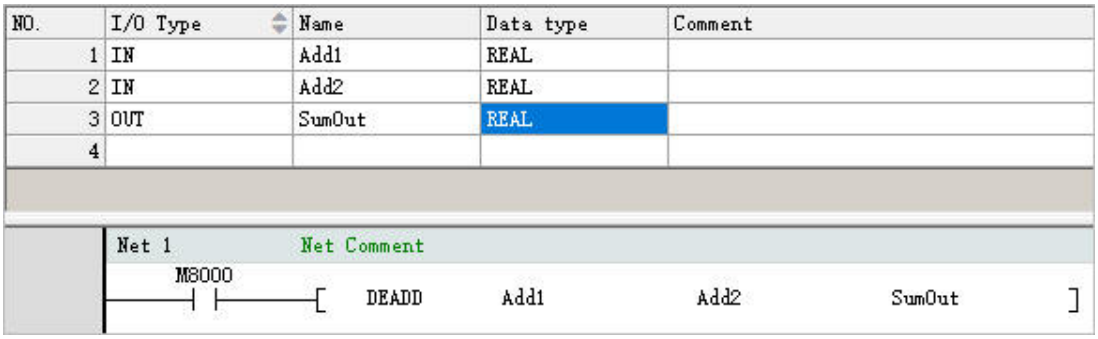
Double-click the created function under "Function (FC)" to go to the function program editing interface. The editing interface of the function program is similar to that of the function block. Compared with ordinary program editing, the function program editing interface has an additional input/output and local variable definition window.



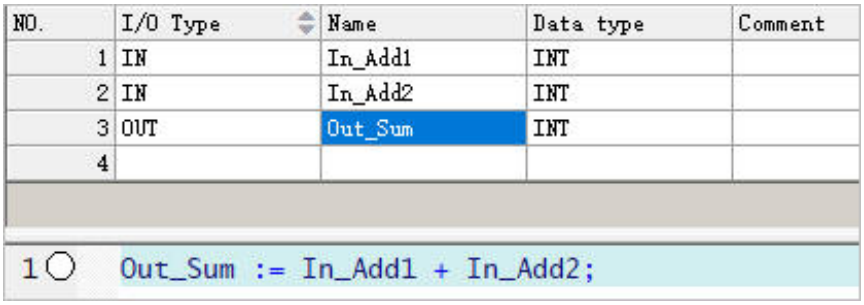
In the input/output and local variable definition window, you can define the input (IN), output (OUT), input/output (INOUT), and local variable (VAR) of a function block. The supported data types include BOOL, INT, DINT, REAL, BYTE, IP, and STRING. You can also define array variables and structure variables. To use structure variables, you need to create structure members in the structure of global variables.

- Compared with variables of function blocks, variables of functions do not support configuration of initial values, and all local variables are non-retentive.
- The function program adopts ladder diagram programming. It can call functions. A function can be called by other functions, function blocks, and programs.
- In addition to variables, the function program can also use M8000 as an always ON variable.
- In a function program, instructions related to states or executed for multiple cycles, such as LDP and MC_Power, cannot be used.

Example 1: Encapsulating the addition function - Ladder Diagram (LD)



Example 2: Encapsulating the addition function with FC - structured text (LiteST)



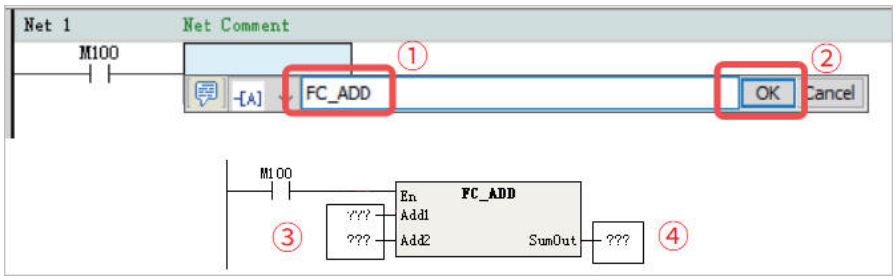
> **CAUTION**
>
> The total size of FB/FC internal definitions is up to 100 K.

Calling the Function

After a FC program is complied, it can be called directly or used in a LD or structured text application.

Ladder Diagram (LD)

- Method 1: Directly enter the function name in the ladder diagram application, press "Enter", and then edit the input/output parameters in the graphic block instruction.

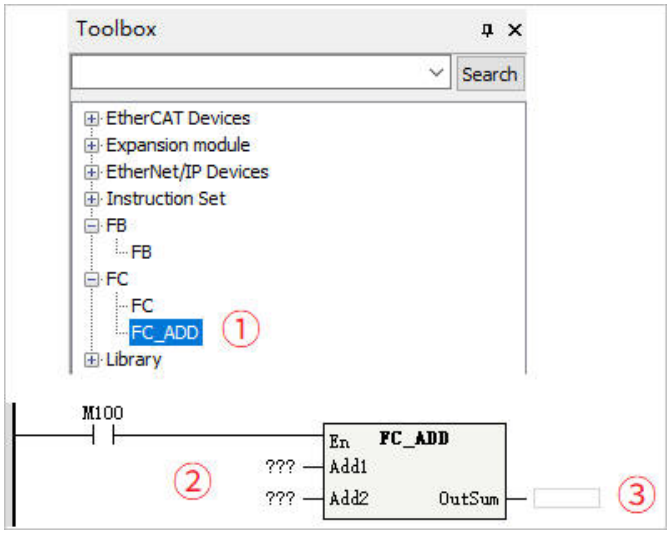


①: Enter the function name.

②: Click "OK".

③/④: Add input/output variables.

- Method 2: After a function program is created, the corresponding instruction is generated under "FC" in the "Toolbox" pane. Double-click the FC instruction under "FC" to add the FC instruction to the selected position in the ladder diagram.



①: Double-click the FC instruction to add it.
②: Add input parameters.

③: Add output parameters.

Structured text (LiteST)

- In an structured text application, enter the FC name directly and then press Tab.

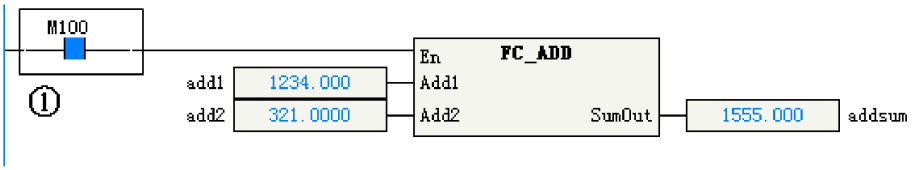
```
1  FC(In_Add1 := ???,In_Add2 := ???,Out_Sum => );
```

- After a function program is created, the corresponding instruction is generated under "FC" in the "Toolbox" pane. Double-click the FC instruction under "FC" to add the FC instruction to the selected position in the structured text application.

Running the Function

Ladder Diagram (LD)

After the function is called, the En of the function is connected to the ladder network. When the En network flow is ON, the function program is executed, and the output of the function is refreshed according to the input state operation. When the En network flow is OFF, the function program is not executed, and output of the function is not refreshed.



①: The function is executed when the En network flow is ON.

Structured text (LiteST)

When the program is executed to the function in the structured text application, the content of the function is executed.

Encapsulating the Function

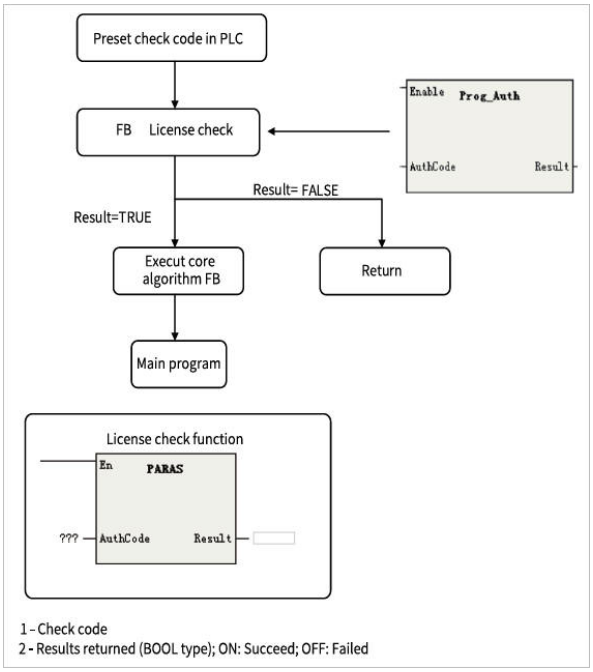
The encapsulation procedure of functions is similar to that of function blocks. For details, see the description of "Encapsulating the Function Block" in 5.11.1 Function Block.

Importing the Function

The import procedure of functions is similar to that of function blocks. For details, see the description of "Importing the Function Block" in 5.11.1 Function Block.

5.11.3 Authorizing a Function Block

By using the Prog_Auth function, the core algorithm function block is controlled and compiled into a library file. Only authorized PLCs that pass the verification can use this library file, thus protecting the intellectual property of the equipment manufacturer, as shown in the following figure.



The differences between the Prog_Auth and PARAS functions are as follows:

- The Prog_Auth function is triggered by contact. After the authorization code passes verification, the Result pin executes the user program in the function block.
- The PARAS function does not need to be triggered by contact. After the authorization code passes verification, the user program in the function block is executed. If verification fails, program execution directly returns.

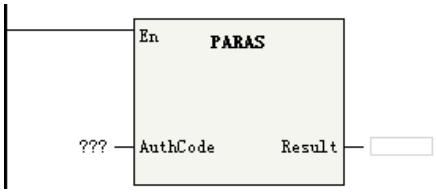
Setting the Authorization Code by Using a Tool

1. Double-click "H5U_AuthManger.exe" in the software installation directory to open it.

2. Select "Ethernet (IP address)" for "Communication type".
3. Enter the IP address of the PLC for which you want to set the authorization code in the "Connected PLC IP" field, enter the 8-digit authorization code, and click "Set Authorization Code".
4. Click "Generate Verification Code". The authorization code is generated for instructions.
5. Click "Check Authorization Code" or "Clear authorization code". (You can perform this operation only after you enter the authorization code.)

Adding a Program Block

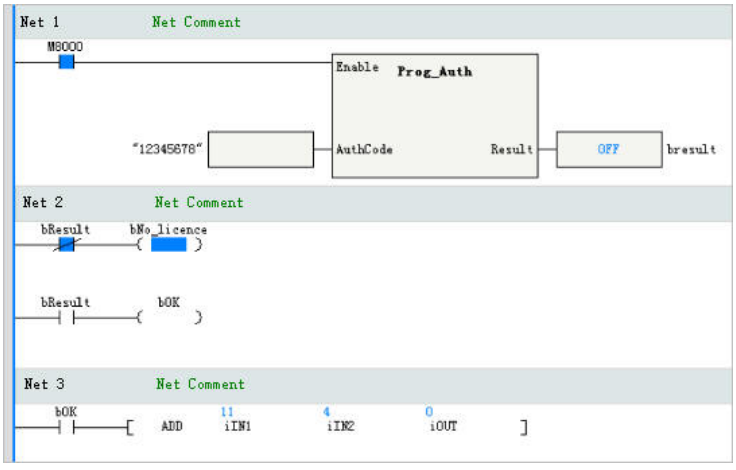
1. Open the function block to be authorized, and add the PARAS function block.



2. Enter the authorization code for instructions generated by the software in the "AuthCode" field.
3. The function block is authorized. If the authorization code of the PLC is inconsistent with that in the function block, the program in the function block cannot run.

Example

OFF is returned because the authorization code obtained by using the Prog_Auth function is different from the preset authorization code in the PLC. In this case, the ADD instruction in the following program is not executed.



5.11.4 Setting Initial Values of the Function Block

The initial values of function block (FB) settings can be modified based on the FB type or FB instance.

- Modifying the initial values based on the FB type is equivalent to modifying the initial values of the type.
- Modifying the initial values based on an FB instance is equivalent to modifying the initial values of the instance.
- If the initial values of an instance are modified, the member variables of the FB instance display the values after modification, and the background color of the cells is yellow.
- If the initial values of an instance are not modified, the member variables of the FB instance display the default values, and the background color of the cells is white.

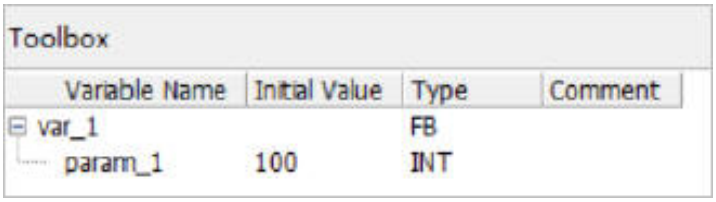
The initial values of the FB type are the default values of the instance. When the initial values of an instance are modified back to the default values, the background color of the cells changes from yellow to white.

Modifying Initial Values When the FB Is Not Nested

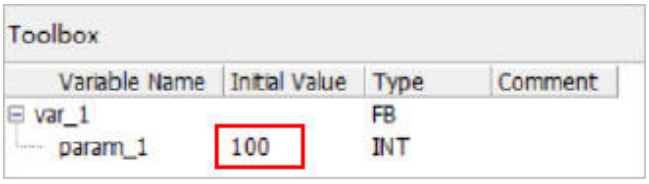
Modify the initial value of the FB type from 0 to 10. Use the default value as the initial value of the FB instance, that is, the initial value 10 of the FB type, as shown in the following figure.

NO.	I/O Type	Name	Data Type	Initial Value	Power Down Hold	Comment
1	VAR	param_1	INT	10	Non Retained	
2						

NO.	Variable ...	Data Type	Initial Value
1	var_1	FB	...
13			



Modify the initial value of the FB instance from 10 to 100. The initial value of the FB instance is 100. At this time, if you attempt to modify the initial value of the FB type to 11, the initial value of the FB instance remains unchanged (still 100), as shown in the following figure.



See the figure below. In the ladder diagram, double-click "FB" to display the FB instance. At this time, the initial value of the FB instance is displayed in the FB view instead of the initial value of the FB type. If the initial value of the variable is modified to be inconsistent with the FB, the background color will be yellow. Modifying the initial value on this interface is the same as modifying the initial value of the instance in the function block instance table.

NO.	I/O Type	Name	Data Type	Initial Value	Power Down Hold	Comment
1	VAR	param_1	INT	100	Non Retained	
2						

Modifying Initial Values When the FB Is Nested

Add a variable fb1 in the FB type, and set the data type to "FB_1". Modify the initial value of FB_1 from 1000 to 1001. The member variable fb1 of the FB type automatically takes the default value 1001 as the initial value, and the member variable fb1 of the FB instance also automatically takes the default value 1001 as the initial value, as shown in the following figure.

NO.	I/O Type	Name	Data Type	Initial Value	Power Down Hold	Comment
1	VAR	param_1	INT	100	Non Retained	
2	VAR	fb1	FB_1	...	Non Retained	
3						

Toolbox

Variable Name	Initial Value	Type	Comment
var_1		FB	
param_1	11	INT	
fb1		FB_1	
param_1	1001	INT	

NO.	Variable Name	Data Type	Initial V...	Comment	Length
1	var_1	FB	...		nBitLen:16
2					

Toolbox

Variable Name	Initial Value	Type	Comment
var_1		FB	
param_1	11	INT	
fb1		FB_1	
param_1	1001	INT	

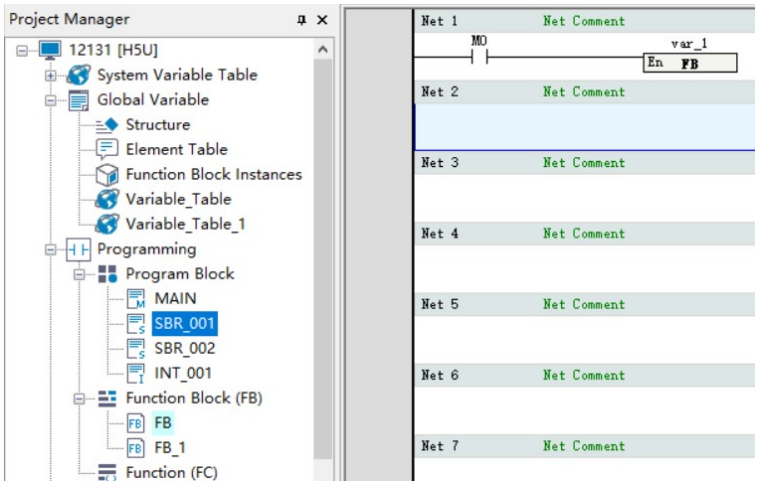
FB is the middle layer between the instance and FB_1. Modify the initial value of FB_1 to 1500 on the FB type interface. Then the initial value of the FB type changes to 1500, and the background color changes to yellow. At this time, the initial value of FB_1 of the FB instance is also 1500, but the background color is white, indicating that the default value is used, as shown in the following figure.

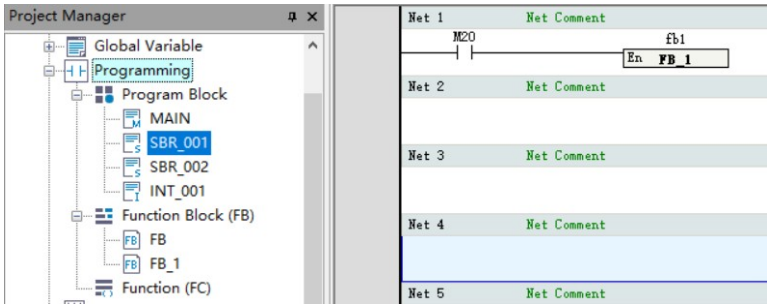
NO.	I/O Type	Name	Data Type	Initial V...	Power Dow...	Comment
1	VAR	param_1	INT	11	Non Retained	
2	VAR	fb1	FB_1	...	Non Retained	

Toolbox

Variable Name	Initial Value	Type	Comment
fb1		FB_1	
param_1	1500	INT	

- Enter the instance interface from the main program. The initial value of the FB instance is displayed. Double-click "FB_1" to enter the FB_1 instance interface, and modify the initial value to 2000. Open the FB_1 type, and the initial value is still 1001. Open the FB instance FB_1, and the initial value is 2000, as shown in the following figure.





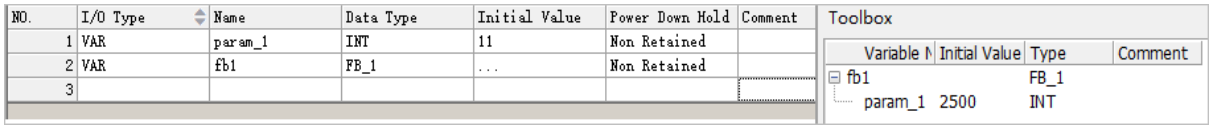
NO.	I/O Type	Name	Data Type	Initial Value	Power Down Hold	Comment
1	VAR	param_1	INT	2000	Non Retained	
2						

At this time, the tab name is "FB_1(var_1.fb1)".

- Double-click "FB" in the "Project Manager" navigation tree. You can see that the initial value of FB_1 on the FB type interface is 1500. Double-click "FB_1" in the ladder diagram of the FB type interface to enter the FB_1 instance interface. You can see that the initial value of the FB_1 instance is 1500. Modify it to 2500. Then return to the FB type interface to check the initial value of FB_1. It also changes to 2500, as shown in the following figure.

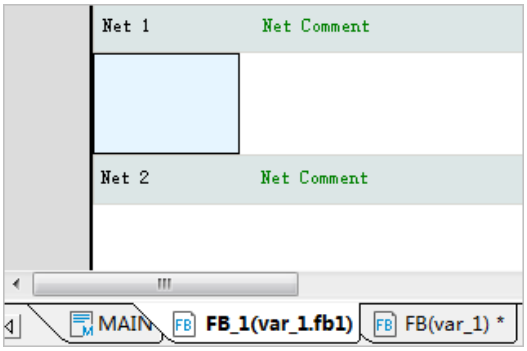
NO.	I/O Type	Name	Data Type	Initial Value	Power Down Hold	Comment
1	VAR	param_1	INT	2500	Non Retained	
2						

At this time, the tab name is "FB_1(fb1)".



Tab at the Bottom of the FB View

The tab displayed at the bottom of the FB view contains the following information from left to right: node name, instance name, and unsaved flag. The node name is the name of the project tree node, and the instance name refers to the instance name in parentheses, as shown in the following figure.



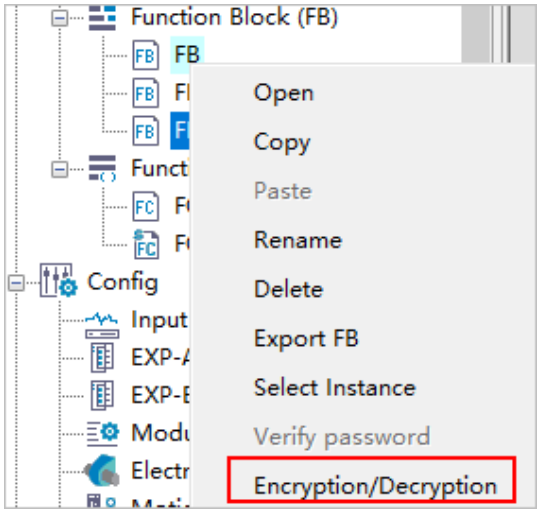
As shown in the preceding figure, "FB" is the node name, "var_1.fb1" is the instance name, and "*" indicates unsaved.

Since the tab needs to be parsed, characters including the period (.), asterisk (*), and parentheses (()) are not allowed when FBs and structures are renamed.

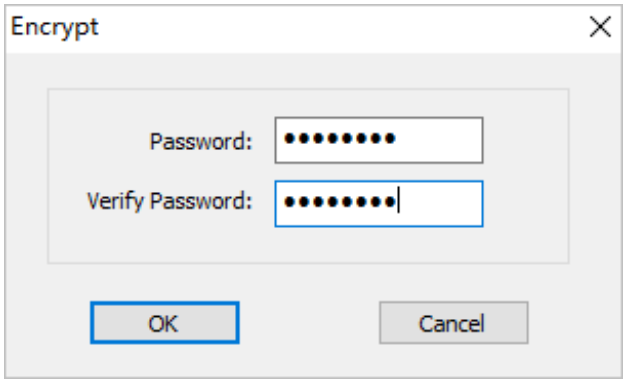
5.11.5 Encrypting the Function Block or Function

This section takes encryption of function blocks (FBs) as an example. The process is similar for encrypting functions (FC). After encryption, the method of calling the function blocks or functions remains unchanged.

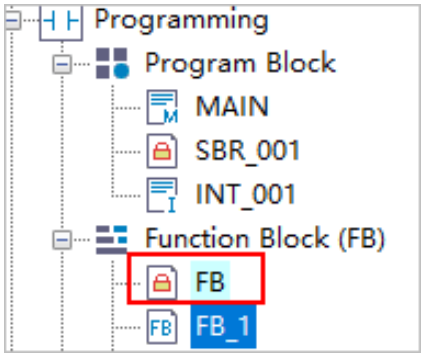
1. Choose "Programming" > "Function Block (FB)" in the "Project Manager" tree, right-click "FB", and select "Encryption/Decryption".



2. Enter and confirm the password in the displayed "Encrypt" dialog box.



The following figure shows a function block after encryption.

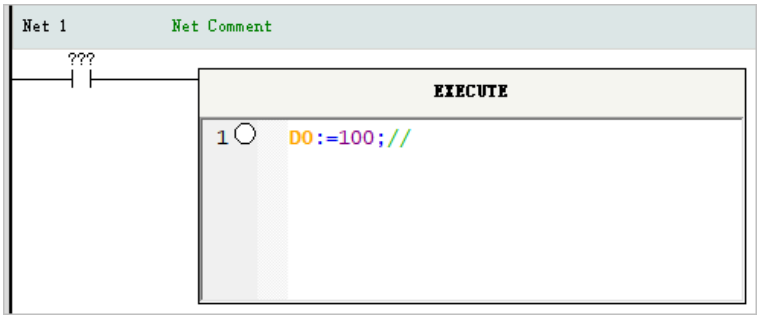


Performing the preceding steps on an encrypted function block will decrypt it and restore it to its original unencrypted state.

To access an encrypted function block, you can double-click the encrypted node, or right-click the encrypted node and choose "Verify password" from the shortcut menu, and enter the correct password in the displayed dialog box.

5.12 Execution Block

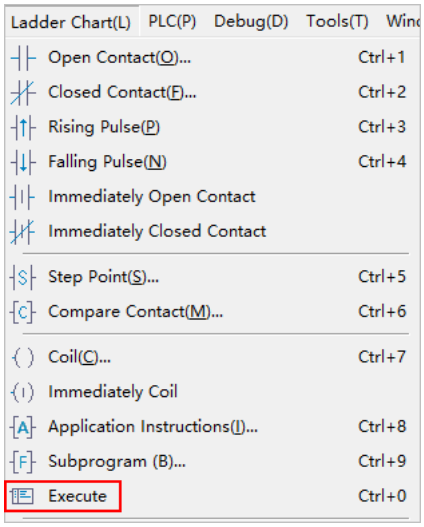
LD supports execution block nesting used to insert the ST snippet. Execution blocks can be inserted in main program, subprogram, and FB/FC.



Inserting an execution block

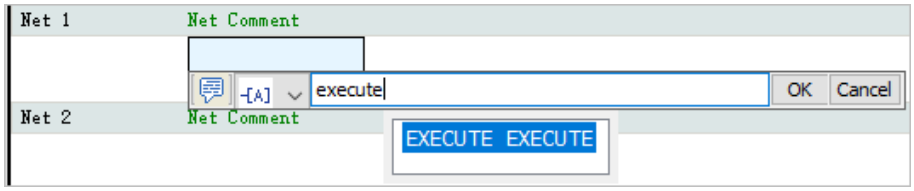
Insert an execution block in any of the following ways.

- Choose "Ladder Chart" > "Execute" in the main menu or press "Ctrl" + 0 to insert an execution block.

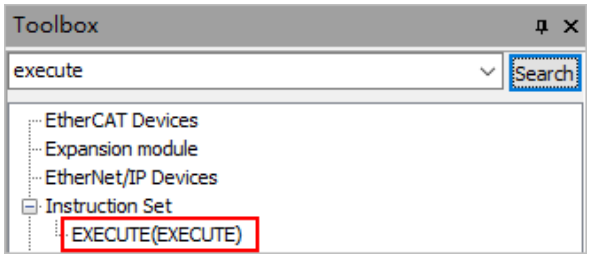


- In the toolbar, click to insert an execution block.

- Run the "EXECUTE" instruction to insert an execution block.

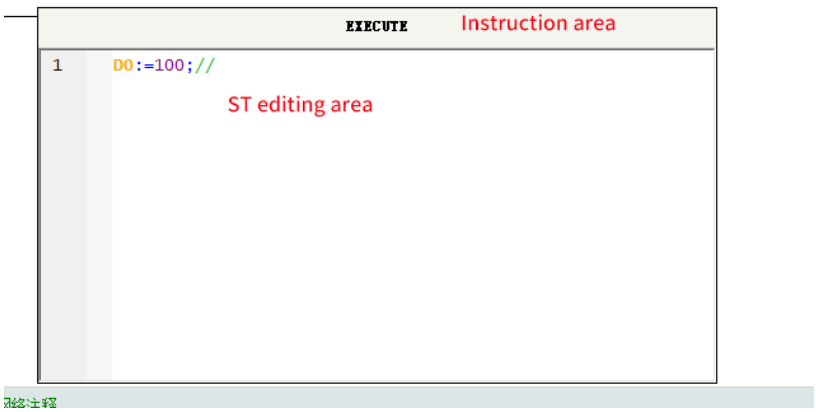


- In the tool box, double-click "EXECUTE" to insert an execution block.



Editing en execution block

You can preform Copy&Paste, Undo&Restore, Cut, and Delete operations on an execution block, as shown in the following figure.



Double-click the instruction area to modify an instruction. Click the editing area to edit the ST statement. Or, select an execution block and press Enter to edit the ST statement and press Esc to exit ST statement editing.

You can select an execution block and then drag it to zoom in the block to up to 9 rows and 12 columns and zoom out the block to at most 2 rows and 2 columns. The zooming operation is prohibited in the monitoring mode and allowed in the online modification mode.

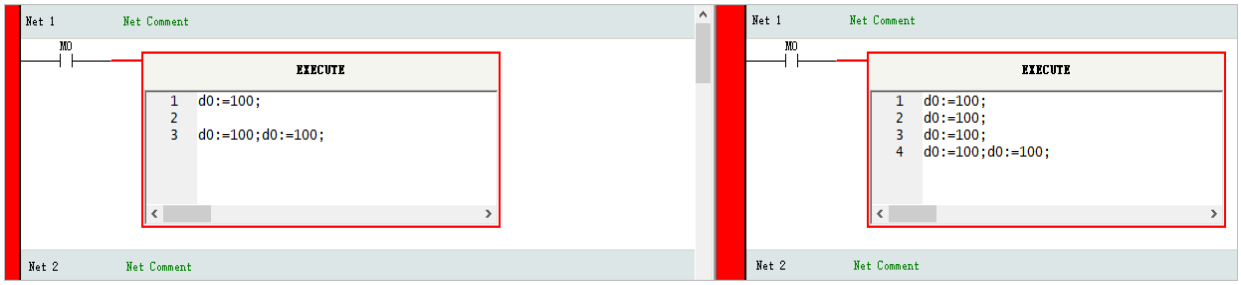
Naming an execution block

When an execution block is created, a globally unique name consisting of "_STB+number" is automatically assigned to the execution block.

- You can search for an execution block by its name.
- You can search for all execution block by the keyword "_STB".
- In the information output window, double-click an entry to jump to the corresponding execution block.

Project comparison - execution block

The project comparison function allows you to compare execution blocks separately. The differences are displayed in red. Double-click the execution block to compare the ST statements in detail inside the execution block. For more information, see 24.1 Project Comparison.



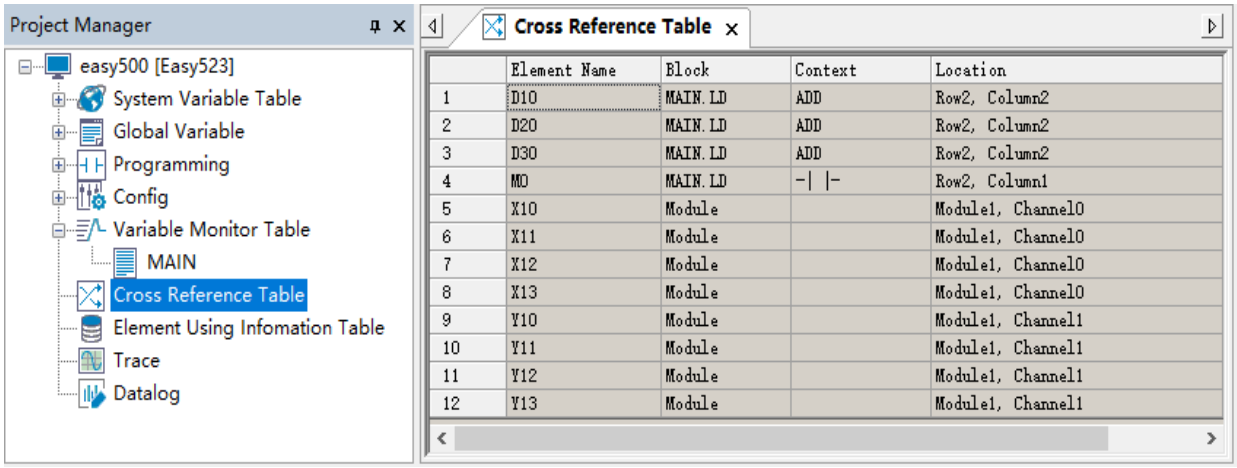
5.13 Cross-Reference

5.13.1 Cross Reference List

The cross reference list identifies all operands used in a program and identifies element names, program blocks, context of operands, as well as their location in the program block. The cross reference list clearly indicates whether and where a symbol name is used as well as the memory allocation in the program.

Using cross reference list

In the "Project Manager" tree, double-click "Cross reference table". Or, right-click "Cross reference table" and then select "Open". The "Cross reference table" page is displayed, as shown in the following figure.

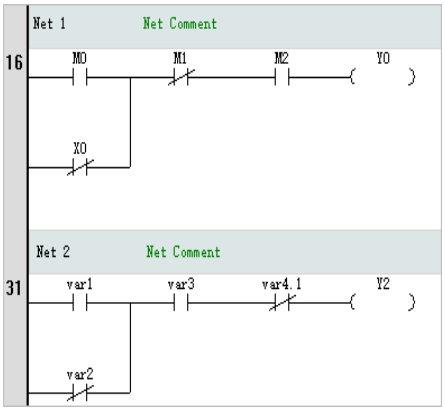
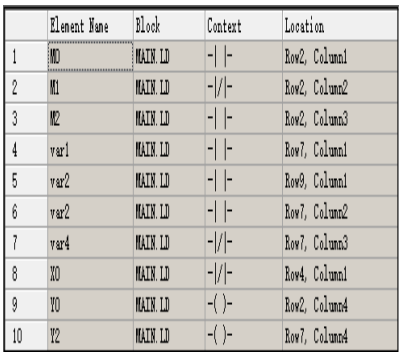
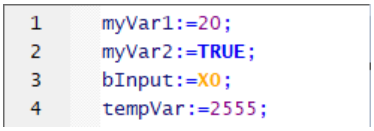
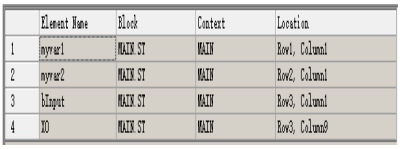


	Element Name	Block	Context	Location
1	D10	MAIN.LD	ADD	Row2, Column2
2	D20	MAIN.LD	ADD	Row2, Column2
3	D30	MAIN.LD	ADD	Row2, Column2
4	M0	MAIN.LD	-| |-	Row2, Column1
5	X10	Module		Module1, Channel0
6	X11	Module		Module1, Channel0
7	X12	Module		Module1, Channel0
8	X13	Module		Module1, Channel0
9	Y10	Module		Module1, Channel1
10	Y11	Module		Module1, Channel1
11	Y12	Module		Module1, Channel1
12	Y13	Module		Module1, Channel1

The headers of the cross reference list are described below.

- Element name: The name of the operand used in the program.
- Program block: The POU of the operand.
- Context: The program instruction using the operand.
- Location: The row and column of the operand in the program instruction.

Use example

Type	Program	Cross Reference List
LD language		
LiteST language		

When you access the cross reference list page again, the "Cross reference table" menu is added to the menu bar.

- Select a record from the cross reference list. In the menu bar, choose "Cross reference table" > "Element location". Then, the system jumps to the specific program location where the element is referenced.
- Double-click a record in the cross reference list. Then, the system jumps to the specific program location where the element is referenced.

NOTE

- For FB, FC, and INT blocks, the cross reference list will only count the references of internal variables and element of the block if the instance is called in the program.
- For SBR subprograms, the cross reference list counts the references of internal variables and element of the block even if the instance is not called.

5.13.2 Browsing Cross-References

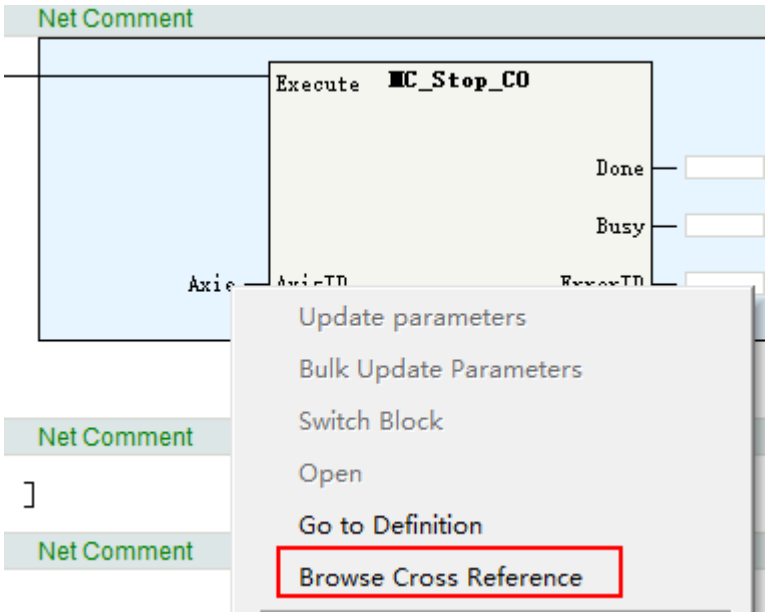
You can browse the cross-references of specified variables in ST, LD, and variable tables.

Preconditions

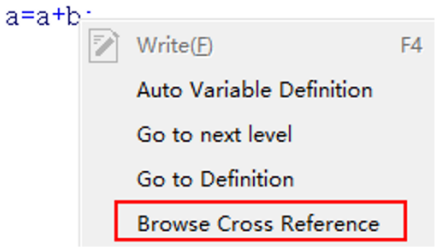
- PLC: Easy500 series, Easy300 series, and H5U series.
- Firmware version: V6.5.0.0 and later
- Software version: AutoShop V4.11.0.0 and later

Procedure

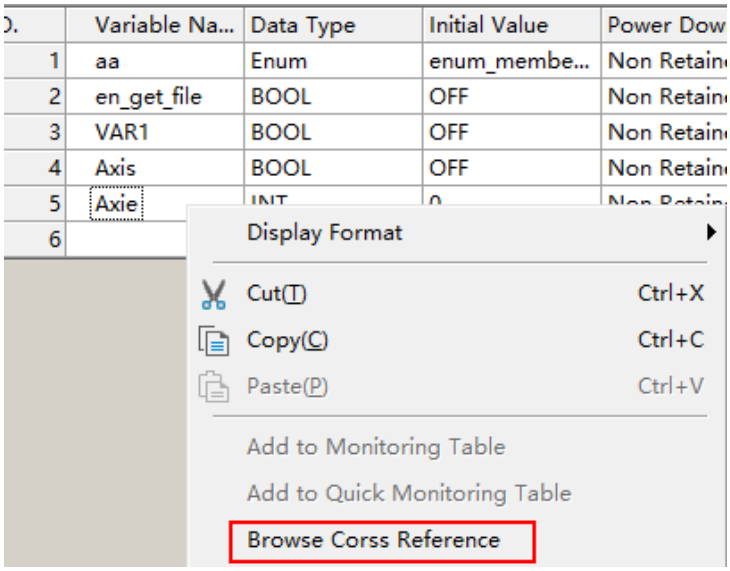
1. Right-click a cross-referenced variable in the view area of an LD program, an ST program, or a variable table, and choose "Browse Cross Reference".
   - LD program area



   - ST program area

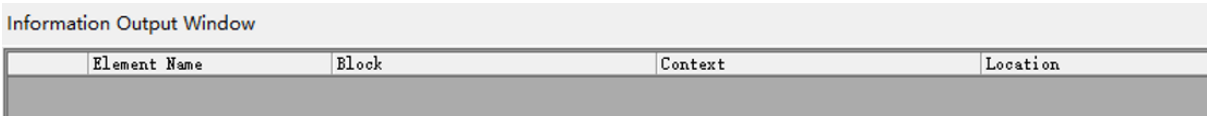


   - Variable table area



2. The "Information Output Window" at the bottom of the interface displays the cross-reference results, including the element name, referenced program block, context, and location. Double-click the row

where the element is located to jump to the specific program location where the element is referenced.



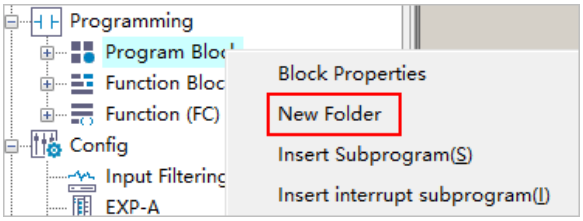
5.14 Folder

Introduction

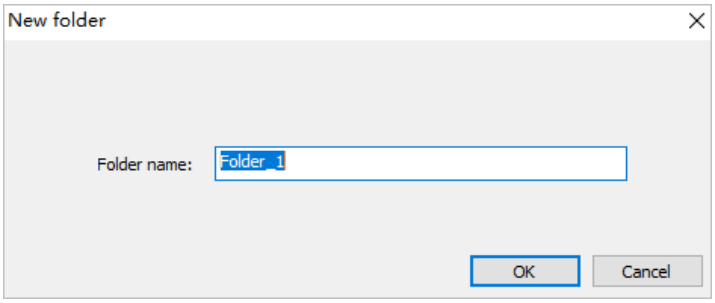
You can use folders to classify and batch operate program blocks, function blocks, and functions.

Creating a folder

1. Right-click "Program Block", "Function Block", or "Function", and select "New folder". For example, right-click "Program Block", as shown in the following figure.



2. In the "New folder" dialog box displayed, set "Folder name" and click "OK". A folder is created.

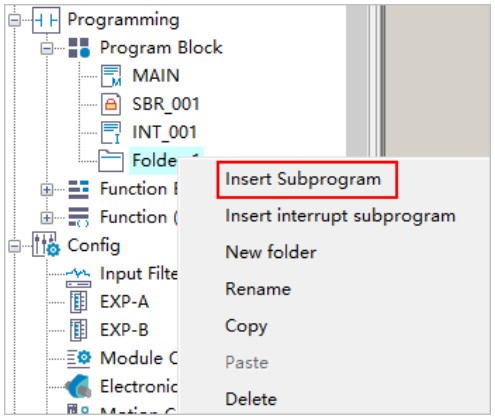


NOTE

Nested folders can contain up to four levels. Each folder name must not:

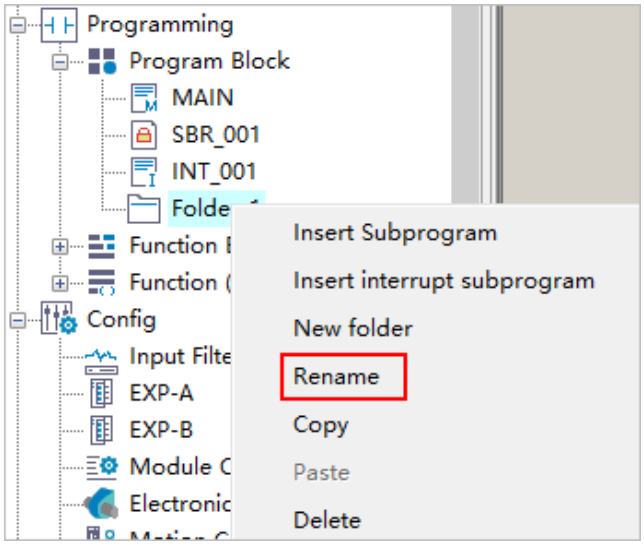
- Be empty.
- Contain special characters such as space, asterisk (*), pipe (|), backslash (\), less-than sign (<), comma (,), period (.), forward slash (/), left parenthesis ((), right parenthesis ()), and question mark (?), or start with underscore (_), numbers, SYS, or _SYS_.
- Be the same as the name of any soft element forms, standard data types, instructions, or constants.
- Be any keywords such as ARRAY, TRUE, FALSE, ON, OFF, and NULL.
- Be the same as the name of any files or folders at the same level.

3. Right-click the new folder and select "Insert Subprogram" to add the new subprogram file.
   You can drag program files from other folders to a specified folder.

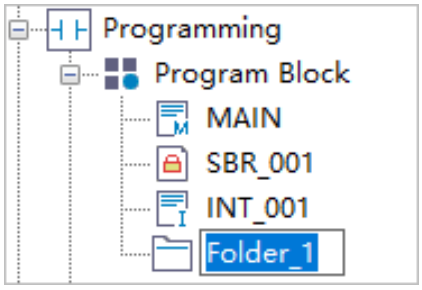


Renaming a folder

1. Right-click a folder to be renamed and select "Rename", as shown in the following figure.

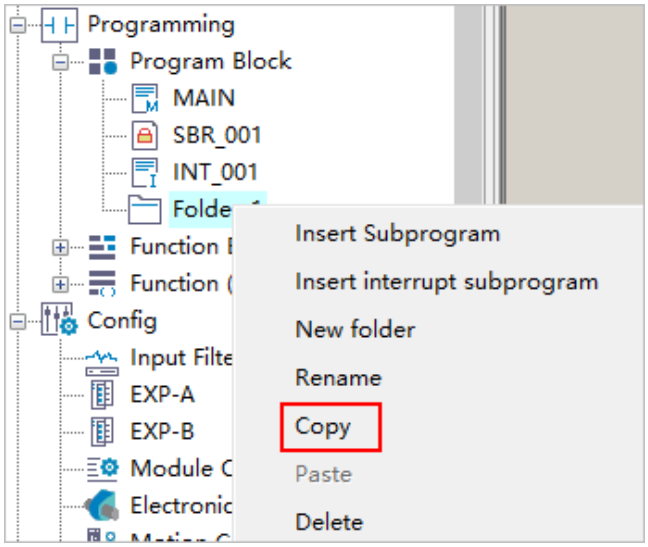


2. Enter the new folder name in the folder name text box and press "Enter", as shown in the following figure.



Copying and pasting a folder

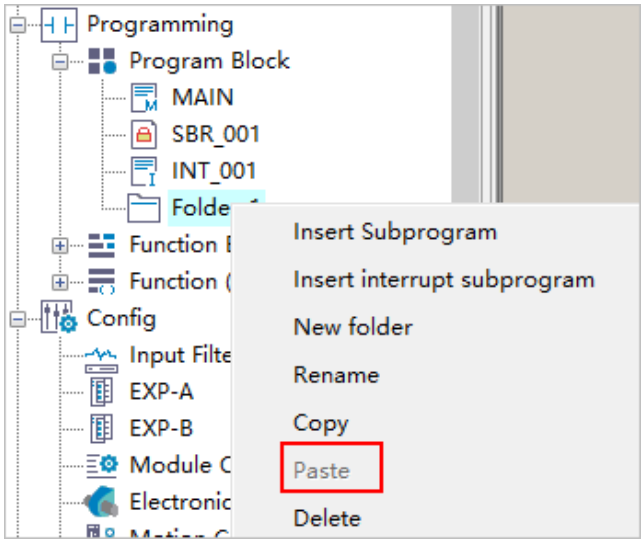
1. Right-click a folder to be copied and select "Copy" to copy the folder and the sub folders and files in the folder to the clipboard, as shown in the following figure.



CAUTION

Folders containing an encrypted program cannot be copied.

2. When the clipboard contains valid content, right-click the target node, and select "Paste" to paste the content in the clipboard to the target node, as shown in the following figure.

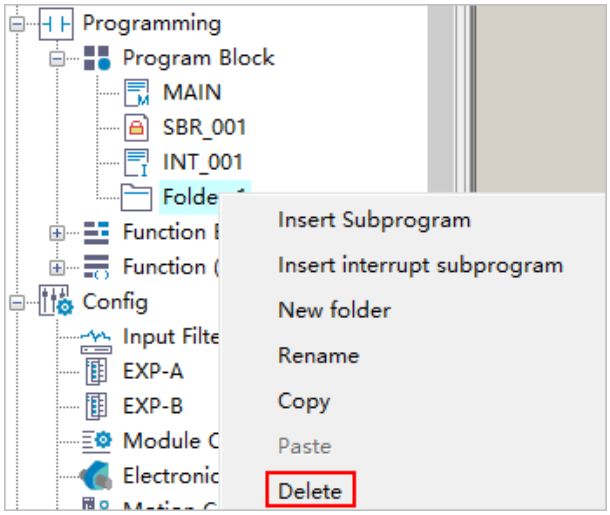


CAUTION

- Pasting fails when the target node contains a file or folder with the same name as the folder in the clipboard.
- Pasting fails if the target node and the folder in the clipboard will produce nested folders of more than four levels.

Deleting a folder

1. Right-click a folder to be deleted and select "Delete", as shown in the following figure.



CAUTION

Folders containing an encrypted program cannot be deleted.

2. In the dialog box displayed, click "OK" to delete the selected folder and sub folders and files in the folder.

5.15 Element Usage Table

You may need to add, delete, or edit soft elements during program commissioning. The element usage table clearly shows usage of soft elements in the program, avoiding repeated application of soft elements. You can refer to the element usage table while modifying a program to reduce program errors and improve the commissioning efficiency.

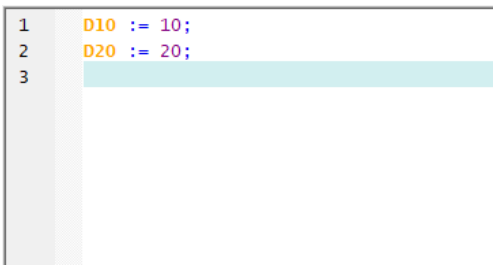
You can view the element usage table after compiling a program. The viewing rules of different program blocks are described below.

- Soft elements used in MAIN and SBR programs can be directly viewed after compilation.
- Soft elements used in INT programs can be viewed after an interrupt event is set and interrupt is enabled.
- Soft elements used in FB/FC programs can be viewed after being called.
- AutoShop is connected with the PLC.

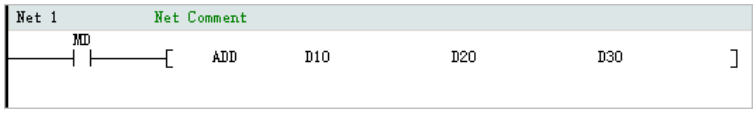
Procedure

1. Compile the program.
   The following takes the ListST program and LD program as an example to describe how to operate the element usage table.

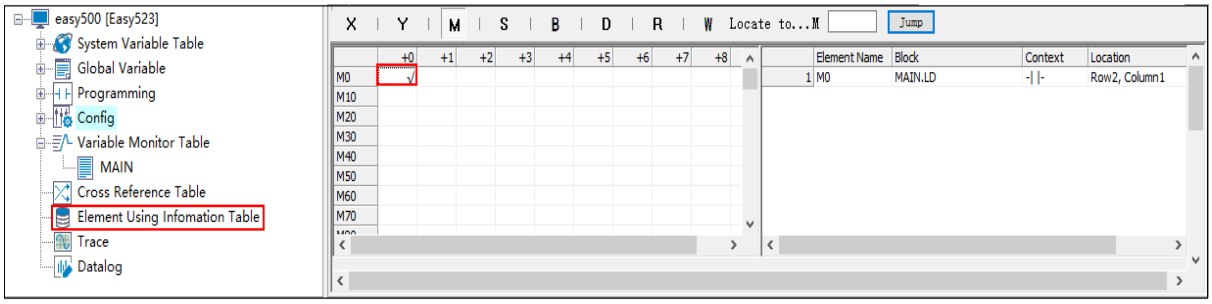
   The following figure shows a ListST program.



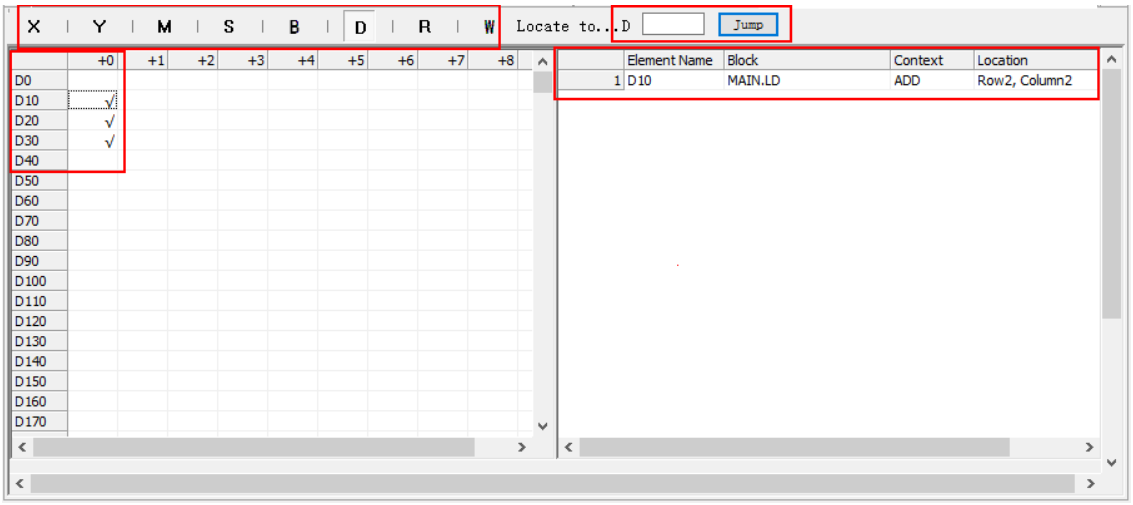
   The following figure shows an LD program.



2. In the "Project Manager" tree, double-click "Element usage table". Or, right-click "Element usage table" and then select "Open". The "Element usage table" page is displayed, as shown in the following figure.



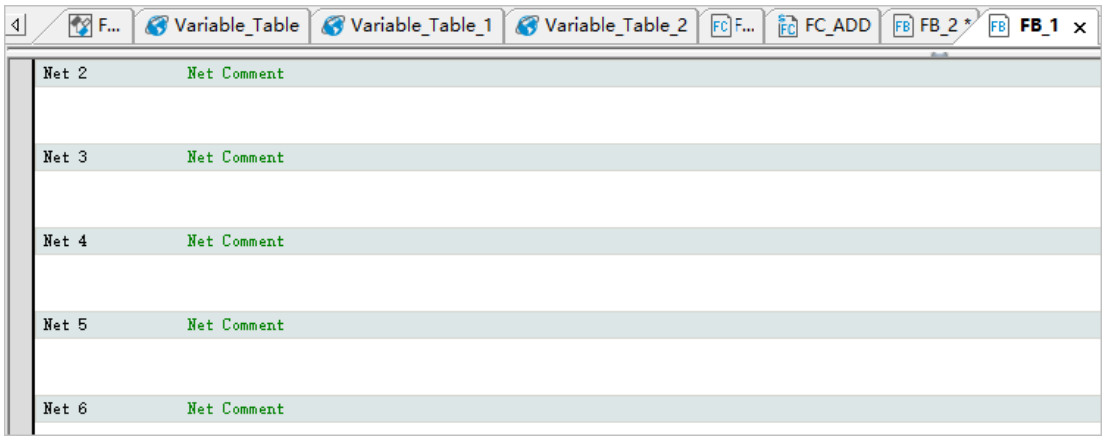
3. View the usage of soft elements, as shown in the following figure.



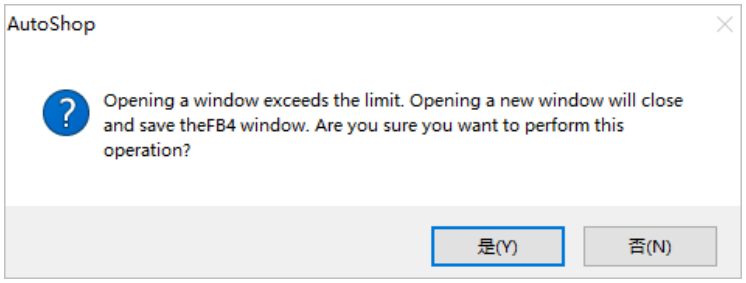
No.	Operation Description
①	To view different types of soft elements, click the corresponding tab to switch to the target type.
②	Input the name of the soft element, and then click "Jump to". The corresponding soft element is displayed.
③	A "✓" is displayed in the cell of the used soft element. Click this cell. Then, the usage details of this soft element are displayed on the right, as shown at ④ in the figure. **NOTE** When a soft element is used only in the ZRST (batch reset) or ZSET (batch set) instruction, a "*" is displayed in the cell of this element. A "✓" is displayed in the cell of elements in all other cases.
④	Double-click a row. Then, the application place of this soft element in the program is automatically displayed.

5.16 Number of Opened Windows

1. The window tabs are displayed at the window top, you can click "X" or right-click a window tab and then select "Close" to close the tab.



2. A maximum number of 32 windows can be opened. If you want to open a new window, the system prompts you to save and close the first opened window, and click "Yes" before the new window can be opened.



CAUTION

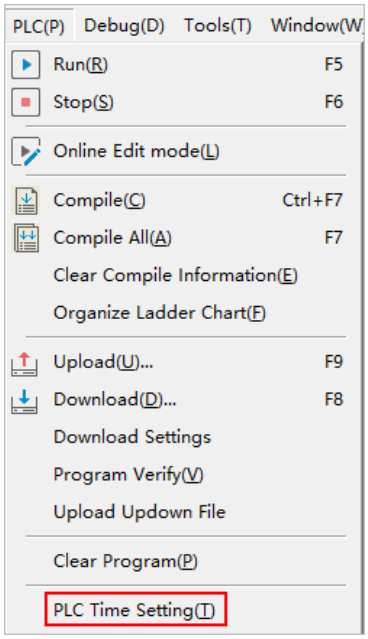
If the first opened window is a MAIN function window, you need to close the window next to the MAIN window before a new window can be opened.

5.17 PLC Time Setting

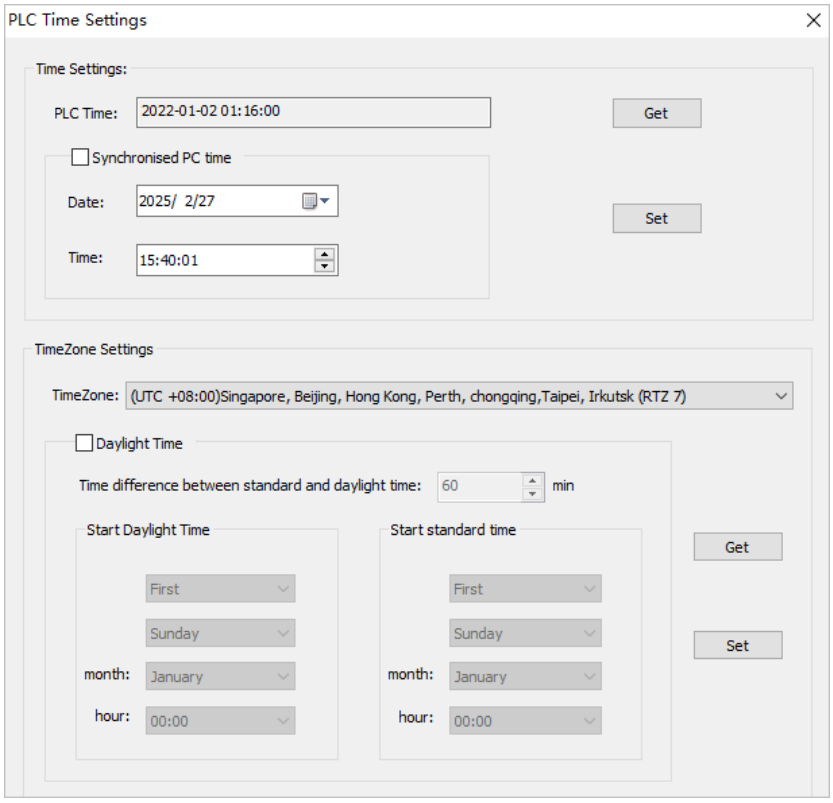
The PLC time setting function allows you to set or read the PLC time, including the time zone, on AutoShop. The time information can also be obtained through the system variable (_DataTime), and time zone information can also be read and set through LD instructions (TZGET and TZSET). See "TZSET" and "TZGET" of clock instructions in H5U and Easy Series Programmable Logic Controller Instructions Guide for details.

- Applicable models
  - Easy series: Easy523, Easy522, Easy521, Easy502, Easy501, Easy320, Easy302, and Easy301
  - H5U series: H5U, H5U-A8, H5U-A8S, and H5U-A16
- Firmware version: V6.4.0.0 and later
- Software version: AutoShop V4.10.2.0 and later
- AutoShop is connected with the PLC.

1. In the menu bar, choose "PLC (P)" > "PLC Time Setting (T)", as shown in the following figure.



2. The "PLC Time Settings" dialog box is displayed.



Time Settings

Name	Description
Read	Read the current PLC time and display the time in the "PLC Time" field.
Set	Set the PLC based on the selected date and time. When "Synchronized PC time" is selected, the date and time controls cannot be edited and the PLC time is automatically synchronized with the current PC time.

Time Zone Settings

- TimeZone: Select a time zone from the drop-down list. Daylight saving time can only be customized by users, and it is not recommended automatically when time zone is selected. The time zone options are shown in the following table.

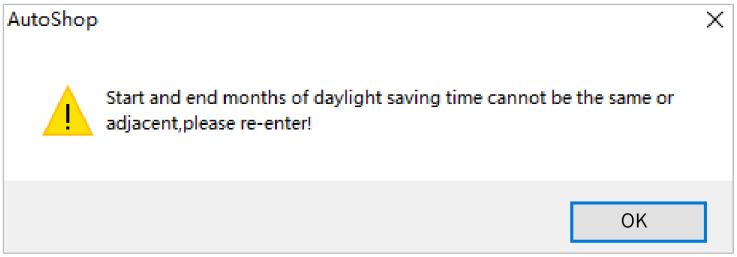
UTC Time	Country/Region
UTC-12:00	Eniwetok Island, Kwajalein Island
UTC-11:00	Midway Island
UTC-10:00	Hawaii
UTC-09:00	Alaska
UTC-08:00	Pacific Standard Time (USA, Canada, and Mexico)
UTC-07:00	Arizona, La Paz, Mazatlan, Mountain Standard Time (USA and Canada)
UTC-06:00	Central Standard Time (USA and Canada), Mexico City, Central America
UTC-05:00	Bogota, Rio Blanc, Eastern Standard Time (USA and Canada), Indiana
UTC-04:00	Caracas, Asunción, Atlantic Standard Time (Canada), San Diego, Georgetown
UTC-03:30	Newfoundland
UTC-03:00	Brasilia, Buenos Aires, Greenland, El Salvador, Montevideo, and Cayenne
UTC-02:00	Coordinated Universal Time-02, Mid-Atlantic
UTC-01:00	Azores, Cape Verde
UTC	Casablanca, Dublin, Edinburgh, London, Coordinated Universal Time, and Monrovia
UTC+01:00	Amsterdam, Berlin, Rome, Vienna, Prague, Budapest, and West and Central Africa
UTC+02:00	Athens, Beirut, Damascus, Kiev, Jerusalem, Cairo, Eastern Europe, and Sofia
UTC+03:00	Baghdad, Moscow, St. Petersburg, Volgograd (RTZ2), Kuwait, and Tehran
UTC+04:00	Abu Dhabi, Baku, Yerevan, Port Louis, and Tbilisi
UTC+04:30	Kabul
UTC+05:00	Ashgabat, Tashkent, Islamabad, Karachi, and Yekaterinburg (RTZ 4)
UTC+05:30	Mumbai, Kolkata, New Delhi, Koti, and Chennai
UTC+05:45	Katmandu
UTC+06:00	Astana, Dakar, and Novosibirsk (RTZ 5)
UTC+06:30	Yangon
UTC+07:00	Bangkok, Hanoi, Jakarta, and Krasnoyarsk (RTZ 6)
UTC+08:00	Singapore, Beijing, Hong Kong, Urumqi, Chongqing, Taipei, and Irkutsk (RTZ 7)
UTC+09:00	Yakutsk (RTZ 8), Osaka, Tokyo, and Seoul
UTC+09:30	Adelaide, Darwin
UTC+10:00	Brisbane, Canberra, Melbourne, Sydney, Hobart, and Vladivostok (RTZ 9)
UTC+11:00	Solomon Islands, Magadan, and New Caledonia

UTC Time	Country/Region
UTC+12:00	Auckland, Wellington, and Fiji, Coordinated Universal Time+12
UTC+13:00	Nuku'alofa, Samoa
UTC+14:00	Christmas Island

- Daylight Time: This option is not enabled by default. In this state, time difference, daylight saving time, and standard time cannot be configured.
  - Time difference between standard and daylight time: The value range is 0 min to 120 min. An error is reported if the value exceeds this range. You can manually edit the value or use the number knob control to adjust the value.
  - Daylight saving time or standard time setting: Select an option from the week, weekday, month, and hour drop-down lists respectively. The 24-hour system is used.

CAUTION

The months of "Start Daylight Time" and "End Daylight Time" cannot be the same or adjacent; otherwise, the "The start and end months of daylight saving time cannot be the same or adjacent. Please enter again!" dialog box is displayed, as shown in the following figure.



- Set: Set the time zone of the PLC according to the time zone of the current control, whether daylight saving time is enabled, time difference, daylight saving time and standard time.
- Read: Read the PLC time zone and display it in the corresponding control.

5.18 I/O Refresh

5.18.1 Introduction

I/O refresh, that is, the timing of software and hardware update of inputs and outputs controlled by PLC, can be divided into three types: PLC I/O refresh, module I/O refresh, and remote I/O refresh.

H5U/Easy series PLCs support two I/O refresh modes: batch refresh and immediate refresh.

- Batch refresh: batch access to inputs and outputs at specified time.
- Immediate refresh: the given input and output are refreshed immediately at the instruction execution point.

NOTE

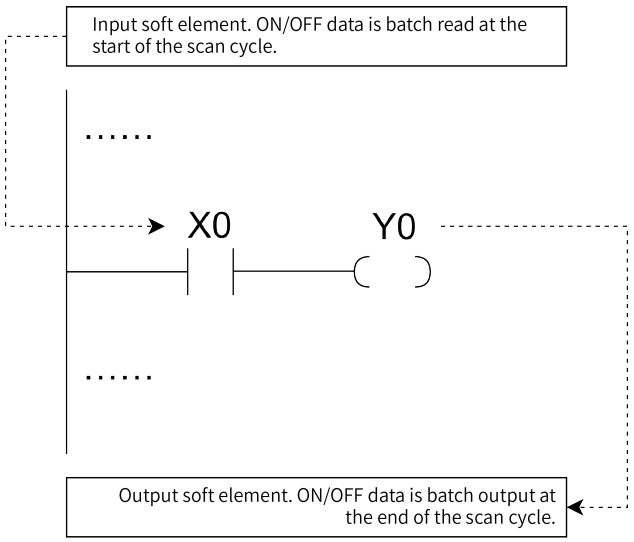
Versions support immediate input/output: AutoShop V4.10.1.0 or later, and PLC firmware V6.3.0.0 or later.

The following table lists the corresponding relationship among tasks I/O batch refresh and immediate refresh on the PLC, on local extension module, and on extension module of a coupler (remote I/O) and task visibility.

Name	Batch Refresh	Immediate Refresh	Task	Task Visibility
PLC I/O	Supported	Supported	Main task	Visible to users
Module I/O	Supported	Not supported	Module task	Invisible to users
Remote I/O	Supported	Not supported	EtherCAT task	Visible to users

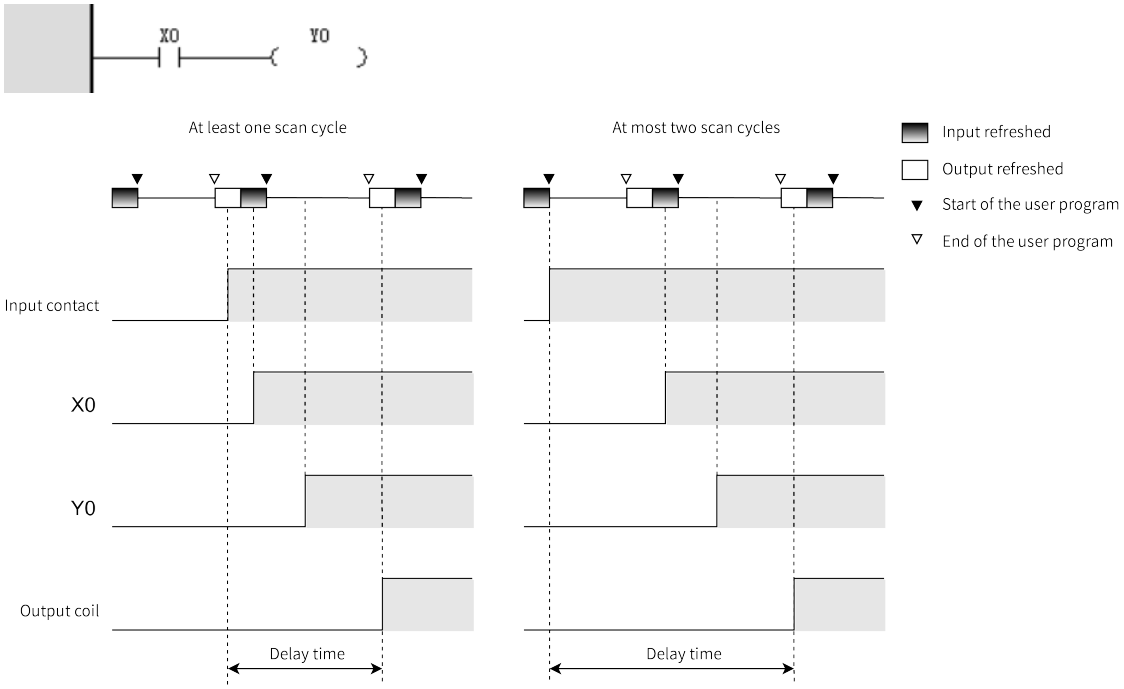
5.18.2 Local I/O Refresh

Batch I/O refresh on PLC

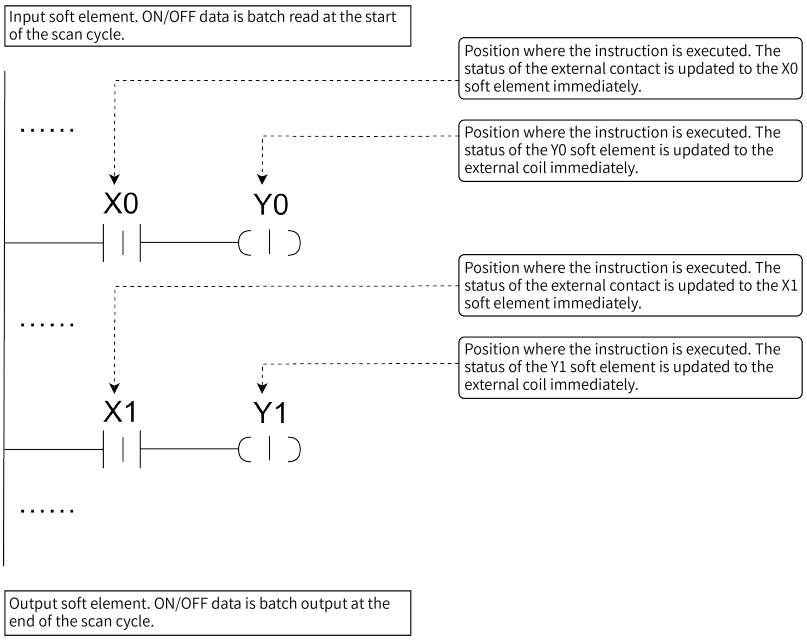


Response delay

The coil state, which changes in response to the input contact state, is delayed by up to two scanning cycles depending on the turning ON timing of the external contact. Take the ladder diagram where output Y0 is ON when input X0 is ON as an example.

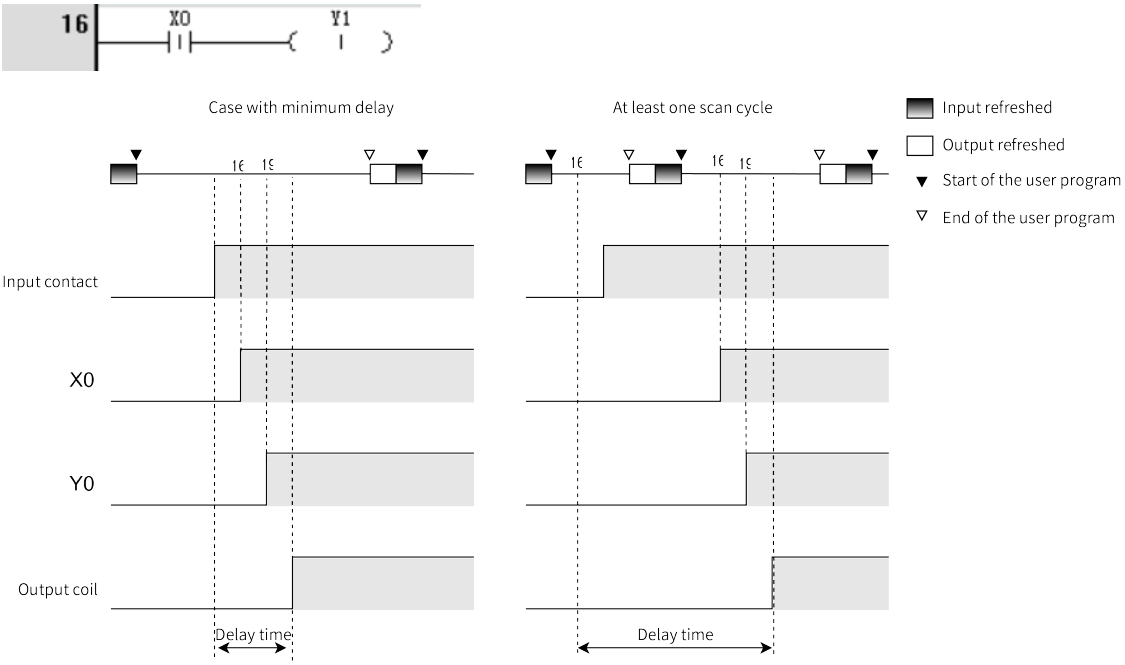


Immediate I/O refresh on PLC



Response delay

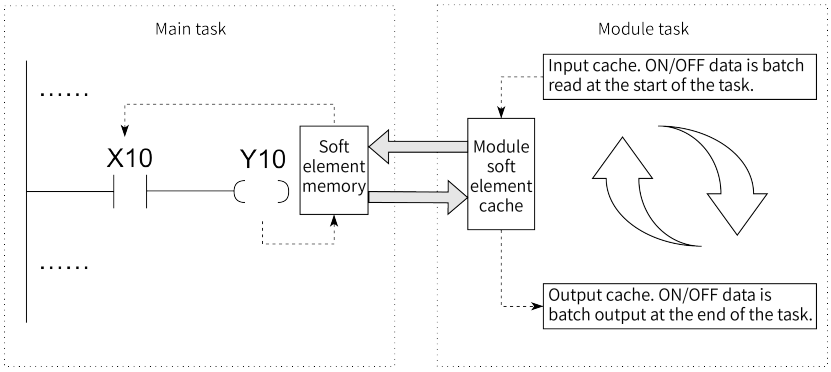
The coil state, which changes in response to the input contact state, is delayed by up to one scanning cycle depending on the turning ON timing of the external contact. The following figure takes the ladder diagram where output Y0 is ON when input X0 is ON as an example.



5.18.3 Module I/O Refresh

Introduction

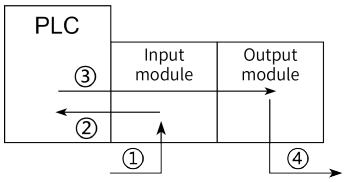
Batch module I/O refresh is supported. Immediate module I/O refresh is not supported. PLC I/O is refreshed in the main task, while module I/O is refreshed in a separate module task invisible to users, and batch input reading is performed at the task start and batch output writing is performed at the task end. The following figure shows the working principle of module I/O refresh.



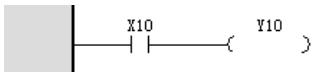
Module Task Cycle	Easy301	Easy302	Easy320	Easy501	Easy502	Easy521	Easy522	Easy523
1 ms	-	-	-	-	-	√	√	√
2 ms	√	√	√	√	√	-	-	-

Response delay

The following figure shows the physical configuration.



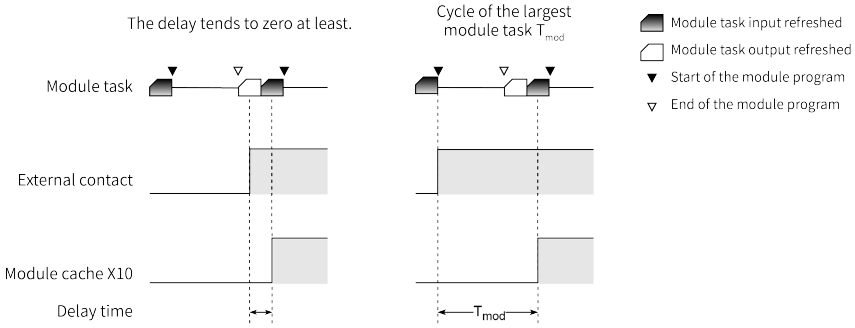
The following takes input X10 being ON and output Y10 being ON as an example to describe the response delay of module I/O.



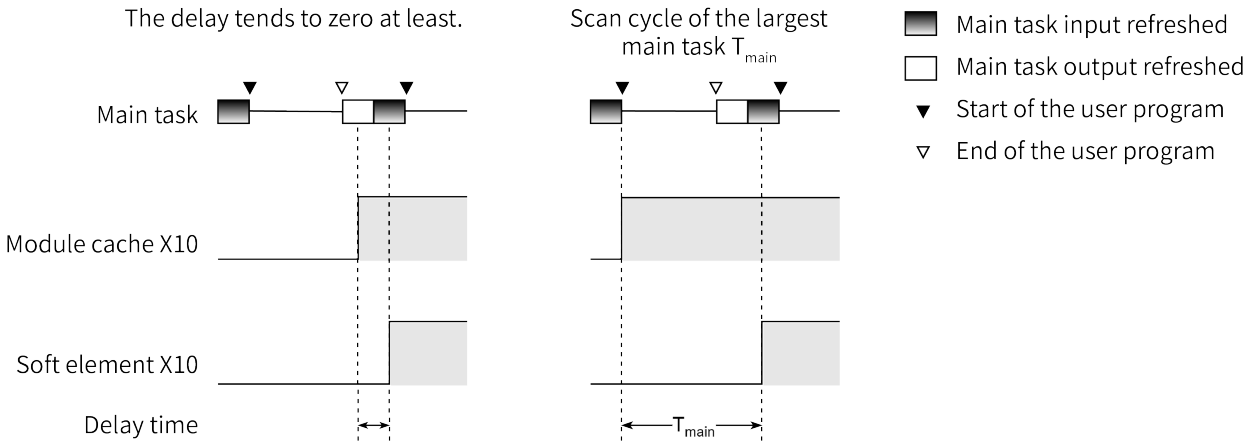
Input delay analysis

The input consists of ① and ②.

- Process ① indicates the signal state from the external contact to the module task, and the delay is shown in the following figure and the cycle of the maximum module task is Tmodule (in the following, Tmodule is shorted to Tmod).



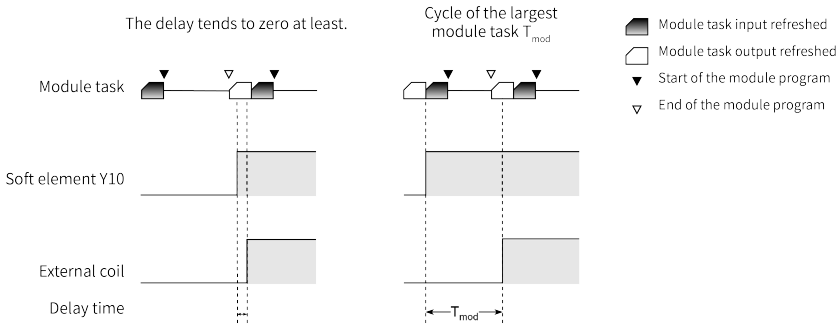
- Process ② indicates the signal state from the module task to the main task. The delay is shown in the following figure, and the maximum scan cycle of the main task is Tmain.



Maximum input delay time TiMax = Tmod + Tmain

Output delay analysis

The output consists of ③ and ④. The delay is shown in the following figure, and the cycle of the maximum module task is Tmod.



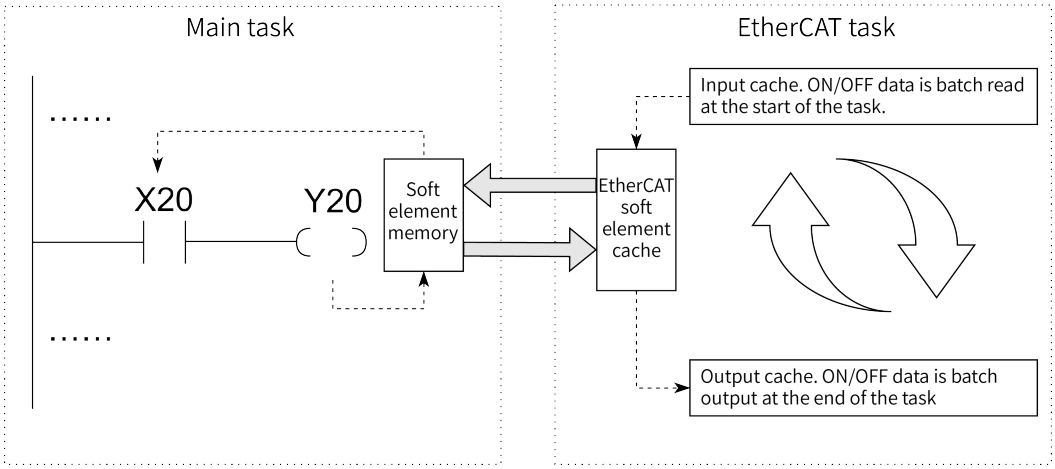
Maximum output delay time ToMax = Tmod.

In general, when the contact of the module X10 hardware is ON, the minimum delay time of the module Y10 hardware → 0 when the coil is ON, and the maximum delay time is Tmax_mod = TiMax + ToMax.

5.18.4 Remote I/O Refresh

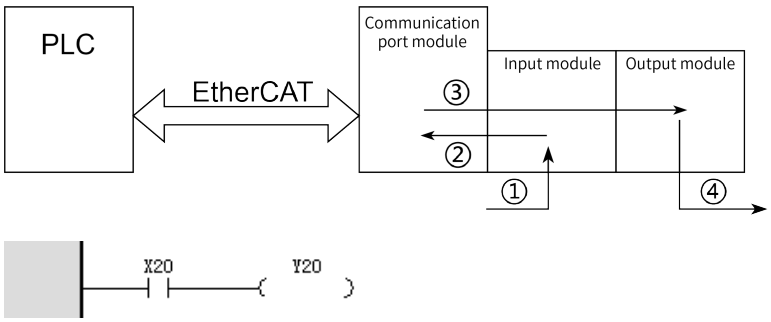
Introduction

Batch remote I/O refresh is supported. Immediate remote I/O refresh is not supported. Module I/O is refreshed in a separate task. Similarly, remote I/O is refreshed in a separate "EtherCAT task", batch input reading is performed at the task start, and batch output writing is performed at the task end. The following figure shows the working principle of remote I/O refresh.



Response delay

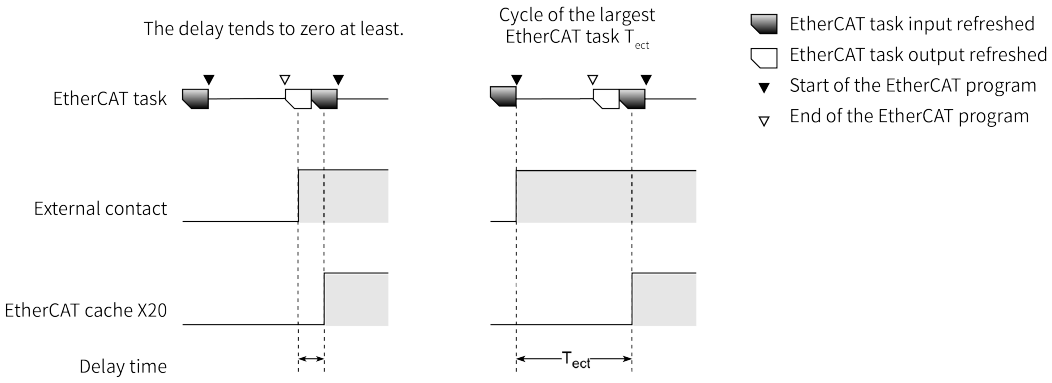
The following figure shows the physical configuration diagram of a remote I/O module mounted on the PLC.



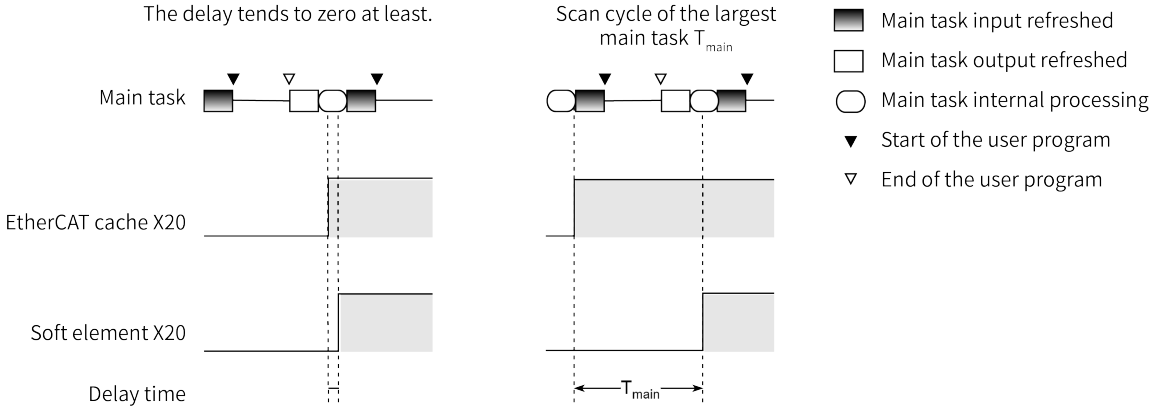
The input delay of the remote I/O consists of the module I/O delay generated by ①②③④, which has been described in 5.18.3 Module I/O Refresh. In the following analysis, the module I/O delay time will be ignored; the input of the EtherCAT task is the actual input and its output is the actual output. The module I/O delay time will be considered for the final maximum delay time.

Input delay analysis

The EtherCAT task reads the input from the external contact, as shown in the following figure.

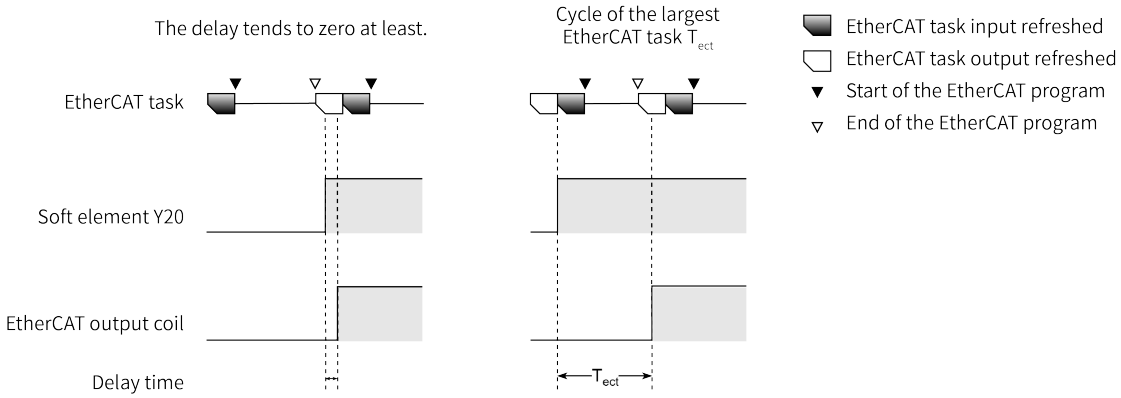


The main task reads the input from the EtherCAT task, as shown in the following figure.



Maximum input delay time TiMax_ect = Tect + Tmain

Output delay analysis



Maximum output delay time ToMax_ect = Tect.

In general, considering the total delay time of module I/O, when the contact of the remote I/O X20 hardware is ON, the minimum delay time of the Y20 hardware → 0 when the coil is ON, and the maximum delay time is Tmax_ect = TiMax_ect + ToMax_ect + Tmax_mod.

6 Programming Language

6.1 Structured Text (LiteST)

6.1.1 Overview

LiteST is a high-level text-based programming language for automation systems. Its syntax structure is similar to that of PASCAL. It provides a simple standard structure to make programming fast and efficient. LiteST uses many traditional characteristics of high-level languages, including variables, operators, and control statements. LiteST provides a freer text-based programming mode than IL because extra placeholders are added to ensure a hierarchical structure of the program frame for easy reading and understanding. LiteST also provides easier migration and repeatability than graphical programming modes such as LD.

Example:

```
IF A>0 THEN
    X:=10;
ELSE
    X:=0;
END_IF;
```

NOTE

- The LiteST programming function requires a firmware version of V5.14.0.0 or later for the H5U series, or a firmware version of V5.67.0.0 or later for the Easy series, and an AutoShop software version of V4.8.1.0 or later.
- One LiteST program can contain up to 1,000 text rows.

6.1.2 Expressions

Block diagrams of different functions are basic elements in the LD programming environment. Similar to LD, expressions are basic elements for LiteST. An expression consists of operators and operands. An operand can be a constant, a variable, a function call, or other expressions. Example:

- Constant, such as 20, 1.43, and 16#10
- Variable, such as iVar and D0:E
- Function call, whose value is the return value of a call, such as Fun1 (1, 2, 4)
- Other expressions: such as 10+3, var1 OR var2, (x+y)/z, and iVar1:=iVar2+22

In an expression, operands are evaluated using operators in sequence defined by a particular operator priority. Operators with top priority must be first used for evaluation. Other operators with lower priority are used by priority in descending order. Operators with the same priority must be used in order from left to right in the expression.

For example, if A, B, C, and D are INT variables and are set to 1, 2, 3, and 4, respectively, A+B-C*ABS(D) must be -9 and (A+B-C)*ABS(D) must be 0.

When an operator has two operands, the leftmost operand must be evaluated first. For example, in SIN(X)*COS(Y), SIN (X) must be evaluated first, then COS(Y), and finally the product of the overall expression.

The following table lists the LiteST operators.

Operation type	Sign	Example	Priority
Parentheses	(Expression)	(A+B/C), (A+B)/C, A/(B+C)	9 (highest)
Function call	Function name (separated by commas (,) in the parameter list)	LN(A), MAX(X,Y)	8
Opposite	-	-A	7
Unary plus (+)	+	+B	7
Negate	NOT	NOT C	7
Multiply	x	A x B	6
Divide	/	A/B	6
Modulo	MOD	A MOD B	6
Plus	+	A+B	5
Minus	-	A-B	5
Compare	<, >, <=, >=	A<B	4
Equal	=	A=B	4
Not equal	<>	A<>B	4
AND	AND	A AND B	3
XOR	XOR	A XOR B	2
OR	OR	A OR B	1 (lowest)

The types supported by LiteST operators are described in the following table.

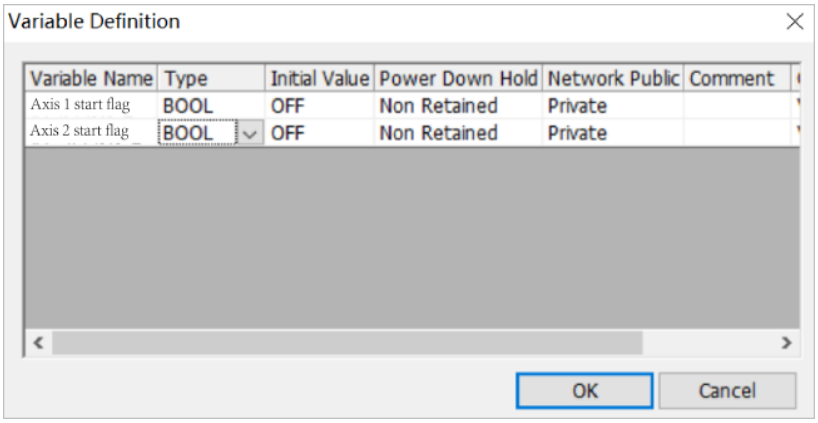
Operators	Bit			Word			Pointer	Constant						Others
	X, Y, M, S, B	Bits of word element	Custom bit variable	D, R, W	Custom word variable	Custom string variable	Pointer variable	BYTE	INT	DINT	REAL	BOOL	STRING	
	-	-	-	√	√		-	-	√	√	√	-	-	-

Operators	Bit			Word			Pointer	Constant						Others
	X, Y, M, S, B	Bits of word element	Custom bit variable	D, R, W	Custom word variable	Custom string variable	Pointer variable	BYTE	INT	DINT	REAL	BOOL	STRING	
Opposite														
Unary plus (+)	-	-	-	√	√	-	-	√	√	√	√	-	-	-
Negate	√	√	√	√	√	-	-	√	√	√	√	✓	-	-
Multiply	-	-	-	√	√	-	-	√	√	√	√	-	-	-
Divide	-	-	-	√	√	-	-	√	√	√	√	-	-	-
Modulo	-	-	-	√	√	-	-	√	√	√	√	-	-	-
Plus	-	-	-	√	√	-	-	√	√	√	√	-	-	-
Minus	-	-	-	√	√	-	-	√	√	√	√	-	-	-
Compare	√	√	√	√	√	√	-	√	√	√	√	√	√	-
Equal	√	√	√	√	√	√	-	√	√	√	√	√	√	-
Not equal	√	√	√	√	√	√	-	√	√	√	√	√	√	-
AND	√	√	√	√	√	-	-	√	√	√	√	√	-	-
XOR	√	√	√	√	√	-	-	√	√	√	√	√	-	-

Operators	Bit			Word			Pointer	Constant						Others
	X, Y, M, S, B	Bits of word element	Custom bit variable	D, R, W	Custom word variable	Custom string variable	Pointer variable	BYTE	INT	DINT	REAL	BOOL	STRING	
OR	✓	✓	√	√	√	-	-	√	✓	✓	✓	√	-	-

6.1.3 Variables

You can compile variables during LiteST program editing and press Enter or click in the area outside the program basic block to display the variable definition box. The default variable type is INT. During a function call or an instruction call, the data type can be automatically identified.



In LiteST programs, characters in orange indicate soft elements, such as X, Y, M, S, B, D, R, or W, and characters in blue indicate an instruction or keyword, such as IF, FOR, AND, DINT, as shown in the following figure.

```
//REAL TYPE
D0:E:=f_Test1;
//DINT TYPE
D2:D:=REAL_TO_DINT(f_Test1);
```

6.1.4 Constants

Constants can be expressed in three ways:

1. A constant is a decimal number by default, for example, a:=100.
2. A constant can contain underlines (_), for example, a:=10#100_10, a:=16#FF_AE_12, and a:=2#1100_1111_11_10, as shown in the following figure.

```
i_Test5 42 :=2#10_10_10;
////Division-by-zero protection: If the divisor is 0, it automatically changes to 1.
//i_Test2:=0;
//f_Test1:=f_Test1/i_Test2;
//
```

3. LiteST also allows LD expression as constants. That is, K100 indicates constant 100, H indicates a 16-bit number, and E indicates a floating-point number.

6.1.5 Function Blocks, Functions, Subprograms, and Interrupts

- FB: In terms of input parameters, only axis parameters are variables requiring pin input and other pins do not need to be input. In addition, output pins do not need to be input. Actual input arguments can be expressions, but actual output arguments cannot.
- FC: Input parameter pins must be input. Otherwise, a compilation error is reported. Output parameters can be left empty. Actual input arguments can be expressions, but actual output arguments cannot.
- Subprogram: A subprogram is called in non-parametric function format, for example, SBR_001().

CAUTION

◦ Up to eight hierarchies can be called for FB and FC nesting.

◦ Up to six hierarchies can be called for SBR nesting.
Interrupt: Interrupts do not need to be manually called. EI() must be called to enable interrupts and DI() must be called to disable interrupts.

NOTE

For software versions of AutoShop V4.11.0.0 or later and firmware versions of V6.5.0.0 or later, actual input arguments of FB, FC, and output instructions can be expressions.

6.1.6 Syntax Instructions

6.1.6.1 Overview

The overall LiteST program consists of instructions separated by semicolons (;). The following table lists the LiteST syntax instructions.

Instruction	Function	Example
:=	Assignment	A := B
Function block call	Function block call and output	TONR(IN := b0,PT := dVar,R := b0,Q => ,ET =>);

IF	Selection	IF A>0 THEN
		X:=10;
		ELSE
		X:=0;
		END_IF;

CASE	Multiway branch	CASE A OF 1: X:=1; 2: X:=2;

Instruction	Function	Example
		3: X:=3; ELSE X:=0; END_CASE;
WHILE	WHILE loop	A := 0; WHILE A <= 1000 DO A := A+7; END_WHILE;
REPEAT	REPEAT loop	A := 1; TOTAL := 0; REPEAT TOTAL := TOTAL + A; A := A+1; UNTIL A>10 END_REPEAT;
FOR	FOR loop	FOR i:=0 TO 100 DO X[i]:=0; END_FOR;
EXIT	EXIT loop	EXIT;
CONTINUE	Interrupting the current loop	CONTINUE;
RETURN	Return	RETURN;
(*Text*)	Comment	(*Comment out multiple lines IF A=3 THEN A:=5; END_IF;*)
//Text	Single-line comment	//A:=5;
;	Empty statement	;

6.1.6.2 Assignment Instructions

In an assignment statement, the evaluation result of the expression is used to replace the current values of one or multiple element variables. An assignment statement must contain a variable reference on the left, followed by the assignment operator ":=" and then the evaluation expression.

Example

```
A:=B x 10
```

After execution, the value of A is 10 times the value of B.

6.1.6.3 Calling a Function Block

Syntax: FB instance name (FB input variable := value, FB output variable => value,... More FB input and output variables);

Example: After you call an instance of the function block with the maximum value evaluated (MAXFB), load input parameters D0 and D1 and the output parameter D2, and execute the function. The result is assigned to the variable maxVar.

```
MYFB(VAR1 := D0,VAR2 := D1,RESULT => D2);
maxVar := MYFB.RESULT;
```

NOTE

myFB is the function block instance of MAXFB.

6.1.6.4 IF Instructions

IF instructions are used to execute relevant statements according to the calculation result of condition expressions.

In IF select statements, only the statements where the condition expression value is boolean 1(TRUE) can be executed. If the condition is set to 0(FALSE), no statement is executed or the statements in the ELSE (or ELSIF) keyword conditions are executed.

1. ELSIF is not required if there is only one condition expression. In addition, ELSE is not required if the Boolean expression indicating no condition is met is not processed. See the following figure.
   IF condition expression THEN

   statement;

   END_IF
2. One IF select statement can contain multiple ELSIF statements. See the following figure.
   IF condition expression1 THEN

   statement;

   ELSIF condition expression2 THEN

   statement;

   ELSIF condition expression3 THEN

   statement;

   ELSIF condition expression4 THEN

   statement;

   ELSE

   statement;

   END_IF

3. IF select statements can be nested. Statement 11 is executed when the condition expressions 1 and 11 are TRUE.

```
IF condition expression1 THEN
IF condition expression11 THEN
        statement 11;
ELSIF condition expression12 THEN
        statement 12;
ELSE
        statement 13;
END_IF
ELSIF condition expression2 THEN
        statement 2;
ELSE
        statement 3;
END_IF
```

Example

```
IF score <60 THEN
        bPass := FALSE;
ELSE
        bPass := TRUE;
END_IF
```

In the example, if the score is less than 60, the test fails. Otherwise, the test passes.

Note

- An IF select statement consists of one IF, one THEN, and one END_IF at least.
- The keyword must be ELSIF rather then ELSEIF.
- In a condition expression, the keyword THEN is used to determine whether the expression ends.

6.1.6.5 CASE Instructions

Function and instruction description

Statements to be executed are selected from multiple statements based on the value of the specified integral expression. An integral expression can be the return value of an INT variable, a DINT variable, an expression, or a function.

```
CASE integral expression OF
  Value 1: statement 1;
  Value 2: statement 2;
```

```
   Value 3, value 4: statement 3;
   Value 5, value 6: statement 4;
:
   ELSE
statement 5;
END_CASE;
```

Like the program expressed by LiteST, CASE instructions can be processed as follows:

- If the value of an integral expression is value 1, statement 1 is executed;
- If the value of an integral expression is value 2, statement 2 is executed;
- If the value of an integral expression is value 3 or value 4, statement 3 is executed;
- If the value of an integral expression is value 5 or value 6, statement 4 is executed;
- Otherwise, statement 5 is executed.

CASE select statements can be nested. Statement 12 is executed when the value of integral expression 1 is value 1 and the value of integral expression 11 is value 2.

```
CASE integral expression 1 OF
1 :
 CASE integral expression 11 OF
 1 : statement 11;
 2 : statement 12;
 ELSE
 statement 1m;
 END_CASE;
2 : statement 2;
3 : statement 3;
ELSE
 statement n;
END_CASE;
```

Example

```
CASE ERROR_CODE OF
      1:ERR_MSG := 'function lacking the right bracket';
      2:ERR_MSG := 'failure to process variables';
      3:ERR_MSG := 'invalid variable initial value';
      …
      255:ERR_MSG := 'function lacking the right bracket';
      ELSE ERR_MSG := 'unknown error';
END_CASE
```

Note

- In CASE statements, values of expressions must be integers.
- ELSE options are optional, and some programs can contain only the CASE...OF…END_CASE structure.

6.1.6.6 WHILE Instructions

Function and instruction description

When the calculation result of a specified condition expression is TRUE, a statement is repeatedly executed.

In a sense, the WHILE loop and REPEAT loop functions are more powerful than the FOR loop function because cycle times do not need to be counted before loop execution. Therefore, only the WHILE loop and REPEAT loop are required in some cases. However, if the cycle times is clear, the FOR loop is better.

Example

```
WHILE Counter<>0 DO
    Var1:= Var1*2;
    Counter := Counter-1;
END_WHILE
```

Note

WHILE must be used with END_WHILE in pair.

6.1.6.7 REPEAT Instructions

Function and instruction description

After a statement is executed once, repeat it before the value of the specified condition expression changes to TRUE. A REPEAT instruction requires statement running before condition expression evaluation. Therefore, the statement must be executed.

Example

```
A := 1;
TOTAL := 0;
REPEAT
 TOTAL := TOTAL + A;
 A := A+ 1;
UNTIL A>10
END_REPEAT;
```

Numbers 1 to 10 are added together, and the result is used for the variable TOTAL.

Note

REPEAT, UNTIL, and END_REPEAT are necessary.

6.1.6.8 FOR Instructions

Functions and instructions

The FOR loop can be used to compile the iterative processing logic.

FOR control variable := Initial value TO Final value{BY incremental value} DO

    statement;

END_FOR;

In the preceding program,

Information in the braces is optional.

The control variable is the counter. The statement will be executed only if the value on the counter is not greater than the final value. Before statement execution, check this condition. If the initial value is greater than the final value, the statement will not be executed.

After the statement is executed the last time, the counter automatically increases the incremental value. An incremental value can be any integer. If the parameter is not set, the default value is 1. When the value on the counter is greater than the final value, the loop stops.

Example

```
sumResult := 0;
factorial := 1;
FOR i :=1 TO 10 BY 1 DO
    sumResult := sumResult + i;
    factorial := factorial * i;
END_FOR;
```

In the preceding example, sumResult (result of adding 1 to 10) and the factorial result are calculated.

Precautions

FOR must be used with END_FOR in pair.

6.1.6.9 EXIT Instructions

Functions and instructions

An EXIT instruction is used to exit the FOR, WHILE, or REPEAT loop. The instruction interrupts iterative processing of the internal FOR, WHILE, or REPEAT instruction, and executes the next step of the iterative processing.

Example

```
IF A THEN
      DATA[3] :=98;
      FOR n := 1 TO 50 BY 1 DO
            DATA[n] := DATA[n] + n;
            IF DATA[n] > 100 THEN EXIT;
            END_IF;
      END_FOR;
A :=FALSE;
END_IF;
```

In the preceding example, the variable value starts from n=1 and increases to 50 by 1 repeatedly to add n to the specified sorting variable DATA[n]. However, after the value of DATA[n] exceeds 100, the operation ends.

Precautions

- This instruction must be used between FOR and END_FOR, WHILE and END_WHILE, or REPEAT and END_REPEAT.
- To interrupt all hierarchical iterative processing (nesting), the number of the EXIT instructions must be the same as that of hierarchies.

6.1.6.10 CONTINUE Instructions

Function and instruction description

A CONTINUE instruction is used to end a FOR, WHILE, or REPEAT loop in advance and start the next loop. It is different to interrupt a loop and exit a loop. When you interrupt a loop, the loop is ignored and the next loop is executed.

Example

```
FOR Counter:=1 TO 5 BY DO
    INT1:=INT1/2;
    IF INT1=0 THEN
        CONTINUE;
    END_IF
Var:=Var1/UBT1L
END_FOR;
```

Note

This instruction must be used between FOR and END_FOR, WHILE and END_WHILE, or REPEAT and END_REPEAT.

6.1.6.11 RETURN Instructions

Function and instruction description

This instruction forcibly ends main programs, subprograms, FBs, or FCs.

Example

```
IF b=TRUE THEN
    RETURN;
END_IF;
a:=a+1;
```

If b is “TRUE”, the statement “a:=a+1;” will not be executed, and POU will be immediately returned.

Note

If this instruction is frequently used, the process will be complex.

6.1.6.12 REGION Instructions

Feature and Instruction Description

The REGION instruction is used to group sections of code, allowing these sections to be flexibly expanded and collapsed. It supports nesting.

Preconditions

- PLC: Easy500 series, Easy300 series, and H5U series.
- Firmware version: V6.5.0.0 and later
- Software version: AutoShop V4.11.0.0 and later

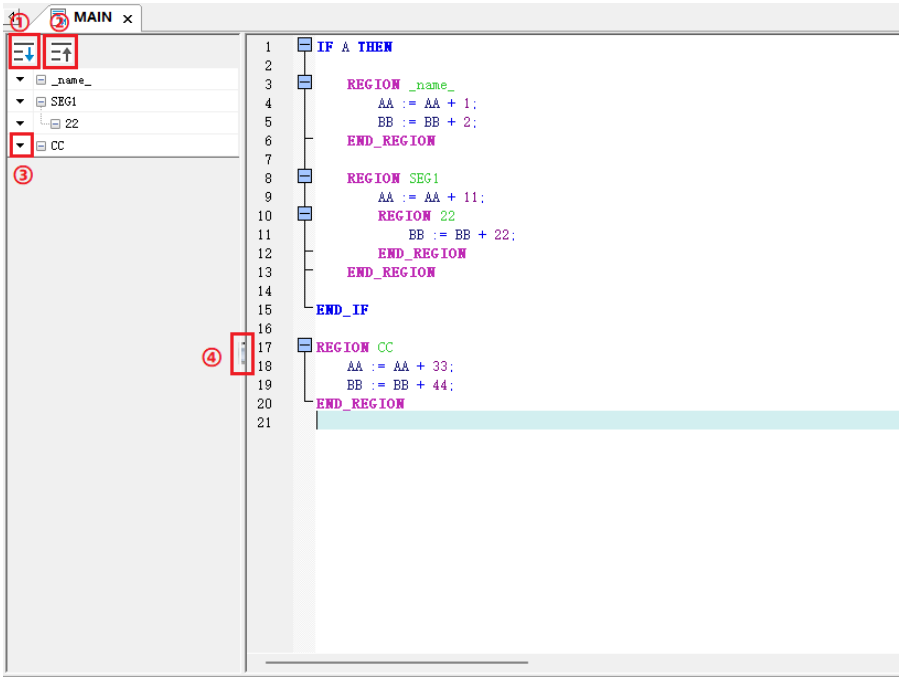
Example

```
IF A THEN
   REGION _name_
      AA := AA + 1;
      BB := BB + 2;
   END_REGION

   REGION program block 1
      AA := AA + 11;
      REGION 22
         BB := BB + 22;
      END_REGION
   END_REGION
END_IF

REGION CC
   AA := AA + 33;
   BB := BB + 44;
END_REGION
```

Description



- Unfold All: Click the button at (1) in the figure to unfold all REGION instructions.
- Fold All: Click the button at (2) in the figure to fold all REGION instructions.

- Unfold/Fold: Click the button at (3) for a specific REGION instruction to unfold or fold it. Clicking the "-"/"+" in the right of the program area will also synchronize the folding state to the corresponding entry in the left REGION outline.
- Show/Hide Outline Pane: Click the button at (4) to hide or show the left REGION outline pane. The REGION outline pane is hidden by default. It automatically becomes visible when a REGION instruction is entered in the program area.
- Clicking a line in the left REGION outline moves the cursor in the right program area to the end of that REGION segment name. Double-clicking a line in the left REGION outline allows you to rename that segment, and the change is synchronized to the program area.

Precautions

- This instruction cannot be used in conjunction with other keywords such as IF, WHILE, and FOR. If it is, the REGION instruction will not take effect.
- It is supported in ST programs for MAIN, SBR, INT, FB, and FC program organization units. Within execution blocks, only the REGION statement itself is supported. The left outline list is not available in this case.

6.1.6.13 Comment

Structured text can be commented in two ways:

- Single-line comment: Start with "//" and end with "//", for example, "// This is a comment.".
- Multi-line comment: Start with "(*" and end with "*)", for example, "(*This is a comment.*)".

Comments can be added to the LiteST editor declaration or any part in implementation.

Comment nesting: A comment can be inserted into another comment.

Example

```
(*
a:=inst.out; //to be checked
b:=b+1;
*)
```

6.1.7 List of Instructions

6.1.7.1 Basic Axis Control Instructions

Instruction	Description
MC_Power	Enable
MC_Reset	Reset
MC_ReadStatus	Read the axis status.
MC_ReadAxisError	Read the axis error.

Instruction	Description
MC_ReadDigitalInput	Read the DI state.
MC_ReadActualPosition	Read the currernt position.
MC_ReadActualVelocity	Read the current velocity.
MC_ReadActualTorque	Read the current torque.
MC_SetPosition	Set the current position.
MC_TouchProbe	Probe
MC_MoveRelative	Relative positioning
MC_MoveAbsolute	Absolute positioning
MC_MoveVelocity	Speed reference
MC_Jog	Jog
MC_TorqueControl	Torque control instruction
MC_Home	ZRN
MC_Stop	Stop
MC_Halt	Halt (Not recoverable)
MC_SetOverRide	Overshoot value reference
MC_MoveFeed	Positioning Interruption
MC_ImmediateStop	Emergency stop
MC_MoveVelocityCSV	CSV-based velocity control with adjustable pulse width
MC_SyncMoveVelocity	CSV-based synchronous velocity control of the PWM waveform
MC_FollowVelocity	CSP-based synchronous velocity control
MC_MoveBuffer	Multi-position
MC_MoveSuperImposed	Motion superposition
MC_SyncTorqueControl	Synchronous torque control
MC_SetAxisConfigPara	Instruction for setting axis parameters
MC_MoveLinear	Linear interpolation
MC_GroupStop	Stop axis group operation

Instruction	Description
MC_MoveCircular	Circular interpolation
MC_GroupPause	Pause axis group operation

6.1.7.2 Cam and Gear Instructions

Instruction	Description
MC_CamIn	Starting cam operation
MC_CamOut	End cam operation
MC_Phasing	Master axis phase shifting
MC_GenerateCamTable	Update cam table
MC_SaveCamTable	Save cam table
MC_GearIn	Start gear operation
MC_GearOut	End gear operation
MC_GetCamTablePhase	Obtain cam table phase
MC_GetCamTableDistance	Obtain cam table displacement
MC_DigitalCamSwitch	Electronic cam tappet control

6.1.7.3 Encoder Instructions

Instruction	Description
HC_Counter	High-speed counter enable
HC_Preset	High-speed counter preset
HC_TouchProbe	High-speed counter probe
HC_Compare	High-speed counter comparison
HC_ArrayCompare	High-speed counter array comparison
HC_StepCompare	High-speed counter step comparison
ENC_SetUnit	Encoder axis setting gear ratio (valid only for the local encoder axis)

Instruction	Description
ENC_SetLineRotationMode	Encoder axis setting linearity rotation mode (valid only for the local encoder axis)
ENC_Counter	Encoder axis enable
ENC_Reset	Encoder axis fault reset (only for the bus encoder axis)
ENC_ResetCompare	Encoder axis reset comparison output (only for the bus encoder axis)
ENC_Preset	Encoder axis preset value
ENC_TouchProbe	Encoder axis probe
ENC_Compare	Encoder axis comparison output (only for the local encoder axis)
ENC_ArrayCompare	Encoder axis array comparison
ENC_StepCompare	Encoder axis step comparison
ENC_GroupArrayCompare	Encoder axis array comparison (bus encoder axis)
ENC_ReadStatus	Encoder axis state read (bus encoder axis)
ENC_DigitalOutput	Encoder axis digital output control (only for the bus encoder axis)

6.1.7.4 Communication Instructions

Instruction	Description
ETC_ReadParameter_CoE	Read the SDO parameter of the ETC slave.
ETC_WriteParameter_CoE	Write the SDO parameter of the ETC slave.
ETC_RestartMaster	Restart the ETC master.

6.1.7.5 Timer Instructions

Instruction	Description
TPR	Pulse timer
TONR	Connection delay timer
TOFR	Off delay timer
TACR	Accumulation timer

6.1.7.6 Interrupt Instructions

Instruction	Description
EI	Enable interrupt
DI	Disable interrupt

6.1.7.7 Operation Instructions

Instruction type	Instruction description	Instruction name
Mathematical operation instruction	Modulo	MOD
Shift instruction	Bitwise left shift	SHL
	Bitwise right shift	SHR
Selection instruction	Either one	SEL
	Maximum value	Max.
	Minimum value	Min.
Math Instructions	Absolute value	ABS
	Square root	SQRT
	Natural logarithm	LN
	Common logarithm	LOG
	Exponentiation	EXPT
	Sine function	SIN
	Cosine function	COS
	Tangent function	TAN
	Arcsine function	ASIN
	Arccosine function	ACOS
	Arctan function	ATAN
Word logic	And	AND
	OR	OR

Instruction type	Instruction description	Instruction name
	Bit reversal	NOT
	Logical exclusive OR	XOR
Data type conversion instructions	BOOL to INT	BOOL_TO_INT
	BOOL to DINT	BOOL_TO_DINT
	BOOL to REAL	BOOL_TO_REAL
	INT to REAL	INT_TO_REAL
	INT to DINT	INT_TO_DINT
	INT to BOOL	INT_TO_BOOL
	DINT to REAL	DINT_TO_REAL
	DINT to INT	DINT_TO_INT
	DINT to BOOL	DINT_TO_BOOL
	REAL to DINT	REAL_TO_DINT
	REAL to INT	REAL_TO_INT
	REAL to BOOL	REAL_TO_BOOL
	To BOOL	TO_BOOL
	To INT	TO_INT
	To DINT	TO_DINT
	To floating-point number	TO_REAL

6.1.7.8 Other Instructions

Instruction	Description
ZSET	Batch setting
ZRST	Batch reset
BITW	16-Bit variable conversion to word variable
WBIT	Word variable conversion to 16-bit variable
MSET	Memory setting

Instruction	Description
MCPY	Memory copying
R_TRIG	Rising edge inspection trigger
F_TRIG	Falling edge inspection trigger

6.1.7.9 Instruction Examples

Use examples of axis control instructions

MC_Power use example

```
MC_Power(Enable := b0, //
    Axis := Axis_0, //Axis name/axis ID
    Status => b4, //Axis enable flag
    Busy => b6, //Busy flag
    Error => b8, //Instruction fault flag
    ErrorID => d0); //Fault code
```

MC_Jog use example

```
MC_Jog(Enable := b0, //
    Axis := Axis_0, //Axis name/Axis ID
    JogForward := b1, //Forward motion, with the valid level for rising edge trigger
    JogBackward := b2, //Reverse motion, with the valid level for rising edge trigger
    Velocity := d0:E, //Target speed
    Acceleration := d2:E, //Acceleration rate
    Deceleration := d4:E, //Deceleration rate
    CurveType := d6, //Curve type. 0: T-type speed curve; 1: Five-segment S-curve
    Busy => b3, //Busy flag
    CommandAborted => b4, //Execution termination
    Error => b5, //Instruction fault flag
    ErrorID => d8); //Fault code
```

Use examples of mathematical operation functions

Max use example

```
c:=MAX(a,b);
```

Use examples of other instructions

R_TRIG (rising edge) use example

NOTE

(*If X0 changes from FALSE to TRUE, the rising edge trigger.Q is TRUE. Execute the M0 assignment statement in the if statement. Otherwise, do not execute the statement.*)

```
Rising edge trigger(CLK := X0,Q => );
if rising edge trigger.Q THEN
    M0:=TRUE;
END_IF;
```

F_TRIG (falling edge) use example

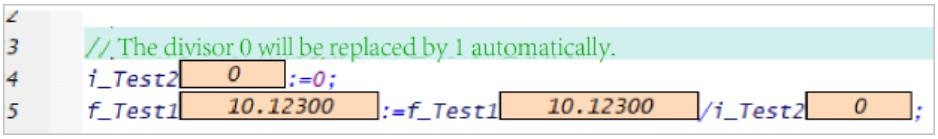
(*If B0 changes from TRUE to FALSE, the falling edge trigger.Q is TRUE. Execute the M0 assignment statement in the if statement. Otherwise, do not execute the statement.*)

```
Falling edge trigger (CLK := B0,Q => );
if falling edge trigger.Q THEN
    M0:=TRUE;
END_IF;
```

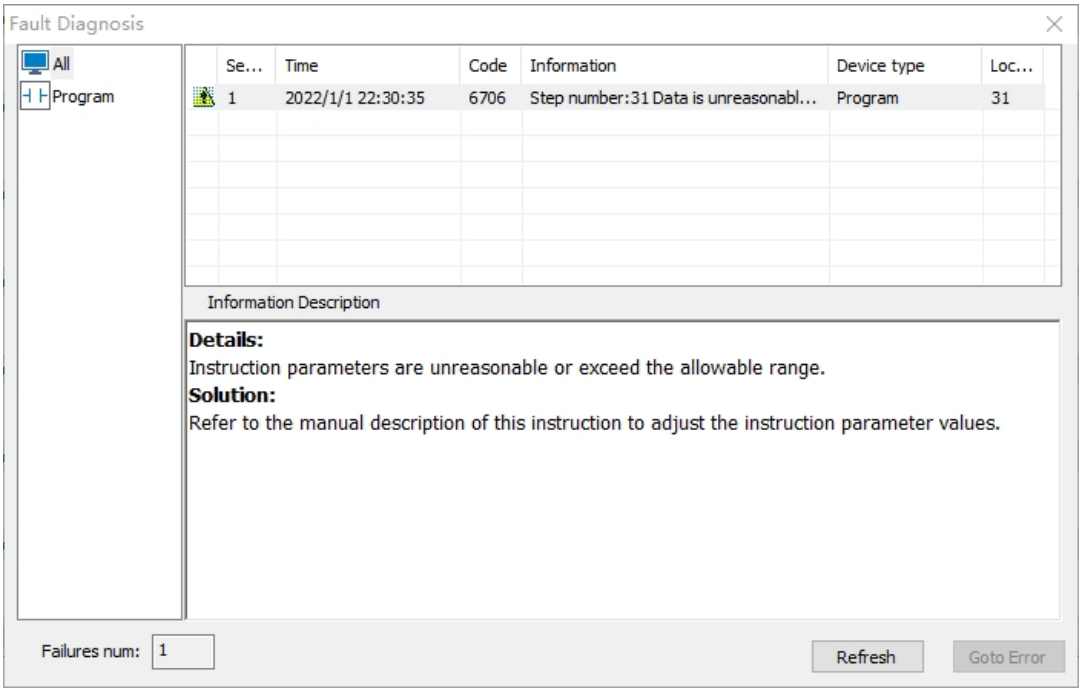
6.1.8 Troubleshooting

6.1.8.1 Protection Against Zero Divisor

If the divisor is 0, it automatically changes to 1, as shown in the following figure.



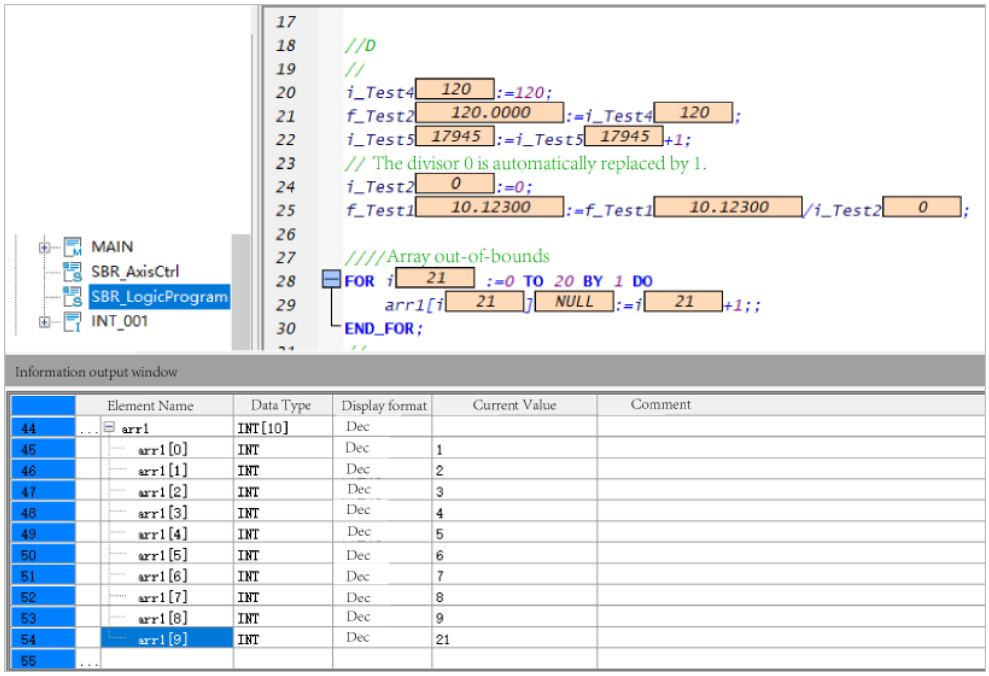
The PLC LED alternatively flashes Er50 and Er81.



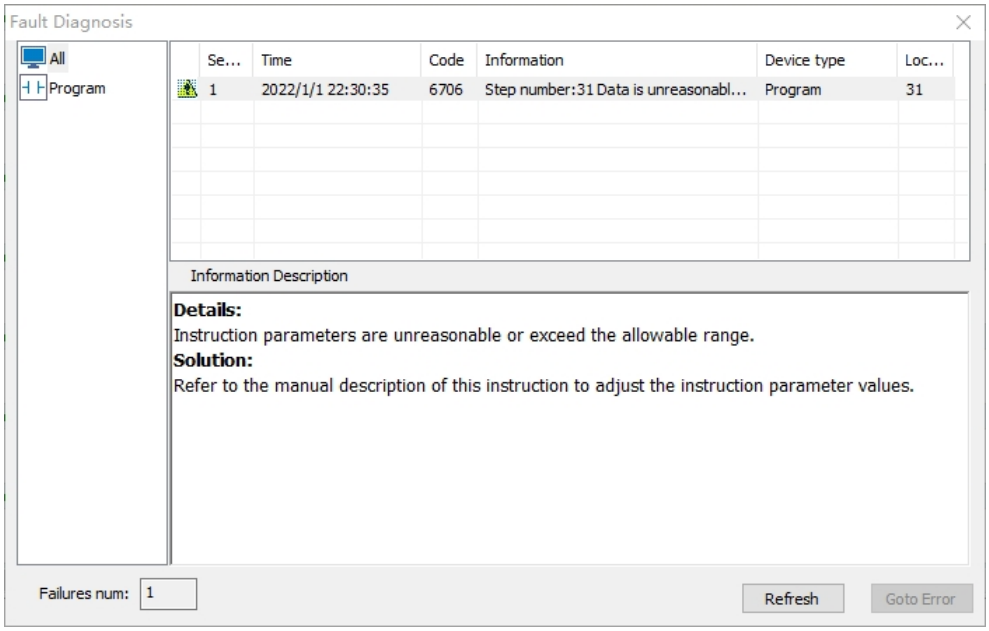
6.1.8.2 Array Out-of-Bounds

Array out-of-bounds is automatically checked during compilation for constants and during running for variables. Values over the upper limit are stored in the element with the maximum subscript in the array, and values below the lower limit are stored in the element with the subscript 0 in the array.

As shown in the following figure, an INT array is defined with up to 10 elements. In case of out-of-bounds, the software reports an error and stores the values out of the range to the element with the subscript (9) in the array.

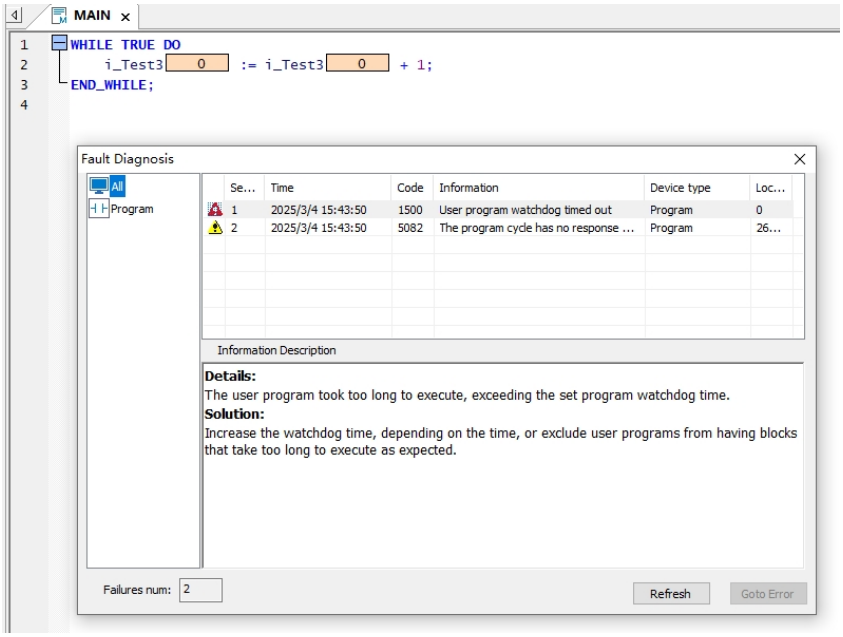


The PLC LED alternatively flashes Er50 and Er81.



6.1.8.3 Infinite Loop

The program is automatically checked for an infinite loop. In case of an infinite loop, an error is reported, the infinite loop is automatically displayed, and the program stop running. The fault diagnosis page shows the error information and the error is located, as shown in the following figure.



The PLC LED alternatively flashes Er1500 and Er5082.

In the error information, Er1500 indicates watchdog timeout and Er5082 indicates an infinite loop alarm.

6.1.8.4 Precautions for Array Subscript Usage

- The subscript of a constant array cannot exceed the array size.
- The subscript of an array cannot be a soft element.
- The subscript of an array in an FB or FC cannot be a global variable.
- The subscript of an array cannot be an expression.
- A variable with a complex structure can contain up to only one variable array sub-index.

6.2 Ladder Diagram (LD)

6.2.1 Overview

Introduction

LD is a graphical programming language. Its structure is similar to that of the circuit diagram. LD contains a series of networks (also called nodes), and each network starts from the vertical line on the left (the power rail and power flow line). A network consists of points of contact, coils, arithmetic blocks (functions, function blocks, programs, execution blocks, actions, and methods), jump instructions, labels, and connecting wires.

- LD mainly includes points of contact, coils, arithmetic blocks, branches, and comments. These elements are inserted, dragged, scribed, and copied and pasted to networks to form the LD execution logic.
- LD provides online commissioning functions such as monitoring, written values, force values, and breakpoints.
- Double-click the blank space in the ladder diagram to pop up the common instruction selection box, and display and edit the ladder diagram comments.
- LD editing supports input expressions and expression verification, and the types of expression operators supported are the same as those supported by ST language.

For details about the LD programming language, see the "AutoShop.chm". To obtain the manual, you can choose "Help" > "Help Manual" in the AutoShop menu bar and double-click "AutoShop.chm" in the "Manual" folder.

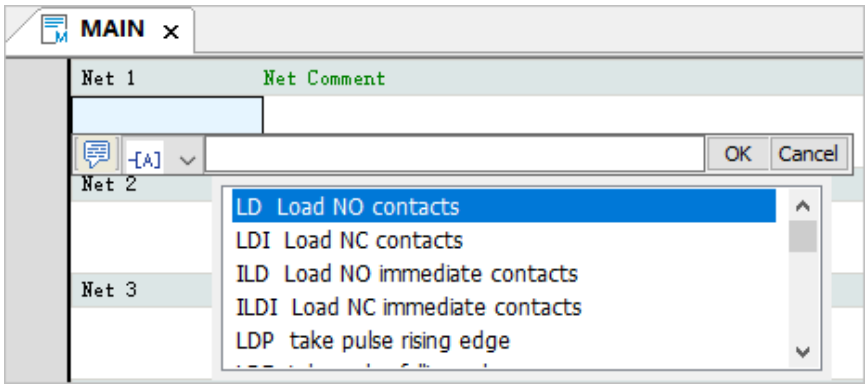
Preconditions

- Applicable models
  - Easy series: Easy523, Easy522, Easy521, Easy502, Easy501, Easy320, Easy302, and Easy301
  - H5U series: H5U, H5U-A8, H5U-A8S, and H5U-A16
- Firmware version: Not involved
- Software version: AutoShop V4.10.2.0 and later

> **NOTE**
>
> The software of AutoShop V4.11.0.0 and later and firmware of V6.5.0.0 and later support expressions and expression verification for LD.

Procedure

1. Double-click the blank space of the ladder diagram in the offline or online modification mode. The common instruction selection box is displayed, as shown in the following figure. This box cannot be displayed by double-clicking the ladder diagram in the monitoring mode.



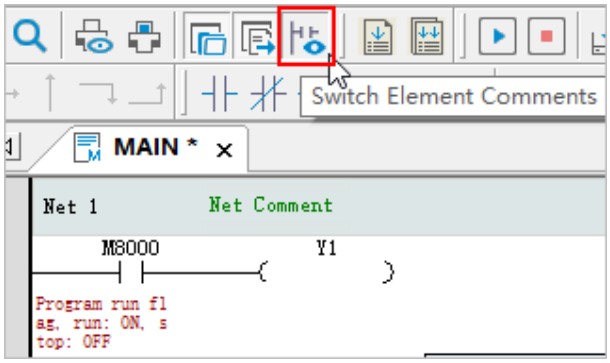
The following table lists contacts, coils, and instructions of common instructions.

Instruction	Description
LD	Load NO contact
LDI	Load NC contact
ILD	Load NO immediate contact
LDI	Load NC immediate contact
LDP	Obtain pulse rising edge
LDF	Obtain pulse falling edge
STL	Start of step ladder diagram
LD&	16-bit logical AND operation
OUT	Output
MOV	16-bit transmission

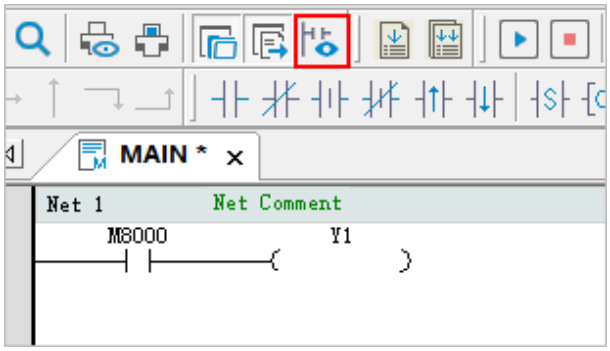
Instruction	Description
DMOV	32-bit transmission
LD<	16-bit LD contact comparison less than
LD<=	16-bit LD contact comparison less than or equal to
LD<>	16-bit LD contact comparison not equal to
LD=	16-bit LD contact comparison equal to
LD>	16-bit LD contact comparison greater than
LD>=	16-bit LD contact comparison greater than or equal to
LDD&	32-bit LD logical AND operation
LDD<	32-bit LD contact comparison less than
LDD<=	32-bit LD contact comparison less than or equal to
LDD<>	32-bit LD contact comparison not equal to
LDD=	32-bit LD contact comparison equal to
LDD>	32-bit LD contact comparison greater than
LDD>=	32-bit LD contact comparison greater than or equal to

2. Edit and display ladder diagram comments.

1. On the main screen of the ladder diagram, an element comment shortcut icon is added. When this icon is depressed, the comment function is enabled, as shown in the following figure.



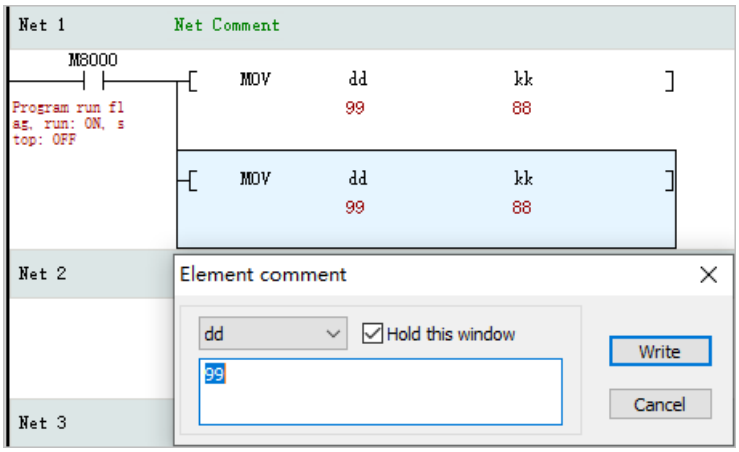
When the icon is normally displayed, the comment function is disabled, as shown in the following figure.



2. Double-click the ladder diagram to modify the comments.

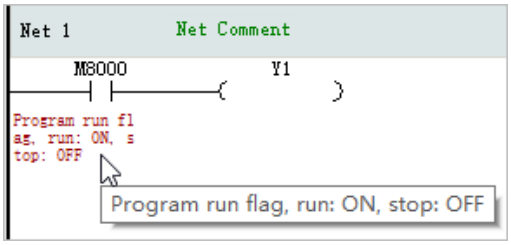
This operation is supported in offline commissioning or online modification mode and is not supported in the monitoring mode (the comment function must be enabled).

Double-click a variable comment. The "Element comment" dialog box is displayed next to the comment, in which you can modify the comment.



3. When a variable comment is enabled, the comment content is displayed in a floating window.

Hover over the variable comment. The comment content is displayed in a floating window, avoiding truncation due to length limit.



6.2.2 En/Eno Instructions

Introduction

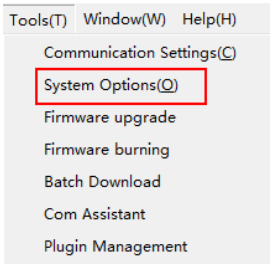
En/Eno supports serial connection with other instructions. By converting one-dimensional instructions into a two-dimensional format, they are displayed as graphic blocks, with an Eno pin added to connect to subsequent instructions.

Preconditions

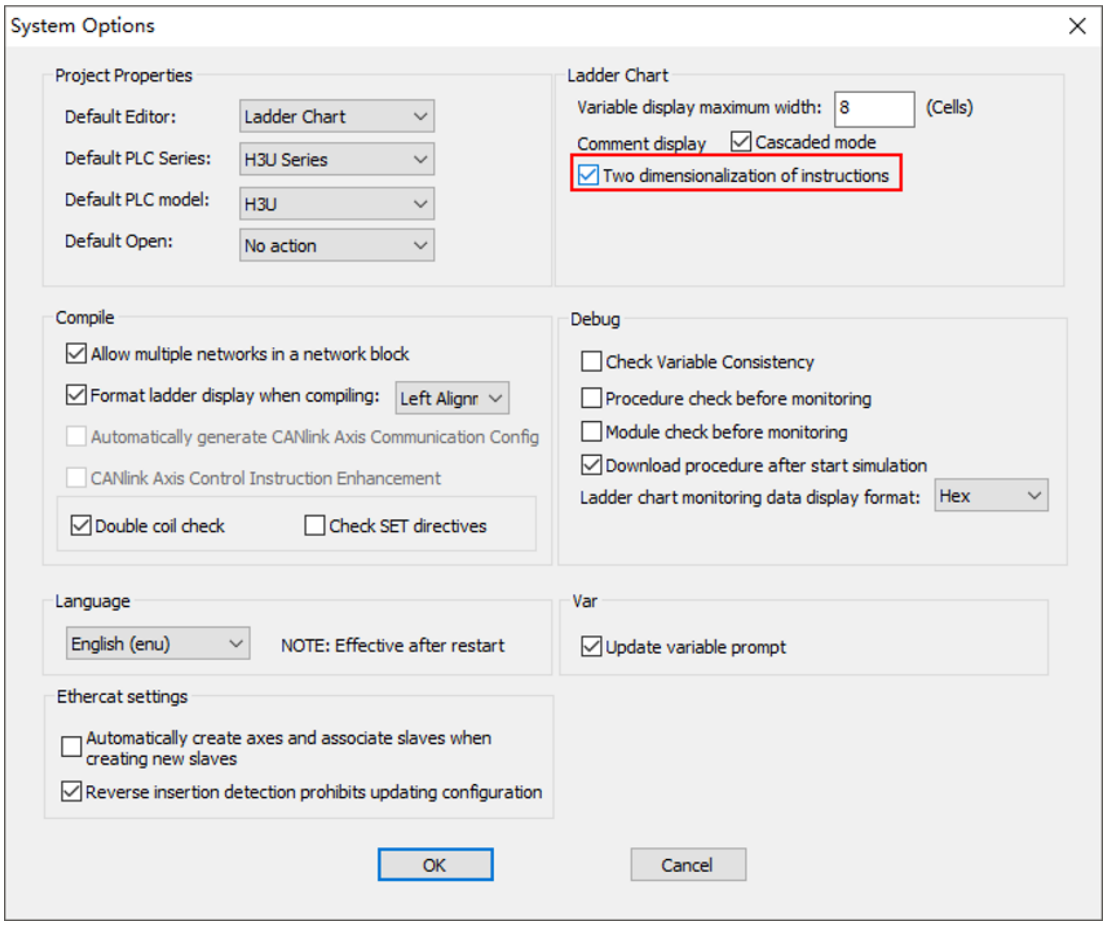
- PLC: Easy Large Point series, Easy500 series, Easy300 series, and H5U series.
- Firmware version: V6.5.0.0 and later
- Software version: AutoShop V4.11.0.0 and later
- The following instruction types support En/Eno:
  - Contact operation instructions
  - Data operation instructions
  - Data processing instructions
  - Matrix instructions
  - String instructions
  - Clock instructions
  - Pointer instructions

Procedure

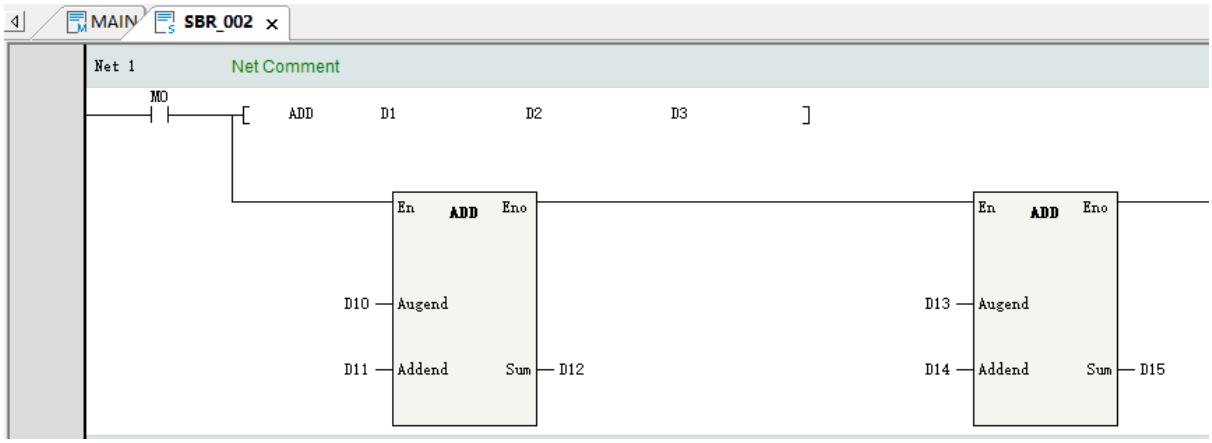
1. In the menu bar, choose "Tools" > "System Options".



2. In the "System Options" dialog box displayed, select "Two dimensionalization of instructions" under "Ladder Chart". LD elements supports En/Eno

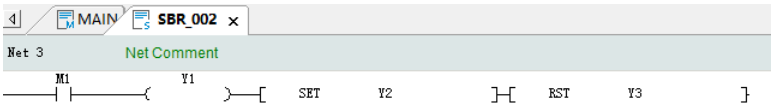


The program interface is displayed. Other elements can be added to the end of the instruction.



NOTE

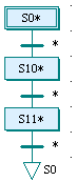
Output elements such as OUT, SET, and RST instructions only support serial connection of output instructions (regardless of whether the two-dimensionalization option is selected, they remain one-dimensional instructions). Subsequent connected elements are not allowed to connect to other elements, nor are they allowed to be connected in parallel.



6.3 Sequential Function Chart (SFC)

A Sequential Function Chart (SFC) is a language in which control is divided into multiple steps and steps are switched over based on the process or operation of mechanical devices. A standard SFC consists of initial step, common steps, conditions for step-to-step switch, jump, and reset. Each step is an operation of mechanical devices. An embedded ladder diagram (LD) can be included in a step, which indicates the operation to be finished in this step. Conditions for switch mean the conditions for finishing an operation and starting the next operation, which are indicated by embedded LDs.

Example



As shown in the preceding SFC, the initial step is S0; if the condition for S0 to-S10 switch is not met, S0 will be executed until the condition is met; when the condition is met, S10 will be executed, and so on; when S11 is finished and the condition for S11-to-S0 switch (reset) is met, S11 will be stopped and S0 will be restarted.

6.3.1 SFC Elements

Stepper

Steppers can be classified into initial steppers and common steppers. They control steps of sequential control programs through the status (S) of internal soft elements.

- **Initial stepper**
  A Sequential Function Chart (SFC) network must start with initial steppers.
  The addresses of S soft elements corresponding to the initial stepper are S0 to S9. That is, diagram element symbols S0* to S9*, which can be added by clicking the initial stepper icon on the toolbar.
- **Common stepper**
  The address of S soft elements corresponding to common steppers are greater than S9. That is, all S elements following the diagram element S10* (including S10) can be added by clicking the common stepper icon on the toolbar.

Ladder Diagram (LD) Block

The diagram element symbol of an LD block is LAD0*. You can click the LD block icon on the toolbar to add an LD block.
An LD block includes an embedded ladder chart, which can be used to execute user-defined code outside the SFC network.

Switcher

The diagram element symbol of a switcher is . You can click the switcher icon on the toolbar to add a switcher.

A switcher includes an embedded ladder chart. You can define code and decide whether to set S soft elements connected to the switcher.

Resetter

The diagram element symbol of a resetter is . You can click the resetter icon on the toolbar to add a resetter.

A resetter is used to reset S soft elements.

Jumper

The diagram element symbol of a jumper is . You can click the jumper icon on the toolbar to add a jumper.

A jumper is used to set specified S soft elements so that the code of embedded ladder chart can be executed.

Connection line

Connection lines connect the preceding SFC elements from top to bottom. In addition to connection, connection lines include selection branches, selection confluences, parallel branches, and parallel confluences.

- **Selection branch**

  The diagram element symbol of a selection branch is . You can click the selection branch icon on the toolbar to add a selection branch.

  A selection branch must start with a stepper and ends with a switcher. It means that if the stepper is successfully executed, a branch should be selected based on the conditions of the switcher.

- **Selection confluence**

  The diagram element symbol of a selection confluence is . You can click the selection confluence icon on the toolbar to add a selection confluence.

  A selection confluence must start with multiple switchers and ends with a common stepper, jumper, or resetter. It means that if conditions of one switcher are met, elements connected to the switcher will be run.

- **Parallel branch**

  The diagram element symbol of a parallel branch is . You can click the parallel branch icon on the toolbar to add a parallel branch.

  A parallel branch must start with a switcher and ends with a common stepper, jumper, or resetter. It means that if conditions of the switcher are met, elements connected to the parallel branch will be run simultaneously.

- **Parallel confluence**

  The diagram element symbol of a parallel confluence is . You can click the parallel confluence icon on the toolbar to add a parallel confluence.

  A parallel confluence must start with common steppers and ends with a switcher. It means that if one of the common steppers is successfully executed, the switcher code will be executed.

6.3.2 SFC Program Compilation

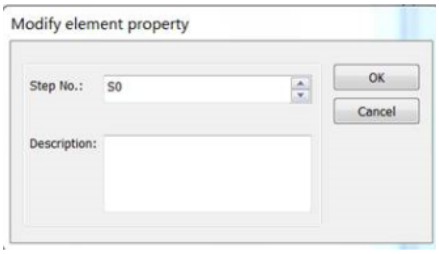
Inputting SFC Elements

All Sequential Function Chart (SFC) elements have corresponding icons or options on the SFC toolbar and in the SFC menu. You can click an icon and enter properties of the SFC element in the **Input element property** dialog box. In addition, shortcut icons ( ) of connection lines are available on the SFC toolbar and in the SFC menu. You can correlate SFC elements based on the programs.

Modifying SFC Elements

The scope of properties to be modified varies by SFC element. For an initial stepper, common stepper, jumper, and resetter, the step number and comments can be modified. For an ladder diagram (LD) block, the element name can be modified. For a switcher, element comments can be modified.

Double-click an element. The **Modify element property** dialog box is displayed. Modify properties of the element. The following figure shows the **Modify element property** dialog box.

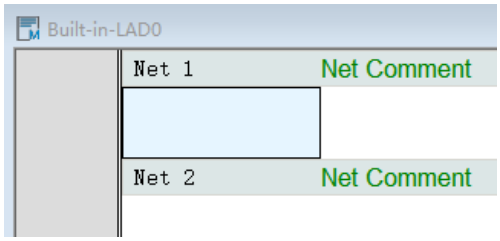


Modifying Embedded LDs of the SFC

In the embedded LD editor, you can embed a custom embedded LD in an LD block, stepper, or switcher.

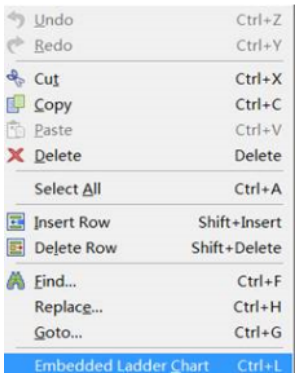
- **Method 1**

  Click the LD block, stepper, or switcher to open the embedded LD editor.



- **Method 2**

  Right-click on the LD block, stepper, or switcher and select **Embedded Ladder Chart**.



Copy/Paste

Select a cell or select an area by mouse dragging, and choose **Edit** > **Copy** or press CTRL+C to copy the element to the clipboard. Then choose **Edit** > **Paste** or press CTRL+V to paste the element to any other directory or SFC.

Insert/Delete Row

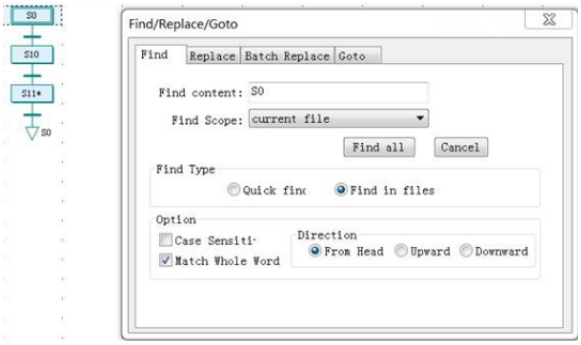
- **Insert Row**
  Insert a blank row into the SFC.
  Operation: Right-double on a cell and select **Insert Row**. The system moves down the row and elements in all rows below to make a blank row.
- **Delete Row**
  Delete a row from the SFC.
  Operation: Right-double on a cell and select **Delete Row**. The system deletes the row and moves a row up the elements in all rows below.

6.3.3 Find, Replace, and Goto

A Sequential Function Chart (SFC) has standard Find, Replace, and Goto functions.

Find

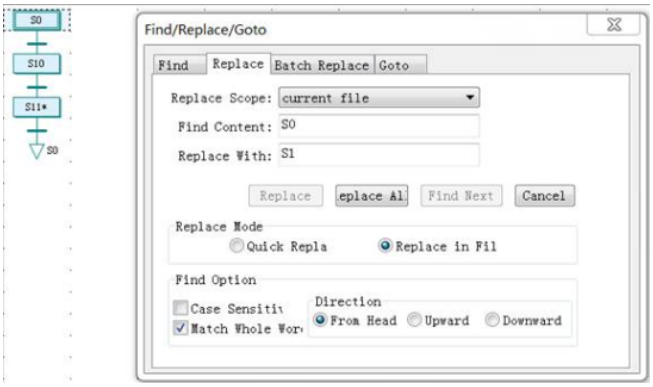
Press Ctrl+F or choose **Edit** > **Find**. The following window is displayed.



Enter strings in the **Find Content** text box, click **Find Next**. The found content is highlighted. Other options are defined the same way as those in a standard **Find** window. Click **Cancel**. The window is closed.

Replace

Press Ctrl+H or choose **Edit** > **Replace**. The following window is displayed.



Enter strings in the **Find Content** text box and the **Replace With** text box respectively, click **Replace**. The system automatically finds the next strings consistent with those entered in the **Find Content** text box and replaces them with the strings entered in the **Replace With** text box. Click **Replace All**. The system automatically finds all strings consistent with those entered in the **Find Content** text box and replaces them with the specified strings.

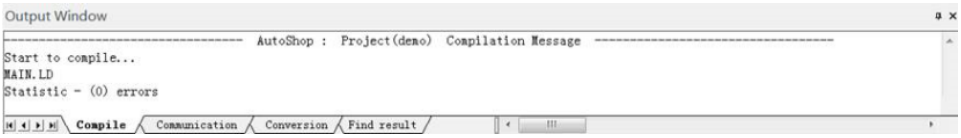
Goto

Move the cursor to the target line. Alternatively, press Ctrl+G or choose **Edit** > **Goto**. The following window is displayed.

Enter the line number in the **Line N** text box and click **Goto**.

6.3.4 SFC Compilation

After editing the Sequential Function Chart (SFC), click on the toolbar to check whether the SFC is correct. If no errors exist, a message indicating successful compilation is displayed in the output window.



6.4 Tips

6.4.1 ST Quick Input

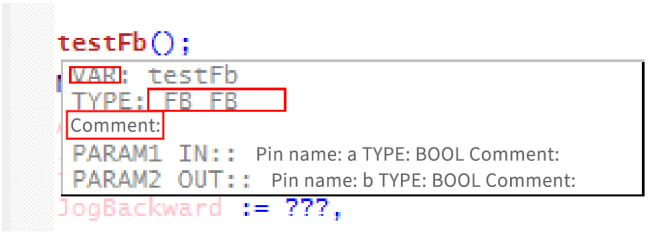
After entering the instruction name, press the tab key to complement the instruction pin. If the default parameter next to the pin is "???", the parameter must be input. Otherwise, the parameter can be input or not as required.

```
MC_Jog(Enable := ???,
       Axis := ???,
       JogForward := ???,
       JogBackward := ???,
       Velocity := ???,
       Acceleration := ???,
       Deceleration:= ,
       CurveType:= ,
       Busy => ,
       CommandAborted => ,
       Error => ,
       ErrorID => );
```

6.4.2 ST Prompt Upon Hovering

- When you hover the mouse over a variable, the variable name, type, and comment are displayed.

- When you hover the mouse over an FB, FC, or instruction, the function name, function type, function comment, input and output parameters, pin name, type, and comment are displayed.



6.4.3 ST Automatic Bracket Completion

Preconditions

- Firmware version: V6.5.0.0 and later
- Software version: AutoShop V4.11.0.0 and later

Introduction

When you enter a left bracket during ST programming, the system automatically completes the right bracket.

6.4.4 Unit Conversion

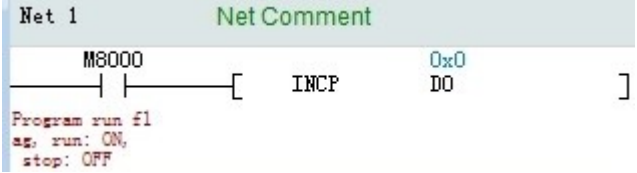
Preconditions

- PLC: Easy Large Point series, Easy500 series, Easy300 series, H5U series, H3U series, H2U series, H1U series, and H0U series
- Firmware version: V6.5.0.0 and later
- Software version: AutoShop V4.11.0.0 and later

Introduction

This function can be implemented in the following three ways.

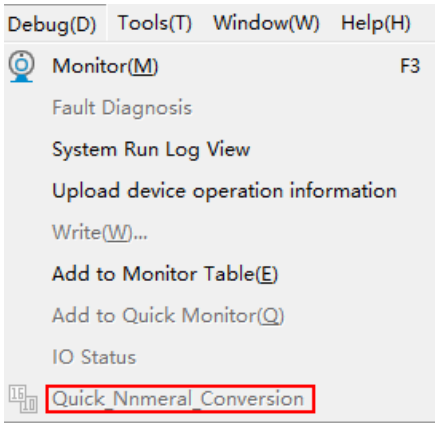
- Method 1: Open the program interface, click the button in the toolbar. The numerical system in the program area automatically switches between decimal and hexadecimal.



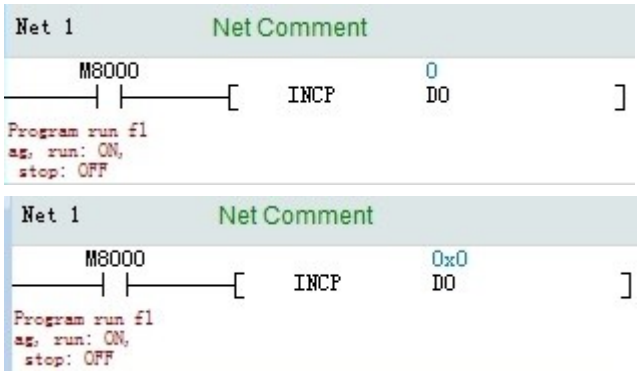
NOTE

- This function can be used during monitoring and online modification.
- Numerical conversion for INT, DINT, and BYTE values is supported.

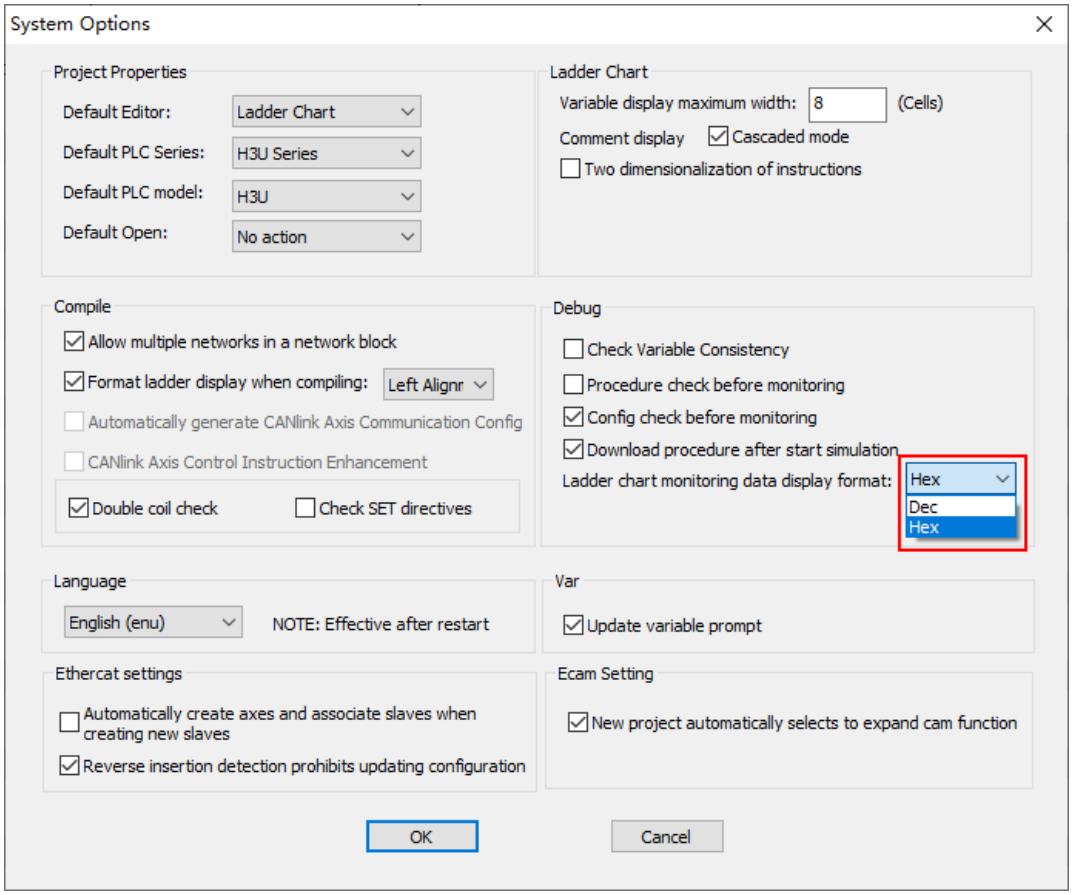
- Method 2: In the menu bar, Choose "Debug" > "Quick Numeral Conversion".



The numerical system in the program area automatically switches between decimal and hexadecimal.



- Method 3: In the menu bar, choose "Tools" > "System Options". In the "System Options" window displayed, modify "Ladder chart monitoring data display format".



7 Extension Modules

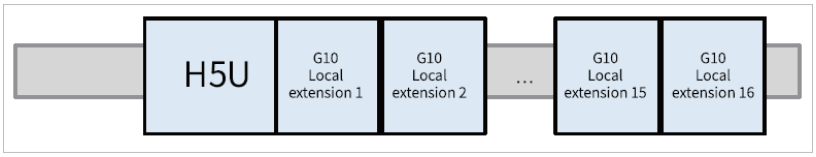
7.1 H5U Extension Module Configuration

7.1.1 Overview

Overview

H5U can carry up to 16 local extension modules, and can access local extension based on module configuration.

The following figure shows the hardware configuration for H5U to connect to local extension modules.



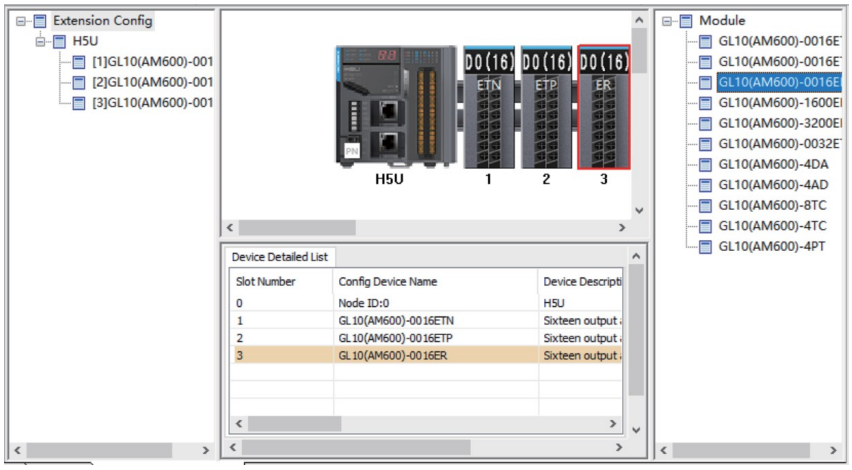
The following table lists the supported models of local extension modules.

Module	Description
GL10-0016ETP	16 digital output (DO) transistor module - PNP
GL10-0016ETN	16 DO transistor module - NPN
GL10-0016ER	16 DO relay module
GL10-1600END	16 DI module
GL10-3200END	32 DI module
GL10-0032ETN	32 DO module
GL10-4AD	4 analog input (AI) module
GL10-4DA	4 analog output (AO) module
GL10-4PT	4-in resistance temperature detector (RTD) module
GL10-4TC	4-in thermocouple temperature detection (TC TEMP MEAS) module
GL10-8TC	8-in thermocouple temperature detection (TC TEMP MEAS) module

7.1.2 Hardware Configuration

Local expansion modules are implemented by hardware configuration. To configure hardware, perform the following steps:

1. In the "Project Manager" tree, unfold "Config" and double-click "Module Config". The "Expansion Config" page is displayed.



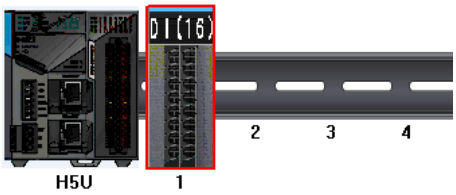
2. Click the position number of an expansion module on the guide rail, double-click the module on the right or drag and move the module to the guide rail, and configure the expansion module.

7.1.3 Configuring an Extension Module

7.1.3.1 Configuring a Digital Input (DI) Module

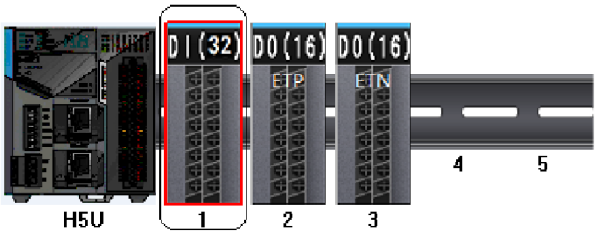
DI modules included GL10-1600END and GL10-3200END. The method of using a DI module as a local extension module is as follows:

1. In the module list, select a module to be added, and double-click the module for automatic extension on the extension rack, or drag the module to the extension rack.

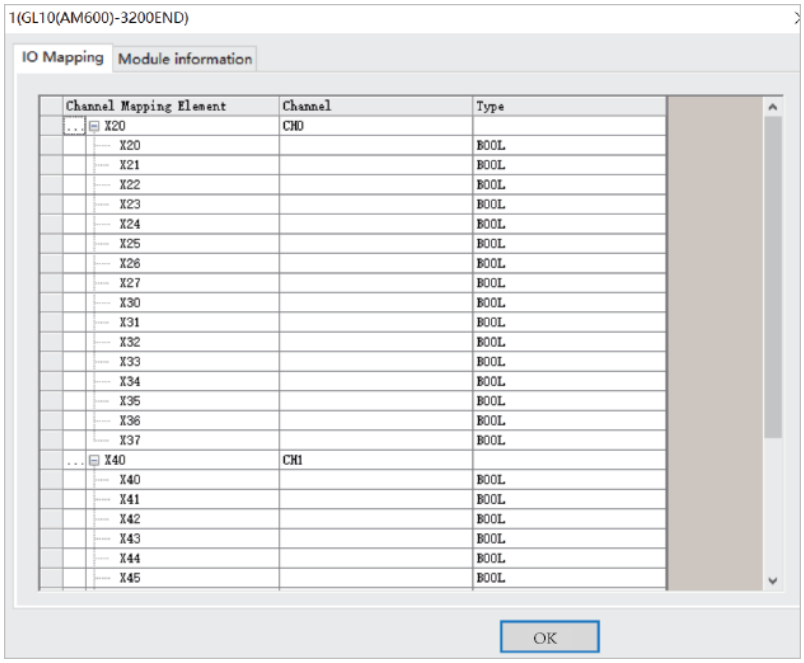


2. After local DI extension modules are connected to the master module without port configuration, numbers of input X ports on the extension modules follow the number of the input X port on the master module in sequence.
   For example, if the master module is a general H5U model, and number of the last X port on the master module is X37 after GL10-1600END connection, numbers of 16 input X ports on the extension module range from X40 to X47 and X50 to X57 during programming. This also applies to other DI extension modules by analogy.

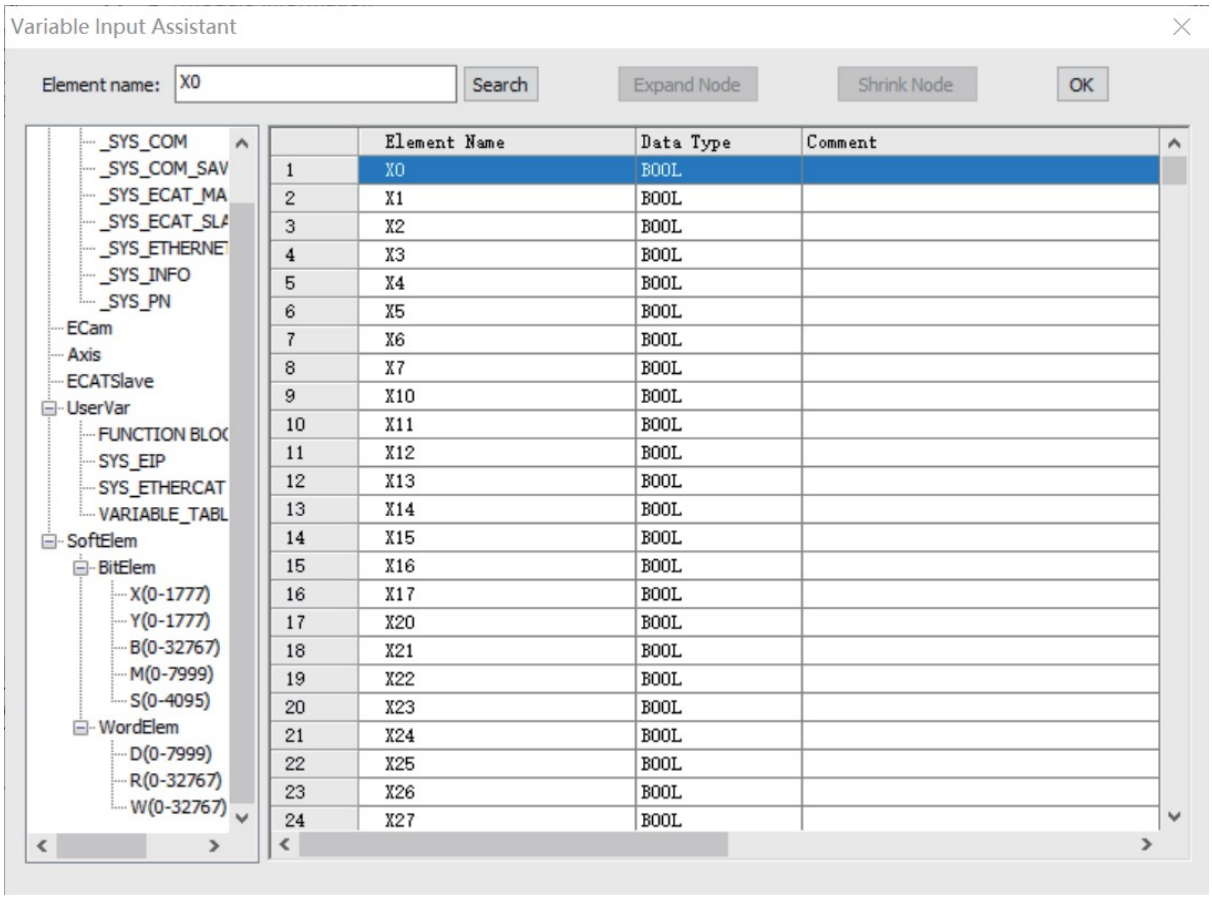
   H5U series PLCs allow manual configuration of port numbers. You can double-click a module on the module configuration page to access the port configuration page.



Use module 1 as an example. Double-click module 1 to access the module configuration page.



Click "..." to configure the port number as required.



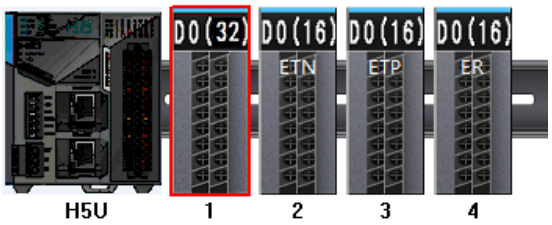
NOTE

- The port mapping of the H5U module is determined by the configuration. If there is no special configuration, the software arranges the port mapping in order. Even if the front-end module is deleted after the configuration is completed, the port mapping of the subsequent modules will not change.
- The relay output extension module can be connected to the relay or transistor main module. Similarly, the transistor input extension module can be connected to the transistor or relay main module.

7.1.3.2 Configuring a Digital Output (DO) Module

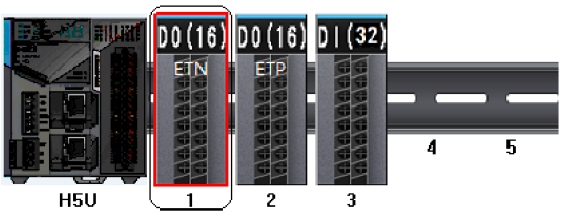
DO modules include GL10-0016ETP, GL10-0016ETN, GL10-0016ER, and GL10-0032ETN. The method of using a DO module as a local extension module is as follows:

1. In the module list, select a module to be added, and double-click the module for automatic extension on the extension rack, or drag the module to the extension rack.

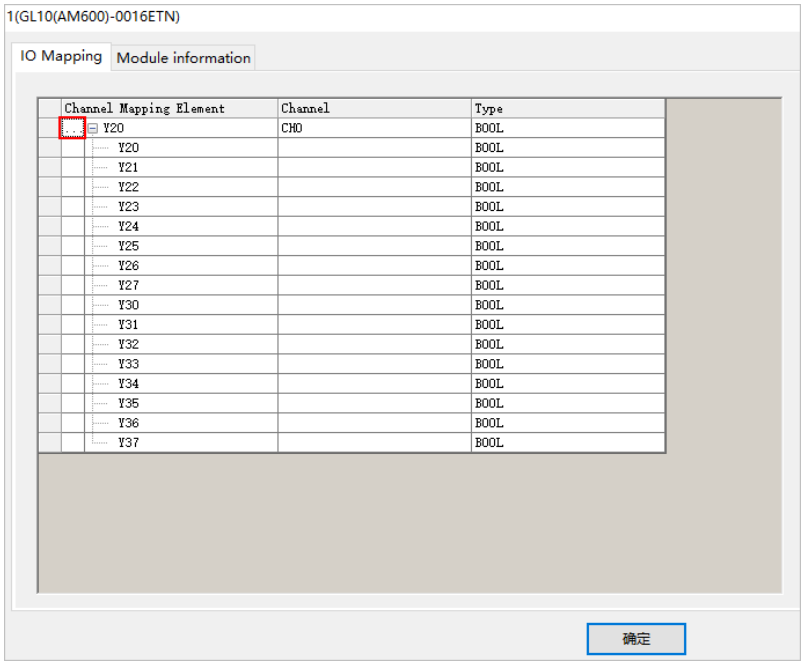


2. After local DO extension modules are connected to the master module, numbers of output Y ports on the extension modules follow the number of the Y port on the master module in sequence.
   For example, if the master module is a general H5U model, and number of the last Y port on the master module is Y37 after GL10-0016END connection, numbers of 16 output Y ports on the extension module range from Y40 to Y47 and Y50 to Y57 during programming. This also applies to other DO extension modules by analogy.

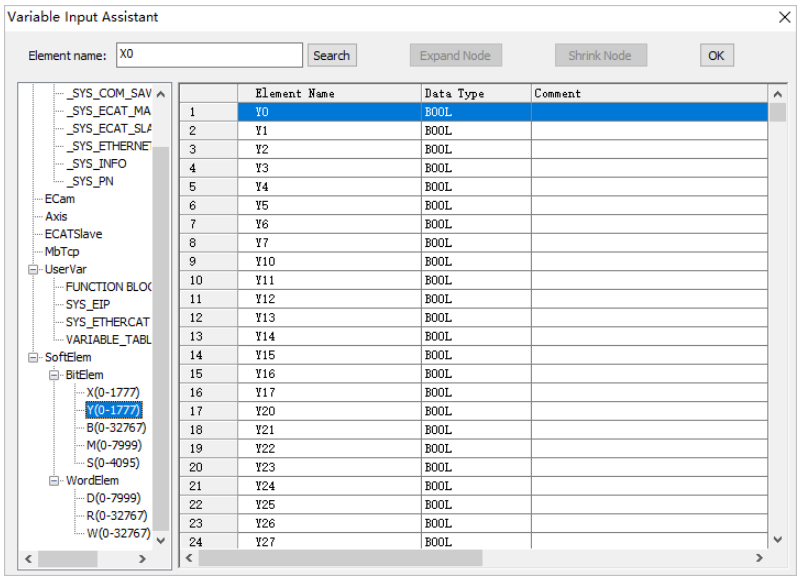
   H5U series PLCs allow manual configuration of port numbers. You can double-click a module on the module configuration page to access the port configuration page.



   Use module 1 as an example. Double-click module 1 to access the module configuration page.



Click "..." to configure the port number as required.



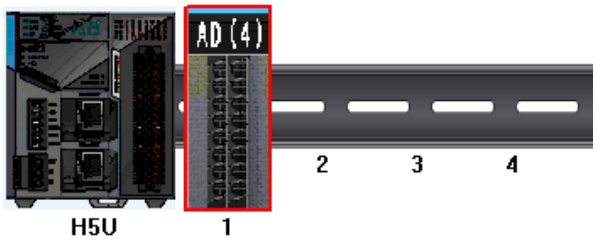
NOTE

- The port mapping of the H5U module is determined by the configuration. If there is no special configuration, the software arranges the port mapping in order. Even if the front-end module is deleted after the configuration is completed, the port mapping of the subsequent modules will not change.
- The relay output extension module can be connected to the relay or transistor main module. Similarly, the transistor input extension module can be connected to the transistor or relay main module.

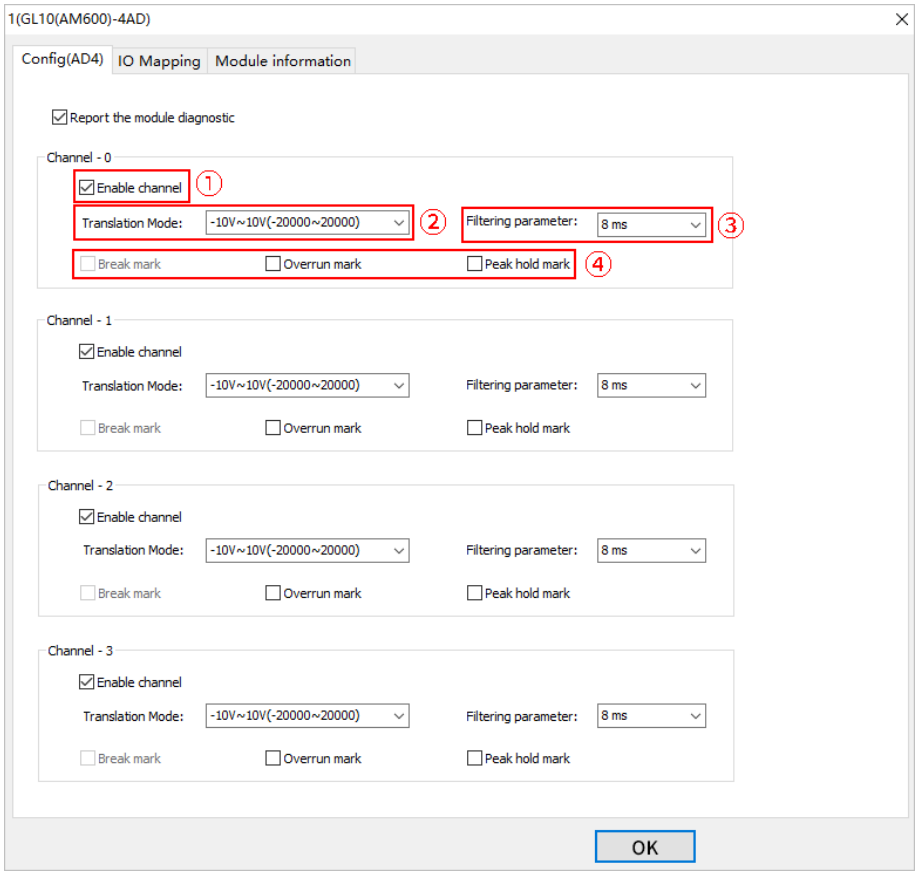
7.1.3.3 Configuring an Analog Input (AI) Module

The method of using the GL10-4AD AI module as a local extension module is as follows:

1. In the module list, select a module to be added, and double-click the module for automatic extension on the extension rack, or drag the module to the extension rack.



2. Double-click the GL10-4AD module on the rack. The page shown in the following figure is displayed.



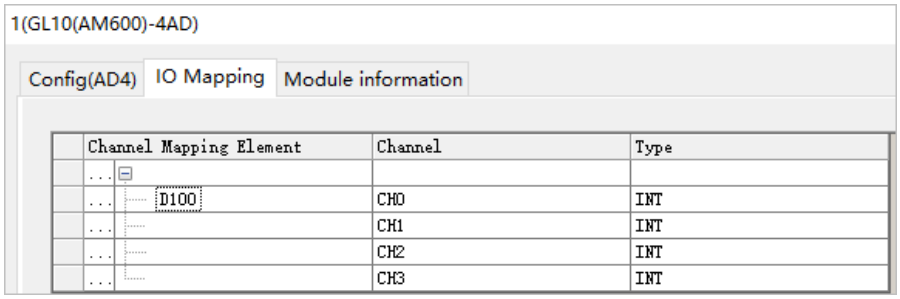
① Determine whether to select "Enable channel". If not, deselect it to save the scanning time.

② Select the corresponding span and resolution.

③ Set "Filtering parameter" to a value in the range of 1 ms to 255 ms.

④ Leave the auxiliary function items empty.

3. On the "IO Mapping" tab page, map CH0 of the 4AD module to the D element D100. In H5U, you can also map the module to a customized variable, as shown in the following figure.

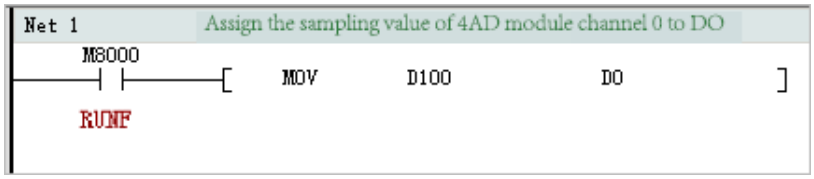


The following table lists the relationships between the mapped variables and actual input analog values.

Input Type	Rated Input Range	Rated Digital Value	Input Limit Range	Digital Value Limit
	-10 V to +10 V	-20000 to +20000	-11 V to +11 V	-22000 to +22000

Input Type	Rated Input Range	Rated Digital Value	Input Limit Range	Digital Value Limit
Analog voltage input	0 V to 10 V	0 to 20000	-0.5 V to +10.5 V	-1000 to +21000
	-5 V to +5 V	-20000 to +20000	-5.5 V to +5.5 V	-22000 to +22000
	0 V to 5 V	0 to 20000	-0.25 V to +5.25 V	-1000 to +21000
	1 V to 5 V	0 to 20000	0.8 V to 5.2 V	-1000 to +21000
Analog current input	-20 mA to +20 mA	-20000 to +20000	-22 mA to +22 mA	-22000 to +22000
	0 mA to 20 mA	0 to 20000	-1 mA to +21 mA	-1000 to +21000
	4 mA to 20 mA	0 to 20000	3.2 mA to 20.8 mA	-1000 to +21000

4. Use the LD programming language to program AD sampling, and assign the voltage sampling value of CH0 from D100 to D0, as shown in the following figure.

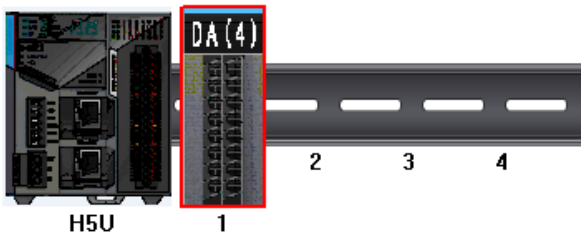


5. After successful compilation, download the project and run it.

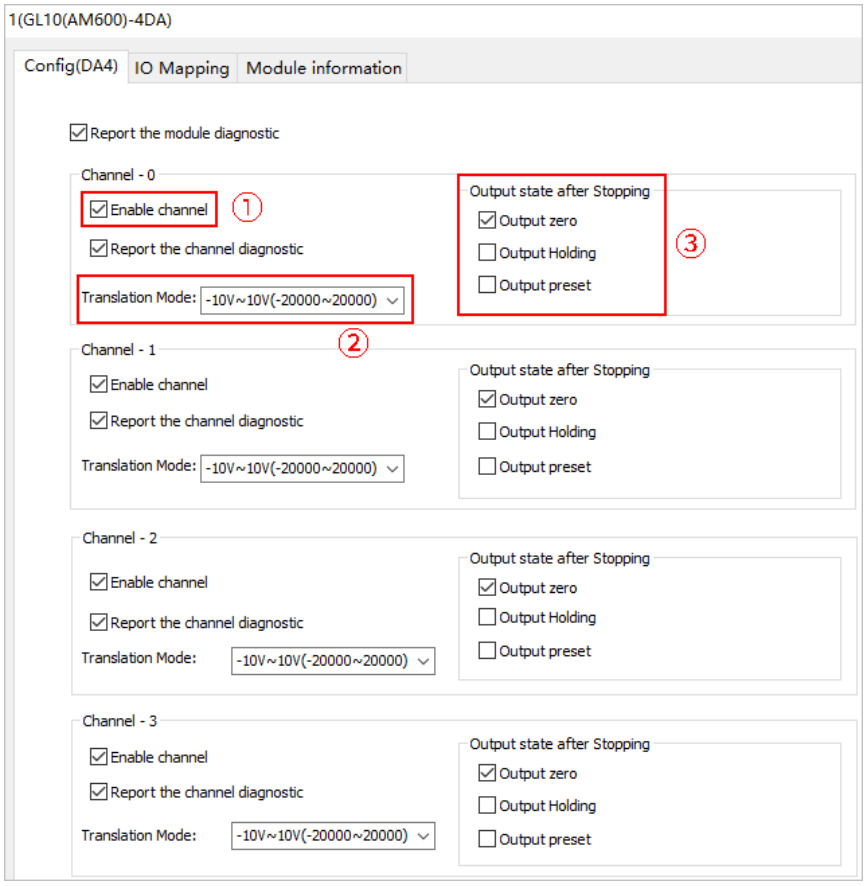
7.1.3.4 Configuring an Analog Output (AO) Module

The method of using the GL10-4DA AO module as a local extension module is as follows:

1. In the module list, select a module to be added, and double-click the module for automatic extension on the extension rack, or drag the module to the extension rack.



2. Double-click the GL10-4DA module on the rack. The "Config(DA4)" tab page is displayed, as shown in the following figure.

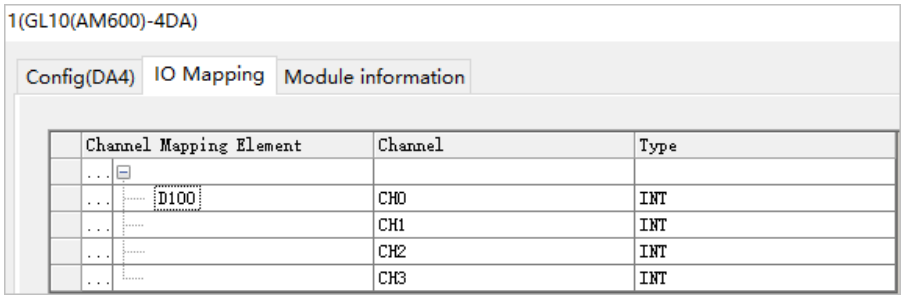


① Determine whether to select "Enable channel". If not, deselect it to save the scanning time.

② Set "Translation Mode" to select the output type and range.

③ Set "Output state after Stopping" to "Output zero", "Output Holding", or "Output preset" when the PLC is in the Stop state.

3. On the IO mapping page, map CH0 of the 4DA module to the D element D100. In H5U, you can also map the module to a customized variable.

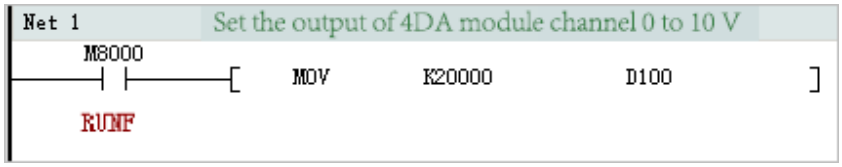


The following table lists the relationships between the mapped variables and actual output analog values.

Output type	Rated Output Range	Rated Digital Value	Output Limit Range	Digital Value Limit
Analog voltage output	-10 V to +10 V	-20000 to +20000	-11 V to +11 V	-22000 to +22000
	0 V to 10 V	0 to 20000	-0.5 V to +10.5 V	-1000 to +21000
	-5 V to +5 V	-20000 to +20000	-5.5 V to +5.5 V	-22000 to +22000
	0 V to 5 V	0 to 20000	-0.25 V to +5.25 V	-1000 to +21000

Output type	Rated Output Range	Rated Digital Value	Output Limit Range	Digital Value Limit
	1 V to 5 V	0 to 20000	0.8 V to 5.2 V	-1000 to +21000
Analog current output	0 mA to 20 mA	0 to 20000	0 mA to 21 mA	0 to 21000
	4 mA to 20 mA	0 to 20000	3.2 mA to 20.8 mA	-1000 to +21000

4. Use ladder graphic programming language to program DA output. As -10 V to +10 V corresponds to the digital of -20000 to +20000, assign 20000 to D0, and the output voltage of the module's channel 0 is +10 V.

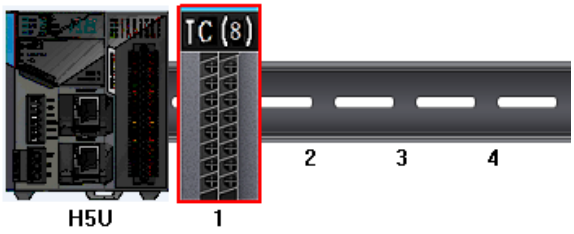


5. After successful compilation, download the project and run it.

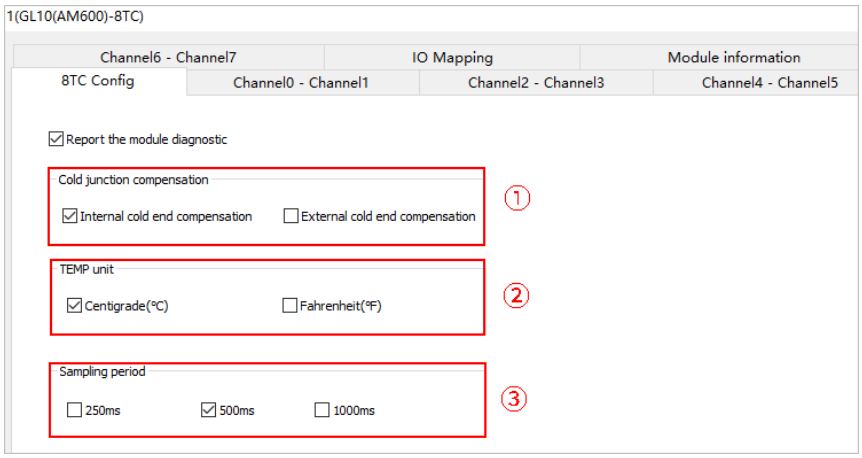
7.1.3.5 Configuring a Temperature Detection Module

Use H5U as the control master module, sample the temperature of the K thermocouple through CH0 of the GL10-8TC module, and assign the sampling value for the corresponding variable.

1. In the module list, select a module to be added, and double-click the module for automatic extension on the extension rack, or drag the module to the extension rack.



2. Double-click the GL10-8TC module on the rack. The "8TC Config" tab page is displayed, as shown in the following figure.

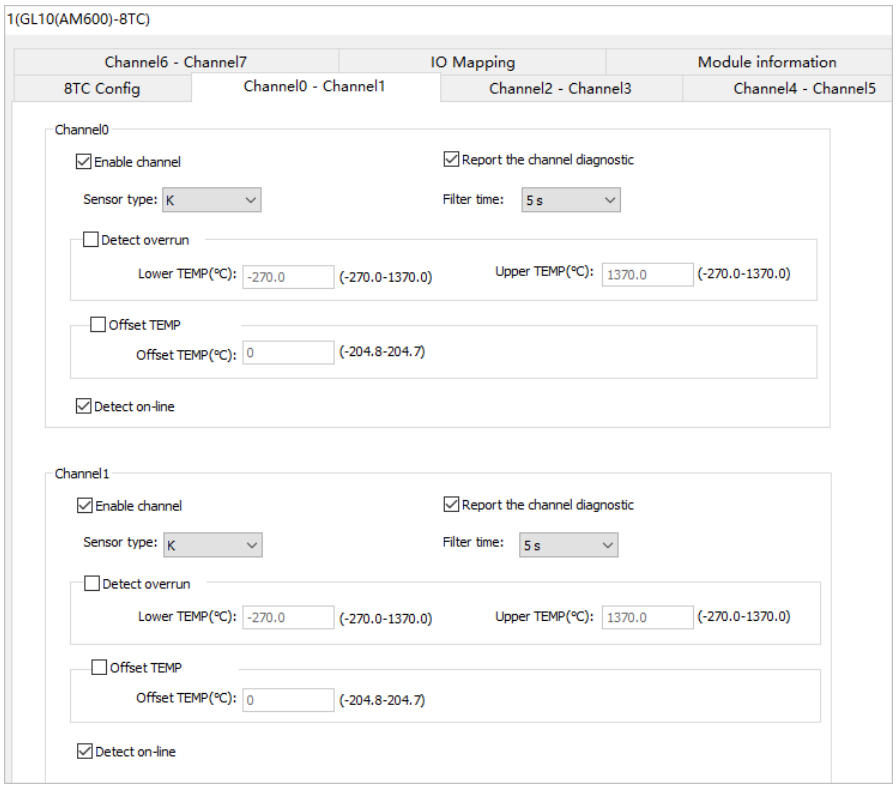


① To use external cold junction compensation, select "External cold end compensation". When external cold junction compensation is used, CH7 of the 8TC module is used for input of the external cold junction compensation sensor (PT100) but cannot be used to measure temperature for the thermocouple.

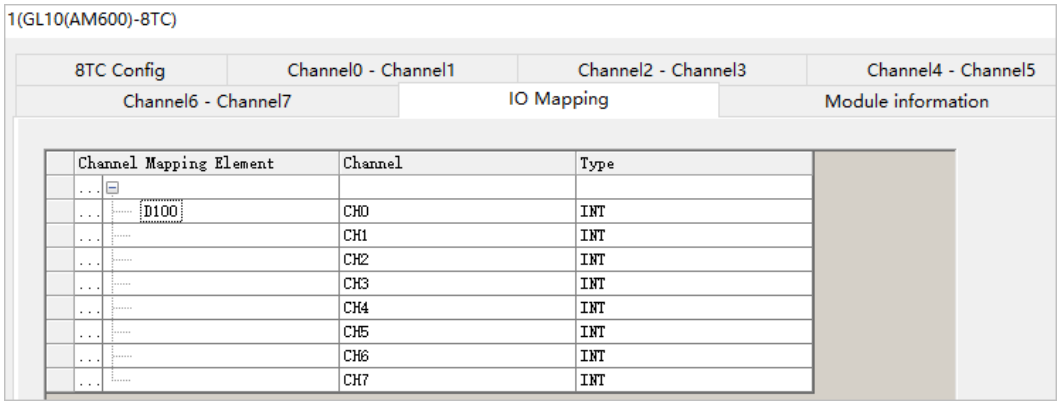
② Set "TEMP unit" to the corresponding temperature unit.

③ Set "Sampling period".

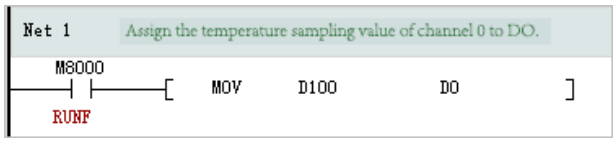
3. On the "Channel 0- Channel 1" tab page, select "Enable channel' for CH0 and set "Sensor type" to "K".



4. On the IO mapping page, map CH0 of the 8TC module to the D element D100. In H5U, you can also map the module to a customized variable.



5. Use the LD programming language to program 8TC sampling, and assign the temperature sampling value of CH0 from D100 to D0.



6. After successful compilation, download the project and run it.

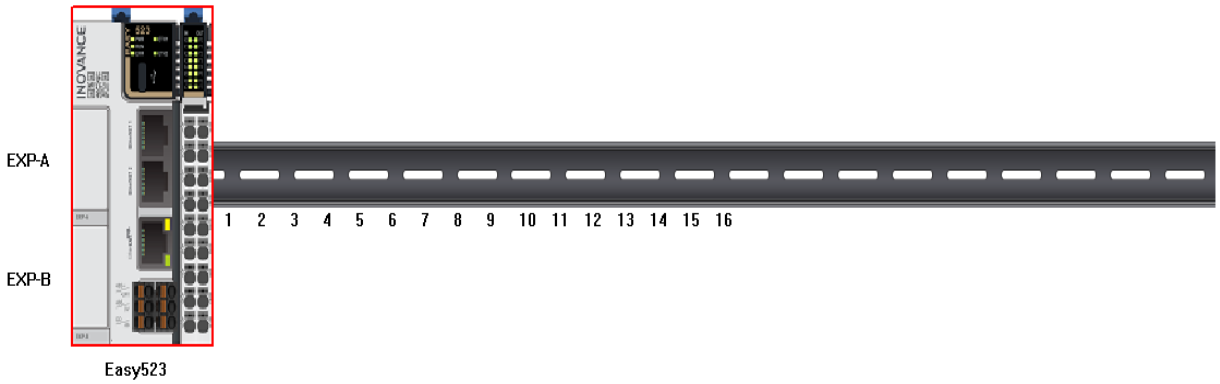
7.2 Easy Extension Module and Expansion Card Configuration

7.2.1 Overview

Easy extension modules

An Easy series host can carry up to 16 modules to access local modules based on module configuration.

The following figure shows how to connect an Easy series host to a local module.



The following table lists supported extension table.

Model	Description
GL20-1600END	16 DI module
GL20-0800END	8 DI module
GL20-3200END	32 DI module
GL20-0016ETN	16 DO module
GL20-0016ETP	16 DO module
GL20-0008ETN	8 DO module
GL20-0008ETP	8 DO module
GL20-0008ER	8 DO module
GL20-0032ETN	32 DO module
GL20-0808ETN	8 DI module and 8 DO module
GL20-3232ETN	32 DI module and 32 DO module
GL20-4AD	4 AI module
GL20-4DA	4 AO module
GL20-4PT	4-in resistance temperature detector (RTD) module
GL20-4TC	4-in thermocouple temperature detection (TC TEMP MEAS) module
GL20-2SCOM	2 serial communication module (configurable for RS232, RS422, and RS485)
GL20-2S485	2 RS485 serial communication module
GL20-2SSI	2 SSI module
GL20-8ADV	8-channel analog input module (voltage)

Model	Description
GL20-8ADI	8-channel analog input module (current)
GL20-0404ETP-5V	5 V power I/O module
GL20(GL20S)-0004ER	4-channel DO module
GL20(GL20S)-0004ETP-2A	4-channel DO module
GL20(GL20S)-PS2	Power supply module
GL20-2HC	2-channel high-speed count module
GL20(GL20S)-8DAV	A module with 8 DA terminals, supporting voltage output
GL20(GL20S)-8DAI	A module with 8 DA terminals, supporting current output

Easy extension cards

An Easy series host can carry up to 2 cards. Ten types of extension cards are supported. The following table lists extension slots supported by different extension cards.

No.	Name (Model)	Function	EXP-A Extension Card Slot	EXP-B Extension Card Slot
1	GE20-232/485-RTC	• Providing either the RS232 or RS485 extension interface • Providing RTC extension	Not supported	Supported
2	GE20-232/485	Providing either the RS232 or RS485 extension interface	Supported	Supported
3	GE20-CAN-485	Providing the CAN extension interface and RS485 extension interface	Supported	Not supported
4	GE20-2AD1DA-I	Providing 2-in AD and 1-out DA extension interface (current)	Supported	Supported
5	GE20-2AD1DA-V	Providing 2-in AD and 1-out DA extension interface (voltage)	Supported	Supported
6	GE20-4DI	Providing 4-in I/O extension interface	Supported	Supported
7	GE20-4DO-TN	Providing 4-out I/O extension interface	Supported	Supported

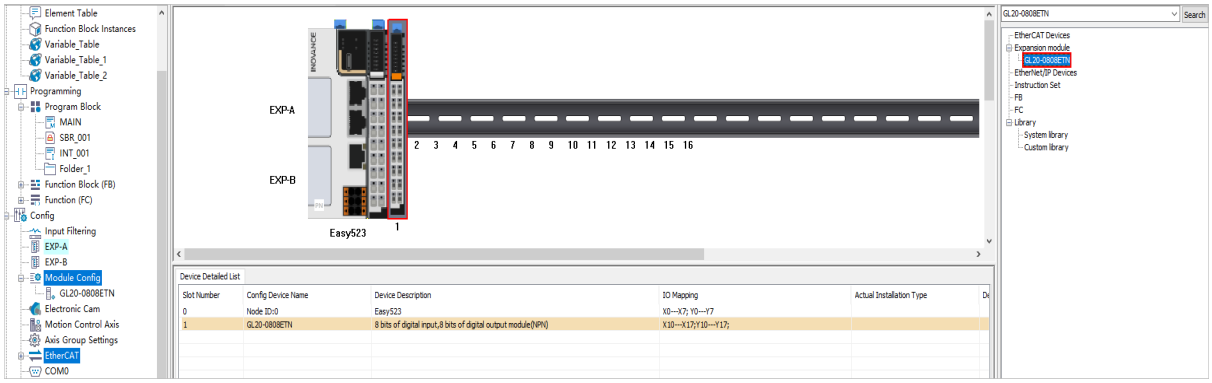
No.	Name (Model)	Function	EXP-A Extension Card Slot	EXP-B Extension Card Slot
8	GE20-RTC	Providing RTC extension	Not supported	Supported
9	GE20-TF	Providing the SD extension card interface	Not supported	Supported
10	GE20-TF-RTC	Providing RTC extension and SD card extension interface	Not supported	Supported

7.2.2 Extension Module Configuration

7.2.2.1 Hardware Configuration

Local extension modules are implemented by hardware configuration. To configure hardware, perform the following steps:

1. In the "Project Manager" tree on the left, double-click "Module Config". The "Module Config" page is displayed.
2. In the toolbox on the left, choose "Expansion module" > "Local module". Double-click the target module, which is GL20-0808ETN in this example. The module is automatically added to the corresponding configuration.



NOTE

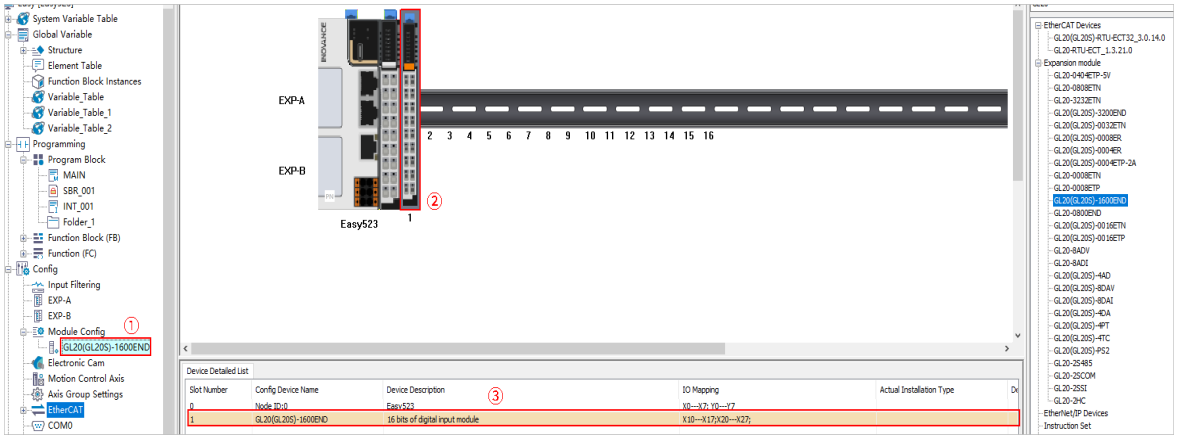
Versions V4.10.1.0 or later can detect the module power consumption. When the sum of the power consumption of added module exceeds the maximum allowed, a prompt is displayed. To view the power consumption of a module, double-click the module picture and then view the power consumption in "Device Information" in the dialog box displayed.

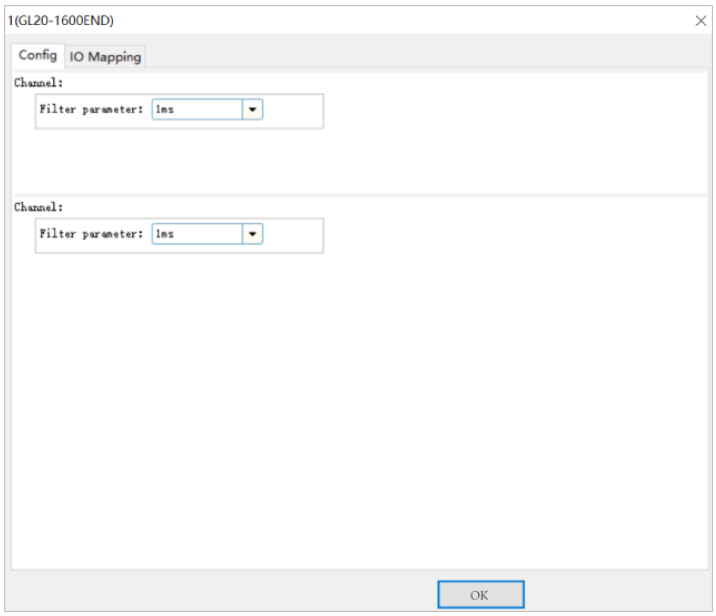
7.2.2.2 Configuring an Extension Module

7.2.2.2.1 Configuring a Digital Input (DI) Module

DI modules include GL20-1600END, GL20-0800END, and GL20-3200END. They are configured in the same way. The GL20-1600END series 16 DI module is used as an example. The method of using a DI module is as follows:

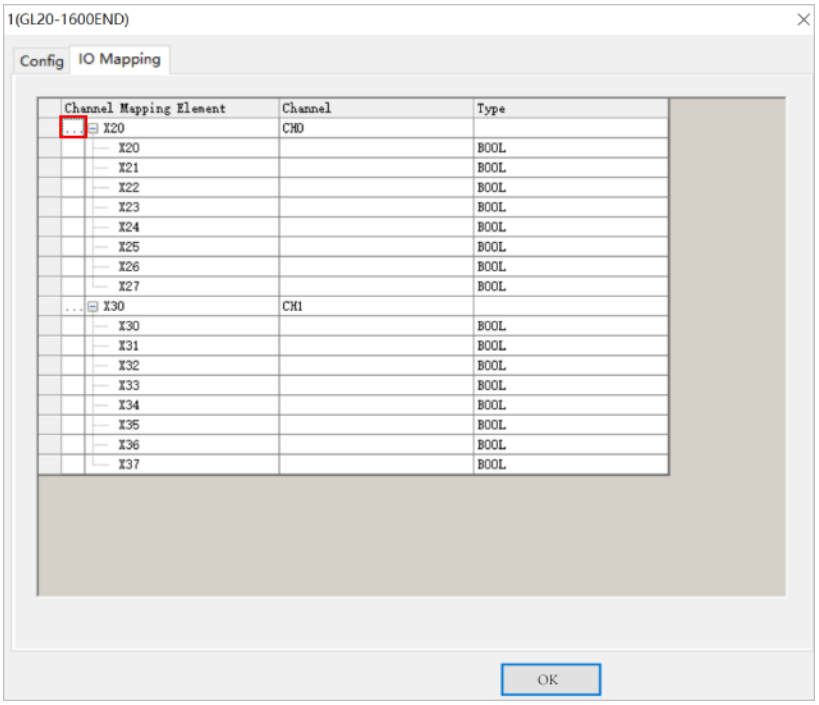
1. Access the configuration page and then the module configuration tab page in any of the following three ways, as shown in the following figure.
   - In the "Project Manager" tree on the left, unfold "Module Config", and then double-click "GL20-1600END".
   - On the configuration page, double-click the GL20-1600END module.
   - In the "Device Detailed List" section, double-click GL20-1600END.

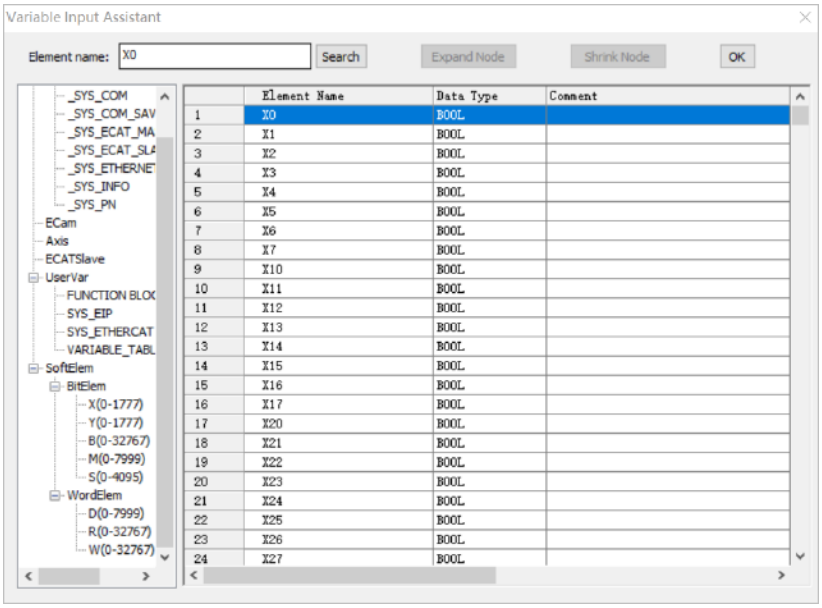




   - Configure module parameters.
     The GL20-1600END module has two channels that are configured in the same way.
     - On the "Config" page, set the filter parameters. You can select a filter time for filtering.
     - Click "IO Mapping" and configure the channel mapping element.
       After local DI extension modules are connected to the master module without port configuration, numbers of input X ports on the extension modules follow the number of the input X port on the master module in sequence. For example, if the extension module is GL20-1600END, and number of the last X port on the PLC host is X7 after module connection, numbers of 16 input X ports on the extension module range from X10 to X17 and X20 to X27 during programming. This also applies to other DI extension modules by analogy.

On the "IO Mapping" page shown in the following figure, click "...". In the "Variable Input Assistant" dialog box displayed, select the mapping element to be modified, and click "OK". Then, the mapping is modified to the corresponding element.





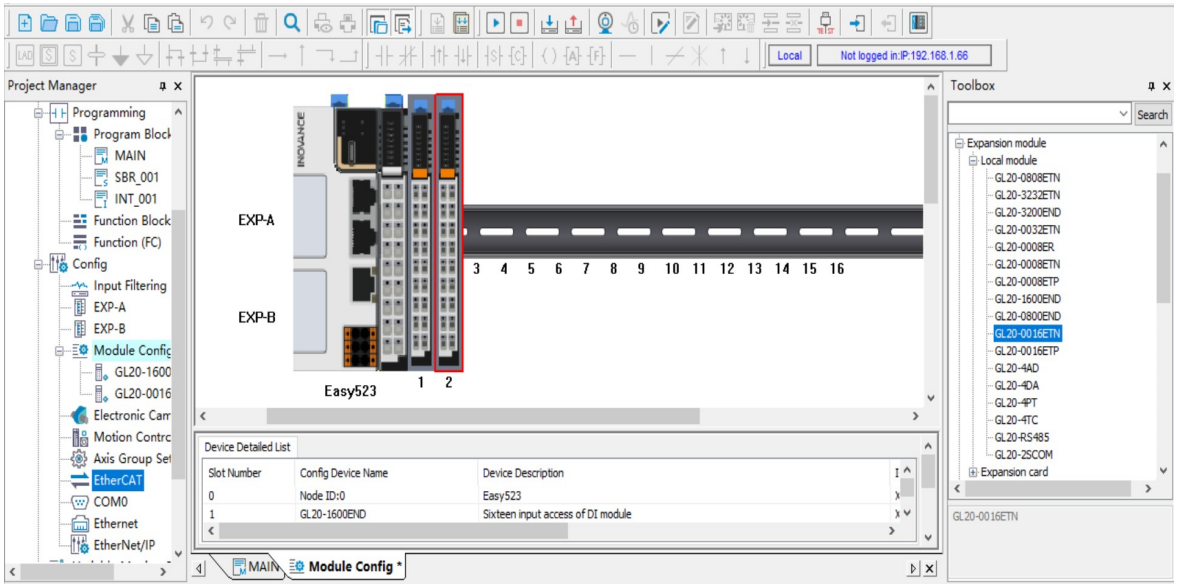
Click "OK".

7.2.2.2.2 Configuring a Digital Output (DO) Module

DO modules include GL20-0016ETN, GL20-0016ETP, GL20-0008ETN, GL20-0008ETP, GL20-0008ER, GL20-0032ETN, GL20(GL20S)-0004ER, and GL20(GL20S)-0004ETP-2A. They are configured in the same way. The GL20-0016ETN series is a 16-DO module and is used as an example. The method of using a DO module is as follows:

1. Access the configuration page and then the module configuration tab page in any of the following three ways, as shown in the following figure.
   - In the "Project Manager" tree on the left, unfold "Module Config", and then double-click "GL20-0016ETN".
   - On the configuration page, double-click the GL20-0016ETN module.

- In the "Device Detailed List" section, double-click GL20-0016ETN.



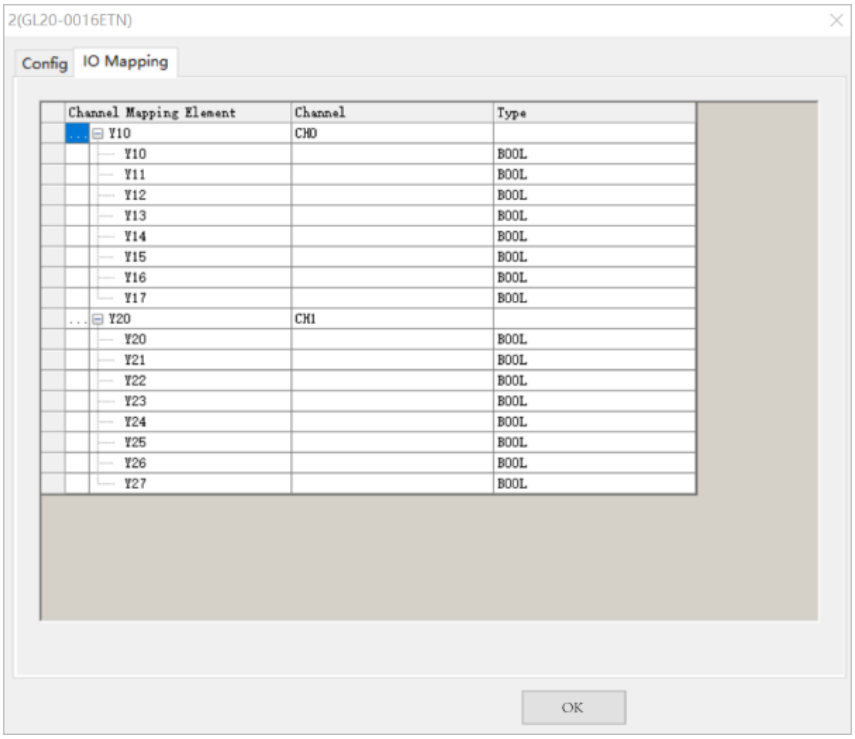
2. Configure module parameters.

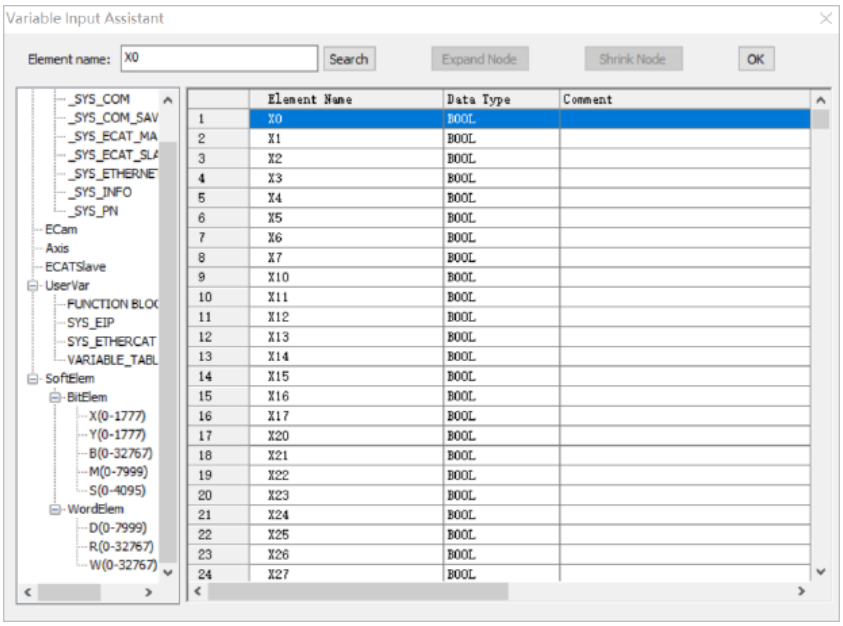
   The GL20-0016ETN module has two channels that are configured in the same way.

   - On the "Config" tab page, you can configure the disconnection and output stop modes for eight I/O elements. You can set or retain the output values of I/O ports when the PLC is disconnected or stopped.
   - Click "IO Mapping" and configure the channel mapping element.

     After local DO extension modules are connected to the master module without port configuration, numbers of output Y ports on the extension modules follow the number of the output Y port on the master module in sequence. For example, if the extension module is GL20-0016ETN, and number of the last Y port on the PLC host is Y7 after module connection, numbers of 16 output Y ports on the extension module range from Y10 to Y17 and Y20 to Y27 during programming. This also applies to other DO extension modules by analogy.

     On the "IO Mapping" page shown in the following figure, click "...". In the "Variable Input Assistant" dialog box displayed, select the mapping element to be modified, and click "OK". Then, the mapping is modified to the corresponding element.





Click "OK".

7.2.2.2.3 Configuring Digital Input and Output Modules

DI and DO modules include GL20-0808ETN and GL20-3232ETN. GL20-0808ETN provides two channels. One channel maps eight consecutive input I/O elements, and the other channel maps eight consecutive output I/O elements, which can be considered as combination of the DI and DO modules. This also applies to GL20-3232ETN.

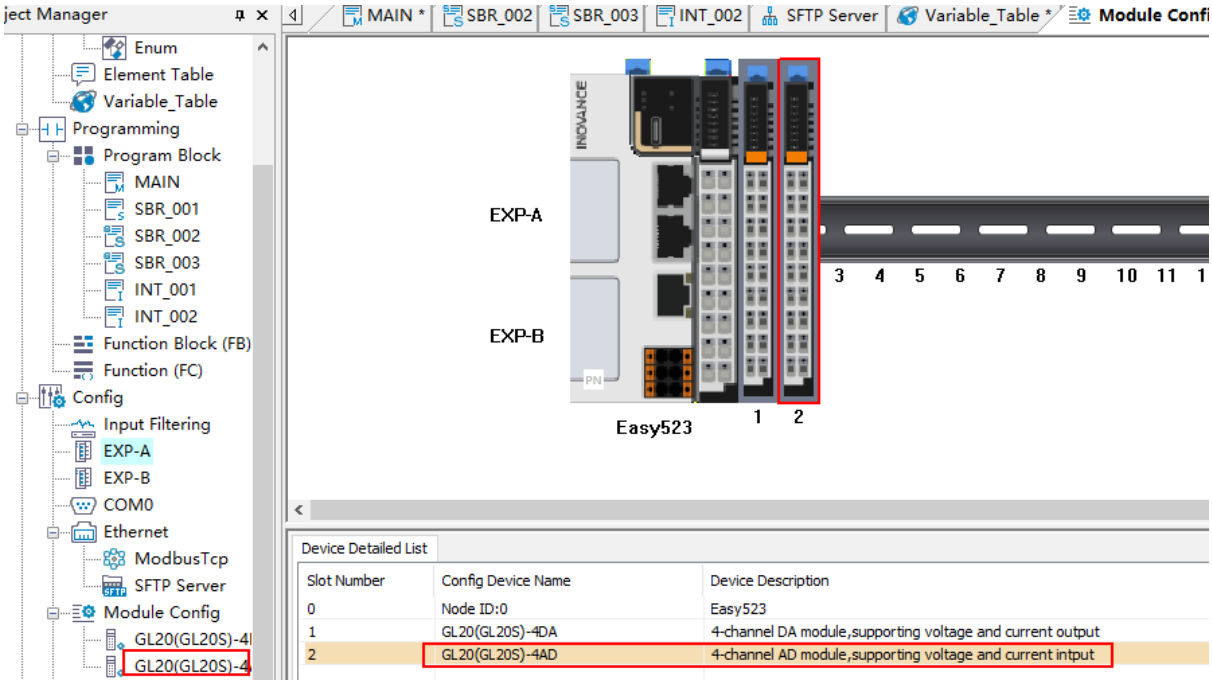
For details about configuring these terminals, see Configuring DI Modules or Configuring DO Modules.

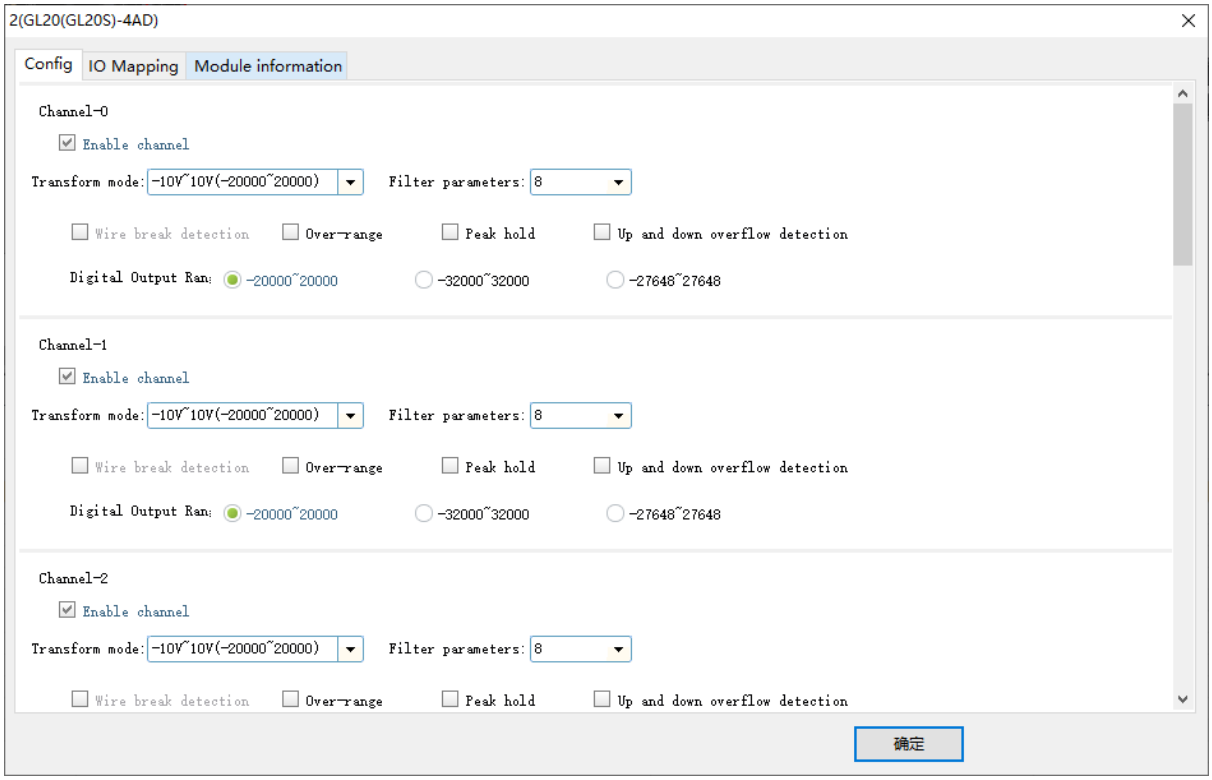
7.2.2.2.4 Configuring an Analog Input (AI) Module

> **NOTE**
>
> Software version: AutoShop V4.11.0.0 and later. Firmware version: V6.5.0.0 and later. When expanding analog modules for Easy500/Easy300/Easy Large Point products, a new "Digital Output Range" option is added.

AI modules include GL20(GL20S)-4AD, GL20-8ADV, and GL20-8ADI, which are configured in the same way. The following takes GL20(GL20S)-4AD as an example to describe how to configure an AI module.

1. Access the module configuration tab page in any of the following three ways, as shown in the following figure.
   - In the "Project Manager" tree on the left, unfold "Module Config", and then double-click "GL20(GL20S)-4AD".
   - On the configuration page, double-click the GL20(GL20S)-4AD module.
   - In the "Device Detailed List" section, double-click GL20(GL20S)-4AD.





2. Configure module parameters.

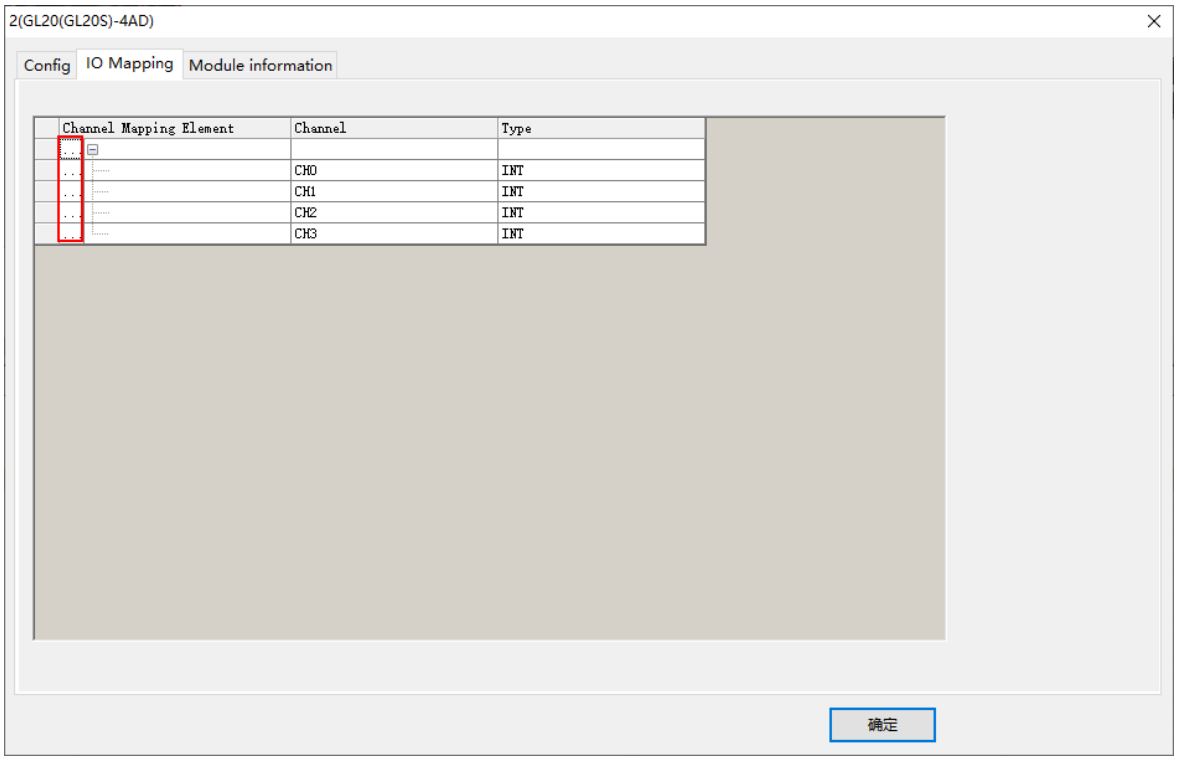
   The GL20(GL20S)-4AD module has four channels that are configured in the same way.

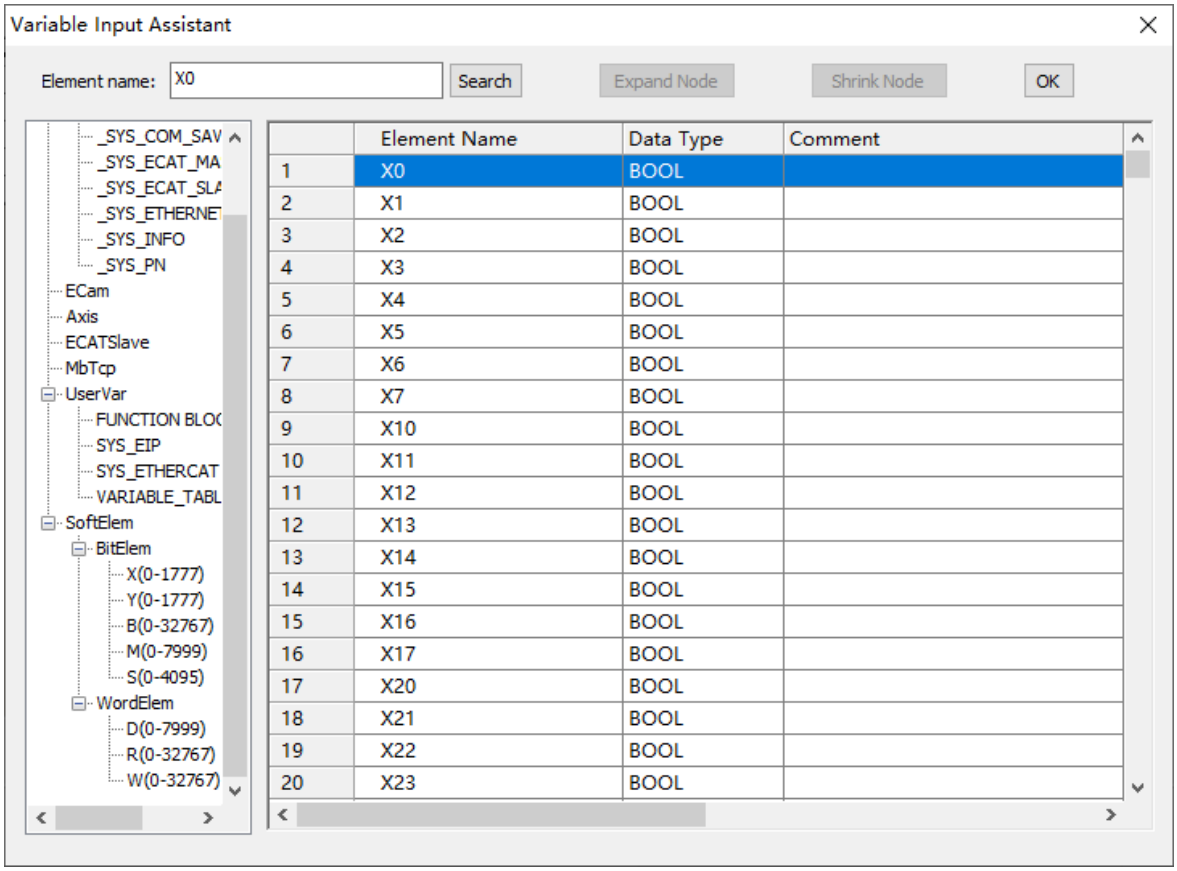
- Configure parameters on the "Config" page. The related parameters are as follows.

Parameter Name	Description
Enable channel	Determine whether to enable this channel. If not, deselect it to save the scanning time. It is selected by default.
Transform mode	Select the corresponding span and resolution.
filter parameters	Set the filter parameter, which ranges from 1 ms to 255 ms.
wire break detection	An alarm is reported when wire disconnection is detected. For models of 1 V to 5 V and 4 mA to 20 mA, this function is supported. This option is not enabled by default.
Over-range	An alarm is reported when the rated range is exceeded. Select this option to enable the function (excluded are 0 V to 10 V, 0 V to 5 V, 1 V to 5 V, 0 mA to 20 mA, and 4 mA to 20 mA. When the value is below -200, the system considers that the value is below the lower limit. When the value is above 20000, the system considers that the value is above the upper limit.) This function is not enabled by default.
peak hold	The maximum value currently collected will not be updated if the later value does not exceed the current value. This function is not enabled by default.
Up and down overflow	After this option is selected, an alarm is reported when temperature measurement range is exceeded. This option is not selected by default.
Digital output range	

Parameter Name	Description
	Sets the analog output range, which determines the range of the values output by this channel for conversion. The available options in the "Conversion Mode" drop-down list correspond to the ranges set by the "Digital output range".

- Click "IO Mapping" and configure the channel mapping element.
  The GL20(GL20S)-4AD module supports consecutive mapping and separate mapping. You can click "..." at a parent node to map elements or variables consecutively or click "..." before a channel to map a single element or variable. In the Variable Input Assistant" dialog box displayed, select the mapping element to be modified and click "OK". Then, the mapping of the corresponding element is modified.





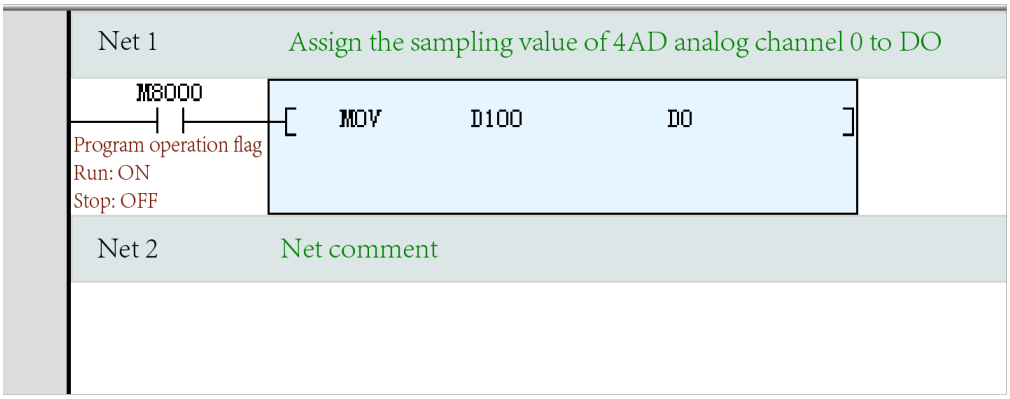
On the "IO Mapping" tab page, map CH0 of the 4AD module to the D element D100. In an Easy series host, you can map it to a customized variable. The following table lists the relationships between the mapped variables and actual input analog values. The EtherCAT bus communication port module supports conversion of three digital spans, including -20000 to +20000, -32000 to +32000, and -27648 to +27648, while the local bus only supports conversion of the span -20000 to +20000.

Input Type	Rated Input Range	Rated Digital Value	Input Limit Range	Digital Value Limit
Analog voltage input	-10 V to +10 V	-20000 to +20000; -32000 to +32000; -27648 to +27648	-11 V to +11 V	-20400 to +20400; -32640 to +32640; -28200 to +28200
	0 V to 10 V	0 to 20000; 0 to 32000; 0 to 27648	-0.5 V to +10.5 V	-1000 to +20400; -1600 to +32640; -1382 to +28200
	-5 V to +5 V	-20000 to +20000; -32000 to +32000; -27648 to +27648	-5.5 V to +5.5 V	-20400 to +20400; -32640 to +32640; -28200 to +28200
	0 V to 5 V	0 to 20000; 0 to 32000; 0 to 27648	-0.25 V to +5.25 V	-1000 to +20400; -1600 to +32640; -1382 to +28200
	1 V to 5 V	0 to 20000; 0 to 32000; 0 to 27648	0.8 V to 5.2 V	-1000 to +20400; -1600 to +32640; -1382 to +28200

Input Type	Rated Input Range	Rated Digital Value	Input Limit Range	Digital Value Limit
Analog current input	-20 mA to +20 mA	-20000 to +20000; -32000 to +32000; -27648 to +27648	-22 mA to +22 mA	-20400 to +20400; -32640 to +32640; -28200 to +28200
	0 mA to 20 mA	0 to 20000; 0 to 32000; 0 to 27648	-1 mA to +21 mA	-1000 to +20400; -1600 to +32640; -1382 to +28200
	4 mA to 20 mA	0 to 20000; 0 to 32000; 0 to 27648	3.2 mA to 20.8 mA	-1000 to +20400; -1600 to +32640; -1382 to +28200

Click "OK".

3. Use the LD programming language to program AD sampling, and assign the voltage sampling value of CH0 from D100 to D0, which can be compiled and downloaded for running.



7.2.2.2.5 Configuring an Analog Output (AO) Module

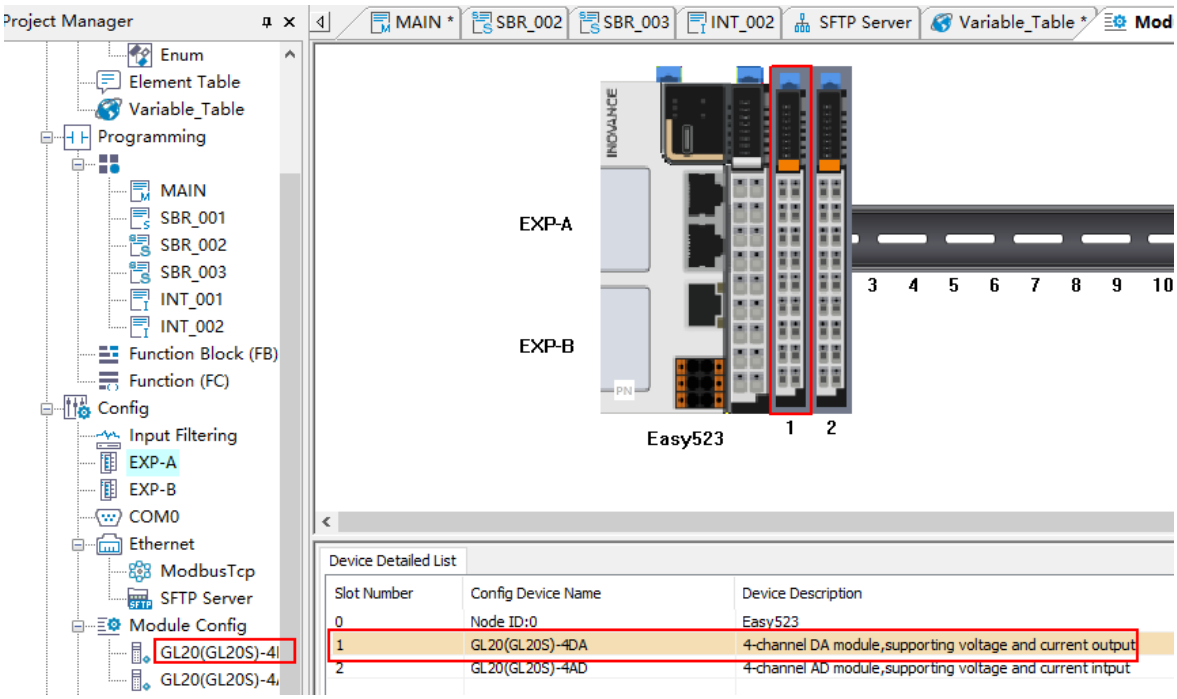
NOTE

Software version: AutoShop V4.11.0.0 and later. Firmware version: V6.5.0.0 and later. When expanding analog modules for Easy500/Easy300/Easy Large Point products, a new "Digital Output Range" option is added.

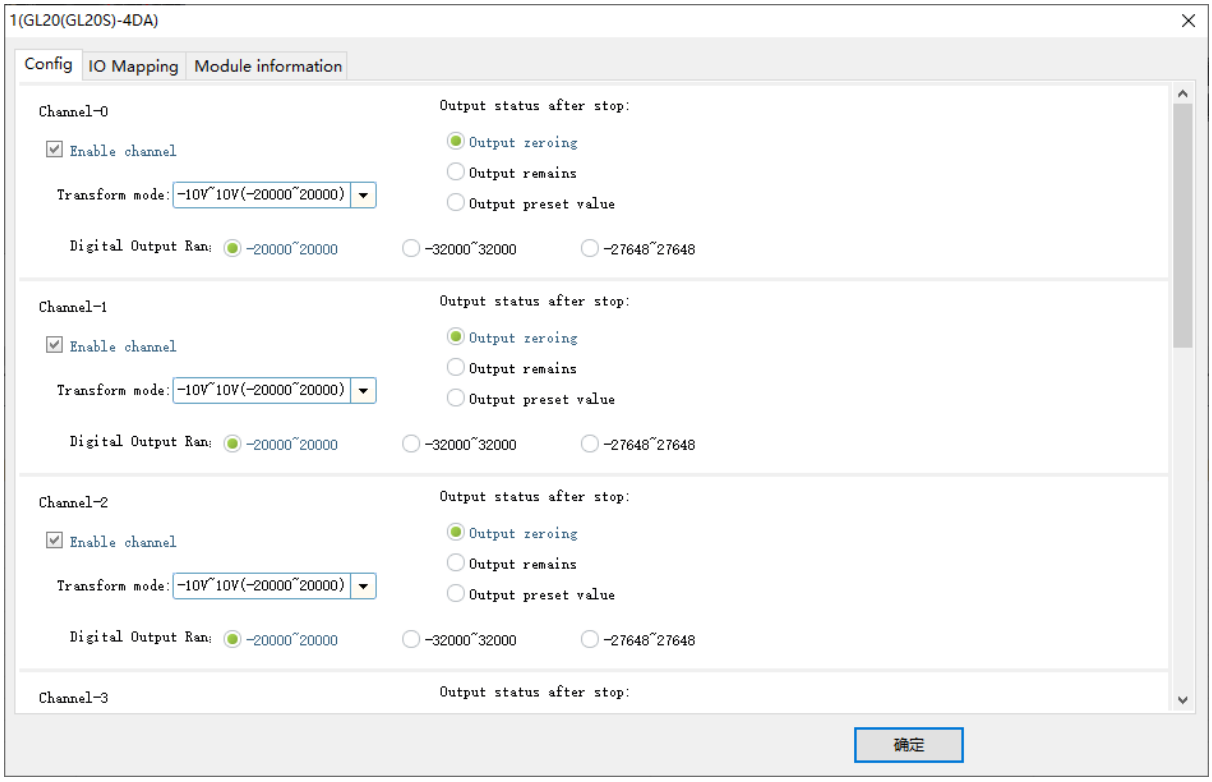
AI modules include GL20(GL20S)-4DA, GL20(GL20S)-8DAV, and GL20(GL20S)-8DAI, which are configured in the same way. The following takes GL20(GL20S)-4DA as an example to describe how to configure an AI module.

1. Access the module configuration tab page in any of the following three ways, as shown in the following figure.
   - In the "Project Manager" tree on the left, unfold "Module Config", and then double-click "GL20(GL20S)-4DA".
   - On the configuration page, double-click the GL20(GL20S)-4DA module.

- In the "Device Detailed List" section, double-click GL20(GL20S)-4DA.



2. Configure parameters on the "Config" page.The GL20(GL20S)-4DA module has four channels that are configured in the same way.



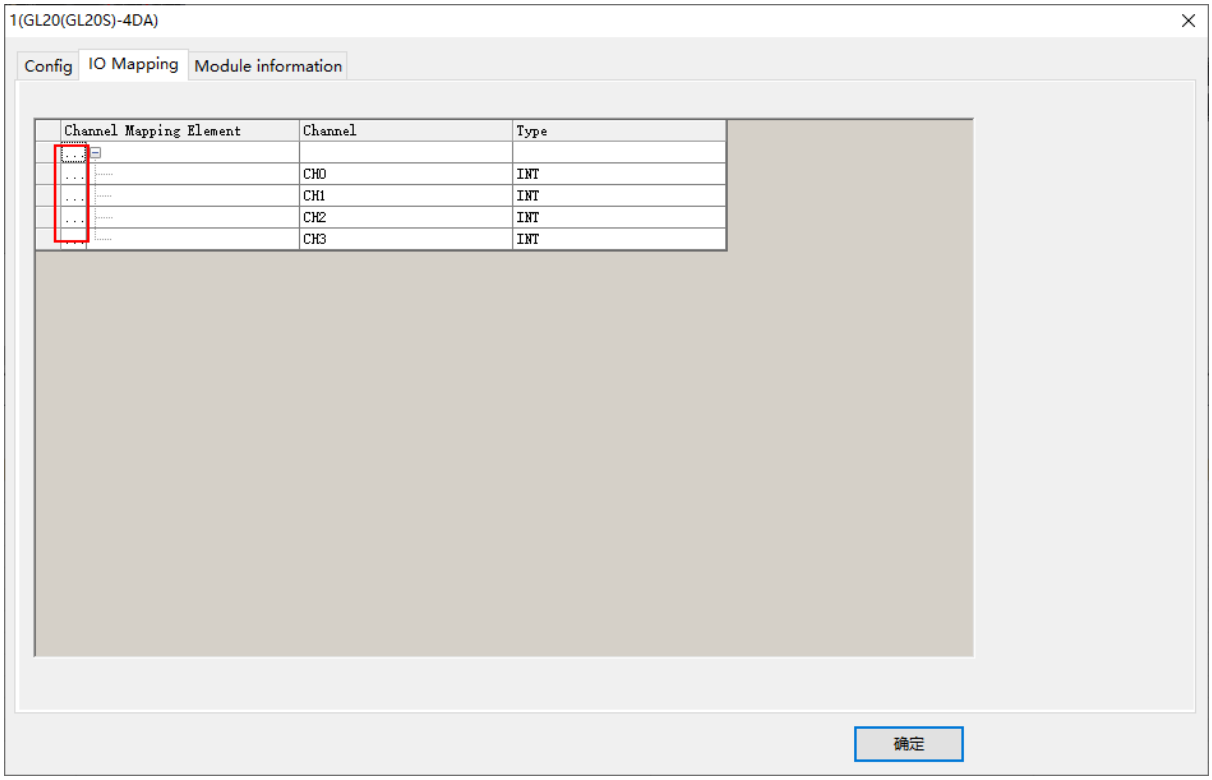
The related parameters are as follows.

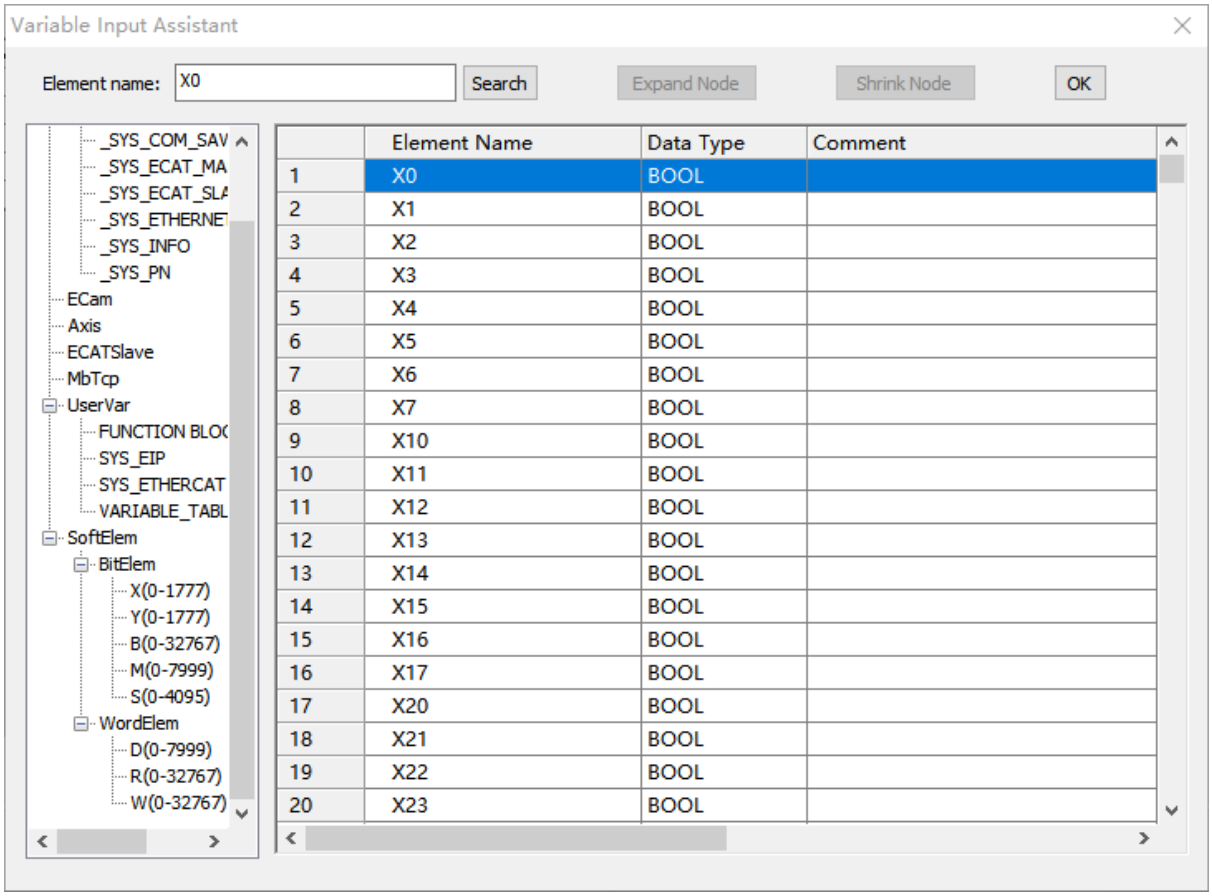
Parameter Name	Description
Enable channel	Determine whether to select "Enable channel". If not, deselect it to save the scanning time. "Enable channel" is selected by default.
Transform mode	Select the output type and range.

Parameter Name	Description
output status after stop	Indicate the module output status when the PLC stops. output clear: The output value is cleared. output hold: The original output value is retained. output preset value: The preset value is output.
Digital output range	Sets the analog output range, which determines the range of the values output by this channel for conversion. The available options in the "Conversion Mode" drop-down list correspond to the ranges set by the "Digital output range".

Click "IO Mapping" and configure the channel mapping element.

The GL20(GL20S)-4DA module supports consecutive mapping and separate mapping. You can click "..." at a parent node to map elements or variables consecutively or click "..." before a channel to map a single element or variable. In the Variable Input Assistant" dialog box displayed, select the mapping element to be modified and click "OK". Then, the mapping of the corresponding element is modified.





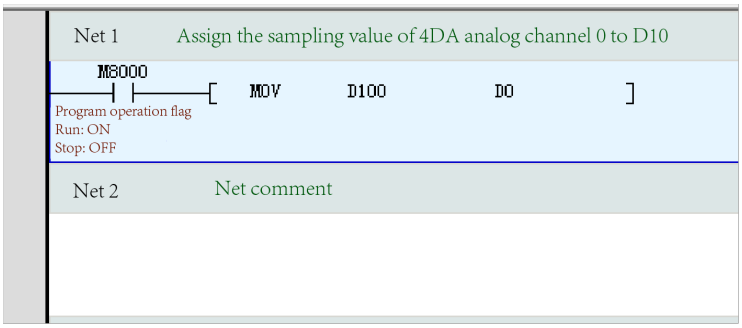
On the "IO Mapping" tab page, map CH0 of the GL20(GL20S)-4DA module to the D element D200. In an Easy series host, you can map it to a customized variable. The following table lists the relationships between the mapped variables and actual output analog values. The EtherCAT bus communication port module supports conversion of three digital spans, including -20000 to +20000, -32000 to +32000, and -27648 to +27648, while the local bus only supports conversion of the span -20000 to +20000.

Output Type	Rated Output Range	Rated Digital Value	Output Limit Range	Digital Value Limit
Analog Voltage Output	-10 V to +10 V	-20000 to +20000; -32000 to +32000; -27648 to +27648	-11 V to +11 V	-22000 to +22000; -32767 to +32767; -29030 to +29030
	0 V to 10 V	0 to 20000; 0 to 32000; 0 to 27648	-0.5 V to +10.5 V	-1000 to +21000; -1600 to +32767; -1382 to +29030
	-5 V to +5 V	-20000 to +20000; -32000 to +32000; -27648 to +27648	-5.5 V to +5.5 V	-22000 to +22000; -32767 to +32767; -29030 to +29030
	0 V to 5 V	0 to 20000; 0 to 32000; 0 to 27648	-0.25 V to +5.25 V	-1000 to +21000; -1600 to +32767; -1382 to +29030
	1 V to 5 V	0 to 20000; 0 to 32000; 0 to 27648	0.8 V to 5.2 V	-1000 to +21000; -1600 to +32767; -1382 to +29030

Output Type	Rated Output Range	Rated Digital Value	Output Limit Range	Digital Value Limit
Analog current output	0 mA to 20 mA	0 to 20000; 0 to 32000; 0 to 27648	0 mA to 21 mA	0 to 21000; 0 to 32767; 0 to 29030
	4 mA to 20 mA	0 to 20000; 0 to 32000; 0 to 27648	3.2 mA to 20.8 mA	-1000 to +21000; -1600 to +32767; -1382 to +29030

Click "OK". Configure module parameters.

3. Use the LD programming language to program DA sampling, and assign the voltage sampling value of CH0 from D200 to D10, which can be compiled and downloaded for running.

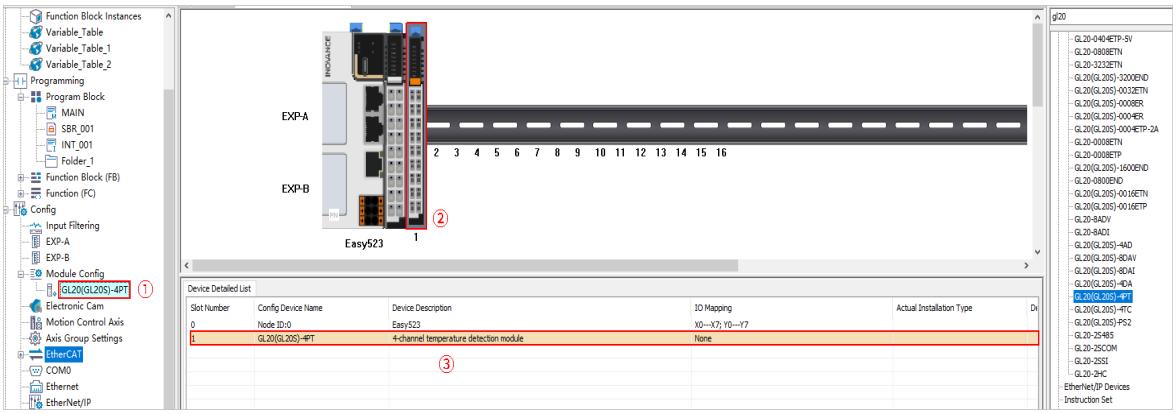


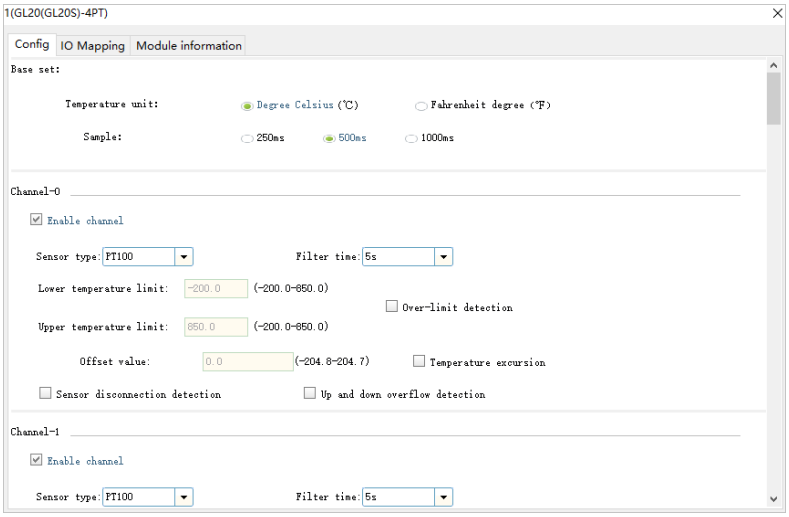
7.2.2.2.6 Configuring a Temperature Detection Module

Temperature detection modules include GL20-4PT and GL20-4TC, which are configured in the same way. The following takes GL20-4PT as an example to describe how to configure a temperature detection module.

1. Access the configuration page and then the module configuration tab page in any of the following three ways, as shown in the following figure.
   - In the "Project Manager" tree on the left, unfold "Module Config", and then double-click "GL20-4PT".
   - On the configuration page, double-click the GL20-4PT module.

- In the "Device Detailed List" section, double-click GL20-4PT.





2. Configure module parameters.

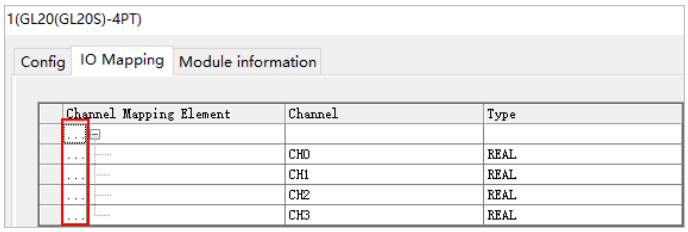
   The GL20-4PT module has four channels that are configured in the same way.

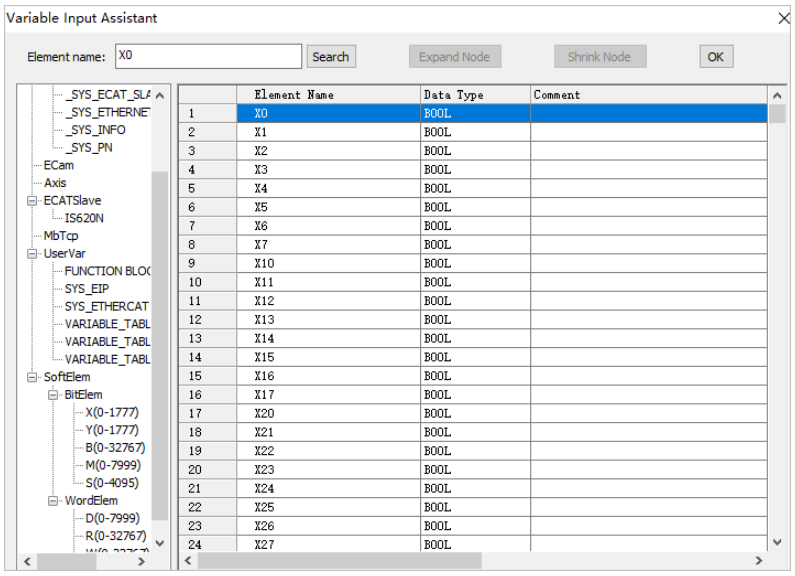
- Configure parameters on the "Config" page. The related parameters are as follows.

Parameter Name	Description
TEMP unit	The temperature unit options are centigrade (°C) and Fahrenheit (°F). Select one as needed.
Sampling period	Options are 250 ms, 500 ms, and 1000 ms. Select one as needed.
Enable channel	Determine whether to enable this channel. If not, deselect it to save the scanning time. It is selected by default.
Sensor type	-
Filter time	-
Over-limit detection	After this option is selected, an alarm is reported when the temperature exceeds the upper or lower limit. This option is not selected by default.
Offset TEMP	The temperature offset. This option is not selected by default.
Detect on-line	After this option is selected, an alarm is reported when the sensor is disconnected. This option is not selected by default.

Parameter Name	Description
Up and down overflow detection	After this option is selected, an alarm is reported when temperature measurement range is exceeded. This option is not selected by default.

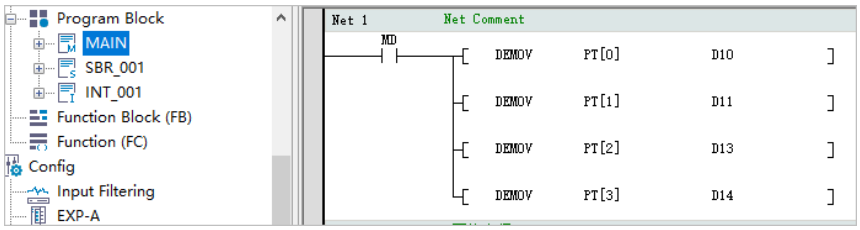
- Click "IO Mapping" and configure the channel mapping element.
  Each channel of the GL20-4PT module can map soft elements or REAL variables consecutively or separately. You can click "..." at a parent node to map elements or variables consecutively or click "..." before a channel to map a single element or variable. In the "Variable Input Assistant" dialog box displayed, select the mapping element to be modified and click "OK". Then, the mapping of the corresponding element is modified.





Click "OK".

3. Use the LD programming language for sampling programming on the temperature of the GL20-4PT module. Map CH0 to CH4 to REAL array PTs in sequence, use D10 to D14 to collect the temperature, and compile, download, and run the program.



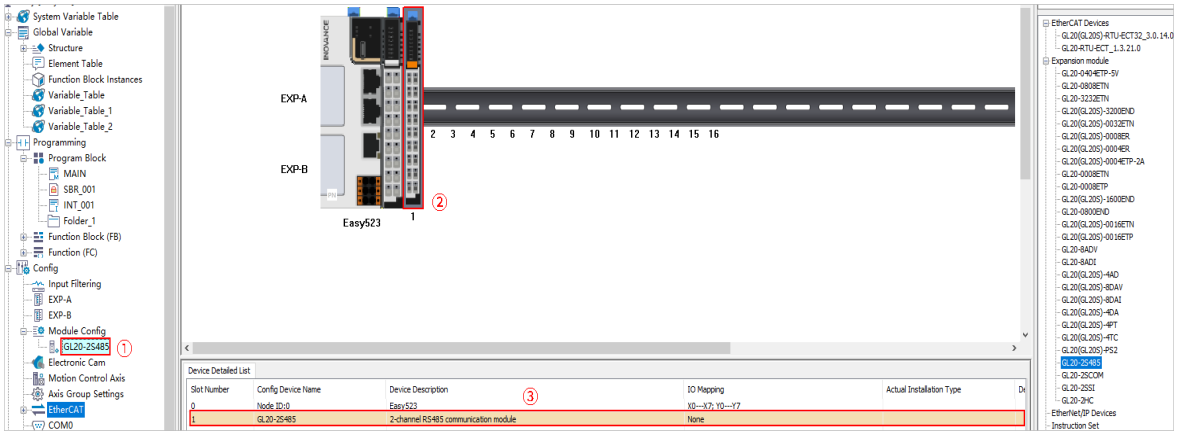
7.2.2.2.7 Configuring a Communication Module

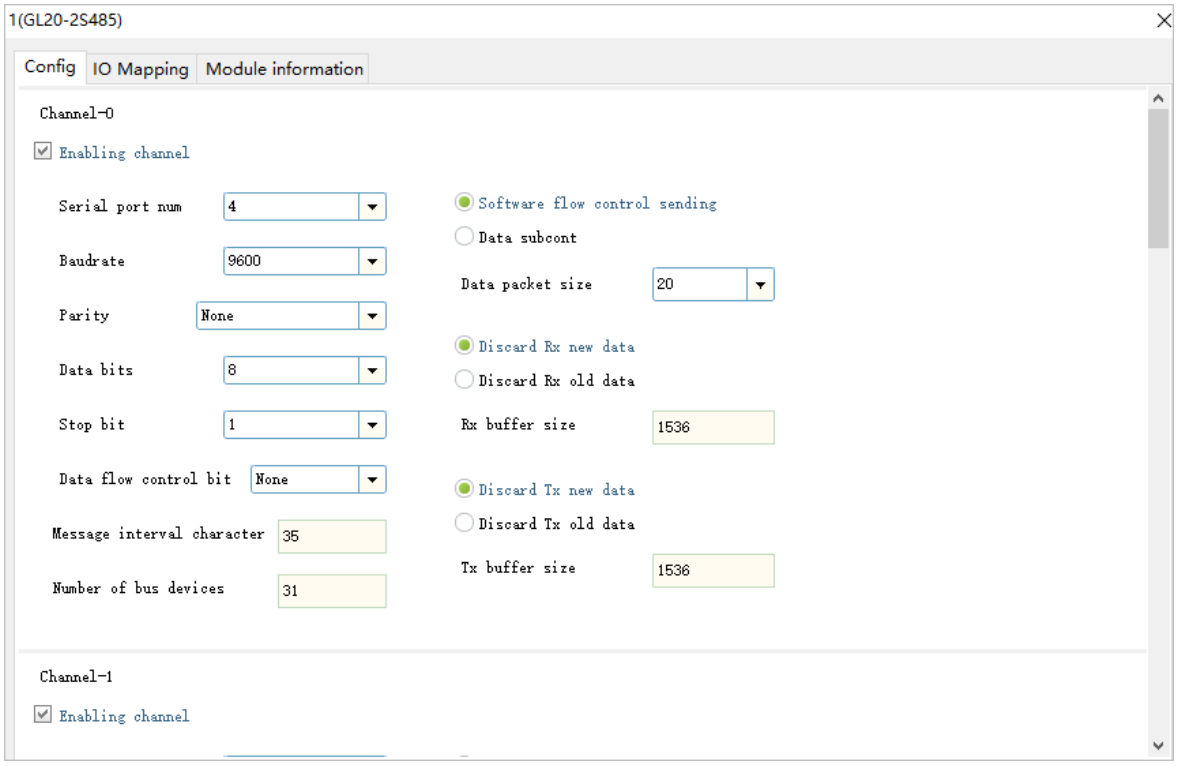
Communication modules include GL20-2SCOM 、 GL20-2S485 and GL20-2CAN, which must be installed in the first four extension slots and are configured in the same way. The following takes GL20-2S485 as an example to describe how to configure a communication module.

NOTE

- Modules GL20-2SCOM and GL20-2S485 must be installed in the first four slots of the expansion slots.
- Notes for configuring the GL20-2CAN module:
  - The module must be installed in the first two slots.
  - Mounting the GL20-2CAN module in local module configuration will affect the response time of other modules.
  - Only the Easy52X model is supported.

1. Access the configuration page and then the module configuration tab page in any of the following three ways, as shown in the following figure.
   - In the "Project Manager" tree on the left, unfold "Module Config", and then double-click "GL20-2S485".
   - On the configuration page, double-click the GL20-2S485 module.
   - In the "Device Detailed List" section, double-click GL20-2S485.





2. Configure module parameters.

   The GL20-2S485 module has two channels that are configured in the same way.

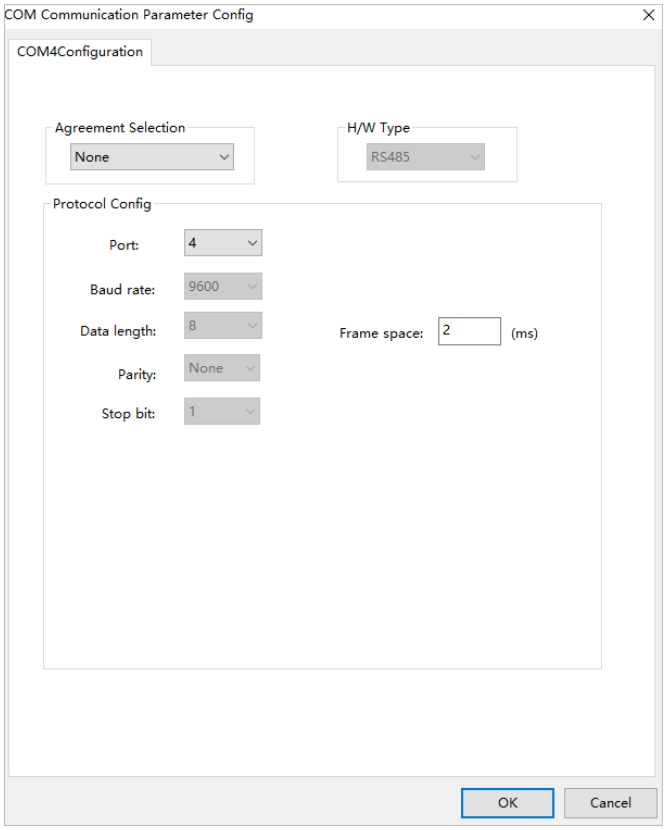
On the "Config" page, configure relevant parameters. Each channel contains basic serial port parameter settings and data transmission settings. In the parameters, "Enabling channel" indicates whether the current channel parameters are enabled; Some parameters that are unavailable, such as "Serial port num" and "Data bits", indicate that these parameters cannot be set on the page. The following table lists the module parameters.

Parameter Name	Description
Serial port num	The number of the serial port for communication connection. It can only be modified by configuring the COM port communication parameters. For details, see step 3.
Baudrate	The communication baud rate. Options are 9600, 19200, 38400, 57600, and 115200. Default: 9600
Parity	The method of verifying communication frames. Options are "None", "Odd", and "Even". Default: Even
Data bits	The actual data bits included in communication frames. It can only be modified by configuring the COM port communication parameters. For details, see step 3.
Stop bit	The last bit in a single message during communication. Options are 1 and 2. Default: 1
Data flow control bit	The bit controlling data flow transmission during communication. Options are "None", "Receive only", and "Send only". Default: None
Hardware Type	The type of hardware of other devices communicating with the module. Options are RS232, RS485, and RS422.
Message interval character	The minimum idle character between two messages. Default: 35
Data packet size	When "Data subcont" is enabled, this parameter is valid and the data packets are sent in this size.
Number of bus devices	The maximum number of devices on the RS485 bus. Default: 31 Note: This parameter is only available for the GL20-2S485 module.

Parameter Name	Description
Software flow control sending	According to the frame length, the packet is automatically packetized and sent. "Software flow control sending" is selected by default.
Data subcont	Packets are sent in fixed length that is set in "Data packet size".
Discard Rx new data	When the receiving cache is full, discard the newly received data. "Discard Rx new data" is selected by default.
Discard Rx old data	When the receiving cache is full, discard the old data at the head of the cache queue.
Discard Tx new data	When the sending cache is full, discard the newly sent data. "Discard Tx new data" is selected by default.
Discard Tx old data	When the sending cache is full, discard the old data at the head of the cache queue.

3. Configure communication parameters of the COM port.
   After the module is added, two COM ports (ranging from 4 to 15) are generated in the "Project Manager" tree by default. As shown in the figure of Step 1, COM4 and COM5 are added, corresponding to channel 0 and channel 1.

   a. The following takes COM4 as an example to describe how to configure communication parameters. In the "Project Manager" tree, double-click "COM4". The "COM Communication Parameter Config" dialog box is displayed.



b. Set the communication protocol and basic serial port parameters (which are serial port number and data bits here).

Parameter Name	Description
Protocol selection	Agreement Selection Options are None, Free Protocol, Modbus-RTU Master, and Modbus-RTU Slave.
H/W Type	The type of hardware of other devices communicating with the module. Options are RS232, RS485, and RS422.
Port	The number of the serial port for communication connection.
Baud rate	The communication baud rate. Options are 9600, 19200, 38400, 57600, and 115200. Default: 9600
Data length	The actual data bits included in communication frames.
Parity	The method of verifying communication frames. Options are "None", "Odd", and "Even". Default: Even
Stop bit	The last bit in a single message during communication. Options are 1 and 2. Default: 1
Station No.	The number of the Modbus-RTU slave. Note: This parameter is available only when "Modbus-RTU" is selected for "Agreement Selection".
Frame interval	The interval between receiving a data frame and sending the next data frame.

c. Click "OK".

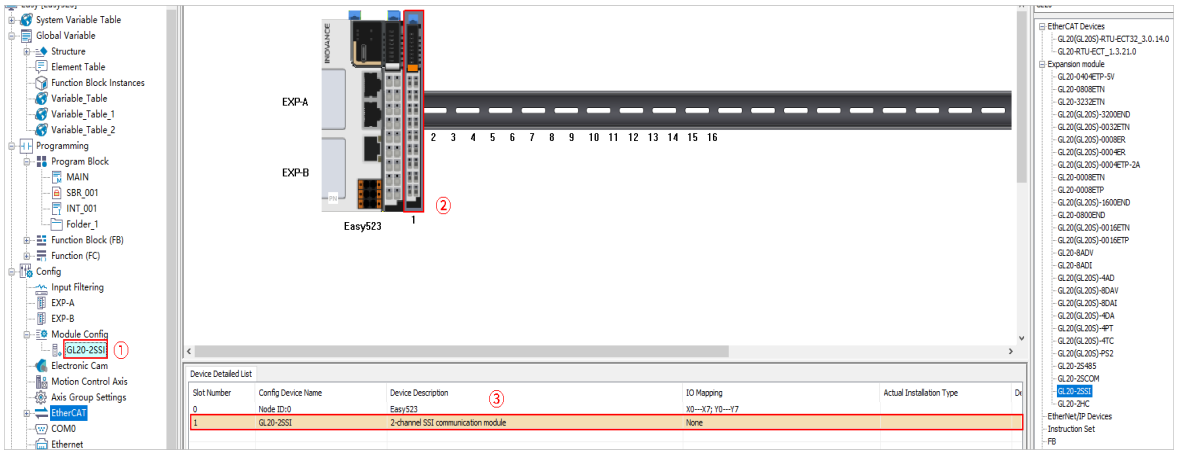
4. (Optional) when "Modbus-RTU" is selected for "Agreement Selection", you need to make Modbus configuration. In the "Project Manager" tree, right-click "COM x", and select "Add Modbus Config" to configure Modbus, which is in the same way as the COM0 port.

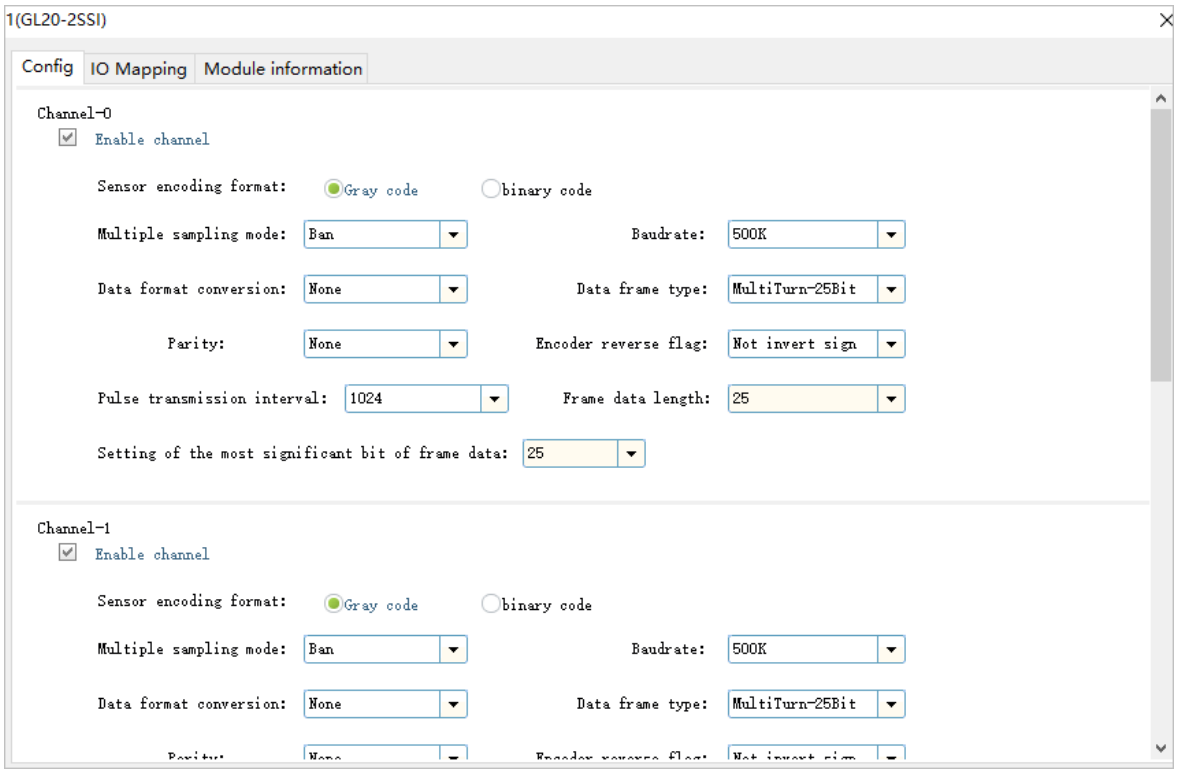
7.2.2.2.8 Configuring a 2-Channel Encoder Input Module

Configure the GL20-2SSI 2-channel encoder input module as follows.

1. Access the configuration page and then the module configuration tab page in any of the following three ways, as shown in the following figure.
   - In the "Project Manager" tree on the left, unfold "Module Configuration", and then double-click "GL20-2SSI".
   - On the configuration page, double-click the GL20-2SSI module.

- In the "Device Detailed List" section, double-click GL20-2SSI.





2. Configure module parameters.

   The GL20-2SSI module has two channels that are configured in the same way.

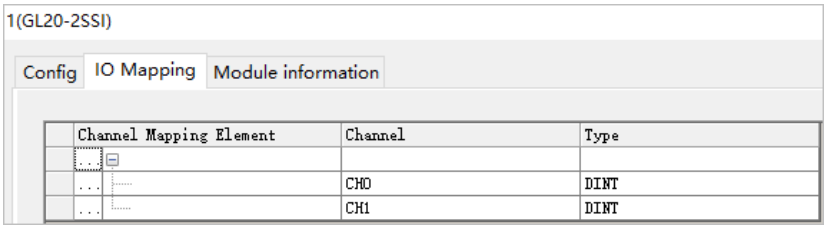
- Configure parameters on the "Config" page. The related parameters are as follows.

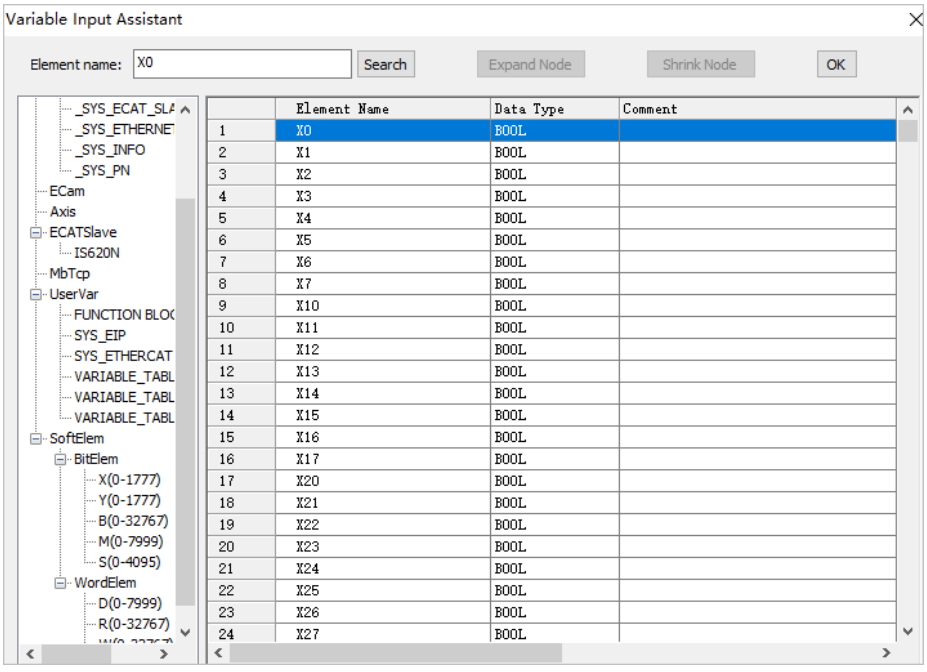
Parameter Name	Description	Value Option/ Range	Default Value
Enable channel	Check this option to enable the channel.	▪ Select ▪ Deselect	Select "Enable channel".
Sensor coding mode	Select the code mode according to encoder type.	▪ Gray code ▪ Binary code	Gray code
Multiple sampling mode	It determines the validity of the encoder position value and sends a fault code to the host if it is invalid.	▪ Disabled ▪ Enabled	Disabled

Parameter Name	Description	Value Option/ Range	Default Value
	Note: The multi-sampling mode can be manually enabled only when PulseTrans Interval is set to 512 or 1024.		
Data format conversion	When "Data format conversion" is set to "Gray", you need to configure this parameter if you want to upload data as binary code. When "Data format conversion" is set to "Binary", use the default value.	▪ None ▪ Gray->Binary	None
Parity	If the assigned parity check is for a 25-bit encoder, 26 bits can be read from the encoder. The bit after the least significant bit will be judged as the parity bit, and parity errors will be sent to the host in the form of a fault code.	▪ None ▪ Odd parity ▪ Even parity	None
Pulse transmission interval	Set the interval according to the encoder specifications. If you need to increase the response speed of the encoder, you can adjust this parameter according to the performance of the encoder. This parameter is strongly related to the encoder performance.	16/32/48/64/512/1024	1024
Setting of the most significant bit of frame data	This parameter is available only when "Data frame type" is set to "Variable".	0 to 32	25
Baudrate	The baud rate affects the accuracy of the encoder as well as the rate of position updates. You can adjust it according to the actual transmission rate and performance of the encoder. This parameter is strongly related to the encoder performance.	100K/200K/ 300K/400K/ 500K/1000K/ 1250K/1500K/ 2000K	500K

Parameter Name	Description	Value Option/ Range	Default Value
Data frame type	It is recommended to select "Variable" when there are status bits in the encoder bits. This parameter must be used in combination with "Setting of the most significant bit of frame data" and "Frame data length".	▪ SingleTurn-13 Bit ▪ MultiTurn-25 Bit ▪ Variable	MultiTurn-25Bit
Encoder reverse flag	The invert function can adjust the motion direction of the absolute encoder to the direction of the motion axis.	No invert sign Invert sign	No invert sign
Frame data length	This parameter is available only when "Data frame type" is set to "Variable".	0 to 32	-

- Click "IO Mapping" and configure the channel mapping element.
  The GL20-2SSI module supports consecutive mapping and separate mapping. You can click "..." at a parent node to map elements or variables consecutively or click "..." before a channel to map a single element or variable. In the Variable Input Assistant" dialog box displayed, select the mapping element to be modified and click "OK". Then, the mapping of the corresponding element is modified.





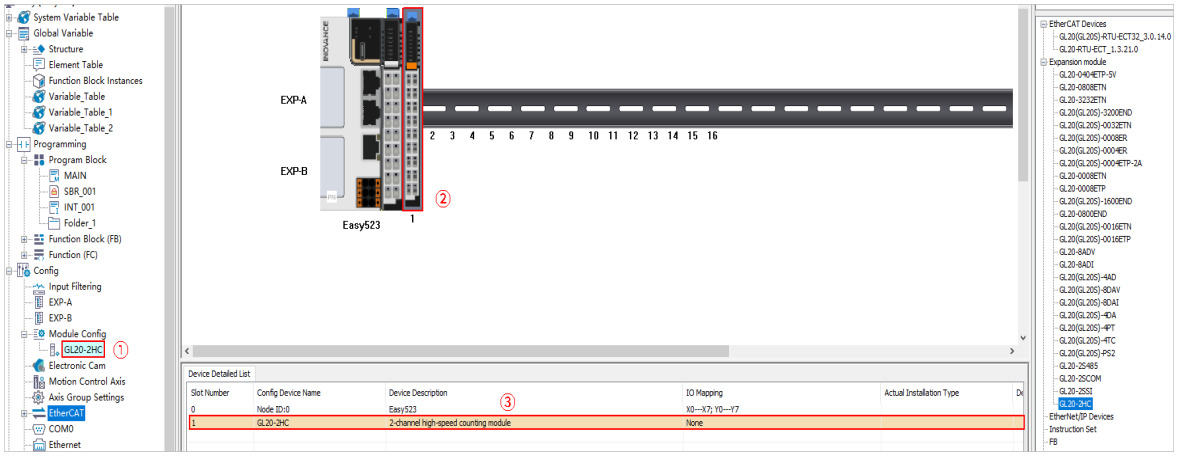
  Click "OK".

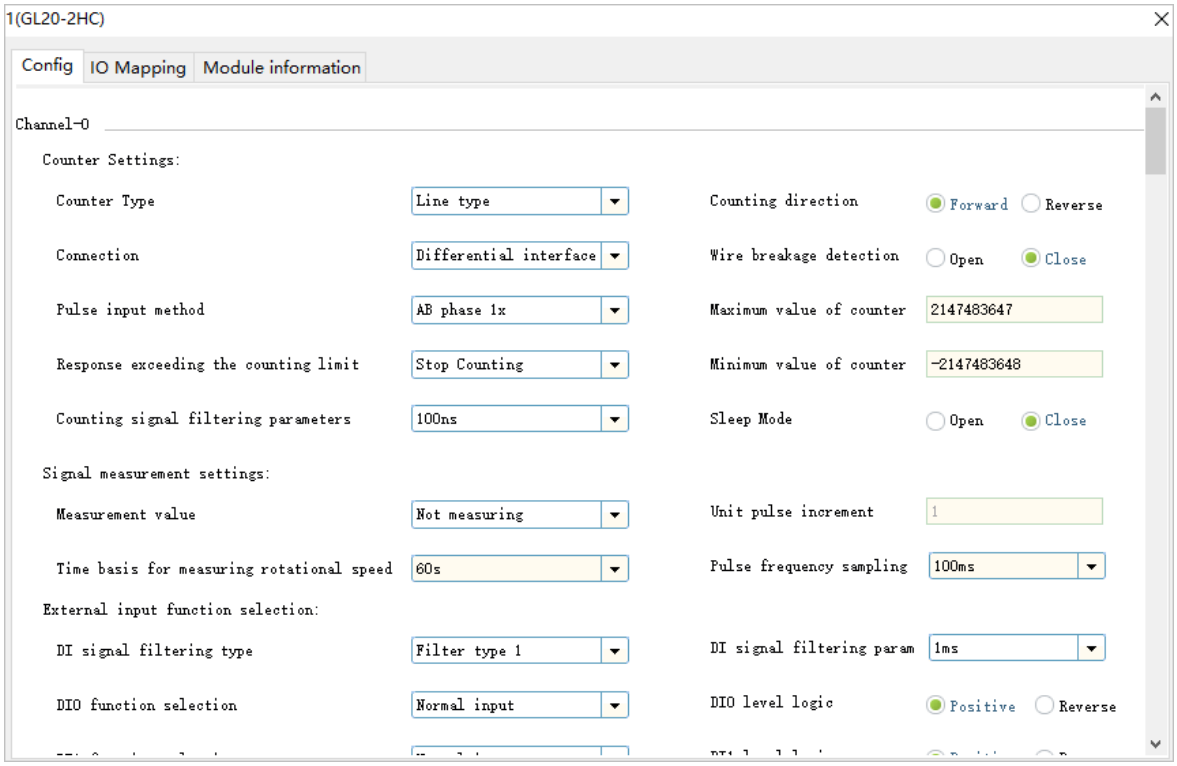
3. After successful compilation, download the project and run it.

7.2.2.2.9 Configuring a 2-Channel High-Speed Count Module

Configure the GL20-2HC 2-channel high-speed count module as follows.

1. Access the configuration page and then the module configuration tab page in any of the following three ways, as shown in the following figure.
   - In the "Project Manager" tree on the left, unfold "Module Config", and then double-click "GL20-2HC".
   - On the configuration page, double-click the GL20-2HC module.
   - In the "Device Detailed List" section, double-click GL20-2HC.





2. Configure module parameters.

   The GL20-2HC module has two channels that are configured in the same way.

- Configure parameters on the "Config" page. The related parameters are as follows.

Parameter Name	Description	Setpoint/Value Range	Default Value
Sleep Mode	Set whether the channel enters the sleep mode.	Select Open or Close.	
Counter Type		• Line type	Line type

Parameter Name	Description	Setpoint/Value Range	Default Value
	• Each channel supports 32-bit counting range, which can be changed by setting the maximum and minimum values. • Line type: Count between the maximum value and the minimum value. After the counting limit is reached, it can be adjusted according to the setting. • Ring type: Count between the maximum and minimum values. When the count-up exceeds the maximum, the value skips to the minimum. When the count-down exceeds the minimum, the value skips to the maximum.	• Ring type	
Counting direction	Set the counter increase direction.	• Forward • Reverse	Forward
Connection	Select the wire connection as needed.	• Differential interface • Single-ended interface	Differential interface
Wire break detection	This function is only available for wire break detection of differential interfaces.	• Open • Close	Close
Pulse input method	• Set this parameter based on the actual counter input signal. • AB phase: Incremental encoder supporting AB phase signal can be connected, and the phase difference between A phase signal and B phase signal should be 90°. • Pulse+direction: Incremental encoder supporting pulse+direction signal can be connected. • CW/CCW: Incremental encoder supporting CW/CCW signal can be connected. • Single-phase A: Pulse encoder without direction signal can be connected. • Single-phase B: Pulse encoder without direction signal can be connected.	• Phase A/B single frequency • Phase A/B double frequency • Phase A/B quadruple frequency • Pulse+Direction • CW/CCW • Single-phase count (A) • Single-phase count (B)	Phase A/B single frequency
Maximum value of counter	The maximum value of the counter.	The value range is -2147483647 to +2147483647.	2147483647

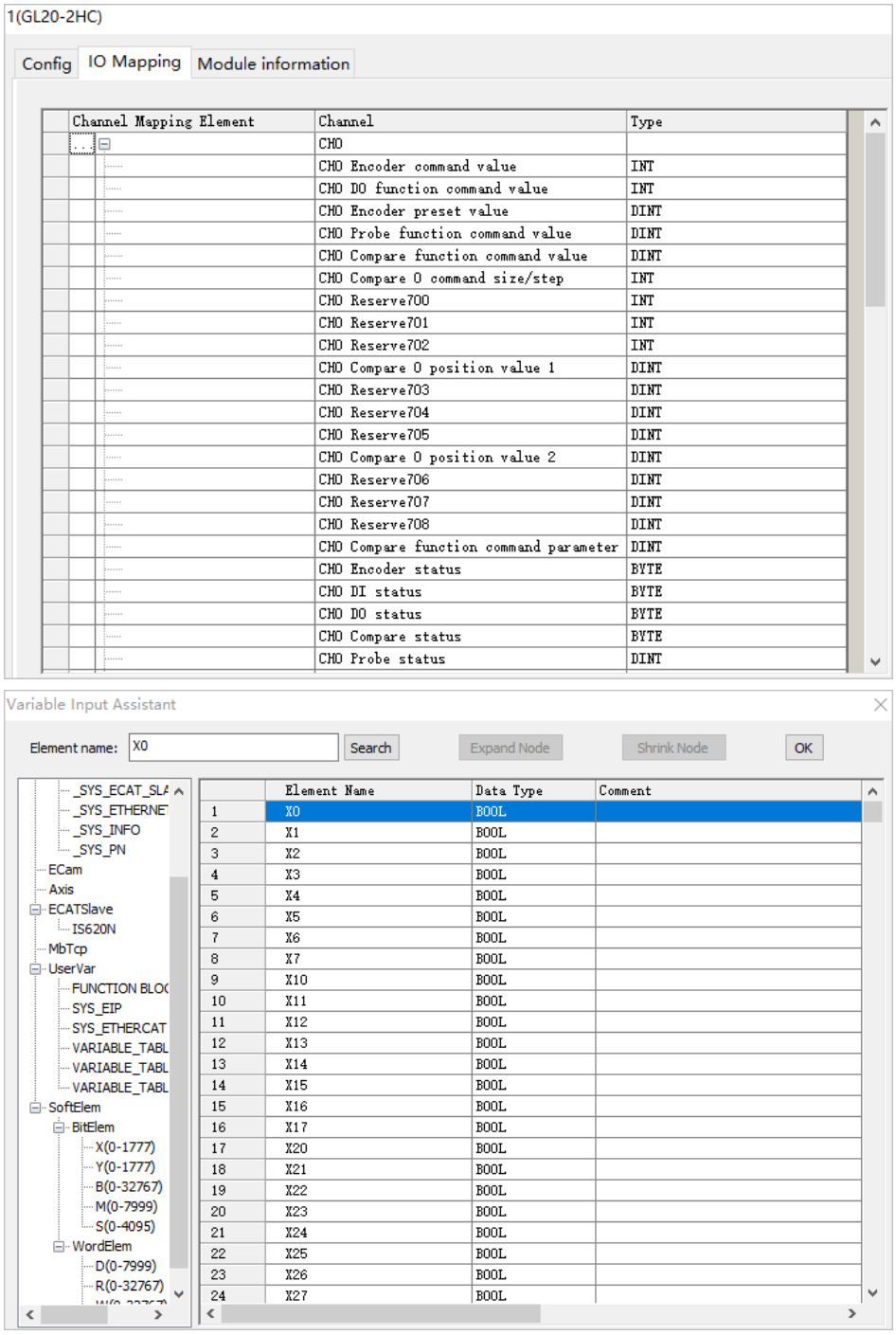
Parameter Name	Description	Setpoint/Value Range	Default Value
Minimum value of counter	The minimum value of the counter.	The value range is -2147483648 to +2147483646.	-2147483648
Response exceeding the counting limit	• This depends on the counter type. Line type: If "Stop Counting" is selected, after the counting limit is reached, counting stops and the counted value overflow alarm is reported. To start counting again, enable the counter again. If "Continue counting" is selected, when the count-up exceeds the maximum, the value skips to the minimum. When the count-down exceeds the minimum, the value skips to the maximum. • Ring type: Invalid	• Stop counting • Continue counting	Stop counting
Counting signal filtering parameters	Through this parameter, the interference between phase A and phase B of counting input can be effectively suppressed, and the signal with holding time less than the set time can be suppressed. It should be between 40% and 60% of the effective signal.	• 50 ns • 100 ns • 200 ns • 500 ns • 1 μs • 2 μs • 5 μs • 10 μs • 20 μs • 50 μs • 100 μs • 200 μs • 500 μs • 1 ms • 2 ms • 5 ms • 10 ms • 20 ms • 50 ms • 100 ms	100 ns
Measurement value	The maximum frequency of external input signal is affected by the connection mode. When "Differential interface" is selected, the maximum	• Not measuring • Frequency	Not measuring

Parameter Name	Description	Setpoint/Value Range	Default Value
	input frequency is 4 MHz. When "Single-ended interface" is selected, the maximum input frequency is 500 KHz. • Not measuring: The signal measuring function and related configuration parameters are all invalid. • Frequency: The frequency of the counter input signal, in Hz. • Period: The period of the counter input signal, in ms. • Speed: The speed of the counter input signal, and the unit depends on the time base of speed measurement.	• Cycle • Speed	
Unit pulse increment	This parameter depends on the measurement value. • Speed: Number of pulses in one turn by encoder • Not measuring, frequency, and period: This parameter is invalid.	The value range is 1 to 65535.	1
Time base for measuring rotational speed	This parameter depends on the measurement value. • Speed: This parameter determines the speed unit, which is RPM by default. • Not measuring, frequency, and period: This parameter is invalid.	• 1 ms • 10 ms • 100 ms • 1s • 60s	60s
Pulse frequency sampling	• Not measuring: This parameter is invalid. • Frequency, period, and speed: The measurement of period and speed is based on the signal frequency, which is strongly related to the frequency calculation. Count the number of input pulse signals within the set time range, so as to calculate the signal frequency. During application, appropriate parameters should be selected, and the sampling time should meet the condition that the period of input signal is less than the sampling time. If conditions permit, the larger the parameter is set, the more accurate the measured value will be.	• 500 μs • 1 ms • 2 ms • 5 ms • 10 ms • 20 ms • 50 ms • 100 ms • 200 ms • 500 ms • 1s • 2s	100 ms

Parameter Name	Description	Setpoint/Value Range	Default Value
		• 5s • 10s • 20s • 25s	
DI signal filtering type	Set the filter type for the DI signal.	Not filtering Filter type 1 Filter type 2	Filter type 1
DI signal filtering param	Through this parameter, the interference at input can be effectively suppressed, and the signal with holding time less than the set time can be suppressed. It should be between 40% and 60% of the effective signal. For DIs, two filter types are available. Select one as needed.	• 100 ns • 500 ns • 1 μs • 100 μs • 250 μs • 500 μs • 1 ms • 2 ms • 3 ms • 8 ms • 16 ms • 32 ms	1 ms
DI0 function selection	Select the function of the DI0 port. This parameter depends on the wiring method. • Single-ended wiring: The DI0 port of the corresponding phase must be set to counting; otherwise, the counter cannot run. • Differential wiring: Invalid parameter	• Normal input • Probe function • Count • Reset to 0 • Preset • Gating	Normal input
DI0 level logic	Select the level logic of the DI0 port. This parameter depends on the wiring method. • Single-ended wiring: Set the level logic of the DI0 signal. • Differential wiring: Invalid parameter	• Positive logic • Negative logic	Positive logic
DI1 function selection		• Normal input • Probe function • Count	Normal input

Parameter Name	Description	Setpoint/Value Range	Default Value
	Select the function of the DI1 port. This parameter depends on the wiring method. • Single-ended wiring: The DI1 port of the corresponding phase must be set to counting; otherwise, the counter cannot run. • Differential wiring: Invalid parameter	• Reset to 0 • Preset • Gating	
DI1 level logic	Select the level logic of the DI1 port. This parameter depends on the wiring method. • Single-ended wiring: Set the level logic of the DI1 signal. • Differential wiring: Invalid parameter	• Positive logic • Negative Logic	Positive logic
DO0 function selection	When comparison is allocated to the DO0 terminal, the comparison function must be configured; otherwise, the comparison function is invalid.	• Normal output • Comparison output	Normal output
DO0 level logic	Set the level logic of the DO0 signal.	• Positive logic • Negative Logic	Positive logic
Output when the module is not running	Set the output status of DO0 when the module is not running.	• Output 0 • Output 1 • Retain	Output 0

- Click "IO Mapping" and configure the channel mapping element.
  The GL20-2HC module supports consecutive mapping. You can click "..." at a parent node to map elements or variables consecutively. In the "Variable Input Assistant" dialog box displayed, select the mapping element to be modified and click "OK". Then, the mapping of the corresponding element is modified.



Click "OK".

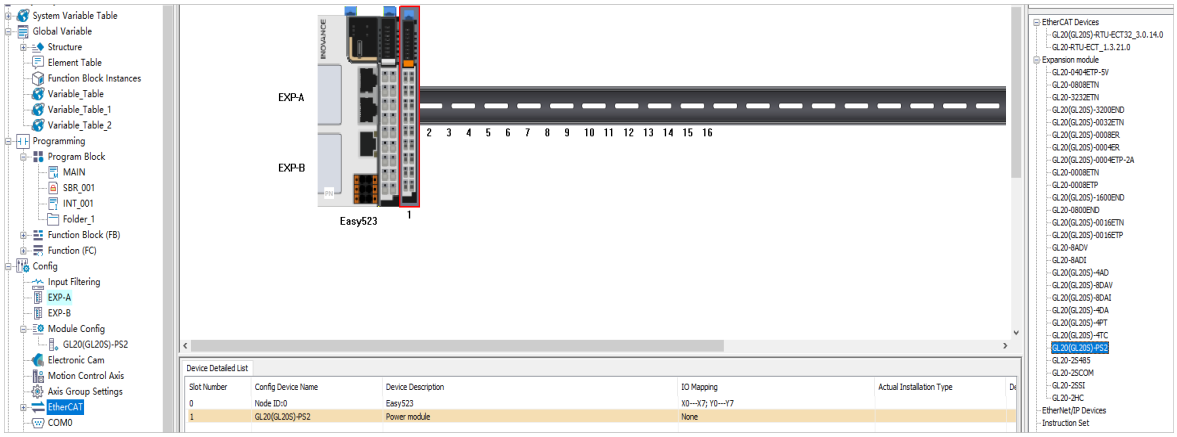
3. After successful compilation, download the project and run it.

7.2.2.2.10 Configuring a Power Module

Configure the GL20(GL20S)-PS2 power module as follows:

1. Access the configuration page and then the module configuration tab page in any of the following three ways, as shown in the following figure.
   - In the "Project Manager" tree on the left, unfold "Module Config", and then double-click "GL20(GL20S)-PS2".
   - On the configuration page, double-click the GL20(GL20S)-PS2 module.

- The GL20(GL20S)-PS2 power module has no configuration page and can be used directly after its actual configuration is accessed.



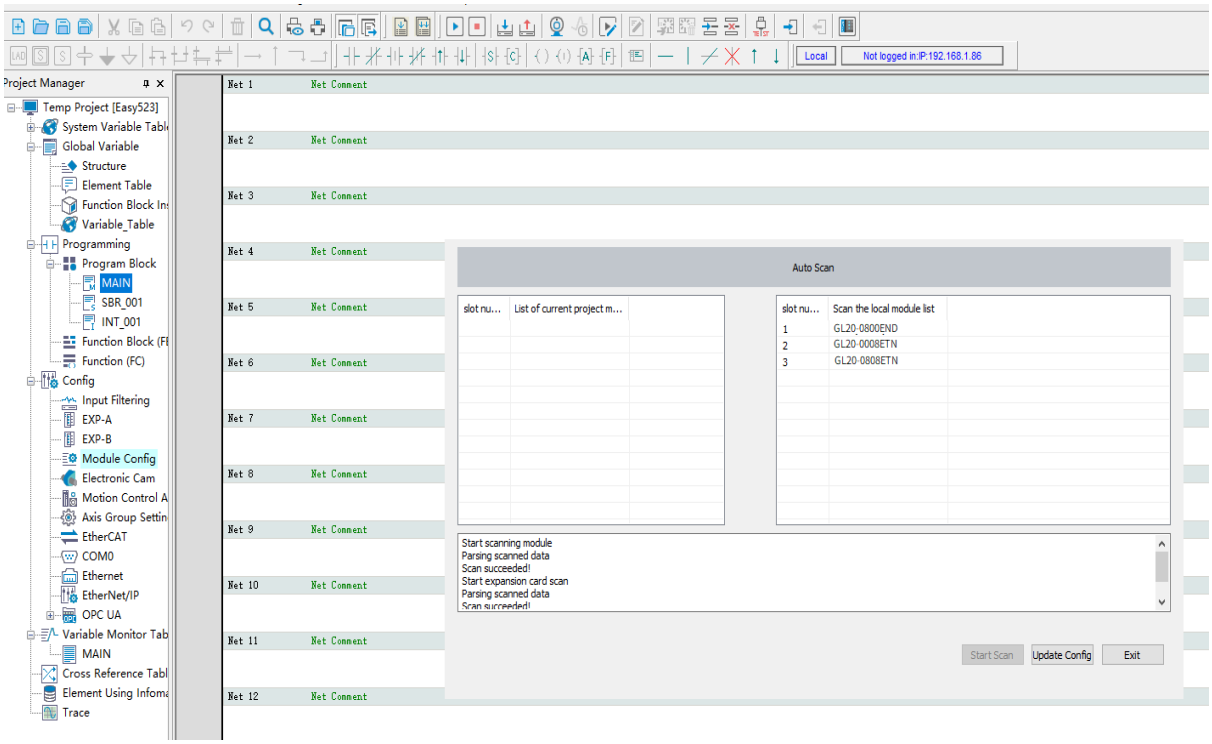
2. In the configuration of the GL20(GL20S)-PS2 power module, you can configure to provide 2 A current output to other modules connected after this module. When 16 non-PS2 modules have been configured, or the actual total current consumption in the configuration exceeds the upper current consumption limit of the PLC, a new power module must be connected.

> **CAUTION**
>
> The GL20(GL20S)-PS2 power module cannot be configured in the last; otherwise, an error will be reported during project compilation.

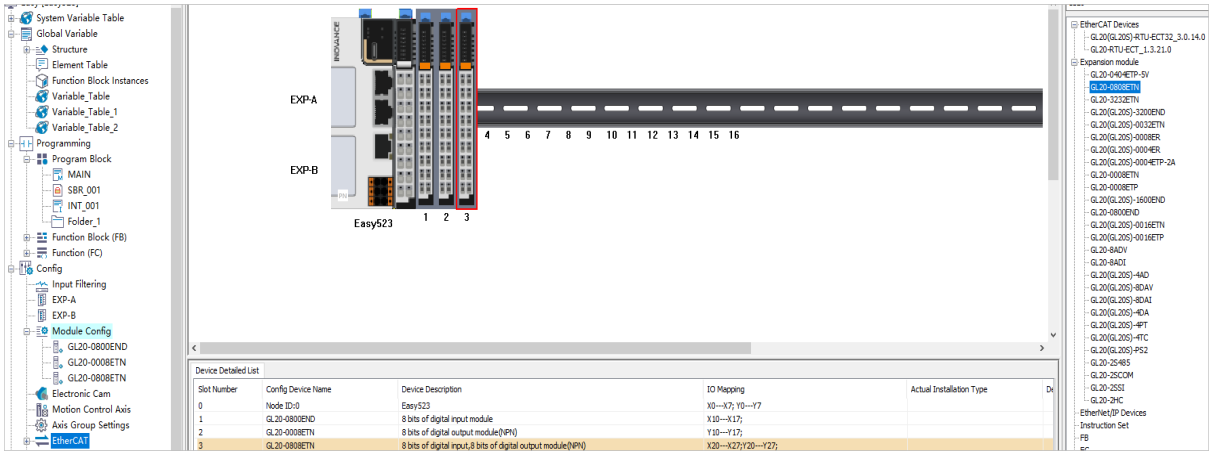
7.2.2.2.11 Application Example

In the following example, GL20-0800END, GL20-0008ETN, and GL20-0808ETN modules are connected to the actual host.

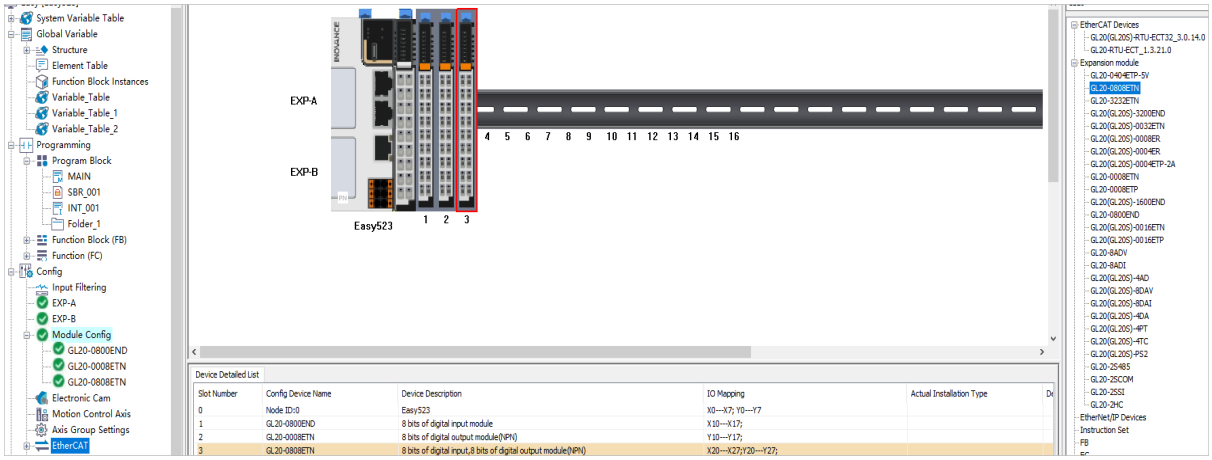
1. In the "Project Manager" tree, right-click "Module Config" and select "Auto Scan" to start scanning. The scan result is displayed as shown in the following figure. The three modules can also be manually added.



2. Click "Update Config" to add the mounted modules to the configuration. The default configuration is added for the modules by default.



3. Download the configuration and run the modules. Nodes of the modules in the "Project Manager" tree turn green and "OK" is displayed on the configuration page, indicating that the modules are running properly.

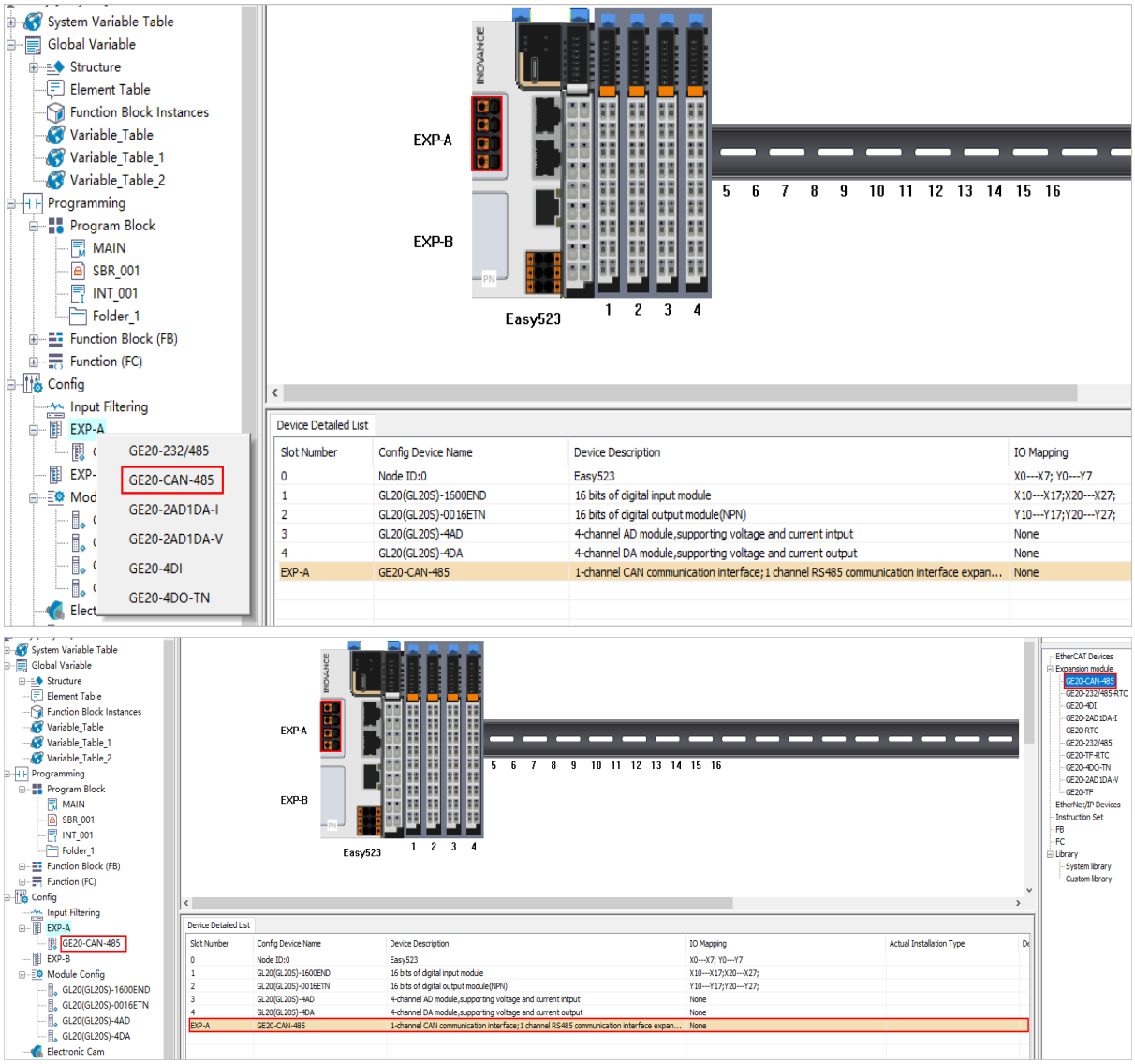


7.2.3 Expansion Card Configuration

7.2.3.1 Hardware Configuration

You can configure hardware in following two ways:

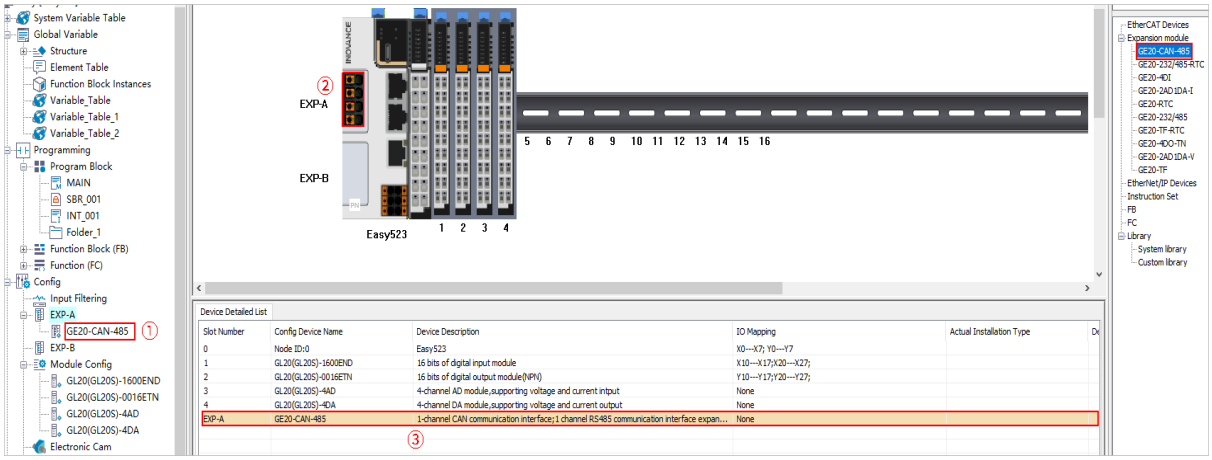
- Unfold the extension module or extension card node on the toolbox, and double-click to add corresponding extension cards.
- In the "Project Manager" tree, right-click "EXP-A" or "EXP-B", and select the extension card type from the shortcut menu displayed. On the configuration page displayed, add corresponding extension cards.



7.2.3.2 Configuring an Expansion Card

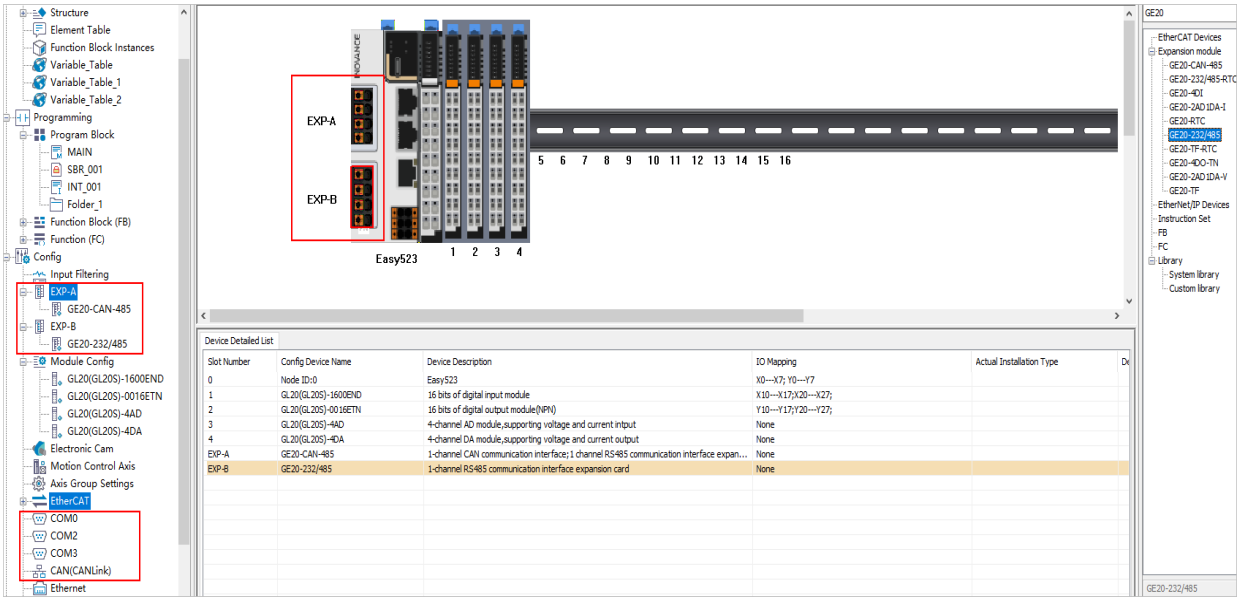
You can access the expansion card configuration page in the following ways:

- Double-click a node under EXP-A or EXP-B.
- Double-click the EXP-A or EXP-B configuration on the configuration page.
- Double-click an item in “Device Detailed List”, as shown in the following figure.



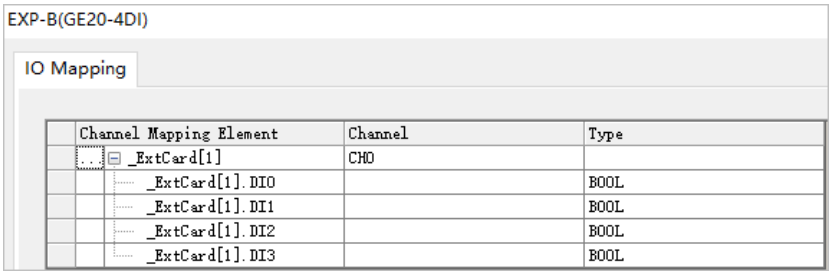
Configurable COM nodes will be generated for the cards with COM ports, such as GE20-CAN-485, GE20-232/485-RTC, and GE20-232/485. EXP-A corresponds to COM2 and EXP-B to COM3. If GE20-CAN-485 is

added for EXP-A, the CAN and COM2 nodes will be generated. If GE20-232/485 is added for EXP-B, the COM3 node will be generated. You can double-click a node for parameter configuration, as shown in the following figure.



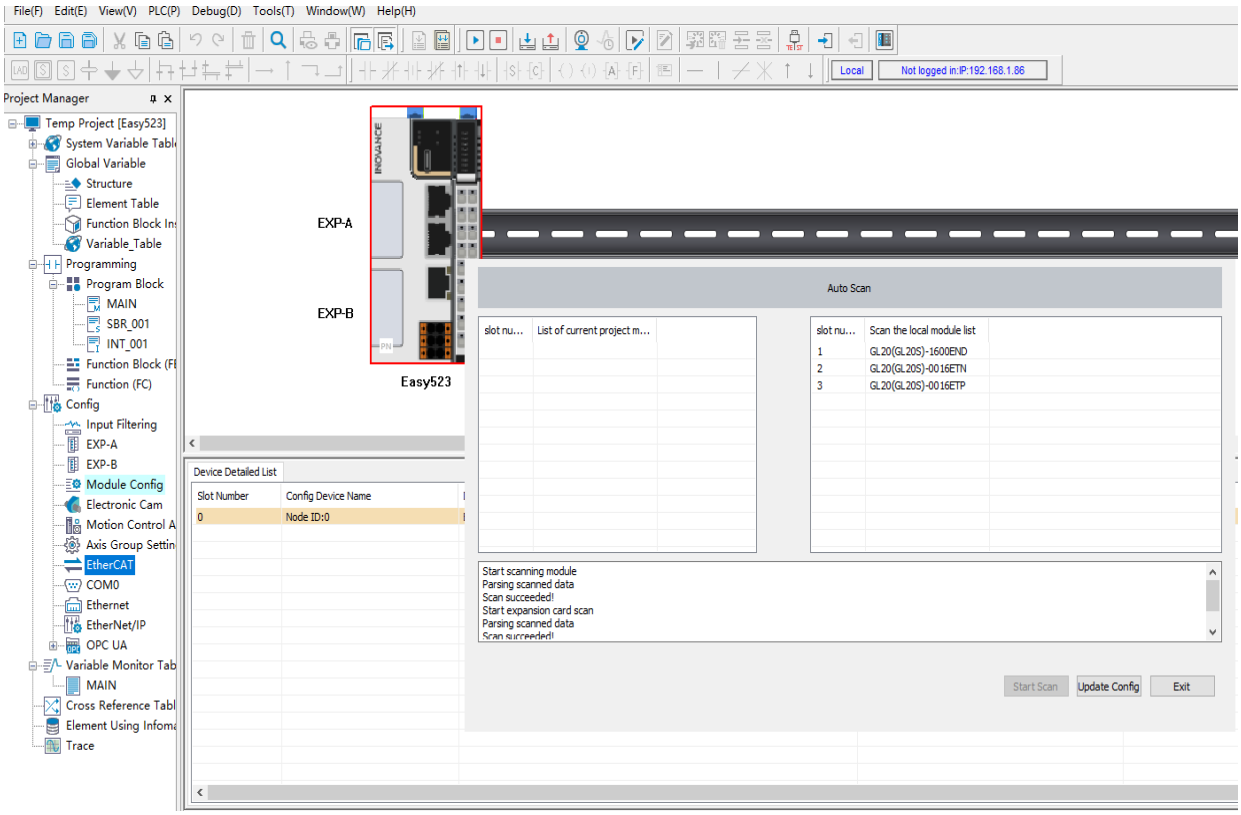
GE20-2AD1DA-I and GE20-2AD1DA-V are current-voltage hybrid expansion cards. If they are added for EXP-A, the COM2 node will be generated by default. If they are added for EXP-B, the COM3 node and three Modbus instructions will be generated by default. You cannot exit COM port parameters and Modbus parameters.

GE20-4DI and GE20-4DO expansion cards are used in the same way as DI and DO modules. The expansion cards provide mappings to system variables by default, which can be automatically modified.

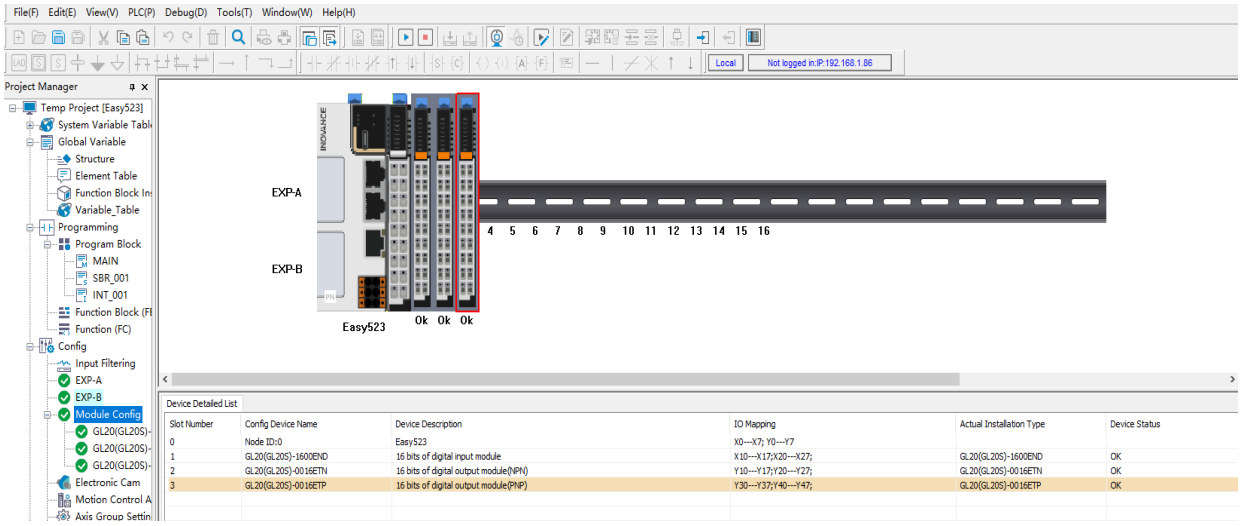


7.2.4 Application Example

After installing the host and module, right-click "Module Config" in the "Project Manager" tree of the software, select "Auto Scan", and use the configuration scanning function to scan modules and extension cards, as shown in the following figure.



On the "Auto Scan" page, click "Update Config" to add the scanned configuration to default configurations, compile and download it, and check the operation status, as shown in the following figure.



7.3 GL20-RTU-ECT Extension Module Configuration

7.3.1 Overview

The GL20 series extension modules can be extended through local bus of the Easy series hosts. They can also be extended through the remote bus of the GL20-RTU-ECT and GL20(GL20S)-RTU-ECT32 communication port modules.

The following table lists modules supported by GL20-RTU-ECT and GL20(GL20S)-RTU-ECT32.

Model	Description
GL20(GL20S)-1600END	16 DI module

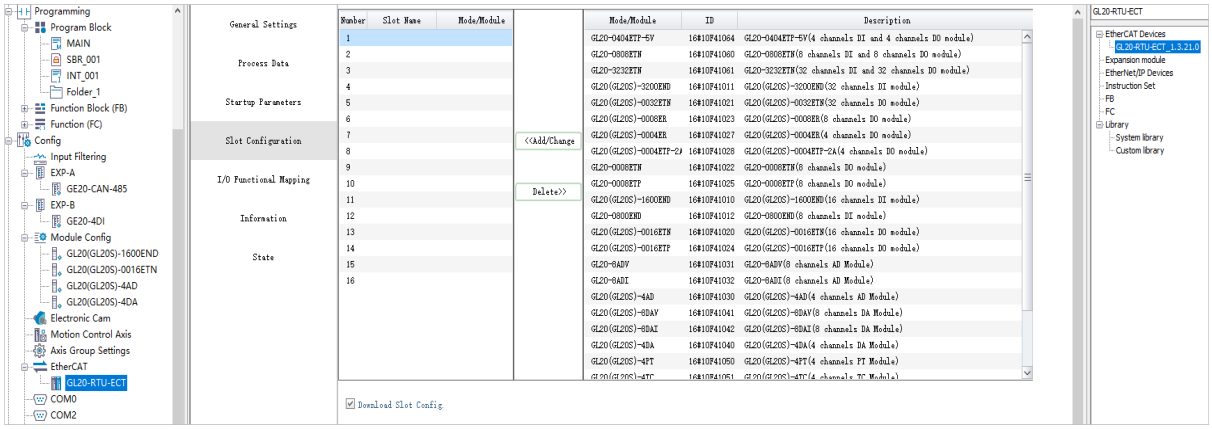
Model	Description
GL20-0800END	8 DI module
GL20(GL20S)-3200END	32 DI module
GL20(GL20S)-0016ETN	16 DO module
GL20(GL20S)-0016ETP	16 DO module
GL20-0008ETN	8 DO module
GL20-0008ETP	8 DO module
GL20(GL20S)-0008ER	8 DO module
GL20(GL20S)-0032ETN	32 DO module
GL20-0808ETN	8 DI module and 8 DO module
GL20-3232ETN	32 DI module and 32 DO module
GL20(GL20S)-4AD	4 AI module
GL20(GL20S)-4DA	4 AO module
GL20(GL20S)-4PT	4-in resistance temperature detector (RTD) module
GL20(GL20S)-4TC	4-in thermocouple temperature detection (TC TEMP MEAS) module
GL20-2SCOM	2 serial communication module (configurable for RS232, RS422, and RS485)
GL20-2S485	2 RS485 serial communication module
GL20-2SSI	2 SSI module
GL20-0404ETP-5V	5 V power I/O module
GL20-8ADV	8-channel analog input module (voltage)
GL20-8ADI	8-channel analog input module (current)
GL20(GL20S)-8DAV	A module with 8 DA terminals, supporting voltage output
GL20(GL20S)-8DAI	A module with 8 DA terminals, supporting current output
GL20(GL20S)-0004ER	4-channel DO module
GL20(GL20S)-0004ETP-2A	4-channel DO module (PNP)
GL20(GL20S)-PS2	Power supply module

Model	Description
GL20-2HC	2-channel high-speed count module

7.3.2 Configuring Hardware and Extension Module

GL20-RTU-ECT is configured in the similar way as GL20(GL20S)-RTU-ECT32. The following takes GL20-RTU-ECT as an example to describe how to configure the module.

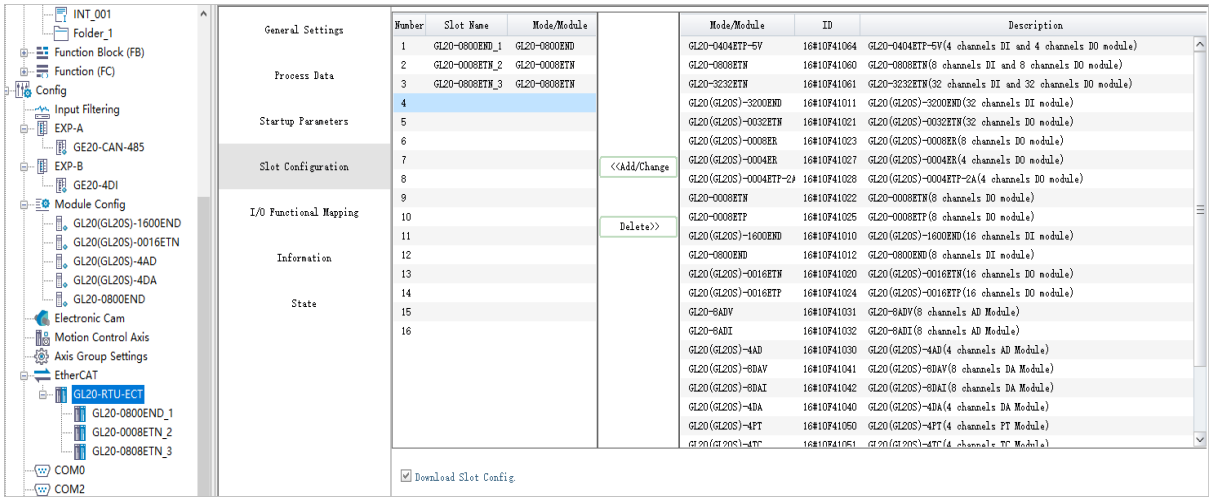
1. Add the GL20-RTU-ECT module.



a. In the "Toolbox" section on the right, choose "EtherCat Devices" > "Inovance Devices" > "IO coupler", and then double-click "GL20-RTU-ECT". In the "Project Manager" tree, a slave node named "GL20-RTU-ECT" is generated under the node "EtherCAT".

b. Double-click "GL20-RTU_ECT" to access its configuration page.

2. Add a GL20 series extension module.



Method 1: Manually add the module

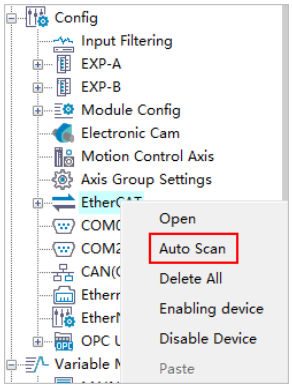
a. Click "Slot Configuration" to access the slot configuration page.

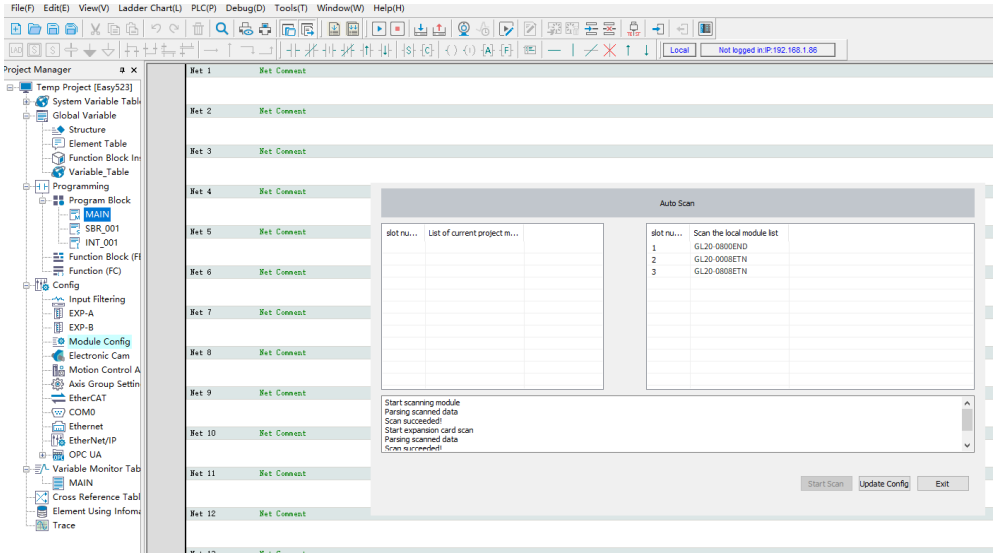
b. Select a module on the right, and click "Add/Change" to add the GL20 series extension module. The following takes GL20-0800END, GL20-0008ETN, and GL20-0808ETN that are actually mounted as an example.

> **NOTE**
>
> - Versions V4.10.1.0 or later can detect the module power consumption. When the sum of the power consumption of added module exceeds the maximum allowed, a prompt is displayed.
> - To delete a module, select a module on the left of the page, and click "Delete".

Method 2: Auto scan

a. After the module cable is connected, power on the PLC, right-click "EtherCAT" in the "Project Manager" tree, and then select "Auto Scan".

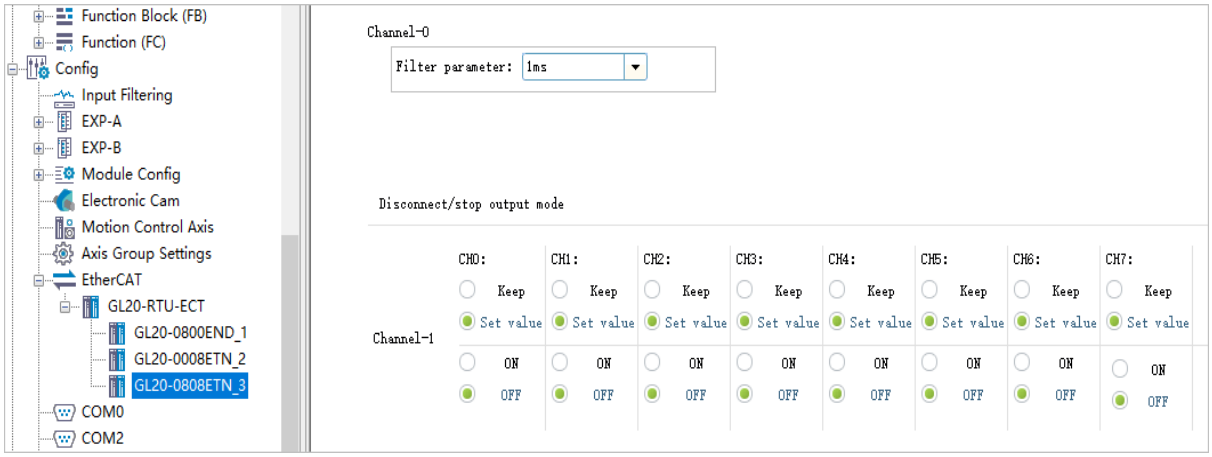




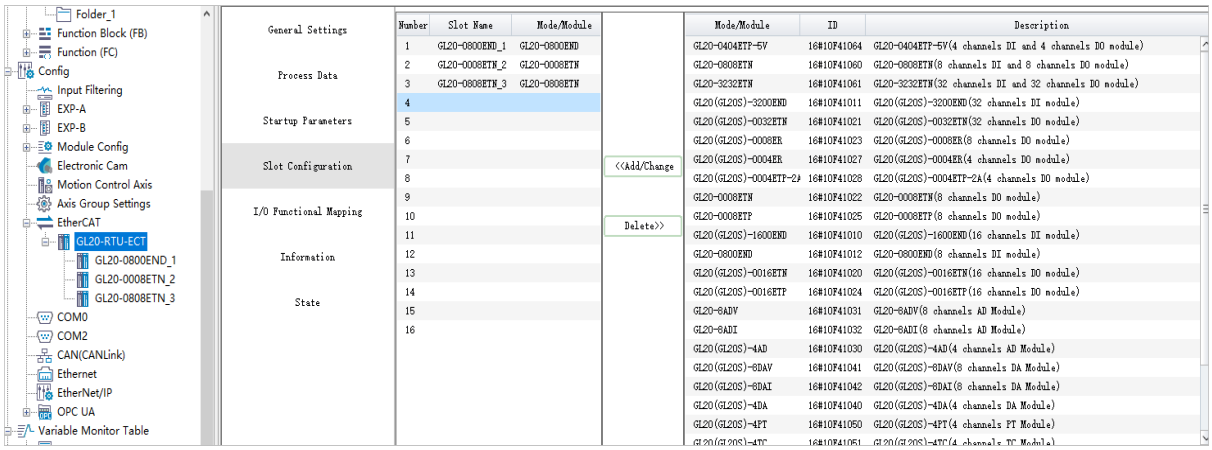
b. Click "Start Scan". The GL20 series extension modules actually mounted are displayed.

> **NOTE**
>
> Versions V4.10.1.0 or later can detect the module power consumption. When the sum of the power consumption of detected modules exceeds the maximum allowed, the configuration can be updated but an alarm is displayed during compilation.

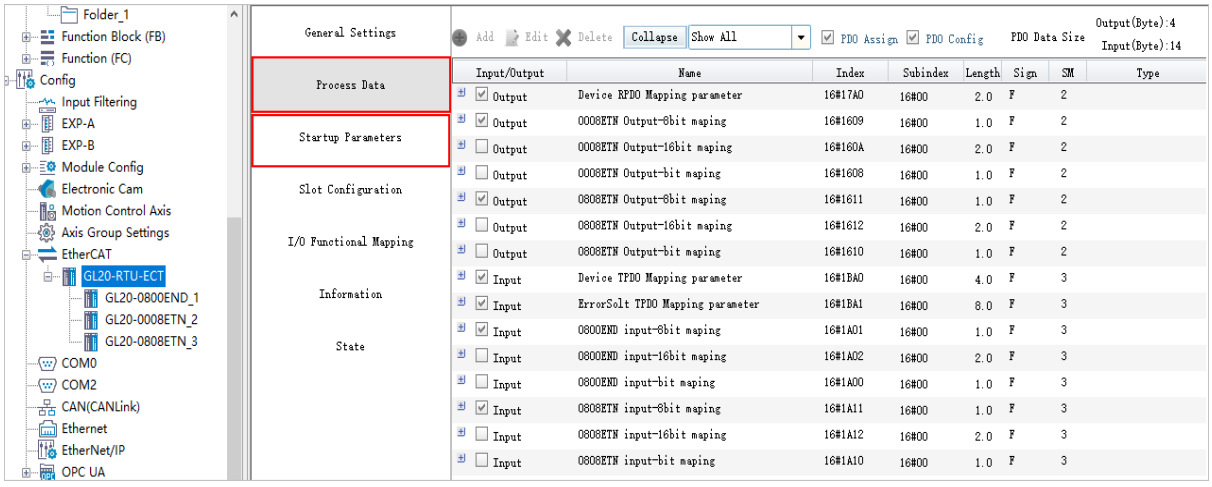
3. Under EtherCAT in the "Project Manager" tree, three nodes are generated under GL20-RTU-ECT, indicating three modules. Double-click each node to access the corresponding module configuration page, and configure module parameters. For details about module parameters, see Configuration of Easy Extension Modules and Extension Cards.



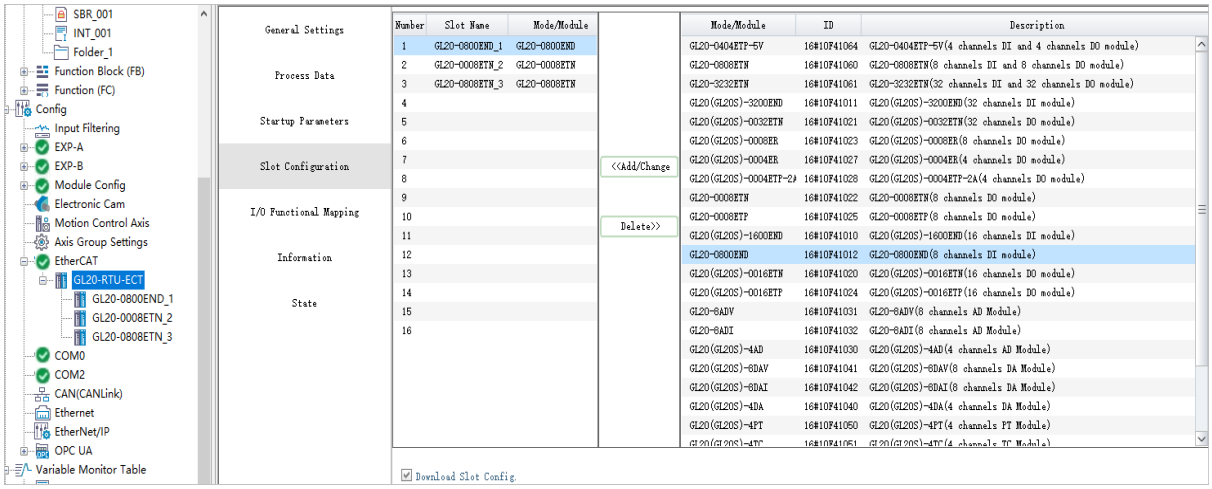
4. Click "Update Config" and select whether to retain current configurations.
   If so, configurations of slave stations scanned for one or more times will be added next to current configurations. If not, current configurations will be deleted and then scanned default configurations need to be added, with the same result as manual adding.



5. Double-click GL20-RTU_ECT and switch to the "Process Data" page and the "Startup Parameters" page to check the generated process data and startup parameters respectively.



6. In the menu bar, click to download the program and click to run the PLC.
   Then, the EtherCAT and GL20-RTC-ECT icons turn green, and no PLC error is reported, indicating that the PLC is running properly.



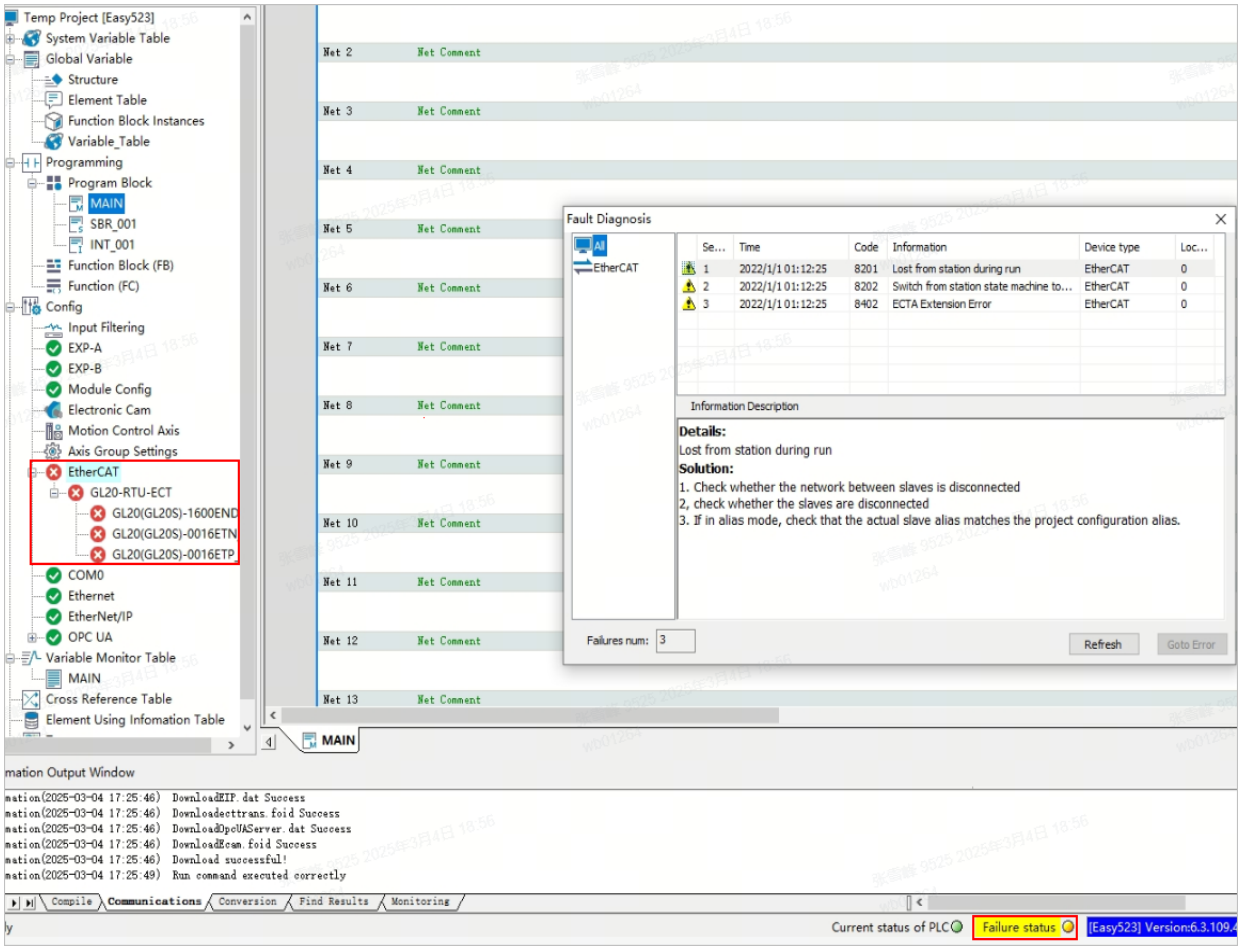
7.3.3 Fault Diagnosis

7.3.3.1 Diagnosis Function

Introduction

The diagnosis function displays information of faults generated by all modules, and provides detailed descriptions and solutions for these faults.

When H5U and GL20-RTU-ECT communication port module are connected normally and run, if the module has a fault, ⊗ will be displayed at the "EtherCAT" node in the "Project Manager" tree on the left, and the "Failure status" in the status bar will be displayed with a yellow background.

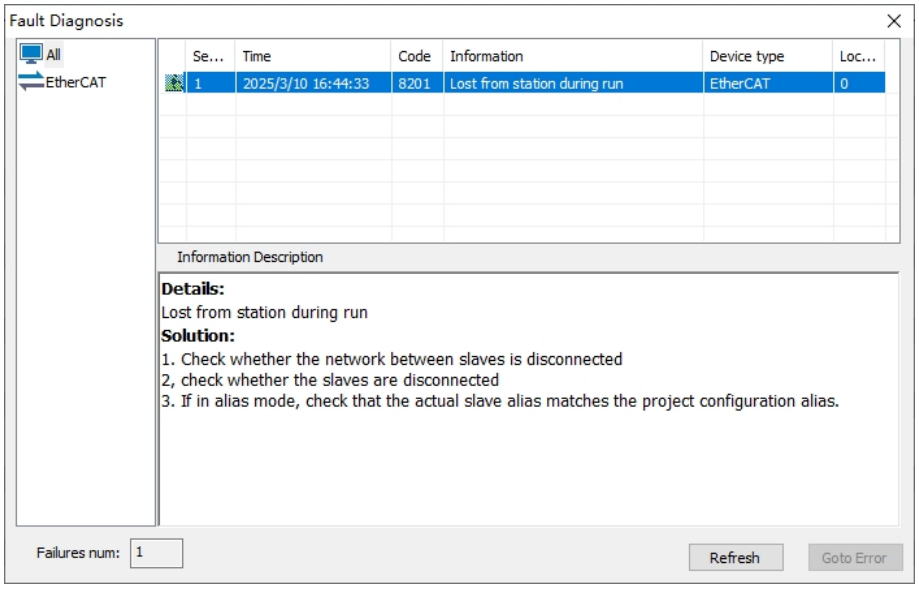


Preconditions

- This function is supported by H5U series with firmware of 6.2.0.0 and later versions.
- This function is supported by Easy series with firmware of 6.3.0.0 and later versions.

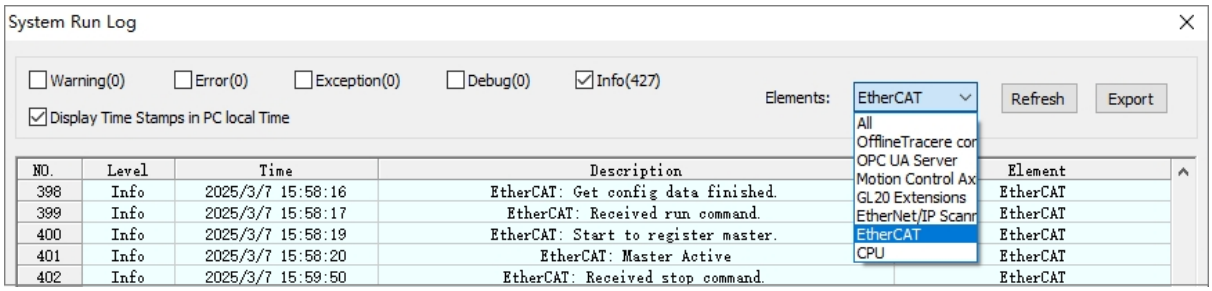
Procedure

1. In the status bar, double-click "Failure status". The "Fault Diagnosis" dialog box is displayed.

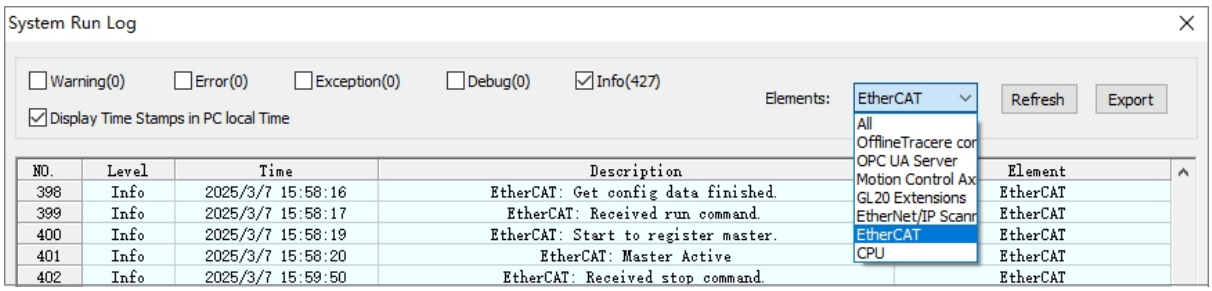


2. Click a fault message. The fault details and solution are displayed in the lower part of the dialog box. Troubleshoot the problem based on the solution.

3. (Optional) To view more fault details, choose "Debug" > "System Run Log View" in the menu bar. The "System Run Log" dialog box is displayed.

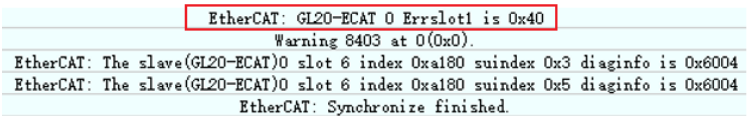


4. Select "EtherCAT" from the drop-down list of "Elements" to view the EtherCAT element log.
   The log contains information such as the fault code (FaultID), slave location (slave), failed slot (slot), diagnosis information (diginfo), and other information.



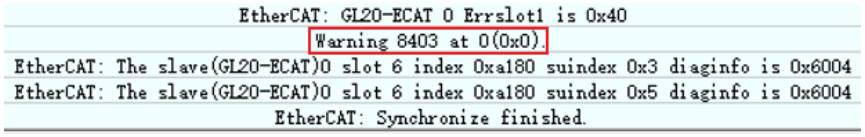
5. • "GL20-ECAT 0" indicates that the failed module is located at the first communication port module, in which 0 indicates that the position of the communication port module is calculated from 0.
   • "0x40" indicates that the module in the 7th slot failed, and "0x40" is converted into binary "1000000", where each bit of the binary represents an extension module, counting from the end, and the last bit

represents the first extension module. The number "0" in the binary bit means no fault, and "1" means a fault.The following figure shows the information of a fault.

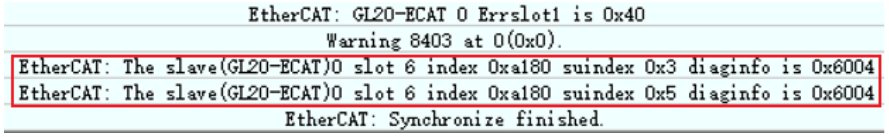


NOTE

You can also view the location of the failed module based on the value of the variable "ErrorSlot1" in the system variable "_RTU_ECT_INFO" in the monitoring mode.



"Warning 8403 at 0(0x0)" indicates that the extension module connected to the GL20-RTU-ECT communication port module failed.



- "The slave(GL20-ECAT)0" indicates that the failed module is located at the first communication port module, in which 0 indicates that the position of the communication port module is calculated from 0.
- "slot 6" indicates that the module in the 7th slot failed, and the module position is calculated from 0.
- "suindex 0x3" indicates that the 2nd channel failed; "suindex 0x5" indicates that the 4th channel failed, and the channel position is calculated from 0x2.
- 0x6004" indicates the module fault code, which means that the data of the module channel exceeds the lower limit. For other fault codes, see Fault Codes.

7.3.3.2 Fault Code

Fault Category	Fault Description		Fault Code
8304 Failed to switch to the SafeOP status (the communication port module status failed to switch to SafeOP due to incorrect module configuration)	The communication protocol conversion module fails.	The number of detected extension modules is 0.	0x1000
		The number of I/O modules configured is more than the number of I/O modules actually scanned.	0x1001
		The number of I/O modules configured is less than the number of I/O modules actually scanned.	0x1002
		The type of I/O modules configured is inconsistent with that of the I/O modules actually scanned.	0x1003
		I/O module configuration failed.	0x1004

Fault Category	Fault Description		Fault Code
		I/O module status switch failed.	0x1005
		The I/O module is offline.	0x1006
8403 Failure of extension module connected to the EtherCAT communication port module.	The I/O module failed.	Power supply overvoltage occurs.	0x5001
		Power supply undervoltage occurs.	0x5002
		The external 24 V power supply fails.	0x5003
		The chip is overheating.	0x5011
		The module ADC device failed.	0x5021
		The module DAC device failed.	0x5022
		The reference channel failed.	0x5031
	I/O channel failure	The channel is disconnected.	0x6001
		The channel is short-circuited.	0x6002
		Data of the channel exceeds the upper limit.	0x6003
		Data of the channel exceeds the lower limit.	0x6004
		The channel data overflows.	0x6005
		The channel data underflows.	0x6006
8402 Extension module failure	The actual configuration is different from the AutoShop configuration.		
8204 Communication port module failure	The type of the actual communication port module is different from that displayed on the AutoShop page.		

7.3.4 Disabling a Module

7.3.4.1 Overview

Precautions

- H5U and Easy series with firmware of version 6.4.0.0 or later support remote module disabling.
- Only the GL20-RTU-ECT32 module (product ID: 16#10F41001) can be disabled.

Function

- Multiple modules of multiple couplers can be disabled. For a single coupler, up to 32 modules can be disabled.
- The disabled module must be deleted from the configuration; otherwise, the startup of the GL20-RTU-ECT32 will be affected, that is, the module configuration type, number, and sequence must be consistent with those of actually connected ones after a module is disabled.
- For the same slave, right-click disabling at the background and disabling through system variables cannot be applied together. In other words, when a slave has been disabled by the context menu operation at the background, it cannot be disabled again through a system variable.
- To disable a slave through a system variable, write the system variable _RTU_ECT_INFO[0].DisableSlotControl within the first scan cycle after startup (that is, after M8000 is set to 1)/ Then, the module disabling setting takes effect, and this system variable cannot be written after the system starts.
- When the GL20-2HC module is configured, no matter whether you want to disable this module, you must arrange this module in any of the first four slots; otherwise, the startup of the coupler will be affected.

7.3.4.2 Disabling a Module Through the Module List

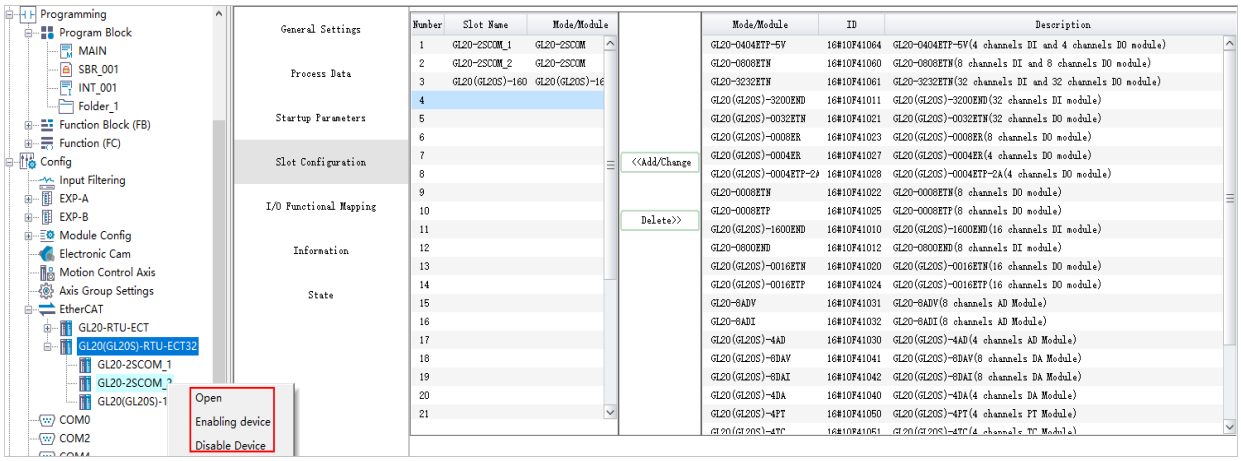
Preconditions

- Firmware version: V6.4.0.0 and later
- Software version: AutoShop V4.10.2.0 and later
- The communication port module, I/O extension module and PLC host have established communication with AutoShop.

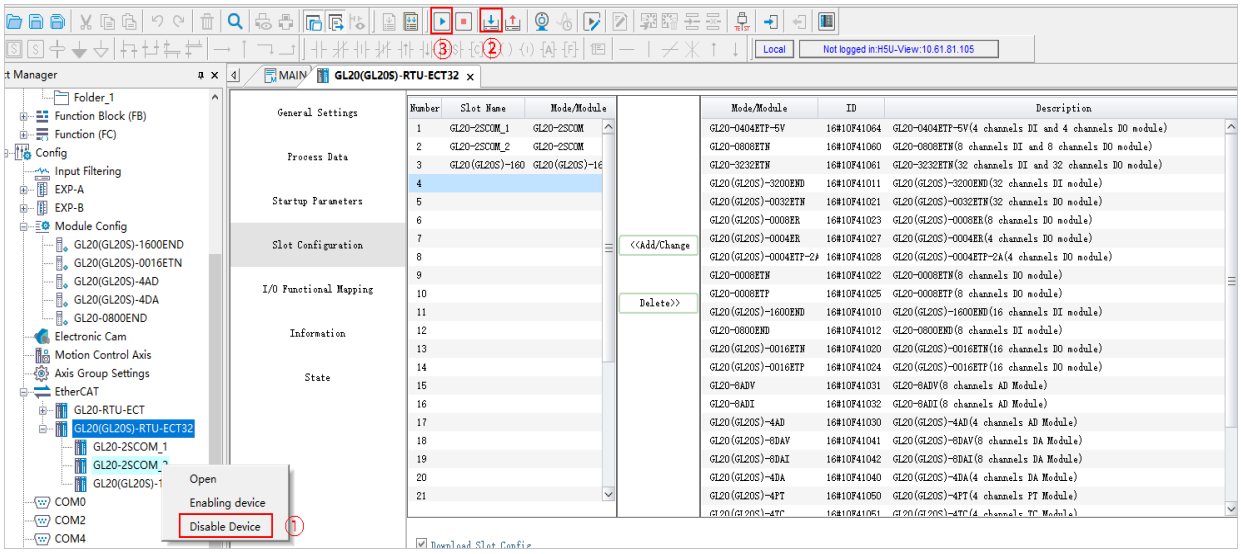
Procedure

In the following example, the H5U PLC communicates with the GL20-RTU-ECT32 communication port module.

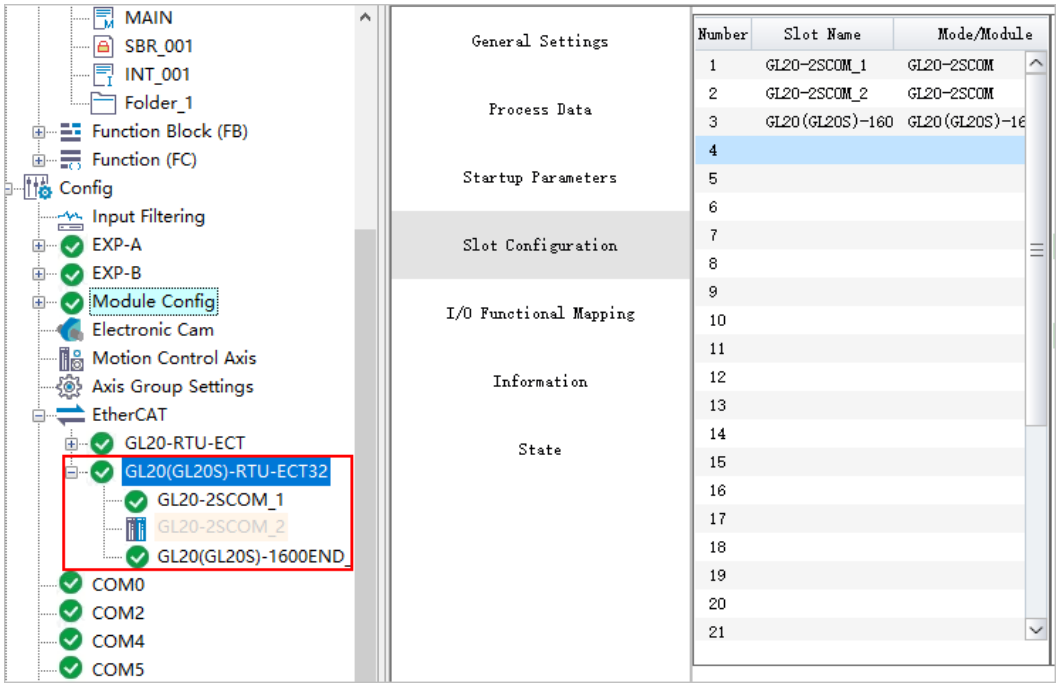
1. In the module list, right-click "GL20(GL20S)0016ETP-2", as shown in the following figure.



2. Select "Disable Device" at ①, and click "Download" at ② to download the program to the PLC. Click "RUN" at ③ to run the PLC, as shown in the following figure.



3. Click "Monitor". The GL20-RTU-ECT32 is enabled, and the disabled module turns gray, indicating that this module is disabled, as shown in the following figure.



7.3.4.3 Disabling a Module Through System Variable

Preconditions

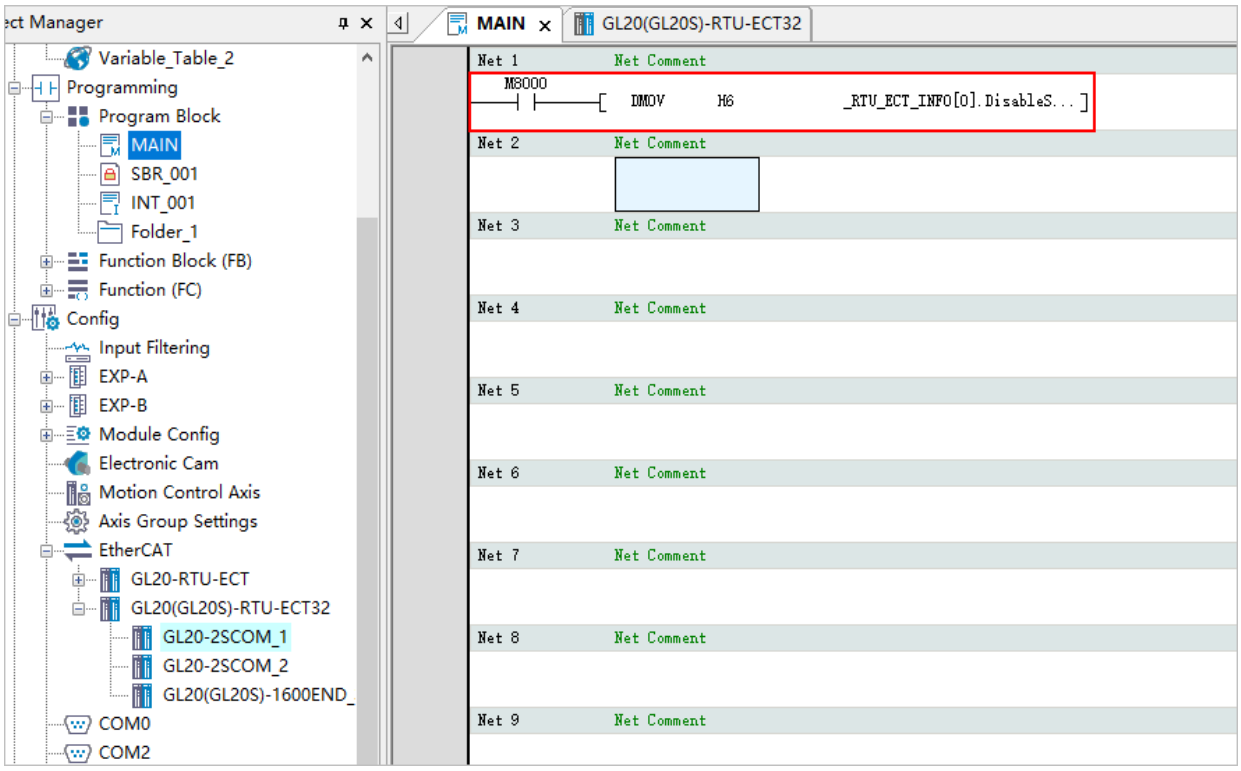
- Firmware version: V6.4.0.0 and later
- Software version: AutoShop V4.10.2.0 and later
- The communication port module, I/O extension module and PLC host have established communication with AutoShop.

Procedure

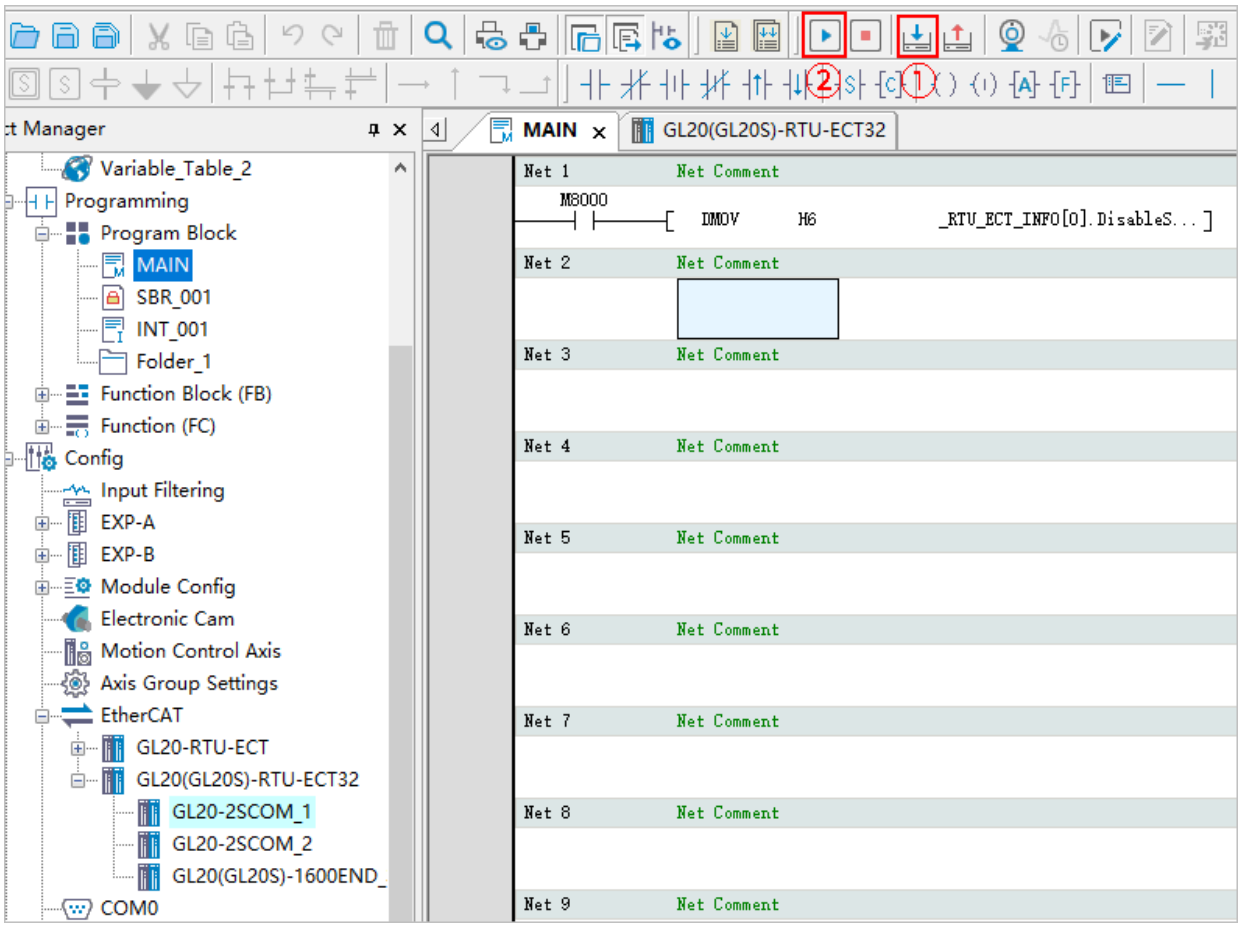
To disable an I/O extension module, use the system variable _RTU_ECT_INFO[0].DisableSlotControl. Each bit of its value indicates a module. If the bit is 1, the module is disabled. If the bit is 0, the module is not disabled.

In the following example, the H5U PLC communicates with the GL20-RTU-ECT32 communication port module.

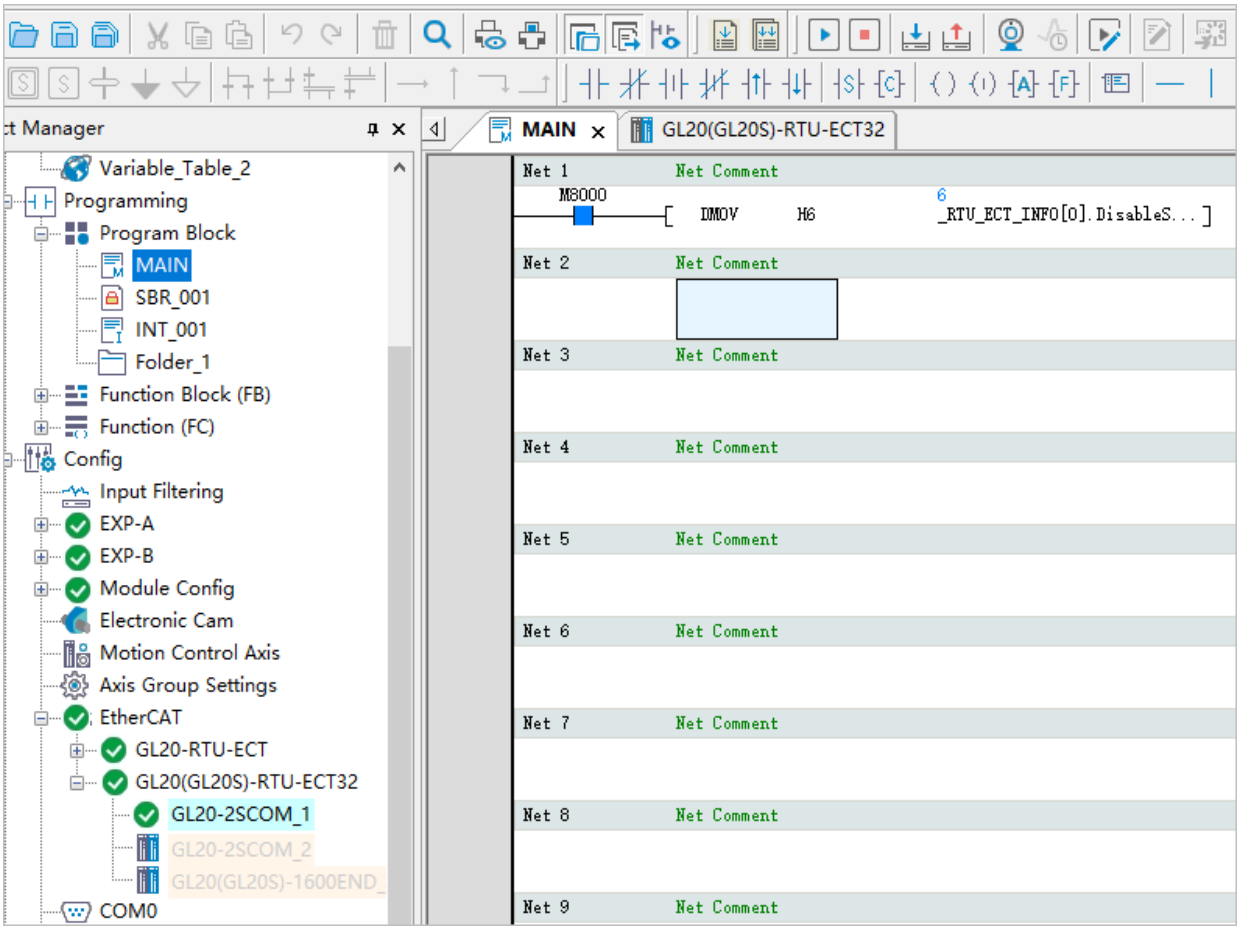
1. Within the first scan cycle after startup, write the value of the system variable _RTU_ECT_INFO[0].DisableSlotControl to 6 (6 in decimal corresponds to 110 in hexadecimal system, indicating that the second and third modules are disabled), as shown in the following figure.



2. Click "Download" at ① to download the program to the PLC. Click "RUN" at ② to run the PLC, as shown in the following figure.



3. Click "Monitor". The GL20-RTU-ECT32 communication port module is enabled, and the disabled I/O extension module turns gray, indicating that this module is disabled, as shown in the following figure.



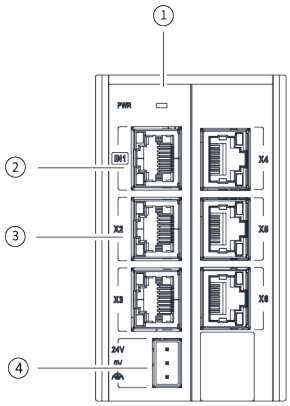
7.4 GR10-EC-6SW Branch Module Configuration

7.4.1 Overview

NOTE

- The GR10-EC-6SW branch module is applicable to H5Us with firmware of version 6.1.0.1 and later.
- The GR10-EC-6SW branch module is applicable to Easy with firmware of version 6.3.0.0 and later.

The GR10-EC-6SW branch module is used to extend extension ports, as shown in the following figure.



No.	Interface	Function		
①	Power indicator	PWR	Green	ON upon power-on
②	EtherCAT input port	IN	Port1 and EtherCAT input port, connecting to the front EtherCAT master station	
③	EtherCAT output port	X2	Port2 and EtherCAT output port, connecting to the rear EtherCAT slave station	
		X3	Port3 and EtherCAT output port, connecting to the rear EtherCAT slave station	
		X4	Port4 and EtherCAT output port, connecting to the rear EtherCAT slave station	
		X5	Port5 and EtherCAT output port, connecting to the rear EtherCAT slave station	
		X6	Port6 and EtherCAT output port, connecting to the rear EtherCAT slave station	
④	24 V power input terminal	Module power input		

The GR10-EC-6SW branch module can be connected to multiple EtherCAT slave devices.

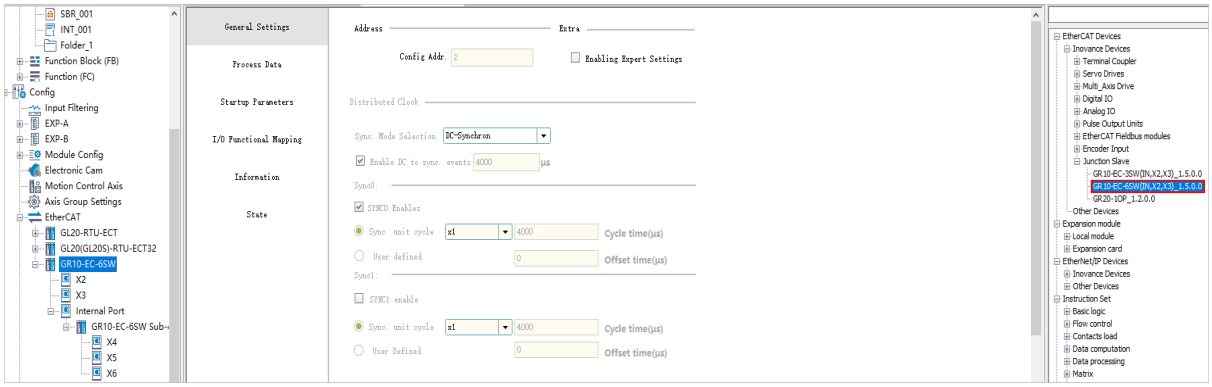
For more information about the GR10-EC-6SW branch module, see the GR10-EC-6SW 6-EtherCAT Branch Module User Guide.

7.4.2 Adding a Branch Module and Its Slave Device

Branch modules and slave devices can be added manually or by automatic scanning.

Manual adding

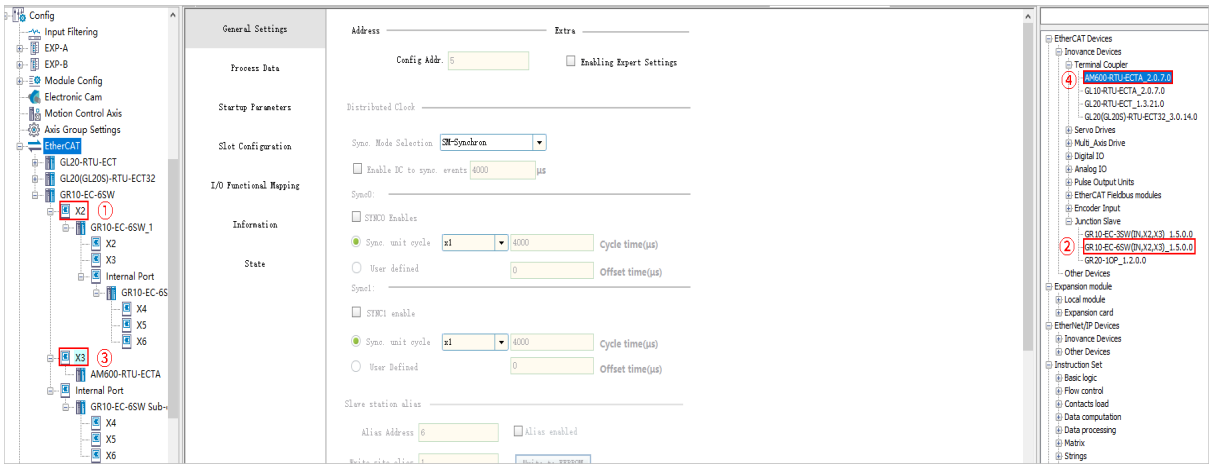
1. In the toolbox, choose “EtherCAT Devices” > “Inovance Devices” > “Junction Slave”, and double-click GR10-EC-6SW.



2. Select the EtherCAT output port with the same physical configuration and add a branch module or slave device.

   For example, to use the X2 EtherCAT output port to add a branch module, select “X2” and double click “GR10-EC-6SW” in the toolbox; to use the X3 EtherCAT output port to add the slave device AM600-RTU-

ECTA, select “X3”, choose “Inovance Devices” > “Terminal Coupler” in the toolbox, and double click “AM600-RTU-ECTA”.



NOTE

- Up to three levels of branch modules can be added.
- Multiple slaves or branch modules can be added under the EtherCAT output port of a branch module.
- If Slave-Disable is turned on for the selected node, all branch nodes under the slave are disabled. Branch nodes also support Slave-Disable. If an EtherCAT output port is disabled, all slaves under the port are disabled.
- Slaves of branch modules do not support copy and paste.

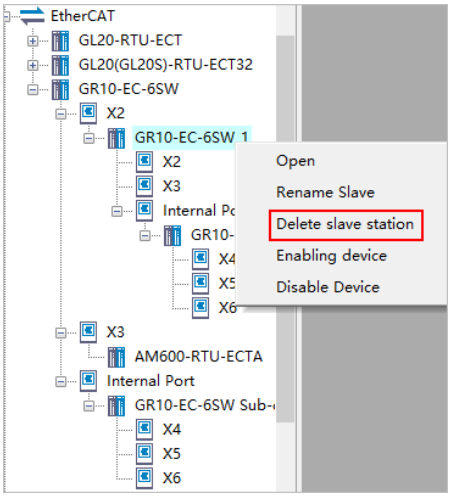
3. For details about the EtherCAT slave station configuration, see 11.3 Slave Configuration.

Automatic scanning

1. In the “Project Manager” tree, right click “EtherCAT” and select “Auto Scan”. Click “Update Config”, select not to retain current configurations, and add the GR10-EC-6SW branch module and its slave device.
2. For details about the EtherCAT slave station configuration, see section 9.3 “Slave Configuration” in chapter 9 “EtherCAT Communication”.

7.4.3 Adding the Branch Module and Its Slave Device

In the “Project Manager” tree on the left, right-click the branch module or slave device under “EtherCAT”, and select “Delete slave station” to delete the branch module or slave device. After the slave station is deleted, data of the slave station and all its subnodes is deleted.



CAUTION

The branch modules as the X2 to X6 nodes, Internal Port, and GR10-EC-6SW Sub-device node must not be deleted.

7.5 GS20-ECT-8L Module Configuration

7.5.1 Overview

The GS20-ECT-8L module is connected to the sensor or activator as an EtherCAT slave station using the IO-Link protocol.

IO-Link is a serial digital communication protocol used to integrate a smart sensor and an activator into an automation system. According to IEC 61131-9, it is the first worldwide standard I/O technology for periodic data exchange between sensors/activators and PLCs. IO-Link is an open point-to-point communication protocol but not a fieldbus protocol. The protocol features ease of use, stabilization, reliability, and plug and play and is used more and more widely with the development of industry 4.0.

The IO-Link topology consists of PLCs, IO-Link master stations (GS20-ECT-8L), IO-Link slave stations (GR20-16EMNL, GR20-16EMPL, GS20-16EMNL, or GS20-16EMPL), sensor, activator, and IO-Link cable, as shown in the following figure. IO-Link master stations are connected to each other and PLCs through EtherCAT communication. IO-Link master stations are connected to IO-Link slave stations and sensors/activators and IO-Link slave stations are connected to sensors/activators through IO-Link communication. IO-Link master stations and slave stations can supply power for sensors/activators directly.

NOTE

The IO-Link ports support two working modes: IO-Link mode and Standard I/O (SIO) mode. The working mode can be set separately on any port.

7.5.2 Module Configuration

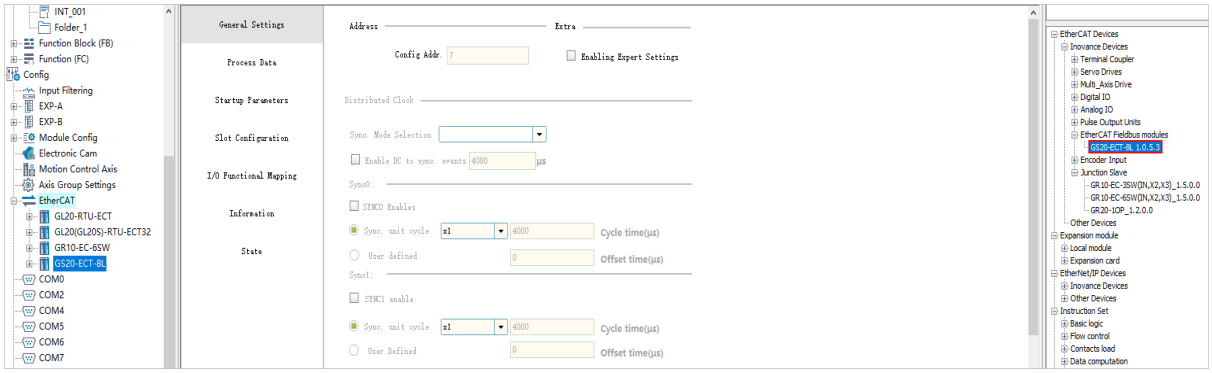
Preconditions

The PLC is connected to the GS20-ECT-8L module through cables and is powered on.

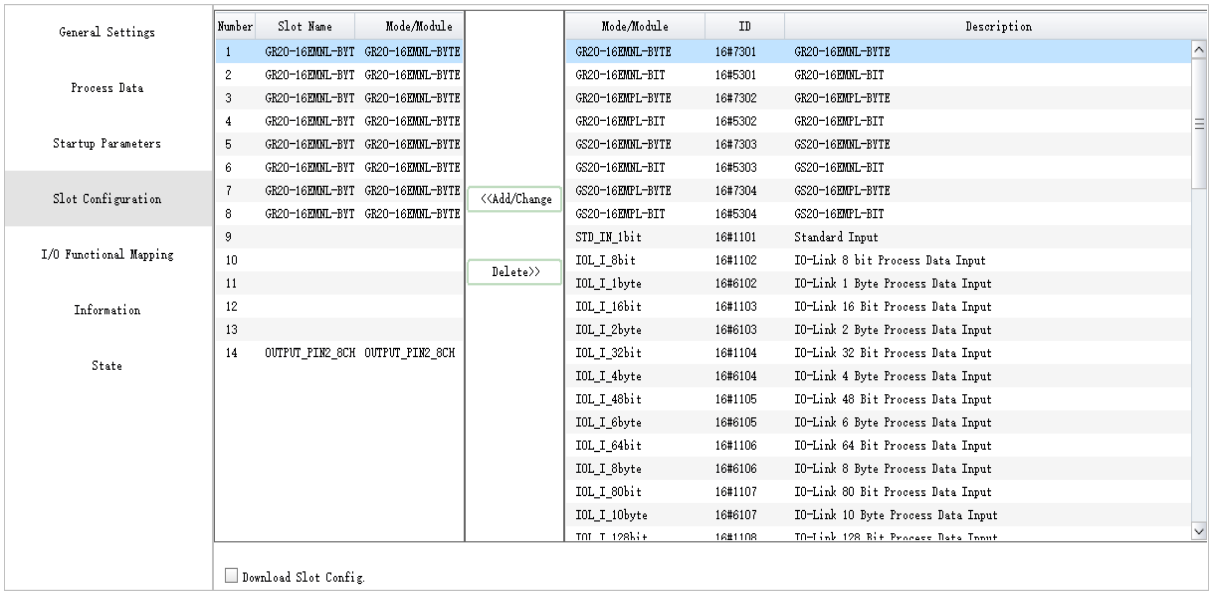
Procedure

1. Add the GS20-ECT-8L module.
   In the toolbox, choose "EtherCAT Devices" > "Inovance Devices" > "EtherCAT Fieldbus modules", and double-click "GS20-ECT-8L" to add the GS20-ECT-8L module. Alternatively, in the "Project Manager" tree, right-click "EtherCAT" and select "Auto Scan". After successful scanning, click "Update Config" to update the device configuration and add the GS20-ECT-8L module.

> **NOTE**
>
> If you choose to discard the current configuration, delete all slave configurations first, and then add the scanned devices. If you choose to retain the current configuration, add the scanned slaves one by one after the existing slaves.



2. Configure slots.
   Click "Slot Configuration". On the page displayed, select corresponding slots and click "Delete" to delete devices from the slots. When the slot area is empty, select target devices on the right and click "Add/Change" to add the selected devices to the slots. The following table lists the slot numbers, names, and description.



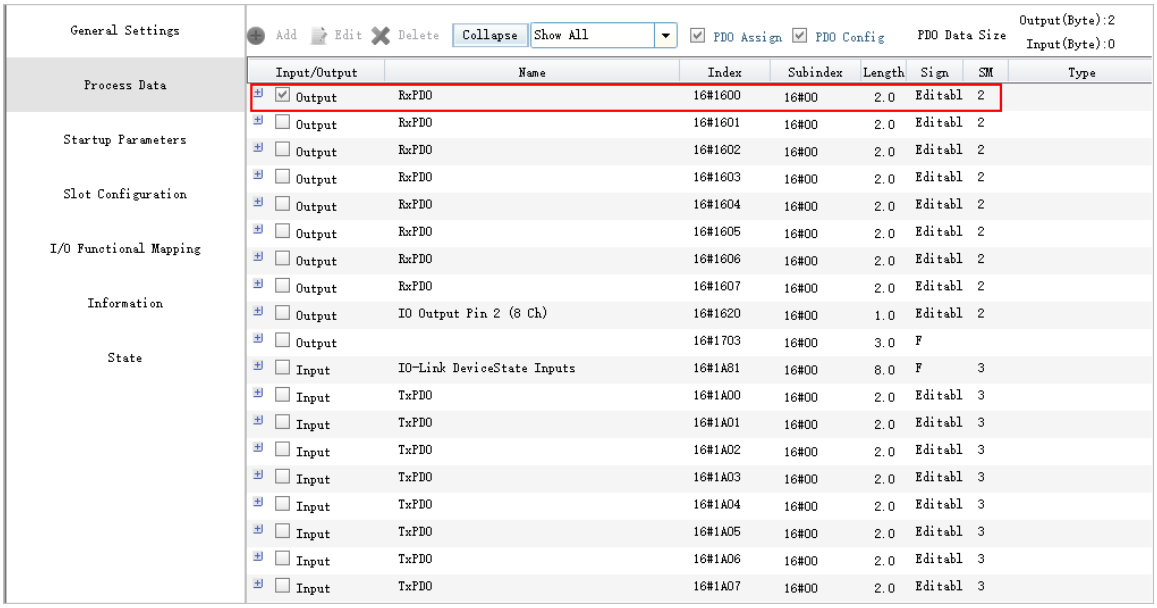
No.	Slot Name	Description
1 to 8 (corresponding to IO-Link physical ports 0 to 7 of the	GR20-16EMNL-BYTE, GR20-16EMNL-BIT, GR20-16EMPL-BYTE, GR20-16EMPL-BIT, GS20-16EMNL-BYTE,	Names of Inovance IO-Link slave modules, where N indicates active low and P indicates active high

No.	Slot Name	Description
GS20-ECT-8L module)	GS20-16EMNL-BIT, GS20-16EMPL-BYTE, GS20-16EMPL-BIT	
	STD_IN_1bit	Standard input
	STD_OUT_1bit	Standard output
	IOL_I_ xbyte/IOL_I_ xbit	IO_Link input, such as IOL_I_1byte, indicating 1-byte IO-Link input
	IOL_O_ xbyte/IOL_O_ xbit	IO_Link output, such as IOL_O_1byte, indicating 1-byte IO-Link output
	IOL_I/O_ x/_ ybyte/IOL_I/O_ x/_ ybit	IO_Link I/O, such as IOL_I/O_1/_1byte, indicating 1-byte IO-Link input and 1-byte IO-Link output
9	INPUT_PIN2_8CH	PIN2 input
10	ACTOR_SHORTCIRCUIT_PIN2_8CH	PIN2 short circuit monitoring
11	ACTOR_SHORTCIRCUIT_PIN4_8CH	PIN4 short circuit monitoring
12	SENSOR_SUPPLY_SHORTCIRCUIT_8CH	PIN1 short circuit monitoring
13	SYSTEM_HARDWARE_MONITOR	Hardware status monitoring
14	OUTPUT_PIN2_8CH	PIN2 output

3. Configure process data and I/O function mapping.

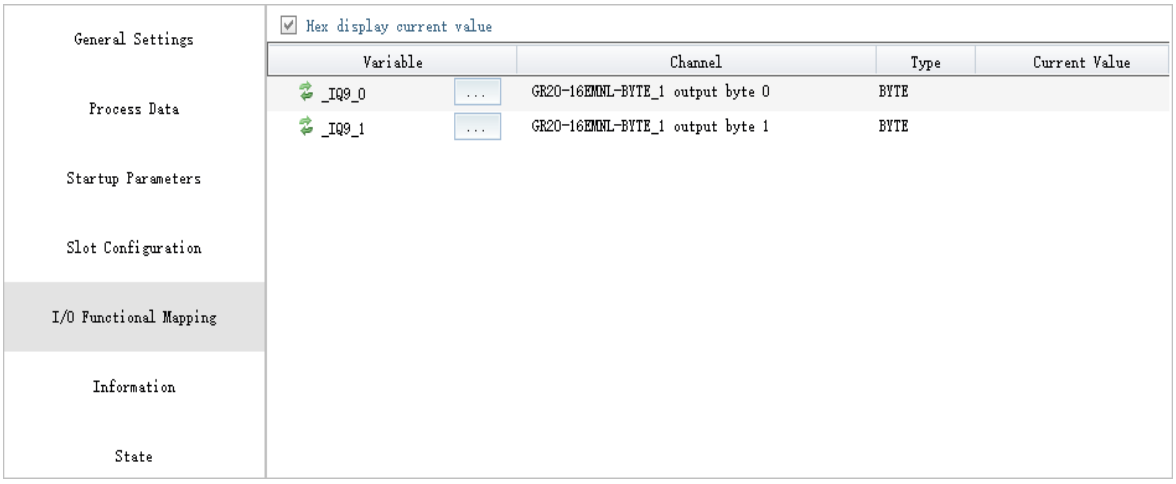
NOTE

- The background DINT is of the signed 32-bit type, so an error occurs when UDINT data is read. If there is no signed number with more than 32 bits, test whether the value read is correct. If decimal display is incomplete, use hexadecimal display. In hexadecimal display, unsigned 32-bit numbers are displayed. Device IDs and other UDINT type data can be viewed in hexadecimal display.
- The BITARR8 type of PDO data is mapped to the BYTE type by default, and can be operated bitwise according to the BYTE type, or mapped to a BOOL array for reading and writing.Click "Process Data" and select corresponding I/O PDOs as required. The corresponding I/O mapping data is generated on the "IO Functional Mapping" page. The PDO index 16#1600 is output as an example.



Input/Output	Name	Index	Subindex	Length	Sign	SM	Type
Output	RxPDO	16#1600	16#00	2.0	Editabl	2	
Output	RxPDO	16#1601	16#00	2.0	Editabl	2	
Output	RxPDO	16#1602	16#00	2.0	Editabl	2	
Output	RxPDO	16#1603	16#00	2.0	Editabl	2	
Output	RxPDO	16#1604	16#00	2.0	Editabl	2	
Output	RxPDO	16#1605	16#00	2.0	Editabl	2	
Output	RxPDO	16#1606	16#00	2.0	Editabl	2	
Output	RxPDO	16#1607	16#00	2.0	Editabl	2	
Output	IO Output Pin 2 (8 Ch)	16#1620	16#00	1.0	Editabl	2	
Output		16#1703	16#00	3.0	F		
Input	IO-Link DeviceState Inputs	16#1A81	16#00	8.0	F	3	
Input	TxPDO	16#1A00	16#00	2.0	Editabl	3	
Input	TxPDO	16#1A01	16#00	2.0	Editabl	3	
Input	TxPDO	16#1A02	16#00	2.0	Editabl	3	
Input	TxPDO	16#1A03	16#00	2.0	Editabl	3	
Input	TxPDO	16#1A04	16#00	2.0	Editabl	3	
Input	TxPDO	16#1A05	16#00	2.0	Editabl	3	
Input	TxPDO	16#1A06	16#00	2.0	Editabl	3	
Input	TxPDO	16#1A07	16#00	2.0	Editabl	3	

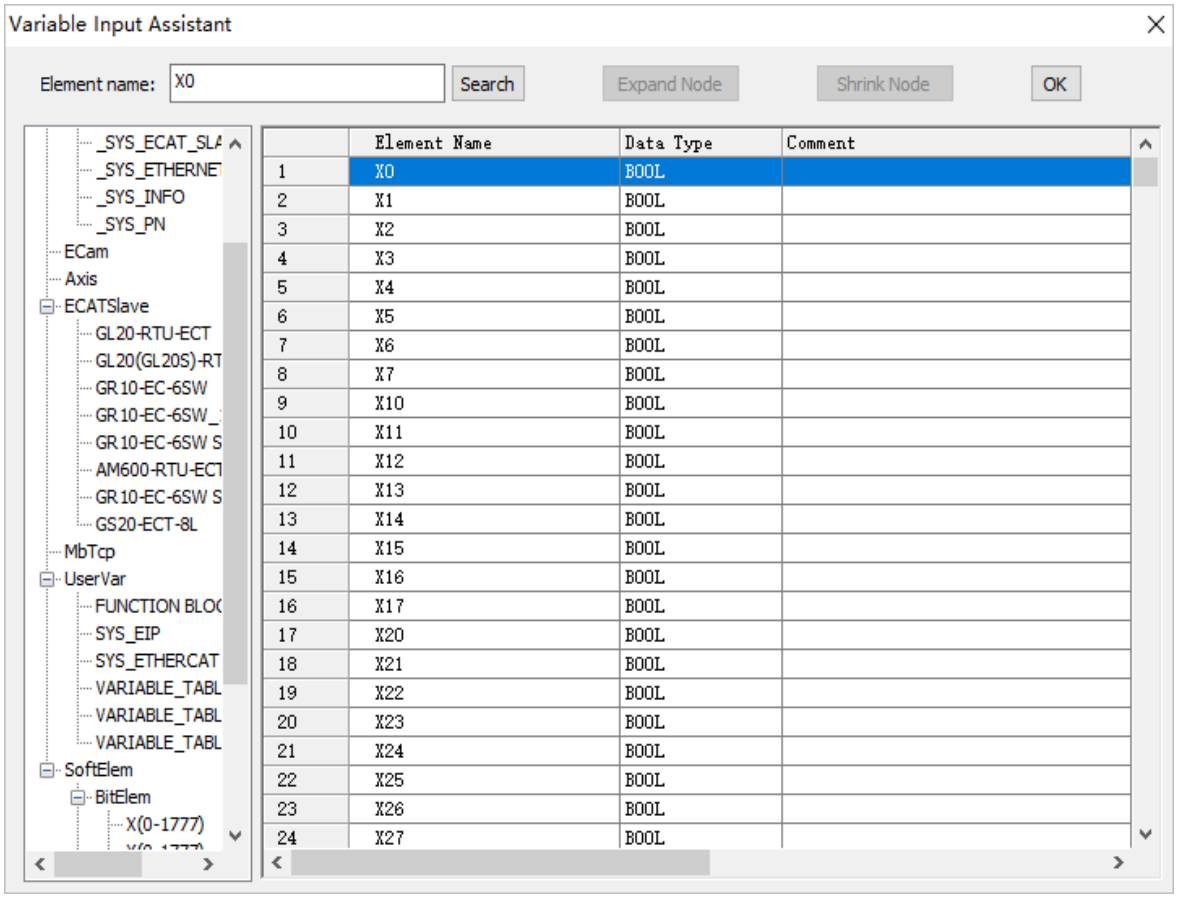
a. Then, channel mappings for "GR20-16EMNL-BYTE_1 output byte 0" and "GR20-16EMNL-BYTE_1 output byte 1" are generated by default.



b. (Optional) Set a customized variable.

Click or . In the dialog box displayed, select a customized variable and click "OK".

> **NOTE**
>
> To use the default mapping variables, skip this step.

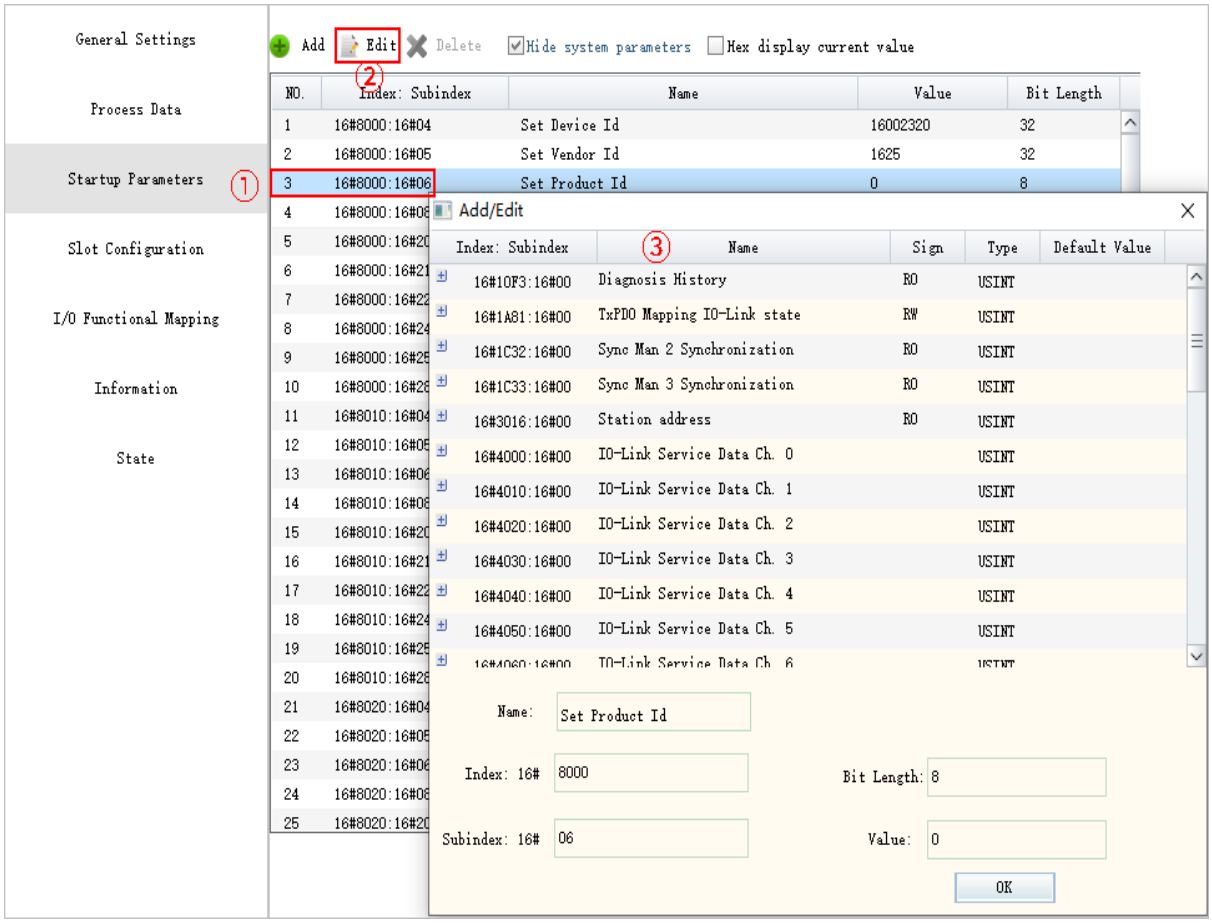


c. For example, configure the GR20_16EMPL byte operation through CH0 and the GR20_16EMPL bit operation through CH1 to map the corresponding input and output variables respectively. Set Output_pin2_chn of the corresponding port if the slave is configured with output data.

d. Download and run the program.
The RUN indicator of the IO-Link master flashes green and then turns steady green. The "0" indicators of the ports configured to the IO-Link mode flash green.

e. Connect the IO-Link slave of the GR20-16EMPL model to port 0 and port 1.
The "0" indicators on port 0 and port 1 turn steady green. The "1" indicators on port 0 and port 1 are steady yellow, for example, Output_pin2_chn. At the slave, the US indicator is steady green and the COM indicator flashes green.

4. (Optional) Modify the startup parameter of the IO-Link slave.
After an IO-Link slave is added, its default startup parameters are generated on the "Start parameter" page. To modify a parameter, select the parameter and click "Edit". In the dialog box displayed, modify the parameter and then click "OK".



5. Configure parameters of the IO-Link slave.
   Basic parameters of the IO-Link slave can be configured by index 0x40n0 (n = 0 to 7). For details about indexes, see . I/O port information of the IO-Link slave is configured as an example as follows:

1. **NOTE**

   - The string type does not support reading or writing.
   - The background DINT is of the signed 32-bit type, so an error occurs when UDINT data is read. If there is no signed number with more than 32 bits, test whether the value read is correct. If decimal display is incomplete, use hexadecimal display. In hexadecimal display, unsigned 32-bit numbers are displayed. Device IDs and other UDINT type data can be viewed in hexadecimal display.
   - The ETC_ReadParameter and ETC_Write_Parameter instructions support reading and writing DINT type data, and do not support reading and writing 32-bit BYTE arrays (BYTE[32]).Run the ETC_ReadParameter and ETC_Write_Parameter instructions to set corresponding pin parameters to read and write corresponding indexes. This section describes operations of the ETC_Write_Parameter instruction, which are the same of the ETC_ReadParameter instruction.

2. (Optional) Read the I/O port configuration information of the IO-Link slave.
   1. Set "Index" to "0X41" (hexadecimal) or "65" (decimal). The default value of "Subindex" of Inovance IO-Link slave is 0, which does not need to be changed.

```
ETC_WriteParameter_CoE(Execute := M0,
                       SlaveID := K0,
                       Index := H4000,
                       SubIndex := K3,
                       DstLength := K4,
                       Data := H41,
                       Done => ,
                       Busy => ,
                       AbortCode => ,
                       Error => Error,
                       ErrorID => ErrorID);

ETC_WriteParameter_CoE(Execute := M0,
                       SlaveID := K0,
                       Index := H4000,
                       SubIndex := K4,
                       DstLength := K4,
                       Data := K0,
                       Done => ,
                       Busy => ,
                       AbortCode => ,
                       Error => Error1,
                       ErrorID => ErrorID1);
```

2. Set "Length" based on the index length of the IO-Link slave in bytes, for example, 2 bytes. For details, see the "Object List" section in the IO-Link Slave User Guide.

```
ETC_WriteParameter_CoE(Execute := M0,
                       SlaveID := K0,
                       Index := H4000,
                       SubIndex := K5,
                       DstLength := K2,
                       Data := K2,
                       Done => ,
                       Busy => ,
                       AbortCode => ,
                       Error => Error2,
                       ErrorID => ErrorID2);
```

3. Set "Control" to 0 and then to 3 to read the I/O port configuration information of the IO-Link slave.

```
ETC_WriteParameter_CoE(Execute := M0,
                       SlaveID := K0,
                       Index := H4000,
                       SubIndex := K1,
                       DstLength := K2,
                       Data := K0,
                       Done => ,
                       Busy => ,
                       AbortCode => ,
                       Error => Error3,
                       ErrorID => ErrorID3);

ETC_WriteParameter_CoE(Execute := M0,
                       SlaveID := K0,
                       Index := H4000,
                       SubIndex := K1,
                       DstLength := K2,
                       Data := K3,
                       Done => ,
                       Busy => ,
                       AbortCode => ,
                       Error => Error4,
                       ErrorID => ErrorID4);
```

3. Configure the I/O port information of the IO-Link slave.
   1. Set "Index" to "0X41" (hexadecimal) or "65" (decimal). The default value of "Subindex" of Inovance IO-Link slave is 0, which does not need to be changed.

```
ETC_WriteParameter_CoE(Execute := M0,
                       SlaveID := K0,
                       Index := H4000,
                       SubIndex := K3,
                       DstLength := K4,
                       Data := H41,
                       Done => ,
                       Busy => ,
                       AbortCode => ,
                       Error => Error,
                       ErrorID => ErrorID);

ETC_WriteParameter_CoE(Execute := M0,
                       SlaveID := K0,
                       Index := H4000,
                       SubIndex := K4,
                       DstLength := K4,
                       Data := K0,
                       Done => ,
                       Busy => ,
                       AbortCode => ,
                       Error => Error1,
                       ErrorID => ErrorID1);
```

2. Set "Length" based on the index length of the IO-Link slave in bytes. For details, see the "Object List" section in the IO-Link Slave User Guide.

```
ETC_WriteParameter_CoE(Execute := M0,
                       SlaveID := K0,
                       Index := H4000,
                       SubIndex := K5,
                       DstLength := K2,
                       Data := K2,
                       Done => ,
                       Busy => ,
                       AbortCode => ,
                       Error => Error2,
                       ErrorID => ErrorID2);
```

3. Set "Data". Value 1 indicates output and value 0 indicates input. For example, configure the low-order 8 bits of the IO-Link slave as output and the high-order 8 bits as input, and set "Data[0]" to 255 and "Data[1]" to 0.

```
ETC_WriteParameter_CoE(Execute := M0,
                       SlaveID := K0,
                       Index := H4000,
                       SubIndex := K6,
                       DstLength := K2,
                       Data := HFF,
                       Done => ,
                       Busy => ,
                       AbortCode => ,
                       Error => Error5,
                       ErrorID => ErrorID5);
```

4. Set "Control" to 0 and then to 2 to configure the I/O port information of the IO-Link slave.

```
ETC_WriteParameter_CoE(Execute := M0,
                       SlaveID := K0,
                       Index := H4000,
                       SubIndex := K1,
                       DstLength := K2,
                       Data := K0,
                       Done => ,
                       Busy => ,
                       AbortCode => ,
                       Error => Error6,
                       ErrorID => ErrorID6);

ETC_WriteParameter_CoE(Execute := M0,
                       SlaveID := K0,
                       Index := H4000,
                       SubIndex := K1,
                       DstLength := K2,
                       Data := K2,
                       Done => ,
                       Busy => ,
                       AbortCode => ,
                       Error => Error7,
                       ErrorID => ErrorID7);
```

6. Verify the configuration correctness.

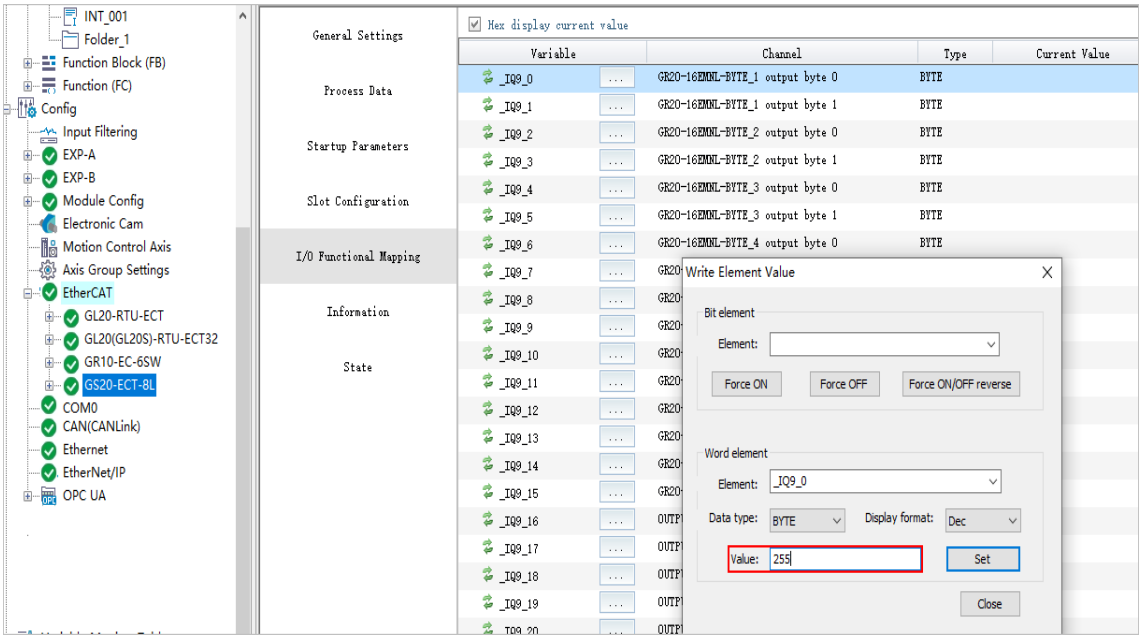
   After the configuration is completed, verify whether the configuration is correct. This section verifies whether port 0 and port 1 of GS20-ECT-8L are correctly configured based on the short connection between the low-order 8 bits (configured to output) and high-order 8 bits (configured to input) of GR20-16EMPL of the IO-Link slave.

   a. Use an I/O cable for short connection between the low-order 8 bits and high-order 8 bits of GR20-16EMPL.

   b. Verify whether port 0 of GS20-ECT-8L is correctly configured.

      α. Use an IO-Link communication cable to connect the IO-Link port of GR20-16EMPL to port 0 of GS20-ECT-8L.

      β. Make the system enter the monitoring mode, and double-click "GS20-ECT-8L". On the page that is displayed, click "IO Functional Mapping", and double-click the low-order 8 bit output data of port 0 of GS20-ECT-8L. In the dialog box that is displayed, set "Value" to 255 to output signal 1 for the eight input ports of the IO-Link slave of GR20-16EMPL.



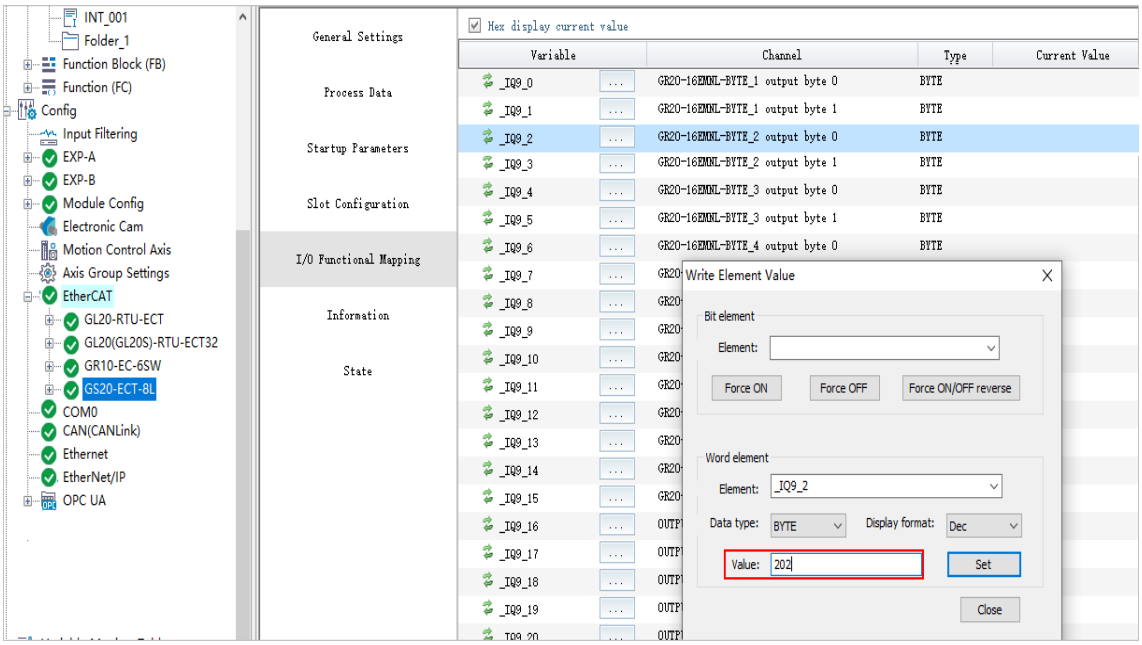
      γ. Check the high-order 8 bit input data of port 0 of GS20-ECT-8L and the status of I/O indicators on the IO-Link slave of GR20-16EMPL.

If the high-order 8 bit input data of port 0 of GS20-ECT-8L is 255 and all the I/O indicators on the IO-Link slave of GR20-16EMPL are turned on, the configuration is correct. Otherwise, the configuration is incorrect. Contact our technical support personnel for help.

c. Verify whether port 1 of GS20-ECT-8L is correctly configured.

α. Use an IO-Link communication cable to connect the IO-Link port of GR20-16EMPL to port 1 of GS20-ECT-8L.

β. Double-click "GS20-ECT-8L". On the page that is displayed, click "IO Functional Mapping", and double-click the low-order 8 bit output data of port 1 of GS20-ECT-8L. In the dialog box that is displayed, set "Value" to 202 as an example to output the "TRUE" or "FALSE" signal for the eight input ports of the IO-Link slave of GR20-16EMPL. By default, BYTE variables are mapped, and the binary code of 202 is 11001010, indicating that the first, third, sixth, and seventh bits are set to TRUE and other bits are set to FALSE.



γ. Check the high-order 8 bit input data of port 1 of GS20-ECT-8L and the status of I/O indicators on the IO-Link slave of GR20-16EMPL.

If the high-order 8 bit input data of port 1 of GS20-ECT-8L is the same as the low-order 8 bit output data, and the I/O indicators on the IO-Link slave of GR20-16EMPL are displayed accordingly, the configuration is correct. Otherwise, the configuration is incorrect. Contact our technical support personnel for help.

7.5.3 Fault Diagnosis

7.5.3.1 EtherCAT Fault Diagnosis

LED indicator		Meaning	Possible cause	Solution
RUN	OFF	The EtherCAT slave station is in the initialization state.	The EtherCAT master station is not connected to the EtherCAT slave station.	Check whether the configuration and parameter distribution are correct.

LED indicator		Meaning	Possible cause	Solution
				Check whether the communication address is configured properly. Check that the network cable specifications (M12 interface, with a shielded Cat5e network cable) and length (within 100 m) meet the requirements.
	Flashing in green	The EtherCAT slave station is in the pre-operational state.	The EtherCAT slave station is in a state other than OP.	Check whether the EtherCAT slave station configuration is correct. Check whether the EtherCAT slave station is faulty. Check whether all EtherCAT slave stations are configured.
	Flashing green only	The EtherCAT slave station is in the safe-operational state.		
ERR	Flashing in red	The EtherCAT communication network is abnormal.	The EtherCAT master station does not exchange data with the EtherCAT slave station. EtherCAT communication receives non-executable state transition commands. EtherCAT synchronization fails. A watchdog error occurs in EtherCAT communication.	Check whether the M12 network cable plus is inserted properly. Check whether the network cable is damaged. Check whether the PDO configuration is correct. Restart the power supply.

7.5.3.2 IO-Link Fault Diagnosis

Type	Error Code	Description
System	0x0002	US supply overvoltage
	0x0003	US supply undervoltage

Type	Error Code	Description
	0x0004	UA supply overvoltage
	0x0005	UA supply undervoltage
	0x0006	MCU temperature over 80°C
	0x000c	Pin2 short circuit
IO-Link master station	0x1800	Port connection to no device
	0x1801	Parameter loading failure
	0x1802	Invalid vendor ID
	0x1803	Invalid device ID
	0x1804	Pin4 short circuit
	0x1805	PHY chip overtemperature
	0x1806	Pin1 short circuit
	0x1807	Pin1 overcurrent
	0x1808	Slave device event overflow
	0x1811	Pin4 short circuit (DO mode)
	0x1813	Pin4 overcurrent (DO mode)
	0x6000	Invalid cycle time
	0x6001	Incorrect version for slave station

7.5.4 Object Lists

7.5.4.1 Process Data

The input data of the IO-Link master station indicates TxPDO of the IO-Link slave station. If an IO-Link slave with TxPDO is connected, the IO-Link master port must be configured with a mode with input data. For the definition of TxPDO data of the IO-Link slave, see the following table. N represents the port number of the IO-Link master.

Index 0xF100 channel n I/O-Link communication state (for 0 ≤ n ≤ 7)				
Index	Name	Meaning	Data type	Default
0xF100:00	Sub-index 00	Highest sub-index supported	USINT	0x08 (8dec)

0xF100:01	Sub-index 01 … Sub-index 08	Master and slave communication state display Bit 0 to bit 3: IO-Link state 0: Port inactive 1: Input mode 2: Output mode 3: Communication OP 4: Communication fault	Bit 4 to bit 7: Error code 00: No error 1: Watchdog error 2: Buffer overflow 3: Invalid device ID 4: Invalid vendor ID 5: Inactive version 6: Invalid frame function 7: Invalid cycle time 8: Invalid input process data length 9: Invalid output process data length 10: No device detected 11: N/A	USINT	0x00 (0dec)
…				…	…
0xF100:08				USINT	0x00 (0dec)

Index 0x20n0 Ch. n Pin2 state monitoring input process data (for 0 ≤ n ≤ 7)

Index	Name	Meaning	Data type	Default
0x20n0:00	Sub-index 00	Highest sub-index supported	USINT	0x02 (2dec)
0x20n0:01	Sub-index 01	Pin2 pin input process data 0: Inactive 1: Active	USINT	0x00 (0dec)
0x20n0:02	Sub-index 02	Short circuit state when pin2 is configured as output 0: No short circuit information 1: Short circuit	USINT	0x00 (0dec)

Index 0x20n1 Ch. N Pin4 and Pin1 short circuit state monitoring (for 0 ≤ n ≤ 7)

Index	Name	Meaning	Data type	Default
0x20n1:00	Sub-index 00	Highest sub-index supported	USINT	0x02 (2dec)
0x20n1:01	Sub-index 01	Short circuit state when pin4 is configured as output 0: No short circuit information 1: Short circuit	USINT	0x00 (0dec)

0x20n1:02	Sub-index 02	Pin1 short circuit state 0: No short circuit information 1: Short circuit	USINT	0x00 (0dec)
Index 0x2A02 Hardware state monitoring				
Index	Name	Meaning	Data type	Default
0x2A02:00	System hardware monitor	Highest sub-index supported	USINT	0x03 (3dec)
0x2A02:01	System hardware monitor_UA state_2A02_01	Actuator power supply voltage detection 00:18V < US < 30.2V 01: 1V < US < 18V 10: US > 30.2V 11: < 11V	USINT	0x00 (0dec)
0x2A02:02	System hardware monitor_US state_2A02_02	System power supply voltage detection 00:18V < US < 30.2V 01: 1V < US < 18V 10: US > 30.2V 11: < 11V	USINT	0x00 (0dec)
0x2A02:03	System hardware monitor_Temperature state_2A02_03	MCU internal temperature detection 00: 0℃ < Internal temperature < 85℃ 01: Internal temperature < 0℃ 10: Internal temperature > 85℃ 11: Reserved	USINT	0x00 (0dec)
Index 0x30n8 Ch.n Pin2 DO mode output process data (for 0 ≤ n ≤ 7)				
Index	Name	Meaning	Data type	Default
0x30n8:00	Sub-index 00	Highest sub-index supported	USINT	0x01 (1dec)
0x30n8:01	Sub-index 01	Pin2 output process data 0: Inactive 1: Active	BIT	0
Index 0x60n0 Ch. n IO-link input process data (for 0 ≤ n ≤ 7)				
Index	Name	Meaning	Data type	Default

0x60n0:00	Sub-index 00	Highest sub-index supported	USINT	0x00 (0dec)
0x60n0:01	TxPDO 01	IO-LINK input process data [00]	UDINT	0x00 (0dec)
…	…	…	…	…
0x60n0:20	TxPDO 32	IO-LINK input process data [32]	UDINT	0x00 (0dec)

Index 0x70n0 Ch. n output process data (for 0 ≤ n ≤ 7)				
Index	Name	Meaning	Data type	Default
0x70n0:00	Sub-index 00	Highest sub-index supported	USINT	0x00 (0dec)
0x70n0:01	RxPDO 01	IO-LINK output process data [00]	UDINT	0x00 (0dec)
…	…	…	…	…
0x70n0:20	RxPDO 32	IO-LINK output process data [32]	UDINT	0x00 (0dec)

The following table lists configuration data of the IO-Link port.

Index 0x20n3 Ch. n Pin4 used as the configuration parameter in DO mode (for 0 ≤ n ≤ 7)					
Index	Name	Meaning	Data type	Mark	Default
0x20n3:00	I/O setting of channels 1 to 8	Highest sub-index supported	USINT	RW	0x1 (1dec)
0x20n3:01	Pin4 safety state	Pin4 is configured as the preset value for the safety state upon output. 0x00: Output 0 upon communication error 0x01: Output 1 upon communication error 0x02: Last output value used upon communication error	UDINT	RW	0x00 (0dec)
Index 0x20n2 Ch. n Pin2 used as the configuration parameter in DO mode (for 0 ≤ n ≤ 7)					

Index	Name	Meaning	Data type	Mark	Default
0x20n2:00	I/O setting of channels 1 to 8	Highest sub-index supported	USINT	RW	0x1 (1dec)
0x20n2:01	Pin2 safety state	Pin2 is configured as the preset value for the safety state upon output. 0x00: Output 0 upon communication error 0x01: Output 1 upon communication error 0x02: Last output value used upon communication error	UDINT	RW	0x00 (0dec)

7.5.4.2 EtherCAT Object Dictionary Data (CoE Objects)

The object dictionary of the EtherCAT IO-Link master station contains SDO-based addressing objects and the standard objects and manufacturer objects supported by the IO-Link master station. The ETG.1000.6: Application Layer protocol specification describes standard objects and ETG.5001-6220 describes the modular equipment profile objects. In addition, the manufacturer objects can be addressed by combination of indexes and sub-indexes. Sub-index 0 indicates the number of sub-indexes or highest-level sub-indexes.

Index 0x1000 EtherCAT slave station device type					
Index	Name	Meaning	Data type	Mark	Default
1000: 00	Device type	Device type of the EtherCAT slave station	UDINT	RO	0x1389 (5001dec)
Index 0x1001 error register					
Index	Name	Meaning	Data type	Mark	Default
1001: 00	Error register	Error register	USINT	RO	0x00
Index 0x1001 EtherCAT slave station device name					

Index	Name	Meaning	Data type	Mark	Default
1008: 00	Device name	Device name	STRING(11)	RO	GS20-ECT-8L
Index 0x1008 EtherCAT slave station hardware version information					
Index	Name	Meaning	Data type	Mark	Default
1009: 00	Hardware version	Hardware version of the EtherCAT slave station	STRING(16)	RO	A00.01
Index 0x100A EtherCAT slave station protocol stack software version information					
Index	Name	Meaning	Data type	Mark	Default
100A:00	Slave version	Software version of the EtherCAT slave station protocol stack	STRING(4)	RO	5.13
Index 0x100B IO-LINK master station product-related version information					
Index	Name	Meaning	Data type	Mark	Default
100B:00	Software version	Highest sub-index supported	USINT	RO	0x04
100B:01	App version	Application software version	UDINT	RO	0x10100000
100B:02	FPGA version	FPGA firmware version	UDINT	RO	0x10100000
100B:03	IOLM version	Software version of the IO-Link master station	UDINT	RO	0x0305
Index 0x1018 EtherCAT slave station identity information					
Index	Name	Meaning	Data type	Mark	Default
1018: 00	Identity	Highest sub-index supported	USINT	RO	0x04 (4dec)
1018: 01	Vendor ID	Vendor ID of the EtherCAT slave station	UDINT	RO	0x00100000

1018: 02	Product code	Product code of the EtherCAT slave station	UDINT	RO	0x10F42EE1
1018: 03	Revision	Firmware version of the product application	STRING(11)	RO	1.1.0.0
1018: 04	Serial number	Production serial number	UDINT	RO	0x15FA66
Index 0x10F3 diagnosis history					
Index	Name	Meaning	Data type	Mark	Default
10F3:00	Diagnosis history	Highest sub-index supported	USINT	RO	0x16
10F3:01	Maximum message	Maximum number of diagnosis messages	USINT	RO	0x14 (20dec)
10F3:02	Newest message	Sub-index of the latest diagnosis message	USINT	RO	0x00000000 (0dec)
10F3:03	Newest acknowledged message	Latest acknowledged message	USINT	RW	0x00000000 (0dec)
10F3:04	New message available	New diagnosis information	BOOL	RO	0
10F3:05	Flags	Setting for send and store diagnosis messages	UINT	RW	0x00000000 (0dec)
10F3:06	Diagnosis message 01	Diagnosis information 1	ARRAY [0..27] OF BYTE	RO	0x00000000 (0dec)
…	…	…	…	…	…
10F3:40	Diagnosis message 64	Diagnosis information 64	ARRAY [0..27] OF BYTE	RO	0x00000000 (0dec)
Index 0x3010 ESC port 0 error counter					
Index	Name	Meaning	Data type	Mark	Default

3010: 00	Port0 error counter	Highest sub-index supported	USINT	RO	0x04 (4dec)
3010: 01	Port0 invalid frame counter	Number of invalid frames for ESC port 0	USINT	RO	0x00 (0dec)
3010: 02	Port0 Rx error counter	Number of error frames for ESC port 0	USINT	RO	0x00 (0dec)
3010: 03	Port0 forwarded Rx error counter	Number of error loopback frames for ESC port 0	USINT	RO	0x00 (0dec)
3010: 04	Port0 lost link counter	Number of lost frames for ESC port 0	USINT	RO	0x00 (0dec)
Index 0x3011 ESC port 0 error counter					
Index	Name	Meaning	Data type	Mark	Default
3011: 00	Port1 error counter	Highest sub-index supported	USINT	RO	0x04 (4dec)
3011: 01	Port1 invalid frame counter	Number of invalid frames for ESC port 1	USINT	RO	0x00 (0dec)
3011: 02	Port1 Rx error counter	Number of error frames for ESC port 1	USINT	RO	0x00 (0dec)
3011: 03	Port1 forwarded Rx error counter	Number of error loopback frames for ESC port 1	USINT	RO	0x00 (0dec)
3011: 04	Port1 lost link counter	Number of lost frames for ESC port 1	USINT	RO	0x00 (0dec)
Index 0x3012 ESC error counter					
Index	Name	Meaning	Data type	Mark	Default
3012: 00	ESC error counter	Highest sub-index supported	USINT	RO	0x04 (4dec)
3012: 01			USINT	RO	0x00 (0dec)

	ECAT processing unit error counter	Error counter for the ESC processing unit			
3012: 02	PDI error counter	PDI error counter	USINT	RO	0x00 (0dec)
3012: 03	Watchdog counter process data	Number of watchdogs for process data	USINT	RO	0x00 (0dec)
3012: 04	Watchdog counter PDI	Number of PDI watchdogs	USINT	RO	0x00 (0dec)

Index 0x3016 station address					
Index	Name	Meaning	Data type	Mark	Default
3016: 00	Station address	Highest sub-index supported	USINT	RO	0x04 (4dec)
3016: 01	Rotary switch value	Values set by DIP switch	USINT	RO	0x00 (0dec)
3016: 02	Configured station address	Configured station address	USINT	RO	0x00 (0dec)
3016: 03	Configured station alias	Configured station alias	USINT	RO	0x00 (0dec)
3016: 04	Alias in EEPROM	Alias in EEPROM	USINT	RO	0x00 (0dec)

The following table lists configuration data of the IO-Link port.

Index 0x80n0 Ch.n IO-Link port configuration data (for 0 ≤ n ≤ 7)					
Index	Name	Meaning	Data type	Mark	Default
0x80n0: 00	IO Settings Ch.1- 8	Highest sub-index supported	USINT	RW	0x28 (40dec)
0x80n0: 04	Device ID	ID of the IO-Link device	UDINT	RW	0x00000000 (0dec)
0x80n0: 05	VendorID	Vendor ID of the IO-Link device	UDINT	RW	0x00000000 (0dec)
0x80n0: 06	Product ID	Product ID of the IO-Link device	USINT	RW	0x00000000 (0dec)
0x80n0: 08	Serial number	Serial number of the IO-Link device	USINT	RW	0x00000000 (0dec)

0x80n0: 20	IO-Link revision	Version of the specification for IO-Link device communication Bit 0 to bit 3: Minor version Bit 4 to bit 7: Major version	USINT	RW	0x00 (0dec)
0x80n0: 21	Frame capability	Reserved	USINT	RW	0x00 (0dec)
0x80n0: 22	Min cycle time	Cycle time① between the IO-Link master station and slave station, which is transmitted by IO-Link data frames of the minimum cycle Bit 6 and bit 7: Time base Bit 0 and bit 5: Ratio 0x00: The IO-Link master station automatically uses the update time of the IO-Link device	USINT	RW	0x00 (0dec)
0x80n0: 23	Offset time	Reserved	USINT	RW	0x00 (0dec)
0x80n0: 24	Process data in length	Number of bits in the input process data transmitted by IO-Link data frames, which is recorded as 255 for 256 bits	USINT	RW	0x00 (0dec)
0x80n0: 25	Process data out length	Number of bits in the output process data transmitted by IO-Link data frames, which is recorded as 255 for 256 bits	USINT	RW	0x00 (0dec)
0x80n0: 26	Compatible ID	Reserved	UINT	RW	0x0000 (0dec)
0x80n0: 27	Reserved	Reserved	UINT	RW	0x0000 (0dec)
0x80n0: 28	Master control	Bit 0 to bit 3: 0: Port not used 1: Port configured as DI mode 2: Port configured as the slave mode 3: Port configured as the IO-Link automatic mode 4: Port configured as the IO-Link verification mode Bit 4 to bit 7:	UINT	RW	0x0000 (0dec)

		0: Invalid process data upon network disconnection 1: Output 0 upon network disconnection 2: Output of the last cycle upon network disconnection Bit 8 to bit 15: 0: Not verifying device parameter information 1: Verifying Vendor ID DEVICE ID for V1.0 2: Verifying Vendor ID DEVICE ID for V1.1 3: Enabling backup and restoration of slave station configuration data 4: Only enabling restoration of slave station configuration data * V1.0 and V1.1 indicate the versions of the IO-Link slave station protocol stack. * If the modes specified by values 3 and 4 are used, the port must be configured to the verification mode.			

Index 0x90n0 Ch. n IO-link port configuration data (for 0 ≤ n ≤ 7)

Index	Name	Meaning	Data type	Mark	Default
0x90n0: 00	IO Settings Ch.1- 8	Highest sub-index supported	USINT	R	0x28 (40dec)
0x90n0: 04	Device ID	Obtained device ID of the IO-Link slave device	UDINT	R	0x00000000 (0dec)
0x90n0: 05	VendorID	Obtained vendor ID of the IO-Link slave device	UDINT	R	0x00000000 (0dec)
0x90n0: 06	Product ID	Obtained product ID of the IO-Link slave device (not supported)	USINT	R	0x00000000 (0dec)
0x90n0: 08	Serial number	Obtained serial number of the IO-Link slave device (not supported)	USINT	R	0x00000000 (0dec)
0x90n0: 20	IO-Link revision	Version of the specification for IO-Link device communication Bit 0 to bit 3: Minor version Bit 4 to bit 7: Major version	USINT	R	0x00 (0dec)

0x90n0: 21	Frame capability	Reserved	USINT	R	0x00 (0dec)
0x90n0: 22	Min cycle time	Cycle time① between the IO-Link master station and slave station, which is transmitted by IO-Link data frames of the minimum cycle Bit 6 and bit 7: Time base Bit 0 and bit 5: Ratio	USINT	R	0x00 (0dec)
0x90n0: 23	Offset time	Reserved	USINT	R	0x00 (0dec)
0x90n0: 24	Process data in length	Obtained number of bits in the input process data transmitted by IO-Link slave station data frames, which is displayed as 255 for 256 bits	USINT	R	0x00 (0dec)
0x90n0: 25	Process data out length	Obtained number of bits in the output process data transmitted by IO-Link slave station data frames, which is displayed as 255 for 256 bits	USINT	R	0x00 (0dec)
0x90n0: 26	Compatible ID	Reserved	UINT	R	0x0000 (0dec)
0x90n0: 27	Reserved	Reserved	UINT	R	0x0000 (0dec)

① For details about the cycle time, see the following table.

Time baseline code	Time baseline value	Calculation method	Cycle time
00	0.1 ms	Multiplier x Time baseline	0.4 ms to 6.3 ms
01	0.4 ms	6.4 ms + Multiplier x Time baseline	6.4 ms to 31.6 ms
10	1.6 ms	32.0 ms + Multiplier x Time baseline	32.0 ms to 132.8 ms
11	Reserved	Reserved	Reserved

7.5.4.3 Configuration Data for Process Data Communication

EtherCAT PDO communication is managed by PDO assignment, PDO mapping, and process data object dictionaries. PDO assignment and PDO mapping are described and sampled as follows:

PDO assignment

PDO assignment, which is divided into two object dictionaries to receive PDO assignment and transmit PDO assignment, is used to configure PDO mapping. Indexes for the two dictionaries are respectively 0x1C12 and 0x1C13.

Index 0x1C12 RxPDO assignment				
Index	Name	Meaning	Data type	Default
0x1C12:00	Sub-index 00	Assignment of output process data	USINT	0x08 (8dec)
0x1C12:01	Sub-index 01	Assignment of output process data of index 1	DT1C12ARR	0x1600 (5632dec)
0x1C12:02	Sub-index 02	Assignment of output process data of index 2	DT1C12ARR	0x1601 (5633dec)
0x1C12:03	Sub-index 03	Assignment of output process data of index 3	DT1C12ARR	0x1602 (5634dec)
0x1C12:04	Sub-index 04	Assignment of output process data of index 4	DT1C12ARR	0x1603 (5633dec)
0x1C12:05	Sub-index 05	Assignment of output process data of index 5	DT1C12ARR	0x1604 (5634dec)
0x1C12:06	Sub-index 06	Assignment of output process data of index 6	DT1C12ARR	0x1605 (5635dec)
0x1C12:07	Sub-index 07	Assignment of output process data of index 7	DT1C12ARR	0x1606 (5634dec)
0x1C12:08	Sub-index 08	Assignment of output process data of index 8	DT1C12ARR	0x1607 (5635dec)
Index 0x1C13 TxPDO assignment				
Index	Name	Meaning	Data type	Default
0x1C13:00	Sub-index 00	Assignment of input process data	USINT	0x08 (8dec)

0x1C13:01	Sub-index 01	Assignment of input process data of index 1	DT1C13ARR	0x1A00 (6656dec)
0x1C13:02	Sub-index 02	Assignment of input process data of index 2	DT1C13ARR	0x1A01 (6657dec)
0x1C13:03	Sub-index 03	Assignment of input process data of index 3	DT1C13ARR	0x1A02 (6657dec)
0x1C13:04	Sub-index 04	Assignment of input process data of index 4	DT1C13ARR	0x1A03 (6658dec)
0x1C13:05	Sub-index 05	Assignment of input process data of index 5	DT1C13ARR	0x1A04 (6658dec)
0x1C13:06	Sub-index 06	Assignment of input process data of index 6	DT1C13ARR	0x1A05 (6659dec)
0x1C13:07	Sub-index 07	Assignment of input process data of index 7	DT1C13ARR	0x1A06 (6659dec)
0x1C13:08	Sub-index 08	Assignment of input process data of index 8	DT1C13ARR	0x1A07 (6660dec)

PDO mapping

PDO mapping is used to map the process data object dictionary that requires communication. PDO mapping is divided into RxPDO mapping and TxPDO mapping, and the index ranges for them are respectively 0x1600 to 0x17FF and 0x1A00 to 0x1BFF.

The object dictionary for PDO mapping contains the values of indexes, sub-indexes, and length of process data in the object dictionary for PDO communication.

Bit	31	……	16	15	……	8	7	……	0
Meaning	Index			Sub-index			Object Length		

Index 0x1A0n Ch. n output process data (for 0 ≤ n ≤ 7)				
Index	Name	Meaning	Data type	Default
0x1A0n:00	Sub-index 00	Input process data mapping	USINT	0x00 (0dec)

0x1A0n:01	Sub-index 01	1. Input process data mapping	UDINT	0x70n0:01.08
…	…	…	…	…
0x1A0n:40	Sub-index 64	64. Input process data mapping	UDINT	0x70n0:40.08

Index 0x1A10 Pin2 input process data mapping (8 Ch)

Index	Name	Meaning	Data type	Default
0x1A10:00	Sub-index 00	Input process data mapping for Pin2	USINT	0x08 (8dec)
0x1A10:01	Sub-index 01	Input process data mapping for Pin2 through CH0	BIT	0x2000:01.01
0x1A10:02	Sub-index 02	Input process data mapping for Pin2 through CH1	BIT	0x2010:01.01
0x1A10:03	Sub-index 03	Input process data mapping for Pin2 through CH2	BIT	0x2020:01.01
0x1A10:04	Sub-index 04	Input process data mapping for Pin2 through CH3	BIT	0x2030:01.01
0x1A10:05	Sub-index 05	Input process data mapping for Pin2 through CH4	BIT	0x2040:01.01
0x1A10:06	Sub-index 06	Input process data mapping for Pin2 through CH5	BIT	0x2050:01.01
0x1A10:07	Sub-index 07	Input process data mapping for Pin2 through CH6	BIT	0x2060:01.01
0x1A10:08	Sub-index 08	Input process data mapping for Pin2 through CH7	BIT	0x2070:01.01

Index 0x1A13 Pin1 short circuit process data mapping (8 Ch)

Index	Name	Meaning	Data type	Default
0x1A13:00	Sub-index 00		USINT	0x08 (8dec)

		Short circuit process data mapping for Pin1		
0x1A13:01	Sub-index 01	Short circuit process data mapping for Pin1 through CH0	BIT	0x2001:02.01
0x1A13:02	Sub-index 02	Short circuit process data mapping for Pin1 through CH1	BIT	0x2011:02.01
0x1A13:03	Sub-index 03	Short circuit process data mapping for Pin1 through CH2	BIT	0x2021:02.01
0x1A13:04	Sub-index 04	Short circuit process data mapping for Pin1 through CH3	BIT	0x2031:02.01
0x1A13:05	Sub-index 05	Short circuit process data mapping for Pin1 through CH4	BIT	0x2041:02.01
0x1A13:06	Sub-index 06	Short circuit process data mapping for Pin1 through CH5	BIT	0x2051:02.01
0x1A13:07	Sub-index 07	Short circuit process data mapping for Pin1 through CH6	BIT	0x2061:02.01
0x1A13:08	Sub-index 08	Short circuit process data mapping for Pin1 through CH7	BIT	0x2071:02.01
Index 0x1A14 system hardware status monitoring process data mapping				
Index	Name	Meaning	Data type	Default
0x1A14:00	Sub-index 00	Process data mapping for status monitoring of system hardware	USINT	0x03 (3dec)
0x1A14:01	Sub-index 01	Process data mapping for status monitoring of system power voltage	UDINT	0x2A02:01.08
0x1A14:02	Sub-index 02		UDINT	0x2A02:02.08

		Process data mapping for status monitoring of activator power voltage		
0x1A14:03	Sub-index 03	Process data mapping for status monitoring of MCU internal temperature	UDINT	0x2A02:03.08
Index 0x1A80 master and slave station communication status monitoring process data mapping				
Index	Name	Meaning	Data type	Default
0x1A81:00	Sub-index 00	Process data mapping for status monitoring of master and slave station communication	USINT	0x08 (8dec)
0x1A81:01	Sub-index 01	Process data mapping for status monitoring of master and slave station communication through CH0	UDINT	0xF100:01.08
0x1A81:02	Sub-index 02	Process data mapping for status monitoring of master and slave station communication through CH1	UDINT	0xF100:02.08
0x1A81:03	Sub-index 03	Process data mapping for status monitoring of master and slave station communication through CH2	UDINT	0xF100:03.08
0x1A81:04	Sub-index 04	Process data mapping for status monitoring of master and slave	UDINT	0xF100:04.08

		station communication through CH3		
0x1A81:05	Sub-index 05	Process data mapping for status monitoring of master and slave station communication through CH4	UDINT	0xF100:05.08
0x1A81:06	Sub-index 06	Process data mapping for status monitoring of master and slave station communication through CH5	UDINT	0xF100:06.08
0x1A81:07	Sub-index 07	Process data mapping for status monitoring of master and slave station communication through CH6	UDINT	0xF100:07.08
0x1A81:08	Sub-index 08	Process data mapping for status monitoring of master and slave station communication through CH7	UDINT	0xF100:08.08
Index 0x160n Ch. n output process data (for 0 ≤ n ≤ 7)				
Index	Name	Meaning	Data type	Default
0x160n:00	Sub-index 00	Output process data mapping	USINT	0x00 (0dec)
0x160n:01	Sub-index 01	1. Output process data mapping	UDINT	0x60n0:01.08
…	…	…	…	…
0x160n:20	Sub-index 32	32. Output process data mapping	UDINT	0x60n0:40.08

Index 0x1620 Pin2 output process data mapping (8 Ch)				
Index	Name	Meaning	Data type	Default
0x1620:00	Sub-index 00	Output process data mapping for Pin2	USINT	0x08 (8dec)
0x1620:01	Sub-index 01	Output process data mapping for Pin2 through CH0	BIT	0x3008:01.01
0x1620:02	Sub-index 02	Output process data mapping for Pin2 through CH1	BIT	0x3018:01.01
0x1620:03	Sub-index 03	Output process data mapping for Pin2 through CH2	BIT	0x3028:01.01
0x1620:04	Sub-index 04	Output process data mapping for Pin2 through CH3	BIT	0x3038:01.01
0x1620:05	Sub-index 05	Output process data mapping for Pin2 through CH4	BIT	0x3048:01.01
0x1620:06	Sub-index 06	Output process data mapping for Pin2 through CH5	BIT	0x3058:01.01
0x1620:07	Sub-index 07	Output process data mapping for Pin2 through CH6	BIT	0x3068:01.01
0x1620:08	Sub-index 08	Output process data mapping for Pin2 through CH7	BIT	0x3078:01.01

7.5.4.4 IO-Link Slave Configuration Data

Index 0x40n0 Ch.n IO-Link port slave configuration parameter read/write (for 0 ≤ n ≤ 7)					
Index	Name	Meaning	Data type	Mark	Default
0x40n0:00	subindex0	Highest sub-index supported	USINT	RO	0x07 (7dec)
0x40n0:01	Control		UDINT	RW	

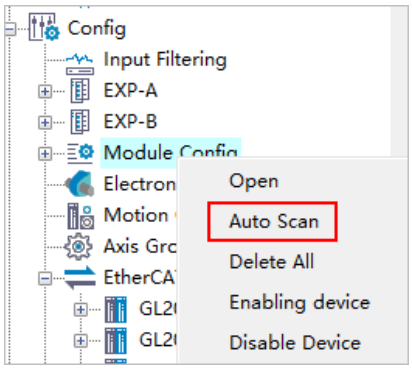
		0: No operation 0→2: Write parameter 0→3: Read parameter			0x00 (0dec)
0x40n0:02	Status	0x00: No error 0x02: Read success 0x40: Error	UDINT	RW	0x00 (0dec)
0x40n0:03	Index	Index number of the slave station	USINT	RW	0x00 (0dec)
0x40n0:04	Subindex	Sub-index number of the slave station	USINT	RW	0x00 (0dec)
0x40n0:05	Length	Data length (in byte)	UINT	RW	0x00 (0dec)
0x40n0:06	Data	Data	ARRAY [0..31] OF BYTE	RW	0x00 (0dec)
0x40n0:07	Error code	0x1: Operation not supported 0x3: Device access failure 0x4: Unauthorized operation 0x5: Slave device in non-OP state 0x34: Length error 0x36: Invalid operation due to busy master station 0x39: Port disabled	UINT	RO	0x00 (0dec)

7.6 Basic Functions

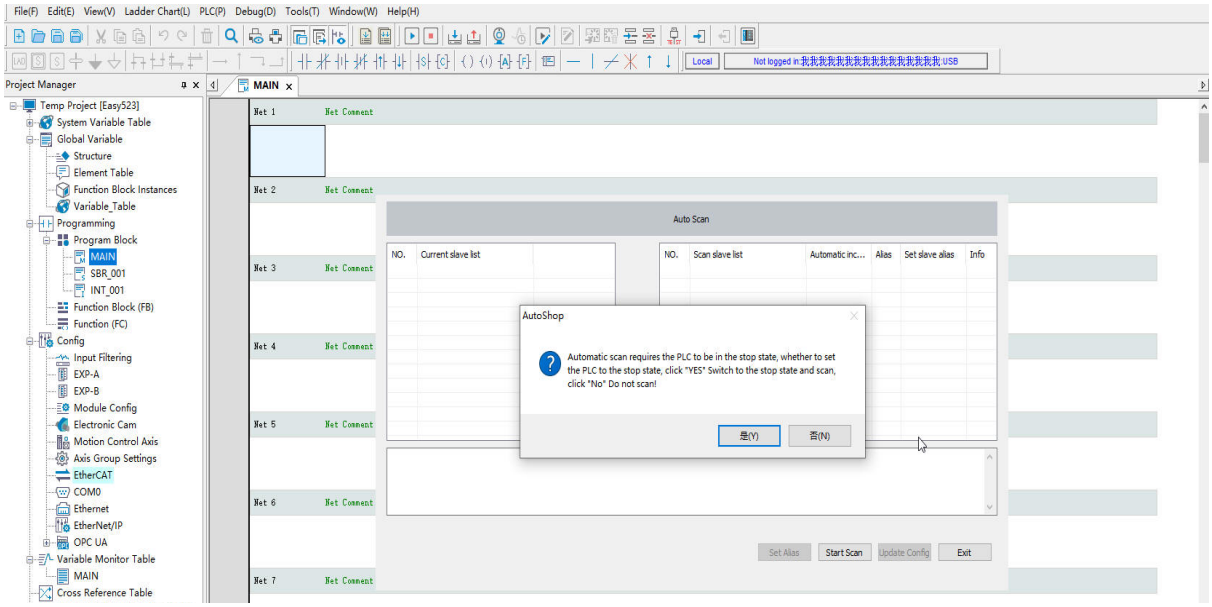
7.6.1 Auto Scan (Easy)

1. In the "Project Manager" tree on the left, right-click "Module Config" and then select "Auto Scan".

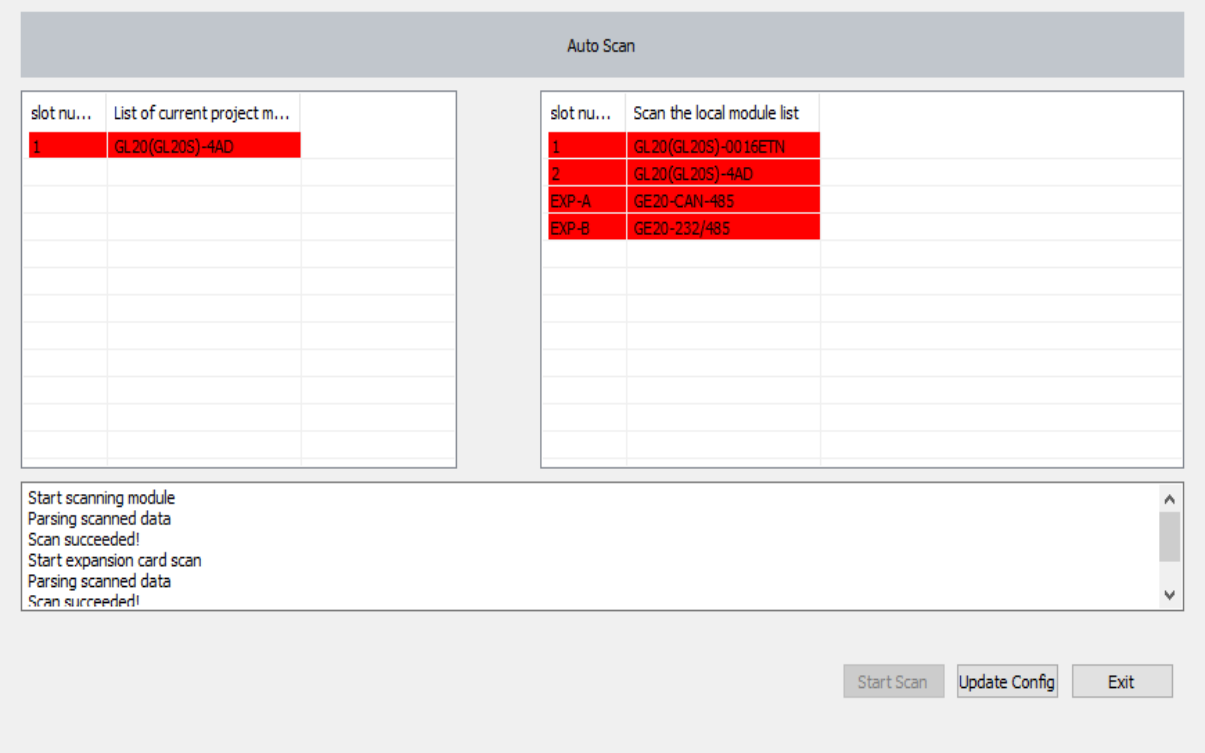
> **NOTE**
>
> Both modules and extension cards are scanned and the scanning follows the same specifications.



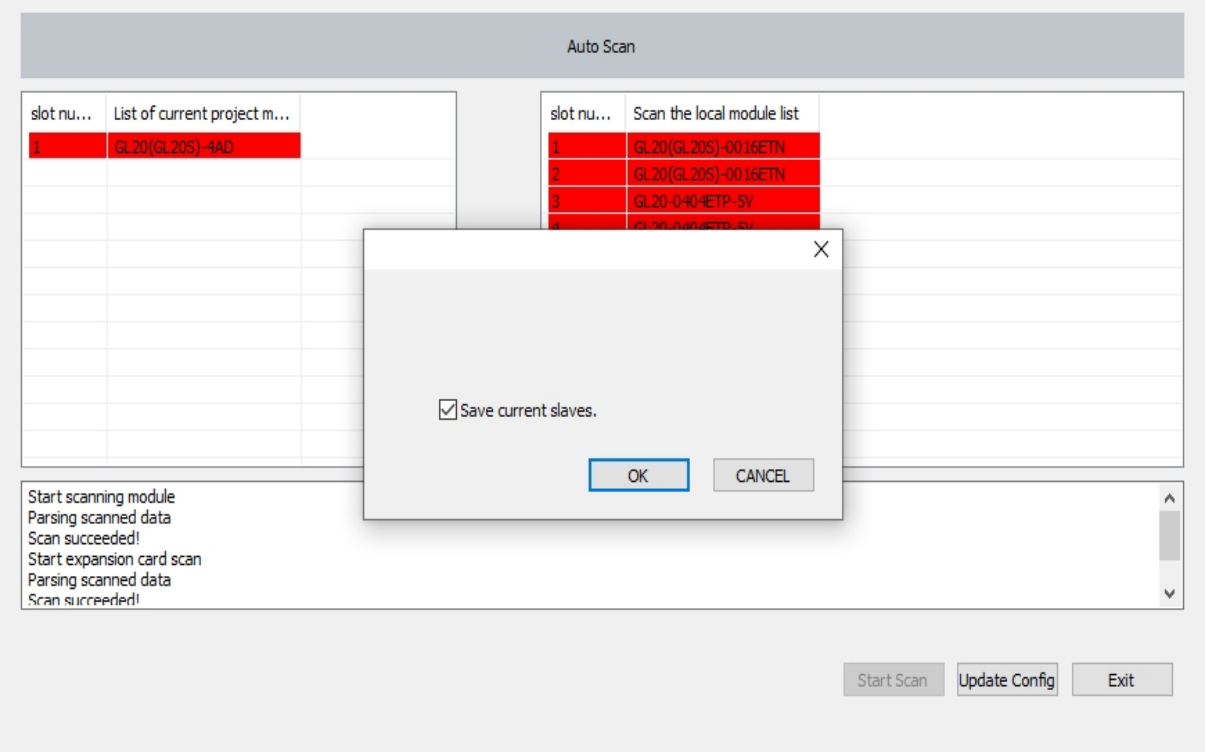
2. In the dialog box displayed, click "Start Scan". If the PLC is running, click "Yes" in the pop-up window to switch to the stopped state.



3. Click "Start Scan". After the scanning is completed, the list of scanned modules is displayed. If the mounted modules are inconsistent with the modules configured in the background, they are marked red.



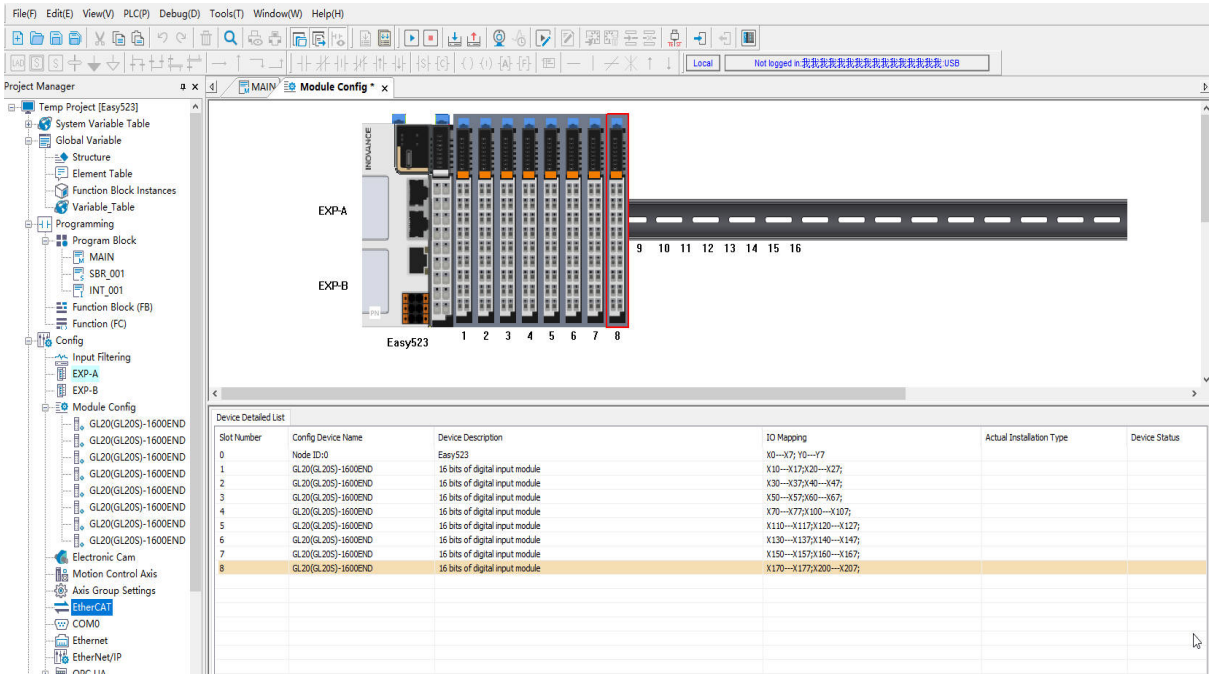
4. Click "Update Config".



The "Save current axises." window is described as follows:

- Deselected: All the background module configurations are deleted. The default configurations of scanned modules are added according to the scan result.
- Selected: Compare the mounted modules with the modules configured in the background. If they are consistent, the mounted modules are not replaced or updated. If they are inconsistent, the background modules of the corresponding slots are deleted, the default configured modules are added, and the modules are compared in sequence by slot.

Click "OK" to add the scan result, as shown in the following figure.

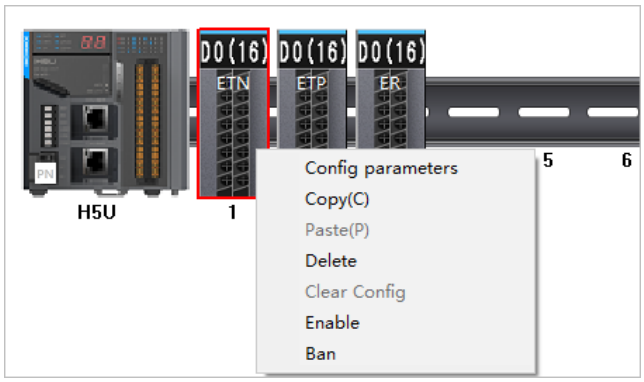


7.6.2 Disabling (H5U and Easy)

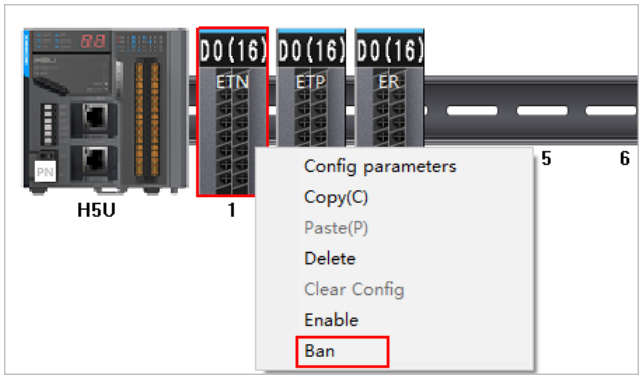
7.6.2.1 Disabling Module Through Right-Click Menu

To use the feature of disabling local extension modules, ensure that the physical extension modules removed are the same as the modules to be disabled in the module list of the software. The procedure of disabling an H5U extension module is as follows:

1. In the module list, select the target module, and right-click the module.



2. In the shortcut menu displayed, select "Ban", and download the program to the PLC device.



3. After downloading the program, restart the PLC device.

7.6.2.2 Disabling Module Through System Variables

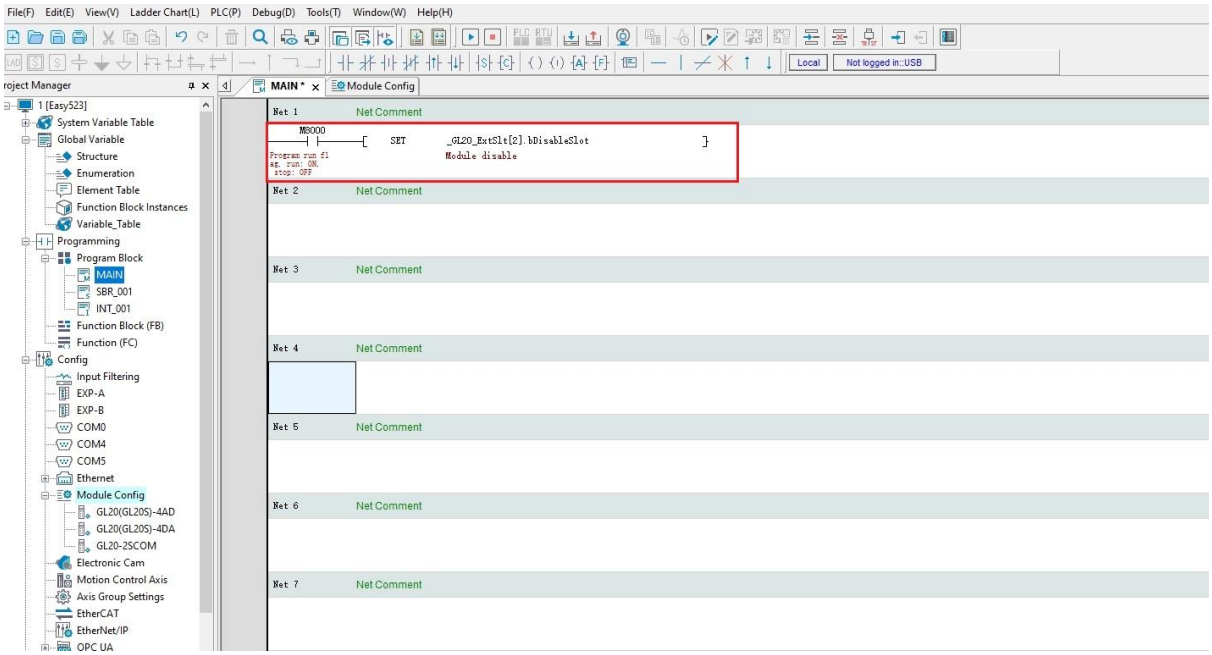
Procedure

The system variable for disabling local extension modules in the Easy series is: _GL20_ExtSlt[i].bDisableSlot. The system variable for disabling H5U local modules is _ExtSlt[i].bDisableSlot. Both of these system variables are of Boolean type. When the value is ON, the i-th module is disabled. When the value is OFF, the i-th module is not disabled. Here, i starts from 0, where 0 represents the first module, and so on.

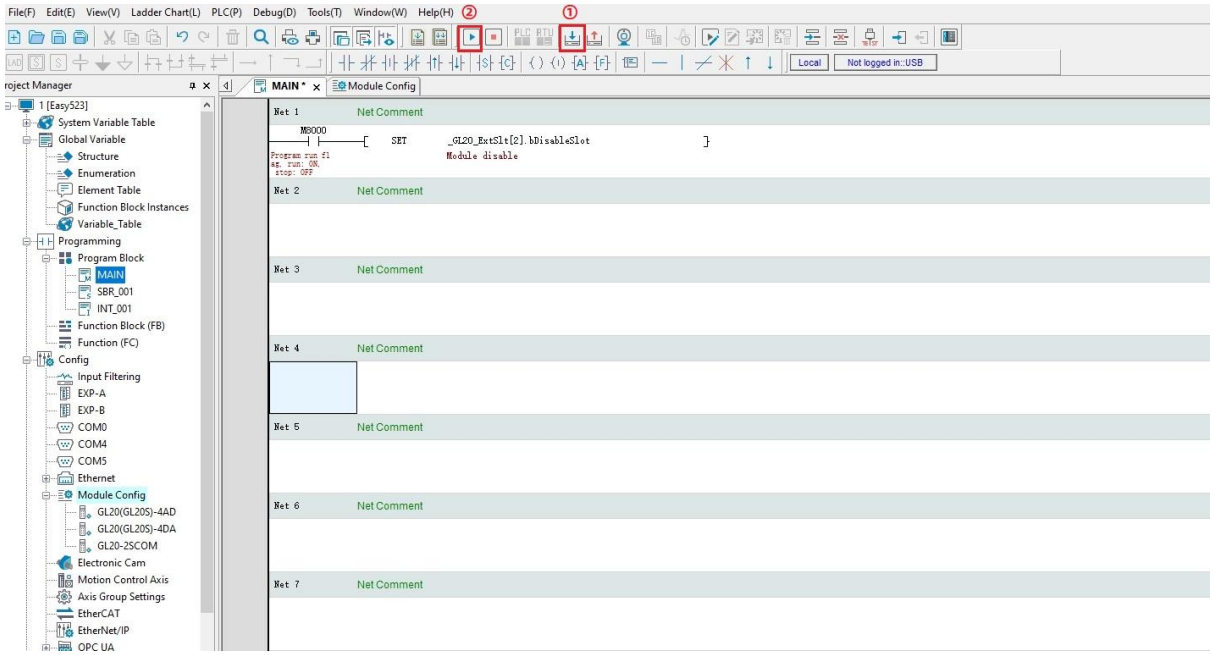
> CAUTION
>
> The system variables for disabling modules take effect only during the first scan cycle of the PLC operation.

Take the Easy523 and the GL20-2SCOM communication interface module as an example.

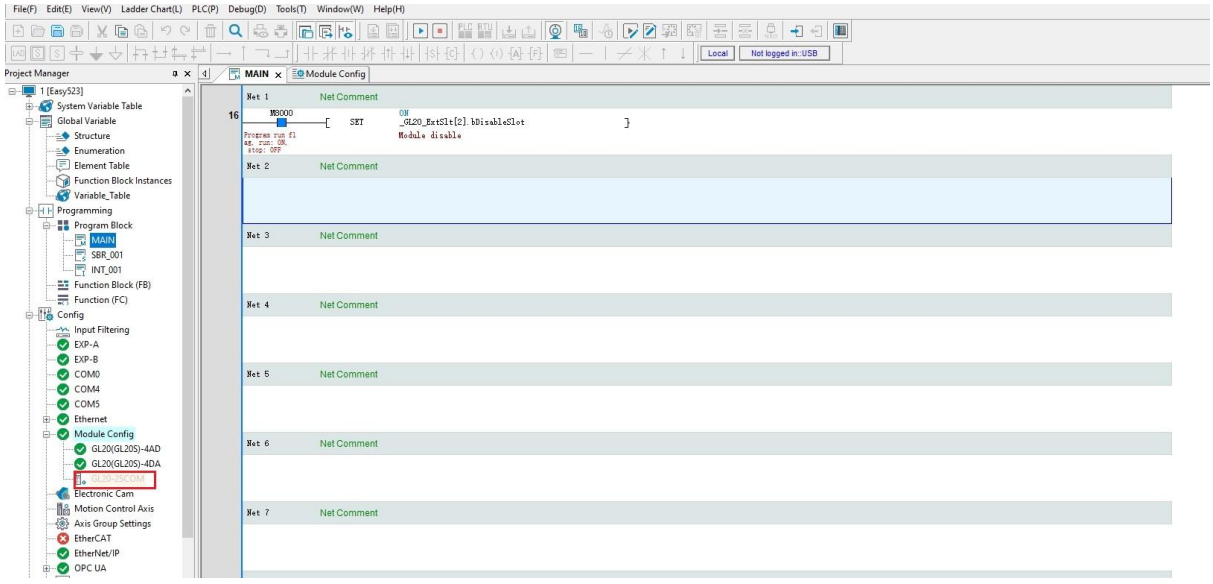
1. To disable the third module, assign ON to the system variable _GL20_ExtSlt[2].bDisableSlot during the first scan cycle after startup (ON means disabled, and OFF means not disabled), as shown in the following figure.



2. Click the icon at (1) to download the program to the PLC. Click the icon at (2) to run the PLC, as shown in the following figure.



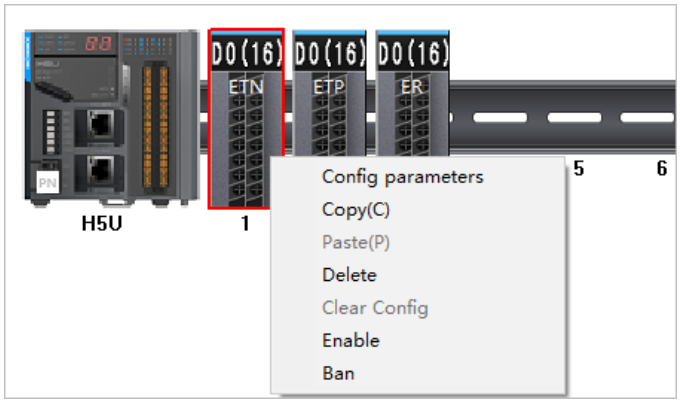
3. Click "Monitor". After the GL20-2SCOM module is disabled, its name will be grayed out, indicating that the module has been successfully disabled. Other modules start up normally, as shown in the following figure.



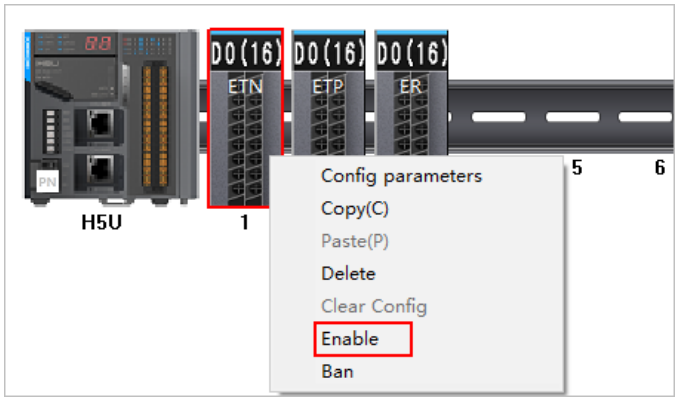
7.6.3 Enabling an Extension Module (H5U&Easy)

To use the feature of enabling local extension modules, ensure that the physical extension modules installed are the same as the modules to be enabled in the module list of the software. The procedure of enabling an H5U extension module is as follows:

1. In the module list, select the target module, and right-click the module.



2. In the shortcut menu displayed, select "Enable", and download the program to the PLC device.



3. After downloading the program, restart the PLC device.

8 Serial Communication

8.1 Overview

H5U provides a serial communication port that supports baud rates of 9600 bps, 19200 bps, 38400 bps, 57600 bps, and 115200 bps.

Serial ports supported by the Easy series are listed in the following table.

Type	Product	Serial Communication
Easy300 series	Easy301	One RS232 port and one RS485 port, supporting free protocols for serial ports and the Modbus protocol
	Easy302	One RS232 port and one RS485 port, supporting up to nine RS232/RS485 extension cards and extension modules; supporting free protocols for serial ports and the Modbus protocol
	Easy320	One RS485 port, supporting up to ten RS232/RS485 extension cards and extension modules; supporting free protocols for serial ports and the Modbus protocol
Easy500 series	Easy501	One RS485 port, supporting up to ten RS232/RS485 extension cards and extension modules; supporting free protocols for serial ports and the Modbus protocol
	Easy502	
	Easy521	
	Easy522	
	Easy523	

Communication Protocol	Description
Free protocol	Freely sends/receives data with the SerialSR instruction
Modbus-RTU master	A standard Modbus-RTU master that reads/writes data from/to a slave station through Modbus configuration
Modbus-RTU slave	A standard Modbus-RTU slave
Modbus-ASCII master	A standard Modbus-ASCII master that reads/writes data from/to a slave through Modbus configuration
Modbus-ASCII slave	A standard Modbus-ASCII slave

8.2 Serial Communication Network

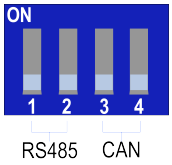
You are recommended to use shielded twisted pairs for the RS485 bus and use twisted pairs to connect RS485+ and RS485-. A 120 Ω termination resistor is connected at both ends of the bus to prevent signal

reflection. The reference grounds (GND) of RS485 signals on all nodes are connected together. A maximum of 31 nodes are supported and the distance between each node and the bus must be less than 3 m.

H5U series

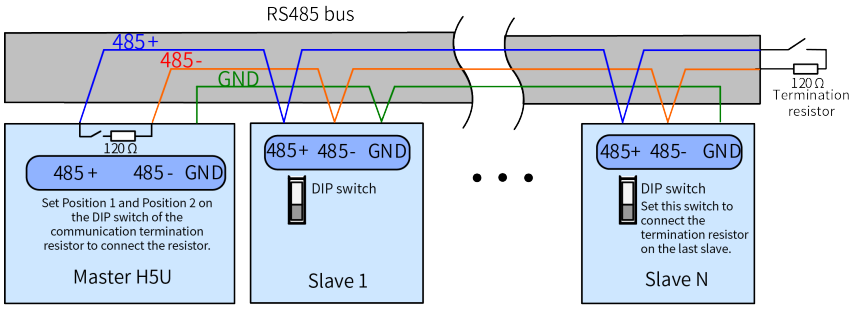
Communication termination resistor DIP switch

The communication termination resistor DIP switch is located in the battery bay. ON means the termination resistor is connected (factory default: OFF). The switch schematic diagram is as follows, in which 1 and 2 are used for RS485 communication, and 3 and 4 are used for CAN communication.



Networking of RS485 serial communication

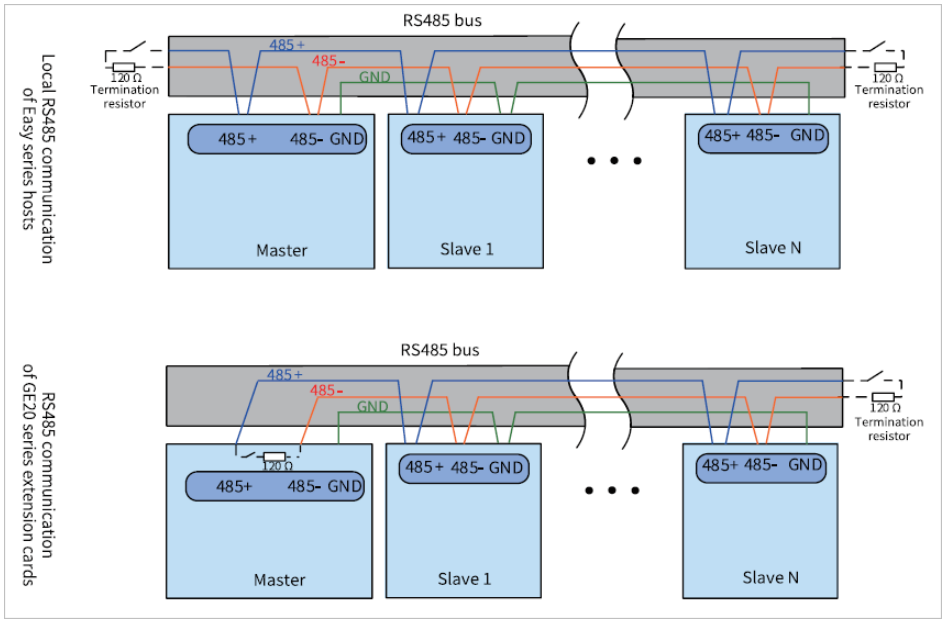
The following figure shows the RS485 bus topology.



The RS485 port of H5U has a 120 Ω termination resistor which can be turned on or off by setting the DIP switch.

Easy series

The Easy series host has no termination resistor. An external termination resistor can be connected if necessary. The RS485 communication extension card of the GE20 series has a termination resistor. The termination resistor can be turned on or off by setting the DIP switch and the default setting is OFF. The following figure shows the RS485 bus topology.

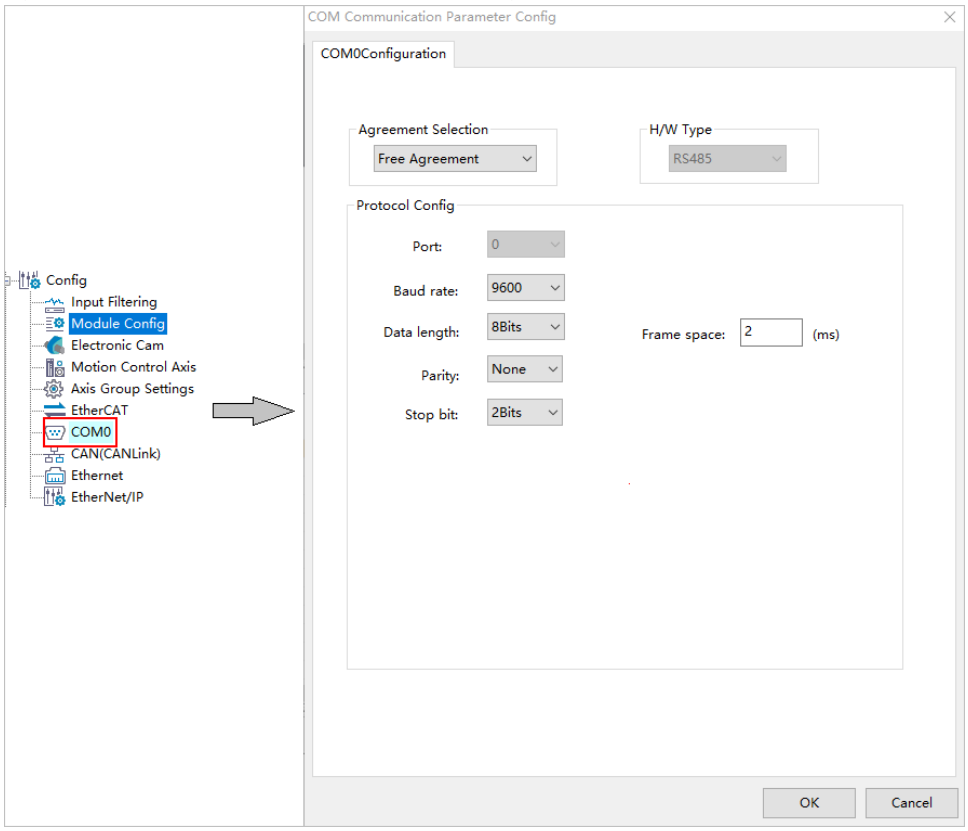


8.3 Free Protocol Configuration

8.3.1 Configuring the Free Protocol

After the free protocol is configured, you can use the SerialSR instruction to send and receive data in the user program.

1. In the "Project Manager" tree on the left, double-click "COM". In the dialog box displayed, select "Free Agreement" from the drop-down list under "Agreement Selection".



2. Set the serial port parameters, and click "OK".
   Then, you can use the SerialSR instruction to send and receive data in the user program.

8.3.2 (Optional) Disabling the Free Protocol

When the free protocol is set for the COM port, you can use the SerialSR instruction to send and receive data over the free protocol. To disable the free protocol, set the system variable _SerialSR.abort to abort a free protocol send/receive process. The modification takes effect immediately, as shown in the following figure.

Variable Name	Initial Value	Type	Comment
_SerialSR		_sSerial_SR	Serial port Free Protocol Inf
port	0	INT	Serial\/Port Number
states	0	INT	Running state
sent	0	INT	Bytes Sent
received	0	INT	Bytes Received
mutexF	0	DINT	Interlock flag
trigger	0	DINT	Trigger Flag
errorid	0	DINT	Error message
timeout	0	DINT	Timeout MS
sendlen	0	DINT	Bytes Sent Length
sendbuf		INT[256]	Send Buffer
recvlen	0	DINT	Bytes of Received Length
recvbuf		INT[256]	Receive Buffer
abort	0	INT	Serial Free Protocol process
startchar_en	0	INT	Receive start characters en
startchar		BYTE[4]	Receive start characters
endchar_en	0	INT	Receive end characters ena
endchar		BYTE[4]	Receive end characters
Bytetimeout_en	OFF	BOOL	Receive idle frame break en
Bytetimeout	0	INT	Receive idle frame break tin

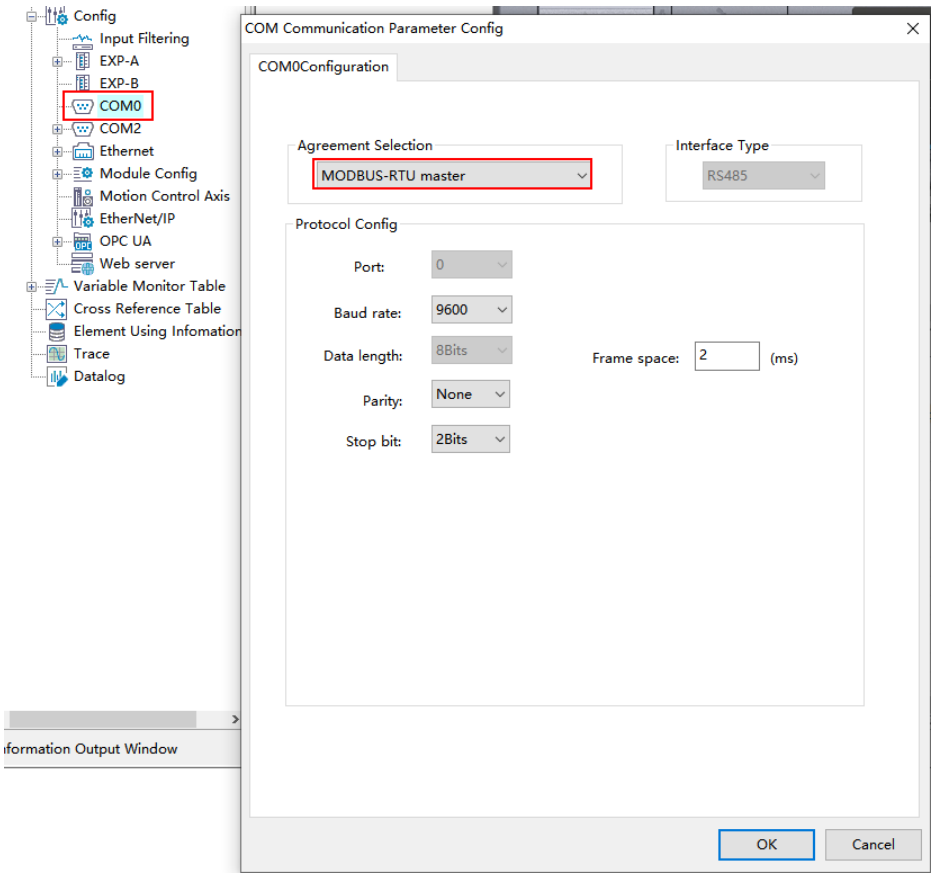
When _SerialSR.abort is set to a non-zero value, the send and receive processes can be aborted within the specified timeout period. After the processes are aborted, their states change to 16-Completed, and the DONE signal is ON.

8.4 Master Configuration

8.4.1 Configuring Modbus-RTU and Modbus-ASCII Masters

1. Configure the serial port protocol.

   In the "Project Manager" tree on the left, double-click "COM0". In the dialog box displayed, select "MODBUS-RTU master" or "MODBUS-ASC master", and set serial port parameters, as shown in the following figure.



NOTE

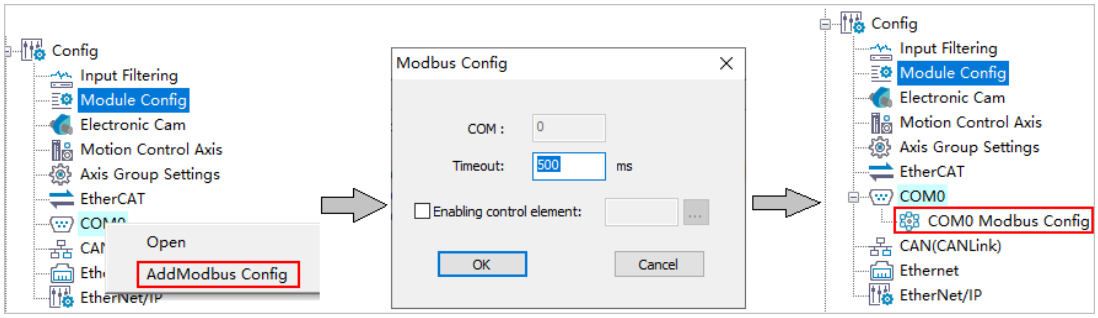
The following devices support low baud rates (600 bps, 1200 bps, 2400 bps, and 4800 bps):

- Easy500 series and Easy300 series models and GE20-232/485 extension cards.
- Software version: AutoShop V4.11.0.0 and later
- Firmware version: V6.5.0.0 and later

The following table list communication protocol and serial port basic parameters.

Parameter Name	Description
Protocol selection	Agreement Selection Options are None, Free Protocol, Modbus-RTU Master, MODBUS-RTU Slave//HMI Download Protocol, MODBUS-ASC Master, and MODBUS-ASC Slave.
H/W Type	The type of hardware of other devices communicating with the module. Options are RS232, RS485, and RS422.
Port	The number of the serial port for communication connection.
Baudrate	The communication baud rate. Options are 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200 Default: 9600
Data length	The actual data bits included in communication frames.
Parity	The method of verifying communication frames. Options are None, Odd, and Even. Default: Even
Frame interval	The interval between receiving a data frame and sending the next data frame.
Stop bit	The last bit in a single message during communication. Options are 1 and 2. Default: 1

2. Add the Modbus configurations, as shown in the following figure.



a. Right-click "COM". In the dialog box displayed, select "Add MODBUS Config".

b. In the dialog box displayed, set the timeout time and enabling control element.

Parameter Name	Description
Timeout	The timeout time for the Modbus-RTU or Modbus-ASCII master to wait for response from a Modbus-RTU or Modbus-ASCII slave, exceeding which an

Parameter Name	Description
	error is reported. It ranges from 50 ms to 10000 ms and can only be a multiple of 50. Default: 500 ms
Enabling control element	Enables or disables the connection. Customized variables are supported. Set this function as needed. Note: If this function is not enabled, the Modbus-RTU or Modbus-ASCII master is enabled by default.

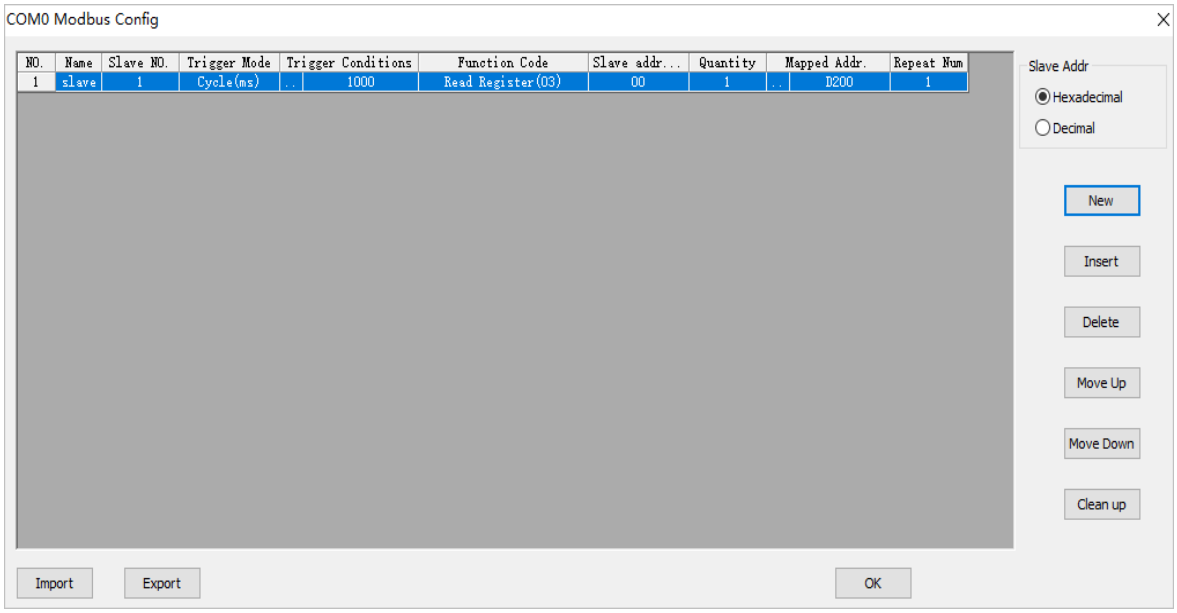
c. Click "OK".

NOTE

Set the CompatMode parameter of the system variable corresponding to the serial port number of the system to 1, so that PLC can analyze the nonstandard Modbus frame of H0U. (For example, set _MbMst.CompatMode of COM0 to 1.)

2. Add and configure the Modbus-RTU or Modbus-ASCII slave.

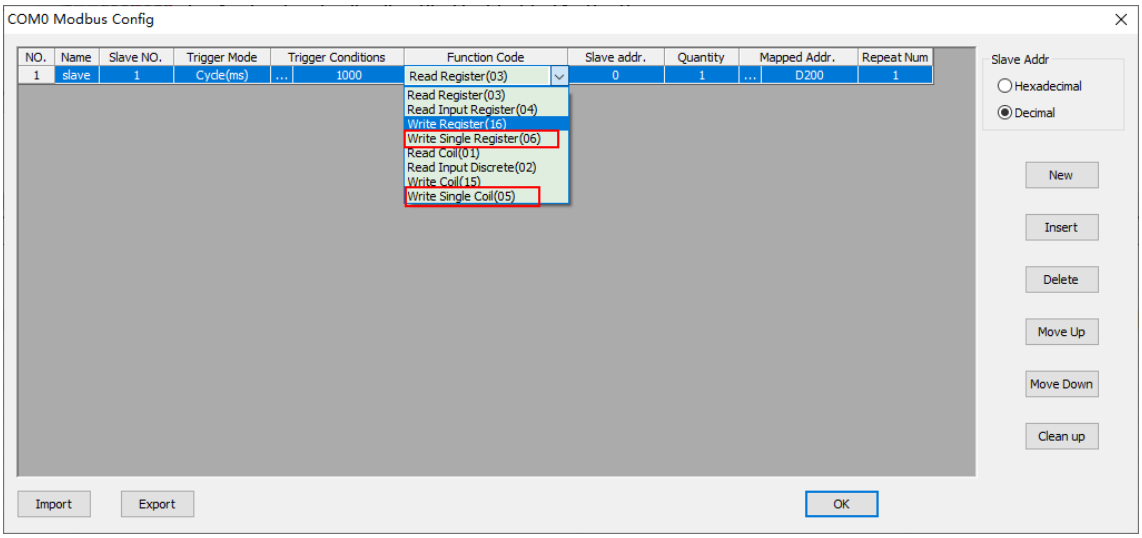
a. In the "Project Manager" tree, double-click "COM0 Modbus Config". The "COM0 Modbus Config" window is displayed, as shown in the following figure.



b. In the dialog box displayed, configure the address mapping relationship between the Modbus slave and Modbus master.

- Name
  The name that labels this condition configuration.
- Slave No.
  The number of the slave you want to access, ranging from 0 to 255. When the slave number of Modbus is configured to 0, the function code supports reading and writing function codes in 05 and

06, and the retransmission times cannot be edited.



> **NOTE**
>
> H5U and Easy models, firmware of V6.5.0.0 and later, and software of AutoShop V4.11.0.0 and later support the functions 05 and 06.

- Trigger Mode and Trigger Conditions
  The communication modes are "Cycle" and "Trigger".

  When "Cycle" is selected, the trigger condition is used to set the cycle time in ms. Then the configurations are executed according to the specified cycle.

  > **NOTE**
  >
  > When the set cycle is smaller than the time required for communication, the configuration is executed according to the time required for communication. For example, if the set cycle is 10 ms and the actual slave response requires 20 ms, the actual execution cycle is 20 ms.

  When "Trigger" is selected, the trigger condition is used to set the trigger condition variable/element. In this mode you can set the trigger condition to trigger a communication. If the slave station responds to the request, the trigger condition is automatically reset; otherwise, the trigger condition remains unchanged. If one trigger variable/element is used to trigger multiple configurations, the trigger condition will be automatically reset after all the triggered configurations are executed and the triggered configurations are not executed again.

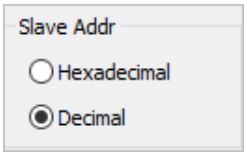
- Function code

Function Code	Definition
0x01 (01)	Read coils
0x02 (02)	Read discrete inputs
0x03 (03)	Read registers
0x04 (04)	Read input registers
0x05 (05)	Single coil

Function Code	Definition
0x06 (06)	Single register
0x0f (15)	Write multiple coils
0x10 (16)	Write multiple registers

- Slave register address
  The slave register address to be accessed.

  You can set the slave register address format to hexadecimal or decimal.



- Quantity
  The number of coils, discrete quantities, or registers to be accessed.

Function Code	Name	Max.
0x01 (01)	Read coils	2000
0x02 (02)	Read discrete inputs	2000
0x03 (03)	Read registers	125
0x04 (04)	Read input registers	125
0x05 (05)	Single coil	1
0x06 (06)	Single register	1
0x0f (15)	Write multiple coils	1968
0x10 (16)	Write multiple registers	123

- Mapped Addr.
  The mapped address of the slave coil, discrete quantity, or register in the master station.
  Customized variables are supported.

- Repeat Num
  The number of retries after the slave station response times out. The value range is 0 to 15 and the default value is 1.

c. Click "OK".

NOTE

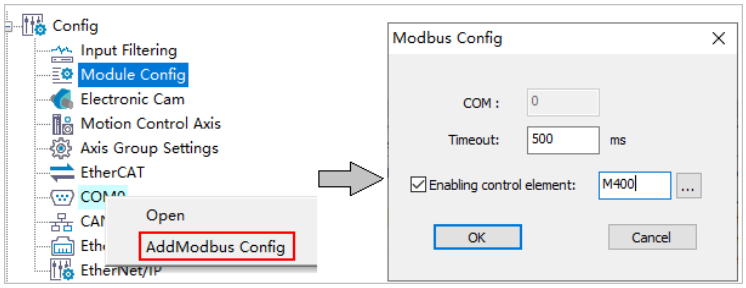
To configure multiple Modbus slaves, repeat steps a to c. When the system load is great, such as CPU utilization greater than 80%, the communication stability may be affected. In this case, increase the number of retries.

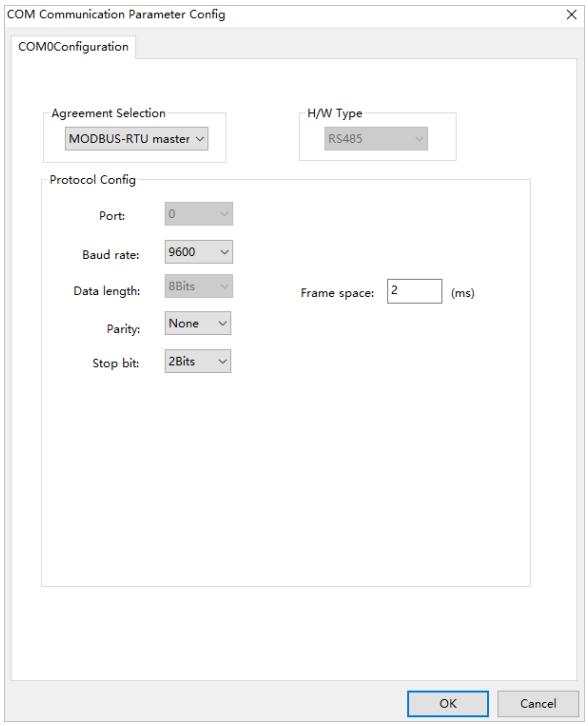
8.4.2 (Optional) Disabling a Modbus-RTU Slave

When the PLC serves as the Modbus-RTU master station, you can use system variables to disable a slave station.

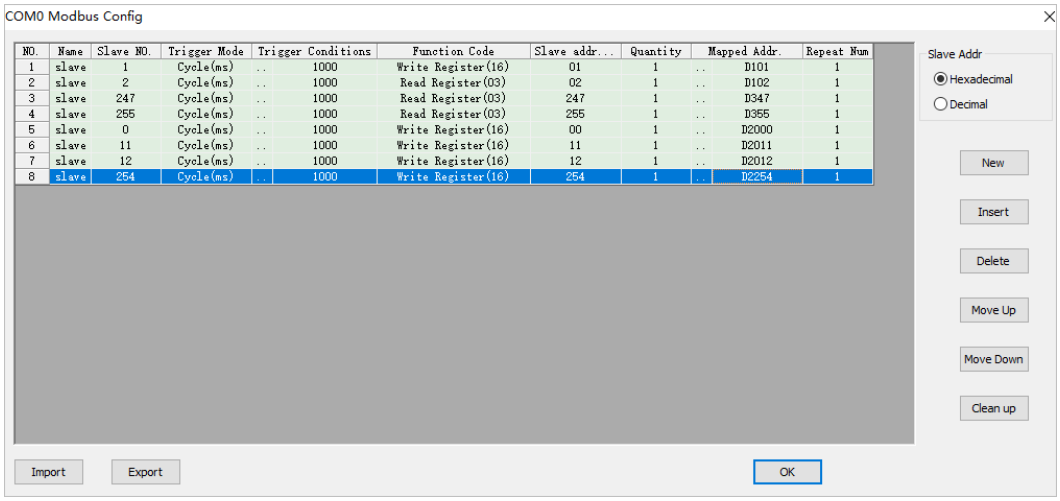
Procedure

1. Set the PLC as the Modbus-RTU master.

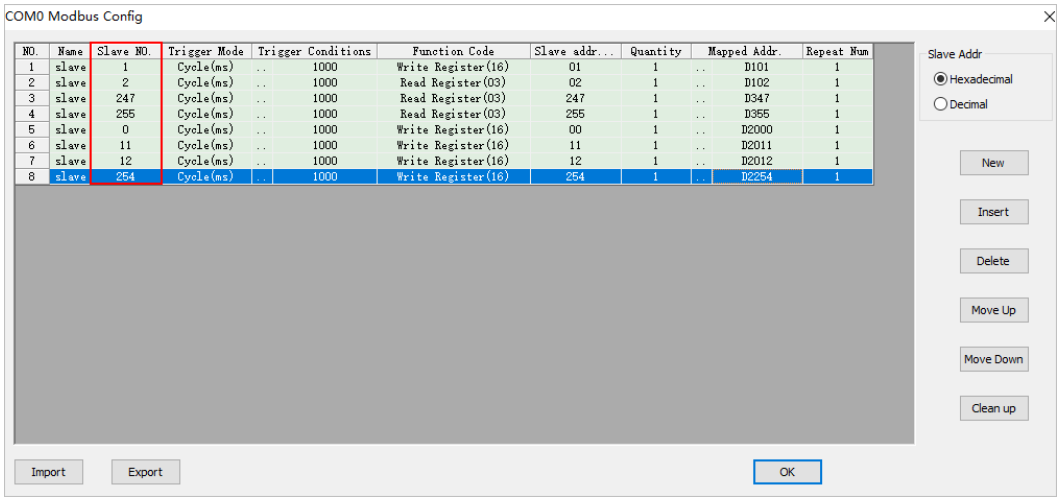




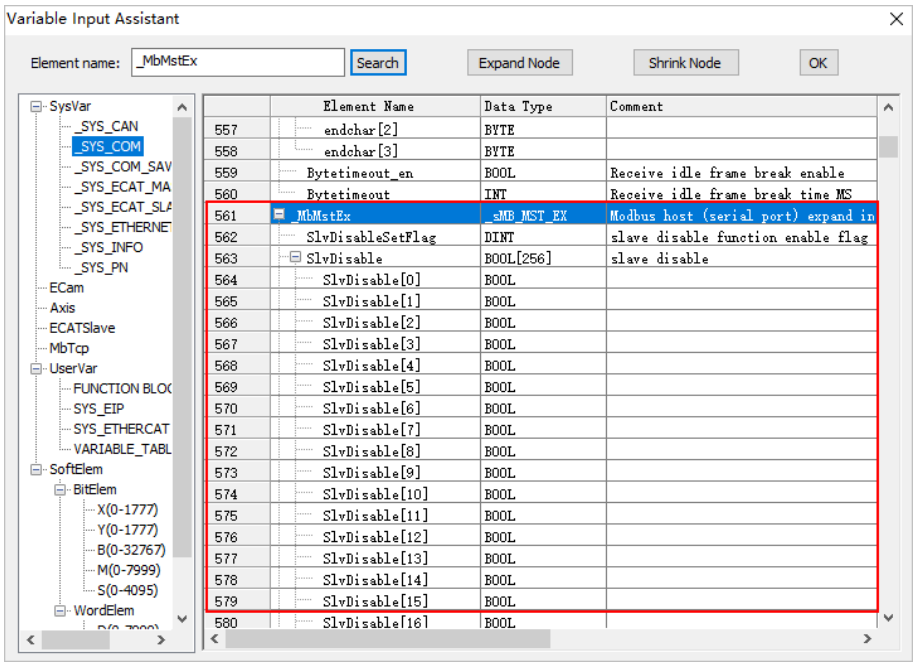
2. Add the configuration table. The following takes COM0 as an example.



3. Locate the slave you want to disable based on the slave No. or slot No..



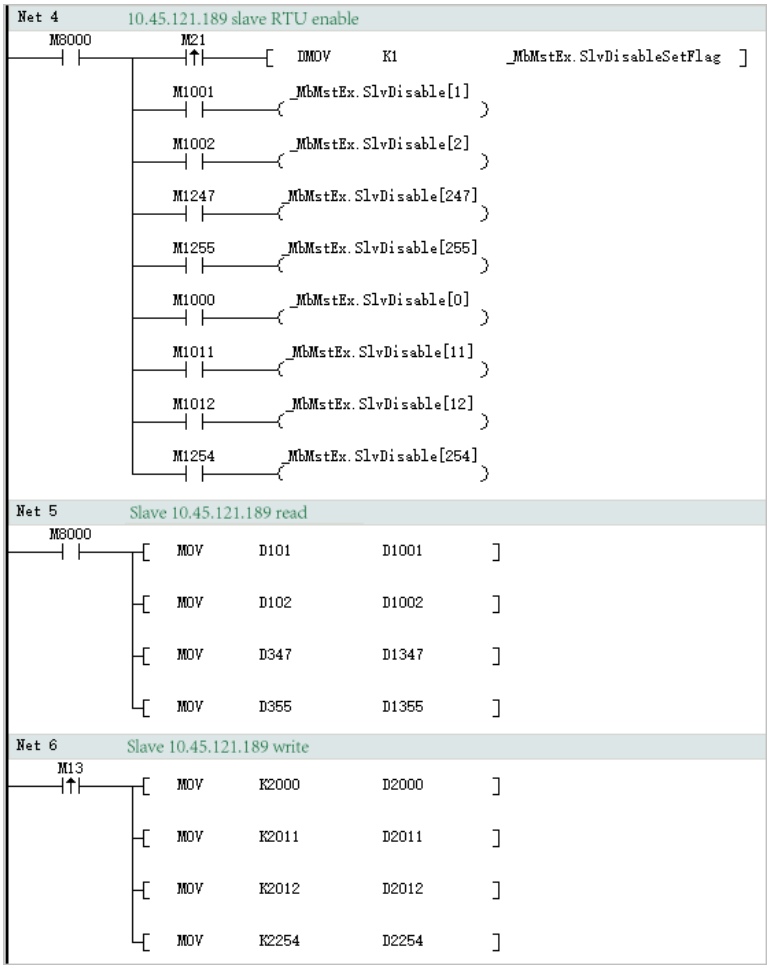
4. Configure the _MbMstEx structure array to disable slaves under the corresponding host.



NOTE

- The configuration corresponding to COM[N] in the _MbMstEx [N] structure array is not retentive at power failure.
- _MbMstEx.SlvDisableSetFlag is used to enable or disable the Slave Disable function. When it is non-zero, the Slave Disable function is enabled.
- In _MbMstEx.SlvDisable[M], M is the Slave Disable flag corresponding to the slave station number mentioned in step 3. It is only valid when _MbMstEx.SlvDisableSetFlag is enabled.

Program example



CAUTION

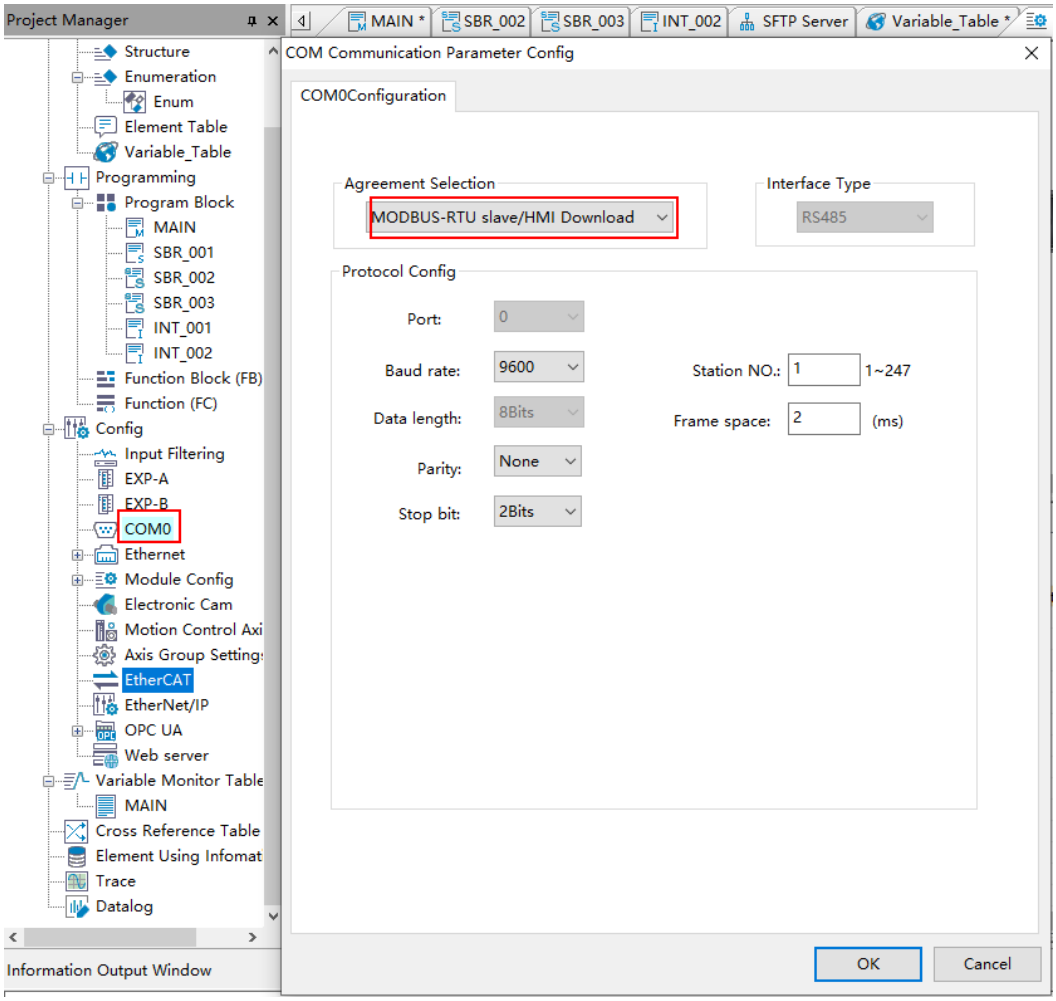
Note: When M1001, M1002 …, and M1254 are used to disable a single slave, the read or write operation on this slave is invalid.

8.5 Slave Configuration

8.5.1 Configuring Modbus-RTU and Modbus-ASCII Slaves

Procedure

1. In the "Project Manager" tree on the left, double-click "COM". The "COM Communication Parameter Config" dialog box is displayed.



NOTE

The following devices support low baud rates (600 bps, 1200 bps, 2400 bps, and 4800 bps):

- Easy500 series and Easy300 series models and GE20-232/485 extension cards.
- Software version: AutoShop V4.11.0.0 and later
- Firmware version: V6.5.0.0 and later

The following table list communication protocol and serial port basic parameters.

Parameter Name	Description
Protocol selection	Agreement Selection Options are None, Free Protocol, Modbus-RTU Master, MODBUS-RTU Slave//HMI Download Protocol, MODBUS-ASC Master, and MODBUS-ASC Slave.
H/W Type	The type of hardware of other devices communicating with the module. Options are RS232, RS485, and RS422.
Port	The number of the serial port for communication connection.
Baudrate	The communication baud rate. Options are 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200 Default: 9600
Data length	The actual data bits included in communication frames.
Parity	

Parameter Name	Description
	The method of verifying communication frames. Options are None, Odd, and Even. Default: Even
Frame interval	The interval between receiving a data frame and sending the next data frame.
Stop bit	The last bit in a single message during communication. Options are 1 and 2. Default: 1

2. Select "MODBUS-RTU slave" or "MODBUS-ASC slave", set serial port parameters and the slave No., and then click "OK".
3. Download the project to PLC.

Function codes and addresses

- When H5U is used as the slave station, the following function codes are supported:

Function Code	Definition
0x01	Read coils
0x02	Read discrete quantities (same as 0x01)
0x03	Read registers
0x04	Read input registers (same as 0x03)
0x05	Write single coil
0x06	Write single register
0x0f	Write multiple coils
0x10	Write multiple registers
0x80 to 0xFF	Standard Modbus fault code

- When H5U is used as the slave station, addresses of coils that can be accessed by Modbus are listed in the following table:

Variable Name	Quantity	Address Range
M0-M7999	8000	0x0000 to 0x1F3F (0 to 7999)
B0-B32767	32768	0x3000 to 0xAFFF (12288 to 45055)
S0-S4095	4096	0xE000 to 0xEFFF (57344 to 61439)
X0-X1777 (octal)	1024	0xF800 to 0xFBFF (63488 to 64511)

Variable Name	Quantity	Address Range
Y0-Y1777 (octal)	1024	0xFC00-0xFFFF (64512 to 65535)

- When H5U is used as the slave station, addresses of registers that can be accessed by Modbus are listed in the following table:

Variable Name	Quantity	Start Address
D0-D7999	8000	0x0000 to 0x1F3F (0 to 7999)
R0-R32767	32768	0x3000 to 0xAFFF (12288 to 45055)

NOTE

W elements and Pointer variables are not supported.

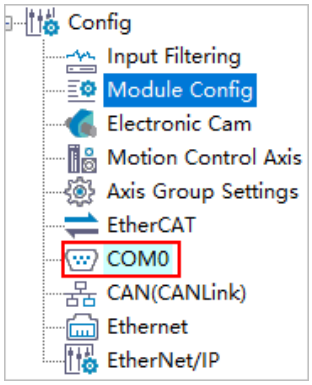
8.6 Application Example

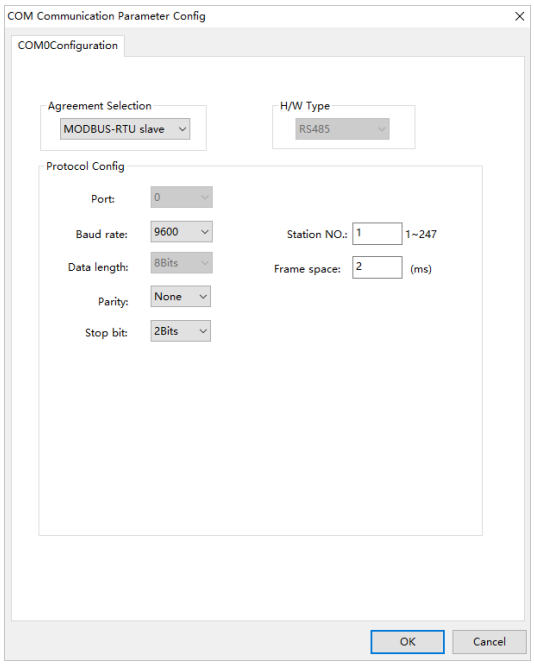
Example requirements

In this example, two H5Us are connected through a serial port and communicate with each other through the Modbus-RTU protocol. The master PLC reads the value in the D100 register of the slave PLC every 10 ms, and the value in D100 of the slave station is added by one every second.

Slave configuration

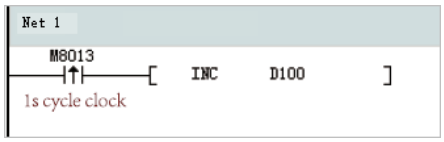
1. In the "Project Manager" tree on the left, double-click "COM". The "COM Communication Parameter Config" page is displayed.





2. Set the serial communication protocol and communication parameters. In this example, the protocol is Modbus-RTU and the communication parameter is 9600-8N2. Then, click "OK" to save the settings.

3. Edit the program so that the value of D100 of the slave station is added by one every second.

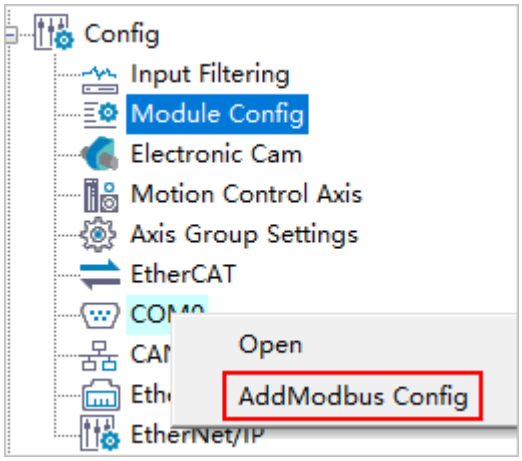


4. Then, click to download the program to the PLC.

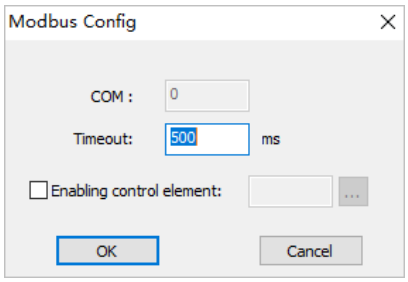
Master configuration

1. In the "Project Manager" tree on the left, double-click "COM". In the window displayed, set the communication protocol to Modbus-RTU master and set the communication parameter to that of the slave.

2. Right-click "COM". In the dialog box displayed, select "Add MODBUS Config".

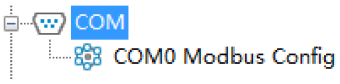


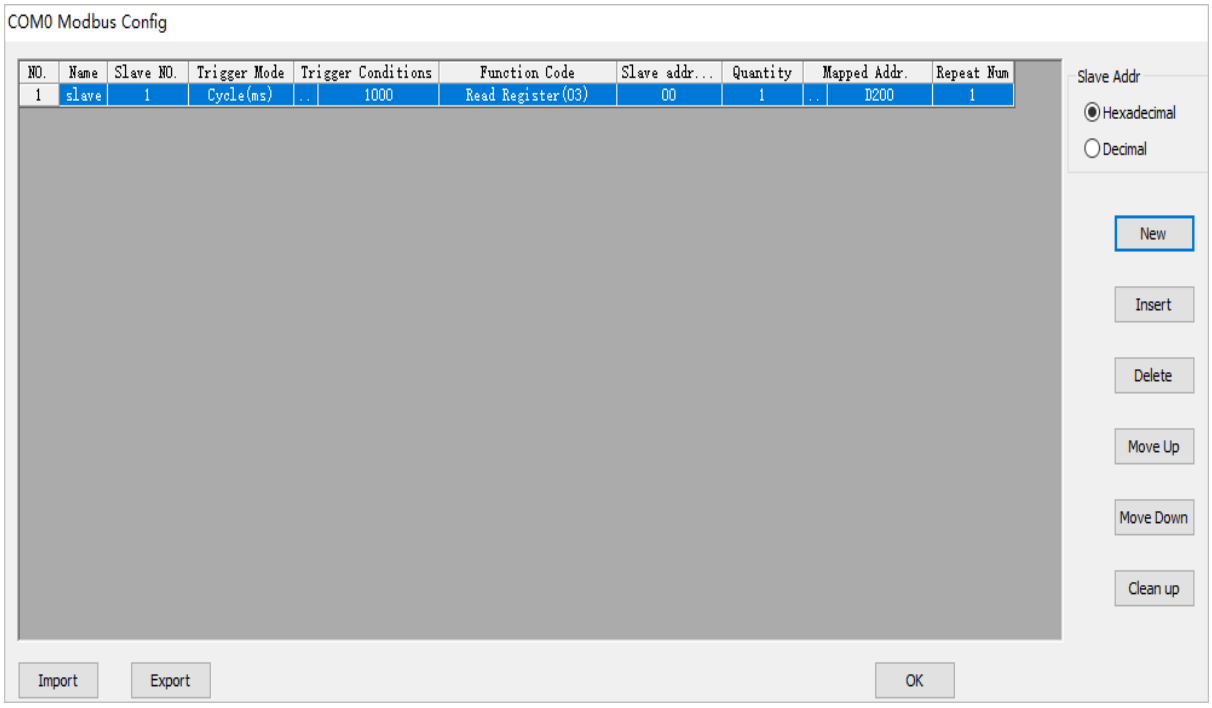
3. In the dialog box displayed, set "Timeout" and "Enabling control element". In this example, the default settings are used: "Timeout" is set to 500 ms, and "Enabling control element" is deselected.



4. Click "OK". The master configuration is generated.

5. Double-click "COM0 Modbus Config". On the "COM0 Modbus Config" page displayed, click "Add" to add the configuration. In this example, the value of D100 of the slave is stored in D200 of the master.





6. Click to download the program to the PLC.

Monitoring the D200 soft element of the Modbus-RTU master

1. In the toolbar, click to make the Modbus-RTU master enter the monitoring state.

2. On the "Information Output Window" page, right-click a blank row in the "Element Name" column, select "Add row", and add the D200 soft element to the monitoring list.
   Then, you can see that the D200 soft element reads the value of the D100 soft element of the Modbus-TCP slave.

	Element Name	Data Type	Display Format	Current Value
1	D200	INT	Dec	1853
2				
3				
4				
5				

8.7 Serial Parameter Modification

8.7.1 Modifying COM Port Parameters

When free protocol and Modbus-RTU/Modbus ASC master-slave are configured for the COM port, you can use the system variable _COMSet to modify the COM port parameters. The effective _COMSet parameter is synchronized to the _COM parameter, as shown in the following figure.

Variable Name	Initial Value	Type	Comment
_COMSet		_sCOMSet	COM0 Parameter setting
SetFlag	0	DINT	Parameter setting enable
BaudRate	0	DINT	BaudRate(bps)
DataBits	0	INT	DataBits
Parity	0	INT	Parity
StopBits	0	INT	StopBits
Interface	0	INT	Interface
Protocol	0	INT	Protocol

Toolbox

Variable N	Initial Value	Type	Comment
_COM		_sCOM	Serial information
BaudRate	0	DINT	Baud Rate (bps)
DataBits	0	INT	Data Bits
Parity	0	INT	Check Bit
StopBits	0	INT	Stop Bit
Interface	0	INT	Physical Interface
Protocol	0	INT	Communication Protocol

Procedure

1. When _COMSet.SetFlag is set to a non-zero value, parameters of the COM port can be modified online.
2. Only the baud rate, data bit, parity bit, and stop bit can be modified. The physical port and communication protocol parameters are read-only and cannot be modified.
3. Modification to these parameters takes effect only after STOP-RUN is executed on the PLC. The effective parameters can be viewed through the system variable _COM.

> **NOTE**
>
> When the baud rate, data bit, parity bit, or stop bit is set to an invalid value, the parameter will be reset by the system to the system background configuration parameter.

4. When the PLC is used as the Modbus-RTU or Modbus-ASC master station, the Modbus configuration must be added; otherwise, modification to the parameter _COMSet is not synchronized to _COM.

5. After the COM parameters of the PLC are modified, parameters of the communication device connected to the port will be automatically synchronized.

8.7.2 Modifying Slave Address Parameters

When a Modbus-RTU or Modbus-ASC slave is set for the COM port, you can use the system variable _COMProtocolSet to modify the slave address parameter SlaveAddress of the COM port. After the modification takes effect, the parameter Address is synchronized to the SlaveAddress in the parameter _MbSlv, as shown in the following figure.

Toolbox

Variable Name	Initial Value	Type	Comment
_COMProtocolSet		_sCOMProto	COM0 Proto
port	0	INT	port number
Reserved0	0	INT	reserved0
AddressSetFlag	0	INT	slave addres
Address	0	INT	slave addres
Reserved1	0	DINT	reserved1
Reserved2		DINT[8]	Reserved2

Toolbox

Variable Name	Initial Value	Type	Comment
_MbSlv		_sMB_SLV	Modbus slave (serial port) information
Port	0	INT	Serial slogan
SlaveAddress	0	INT	Slave Address
Connected	OFF	BOOL	Connection Status

Procedure

1. When _COMProtocolSet.AddressSetFlag is set to a non-zero value, the parameter Address can be modified online.

> **NOTE**
>
> _COMProtocolSet can be used to modify the parameter Address only.

2. Modification to these parameters takes effect only after STOP-RUN is executed on the PLC. The effective parameters can be viewed through the system variable _COM.

9 Ethernet Communication

9.1 Overview

H5U, Easy320, and Easy52X series come with the Modbus-TCP and FINS TCP/UDP protocols and a server and a client. They can easily implement communication and data exchange with devices supporting Modbus-TCP and FINS TCP/UDP. Socket instructions are provided for devices not supporting Modbus-TCP and FINS TCP/UDP to implement application protocols based on TCP/UDP. For details, see the section "Socket Instructions" in the H5U and Easy Series Programmable Logic Controllers Instructions Guide.

H5U provides an Ethernet port. The Easy320 and Easy52X series provide two Ethernet ports. You can use AutoShop to monitor and commission the PLC and download and upload parameters from and to the PLC through Ethernet quickly and conveniently. You can also exchange data with other devices in the network through Ethernet.

H5U, Easy320, and Easy52X series use the RJ45 port, whose specifications are listed in the following table.

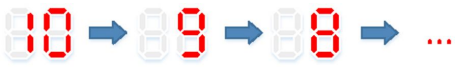
Item	Ethernet Port
Transmission rate	10 Mbps: 10BASE-T 100 Mbps: 100BASE-TX 10 Mbps/100 Mbps self-adaptive
Modulation	Baseband
Topology	Star
Medium	Cat5 twisted pairs or shielded twisted pairs with aluminum foil and braided mesh
Transmission distance	Distance between nodes: 100 m or less
Number of connections	31

9.2 Setting the IP Address

Restoring the factory default IP address (H5U)

The default IP address of H5U is 192.168.1.88. You can press the MFK key on the operating panel to restore the IP address to the default value as follows.

1. Switch the status of H5U to Stop.
2. Press and hold the MFK key until "IP" is displayed on the LED, and then press the MFK key for no more than 2s.
   Then, a countdown is displayed on the LED. When the countdown reaches 0, the IP address is restored to the default value.



> **NOTE**
>
> You can press the MFK key during countdown to cancel the restoration operation.

Setting the IP address through a USB flash drive (H5U and Easy)

For how to set the IP address through a USB flash drive, see the description in the section Connection through USB.

Setting the IP address through Ethernet (H5U and Easy)

For how to set the IP address through Ethernet, see the description in the section Connection through Ethernet.

Setting the IP address through system variables (H5U and Easy)

You can use the system variable to modify the IP address in the running state. For example, to set the variable Ethernet, perform the following operations.

1. Modify the value of the variable _Ethernet.IPCommand to 1.
2. Modify the variable _Ethernet.IPAddress to the target IP address. If the value of this variable is hexadecimal, such as 192.168.1.88, input C0A80158.
3. After inputting the IP address, modify the value of the variable _Ethernet.IPCommand to 2. Then, the value of this variable is automatically changed to 0 (display mode).

> **NOTE**
>
> This variable can only modify the IP address and cannot modify other variables.

Variable Name	Initial Value	Type	Comment
_Ethernet		_sETHERNET	Port Information
MACAddress		INT[3]	Physical Address
MACAddress[0]	0	INT	
MACAddress[1]	0	INT	
MACAddress[2]	0	INT	
IPAddress	0	DINT	IP Address
Mask	0	DINT	Subnet Mask
Gateway	0	DINT	Gateway
IPCommand	0	INT	IP Command, 0: Display, 1: Edit\/Modify, 2: Write\/Effective

9.3 Modbus TCP Communication Configuration

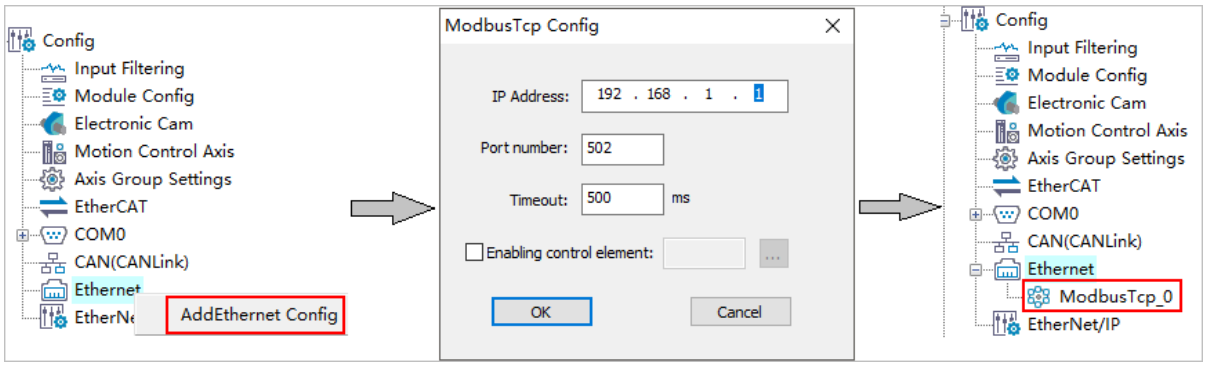
9.3.1 Configuring a Modbus-TCP Master

The Modbus-TCP master is the Modbus-TCP client. One Modbus-TCP master can communicate with a maximum of 31 Modbus-TCP slaves (Modbus-TCP servers) at a time.

Procedure

1. Set the IP address of the PLC. For details, see Setting IP Address.

2. Add a Modbus-TCP slave.



a. In the "Project Manager" tree on the left, right-click "Ethernet" and select "Add Ethernet Config".

b. In the dialog box displayed, set the IP address, port number, timeout time, and enabling control element.

Parameter Name	Description
IP Address	The IP address of the Modbus-TCP slave. Input the actual IP address of the Modbus-TCP slave.
Port number	The port number of the Modbus-TCP slave. The default value is recommended. Default: 502
Timeout	The timeout time for the Modbus-TCP master to wait for response from a Modbus-TCP slave, exceeding which an error is reported. It ranges from 100 ms to 10000 ms. Default: 500 ms
Enabling control element	Enables or disables the connection. Customized variables are supported. Set this function as needed. Note: If this function is not enabled, the Modbus-TCP master is enabled by default.

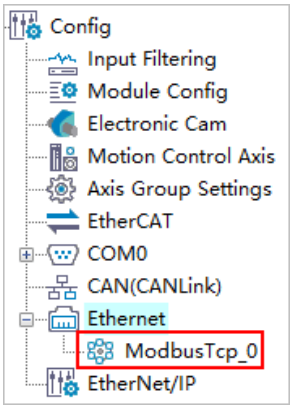
c. Click "OK".

NOTE

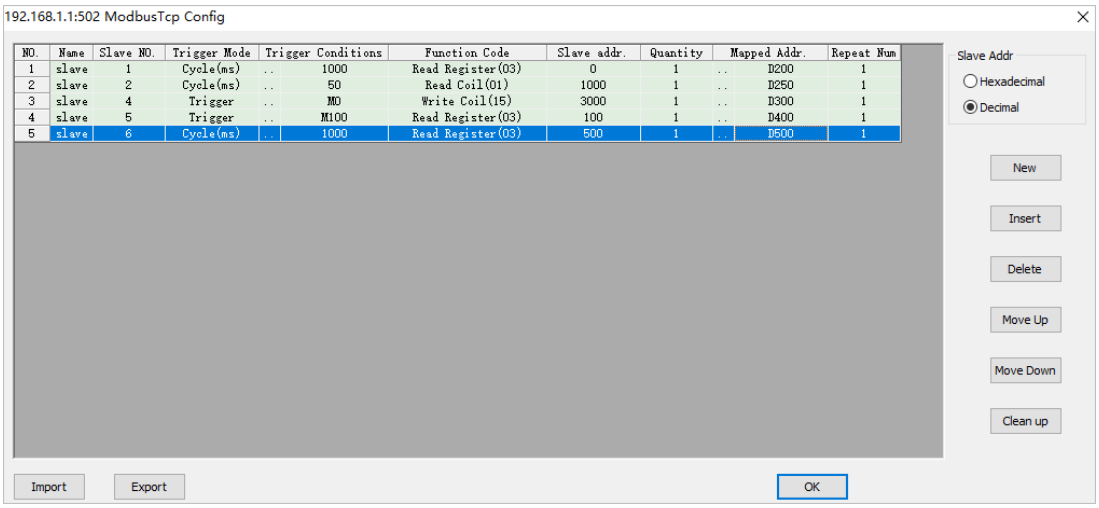
To add multiple Modbus-TCP slaves, repeat steps a to c. When the system load is great, such as CPU utilization greater than 80%, the communication stability may be affected. In this case, increase the number of retries.

3. Configure the added Modbus-TCP slave.

a. In the "Project Manager" tree, double-click the added Modbus-TCP slave.



b. In the dialog box displayed, configure the address mapping relationship between the Modbus slave and Modbus master.



- Name
  The name that labels this condition configuration.
- Slave No.: Identifies the slave.
  The number of the slave you want to access, ranging from 0 to 255.

  Modbus-TCP communication identifies slaves by IP address. The slave No. is not checked. You can use the default slave No..
- Trigger Mode and Trigger Conditions
  The communication modes are "Cycle" and "Trigger".

  When "Cycle" is selected, the trigger condition is used to set the cycle time in ms. Then the configurations are executed according to the specified cycle.

  NOTE

  When the set cycle is smaller than the time required for communication, the configuration is executed according to the time required for communication. For example, if the set cycle is 10 ms and the actual slave response requires 20 ms, the actual execution cycle is 20 ms.

  When "Trigger" is selected, the trigger condition is used to set the trigger condition variable/element. In this mode you can set the trigger condition to trigger a communication. If the slave station responds to the request, the trigger condition is automatically reset; otherwise, the trigger condition remains unchanged. If one trigger variable/element is used to trigger multiple configurations, the trigger condition will be automatically reset after all the triggered configurations are executed and the triggered configurations are not executed again

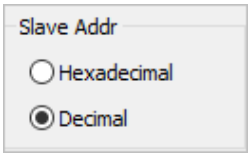
- Function code

Function Code	Definition
0x01 (01)	Read coils
0x02 (02)	Read discrete inputs
0x03 (03)	Read registers
0x04 (04)	Read input registers
0x05 (05)	Single coil
0x06 (06)	Single register
0x0f (15)	Write multiple coils
0x10 (16)	Write multiple registers

- Slave register address

  The slave register address to be accessed.

  You can set the slave register address format to hexadecimal or decimal.



- Quantity

  The number of coils, discrete quantities, or registers to be accessed.

Function Code	Name	Max.
0x01 (01)	Read coils	2000
0x02 (02)	Read discrete inputs	2000
0x03 (03)	Read registers	125
0x04 (04)	Read input registers	125
0x05 (05)	Single coil	1
0x06 (06)	Single register	1
0x0f (15)	Write multiple coils	1968
0x10 (16)	Write multiple registers	123

- Mapped address

  The mapped address of the slave coil, discrete quantity, or register in the master station.

  Customized variables are supported.

- Repeat Num

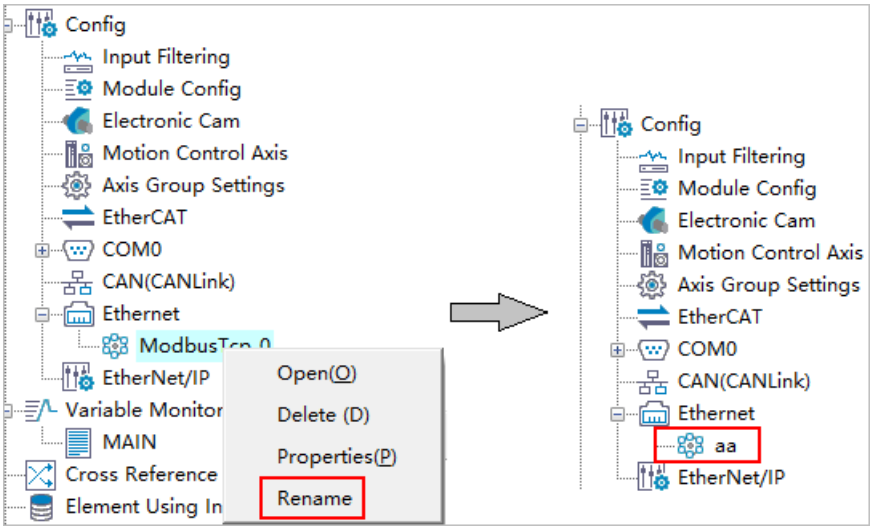
The number of retries after the slave station response times out. The value range is 0 to 15 and the default value is 1.

c. Click "OK".

> **NOTE**
>
> To configure multiple Modbus-TCP slaves, repeat steps a to c.

4. (Optional) Rename the Modbus-TCP slave.

a. In the "Project Manager" tree on the left, right-click the added Modbus-TCP slave and select "Rename".



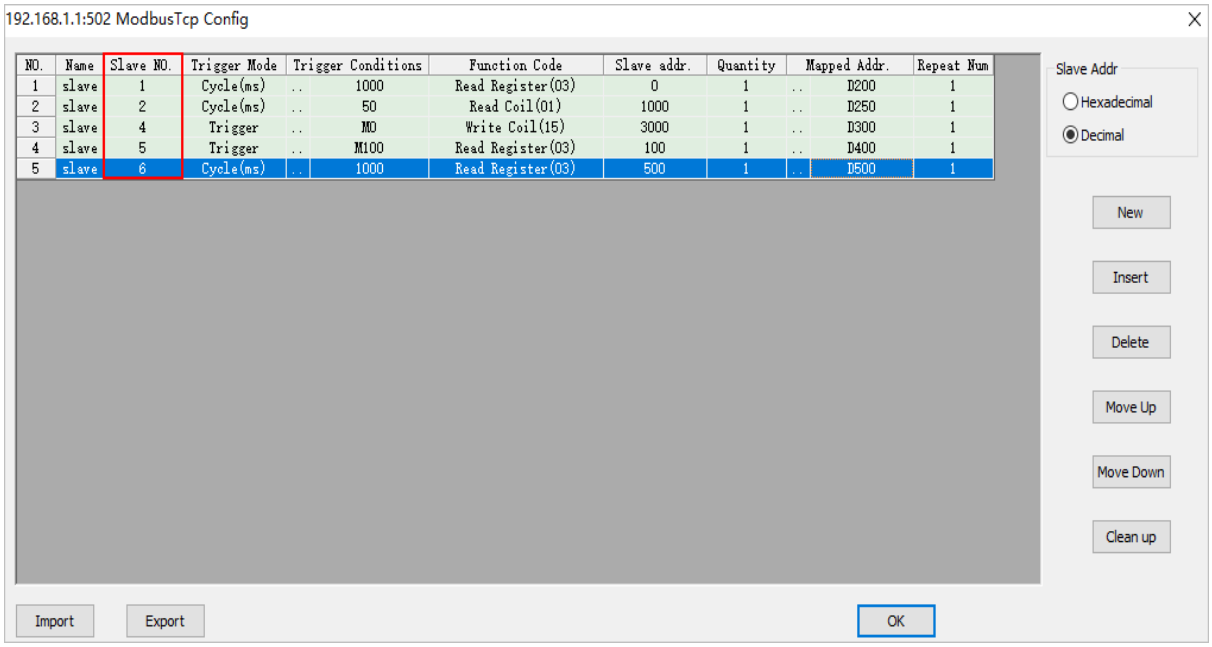
b. In the dialog box displayed, input a name for the Modbus-TCP slave.(Optional) Rename the Modbus-TCP slave.

> **NOTE**
>
> The name of the Modbus-TCP slave can be used as a variable which is bound with the system variable _MbTcpMst[i] and records the slave connection status. In the program, you can use this variable to access the slave status, such as aa.Enable and ModbusTcp_1.Error. This variable is read-only. For information about the _MbTcpMst structure, see _SYS_Ethernet for Ethernet Information.

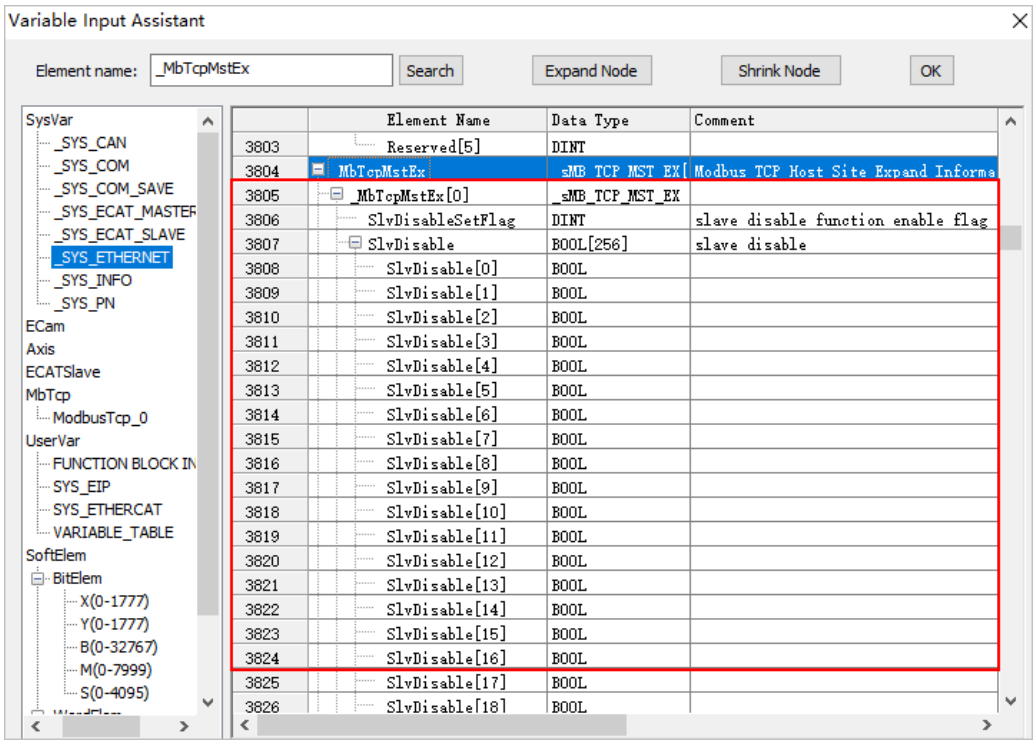
9.3.2 (Optional) Disabling a Modbus-TCP Slave

Procedure

1. In the left tree, double-click the Modbus TCP slave to be disabled. In the dialog box displayed, view the record the slave station number.

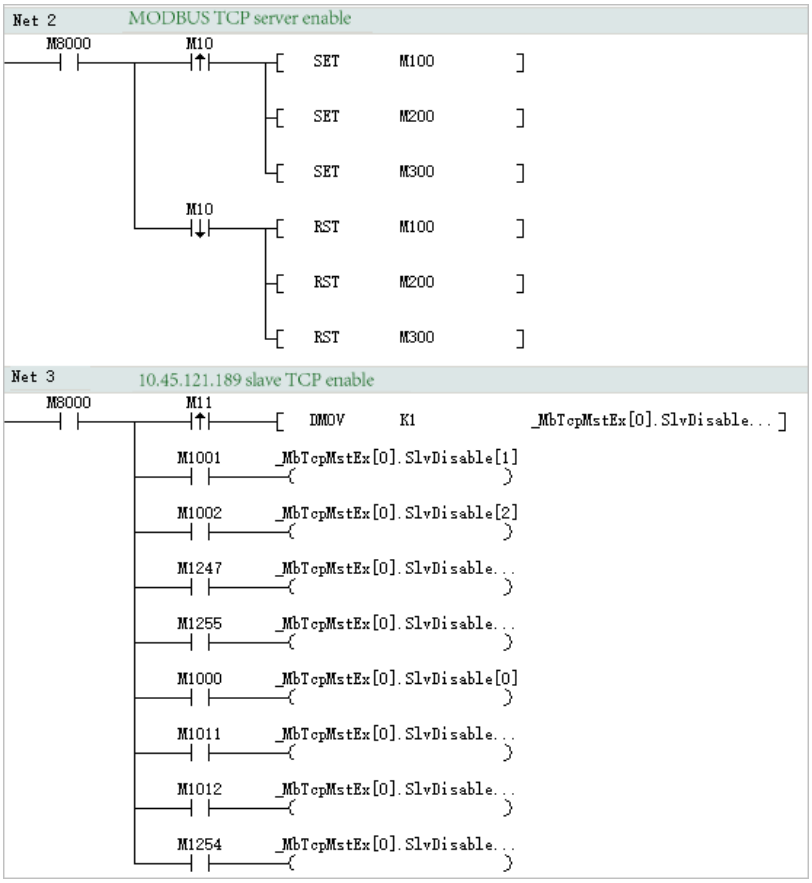


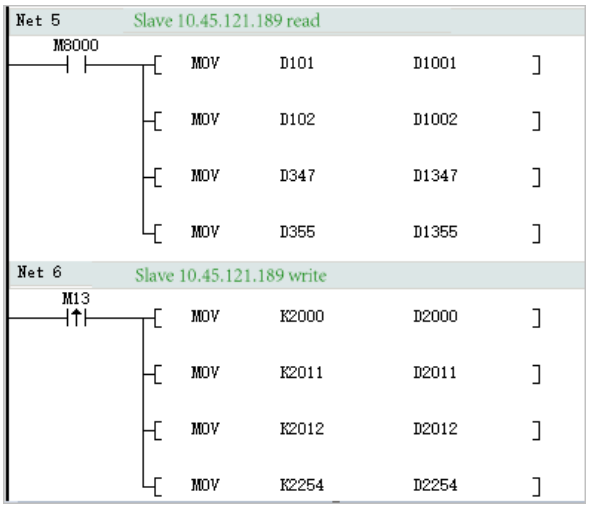
2. Program and configure the _MbTcpMstEx structure array to disable the slave under the corresponding host.



- In the _MbTcpMstEx[N], N is the server number. In this example, N is 0.
- _MbTcpMstEx[N].SlvDisableSetFlag is used to enable or disable the Slave Disable function. When it is non-zero, the Slave Disable function is enabled.
- In _MbTcpMstEx[N].SlvDisable[M], M is the Slave Disable flag corresponding to the slave station number mentioned in step 1. It is only valid when _MbTcpMstEx[N].SlvDisableSetFlag is enabled.

Program example





CAUTION

When M1001 M1002 …, and M1254 are used to disable a single slave, the read or write operation on this slave is invalid.

9.3.3 Configuring a Modbus-TCP Slave

A Modbus-TCP slave is a Modbus-TCP server enabled with Modbus-TCP and port 502 by default. The communication protocol is not mandatory. One Modbus-TCP slave can connect to a maximum of 31 Modbus-TCP clients (Modbus-TCP masters) at a time.

The procedure for setting the IP address and port number of the PLC is the same as that for setting the IP address and port number of a Modbus-TCP slave at the Modbus-TCP master. For details, see Setting IP Address.

Function codes and addresses

- The following table lists the function codes supported by the Modbus-TCP slave.

Function Code	Definition
0x01	Read coils
0x02	Read discrete quantities (same as 0x01)
0x03	Read registers
0x04	Read input registers (same as 0x03)
0x05	Write single coil
0x06	Write single register
0x0f	Write multiple coils
0x10	Write multiple registers
0x80 to 0xFF	Standard Modbus fault code

- When H5U is used as a Modbus-TCP slave, addresses of coils that can be accessed by Modbus-TCP are listed in the following table:

Variable Name	Quantity	Address Range
M0-M7999	8000	0x0000 to 0x1F3F (0 to 7999)
B0-B32767	32768	0x3000 to 0xAFFF (12288 to 45055)
S0-S4095	4096	0xE000 to 0xEFFF (57344 to 61439)
X0-X1777 (octal)	1024	0xF800 to 0xFBFF (63488 to 64511)
Y0-Y1777 (octal)	1024	0xFC00-0xFFFF (64512 to 65535)

- When H5U is used as a Modbus-TCP slave, addresses of registers that can be accessed by Modbus-TCP are listed in the following table:

Variable Name	Quantity	Start Address
D0-D7999	8000	0x0000 to 0x1F3F (0 to 7999)
R0-R32767	32768	0x3000 to 0xAFFF (12288 to 45055)

W elements and Pointer variables are not supported.

9.3.4 Application Example

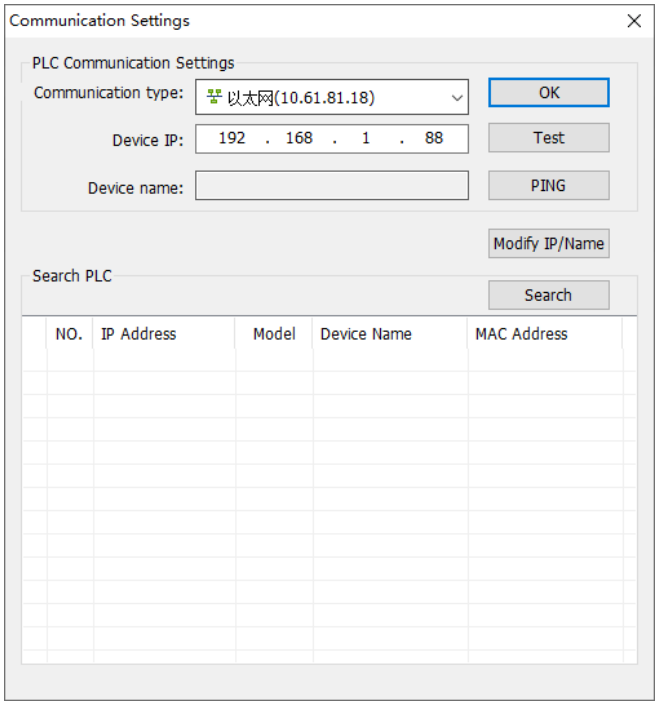
Example Requirements

In this example, two H5Us are connected through an Ethernet port and communicate with each other through the Modbus-TCP protocol. The master PLC reads the value in the D100 register of the slave PLC every 10 ms, and the value in D100 of the slave is added by one every second.

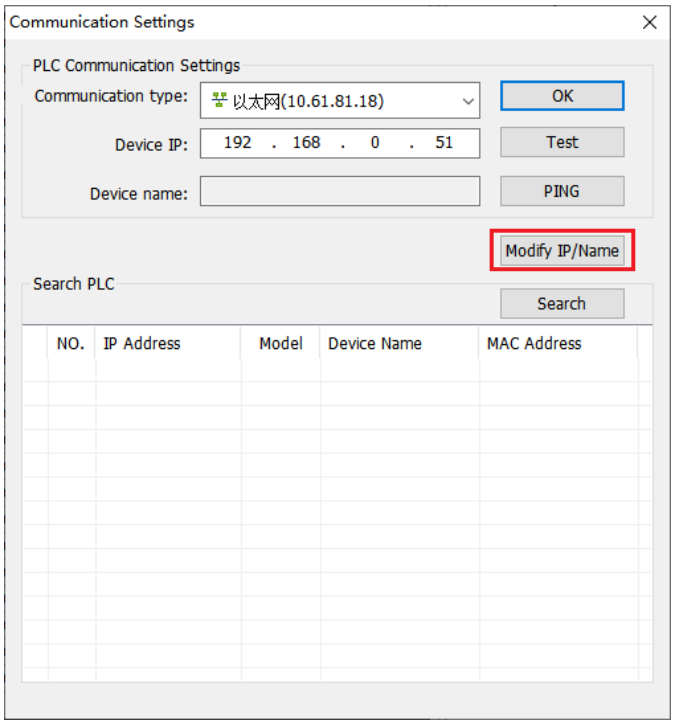
> NOTE
>
> This DHCP function requires firmware of V6.5.0.0 or later and software of AutoShop V4.11.0.0 or later.

Procedure

1. Configure the Modbus-TCP slave.
   a. Choose "Tools" > "Communication Settings" in the menu bar or click in the toolbar. The "Communication Settings" dialog box is displayed.

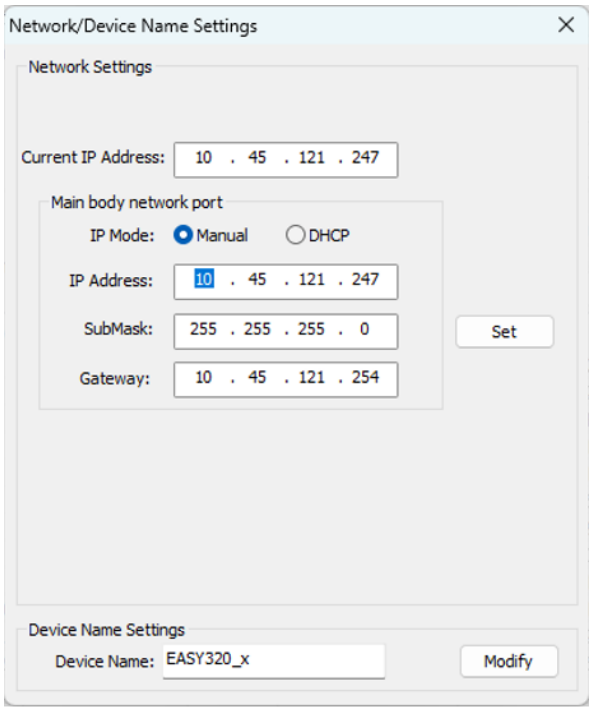


b. Click "Modify IP/Name".

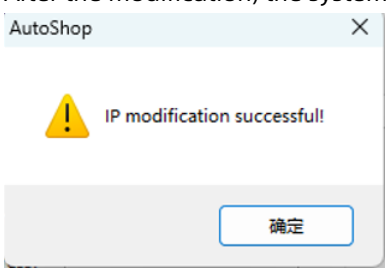


c. The "Network/Device Name Settings" dialog box is displayed. Set a new IP address, subnet mask, and default gateway for the Modbus-TCP slave. Then, click "Set".

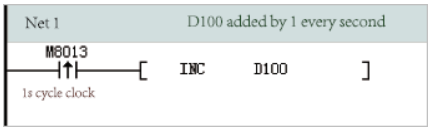
In this example, the IP address, subnet mask, and default gateway of the Modbus-TCP slave are 192.168.1.100, 255.255.255.0, and 192.168.1.1.



d. After the modification, the system prompts "IP modification successful!". Click "OK".



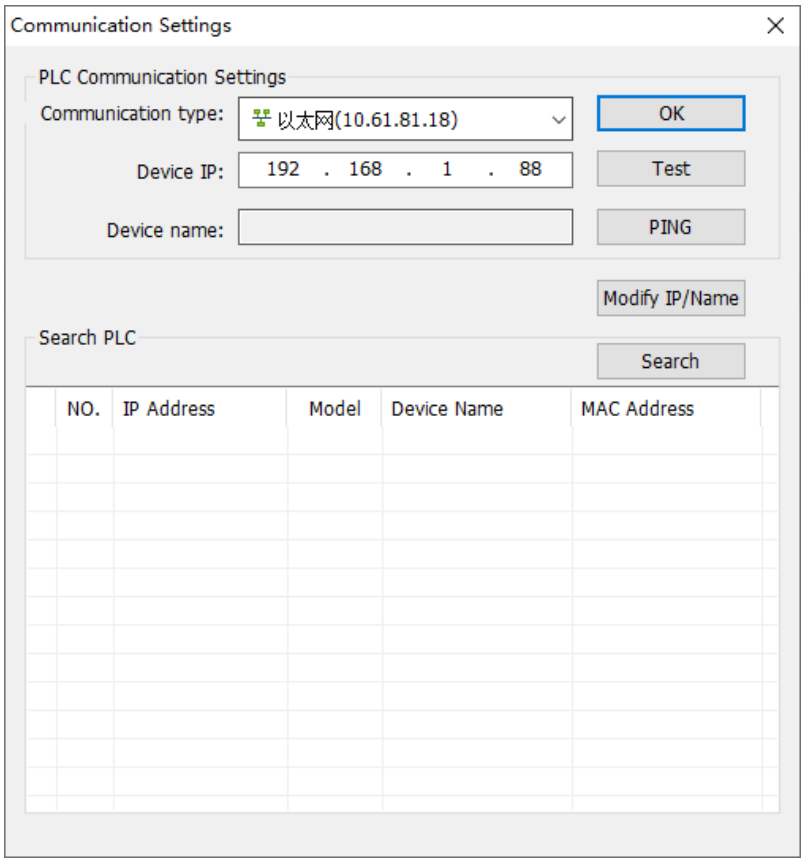
e. In the "Project Manager" tree on the left, double-click "MAIN". On the program editing page displayed, write a program to increase the value of D100 of the Modbus-TCP slave by 1 per second.



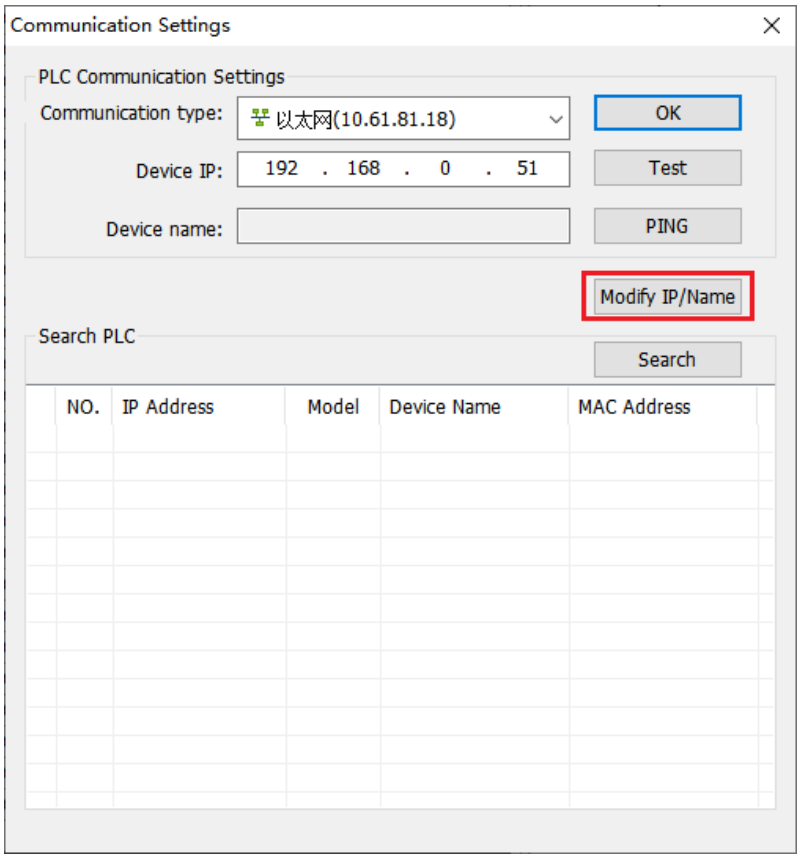
f. Then, click to download the program to the PLC.

2. Configure the Modbus-TCP master.

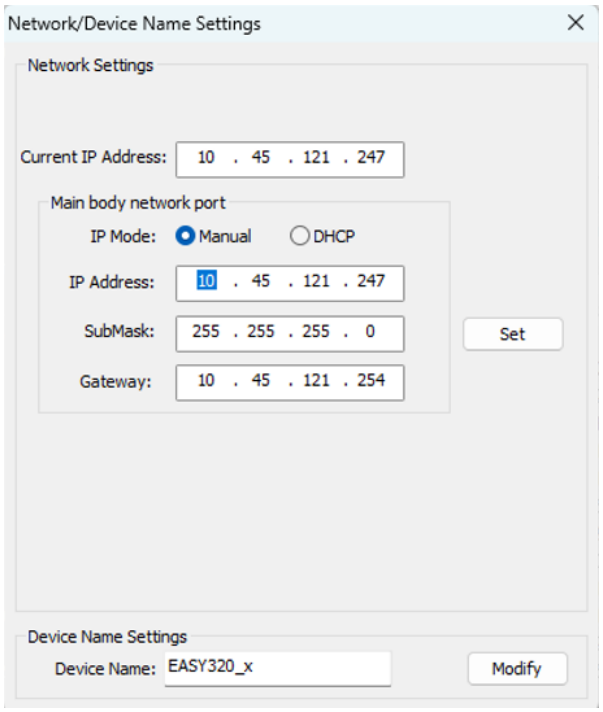
a. Choose "Tools" > "Communication Settings" in the menu bar or click in the toolbar. The "Communication Settings" dialog box is displayed.



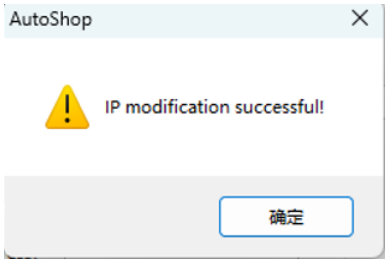
b. Click "Modify IP/Name".



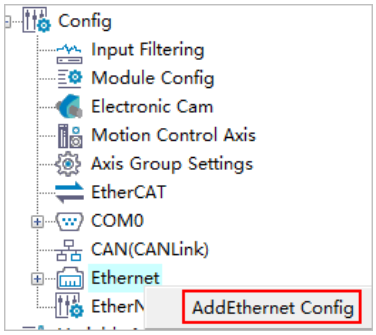
c. The "Network/Device Name Settings" dialog box is displayed. Set a new IP address, subnet mask, and default gateway for the Modbus-TCP master. Then, click "Modify IP".
In this example, the IP address, subnet mask, and default gateway of the Modbus-TCP master are 192.168.1.99, 255.255.255.0, and 192.168.1.1.



d. After the modification, the system prompts "IP modification successful!". Click "OK".

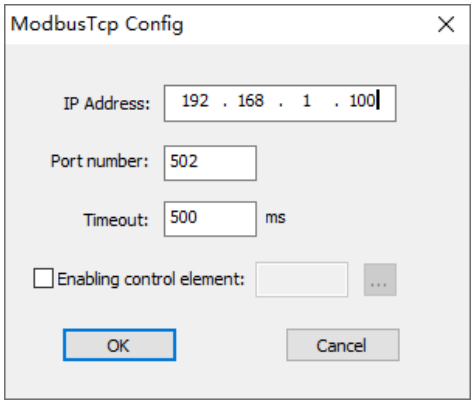


e. In the "Project Manager" tree on the left, right-click "Ethernet" and select "Add Ethernet Config".

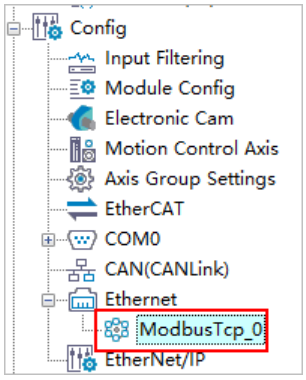


f. In the dialog box displayed, configure the IP address, port ID, timeout time, and enabling control element for the Modbus-TCP slave.

In this example, the IP address of the Modbus-TCP slave is 192.168.1.100 and other parameters are defaulted, that is, the port ID is 502, the timeout time is 500 ms, and the enabling control element is not used.

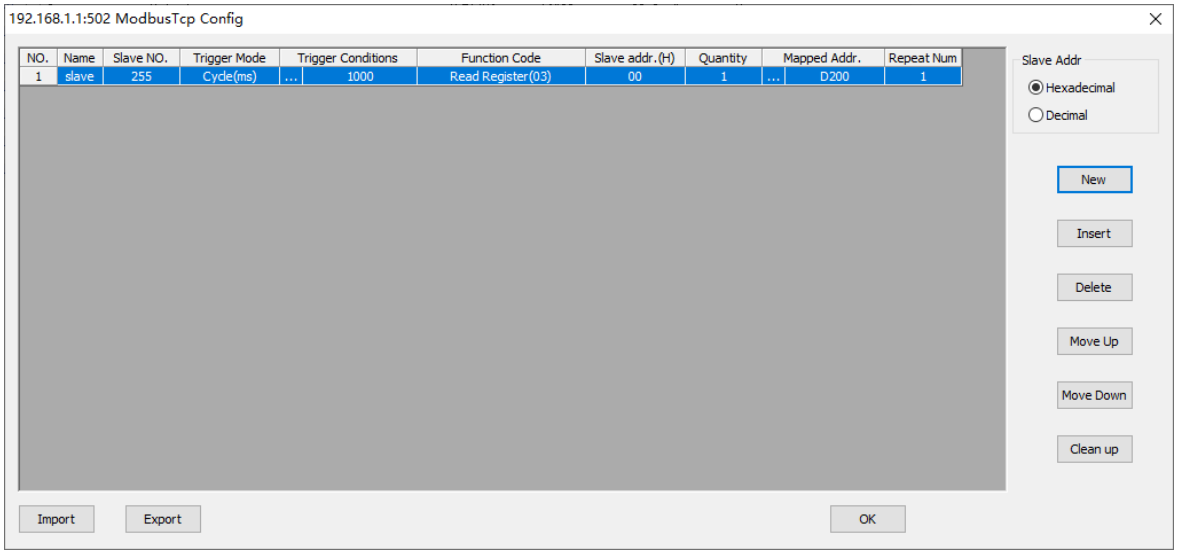


g. Click "OK". Then, the configuration of the Modbus-TCP master is generated. The default name of the Modbus-TCP slave is "ModbusTcp_0".



h. Double-click "ModbusTcp_0". On the configuration page displayed, click "New", modify the configuration, and then click "OK".

In this example, the D100 value of the Modbus-TCP slave is saved to D200 of the Modbus-TCP master.



i. Click to download the program to the PLC.

3. Monitor the D200 soft element of the Modbus-TCP master.

a. In the toolbar, click to make the Modbus-TCP master enter the monitoring state.

b. On the "Information Output Window" page, right-click a blank row in the "Element Name" column, select "Add row", and add the D200 soft element to the monitoring list.
Then, you can see that the D200 soft element reads the value of the D100 soft element of the Modbus-TCP slave.

	Element Name	Data Type	Display Format	Current Value
1	D200	INT	Dec	470
2				
3				
4				

9.4 FINS TCP&UDP Communication Configuration

9.4.1 FINS TCP&UDP Configuration

> **CAUTION**
>
> H5U and Easy PLCs can only serve as a FINS TCP&UDP slave.

A FINS TCP&UDP slave is a FINS TCP&UDP server enabled with FINS TCP&UDP and port 9600 by default. The communication protocol is not mandatory. One FINS TCP slave can be connected to up to 32 FINS TCP masters (FINS TCP clients) at a time.

Configuring a FINS TCP&UDP slave

Set the IP address of the PLC. For details, see Setting IP Address.

Configuring a FINS TCP&UDP master

On the FINS TCP&UDP master, configure the IP address, port 9600, read/write address, and read/write count of the FINS TCP&UDP slave. Then, the master can communicate with the FINS TCP&UDP slave.

Function codes and addresses

- When the PLC serves as a FINS TCP&UDP slave, the DM area (X, Y, M, D, and R soft elements) of the I/O storage area of the PLC is accessed through a FINS protocol command. The following table lists the codes of main communication commands.

Type	Primary Command	Secondary Command	Function	Description
Read/write command of I/O storage area (DM area)	0x01	0x01	Memory reading	Read I/O memory area data consecutively.
	0x01	0x02	Memory writing	Write I/O memory area data consecutively.

- When the PLC serves as a FINS TCP&UDP slave, map the PLC soft elements to the DM area, and access the elements in the format of word elements in the address range 0 to 32767 in the DM area. The following figure lists the soft element address mappings supporting access over FINS protocol.

Variable Name	Address Range	Number of Soft Elements	Note
D0 to D7999	0x0000 to 0x1F3F (0 to 7999)	8000	16-bit register
R0 to R19999	0x2000 to 0x6E1F (8192 to 28191)	20000	16-bit register
X0 to X1777	0x6F00 to 0x6F3F (28416 to 28479)	1024	Bit element
Y0 to Y1777	0x6F50 to 0x6F8F (28496 to 28559)	1024	Bit element
M0 to M7999	0x7000 to 0x71F3 (28672 to 29171)	8000	Bit element

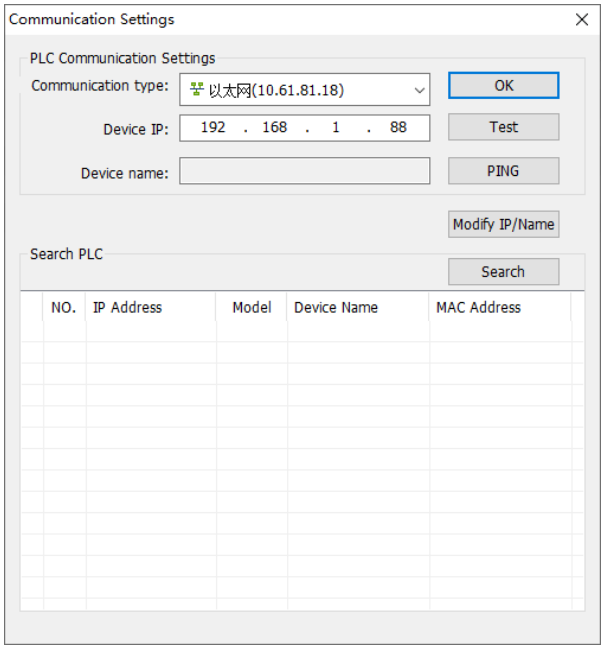
NOTE

- Bit elements within the address range 0x6F00 to 0x71F3 (X/Y/M) are still accessed in WORD mode, that is, each FINS address contains 16 bit elements. For example, when the 0X7000 address is read/written, the actual one read/written is M0 to M15.
- The number of coils corresponding to X and Y variable names (X0 to X1777/Y0 to Y1777) are in octal, instead of in decimal. For example, when the 0X6F50 address is read/written, the actual one read/written is Y0 to Y7 and Y10 to Y17.
- Not the FINS protocol addresses of each element is completely continuous and so do not access them across areas. When reading and writing multiple elements, read and write them in multiple frames.
- One FINS TCP/UDP network can have up to 254 nodes (the last bit of the IP address).
- FINS UDP communication is valid when the PLC is in the RUN state.
- When the PLC serves as a FINS TCP/UDP slave, the maximum number of elements that can be read/written consecutively is 2000 (count by word). The number of soft elements that can be read/written is configured by the FINS TCP/UDP master.

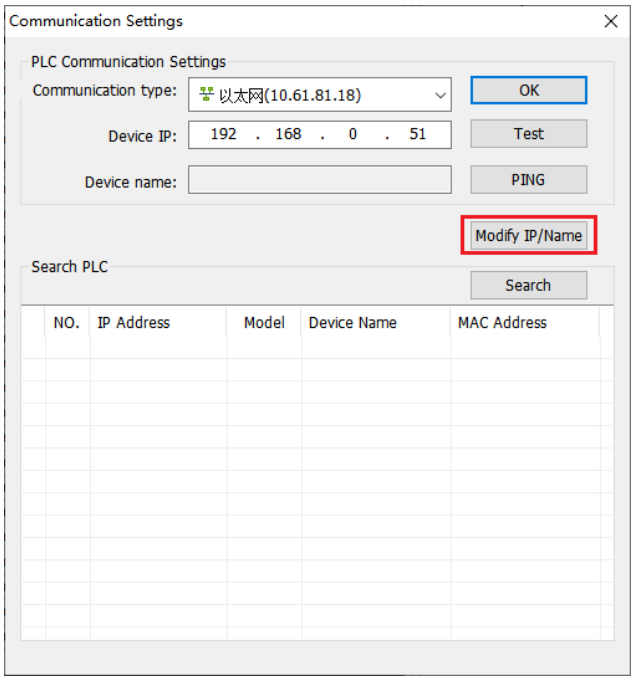
9.4.2 Application Example

In this example, an H5U device and an IT7100E HMI device are used for Ethernet connection, and they communicate through FINS TCP protocol. HMI as the master reads the values of slave registers D0 to D2, and the cycle period is 100 ms.

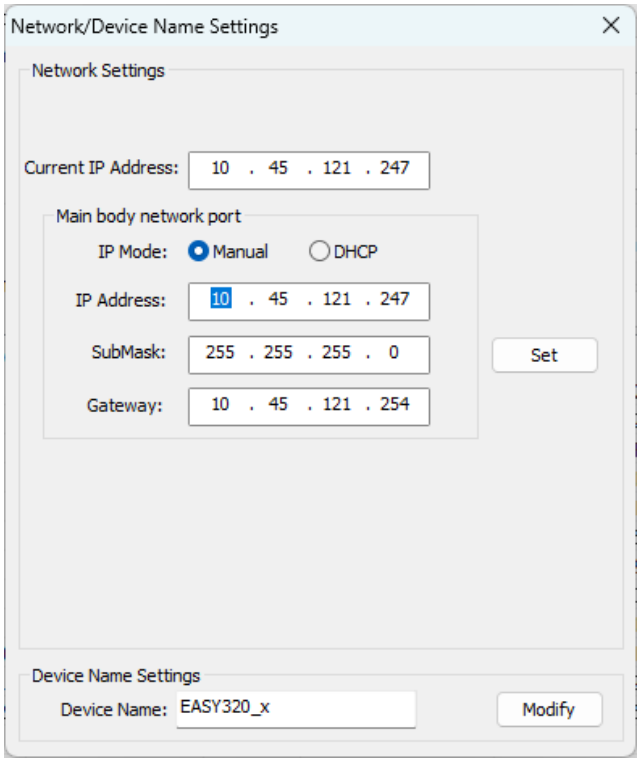
1. Configure the FINS TCP slave.
   a. Choose "Tools" > "Communication Settings" in the menu bar or click in the toolbar. The "Communication Settings" dialog box is displayed.



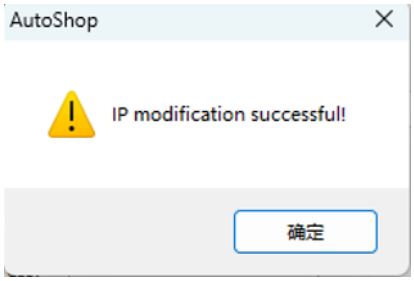
b. Click "Modify IP/Name".



c. The "Network/Device Name Settings" dialog box is displayed. Select "Manual" or '"DHCP" for "IP Mode". Set an IP address, subnet mask, and default gateway for the FINS TCP slave. Then, click "Set".
In this example, the IP address, subnet mask, and default gateway of the FINS TCP slave are 192.168.1.67, 255.255.255.0, and 192.168.1.1.



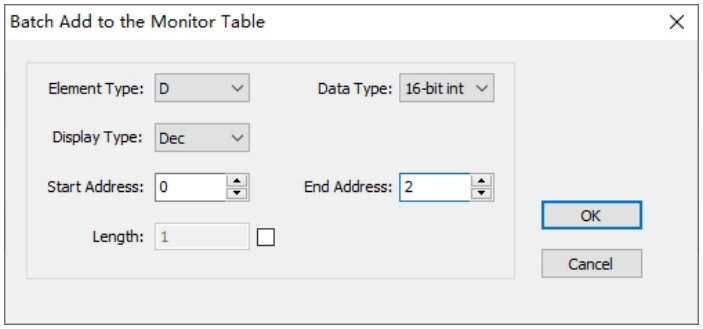
d. After the modification, the system prompts "IP modification successful!". Click "OK".



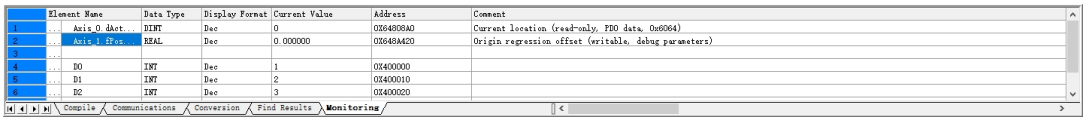
e. In the toolbar, click to make the FINS TCP slaver enter the monitoring state.

f. On the "Information Output Window" page, right-click a blank row in the "Element Name" column, and select "Batch add".

g. In the dialog box displayed, set "Element Type" to D, input 0 in the "Start Address" field and 2 in the "End Address" field, and then click "OK". The soft elements D0 to D2 are monitored.



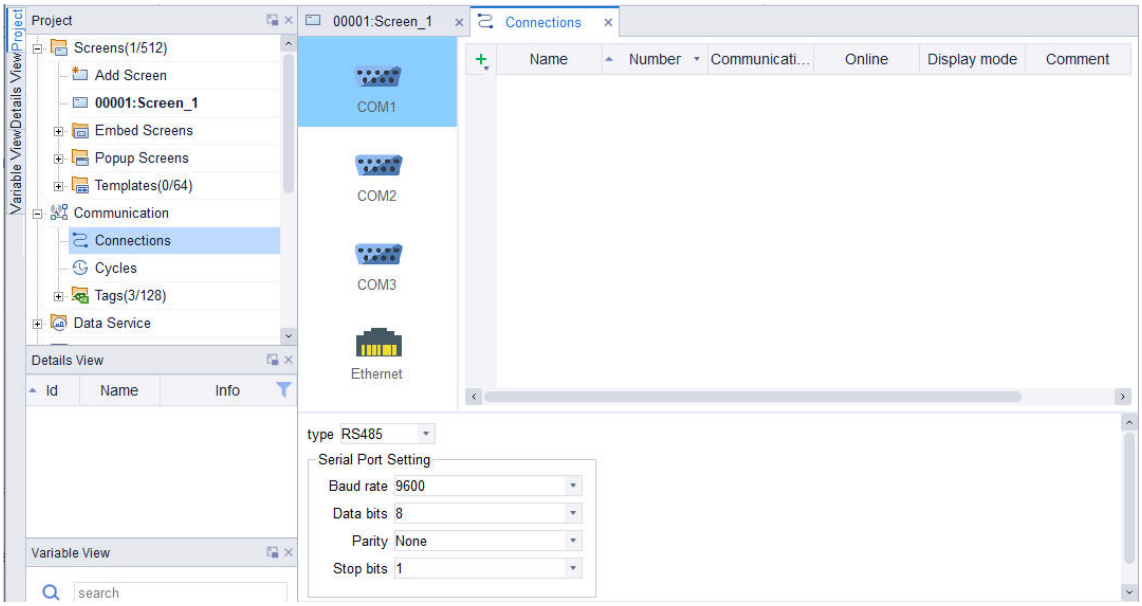
h. In the "Current Value" column, input 1, 2, and 3 for D0, D1, and D2, respectively.



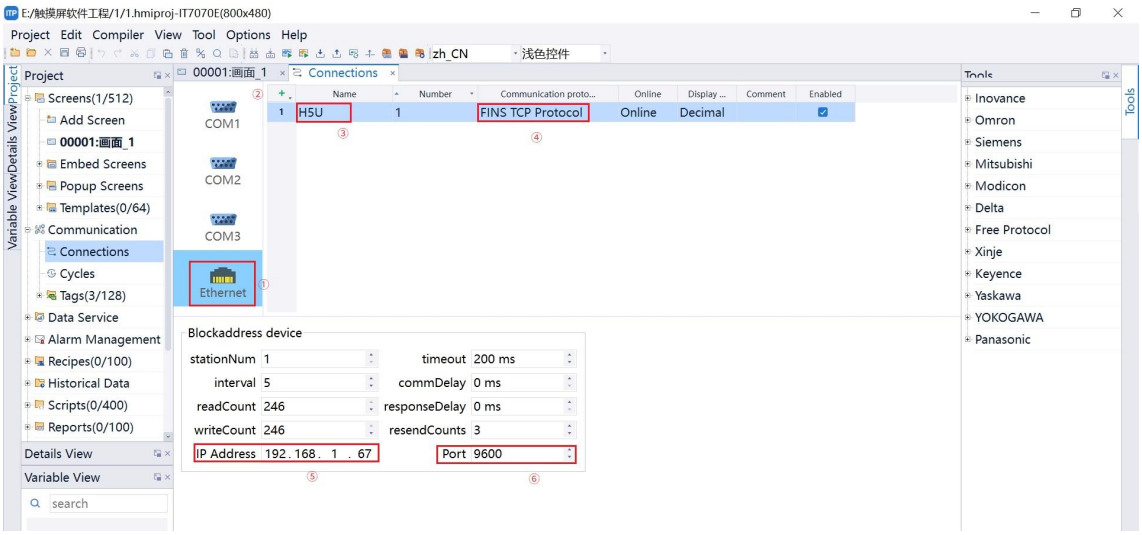
1. Configure the FINS TCP master.

a. Create a connection.

i. In the "Project" tree of InoTouchPad, unfold "Communication", and double-click "Connections" or right-click "Connections". Select "Open editor". The "Connections" page is displayed.



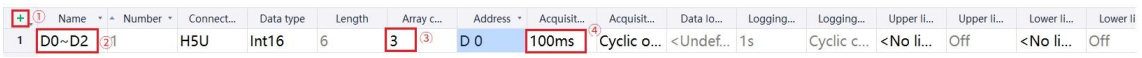
ii. Click "COM" and then click . Input a name for the connection, select "FINS TCP Protocol" in the "Communication protocol" column, and set the IP address and port number of the FINS TCP slave to 192.168.1.67 and 9600.



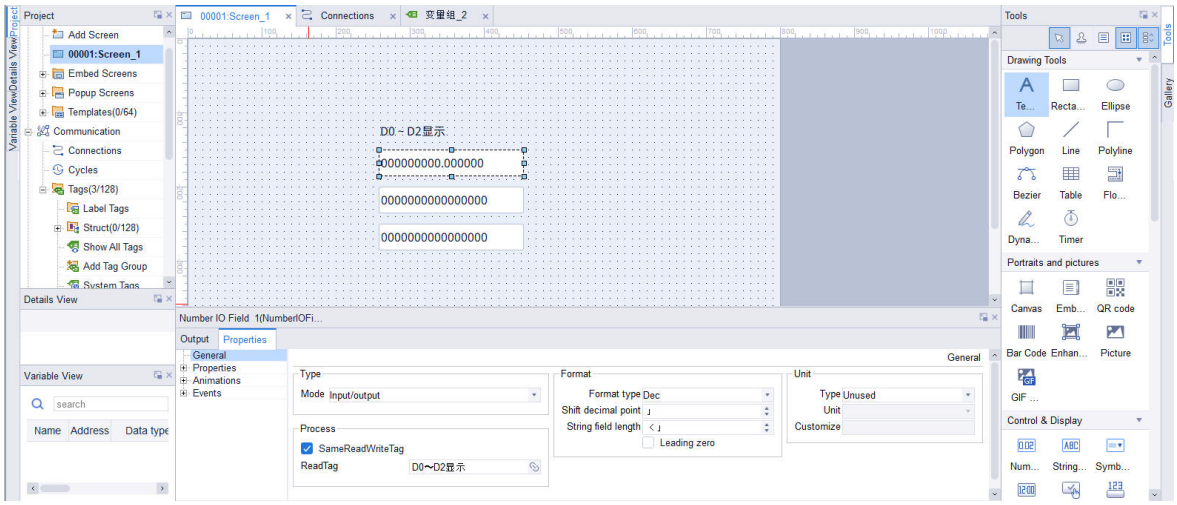
b. Create a variable.

i. Unfold "Tags" under "Communication", double-click "Tag group_2" or right-click "Tag group_2", and select "Open editor". The "Tag group_2" page is displayed.

ii. Click , input a name, set "Array Count" to 3 and "Acquisition Cycle" to 100 ms.



c. Configure the screen.



i. In the "Project" tree, unfold "Screens", and double-click "Screen_1". The "Screen_1" page is displayed.

ii. In the tools area, drag a text field to the editing area and named it as "Display of D0 to D2".

iii. In the tools area, drag the "numeric I/O field" to the editing area three times and align them vertically. Associate their read variables with D0, D1, and D2, respectively.

d. Compile and download the project to HMI and run it.

i. Choose "Compile (M)" > "Compile (M)" in the menu bar, click in the tool bar, or press F5 to compile the project.

ii. Choose "Compile (M)" > "Download Project..." in the menu bar, click in the tool bar, or press F7 to download the project to HMI and run it.

2. On the HMI display, the values of the three "numeric I/O fields" are 1, 2, and 3, means that the values of D0, D1, and D2 of the FINS TCP slave area read successfully.

9.4.3 Error Codes

In the program, common errors in the communication process can be checked through the error codes of FINS TCP/UDP slave information system variables. FINS error codes are displayed only when the communication is abnormal, so it is necessary to check the corresponding error code through Trace monitoring.

The information of FINS TCP/UDP slave information system variables is shown in the following figure.

Variable Name	Initial Value	Type	Comment
_FinsSlv		_sFINS_SLV	FINS TCP/UDP slave information
Port	0	INT	Port number
TcpErrorCode	0	INT	FINS TCP error code
UdpErrorCode	0	INT	FINS UDP error code
LogEnable	OFF	BOOL	FINS Log information switch
Reserved		DINT[40]	Reserved Field

The following table lists FINS TCP/UDP error codes.

Error Code	Description
1	The FINS TCP data frame header is incorrect.
2	The FINS TCP handshaking command frame is incorrect (Error Code is not 0).
3	The number of FINS client nodes (last bit of IP address) exceeds the limit.
4	Handshaking between the FINS TCP client and server failed.
5	Check of the fixed value of FINS command frame failed.
6	The address I/O storage area (non-DM area) accessed by read/write operation is incorrect.
7	Parameter length of FINS command frame incorrect
8	The number of continuous FINS read/write operations exceeds the limit.
9	The FINS access address range is incorrect.
10	The FINS function request code is incorrect.

If Trace monitoring TcpErrorCode/UdpErrorCode does not display any error code when a communication error occurs, use LogEnable to enable FINS log printing to troubleshoot the communication error.

> **CAUTION**
>
> When a communication error occurs, frequent FINS log printing will affect the normal operation of the PLC; therefore, do not enable FINS log printing during normal operation.

9.5 SFTP Function

The SFTP communication function is a secure file transfer service based on the SSH (Secure Shell) protocol, primarily used for encrypted file transfer between a client and a server.

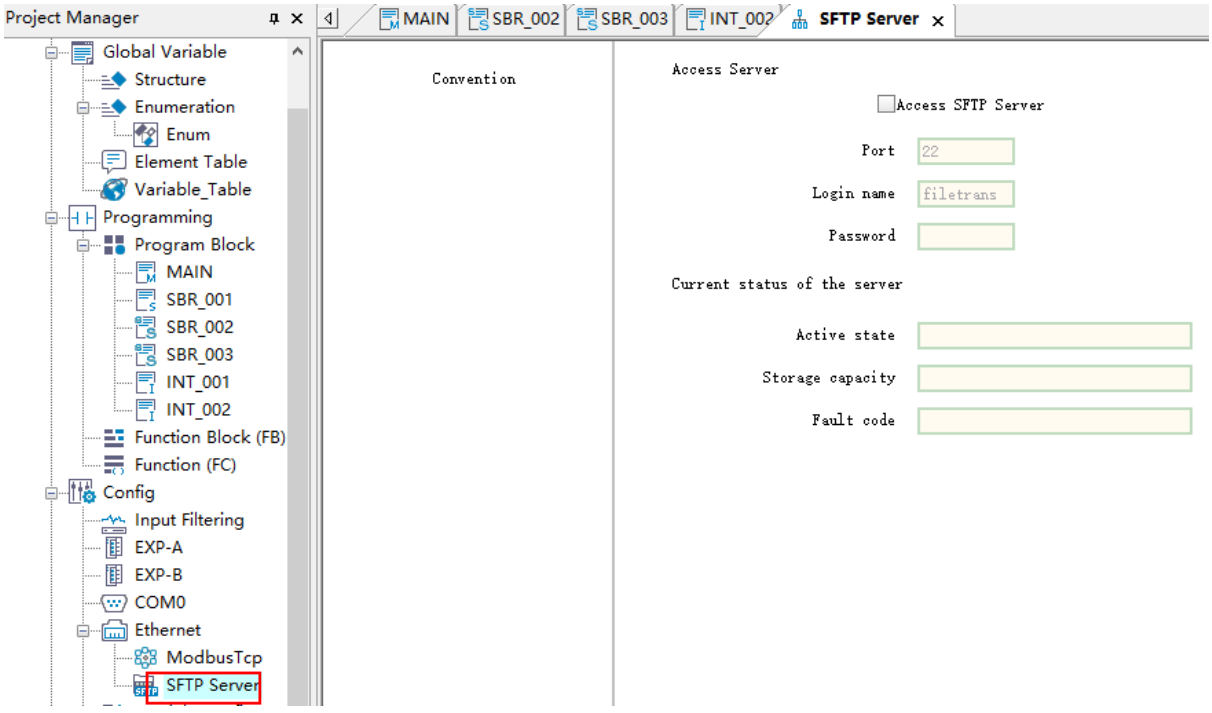
> **CAUTION**
>
> - The SFTP function must be used with an SD card (GE20-TF-RTC extension card). Use of or access to the PLC's local storage space is not permitted.
> - When the SFTP function is enabled, the mounted SD card must have a capacity greater than 256 MB.
> - A maximum of 10 connections are allowed.

Preconditions

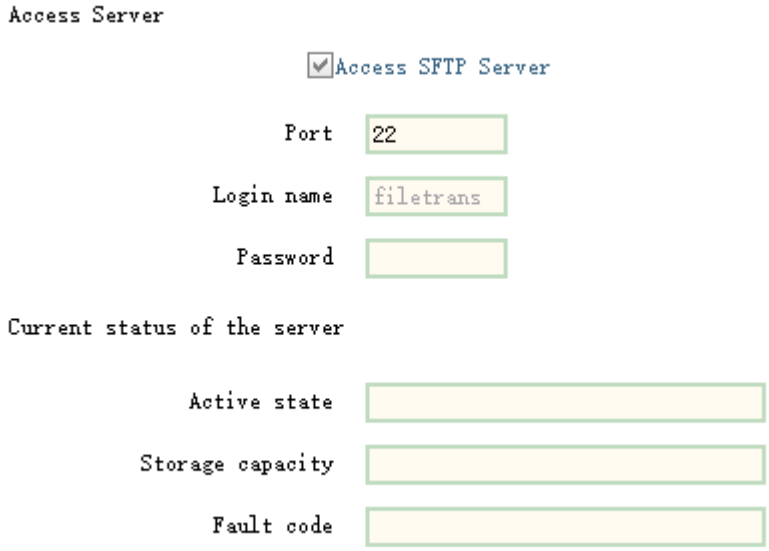
- PLC: Easy series
  - The Easy511 series only supports local Ethernet port configuration, not expanded Ethernet port configuration.
  - Easy series models without an Ethernet port are not supported.
- Firmware version: V6.5.0.0 and later
- Software version: AutoShop V4.11.0.0 and later

Procedure

1. In the "Project Manager" tree on the left, choose "Config" > "Ethernet" > "SFTP Server". The SFTP server configuration page is displayed.



2. Select "Access SFTP Server", and set the port, login name, and password.



The following table lists the speed compensation parameters.

Parameter Name		Description
Access SFTP Server	Activate the SFTP server.	After this option is selected, the SFTP server is started. It is not selected by default.
	Port	The port number of the SFTP server to access. The port number for accessing the SFTP server. Range: 22, 1024 to 49151. It cannot be set to 2222, 9600, 12939, 12940, 34980, or 44818. The default is 22.
	Login name	The username used for authentication when accessing the PLC SFTP server. It supports an 8-character combination of numbers and letters. The default is filetrans and cannot be changed.
	Password	The password of the account used for authentication when accessing the PLC SFTP server. It can be a combination of any numbers or letters. Special characters are not supported. It must be 8-characters long.
Current status of the server	Active state	• Not Activated: Indicates the SFTP server is not started. • Activated: Indicates the SFTP server has been activated. • Version Not Supported: Indicates the connected PLC does not support the SFTP server function.
	Storage capacity	• No SD Card: Indicates no SD card is inserted into the GE20 extension module. • Displays the Available Capacity/Total Capacity values. ▪ Available Capacity refers to the remaining usable space for the current SFTP server. ▪ Total Capacity refers to the storage space allocated by the PLC to the SFTP server.

Parameter Name		Description
	Fault code	• "No Error" is displayed when normal, indicating the SFTP server is running without error. • The following fault information is displayed when abnormal: ▪ Configuration parameter obtaining error ▪ Port number error ▪ Username error ▪ Failed to change password ▪ Failed to add a new user ▪ SD card error ▪ Failed to mount the SD card ▪ Failed to start the SFTP server ▪ SFTP server exited abnormally ▪ Failed to modify the SFTP server configuration ▪ SFTP server configuration error

3. After the project is downloaded and the SFTP server is started, the "Current status of the server" section displays online monitoring information.

9.6 (Optional) Network Firewall

9.6.1 Network Firewall

Network firewalls can identify devices with specified IP address or MAC address in Ethernet, and screen or restrict unauthorized devices from accessing the PLC, thus ensuring the data and operation safety of the PLC.

9.6.2 Configuring a Network Firewall

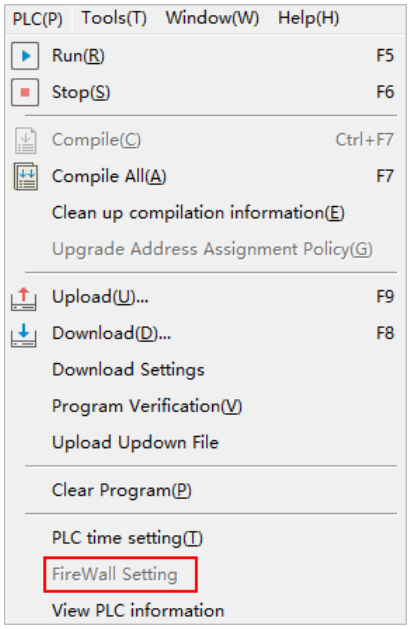
On the network firewall configuration page of AutoShop, set a device allowlist and blocklist, and download the firewall configurations to the PLC. Then, only the specified devices can access the PLC.

Background

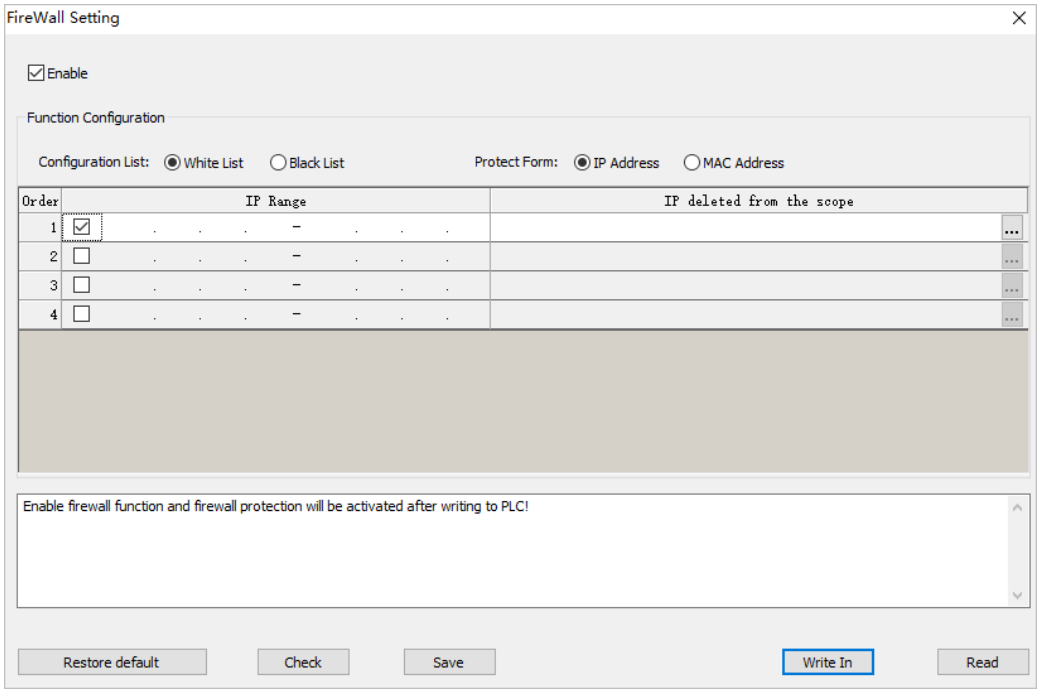
- Only Easy series small PLCs support the network firewall function, and the specific models are Easy320, Easy521, Easy522, and Easy523.
- PLCs with firmware version of 6.3.0.0 and later support the network firewall function.
- Ensure that the connection between the background and the PLC is normal; otherwise, it may fail to read and write firewall configuration data from and to the PLC.

Procedure

1. In the menu bar, choose "PLC" > "FireWall Setting" to access the "FireWall Setting" page.
   This button is grayed for projects not supporting network firewall.



2. In the "FireWall Setting" dialog box, select "Enable" to enable this function and edit configurations.



3. Select "White List" or "Black List" for "Configuration List". The allowlist contains devices which are authorized to access the PLC. The blocklist contains devices which are not allowed to access the PLC.

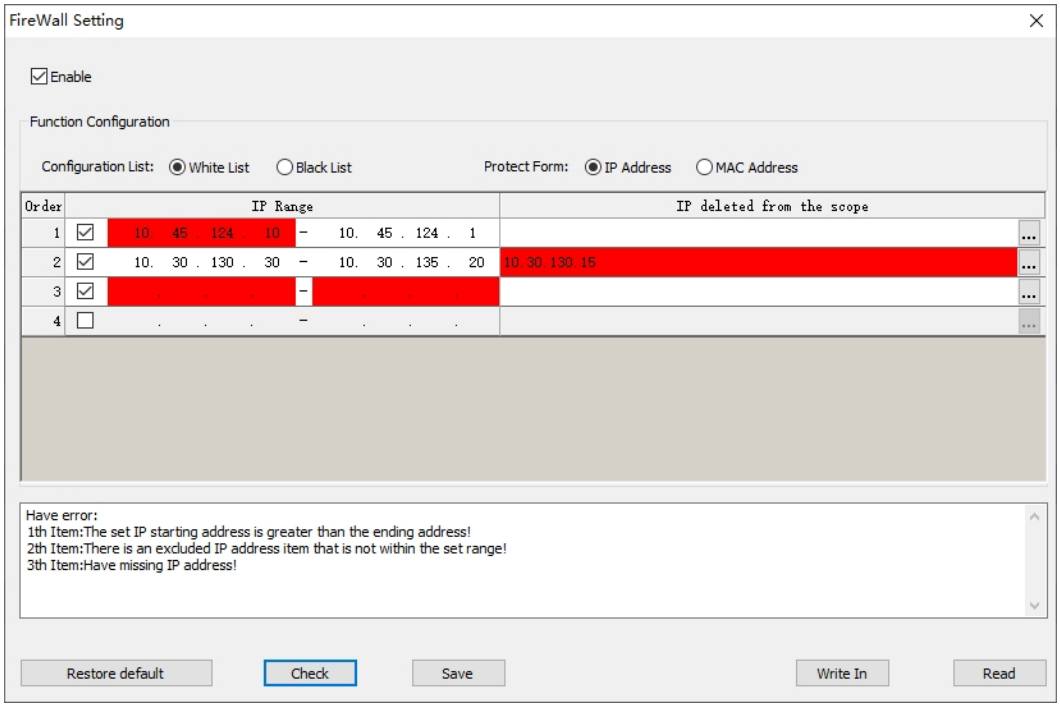
4. Set the protection form.
   - IP Address: Select "IP Address" for "Protect Form", and select items in the table below to edit the information.
     - In the "IP Range" column, select the start and end IP addresses. IP addresses in different network segments can be selected.
     - In the "IP deleted from the scope" column, set IP addresses to be excluded from the IP address range. Double-click this column or click the button on the right of this row to enter the settings page.
       Up to 16 IP address exceptions can be set and the object IP address range can be modified.

       After setting, click "OK". The configurations are updated to the parameters on the main settings page. Click "Cancel", and then the current setting is canceled.
   - MAC Address: Select "MAC Address" for "Protect Form", and select items in the table below to edit the information.
     You can set MAC addresses of up to 100 specified addresses and the MAC addresses must be in the same network segment.

   Right-click the list to perform operations such as Copy, Paste, Import, and Export.

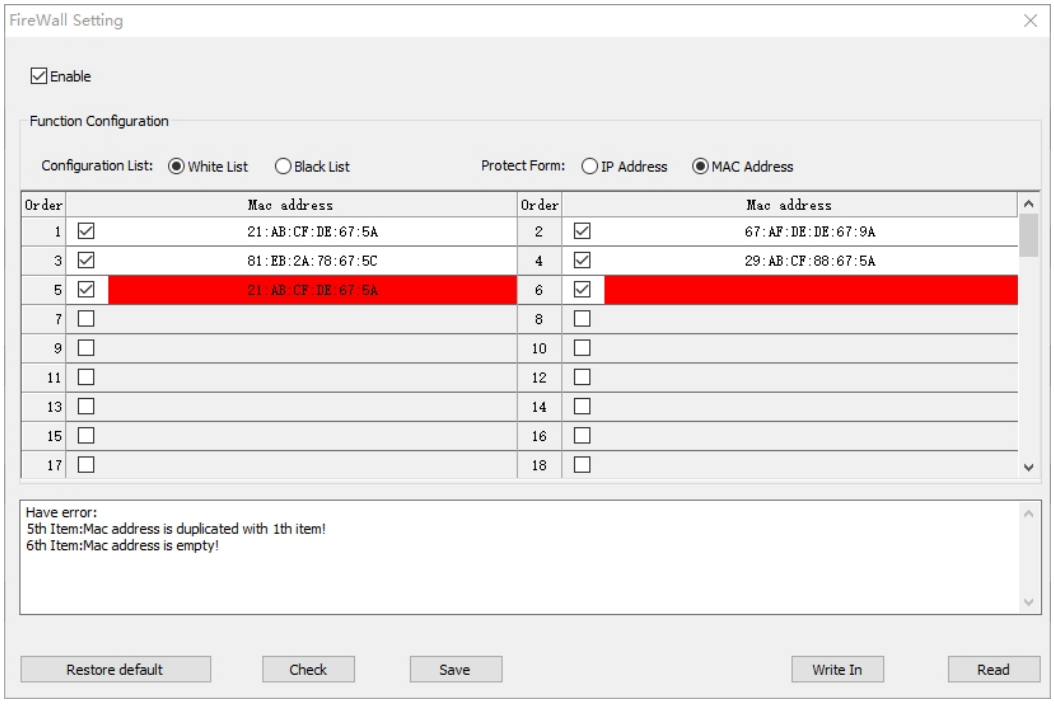
   - When you select "Export", the IP address configuration table or MAC address configuration table is exported to a CSV file.
   - When you select "Import", the current table is cleared and the operation cannot be canceled. If any data to be imported contains incorrect information, an error is reported in the information output window and this data is not imported. After data import, information on the configuration page is overwritten and updated.
5. Click "Restore default". In the dialog box displayed, click "OK". All the data on the configuration page is cleared and restored to the status when the item is not enabled.
6. Click "Check". The system checks the validity of the data on the configuration page and prompts the error if any, to prevent illegal configurations from being written to the PLC.
   - The following describes check of the IP address.
     - The start IP address must be smaller than or equal to the end IP address; otherwise, an error is reported.
     - When the excluded IP addresses are not within the target IP address range specified, an error is reported.
     - When IP address ranges overlap, an error is reported.
     - When the start or end IP address of an enabled item is not set, an error is reported.

     The following figure is an IP address check example.



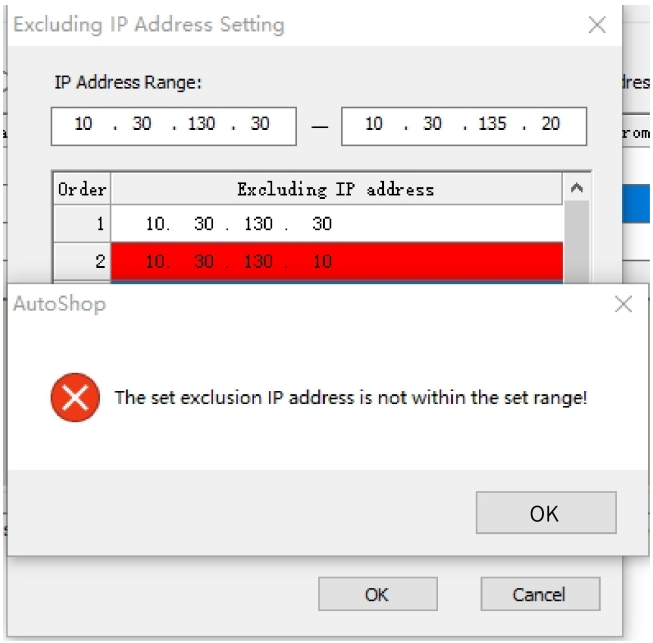
- The following describes check of the MAC address.
  - A MAC address consists of 12 hexadecimal numbers, and every two hexadecimal numbers are separated by ":".
  - The enabled item cannot be empty.
  - MAC addresses cannot be repeated.

  The following figure is a MAC address check example.



- The following describes check of the excluded IP addresses.
  - The target start IP address modified on the excluded IP address page must be smaller than the end address and cannot be empty.
  - The excluded IP addresses must be within the target IP address range.
  - Excluded IP addresses cannot be repeated.

As shown in the following figure, in the "Excluded IP address settings" dialog box, click "OK". Then the system checks the settings and displays the incorrect information in red.

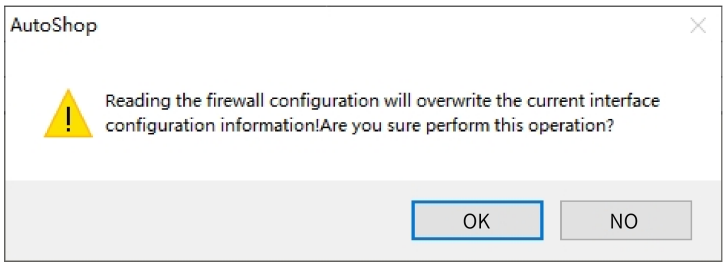


7. Click "Write In" or "Read" to read or write settings of the network firewall from or to the PLC in real time.

- Before writing data to the PLC, the system first checks the content validity of the current IP address or MAC address configuration page to prevent illegal information from being written to the PLC.
  When data is written to PLC or firmware that does not support network firewall, the following prompt is displayed.

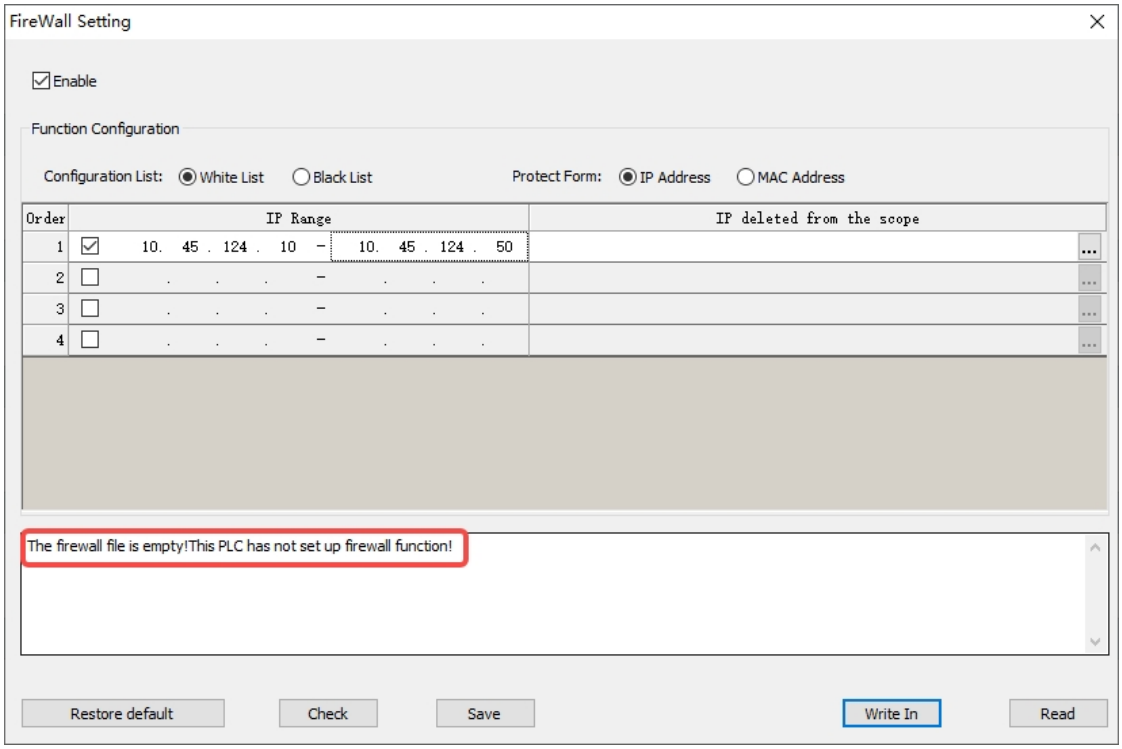
  After data writing, the network firewall takes effect immediately. When an allowlist is configured, only devices with IP addresses or MAC addresses on the list can access the PLC through Ethernet. When a blocklist is configured, devices on the list cannot access the PLC.

- When reading the network firewall configurations from the PLC, the following prompt is displayed.



  When you click "Yes", after data reading, configurations on the current page are cleared, overwritten, and updated. When you click "No", the reading operation is canceled.

  When the PLC is not enabled with the network firewall function of the configurations are cleared after firmware programming, the system prompts "Firewall file is empty! The PLC has no firewall function!", and the configurations on the page are retained, as shown in the following figure.

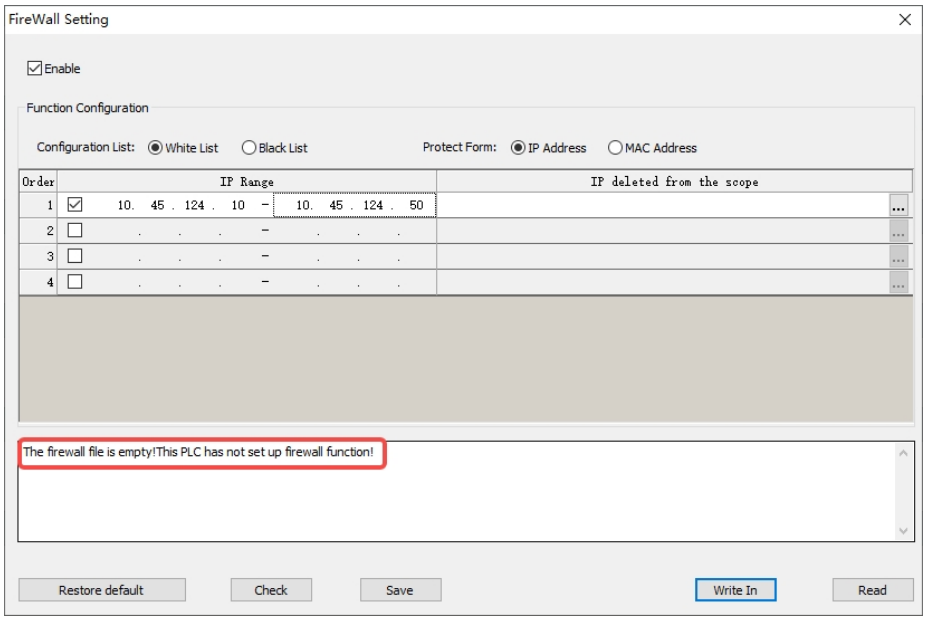


8. When the network firewall function is enabled, you can perform operations such as Copy, Paste, Cut, Delete, All, Undo, and Restore through shortcut keys or context menu.

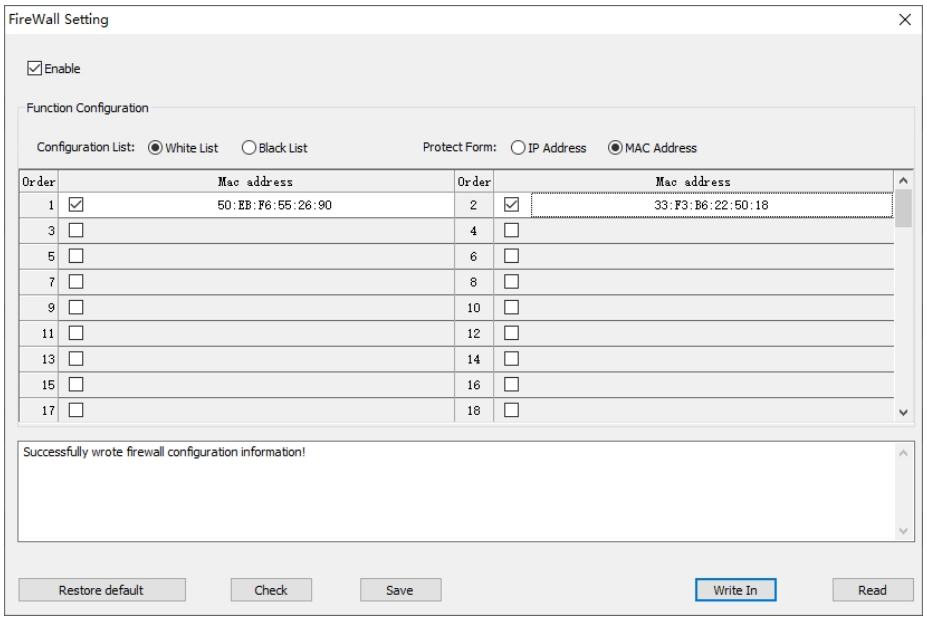
9. The network firewall configurations are packed into the Down or Updown file generated. When the Down or Updown file containing network firewall configurations is downloaded to the PLC through an SD card, the network firewall configurations are overwritten and updated.
   Before generating the Down file, the system first checks the validity of configurations and displays the incorrect information in the information output window.

9.6.3 Application Example

- Configure an IP address allowlist
  Configure the network firewall and write the configuration to the PLC, as shown in the following figure. Then, only devices with an IP address within the segment 10.45.126.10 to 10.45.126.80 (with 10.45.126.11, 10.45.126.12, and 10.45.126.13) can access this PLC through Ethernet, and other devices cannot access this PLC.



- Configure a MAC address blocklist
  Configure the network firewall and write the configuration to the PLC, as shown in the following figure. Then, devices with a MAC address of 50:EB:F6:55:26:90 or 33:F3:B6:22:50:18 in the same network segment of the PLC cannot access this PLC. Other devices can access this PLC. Devices in other network segments instead of the MAC address segment are not restricted by the firewall.



10 CAN Communication

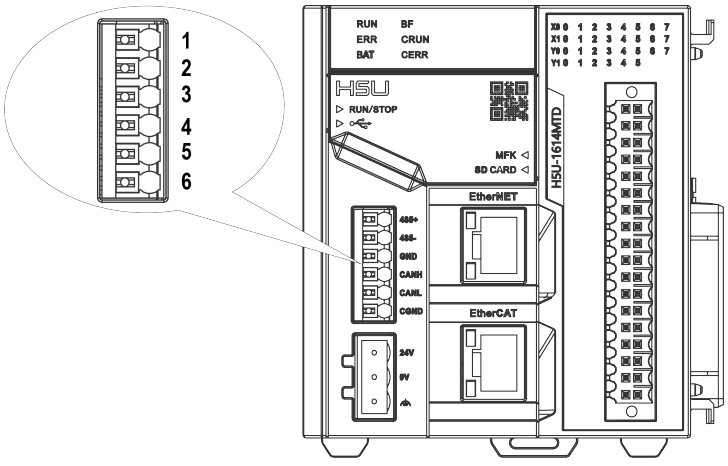
10.1 Overview

- H5U provides a CAN communication port that supports the CANlink and CANopen protocols. The CANlink protocol supports up to 62 extension slaves, and CANopen protocol supports up to 30 extension slaves.
- The Easy series (Easy302/Easy320/Easy501/Easy502/Easy521/Easy522/Easy523) support the CANlink and CANopen protocols. They support one master. The CANlink protocol supports up to 62 extension slaves, and CANopen protocol supports up to 30 extension slaves.

> **NOTE**
>
> This function requires a firmware version of V5.65.2.0 or later and a software version of V4.6.5.0 or later for the Easy302, Easy320, Easy501, and Easy502 series.
>
> It requires a firmware version of V5.66.0.0 or later and a software version of V4.8.0.0 or later for the Easy521, Easy522, and Easy523 series.

10.2 Hardware Interface

The CAN communication port and RS485 port of H5U are integrated to a 6-pin port. CAN extension cards of the Easy series use the RJ45 network port. For details, see the GE20-CAN-485 Communication Extension Card User Guide.



The following table lists the assignments of the six pins.

Pin	Signal Definition	Description
1	485+	Positive signal of the RS485 differential pair for COM0
2	485-	Negative signal of the RS485 differential pair for COM0
3	GND	Power ground of COM0
4	CANH	CAN communication data receiving terminal

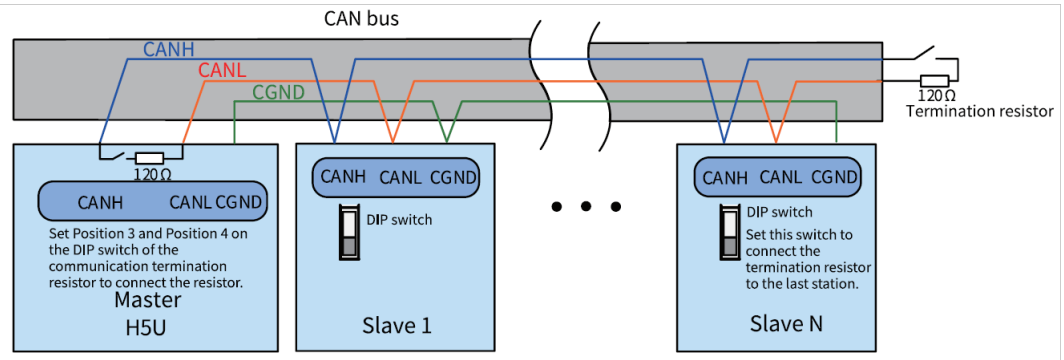
Pin	Signal Definition	Description
5	CANL	CAN communication data sending terminal
6	CGND	CAN communication ground

10.3 CAN Network

10.3.1 CAN Communication Networking

The three wires of each device must be interconnected to form a CAN. 120 Ω termination resistor must be provided at both sides of the CAN bus (both the H5U and CAN extension card have a built-in resistor, which can be connected by setting the DIP switch. The default value is ON).

The multi-device CAN bus wiring diagram of H5U is as follows.



NOTE

The CGND terminals of all the devices must be connected together.

10.3.2 Relationship Between CAN Communication Distance and Baud Rate

The following table lists relationship between CAN communication distances and baud rates.

Baud rate (kbps)	Distance (m)	Minimum cable diameter (mm^2)	Maximum number of access points
1000	20	0.3	18
500	80	0.3	31
250	150	0.3	31
125	300	0.5	31
100	500	0.5	31
50	1000	0.7	31

10.3.3 System Variable of CAN Interface

H5U provides a system variable named “_CAN” to view or monitor the status of CAN port. “_CAN” is a structure variable whose data type is “_sCAN”. The following table lists its members and definitions.

Member	Data type	Description
BaudRate	INT	Baud rate, in the unit of kbps
LoadRate	INT	Network load rate, in the unit of %
RxPexSec	INT	Number of messages received per second, in the unit of FPS
TxPexSec	INT	Number of messages sent per second, in the unit of FPS
RxErrCnt	INT	Count of errors received by the CAN controller
TxErrCnt	INT	Count of errors sent by the CAN controller
Protocol	INT	Communication protocol. 0: CANlink; 1: CANopen

10.4 CANlink Communication

10.4.1 Principle of CANlink3.0 Communication

CANlink3.0 communication is implemented through CAN network configuration. When downloading user programs, you need to download CAN network configurations to the PLC.

Understanding the principle of CANlink3.0 network configuration can help you complete the CAN configuration table.

One CANlink3.0 network can have only one master station but can have one or more slave stations.

Master and slave stations on a CANlink3.0 network communicate with each other by automatically sending and writing data rather than in query-response mode.

Example:

- To send data to slave stations, the master station "writes" register data in slave registers based on CANlink communication configurations when trigger conditions are met.
- Slave stations automatically send data to the master station and "write" the data in the receiving unit of the master station based on CANlink communication configurations.
- Slave stations automatically send data to each other and "write" the data in receiving units of slave stations based on CANlink communication configurations.
- To send data to multiple stations, a station automatically sends the "write operation" data to itself (equivalent to broadcasts) based on CANlink communication configurations, while the other stations selectively receive the data and automatically store it in their receiving units.
- For efficient data exchange during network communication, master and slave stations save "heard" broadcast data sent by other stations. You need to click "Receive Config" to set receiving slave station numbers and addresses. In this way, the stations configured as receiving stations will ignore the broadcast data from stations not configured as sending stations.

You do not need to configure CANlink3.0 slave stations because CANlink configurations can be transmitted to slave stations through a master PLC. Therefore, CANlink3.0 communication configuration items for slave stations are forwarded by the CANlink master station through configuration frames.

Upon startup, the master station sends configuration frames to CANlink slave stations and assigns the list of communication tasks. Slave stations automatically send data based on the list.

CANlink3.0 configuration items include the address of the sent register, address of the target receiving slave station, number of data entries, address of the received register, interval for sending, and trigger condition, which are required by common communication instructions. Different from common communication operations, "communicate-write" operations do not need responses.

In communication scenarios where multiple slave stations must synchronously respond (for example, servo-driven synchronous multi-axes control and position-controlled high-speed movement), you need to set Synchronous Write for the master station. The master station writes data to slave stations and then sends broadcast command frames to make slave stations run simultaneously.

10.4.2 System Variable of CANlink Interface

The following table lists CANlink ports.

Name	Data Type	Description	R/W	Comparison with H3U
_CANLink.Address	INT	Station No. or address	R	D8284
_CANLink.Heartbeat	INT	Heartbeat time (ms)	R	D8282
_CANLink.NetworkStart	BOOL	Network startup	R	M8290
_CANLink.SyncTrigger	BOOL	Sync triggering	R	M8291
_CANLink.SyncWrErr	INT	Synchronous write error code	R	D8307
_CANLink.ConfigErr	INT	Configuration error code	R	D8308
_CANLink.NodeState	INT[64]	Station status (=2: online; ≠2: offline)	R	-
_CANLink.NodeState[0]	INT	Local station status (=2: online; ≠2: offline)	R	D7800
_CANLink.NodeState[1]	INT	1# station status (=2: online; ≠2: offline)	R	D7801
_CANLink.NodeState[…]	INT	-	R	-
_CANLink.NodeState[63]	INT	63# station status (=2: online; ≠2: offline)	R	D7863
_CANLink.Online	DINT[2]		R	-

Name	Data Type	Description	R/W	Comparison with H3U
		Station status on the CANlink configuration monitoring page of AutoShop, with one bit indicating a station		
_CANLink.Online[0]	DINT	-	R	D8241
_CANLink.Online[1]	DINT	-	R	D8242
_CANLink.Disable	BOOL[64]	Enables (set to ON) or disables (set to OFF) a slave. The array elements of 0 and the master No. are meaningless.	R/W	M8290
_CANLink.Disable[0]	BOOL	-	R/W	-
_CANLink.Disable[1]	BOOL	-	R/W	-
_CANLink.Disable[…]	BOOL	-	R/W	-
_CANLink.Disable[63]	BOOL	-	R/W	-

10.4.3 CANlink Indicators

You can judge the CANlink communication state based on the CANlink indicator, as shown in the following table .

Indicator	Status	Description
Communication (green)	Gray	CANlink bus not connected or disconnected
	ON	CANlink bus connected (remote frames received on the node)
	Flashing (≤ 3Hz)	During CANlink communication, one flashing per frame of bus data sent or received
	Blinking (≤ 5Hz)	Flag monitor
Fault (red)	Gray	No fault
	ON	Monitor timeout (node), no node (monitor)
	Blinking (≤ 0.5Hz)	CANlink configuration error (for the configurator)
	Blinking (≤ 1Hz)	Node lost or crash (for the monitor)

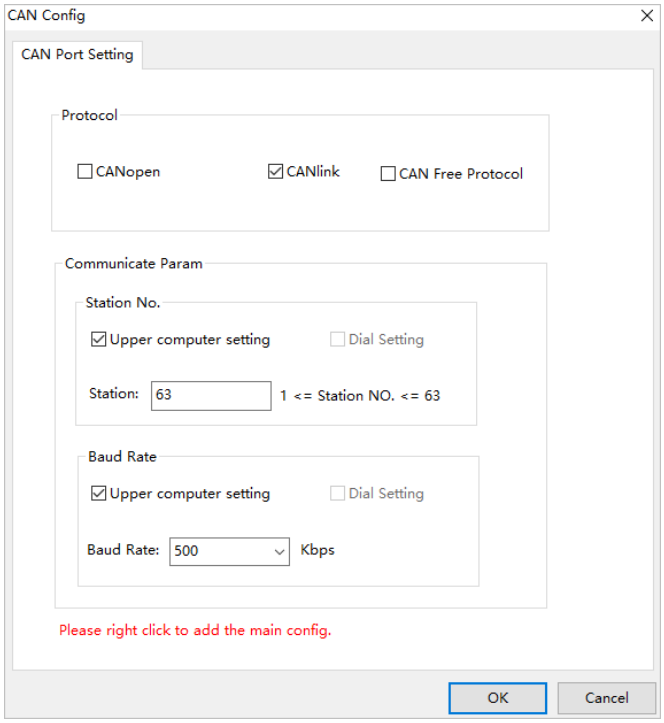
Indicator	Status	Description
	Blinking (≤ 5Hz)	CANlink address conflict

10.4.4 Configuring a CANlink Network

Take the following steps to configure a CANlink network

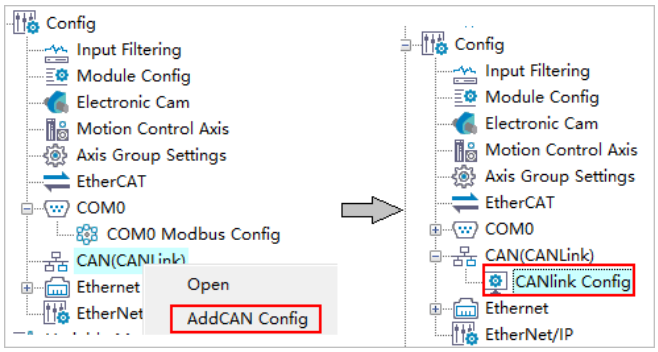
1. Configure the CANlink network through AutoShop and define the data to be exchanged.
2. Download configurations to the H5U series PLC.

After creating a project, choose "Project Manager" > "Config", and then double-click "CAN". In the window displayed, select "CANlink" as the communication protocol. Then, the system automatically determines whether the current PLC is the CANlink master station or the CANlink slave station based on the presence of the CANlink configuration.



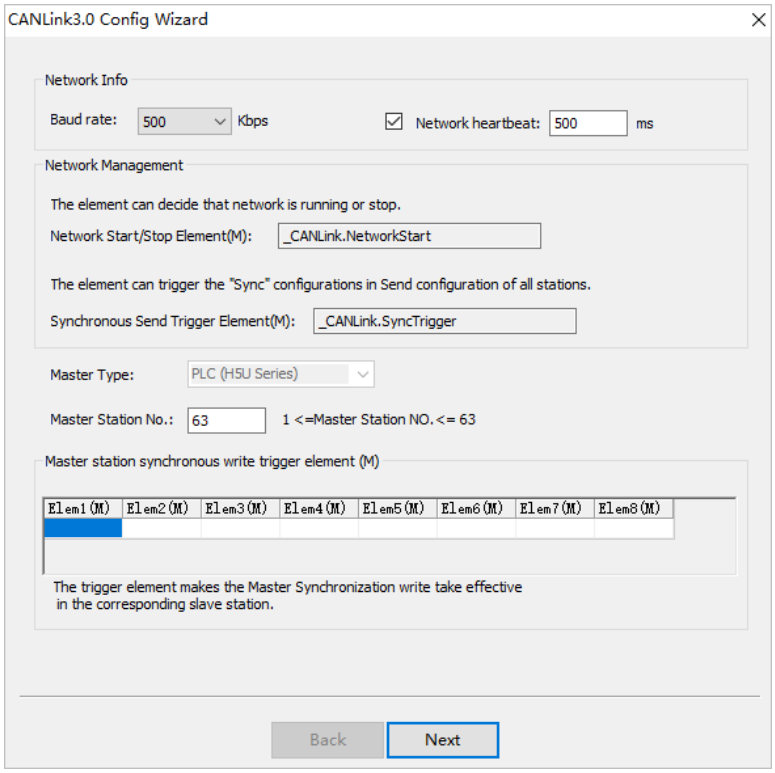
Select "CANlink", set the slave No. and baud rate, and then click "OK".

In this example, CAN is configured as the CANlink slave station. Choose "Project Manager" > "Config". Right-click "CAN". In the window displayed, select "Add CAN Config" to configure it as the CANlink master station, as shown in the following figure.



CANlink3.0 Config Wizard page

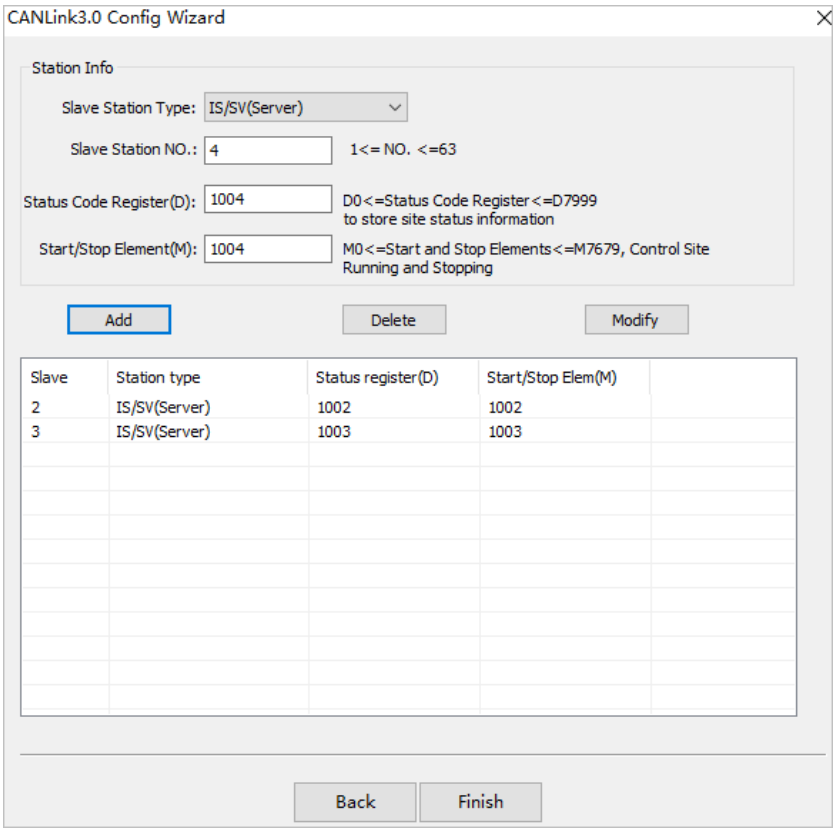
Double-click "CANlink Config". The "CANlink3.0 Config Wizard" page is displayed, as shown in the following figure.



- Baud rate (required)
  Eight options are available for different scenarios: 20 kbps, 50 kbps, 100 kbps, 125 kbps, 250 kbps, 500 kbps, 800 kbps, and 1 Mbps. You can select the desired option from the drop-down list, and then download the configuration to the master station (this parameter is valid for the master station only, and needs to be manually modified on a slave station). You can select the baud rate based on the bus load and communication distance.
- Network heartbeat (optional)
  All slave stations send heartbeats to the master station at a specified interval. The master station monitors the state (online or offline) of each slave station through the heartbeat mechanism. Slave stations monitor the status of the master station through its heartbeats. (An interval of more than 200 ms is recommended.) If you deselect this parameter, the heartbeat function is disabled and the system cannot monitor the network.
- Master Station NO. (required)
  In this example, the master station No. is the number of the PLC that serves as a master station. The number cannot be changed. If the number entered is inconsistent with the actual number, the PLC will determine that the downloaded configuration is invalid.

  For example, if you enter 7, the configuration is valid only when downloaded to station 7. Station 7 then assigns the configuration to other stations. The CANlink network configuration is downloaded to the master station and then assigned to slave stations. In this way, the system can monitor and manage the entire network through the master station in the background.
- Master station synchronous write trigger element (optional)
  It is an element triggering Synchronous Write for the master station. When a trigger element (M) is set, the corresponding configuration takes effect. The element is automatically reset after data is sent.

Click "Next". The window for adding a slave station is displayed as follows.



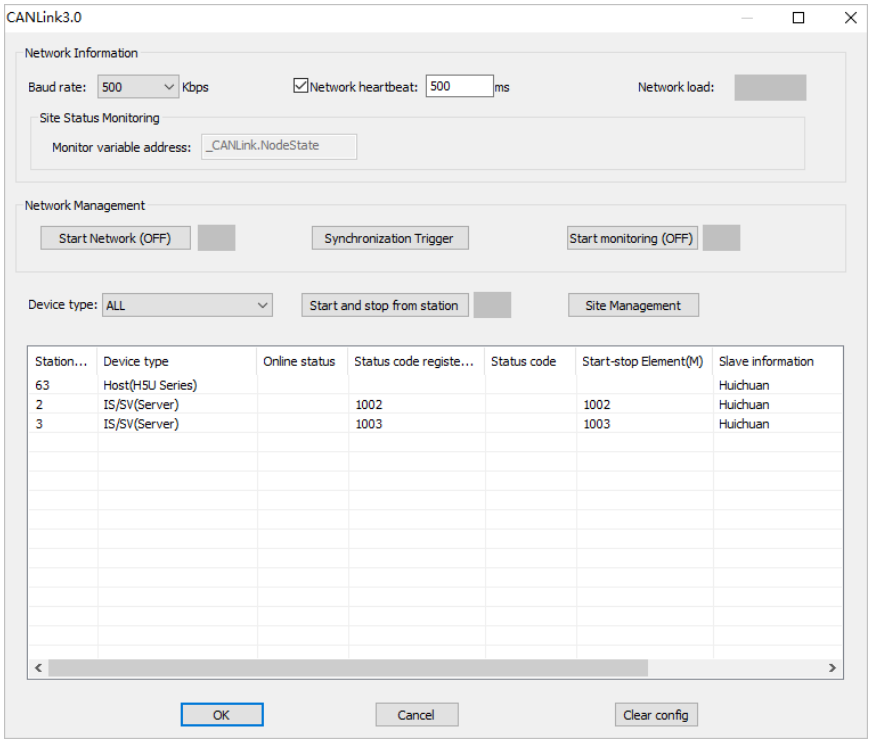
Station information page

- Add
  After configuring a slave station, click "Add". The station is added to the list.
- Delete
  Select a station and click "Delete". In the confirmation dialog box "Delete it?", click "OK" (you can delete multiple stations at a time).
- Modify
  Select a slave station, modify parameters on the "Station Info" page, and then click "Modify". Do not modify the station type.
- Slave Station NO.
  Set the CANlink slave station number.
- Status Code Register (D)
  It is used to save the status of a slave station fed back through heartbeat frames of the slave station.
- Start/Stop Element (M)
  It is an M element used to start or stop communication. When M is ON, communication is started. When M is OFF, communication is stopped.

NOTE

In the configuration wizard, click "OK" to save the modification made in the wizard and exit; click the "X" button in the upper right corner of the wizard to cancel the modification and exit.

Then, click "Finish". The following window is displayed.



- ◦ Network load ≤ 50: Green (good)
  - ◦ 50 < Network load ≤ 75: Yellow (warning)
  - ◦ 75 < Network load ≤ 90: Red (major warning)
  - ◦ Network load > 90: ERR, red background (error)Network Information
    Baud rate: Indicates the baud rate of the master station.

    Network heartbeat: Indicates that the heartbeat function is enabled after this parameter is selected.

    Network load: Calculates the real-time load of the network (this parameter is displayed only when the network load is monitored during running of devices).

- Site Status Monitoring
  The online state of the station will be updated to the system variable _CANlink.NodeState[64], in which _CANlink.NodeState[0] is the status of the local station, while _CANlink.NodeState[station No.] is the state of the slave station.

Status Value	Definition
1	Configurations of the slave station are available.
2	The slave station is running.
5	The slave station is disconnected.

NOTE

If the heartbeat function is disabled, the station monitoring function is meaningless.

- Network Management
  Start/Stop Network (OFF) (enabled when monitoring is enabled): Starts and stops network communication.

Synchronization Trigger: Synchronization will be triggered. You can enable the function in the user program by setting _CANlink.SyncTrigger. After synchronous data frames are sent, _CANlink.SyncTrigger will be automatically reset.

Start monitoring/Stop monitoring (OFF): Starts and stops network monitoring.

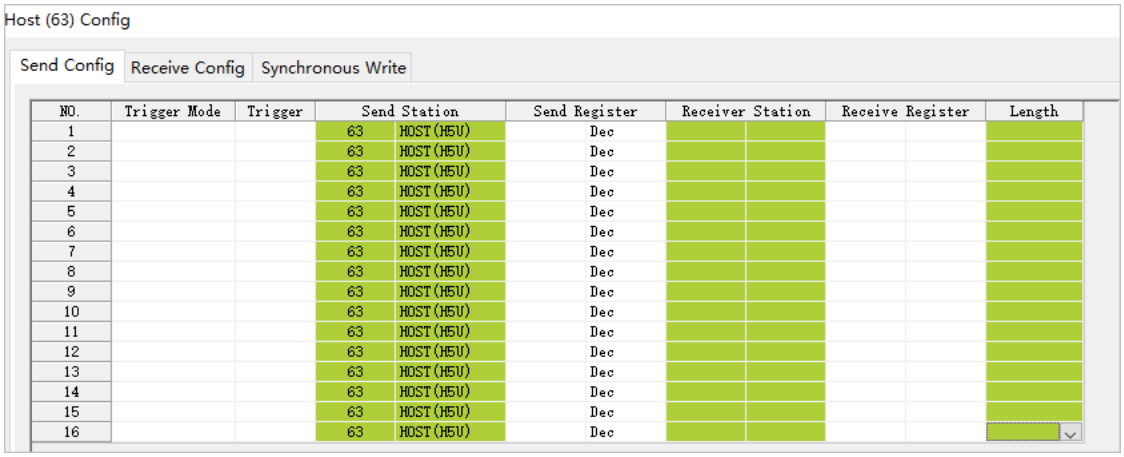
Device type: Filters displayed stations.

Start and stop from station: Select a slave station and control its start/stop of communication.

Site management: Click "Site Management". The initialization wizard page is displayed. You can modify parameters of the master or slave station.

Station configuration: On the main screen, double-click a station. The communication configuration window is displayed. Communication configuration includes sending configuration, receiving configuration, and synchronization configuration (for the master station only).

Sending configuration:



- Trigger Mode
  Time (ms): It is applicable to all devices. The station applies the configuration at a fixed interval. The value ranges from 1 ms to 30000 ms.

  Event (M): It is applicable to the host and PLC. The station applies the configuration when the trigger condition (M element) is set. Multiple configurations can be triggered by the same M element. The element is automatically reset after data is sent. Edge trigger instructions must be used to operate M elements; otherwise, the network load will be excessive.

  Synchronization: It is applicable to all devices. The master station applies the configuration when the system variable _CANlink.SyncTrigger is set. The element is automatically reset after data is sent.

  Event (ms): It is applicable to IS, MD, and remote extension modules (TCM/NTCM). The station applies the configuration when it detects the changed value of the sent register and the trigger condition (disabling time) is met.

  The disabling time indicates the minimum interval for sending the same configuration.

  Maximum number of configuration items for one station: 256 for the host (master station), 16 for one slave station, and 256 for all slave stations.

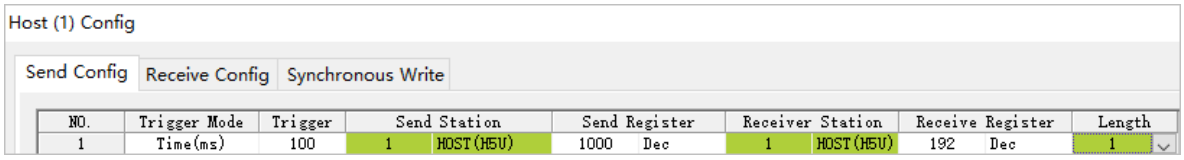
  If you select a configuration item and press "Insert", an empty configuration line will be added following the item. If you select a configuration item and press "Delete", the item will be deleted. In addition, you can press shortcut keys or right-click an item to copy-paste or delete it, and insert or delete a line.

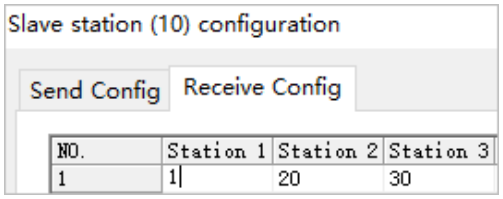
- Register

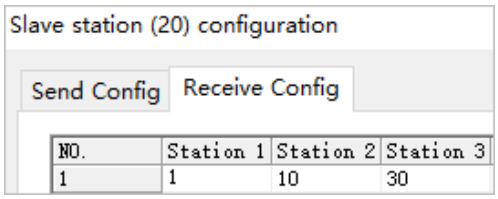
Host and PLC register values correspond to D components. IS and MD register values correspond to parameter numbers. TCM/NTCM corresponds to BFM.

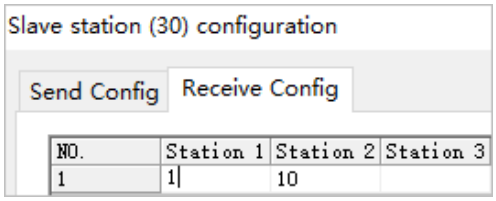
- Reg Count
  It is the number of sent or received consecutive D elements or function codes.
- Point-to-multi-point configuration
  When a sending station is also a receiving station, the station applies the point-to-multi-point configuration, in which no receiving station is specified. If you enter the sending station number into the receiving configuration table, the configured station can receive data sent by the sending station. The received register is the D element or function code corresponding to the receiving station.
- Received data
  The entries in the gray background indicate data received from other stations, including point-to-point and point-to-multi-point data. You can see which element or function code of which station will affect the configured station.
- Receiving configuration
  Receiving configuration applies to receiving point-to-multi-point data from other stations. Each station can receive point-to-multi-point data from eight stations.

  Example:





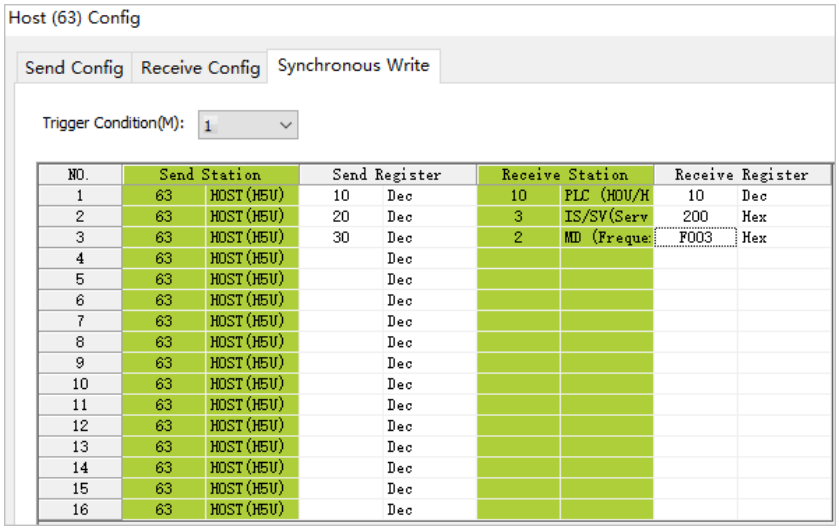




In this example, the master station (station 1) sends the D1000 value in the form of a point-to-multi-point frame to the receiving station D192 at an interval of 100 ms. Based on the receiving configuration of slave stations 10, 20, and 30, slave stations 10, 20, and 30 will receive and write the frame in D192.

> **NOTE**
>
> The point-to-multipoint configuration enables simultaneous application of data. This is equivalent to master synchronous configuration, but does not limit the data sending capability to the master alone. Each station can receive point-to-multipoint data from up to eight different stations, but the number of stations each station can send point-to-multipoint data to is not limited. In other words, all nodes in the network except for the sending station can receive such point-to-multipoint data. However, to receive point-to-multipoint data from a station, the receiver must be configured to allow receiving such data from the station.

Synchronous write configuration for the master station



NO.	Send Station		Send Register		Receive Station		Receive Register	
1	63	HOST(H5U)	10	Dec	10	PLC (HDU/H	10	Dec
2	63	HOST(H5U)	20	Dec	3	IS/SV(Serv	200	Hex
3	63	HOST(H5U)	30	Dec	2	MD (Freque	F003	Hex
4	63	HOST(H5U)		Dec				
5	63	HOST(H5U)		Dec				
6	63	HOST(H5U)		Dec				
7	63	HOST(H5U)		Dec				
8	63	HOST(H5U)		Dec				
9	63	HOST(H5U)		Dec				
10	63	HOST(H5U)		Dec				
11	63	HOST(H5U)		Dec				
12	63	HOST(H5U)		Dec				
13	63	HOST(H5U)		Dec				
14	63	HOST(H5U)		Dec				
15	63	HOST(H5U)		Dec				
16	63	HOST(H5U)		Dec				

When the trigger condition (M) is set, the Synchronous Send configuration for the master station takes effect. You can select different trigger conditions (M) to display, add, modify, or delete synchronization configurations. Synchronization configuration is applicable to scenarios in which an operation needs to be initiated synchronously.

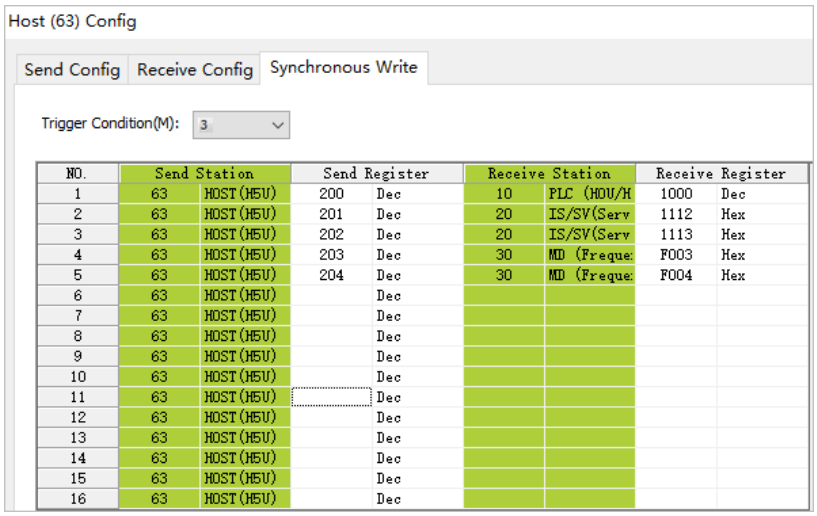
As shown in the figure, when M1 is 1, the master station sends the three configuration items successively. Upon receipt of the items, slave stations store them in the buffer. After the last data entry is sent, the master station automatically sends a configuration application command. Upon receipt of the command, all slave stations automatically write the data in the buffer in corresponding elements or function codes. As shown in the figure, PLC 10 writes the D10 value in D10, servo 20 writes the D20 value in H200, and AC drive 30 writes the D30 value in HF003. All these values are synchronously written when slave stations receive the configuration application command. After the command is sent, the master station automatically resets the trigger element M1. Edge trigger instructions must be used to operate M elements; otherwise, the network load will be excessive.

Precautions for the trigger condition (M):

- Each trigger condition associates a maximum of 16 configuration items. It determines whether the associated synchronization configuration is valid. A maximum of 8 trigger conditions (M) are allowed.
- You can select a trigger condition from the drop-down list.
- During synchronization configuration of a 32-bit servo register, data must correspond to high-order 16 bits and low-order 16 bits respectively for the same trigger element. That is, two data entries must be written for one trigger element, one corresponding to high address bits of the 32-bit function code, and the other corresponding to low address bits. If only one entry is written or two entries are written for two trigger elements respectively, the servo will return an error, and the configuration cannot continue.

Example of 32-bit servo register synchronization configuration:

As shown in the following figure, H1112 is a 32-bit function code of the servo. During configuration of the function code, two data entries must be written, corresponding to high and low address bits respectively. When M3 is set, the master station writes D201 and D202 values in H1112. When all of the five data entries of M3 are sent, the master station sends a command to enable the slave stations and apply the configurations. Then M3 is automatically reset.



If only one address is processed for one trigger element, the servo will return an error so that synchronization cannot continue. The error will be recorded in D8307 of the master station. Fault codes are listed in Fault Codes.

- Master station fault codes and processing
  The following table lists configuration errors and causes. You can use the system variable _CANlink.ConfigErr to view the details.

Error Code	Cause	Solution
XX00	Reserved	None
XX01	Incorrect code	Check whether the internal definition is correct.
XX02	Incorrect index	Check whether the device type is correct.
XX03	Incorrect information	Check whether the address is valid and check the read-write property.
XX04	Reserved	Reserved
XX05	Incorrect data length	Check whether the data length exceeds the limit.
XX06	Configuration frames failing to respond within a specified time	Check whether the connection is normal.

  The following table lists abnormality codes and causes. You can use the system variable _CANlink.SyncWrErr to view the details.

Error Code	Cause	Solution
XX00	Reserved	Reserved

Error Code	Cause	Solution
XX01	Invalid command code	Check whether the internal definition is correct.
XX02	Abnormal address	Check whether the address is normal or whether the address can be accessed.
XX03	Abnormal data	Check whether the data is within a specified range.
XX04	Invalid operation	Check whether the operation is authorized.
XX05	Invalid length	Check whether the data length exceeds the limit.
XX06	Responding timeout	Check whether the connection is normal.

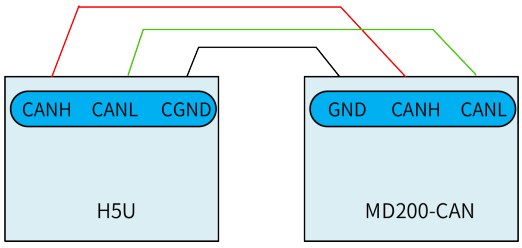
NOTE

- The fault codes are in decimal, where XX indicates the station number. Specifically, a fault code indicates that an error occurred when configuring XX station or sending commands to XX station.
- Fault codes of PLC slaves are similar to those of the master, except that the fault codes of PLC slaves do not contain a station number.

10.4.5 CANlink Application Example

Use one H5U and one MD200-CAN AC drive to control the start/stop and write the frequency of the AC drive through the CANlink bus.

1. The following figure describes the CANlink bus connection.



2. Configure function codes for the slave MD200 AC drive, as shown in the following table.

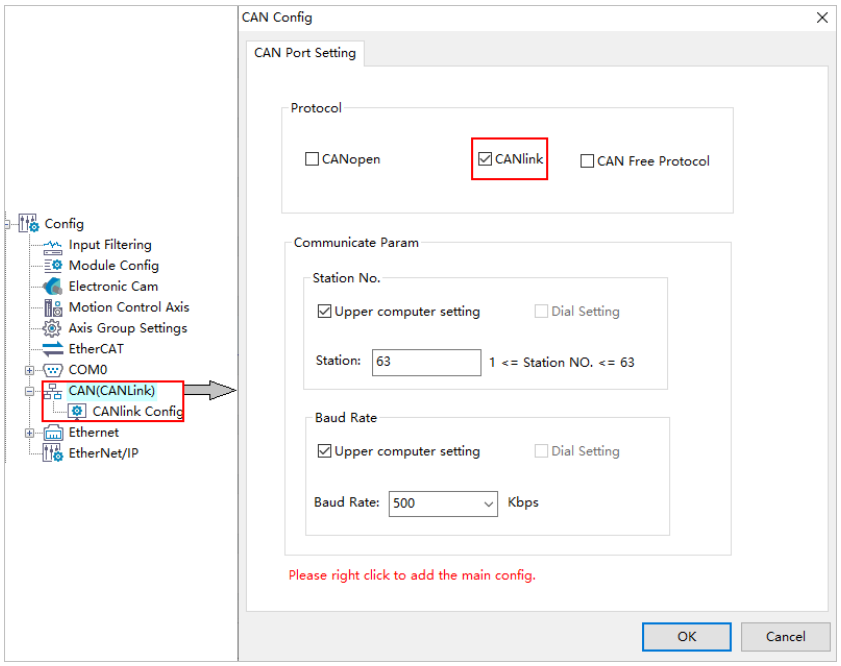
Function Code	Name	Value	Description
FD-00	Baud rate	5005	CANlink baud rate: 500 kbps
FD-02	Local address	1	Local station No. is 1.
F0-02	Command source selection	2	Communication set value
F0-03	Main frequency reference source	**9**	Communication set value

Control parameter addresses when MD200-CAN is used for communication with the host controller PLC:

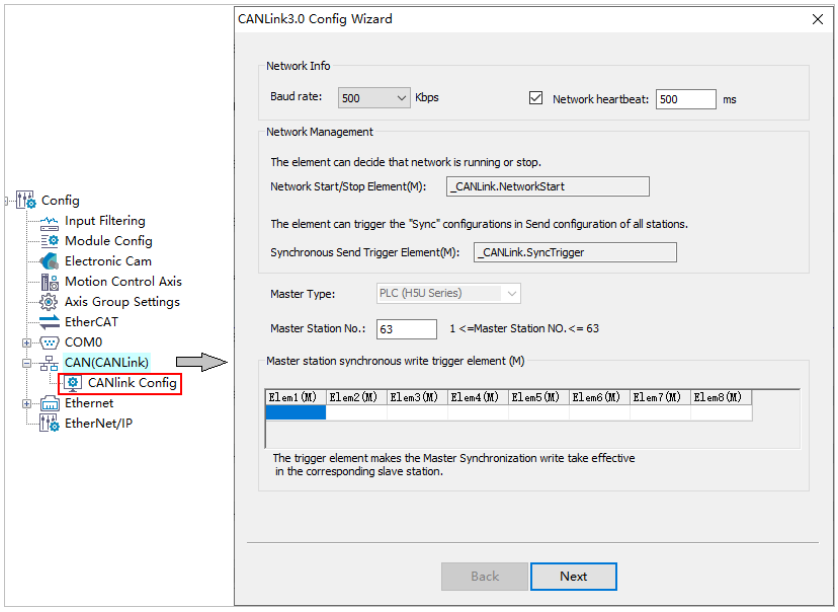
Para. Address	Name	Description
H1000	Frequency set through communication	-10000 to +10000 (decimal)
H1001	Feedback running frequency	-
H2000	Control commands	0001: Forward running 0002: Reverse running 0003: Forward jog 0004: Reverse jog 0005: Coast to stop 0006: Decelerate to stop 0007: Fault reset

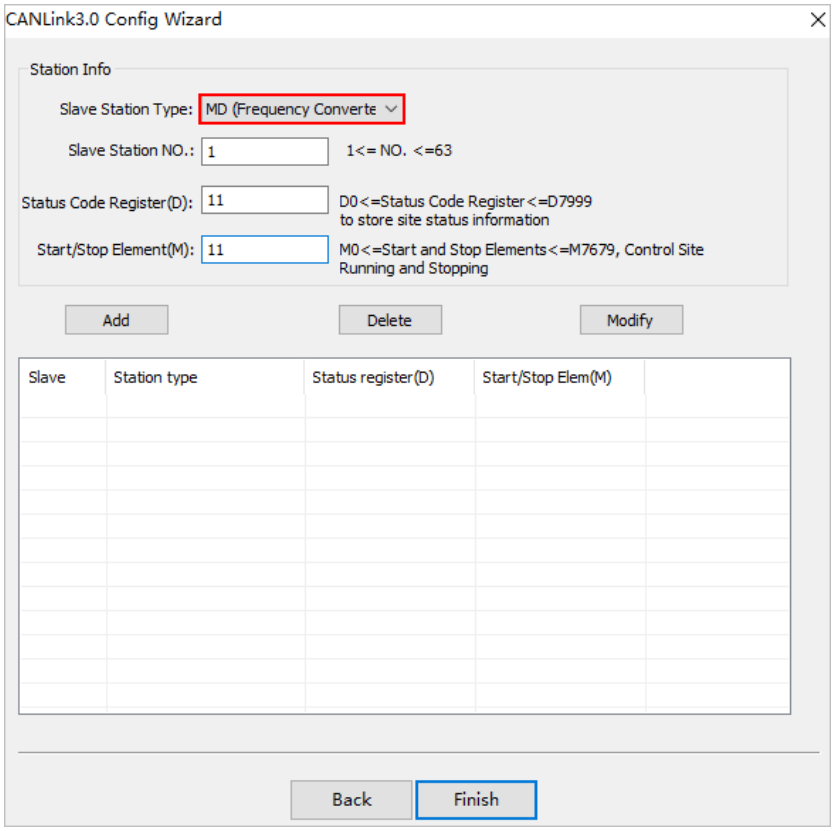
3. Configure the H5U master station

- In AutoShop, right-click CANlink, set the station No. of the H5U master station to 63, and set the baud rate to 500 kbps.



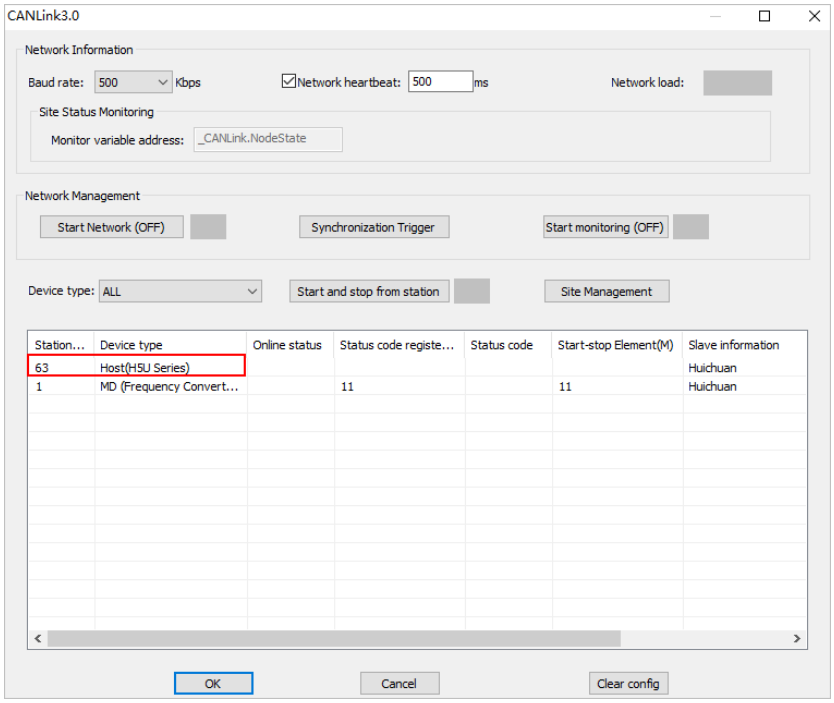
- Add an MD AC drive, as shown in the following figure.

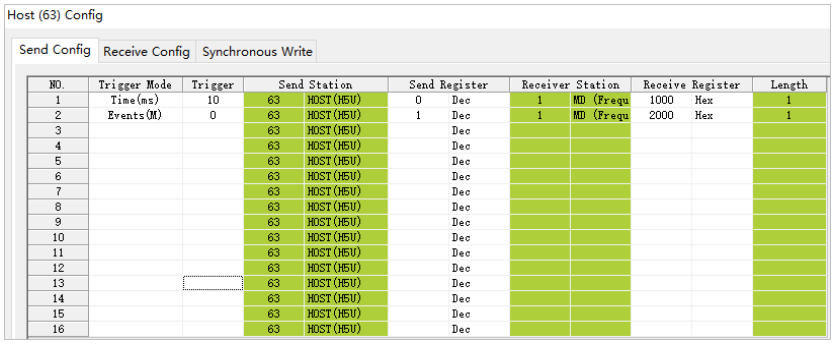




Set the slave station No. according to that defined by FD-02 of the AC drive.

- Configure the master station to send data, as shown in the following figure.



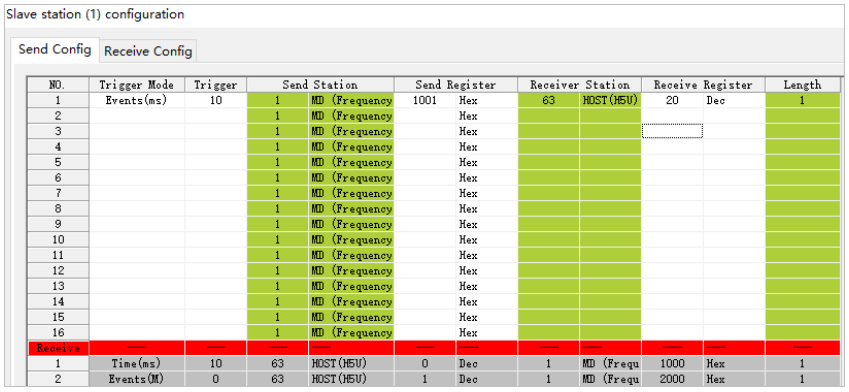


Then:

The H5U master station writes the value of the D0 element to the 1000 (frequency reference) address register of the slave station 1# at an interval of 10 ms.

When the M0 element is ON, the H5U master station writes the value of the D1 element to the 2000 (AC drive control word) address register of the slave station 1#.

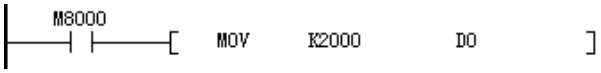
- Configure the slave station to send data, as shown in the following figure.



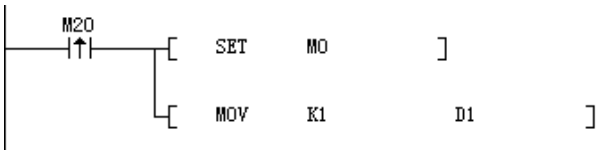
The slave AC drive 1# converts the value in the 1001 (running frequency) address register and then writes the value to the D20 element of the master station 63# at an interval of 10 ms.

- PLC Programming

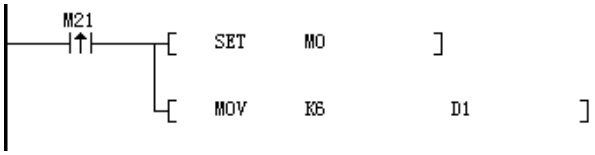
  Set the AC drive running frequency to 20 Hz, as shown in the following figure.



Set M20 to ON to start the AC drive and make it rotate in the forward direction, as shown in the following figure.

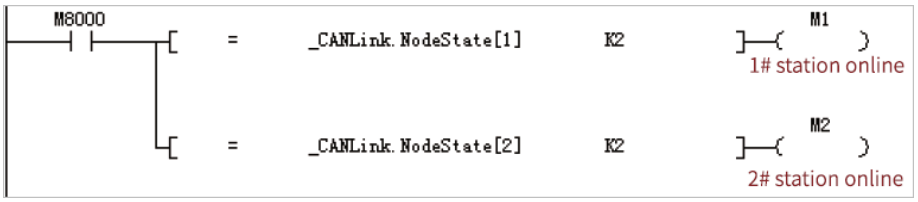


Set M21 to OFF to stop the AC drive, as shown in the following figure.



Determination on the online status of slave stations

The online state of the slave station will be updated to the system variable _CANLink.NodeState[64], in which _CANLink.NodeState[1] is the state of the station 1, as shown in the following figure.



10.4.6 Troubleshooting for CANlink Communication

Check the following items when a CANlink communication error occurs.

- Check the termination resistor
  Power off all devices. Use a multimeter to measure the resistance between CANH and CANL. The resistance should be about 60 Ω. If the resistance is too small, there are termination resistors incorrectly connected at other locations. In this case, disconnect these termination resistors. If only one termination resistor is available, the resistance is about 120 Ω, and the network connection is bad. If no termination resistor is available, communication fails. Provide termination resistors between the stations at both ends of the network.
- Check the baud rate
  Check whether the baud rate is normal. Baud rates of all stations in the network must be the same; otherwise, communication fails. Power off and then on the device or switch it from STOP to RUN so that the baud rate can take effect.
- Others
  In case of strong interference, reduce the baud rate.

10.5 CANopen Communication

10.5.1 CANopen Communication Protocol

The H5U supports the CANopen communication standard protocol DS301.

Software Function Module	Slave	Master
Supported protocol	DS301 V4.02	DS301 V4.02

Software Function Module	Slave	Master
Maximum number of TPDOs	8	64
Maximum number of RPDOs	8	64
Number of slave station nodes	/	30
Baud rate and communication distance	1 Mbps/25 m 800 kbps/50 m 500 kbps/100 m 250 kbps/250 m 125 kbps/500 m 50 kbps/1000 m 20 kbps/2500 m 100 kbps 10 kbps	1 Mbps/25 m 800 kbps/50 m 500 kbps/100 m 250 kbps/250 m 125 kbps/500 m 50 kbps/1000 m 20 kbps/2500 m 100 kbps 10 kbps
Soft element for data exchange	W300 to W363	D0 to D7999 (configurable)

10.5.2 List of CANopen Axis Control Instructions

The following table lists CANopen axis control instructions supported by H5U. See H5U Series Programmable Logic Controller Instructions Guide for detailed usage of related instructions.

Instruction	Function
MC_Power_CO	Instruction for enabling the communication control servo axis
MC_Reset_CO	Instruction for resetting faults of the communication control servo axis
MC_ReadActualPosition_CO	Instruction for reading the current position of the communication control axis
MC_ReadActualVelocity_CO	Instruction for reading the current velocity of the communication control axis
MC_Halt_CO	Instruction for halting the motion of the communication control servo axis
MC_Stop_CO	Instruction for stopping the communication control servo axis
MC_MoveAbsolute_CO	Instruction for obtaining the absolute position of the communication control axis
MC_MoveRelative_CO	Instruction for obtaining the relative position of the communication control axis
MC_MoveVelocity_CO	

Instruction	Function
	Instruction for selecting the velocity operation mode of the communication control axis
MC_Jog_CO	Instruction for communication control axis jogging
MC_Home_CO	Instruction for communication control axis homing
MC_WriteParameter_CO	Instruction for writing parameters of the communication control axis
MC_ReadParameter_CO	Instruction for reading parameters of the communication control axis

10.5.3 Related Terms

- NMT: Network Management
  Network management includes management of application layers, network states, and node ID allocation. It is implemented in master-slave communication mode. That is, on a CAN network, only one NMT master station exists with one or more slave stations. The service is used to control the slave station state.
- SDO: Server Data Object
  An SDO can access the data in the slave station object dictionary through indexes and sub-indexes. SDOs are used for slave station configuration. Each frame of an SDO request must be answered.
- PDO: Process Data Object
  PDOs are used to transmit real-time data. The data length ranges from one to eight bytes. Data can be transmitted in synchronous and asynchronous modes. PDO frames are primary data exchange frames after slave stations are started.
- SYNC: Synchronous
  Synchronization is implemented in master-slave communication mode. The master SYNC node regularly sends SYNC objects, and the SYNC slave node synchronously executes tasks upon receipt of the objects. SYNC frames are used for synchronous transmission through PDOs.
- COB-ID: Communication Object Identifier
  Each CANopen frame starts with a COB-ID, which is the communication target identifier of the CAN frame. A COB-ID is not the slave station number. However, it is associated with the slave station number by default.

10.5.4 CANopen Indicators

You can judge the CANopen communication state based on the CANopen indicator.

LED Indicator	CAN RUN (Green)	CAN ERR (Red)
OFF	None	No error
ON	Operational	Bus disconnected
Flashing slowly (0.8-second cycle)	Pre-operational	Pre-operational

LED Indicator	CAN RUN (Green)	CAN ERR (Red)
Flashing once quickly (1.2-second cycle)	Stopped	At least one error counter of the CAN controller hitting or exceeding the threshold (too many error frames)
Flashing twice quickly (1.6-second cycle)	None	Incorrect control event (node protection or heartbeat timeout)

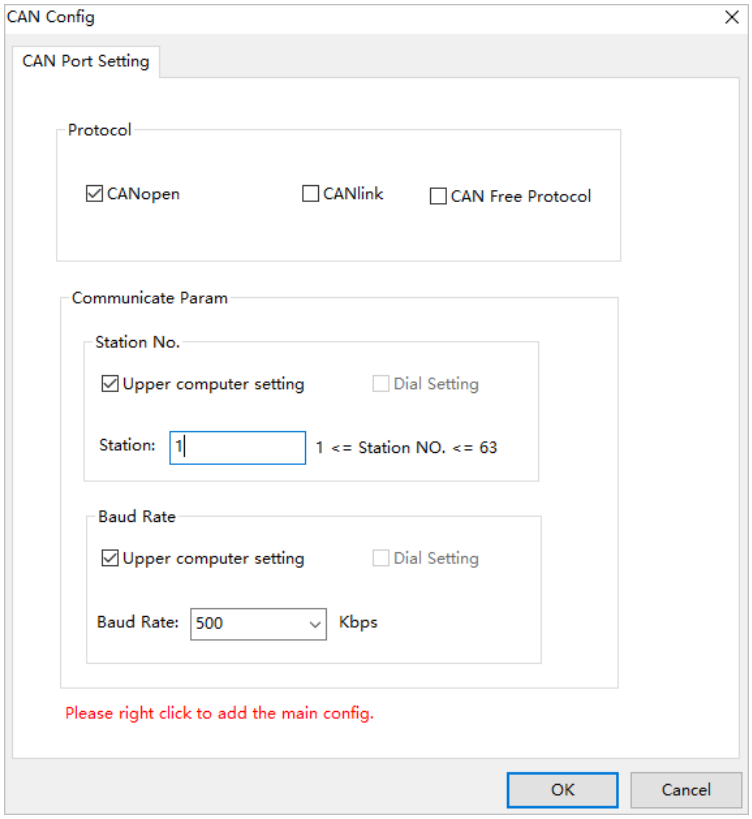
10.5.5 CANopen Configuration

10.5.5.1 Configuring a CANopen Master

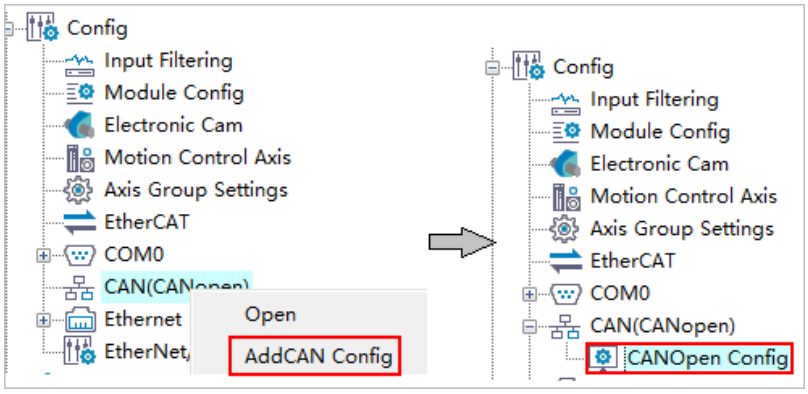
Procedure

When "CANopen" is selected, the system automatically determines whether the current PLC is the CANopen master station or the CANopen slave station based on the presence of the CANopen configuration.

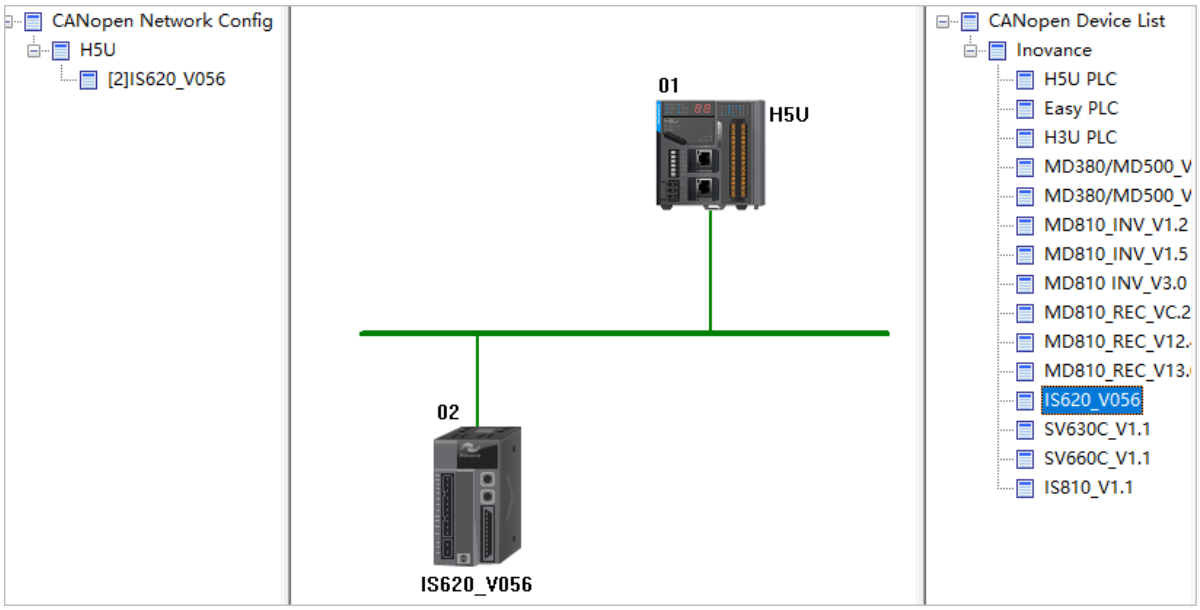
1. After creating a project, choose "Project Manager" > "Config", and then double-click "CAN". The following window is displayed.



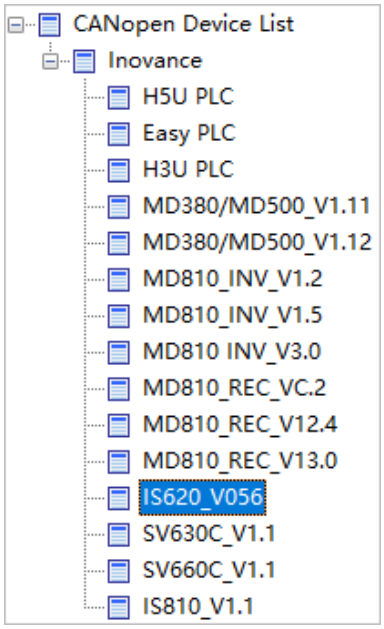
2. Select "CANopen", set the station number and baud rate, and click "OK".
   In this example, CAN is configured as the CANopen slave station. Choose "Project Manager" > "Config". Right-click "CAN". In the window displayed, select "Add CAN Config" to configure it as the CANopen master station, as shown in the following figure.



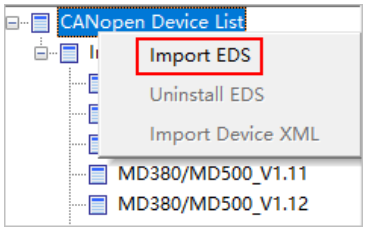
3. Double-click "CANOpen Config". The following CANopen configuration page is displayed:



4. Double-click or drag the CANopen slave station you want to add in the device list.

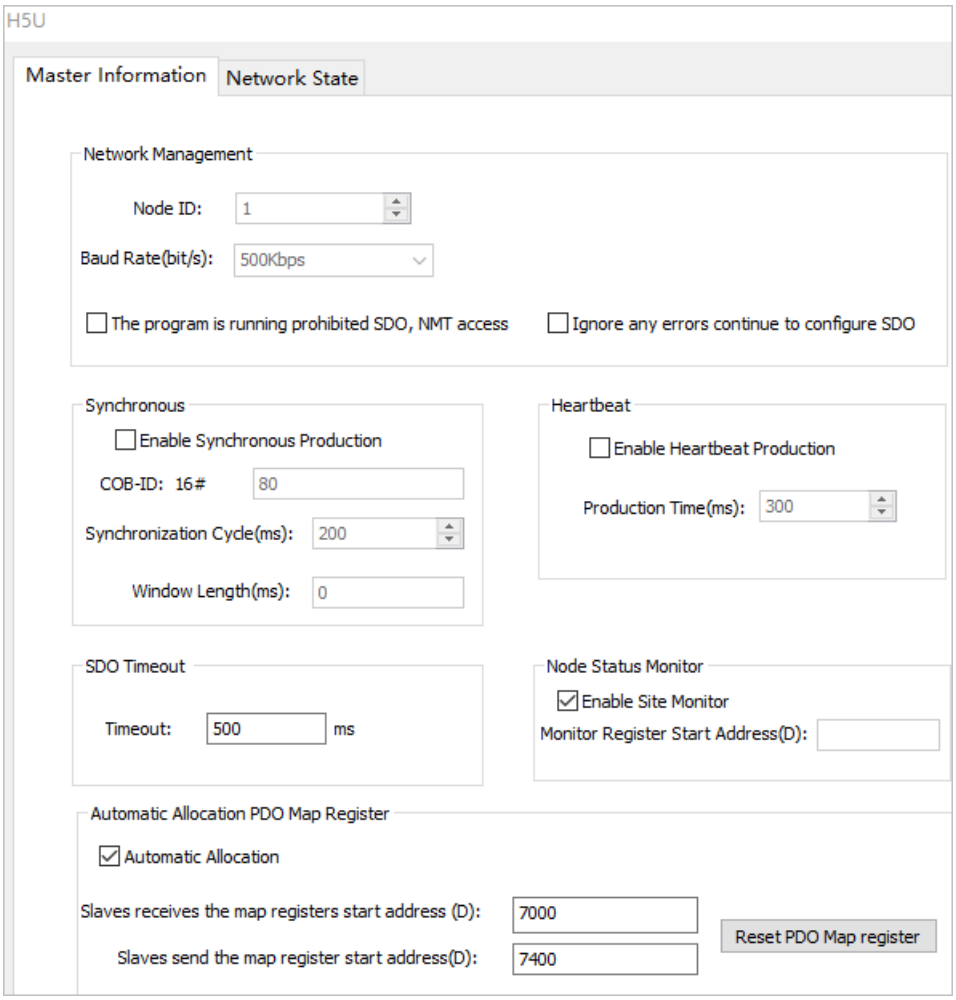


5. If the target slave station is not in the list, right-click the CANopen device list, and click "Import EDS" to import the EDS file, which can be obtained from the device supplier.



Master information page

Set the master parameters. Double-click the H5U master station in the network. The following window is displayed.



- Network Management
  Node ID: Indicates the master station number. If the station number is identical to the PLC number, the PLC will be initialized as the CANopen master station. If the station number is different from the PLC number, the PLC will be initialized as a CANopen slave station.

  Baud Rate: Indicates the communication baud rate valid for the master station.

  The program is running prohibited SDO, NMT access: If this option is selected, online commissioning is disabled during running of the program. The function only applies to background software.

  Ignore any errors continue to configure SDO: After this option is selected, if SDO configuration errors occur, configuration will continue. The function is valid for all slave stations. If the option is not selected, when SDO errors occur, the master station will reset slave stations through broadcasts.

- Synchronous
  Enable Synchronous Production: If this option is selected, the configured station will send a sync frame repeatedly in the set synchronization cycle.

COB-ID: Indicates the ID for sync frame sending. The default value is 0x80. The parameter cannot be configured.

Synchronization Cycle (ms): Indicates the cycle for sync frame sending. The default value is 200, in the unit of ms.

Window Length (ms): The value is 0 by default. The parameter cannot be configured.

> **NOTE**
>
> Only one synchronous frame transmission can exist in one network.

- Heartbeat
  Enable Heartbeat Production: If this option is selected, the configured station will send heartbeat frames repeatedly in the set cycle.

  Production Time (ms): Indicates the cycle for heartbeat sending. The default value is 300, in the unit of ms.

  > **NOTE**
  >
  > The default heartbeat monitoring consumption time of the master is 2.5 times the heartbeat production time. (The timeout threshold for heartbeat monitoring is 2.5 times the heartbeat production time.)

- SDO Timeout
  Timeout: Indicates the SDO waiting time. The default value is 500, in the unit of ms. SDO frames are used for network configuration. If the SDO fails to receive return frames after the third try, the master station determines that configuration times out. The waiting time for each frame is called SDO timeout.

- Node Status Monitor
  The online status of the station will be updated to the system variable _CANOpen.NodeState[64], in which _CANOpen.NodeState[0] is the status of the local station, while _CANOpen.NodeState[station No.] is the status of the slave station.

Value	Status
0	Initializing
4	Stopped
5	Operational
127	Pre-operational
255	Offline

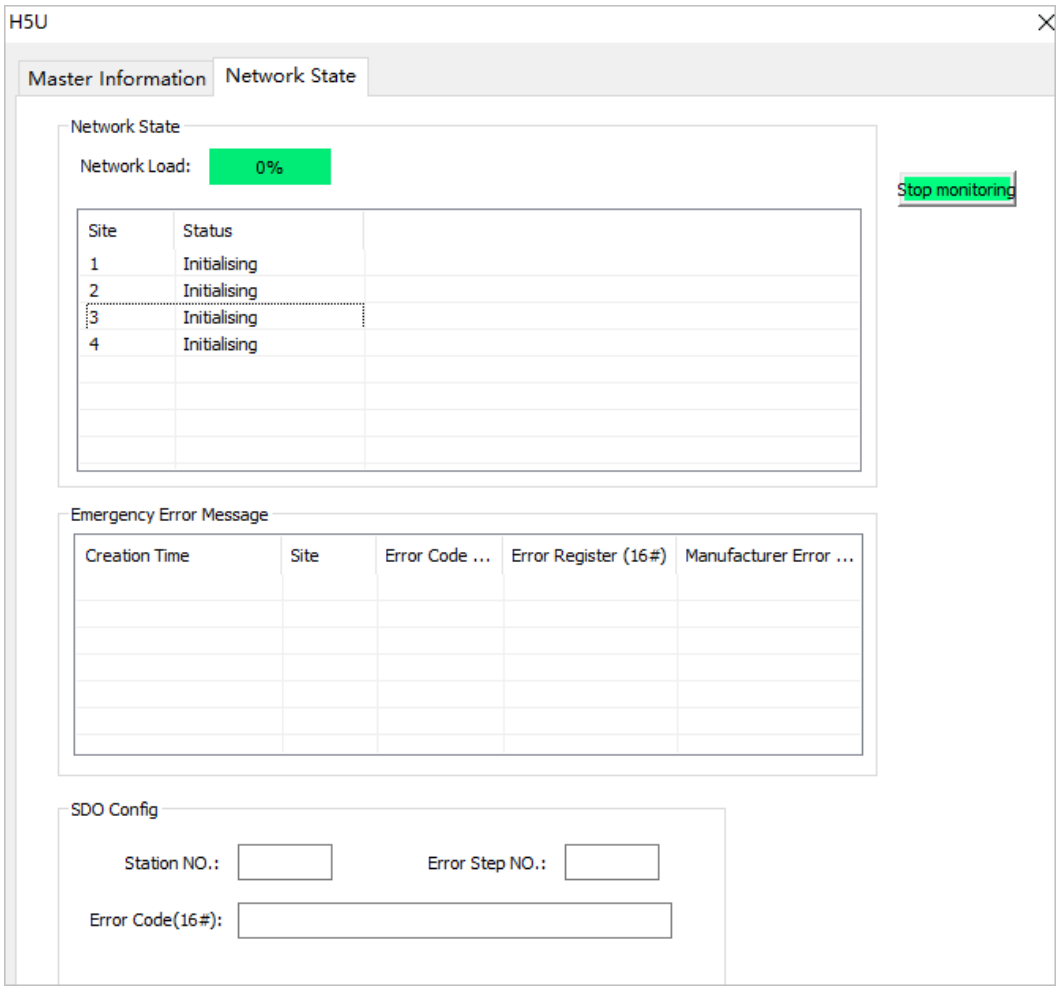
> **NOTE**
>
> If the corresponding slave does not exist, the corresponding register will not be updated. For example, if station 3 does not exist, the data of _CANOpen.NodeState[3] will not be updated.
>
> This function works only when the heartbeat or node protection function is set on the slave, because the relevant status is fed back by the heartbeat or node protection frame of the slave.

- Automatic Allocation PDO Map Register
  Automatic Allocation: If this option is selected, the system will automatically assign the address of the register for master-slave data exchange. If this option is not selected, you need to configure the start address for data exchange (by configuring the start address of each PDO). This option is selected by default.

  Slaves receive the map registers start address: Indicates the automatically assigned start address of data sent by the master station ("Automatic Allocation" must be selected).

  Slaves send the map register start address: Indicates the automatically assigned start address of data received by the master station ("Automatic Allocation" must be selected).

Network State



Start monitoring/Stop monitoring: Information monitoring is enabled by clicking this option. Monitoring is disabled by clicking the option again.

Network Load: Monitors the network load in real time.

Network state table: Displays the station state. The table is applicable only to the master station. The state value is from the node state monitoring register.

- Emergency Error Message
  The table lists emergency error messages on the network. It is applicable only to the master station. The master PLC only caches the latest error message. If background programs are not shut down, a maximum of five messages will be cached in the background.
- SDO Config
  Station NO.: Indicates the number of the station with SDO configuration errors.

Error Step NO.: Indicates the SDO error number. To check numbers of slave stations with parameter errors, click the SDO tab.

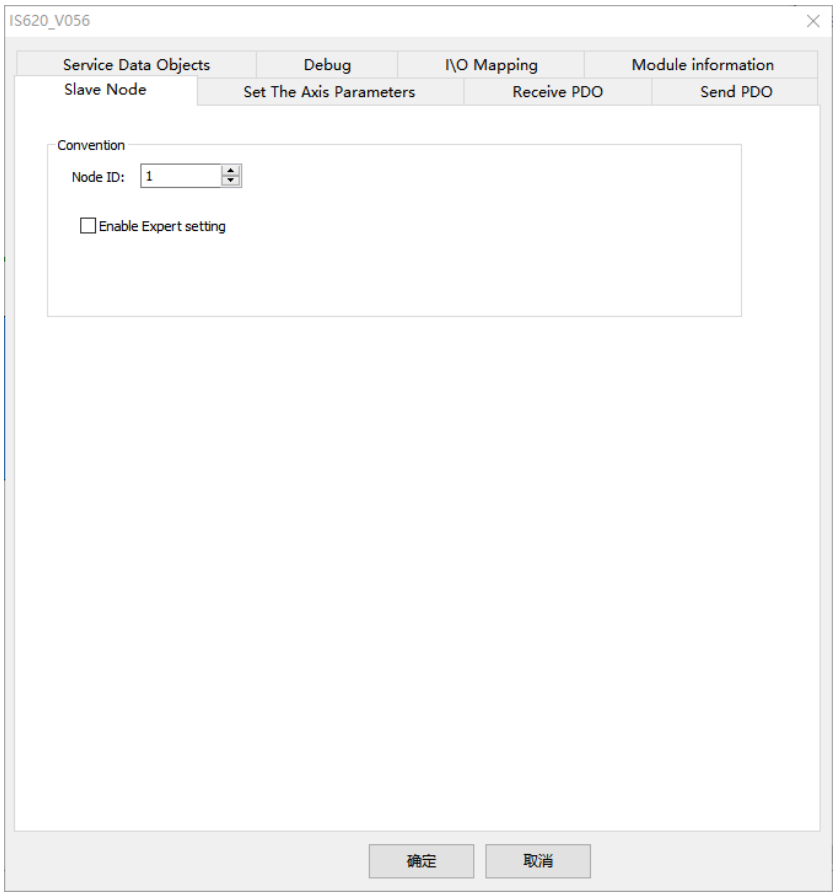
Error Code: Indicates the SDO fault code (standard CANopen fault code).

10.5.5.2 Configuring a CANopen Slave

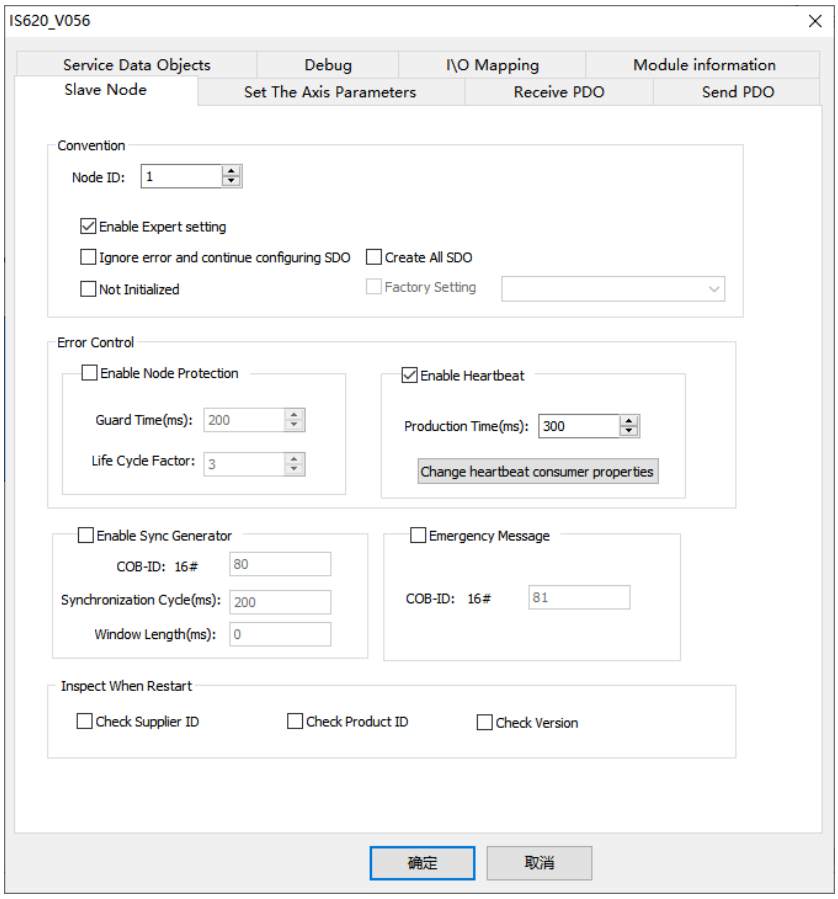
This section takes the IS620 slave station as an example to describe how to configure the CANopen slave and its parameters.

Configuring a slave node

1. Double-click a slave in the network. The following dialog box is displayed.



2. Select "Enable Expert setting" and set parameters on the page. (By default, this option is not selected.)



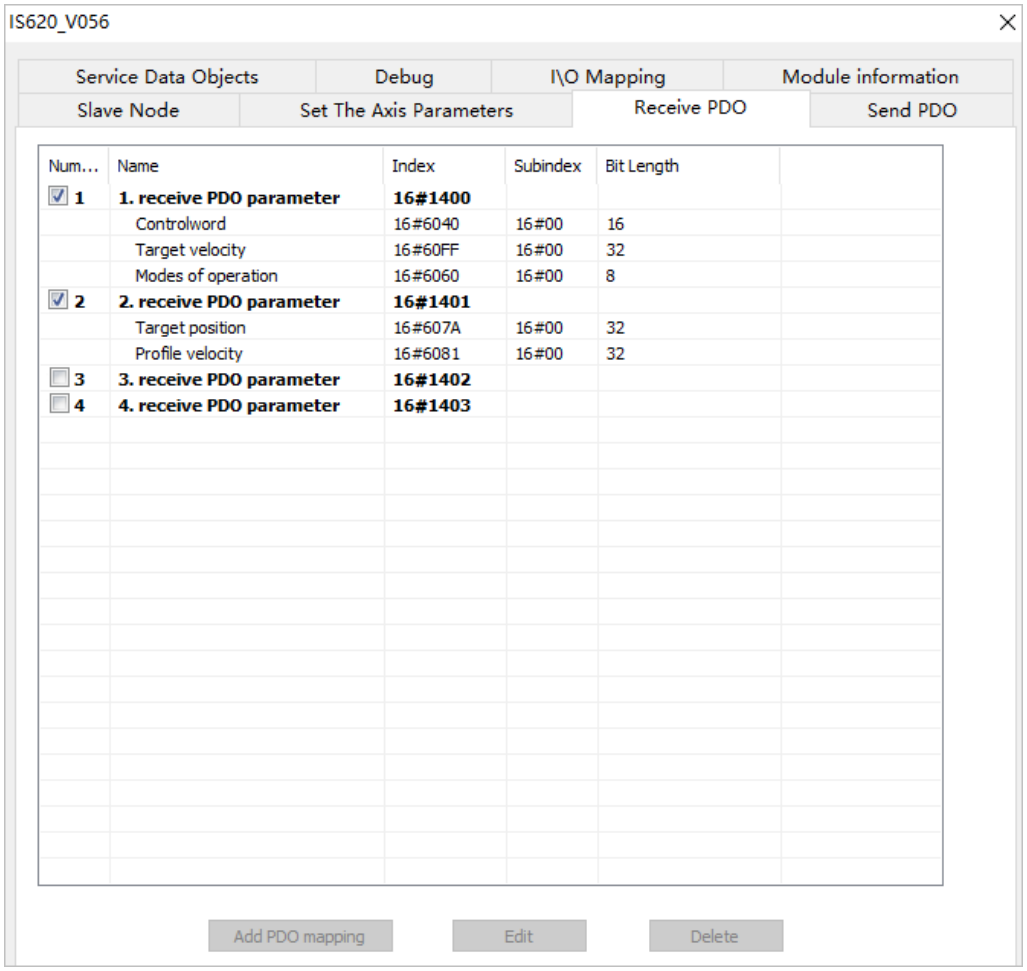
Parameter Name		Description
Convention	Node ID	Indicates the ID of a slave node.
	Enable Expert setting	When this option is selected, detailed configurations of the slave are displayed. By default, this option is not selected.
Ignore error and continue configuring SDO		• Selected: When a configuration error (other than a check type error) occurs, configuration continues. • Deselected: When a configuration error occurs, configuration cannot continue, and the entire network is disconnected. By default, this option is not selected.
Create All SDO		If this option is selected, all writable object dictionaries in the EDS will be added and initialized. By default, this option is not selected.
Not Initialized		If this option is selected, the slave station will not be initialized (this option can be selected only when the station applies the default configuration). By default, this option is not selected.
Factory Setting		If this option is selected, you can select options from the drop-down list.

Parameter Name		Description
		By default, the option is not selected. This option can be selected only when the slave supports this function.
Error Control	Enable Node Protection	Enable Node Protection: If this option is selected, node protection will be enabled. By default, the option is not selected. Node protection timeout = Guard time x Life cycle factor Node protection provides a network evaluation platform on which master station and slave station monitor each other with return frames. Either the heartbeat or node protection function can be selected. Guard Time (ms): Indicates the node protection time, which is 200 ms by default. Life Cycle Factor: Indicates the node protection factor, which is 3 by default.
	Enable Heartbeat	Enable Heartbeat: If this option is selected, heartbeats will be generated. By default, this option is selected. When this option is selected, the master station will monitor the heartbeat state by default. Production Time (ms): Indicates the cycle for heartbeat sending. Change heartbeat consumer properties: It is used to set heartbeats of other stations to be monitored by the configured station. This function is disabled by default. The function can be enabled only when the slave station supports heartbeat monitoring.
	Enable Sync Generator	Enable Sync Generator: If this option is selected, the configured station will send a sync frame repeatedly in the set synchronization cycle. COB-ID: Indicates the ID for sync frame sending. The default value is 0x80. The parameter cannot be configured. Synchronization Cycle (ms): Indicates the cycle for sync frame sending. The default value is 200, in the unit of ms. Window Length (ms): The value is 0 by default. The parameter cannot be configured. Note: Only one synchronous frame transmission can exist in one network.
	Emergency Message	Emergency Message: If this option is selected, you can set the COB-ID of an emergency message. By default, this option is not selected.
	Inspect When Restart	If "Check Supplier ID", "Check Product ID", or "Check Version" is selected, corresponding data will be checked before configuration of the slave station. If the check fails, the network cannot be connected.

Receive PDO/Send PDO

Receive PDO indicates the data sent by the master to a slave. Send PDO indicates the data sent by a slave to the master.

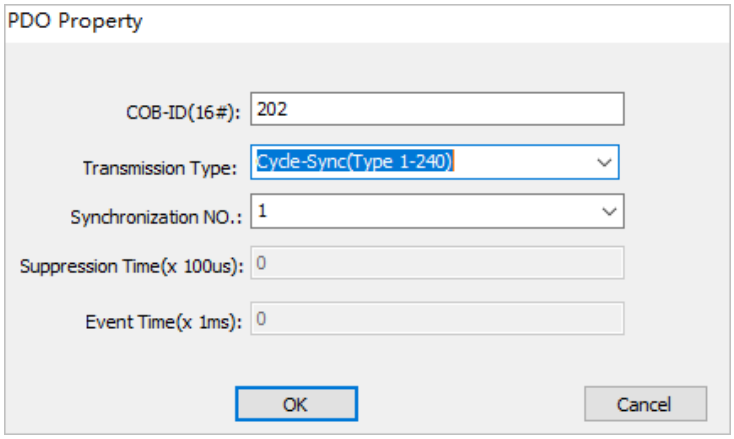
1. Click "Receive PDO" or "Send PDO" to access the corresponding page.



2. Enable PDO.
   Check the box to enable PDO. The PDOs in the EDS file that take effect are checked by default.

3. Add/edit the PDO mapping.
   Click "Add PDO Mapping" or "Edit" to add or edit the PDO mapping.

> **NOTE**
>
> To delete a PDO, select it and then click "Delete".



- COB-ID
  Indicates the ID for sending a PDO parameter. Based on the CANopen DS301 protocol, default COB-IDs are available for the first four PDO parameters. For other PDO parameters, you need to configure their COB-IDs (if supported by the slave). The COB-IDs must be unique within the entire network, ranging from 0x180 to 0x57F.
- Transmission Type

Type	Condition for Data Sending	Condition for Valid Data
Loop-synchronization (Type 0)	Data is changed, and a sync frame is received.	Data does not take effect immediately but takes effect after a sync frame is received.
Loop-synchronization (Types 1 to 240)	Data is sent after the corresponding "number of synchronizations" frame is received.	Data does not take effect immediately but takes effect after a sync frame is received.
Asynchronization-only RTR (Type 252)	Not supported	Not supported
Asynchronization-only RTR (Type 253)	Not supported	Not supported
Asynchronization-specified by manufacturers (Type 254)	Manufacturer-defined	Manufacturer-defined
Asynchronization-specified by the configuration file (Type 255)	Data is changed or the event time is correct, and the change cycle is shorter than the suppression time.	Immediately

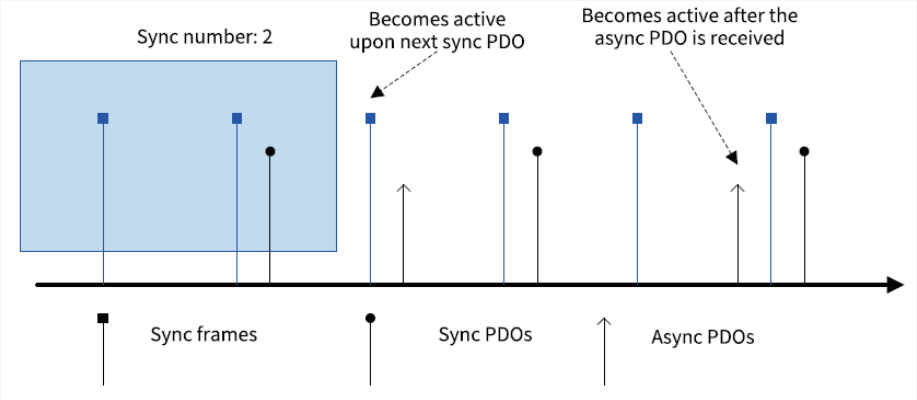
NOTE

To use the synchronous type, it is necessary to enable synchronous production on a station, usually the master.

- Synchronization NO.
  The number of synchronizations takes effect after "loop-synchronization (types 1 to 240)" is selected.

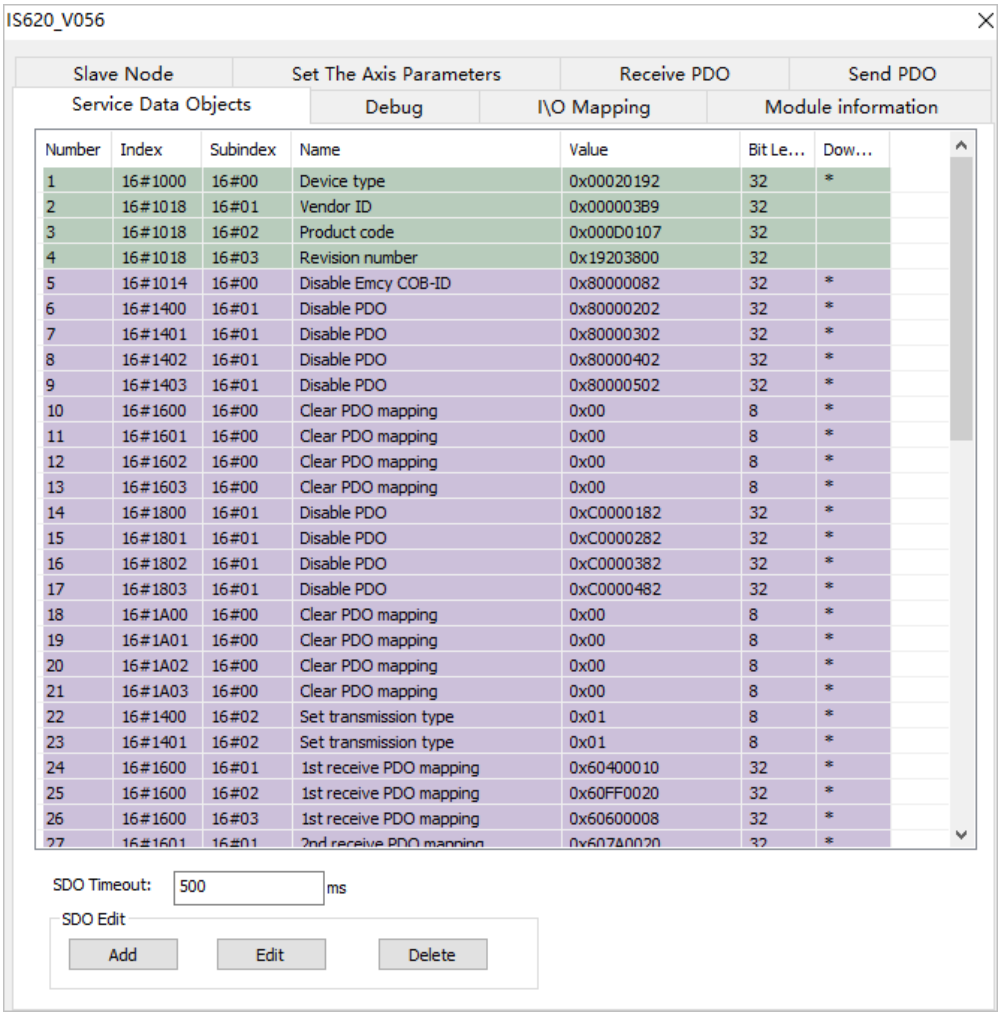
- Suppression Time
  The suppression time can be set after "asynchronization-specified by the configuration file (Type 255)" is selected. If the value is 0, the function is disabled. If the value is not 0, the suppression time is the minimum interval for frame sending.
- Event Time
  The event time can be set after "asynchronization-specified by the configuration file (Type 255)" is selected. If the value is 0, the function is disabled. If the value is not 0, the event time is the cycle for data sending. (Data sending is limited by the suppression time.)

The following figure shows the example of loop-synchronization (Type 2).



Service Data Object (SDO)

1. Click "Service Data Objects" to access the "Service Data Objects" page.
   This page lists SDO configurations automatically generated based on user settings.



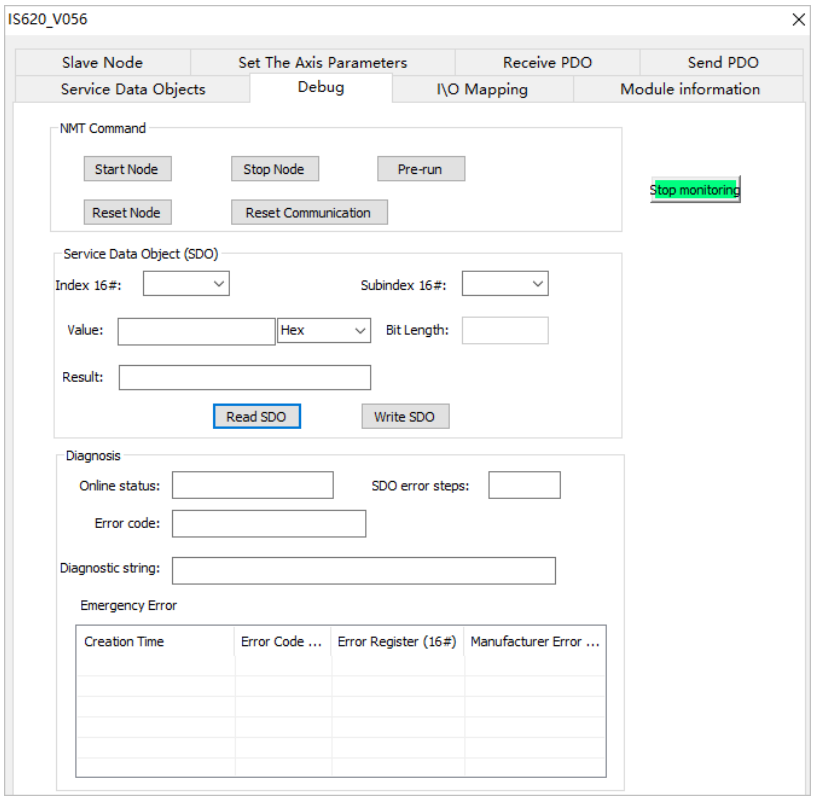
Number	Index	Subindex	Name	Value	Bit Le...	Dow...
1	16#1000	16#00	Device type	0x00020192	32	*
2	16#1018	16#01	Vendor ID	0x000003B9	32	
3	16#1018	16#02	Product code	0x000D0107	32	
4	16#1018	16#03	Revision number	0x19203800	32	
5	16#1014	16#00	Disable Emcy COB-ID	0x80000082	32	*
6	16#1400	16#01	Disable PDO	0x80000202	32	*
7	16#1401	16#01	Disable PDO	0x80000302	32	*
8	16#1402	16#01	Disable PDO	0x80000402	32	*
9	16#1403	16#01	Disable PDO	0x80000502	32	*
10	16#1600	16#00	Clear PDO mapping	0x00	8	*
11	16#1601	16#00	Clear PDO mapping	0x00	8	*
12	16#1602	16#00	Clear PDO mapping	0x00	8	*
13	16#1603	16#00	Clear PDO mapping	0x00	8	*
14	16#1800	16#01	Disable PDO	0xC0000182	32	*
15	16#1801	16#01	Disable PDO	0xC0000282	32	*
16	16#1802	16#01	Disable PDO	0xC0000382	32	*
17	16#1803	16#01	Disable PDO	0xC0000482	32	*
18	16#1A00	16#00	Clear PDO mapping	0x00	8	*
19	16#1A01	16#00	Clear PDO mapping	0x00	8	*
20	16#1A02	16#00	Clear PDO mapping	0x00	8	*
21	16#1A03	16#00	Clear PDO mapping	0x00	8	*
22	16#1400	16#02	Set transmission type	0x01	8	*
23	16#1401	16#02	Set transmission type	0x01	8	*
24	16#1600	16#01	1st receive PDO mapping	0x60400010	32	*
25	16#1600	16#02	1st receive PDO mapping	0x60FF0020	32	*
26	16#1600	16#03	1st receive PDO mapping	0x60600008	32	*
27	16#1601	16#01	2nd receive PDO mapping	0x607A0020	32	*

2. Click "Add" to add configurations. It is used to assign initial values to object dictionaries of a slave station.

> **NOTE**
>
> - Edit: Edits configurations.
> - Delete: Deletes configurations.

Commissioning

1. Click "Debug" to access the "Debug" page.



Parameter Name		Description
NMT Command	Start Node	Sends a command to the slave station to start a node.
	Stop Node	Sends a command to the slave station to stop a node.
	Pre-run	Sends a command to the node to pre-run it.
	Reset Node	Sends a command to the node to reset it.
	Reset Communication	Sends a command to the node to reset communication.
Service Data Object (SDO)	Index and sub-index	You can only select object dictionaries in the EDS as indexes or sub-indexes.
	Value	Indicates sent or returned data.

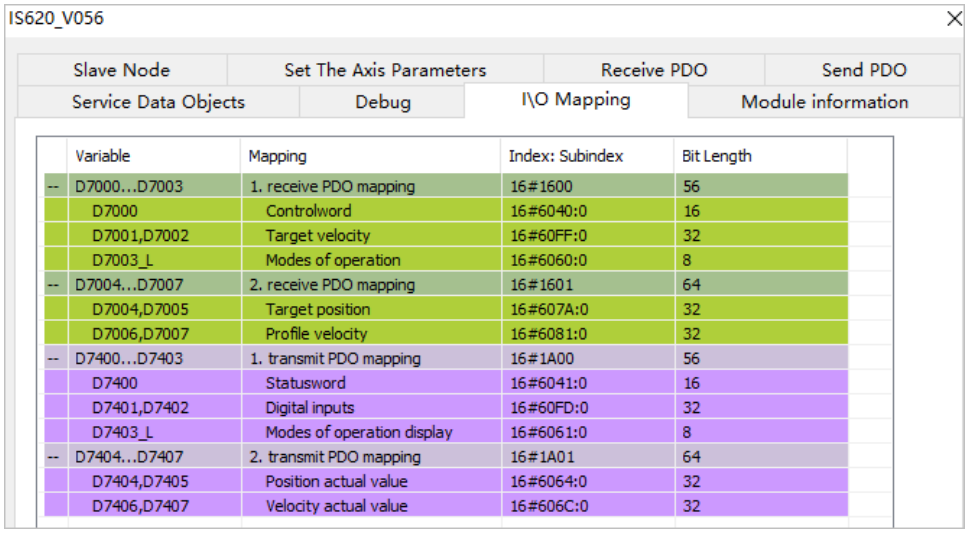
Parameter Name		Description
	Bit Length	It is automatically generated based on an object dictionary in the EDS. It must not be modified.
	Result	Error information.
	Read SDO and write SDO	Reads and writes object dictionaries.
Diagnosis	Online status	Indicates the status of the slave station (fed back based on heartbeat or node protection).
	SDO error steps	Indicates the SDO error number. This number corresponds to the "Service Data Objects" tab.
	Diagnostic string	Indicates the error message (SDO error).
	Emergency Error	Indicates an emergency error frame (the system monitors real-time errors and caches five error messages in the background; the PLC only caches the latest error message) (emergency error).

NOTE

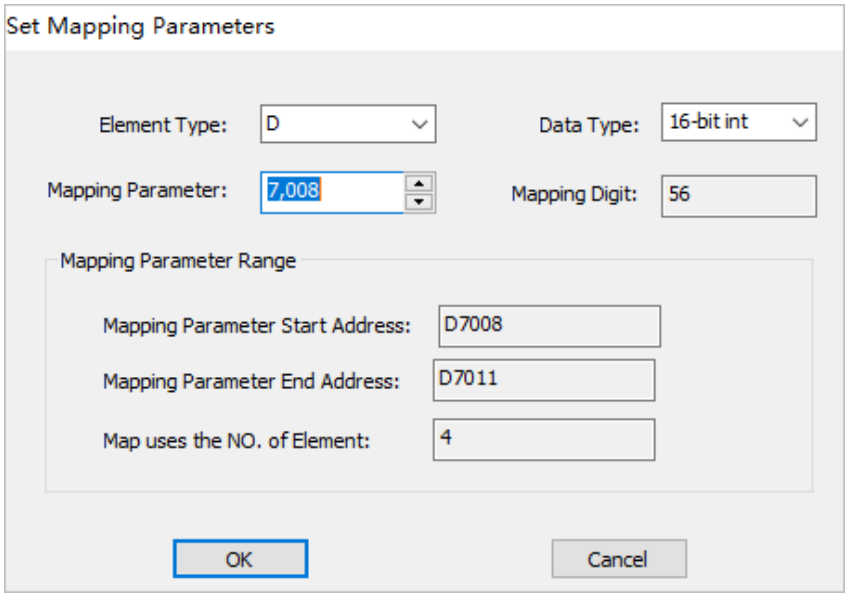
This function cannot be used if "Prohibit SDO and NMT access during program operation" is selected on the master.

I/O Mapping

1. Click "I/O Mapping" to access the "I/O Mapping" page.
   This tab is used to set the communication relationship between master and slave PDOs.



2. If "Automatic Allocation" is not selected, when you double-click an item, the "Set Mapping Parameters" page is displayed.



3. You can configure the start register address for the master station corresponding to a slave PDO.

Module Information

Click "Module information" to access the "Module information" page.

Device information can be obtained from the EDS file.

10.5.6 Troubleshooting for CANopen Communication

10.5.6.1 Troubleshooting

- Check the termination resistor
  Power off all devices. Use a multimeter to measure the resistance between CANH and CANL. The resistance should be about 60 Ω. If the resistance is too small, there are termination resistors incorrectly connected at other locations. In this case, disconnect these termination resistors. If only one termination resistor is available, the resistance is about 120 Ω, and the network connection is bad. If no termination resistor is available, communication fails. Provide termination resistors between the stations at both ends of the network.
- Check the baud rate
  Check whether the baud rate is normal. Baud rates of all stations in the network must the same; otherwise, communication fails. Power off and then on the device or switch it from STOP to RUN so that the baud rate can take effect.

For the relationship between communication distance and baud rate, see Relationship Between CAN Communication Distance and Baud Rate.

- Check cable connections
  Interconnect CGND pins of all CAN devices to ensure that all devices share one power supply CGND port of CAN communication.

  Check whether the communication cable, shielded cable, and power supply are short-circuited.
- Others
  In case of strong interference, reduce the baud rate.

10.5.6.2 Error Codes

SDO error code

Abort code	Function	Abort code	Function
0503 0000	Trigger bits not alternated	0601 0002	Attempt to write a read-only object
0504 0000	SDO protocol timed out.	0602 0000	Object unavailable in the OD
0504 0001	Invalid or unknown client /server command word	0604 0041	The object cannot be mapped to the PDO.
0504 0002	Invalid block size (for the block transfer mode only)	0604 0042	Number and length of the objects to be mapped exceeding PDO length
0504 0003	Invalid serial number (for the block transfer mode only)	0604 0043	General parameter incompatibility
0503 0004	CRC error (for the block transfer mode only)	0604 0047	General internal incompatibility
0503 0005	Memory overflow	0606 0000	Failure to access objects because of hardware errors
0601 0000	Inaccessible object	0606 0010	Incorrect data type: incorrect service parameter length
0601 0001	Attempt to read a write-only object	0606 0012	Incorrect data type: service parameter too long
0601 0002	Attempt to write a read-only object	0606 0013	Incorrect data type: service parameter too short
0602 0000	Object unavailable in the OD	0609 0011	Sub-index nonexistent
0604 0041	The object cannot be mapped to the PDO.	0609 0030	Beyond the value range (for write access)

Abort code	Function	Abort code	Function
0503 0000	Trigger bits not alternated	0609 0031	Written parameter value too large
0504 0000	SDO protocol timed out.	0609 0032	Written parameter value too small
0504 0001	Invalid or unknown client /server command word	0609 0036	Maximum value is less than minimum value.
0504 0002	Invalid block size (for the block transfer mode only)	0800 0000	General error
0504 0003	Invalid serial number (for the block transfer mode only)	0800 0020	Data unable to be transmitted or saved to the application
0503 0004	CRC error (for the block transfer mode only)	0800 0021	Data unable to be transmitted or saved to the application because of local control
0503 0005	Memory overflow	0800 0022	Data unable to be transmitted or saved to the application because of the device state
0601 0000	Inaccessible object	0800 0023	OD error or OD unavailable (For example, an OD is generated through a file, but an error occurs because the file is corrupted.)
0601 0001	Attempt to read a write-only object		

Critical error code

Main table 1 (hexadecimal)

Critical error code	Description	Critical error code	Description
00xx	No error	50xx	Hardware
10xx	General error	60xx	Software
20xx	Current	61xx	Internal software
21xx	Current at input end	62xx	User software
22xx	Internal current	63xx	Data setting
23xx	Current at output end	70xx	Extra module
30xx	Voltage	80xx	Monitoring
31xx	Power supply voltage	81xx	Communication
32xx	Internal voltage	82xx	Protocol error

Critical error code	Description	Critical error code	Description
33xx	Output voltage	90**	External error
40xx	Temperature	F0**	Extra function
41xx	Ambient temperature	FF**	Special device
42xx	Device temperature		

Main table 2 (hexadecimal)

Critical error code	Description	Critical error code	Description
0000	Incorrect reset or no error	6300	Data setting
1000	General error	7000	Extra module error
2000	Current error	8000	Monitoring error
2100	Input current	8100	General communication error
2200	Internal current	8110	CAN communication overload
2300	Output current	8120	Incorrect CAN passive method
3000	The voltage is incorrect.	8130	Node protection or heartbeat error
3100	Power supply voltage	8140	Bus disconnection
3200	Internal voltage	8150	CAN-ID impulse
3300	Output voltage	8200	Protocol error
4000	Temperature error	8210	Incorrect PDO length
4100	Ambient temperature	8220	Excessive PDO length
4200	Device temperature	8240	Unidentifiable sync data length
5000	Hardware error	8250	RPDO timeout
6000	Software error	9000	External error
6100	Internal software	F000	Extra function error
6200	User software	FF00	Special device error

11 EtherCAT Communication

11.1 Overview

EtherCAT is an open industrial field technology based on Ethernet. It features short communication refresh cycles, low synchronization jitter, and low hardware costs. For details about EtherCAT principles and related technologies, see the book "Industrial Ethernet Fieldbus EtherCAT Driver Design and Applications" or visit the official website of the EtherCAT Technical Group at https://www.EtherCAT.org.cn.

The H5U and Easy500 series support standard EtherCAT ports (one RJ45 port). In the linear topology, they support a maximum of 72 EtherCAT slave stations and a minimum EtherCAT bus cycle of 1 ms.

The following table lists the EtherCAT port specifications.

Item	Specifications
Transmission rate	100 Mbps: 100BASE-TX
Modulation	Baseband
Topology	Linear and daisy chain
Medium	Cat5 twisted pairs or shielded twisted pairs with aluminum foil and braided mesh
Transmission distance	Distance between nodes: 100 m or less
Number of connections	72

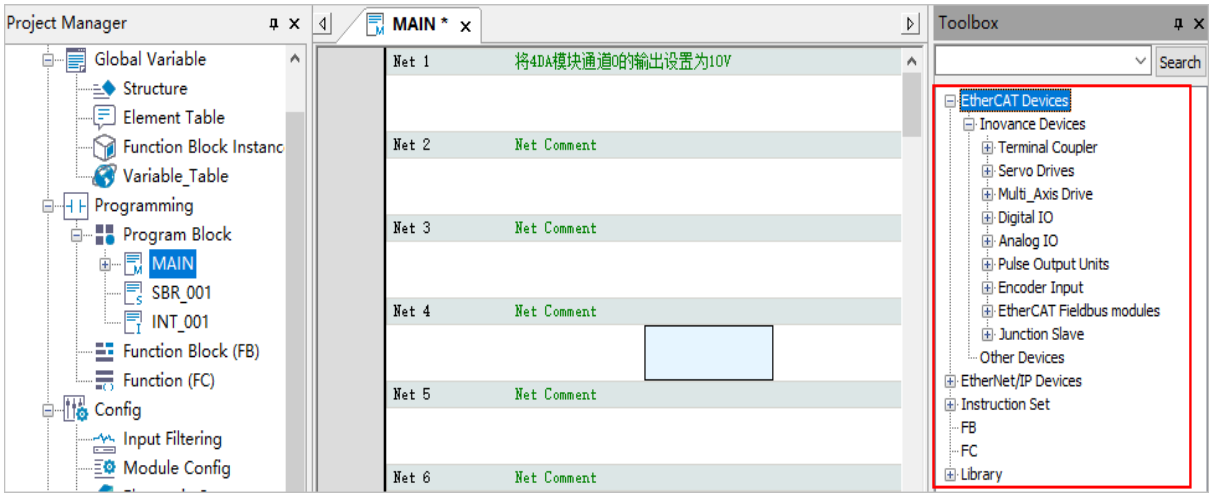
11.2 Master Configuration

11.2.1 Importing Device Description XML

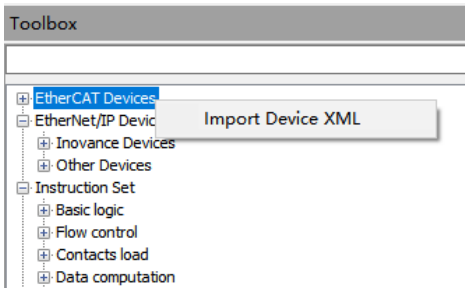
Importing device description XML means importing the device description file with the suffix ".XML" that meets standards of the EtherCAT Technical Group (ETG) into the programming software AutoShop, in which, the file is parsed into EtherCAT configuration devices that can be added or deleted by users.

AutoShop provides built-in EtherCAT slave stations of Inovance, and therefore the device description files do not need to be installed. To use third-party EtherCAT devices, their description files must be installed. The following section takes importing the description file of Inovance bus motor drive SV520N as an example.

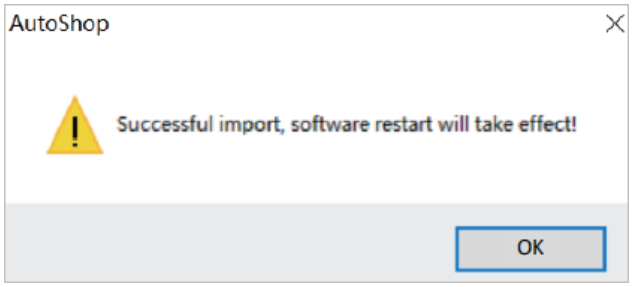
1. Create a project, open the toolbox, and locate "EtherCAT Devices".



2. Right-click "EtherCAT Devices". In the dialog box displayed, select and import the target XML file.

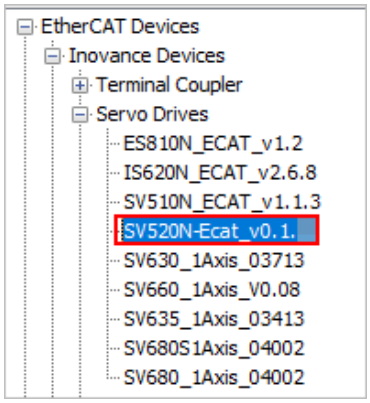


3. Restart the software to make the imported XML file take effect. To import the XML files of multiple devices, repeat Step 2 and then restart the device.



You must manually restart the software before the added devices take effect.

4. The added devices are added to the list after the software is reaccessed.



11.2.2 Scanning a Device

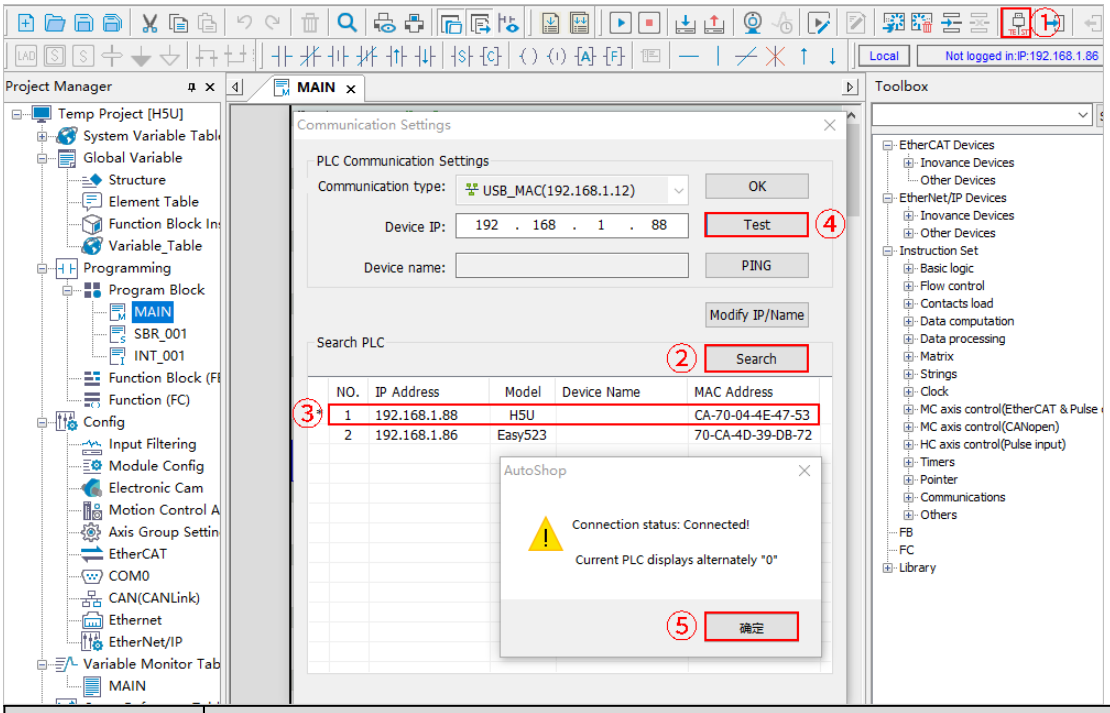
11.2.2.1 Offline Scanning

The following section describes how to scan a device when the communication mode is Ethernet.

NOTE

EtherCAT slaves can only be scanned when the PLC is in the stopped state.

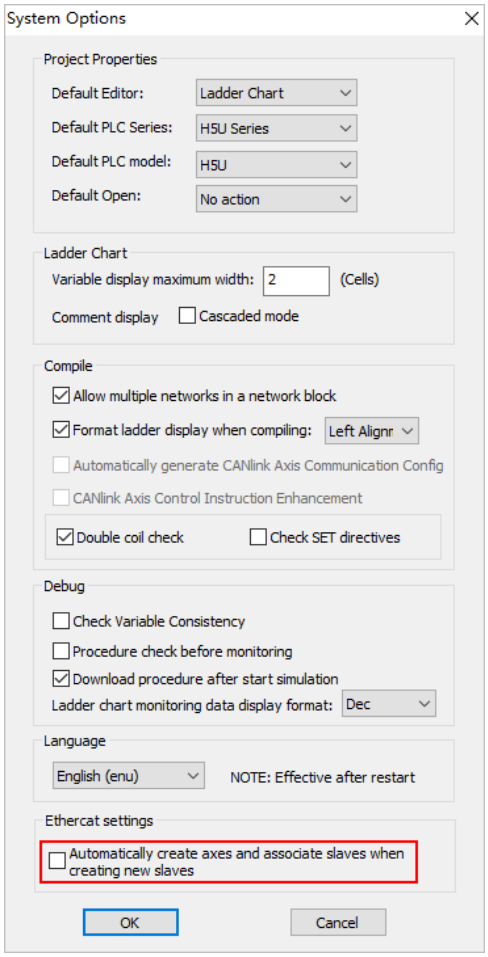
1. Select the target host.



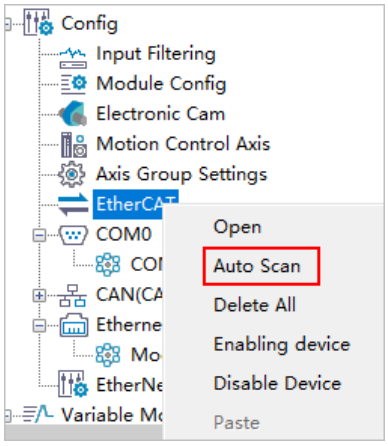
No.	Operation Description
①	Click the icon to test the communication state.
②	On the page displayed, click "Search".
③	Select the target host.
④	Click "Test".
⑤	After confirming that the host is connected, click "OK".

2. Determine whether to automatically associate the motion control axis.

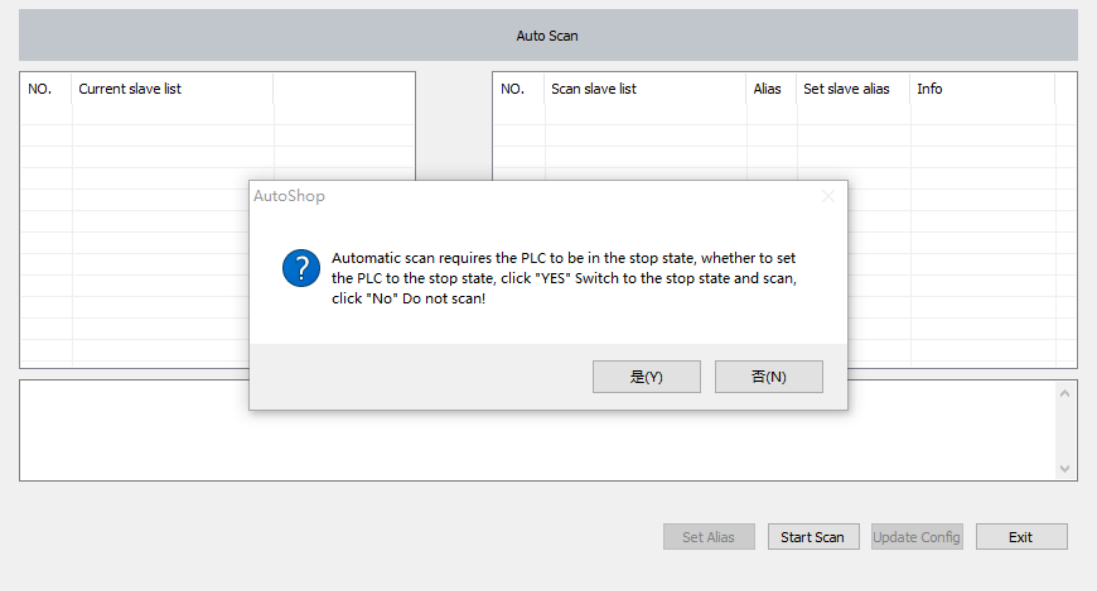
   If "Automatically create axes and associate slaves when creating new slaves" is selected, each time an EtherCAT slave of the drive type (such as IS620N) is added, a motion control axis is automatically added. In this case, this option is not selected.



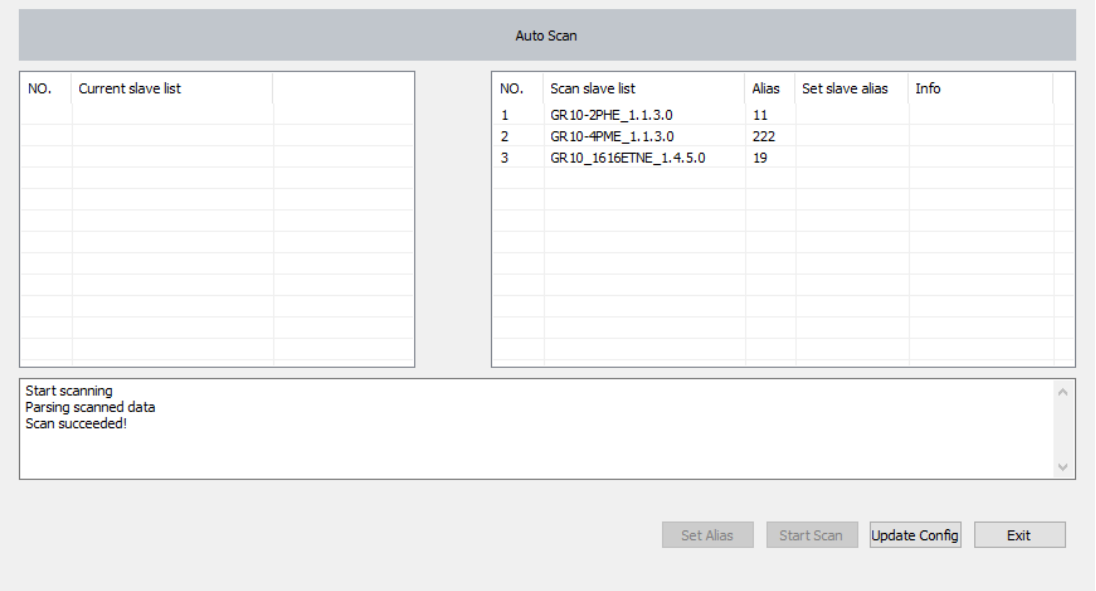
3. Right-click EtherCAT and then select "Auto Scan".



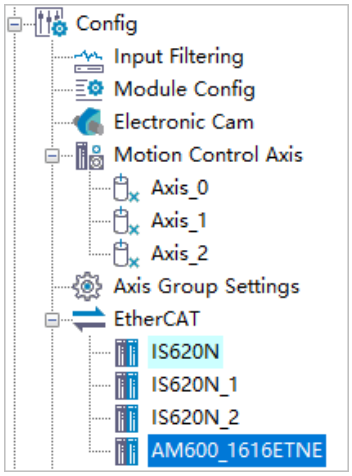
4. In the dialog box displayed, click "Start Scan" at ①. If the PLC is running, click "Yes" at ② to stop the PLC first.



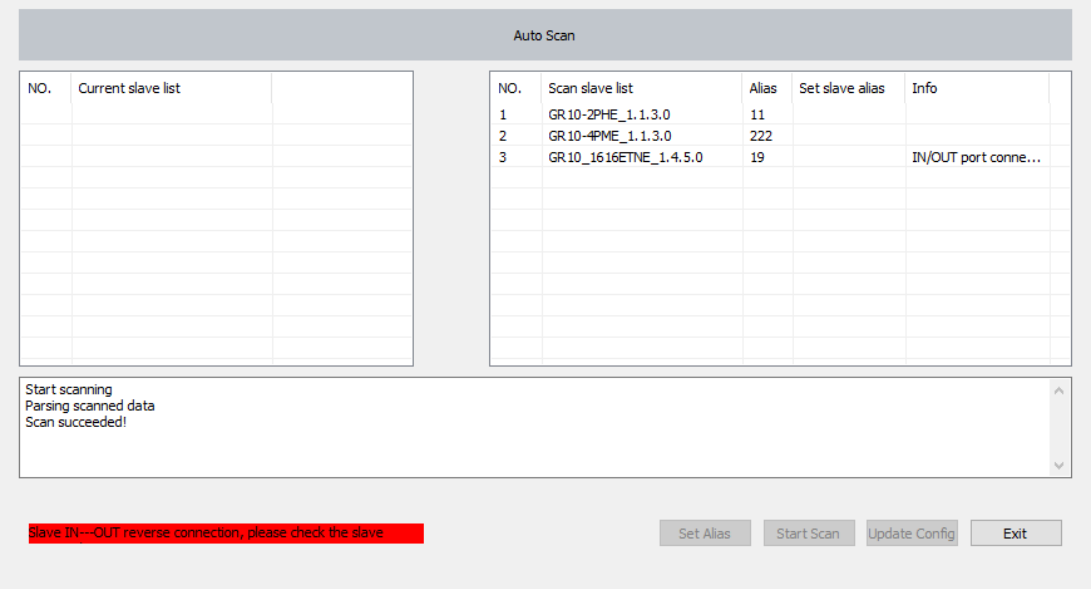
5. When scanning is completed, the found slaves are displayed. Click "Update Config" to add the found devices to the configuration list, or click "Exit" to not add the found devices to the configuration list.



After the configuration list is updated, the following figure is displayed and IS620N is automatically associated with the motion control axis.



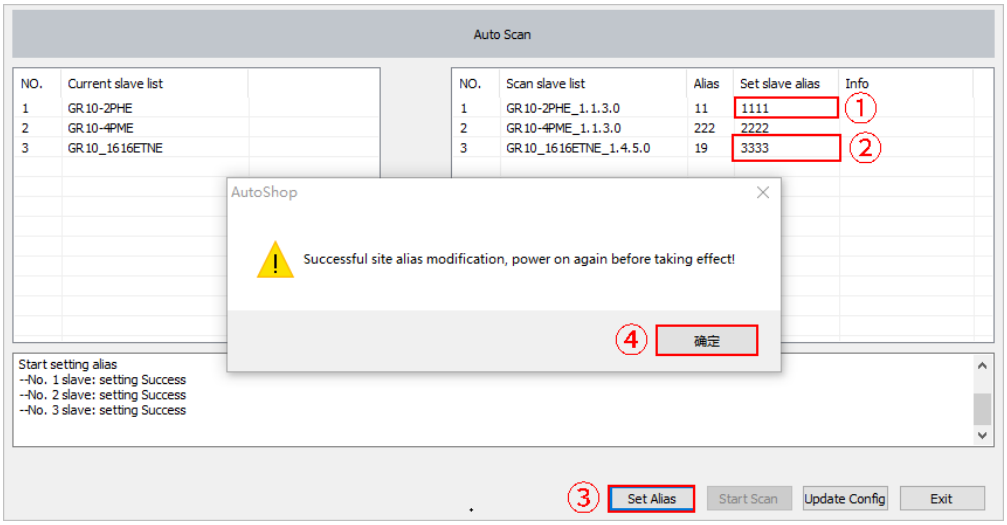
When the IN/OUT pins of the slave station are reversed, the message "IN/OUT port connection error" is displayed in the "Message" column. In this case, the "Update Config" button is grayed out, and you need to manually check the connection of the physical link and perform scanning again.



NOTE

- The H5U series with firmware of version 6.1.0.1 or later support IN/OUT port reversing connection detection.
- The Easy series with firmware of version 6.3.0.0 or later support IN/OUT port reversing connection detection.

6. After auto scan is completed, you can set alias for stations in batch.



No.	Operation Description
①	In the "Write site alias" column, double-click the cell of the target slave, and write an alias address.
②	Write the alias addresses for all target slaves in sequence.
③	Click "Set Alias". If the "Write site alias" column is empty, this button is grayed.
④	After alias addresses are written and modification success is prompted, click "OK". If modification failure is prompted, set alias again as instructed.

Restart the corresponding slaves to make the settings take effect.

NOTE

- Batch station alias setting is support by H5U with a firmware version of 6.3.0.0 or later and by Easy with a firmware version of 6.3.0.0 or later.
- If no EtherCAT slave is detected by auto scan, the batch alias setting function is not available.
- The value of cells in the "Write site alias" column is blank by default, and only be numbers ranging from 1 to 65535.
- If the aliases of two slaves in the "Write site alias" column are the same, an error will be reported during program compilation and download in the alias mode.

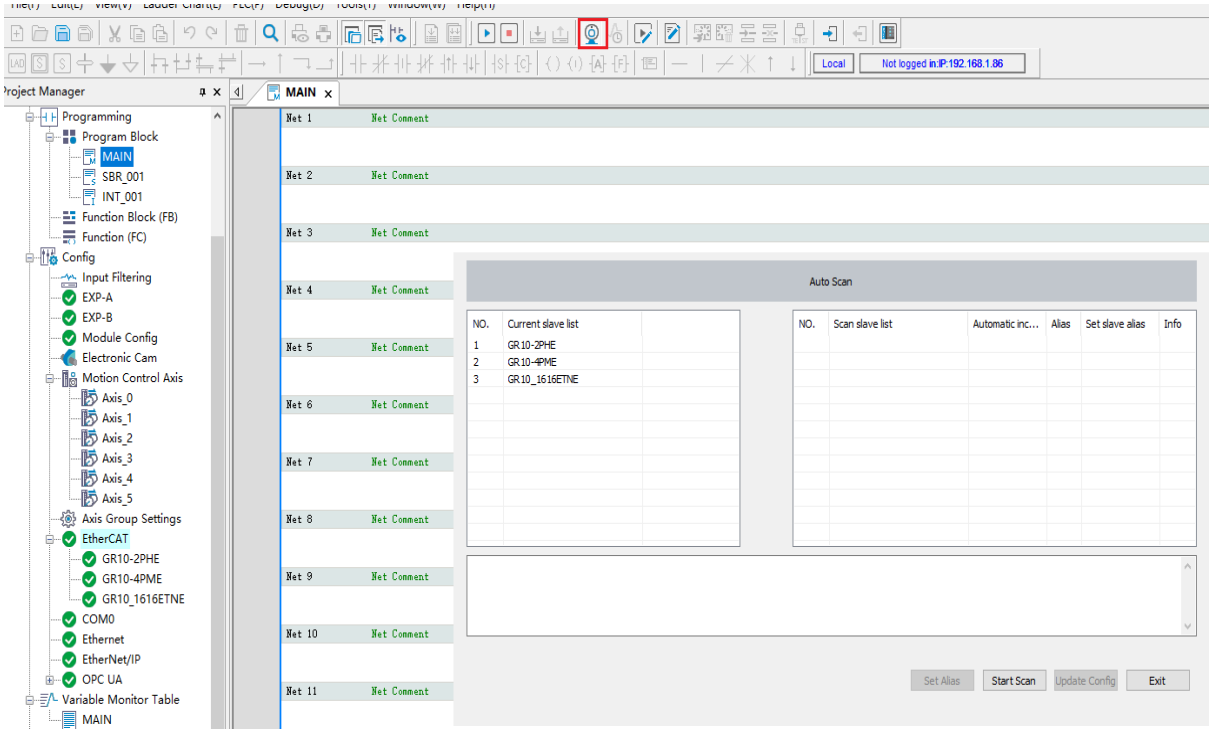
11.2.2.2 Online Scanning

Preconditions

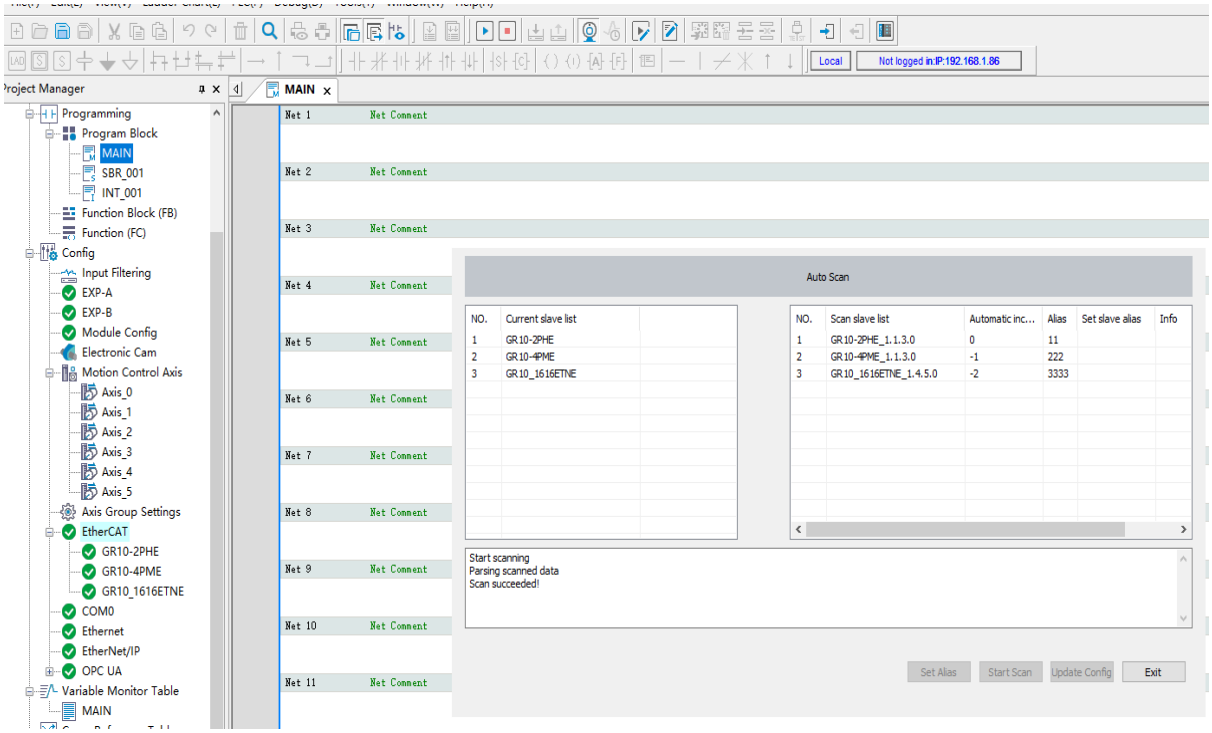
- The Easy firmware version is 6.4.0.0 or later.
- Online scanning does not support configuration upgrade and batch alias setting.

Procedure

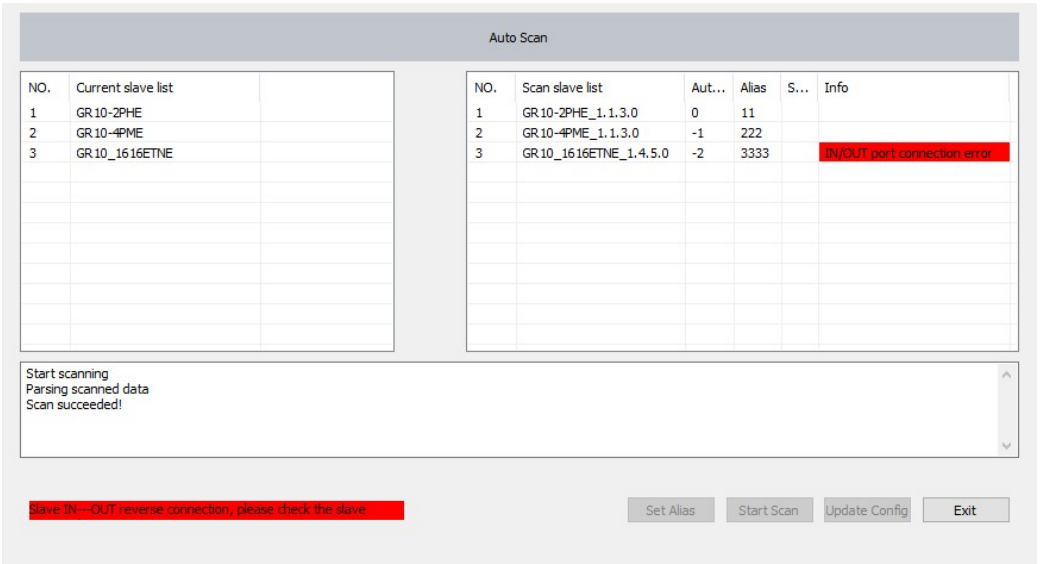
1. When the PLC is running, click "Monitor" to enter the monitoring state.
2. In the "Project Manager" tree, right-click "EtherCAT" and select "Auto Scan", as shown in the following figure.
3. In the dialog box displayed, select "Start Scan", as shown in the following figure.



4. After scanning is completed, the slaves detected can be viewed. Click "Exit" to exit online scanning, as shown in the following figure.

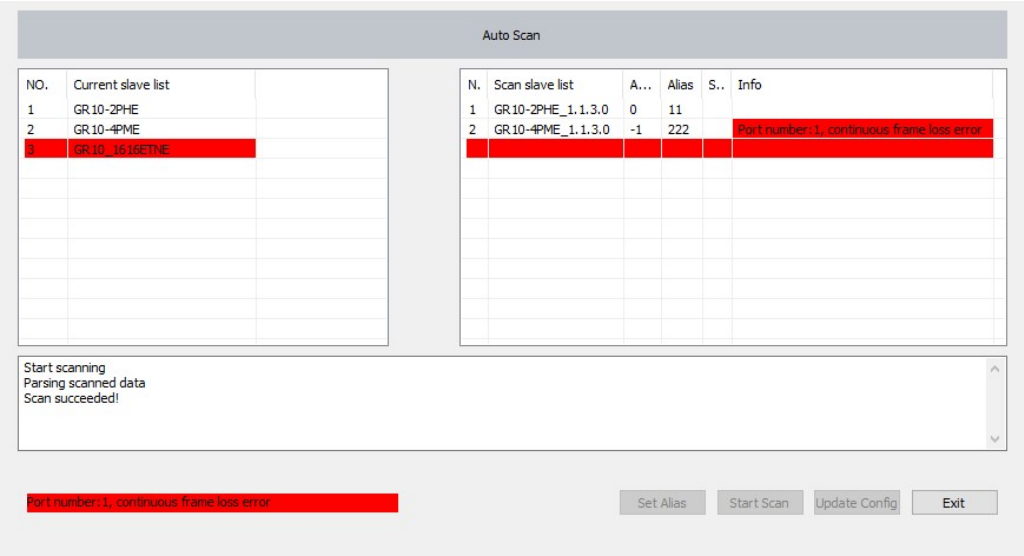


- When IN/OUT port reversing connection is detected, "IN/OUT port connection error" is displayed in the information bar of the corresponding slave, as shown in the following figure.



> **NOTE**
>
> The Easy series with firmware of version 6.4.0.0 or later support IN/OUT port reversing connection detection.

- If continuous frame loss occurs in any link, the system displays the port with continuous frame loss in the information bar of the corresponding slave, as shown in the following figure.

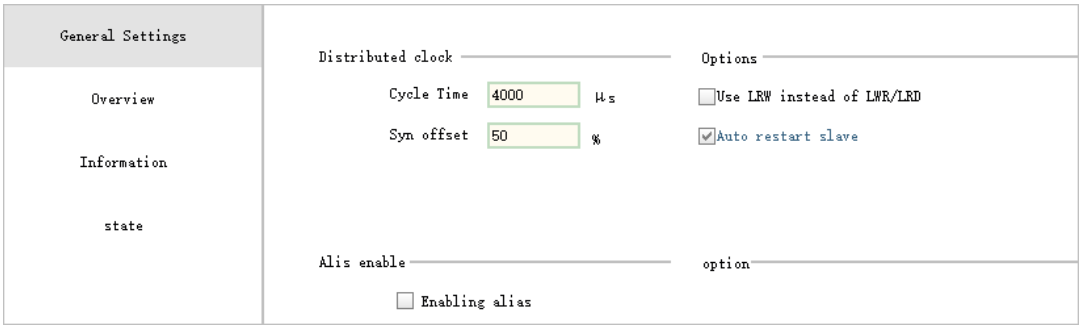


> **NOTE**
>
> The Easy series with firmware of version 6.4.0.0 or later support continuous frame loss detection.

11.2.3 Configuring a Master

Procedure

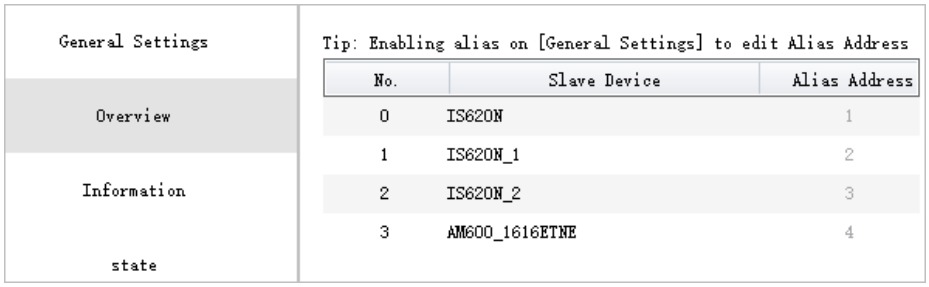
1. In the "Project Manager" tree, double-click "EtherCAT". On the "General Settings" page, set the cycle time, synchronization offset, and alias enabling for the EtherCAT master, as shown in the following figure.



Parameter Name	Description
Cycle Time	Indicates the EtherCAT data frame sending interval and the EtherCAT task cycle time.
Syn offset	Indicates the relative offset (in percentage) of the EtherCAT task with respect to the Sync0 interrupt of the slave.

Parameter Name	Description
Use LRW instead of LWR/LRD	This option is not selected by default, that is, logic read is separated from logic write. If this option is selected, logic read/write is applied. Do not select this option for slaves that only support separate logic read or logic write. This option is supported in H5U and Easy firmware of versions 6.3.0.0 and later.
Auto restart slave	When this option is selected, the EtherCAT slave station is automatically restarted when it attempts to connect to the network after disconnection. Note: This function is supported by AutoShop 4.0.0.0 matching the PCB software 3.0.0.0.
Enabling alias	After this option is selected, the aliases of all EtherCAT slave stations are enabled. In this case, even "Enabling alias" is not selected for some slave stations, such slave stations are stilled started with their aliases. When the network contains branch modules, you are recommended to select this option. If "Enabling alias" is not selected for the master station, when you select "Enabling alias" for a single slave station, the alias mode is applied only to this slave station. Then, aliases are mixed with formal names, causing a communication error.

2. In the "Project Manager" tree, double-click "EtherCAT". Click "Overview" to modify the slave alias on this tab page, as shown in the following figure.



This tab page displays information such as the slave No. and slave device and alias address configured in the background. The slave No. and slave device cannot be edited. If "Enabling alias" is selected on the "General Settings" tab page, the alias address can be edited, which can only be numbers and must be within the range of 1 to 65535. Numbers exceeding this range will be automatically modified to 65535.

CAUTION

- On the "Overview" tab page, when you edit the slave alias, the alias address, instead of the actual alias, of the EtherCAT slave under the current project is modified (the actual alias is stored to the slave EEPROM). If the modification is inconsistent with the actual alias of the slave, this slave cannot start when the PLC runs.
- When the slave is in the state of monitoring, online modification, or offline simulation, its alias address cannot be edited.
- The alias modification made on the "Overview" tab page is synchronized to the page of the corresponding slave. The alias modification made on the page of a slave is synchronized to the "Overview" tab page. The slave adding, deletion, and moving operations in the project tree on the right of the slave page are synchronized to the "Overview" tab page.

11.2.4 Device Start/Stop, Disable, and Enable

Start/Stop control

You can start or stop an EtherCAT bus but cannot start or stop a single slave station. The procedure is as follows.

- When the PLC state changes from STOP to RUN, the EtherCAT master station starts to run automatically.
- When the PLC state changes from RUN to STOP, the EtherCAT master station stops automatically.
- When the PLC is running, you can use the system variables to start or stop the EtherCAT master station.

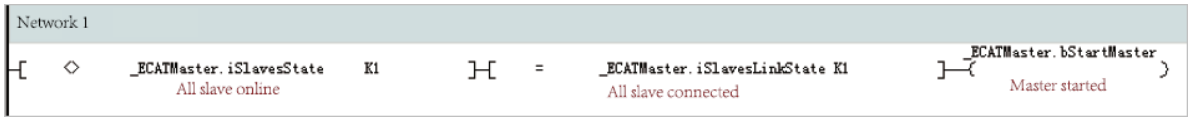
NOTE

Only the startup and stop of EtherCAT bus are supported, while the startup and stop of individual slaves are not supported.

System variable	Data type	Function
bStopMaster	BOOL	Stop of the EtherCAT master station The EtherCAT master station stops at the rising edge of the variable input and then the variable is automatically reset.
bStartMaster	BOOL	Startup of the EtherCAT master station When the EtherCAT bus fails or stops, the EtherCAT master station is restarted at the rising edge of the variable input and then the variable is automatically reset.

Auto Restart

For AutoShop 4.0.0.0 matching PCB software of a version earlier than 3.0.0.0, you can run the PLC program together with system variables iSlavesState, iSlavesLinkState, and bStartMaster to automatically restart the EtherCAT bus.

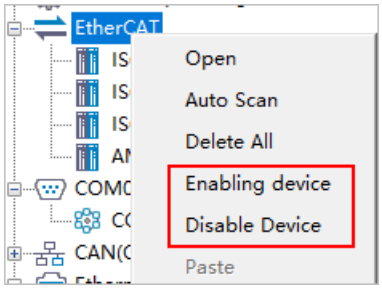


For AutoShop 4.0.0.0 matching PCB software of version 3.0.0.0 or later, you can select “Auto restart slave” on the “General Settings” tab page to automatically restart the slave station.

Disable and Enable

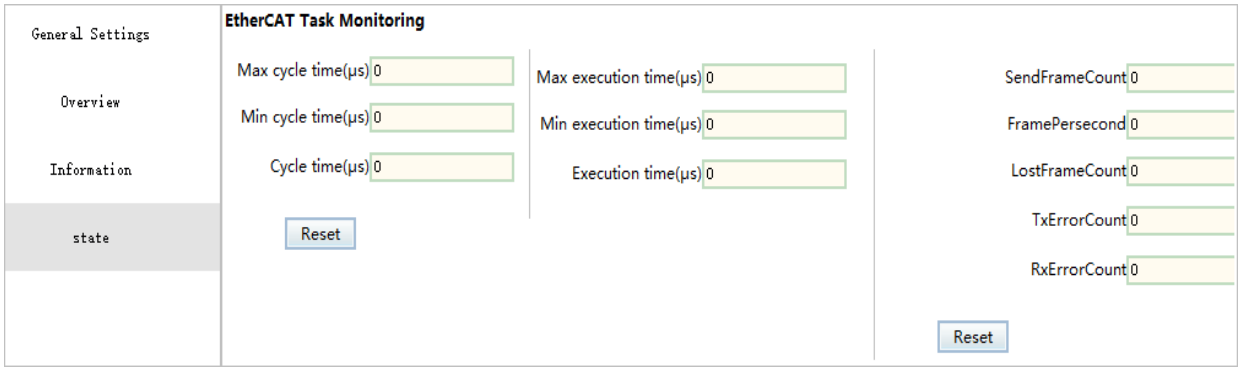
When “Motion Control Axis > Virtual Axis Mode” is selected, even if you disable the EtherCAT master station (all the master and slave stations under this bus are disabled), you can still control the motion control axis through the program. For details, see description of the virtual axis mode of the motion control axis. Note that after the EtherCAT master station is disabled, all EtherCAT slave stations are disabled and bus servo axes associated with the slave stations cannot move.

Right-click the EtherCAT master station, and then select “Enabling device” or “Disable Device”.



11.2.5 Monitoring the Master Status

On the “Status” page, you can view the running information of the EtherCAT bus, as shown in the following figure.



- The left column displays the execution information of the EtherCAT task. The following table lists functions and corresponding system variables.

System variable	Data type	Function
dMaxCycleTime	DINT	Maximum cycle time of the EtherCAT task
dMinCycleTime	DINT	Minimum cycle time of the EtherCAT task
dCycleTime	DINT	Cycle time of the EtherCAT task in the previous period
dMaxExeTime	DINT	Maximum execution time of the EtherCAT task
dMinExeTime	DINT	Minimum execution time of the EtherCAT task

System variable	Data type	Function
dExeTime	DINT	Execution time of the EtherCAT task in the previous period
bResetTime	BOOL	Execution time and cycle time for resetting

- The right column displays the data sending and receiving status of the EtherCAT bus.

System variable	**Data type**	**Function**
dtx_frames	DINT	Number of EtherCAT frames sent
dtx_frames_rates	DINT	EtherCAT frame sending rate
dloss_frames	DINT	Number of lost EtherCAT frames
dtx_error_cnt	DINT	Number of EtherCAT frame sending errors
drx_error_cnt	DINT	Number of EtherCAT frame receiving errors

- You can also use system variables to monitor the running, stop, or connection status of the master station.

System variable	Data type	Function
bMasterRunState	BOOL	Running status of the EtherCAT master station After the EtherCAT master station receives the RUN command and all slave stations are started, this variable becomes TRUE. Note: If some slave stations are disconnected during EtherCAT running, this variable is still TRUE.
bLinkState	BOOL	Connection status of the master station The variable is ON as long as one slave station is physically connected to the master station. The variable is OFF if no slave station is physically connected to the master station.
iSlavesState	INT	Online status of all slave stations When all the configured slave stations are running, the value is 1. When any slave station is not running, the value is 0.
iFirstErrorSlave	INT	When a configured slave station fails (the state machine switches to a non-OP state or is offline), this variable displays the configuration location of the first disconnected slave station.
iSlavesLinkState	INT	Physical connection status of all slave stations When the physical connections of all configured slave stations are normal, the value is 1; if the physical connection of any slave station is abnormal, the value is 0.

11.2.6 List of System Variables

The following table lists system variables of EtherCAT communication.

System variable	Data type	Function
bMasterRunState	BOOL	Running status of the EtherCAT master station After the EtherCAT master station receives the RUN command and all slave stations are started, this variable becomes TRUE. Note:If some slave stations are disconnected during EtherCAT running, this variable is still TRUE.
bLinkState	BOOL	Connection status of the master station The variable is ON as long as one slave station is physically connected to the master station. The variable is OFF if no slave station is physically connected to the master station.
bHeartBeat	BOOL	EtherCAT real-time task heartbeat Flip once per EtherCAT real-time task cycle
bBolckHeartBeat	BOOL	EtherCAT non-real-time task heartbeat Flip once per EtherCAT non-real-time task cycle
dMaxCycleTime	DINT	Maximum cycle time of the EtherCAT task
dMinCycleTime	DINT	Minimum cycle time of the EtherCAT task
dCycleTime	DINT	Cycle time of the EtherCAT task in the previous period
dMaxExeTime	DINT	Maximum execution time of the EtherCAT task
dMinExeTime	DINT	Minimum execution time of the EtherCAT task
dExeTime	DINT	Execution time of the EtherCAT task in the previous period
dtx_frames	DINT	Total number of sent frames
drx_frames	DINT	Total number of received frames
dtx_frames_rates	DINT	Frame sending rate (frame/s)
drx_frames_rates	DINT	Frame receiving rate (frame/s)
dtx_bytes_rate	DINT	Frame sending rate (byte/s)
drx_bytes_rate	DINT	Frame receiving rate (byte/s)
dloss_frames	DINT	Number of lost data frames by EtherCAT
bResetTime	BOOL	Execution time and cycle time for resetting
bStopMaster	BOOL	Stop of the EtherCAT master station

System variable	Data type	Function
		The EtherCAT master station stops at the rising edge of the variable input and then the variable is automatically reset.
bStartMaster	BOOL	Startup of the EtherCAT master station When the EtherCAT bus fails or stops, the EtherCAT master station is restarted at the rising edge of the variable input and then the variable is automatically reset.
bClearFrameCounter	BOOL	Resetting of the EtherCAT data frame counter register
bDisableMaster	BOOL	Master station disabled
iSlavesState	INT	Online status of all slave stations When all the configured slave stations are running, the value is 1. When any slave station is not running, the value is 0.
iFirstErrorSlave	INT	When a configured slave station fails (the state machine switches to a non-OP state or is offline), this variable displays the configuration location of the first disconnected slave station.
dLibVersion	DINT	Version of EtherCAT system library software
dMstVersion	DINT	Version of EtherCAT master station software
dDriveVersion	DINT	Version of EtherCAT network adapter drive board software
dtx_error_cnt	DINT	Number of EtherCAT data frame sending errors
drx_timeout_cnt	DINT	Number of EtherCAT data frame receiving timeout times
drx_corrupt_cnt	DINT	Number of invalid frame receiving events by EtherCAT
drx_unmatch_cnt	DINT	Number of mismatched frame receiving events by EtherCAT
dRxPDOLength	DINT	Total length of received PDOs in the configuration (bytes)
dTxPDOLength	DINT	Total length of send PDOs in the configuration (bytes)
dConfigureState	DINT	For internal use of configuration status
dDelay	DINT	Adjustment value of EtherCAT master station synchronization regulator
iSlavesLinkState	INT	Physical connection status of all slave stations When the physical connections of all configured slave stations are normal, the value is 1; if the physical connection of any slave station is abnormal, the value is 0.
iUnused1	INT	Reserved 1

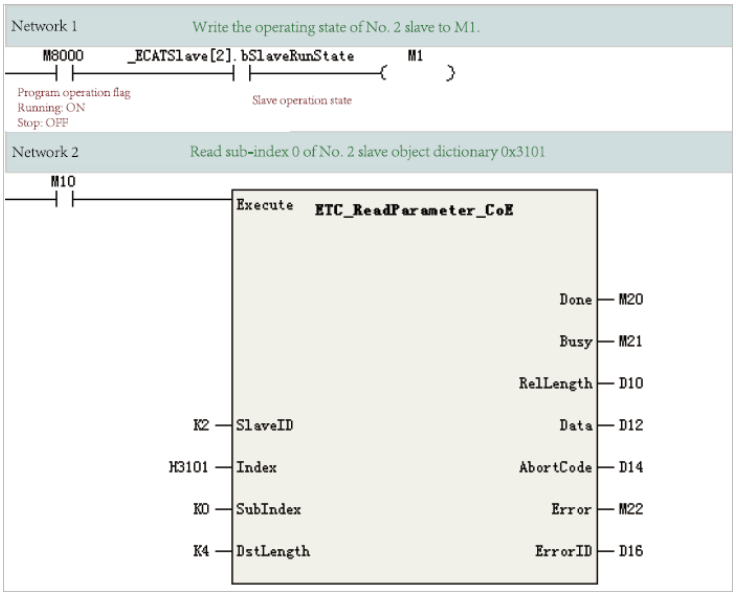
System variable	Data type	Function
iUnused2	INT	Reserved 2
iDisableState	INT	Master station disabled
drx_error_cnt	DINT	Number of EtherCAT data frame receiving errors
ichange_finished	INT	The state switchover of all slaves is completed.

11.3 Slave Configuration

11.3.1 General Settings

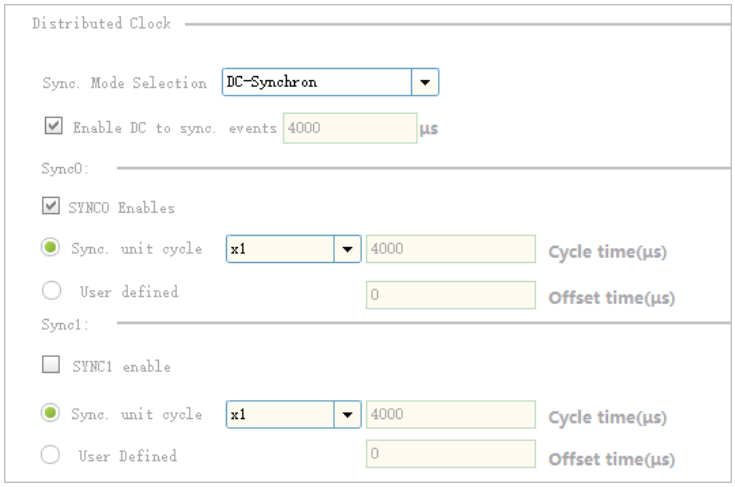
Configuration address

Configuration addresses of slave stations are sequential addresses in the AutoShop device tree, which increase from 0. The configuration address can be used as a subscript of the slave system variable array or as the slave address for reading and writing SDO instructions.



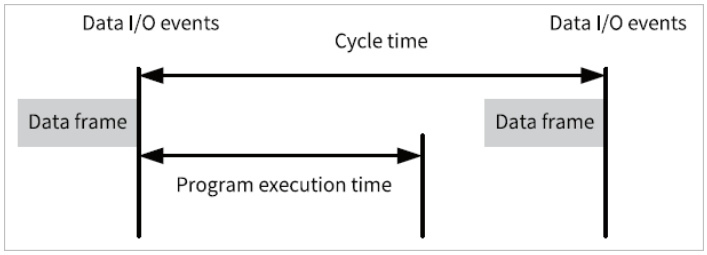
Distributed Clock

You can set the synchronization running mode for a slave station on the following page.

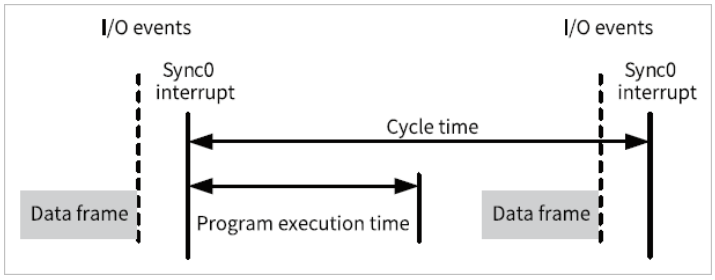


Sync. Mode Selection: For an EtherCAT slave station, options are "FreeRun", "SM-Synchron", and "DC-Synchron". The available options vary with the selected slave station.

For example, the GL10-RTU-ECTA module only supports SM-Synchron, and does not support SYNC0 and SYNC1. There is only one interrupt for data input and output events inside the clock slave station, and the internal processing logic of the slave station is shown in the following figure.



Take the GR10-4PME module that supports only DC-Synchron as an example. In this mode, the Sync interrupt of the slave station can be configured. The DC Sync event is enabled by default, the SYNC0 interrupt is enabled, the period of the SYNC0 interrupt is the same as the cycle time of the EtherCAT master station, and the SYNC1 interrupt is not enabled.



NOTE

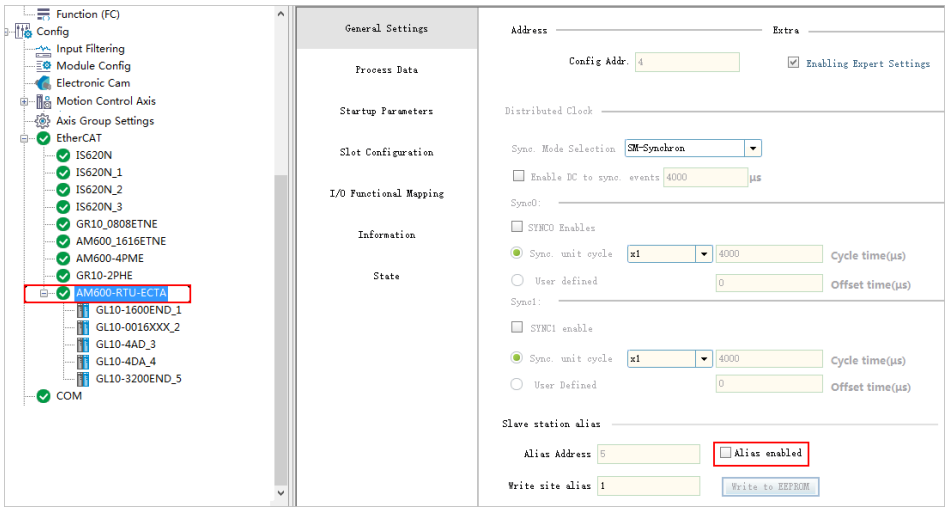
Users not familiar with the EtherCAT communication principles shall not modify the default configuration in DC-Synchron mode.

Setting station alias

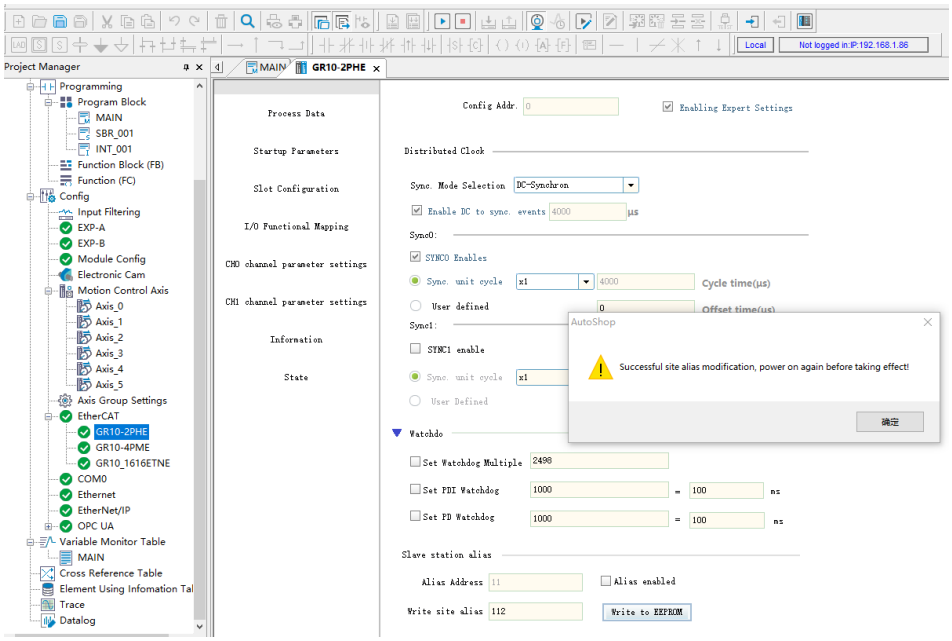
An alias can be set for a slave station only when the expert mode is enabled. To enable the station alias function, you must set an alias for the slave station first. You can set parameters to set an alias for Inovance

servo, use the DIP switch to set an alias for the GR10-0808ETNE module, and set the EtherCAT master station to set an alias for GL10-RTU-ECTA. Set an alias for an ECTA module as follows.

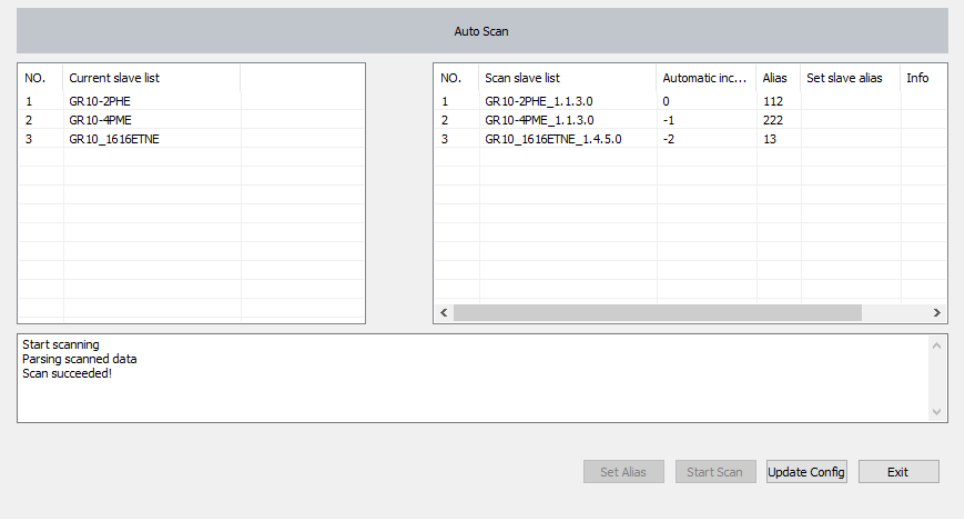
1. Create a configuration, do not select "Alias enabled", download the program, and wait until the slave station is started.



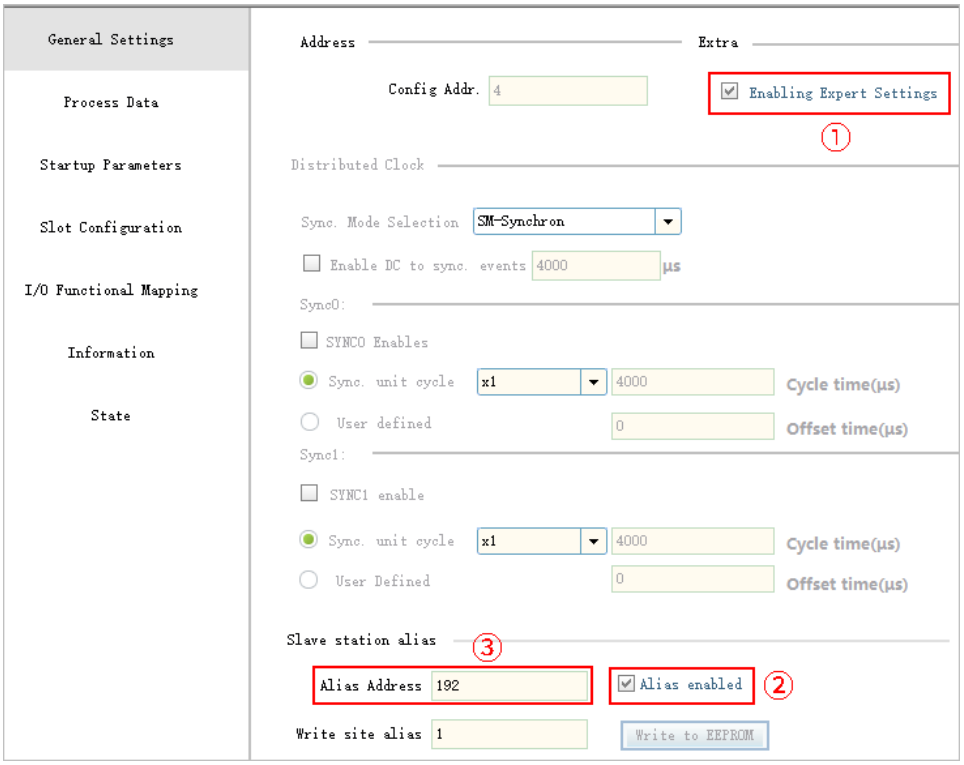
2. Select "Enable Expert Settings" at ①. In the "Write site alias" text box, input an alias at ②, enter the monitoring status, and then click "Write to EEPROM" at ③. Wait until the write operation is completed, click "OK" at ④, and start the master. The following takes writing 1 as an example.



3. Power on the slave station again. You can perform auto scanning to check whether the alias is written.



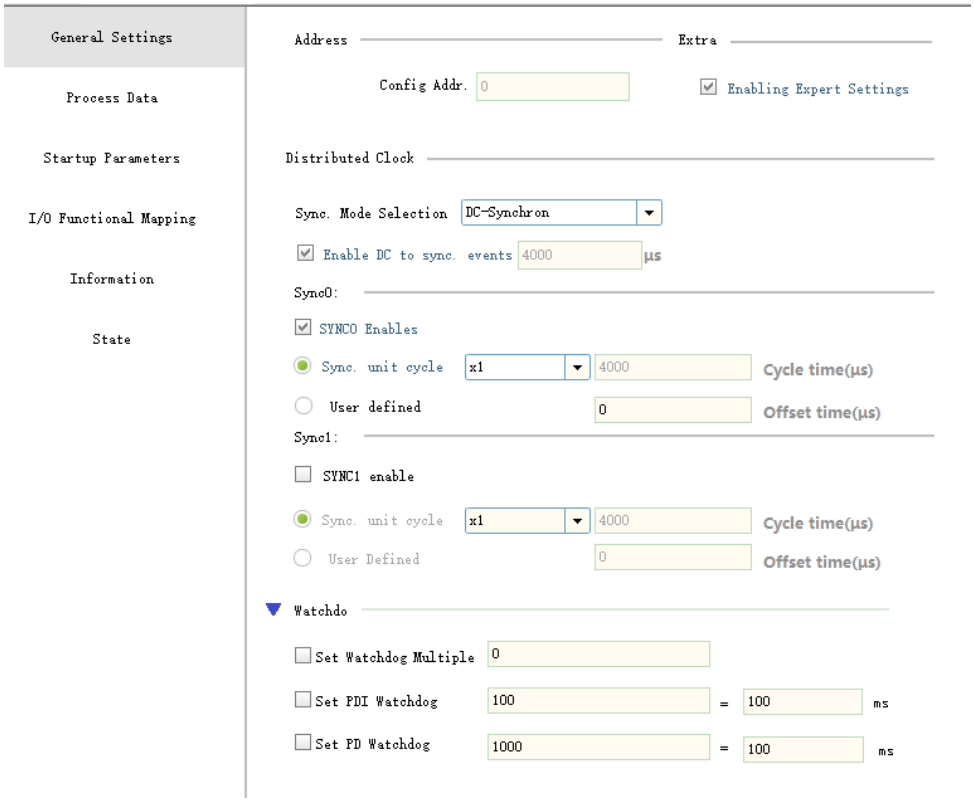
4. In the configuration, select "Alias enabled" at ①. In the "Alias address" text box at ②, input the actual alias of the current slave station at ③, and then download the program, as shown in the following figure.



NOTE

- Avoid mixing the use of station alias and station names. Avoid alias conflicts when using aliases.
- It is recommended t enable the alias function for all slaves when setting up branch module networking.
- After the alias function is enabled for slaves, function blocks for reading and writing SDOs and system variables for accessing the slaves still use the configuration addresses.

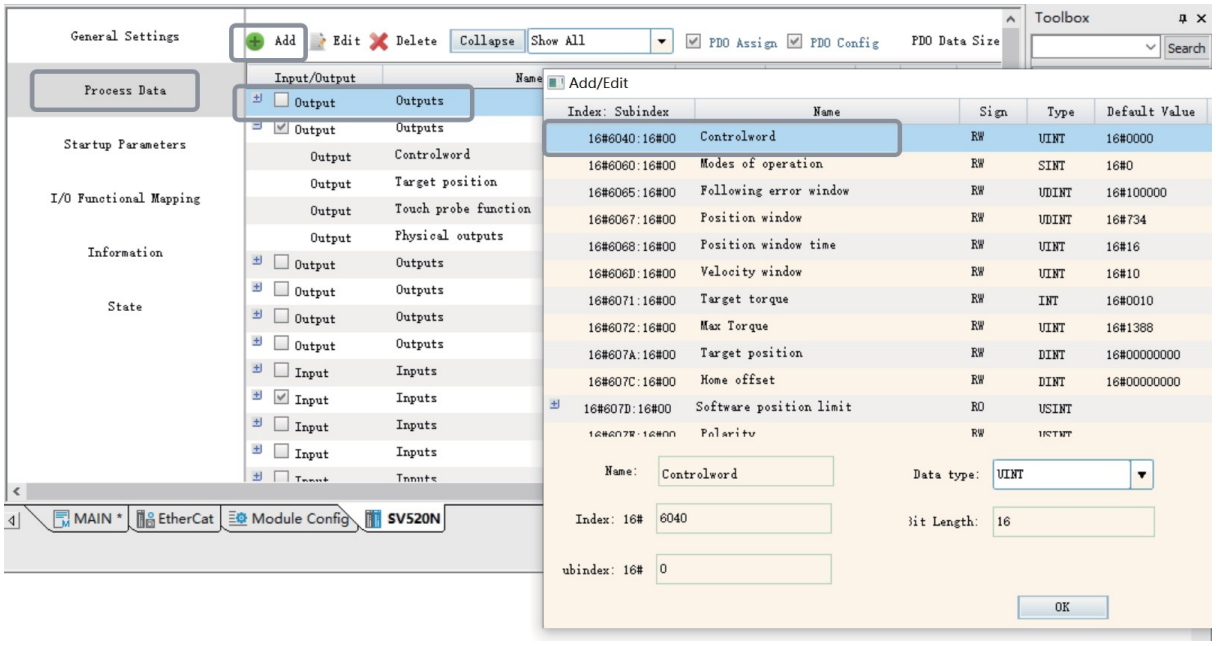
5. An alias can be set for a slave station only when the "Enabling Expert Settings" is enabled.



Name	Description
Set Watchdog Multiple	Sets the frequency multiplication ratio of the watchdog timer to determine the minimum increment unit. Default: 2498
Set PDI Watchdog	If the process data interface (PDI) watchdog is enabled, when the PDI communication time of the EtherCAT slave exceeds the specified value, the watchdog is triggered. Default: 100 ms
Set PD Watchdog	Activate the PD watchdog. When the process data (PD) communication time of the EtherCAT exceeds the specified value, the watchdog is triggered. Default: 100 ms

11.3.2 Process Data

You can edit PDOs on the "Process Data" page, which is shown in the following figure.



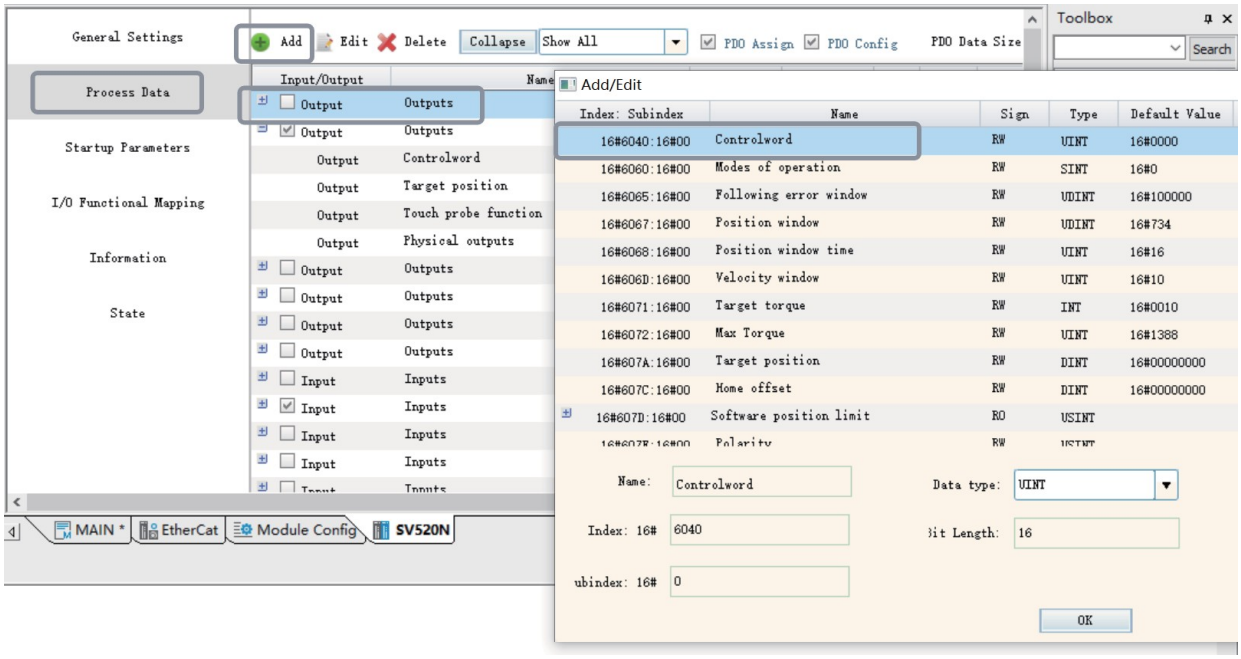
No.	Description
①	PDO editing button
②	PDO configuration downloading selection area
③	PDO display area

PDOs are divided into output PDOs and input PDOs by data flow direction. Output PDOs indicate process data sent from the EtherCAT master station to an EtherCAT slave station, such as the control word 0x6040. Input PDOs indicate process data sent from an EtherCAT slave station to the master station.

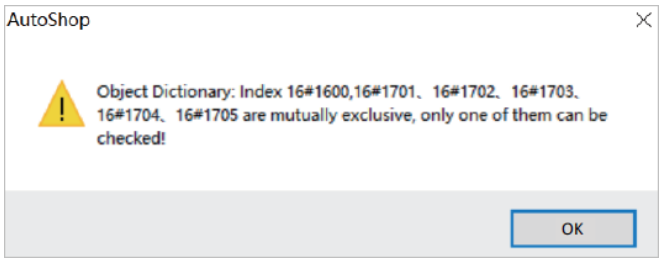
Each slave station may have one or more groups of PDOs. As shown in the preceding figure, the first group of input PDOs and the first group of output PDOs can be added, edited, or deleted.

The following describes how to add a PDO.

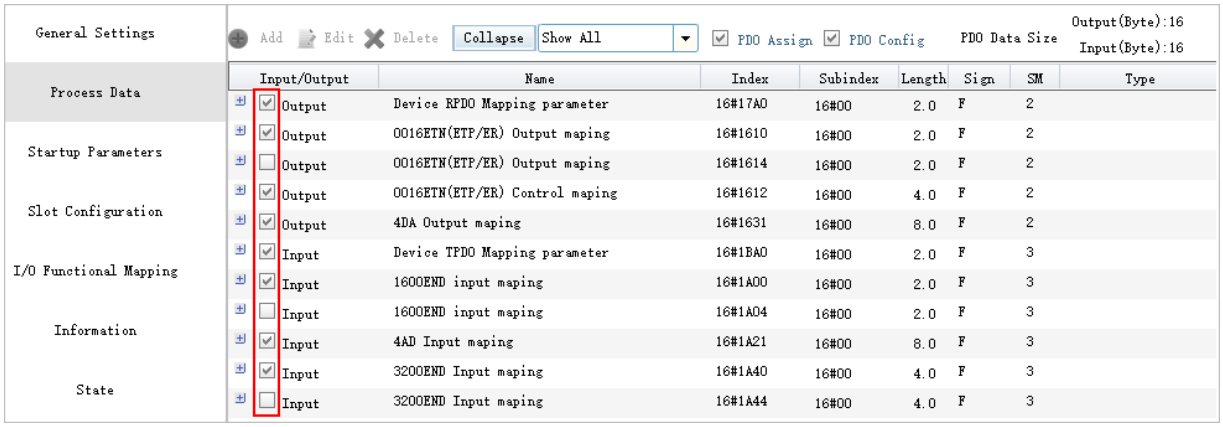
Select a PDO in the first group at ①. Click "Add" at ② and select 6060 at ③. Then, click "OK", as shown in the following figure.



When a slave station has multiple groups of PDOs, such PDO groups are exclusive, such as IS660N. You can select one group each time.

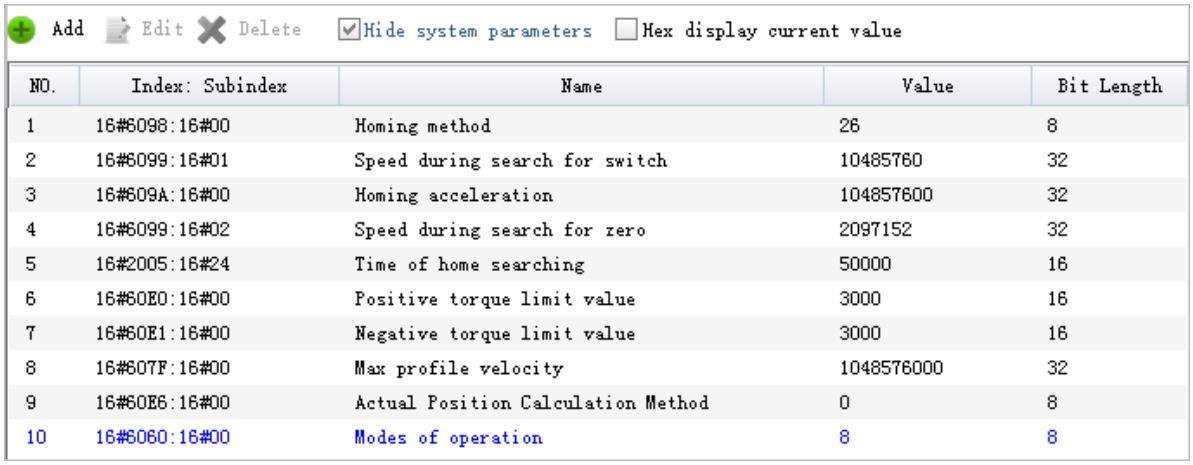


Such mutually exclusive relationship varies with the slave station. For example, you can select multiple PDO groups for GL10-RTU-ECTA.



The master station downloads the PDO configuration relationships to EtherCAT slave stations in the form of startup parameters through PDO allocation and PDO mapping.

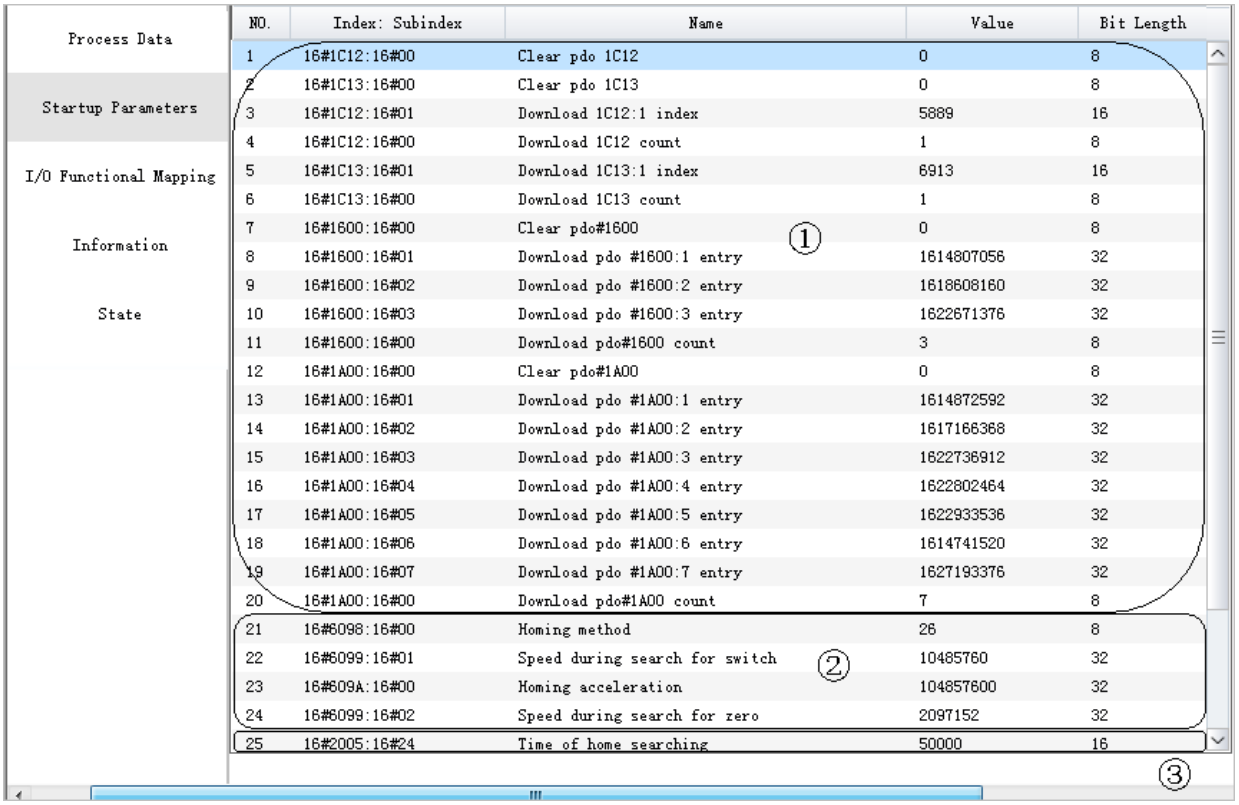
"PDO Assign" is used to download the selected PDO group number to the slave station, while "PDO Config" is used to download editable PDOs of a group to the slave station. When PDOs are modified but "PDO Assign" and "PDO Config" are not selected, the slave station may not be started. The configurations can be viewed in the startup parameter list.



11.3.3 Startup Parameters

Startup parameters are used to write slave station PDO configuration, factory settings, and parameters specified by some protocols (such as the 402 protocol) to the slave station through writing SDO when the slave station is in the PreOP state.

Take IS620N as an example:



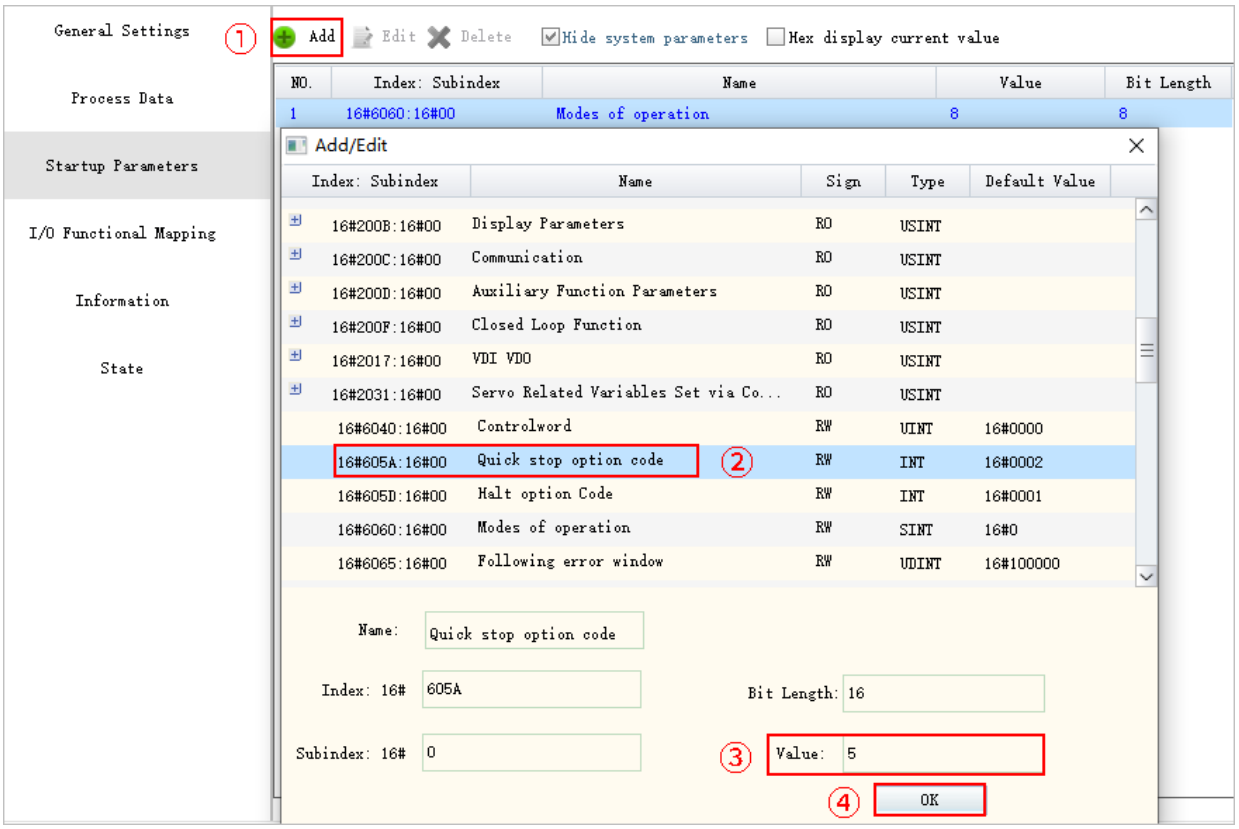
① PDO configuration parameters

② 402 protocol parameters

③ Factory parameters

On this page, you can add startup parameters as required. For example, you can add the object dictionary 0x605a and modify its value to 5 as follows.

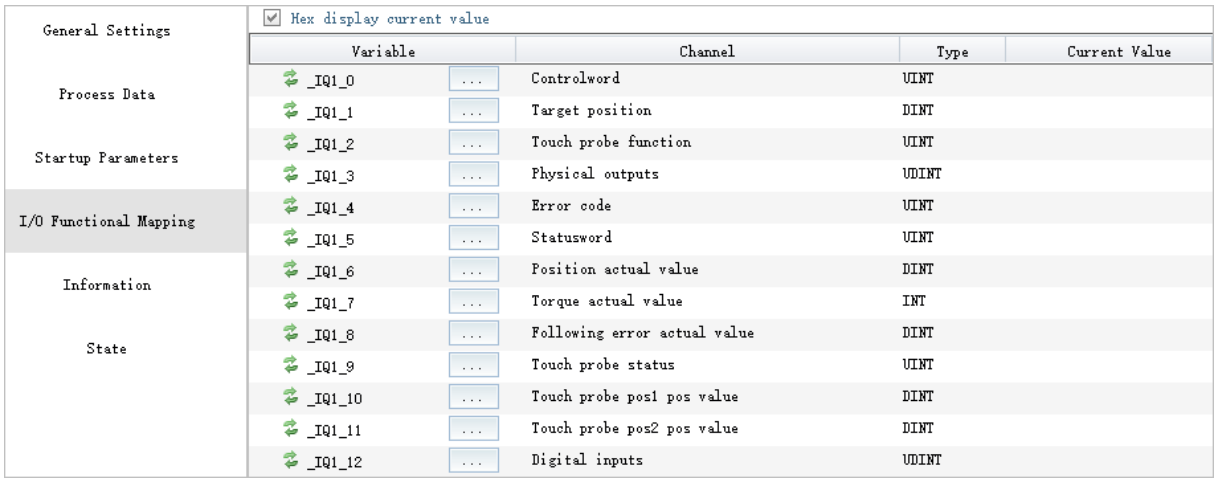
Click "Add" at ①, select "605A", modify the value to 5, and click "OK". Then the object dictionary 0x605a is added and its value is changed to 5.



11.3.4 IO Functional Mapping

You can control an EtherCAT slave station module by controlling the operation variables only after the PDO data is connected to the PLC.

The "IO Functional Mapping" page is as follows.



Each time a slave station is added, a group of internal variables are automatically created and connected to the PDO of the slave station, such as IQ2_0.

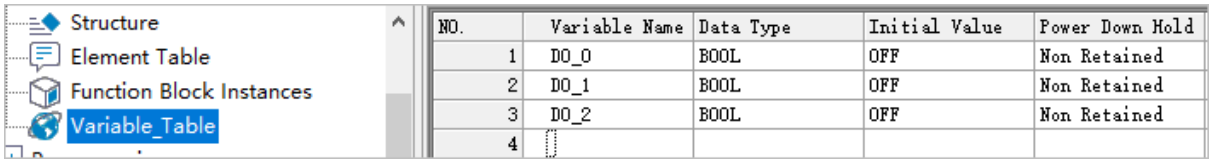
NOTE

- The automatically generated variables change when the module position is changed or any PDO is added, deleted, or modified.
- If a slave is associated with a motion control axis, such as SV660N, these variables can only be controlled through axis instructions.

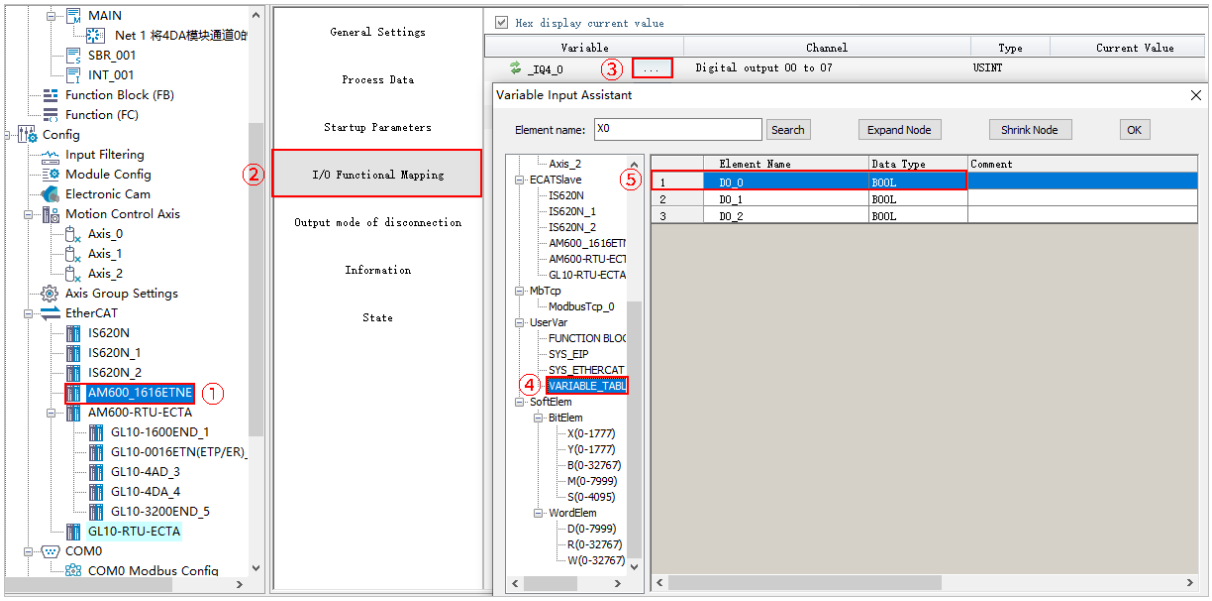
Associated variable

To modify an associated variable, perform the following operations (taking the GR10-1616ETNE module as an example).

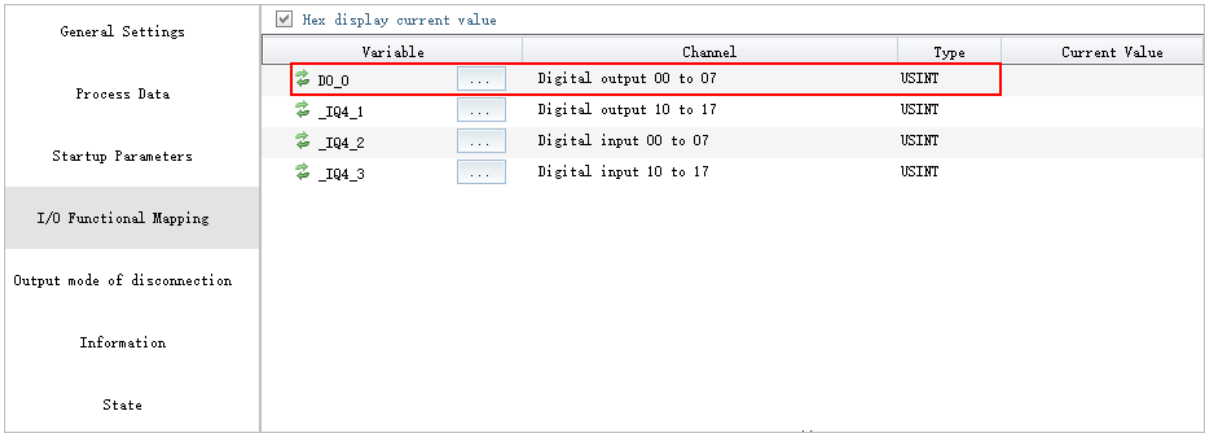
1. Open the variable table and add a variable.



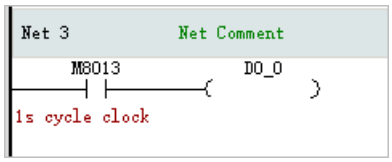
2. Associate the variable on the "IO Functional Mapping" page.
   Access slave device ①, select "I/O Functional Mapping", click "...", select the variables as required, and click "OK", as shown in the following figure.



3. The following page is displayed.



4. The PLC program controls DO_0 to flash once every second.



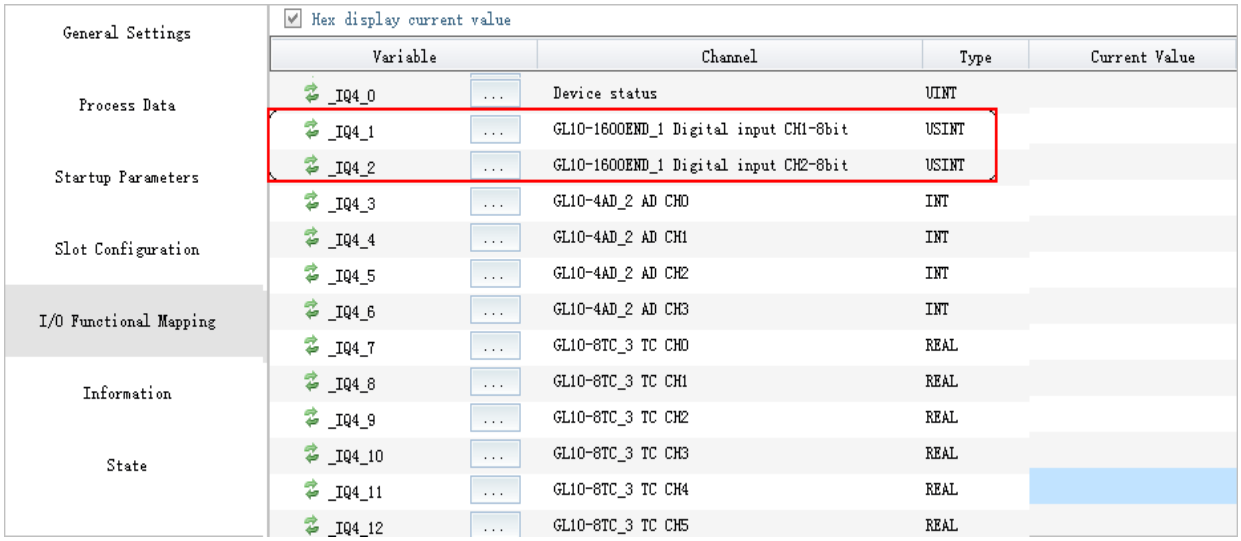
Mapping Rules

Data types supported by customized variables are BOOL, BYTE, INT, DINT, and REAL. PDO variables of EtherCAT slave stations support more data types. The following table lists the mapping rules available on the “IO Functional Mapping” page.

EtherCAT slave station type	Bit length	Mapping rules
BOOL	1	BOOL
BYTE	8	• INT: Low-order 8 bits are valid. High-order 8 bits are reserved. • BOOL[8]: An 8-bit BOOL-type array is used. • BYTE: 8-bit BYTE-type mapping is used.
SINT	8	• INT: Low-order 8 bits are valid. High-order 8 bits are reserved. • BOOL[8]: An 8-bit BOOL-type array is used.

EtherCAT slave station type	Bit length	Mapping rules
		• BYTE: 8-bit BYTE-type mapping is used.
USINT	8	• INT: Low-order 8 bits are valid. High-order 8 bits are reserved. • BOOL[8]: An 8-bit BOOL-type array is used. • BYTE: 8-bit BYTE-type mapping is used.
BITARR8	8	• INT: Low-order 8 bits are valid. High-order 8 bits are reserved. • BOOL[8]: An 8-bit BOOL-type array is used. • BYTE: 8-bit BYTE-type mapping is used.
BIT8	8	• INT: Low-order 8 bits are valid. High-order 8 bits are reserved. • BOOL[8]: An 8-bit BOOL-type array is used. • BYTE: 8-bit BYTE-type mapping is used.
INT	16	INTBOOL[16]: A 16-bit BOOL-type array is used.
UINT	16	INTBOOL[16]: A 16-bit BOOL-type array is used.
WORD	16	INT BOOL[16]: A 16-bit BOOL-type array is used.
BITARR16	16	INT BOOL[16]: A 16-bit BOOL-type array is used.
DINT	32	DINTBOOL[32]: A 32-bit BOOL-type array is used.
UDINT	32	DINTBOOL[32]: A 32-bit BOOL-type array is used.
DWORD	32	DINT BOOL[32]: A 32-bit BOOL-type array is used.
BITARR32	32	DINT BOOL[32]: A 32-bit BOOL-type array is used.
REAL	32	REAL

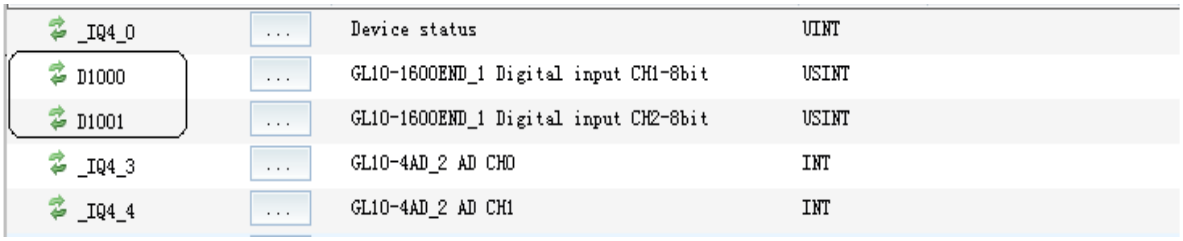
The following figure shows an example with the data type of USINT.



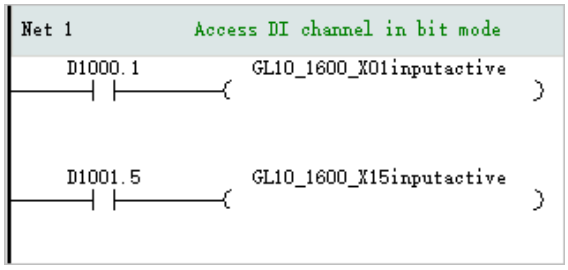
To detect the DI channel of the GL10-1600END module, you must allocate a variable to both GL10-1600END_1 Digital input- 8bit and GL10-1600END_1 Digital input CH2-8bit (you can also use the default variables IQ4_1 and IQ4_2, but they are not easy to expand and maintain). The configuration can be displayed in the following three ways.

Solution 1: Associate the D element.

1. Create a mapping relationship. (That is, create a relationship between the two collections so that each input element is associated with one exclusive output element.)



2. Call it in the program.

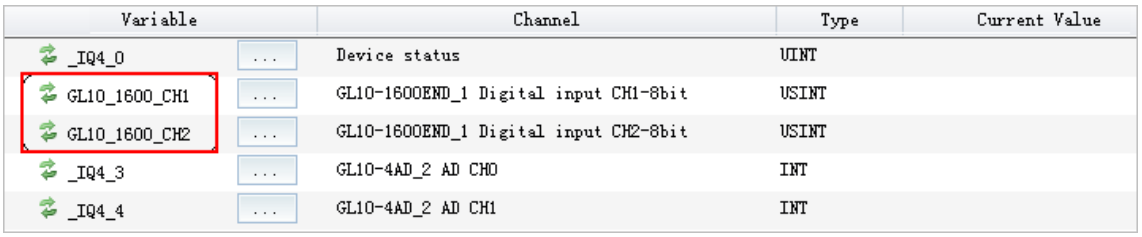


Solution 2: Associate customized variable of the INT type.

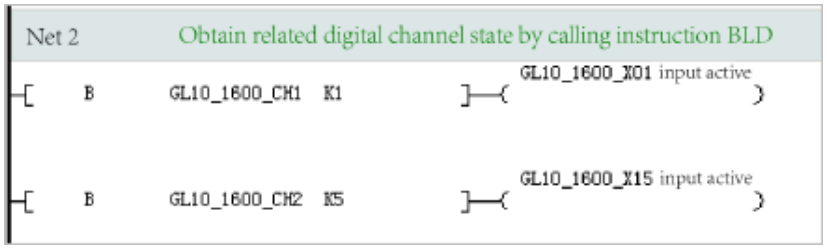
1. Create a global variable.

NO.	Variable Name	Data Type	Initial Value	Power Down Hold	Network Pubilc	Comment	Element Addr.	Length
1	GL10_1600_CH1	INT	0	Non Retained	Private			nBitLen:16
2	GL10_1600_CH2	INT	0	Non Retained	Private			nBitLen:16

2. Create a mapping relationship.



3. Call it in the program.



Solution 3: Associate an array of the BOOL type.

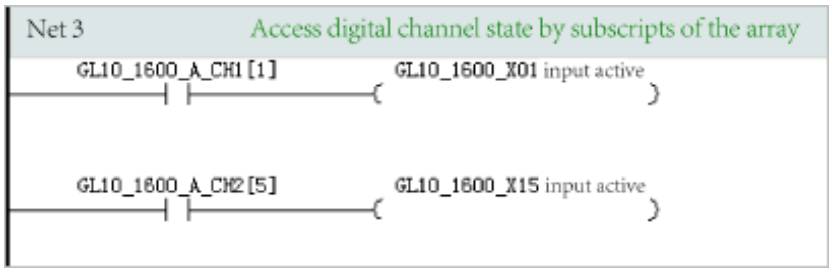
1. Create a global variable array.

NO.	Variable Name	Data Type	Initial Value	Power Down Hold	Network Pubilc	Comment	Element Addr.	Length
1	GL10_1600_CH1	INT	0	Non Retained	Private			nBitLen:16
2	GL10_1600_CH2	INT	0	Non Retained	Private			nBitLen:16
3	GL10_1600_A_CH1	BOOL[8]	...	Non Retained	Private			nBitLen:8
12	GL10_1600_A_CH2	BOOL[8]	...	Non Retained	Private			nBitLen:8

2. Create a mapping relationship.

Variable	Channel	Type	Current Value
_IQ4_0	Device status	UINT	
GL10_1600_A_CH1	GL10-1600END_1 Digital input CH1-8bit	USINT	
GL10_1600_A_CH2	GL10-1600END_1 Digital input CH2-8bit	USINT	
_IQ4_3	GL10-4AD_2 AD CH0	INT	
_IQ4_4	GL10-4AD_2 AD CH1	INT	

3. Call it in the program.



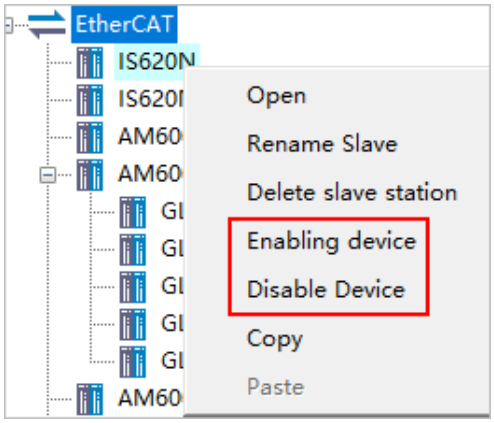
11.3.5 Device Start/Stop, Disable, and Enable

For AutoShop 4.0.0.0 matching PCB software of a version earlier than 3.0.0.0, you cannot start or stop a single slave station, but can only start or stop the entire EtherCAT bus.

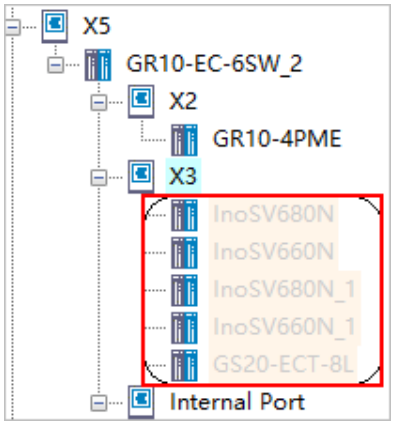
For AutoShop 4.0.0.0 matching PCB software of version 3.0.0.0 or later, you can select “Auto restart slave” on the EtherCAT master station configuration page to automatically restart a slave station.

When the number of configured slave stations is greater than the number of connected ones, you can disable unavailable slave stations in the configuration.

To enable or disable a slave station, right-click the target slave station, and then select “Disable Device” or “Enabling device”.



When the network contains branch modules, if a branch port is disabled, all slave stations mounted to this port are disabled.



CAUTION

A disabled slave must be removed from the actual physical link. Otherwise, the slave will affect the startup of all subsequent slaves.

11.3.6 Disabling a Slave Through Instruction

You can disable EtherCAT slaves during programming by using the ETC_RestartMaster instruction and relevant system variables.

System variables

- The following table lists system variables used to enable or disable specified EtherCAT slaves.

Name	Unit	Description	Retentive upon Power Failure
bDisableEnable	BOOL	Disable/Enable OFF: Enabled ON: Disabled	No
wDisableState	INT	Configuration status 0: Reserved 1: Enabled 2: Disabled	No

- The following table lists system variables of EtherCAT masters used to enable or disable the entire EtherCAT bus.

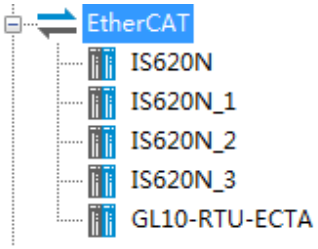
Name	Unit	Description	Retentive upon Power Failure
bDisableMaster	BOOL	Disable/Enable OFF: Enabled ON: Disabled	No
iDisableState	INT	Configuration status 0: Reserved 1: Enabled 2: Disabled	No

Usage

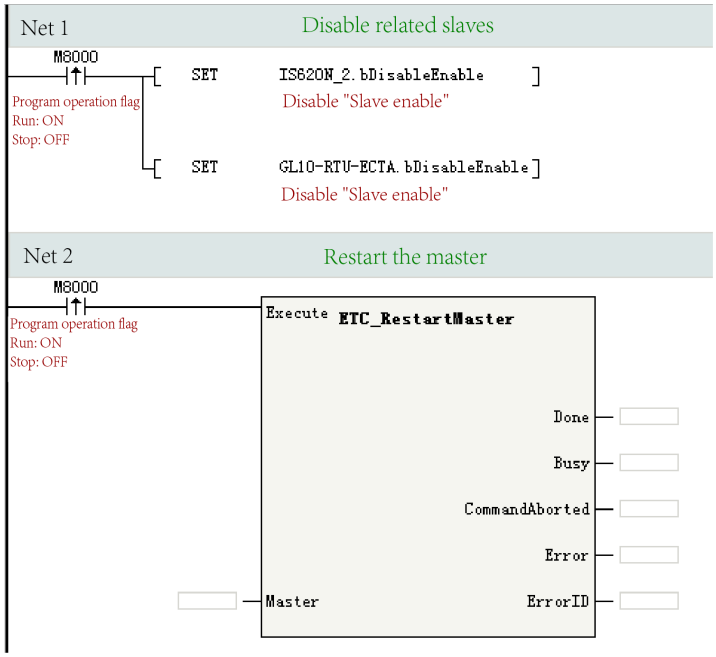
1. After power-on, the PLC initializes the bDisableEnable variable based on the background configuration, updates the configuration list based on the value of the bDisableEnable variable, starts the master, and then writes the disabling status of the slave to the wDisableState variable.
2. Wait until the PLC parses the program configuration.
3. Use the PLC program to set the value of the bDisableEnable variable.
4. Use the ETC_RestartMaster instruction to restart the EtherCAT master.
5. After the master is restarted, use the bDisableEnable variable to update the configuration list and write the disabling status of the slave to wDisableState.

Example

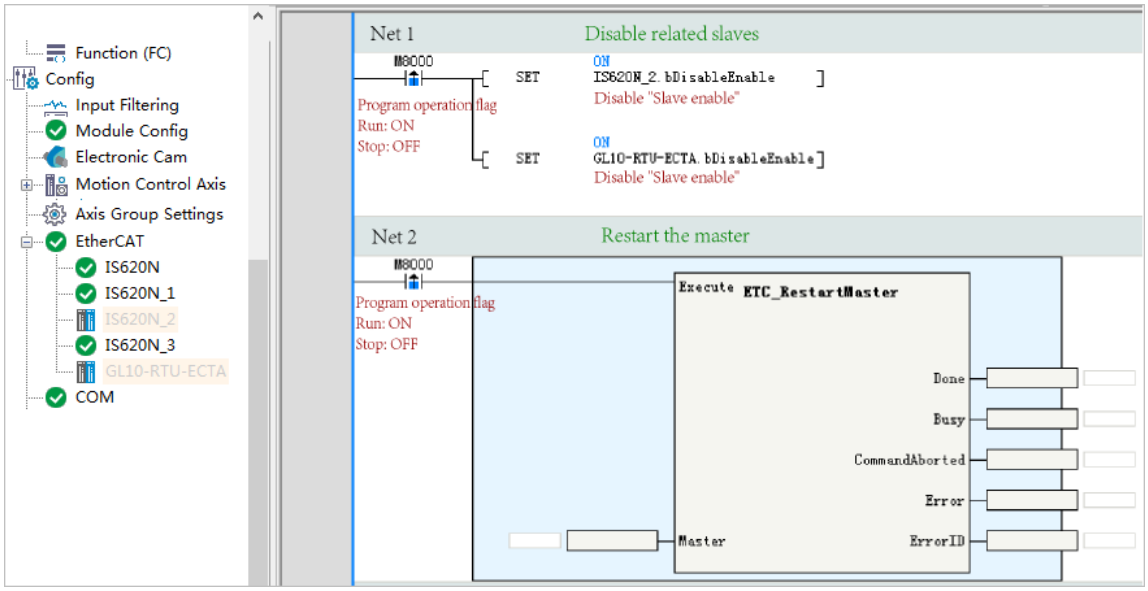
1. Create a configuration by using the background software, enable the EtherCAT communication, and configure four IS620N slaves and one AM600-RTU-ECTA slave.



2. Set the system variables _DisableEnable and bDisableEnable to disable the slaves IS620N_2 and AM600-RTU-ECTA, and then use the ETC_RestartMaster instruction to restart the master.



3. Download the project to the controller. The slaves IS620N_2 and AM600-RTU-ECTA are disabled after start.



11.3.7 List of System Variables

The following table lists system variables of EtherCAT slaves.

System Variable	Data Type	Function Description
bSlaveRunState	BOOL	Running status of slave The value is ON when the slave is in the OP state; otherwise, the value is OFF.
bSetAliasState	BOOL	Set the alias status (for the use of the background only) ON: Busy OFF: Idle or setting completed
bSetAliasError	BOOL	Failed to set the alias status (for the use of the background only) ON: Failed to set the alias status

System Variable	Data Type	Function Description
		OFF: No fault
bMatchState	BOOL	Slave type match status ON: The type of the slave at this position in actual configuration is different from that of the slave at this position in the background configuration. OFF: Slave types consistent
bConfigError	BOOL	Slave configuration status ON: The slave configuration at this position is incorrect OFF: The slave configuration at this position is correct
bSetAlias	BOOL	Set the station alias (for the use of the background only) The value of wTarAlias is written to the slave at the rising edge of the variable.
bDisableEnable	BOOL	Disable/Enable OFF: Enabled ON: Disabled
wALState	INT	Status of the EtherCAT slave state machine 1: INIT 2: PreOP 4: SafeOP 8: OP
wAlCode	INT	Fault code of failure to convert the slave state machine. For details, see the slave guide.
wActAlias	INT	Actual alias of the slave. Initialization is performed once upon power-on, and the modification does not take effect.
wTarAlias	INT	Station alias to be written (for the use of the background only)
wStationAddress	INT	Sequential address of the slave. Initialization is performed once upon power-on, and the modification does not take effect.
wSlaveRingPos	INT	Configuration address, starting from 0
wSDOErrcnt	INT	Number of startup parameter configuration errors
dCfgErrorCode	DINT	Configuration error code. For the code definition, see Fault Codes.
wDisableState	INT	Configuration status 0: Reserved

System Variable	Data Type	Function Description
		1: Enabled 2: Disabled

11.4 Fault Diagnosis

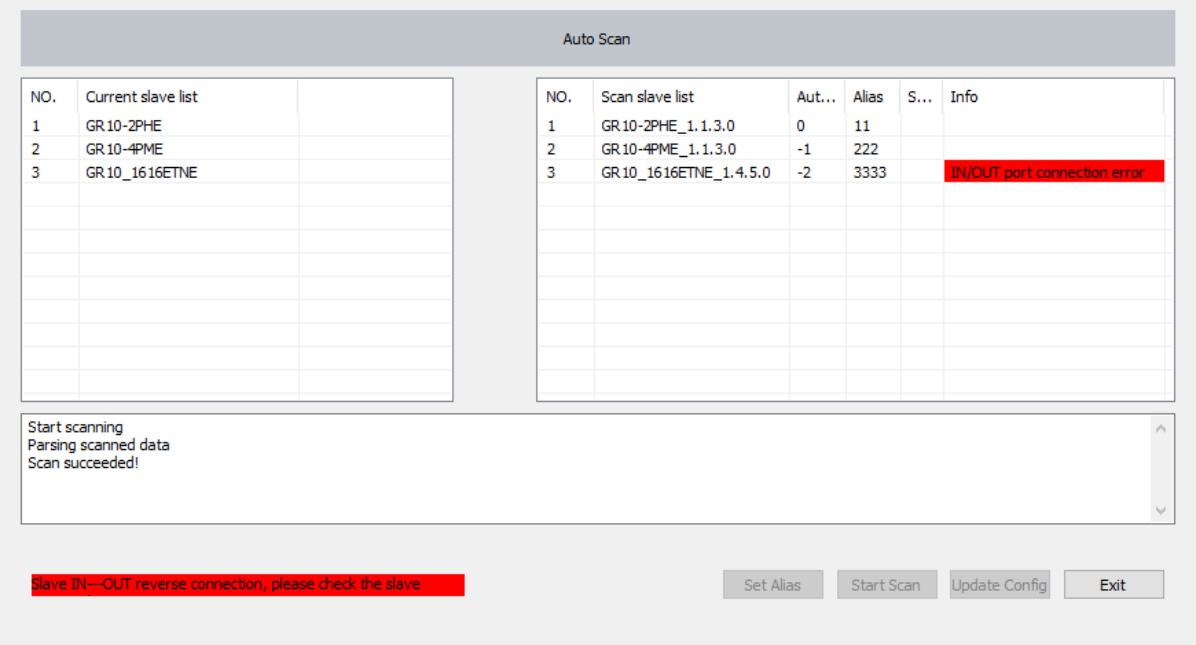
11.4.1 Fault Diagnosis for IN-OUT Reverse Connection

Preconditions

- Applicable model: Easy523
- Firmware version: V6.4.0.0 and later
- Software version: AutoShop V4.10.2.0 and later

Symptoms

During scanning, "IN/OUT port connection error" is displayed at the slave in the information section, and "slave IN-OUT reverse connection" is displayed in the left-down corner on the "Auto Scan" page, as shown in the following figure.

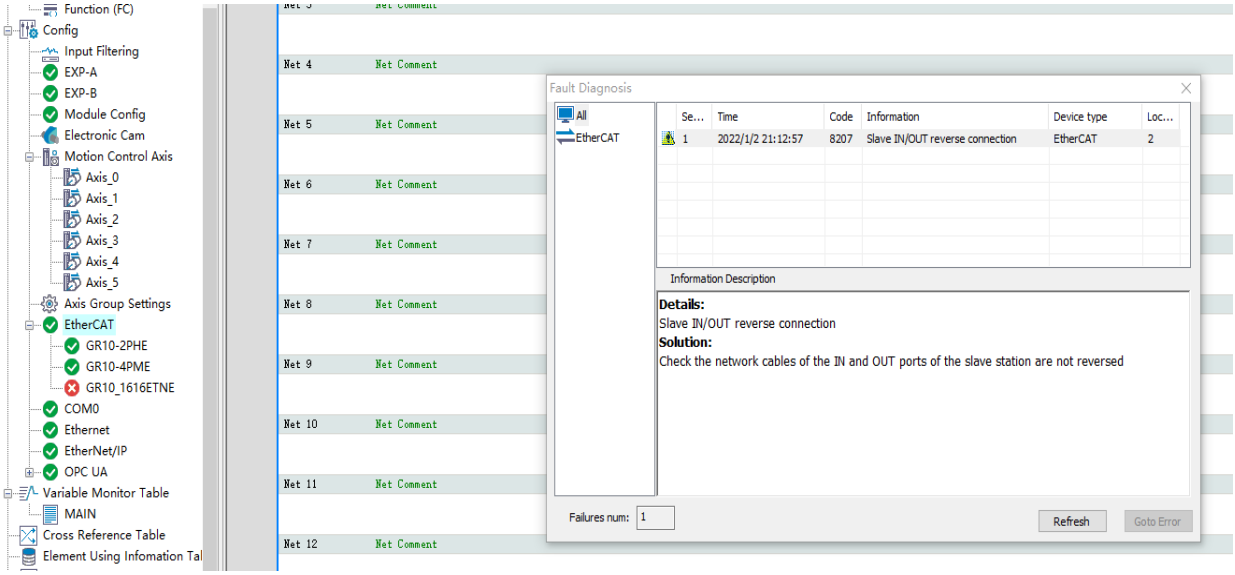


Possible cause

During scanning, the IN port and OUT port of the slave are reversed.

Solution

Connect the IN and OUT ports of the slave correctly according to the prompt. Then, the EtherCAT bus can run normally, as shown in the following figure.



11.4.2 Fault Diagnosis for Continuous Frame Loss

Preconditions

- Applicable model: Easy523
- Firmware version: V6.4.0.0 and later
- Software version: AutoShop V4.10.2.0 and later

Symptoms

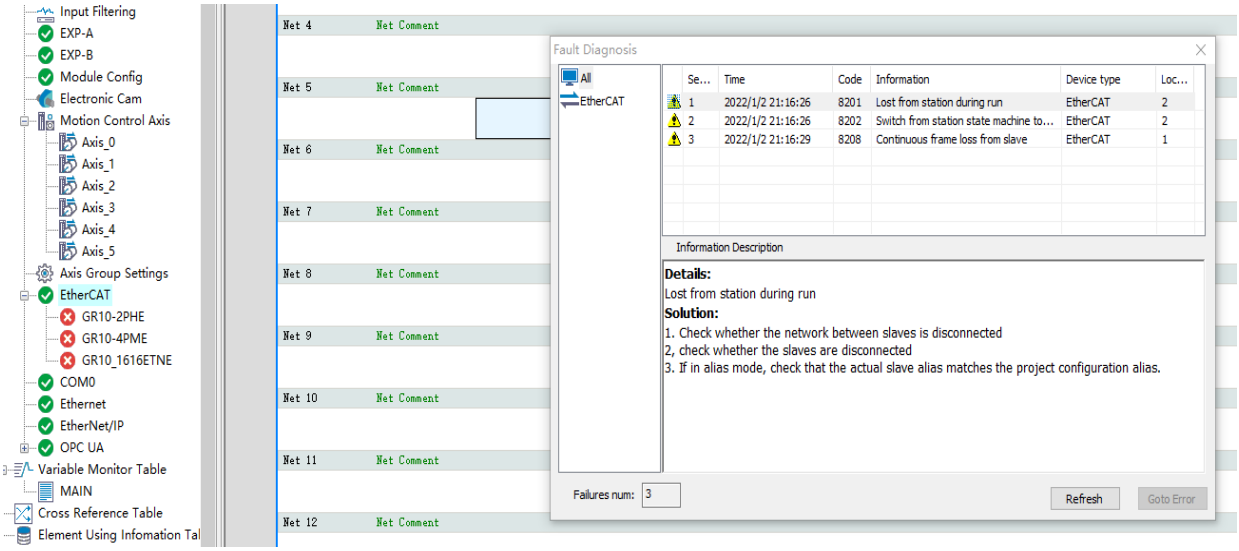
If continuous frame loss occurs in any link during scanning, the system displays the port with continuous frame loss in the information bar of the corresponding slave, as shown in the following figure.

Possible cause

Continuous frame loss occurs in the configuration link.

Solution

Connect the corresponding ports and network of the slave correctly according to the prompt. Then, the EtherCAT bus can run normally, as shown in the following figure.



During running, if any configuration link has continuous frame loss, the slave with continuous frame loss will be displayed in the fault status column.

In this case, connect the corresponding ports and network of the slave correctly according to the prompt. Then, the EtherCAT bus can run normally.

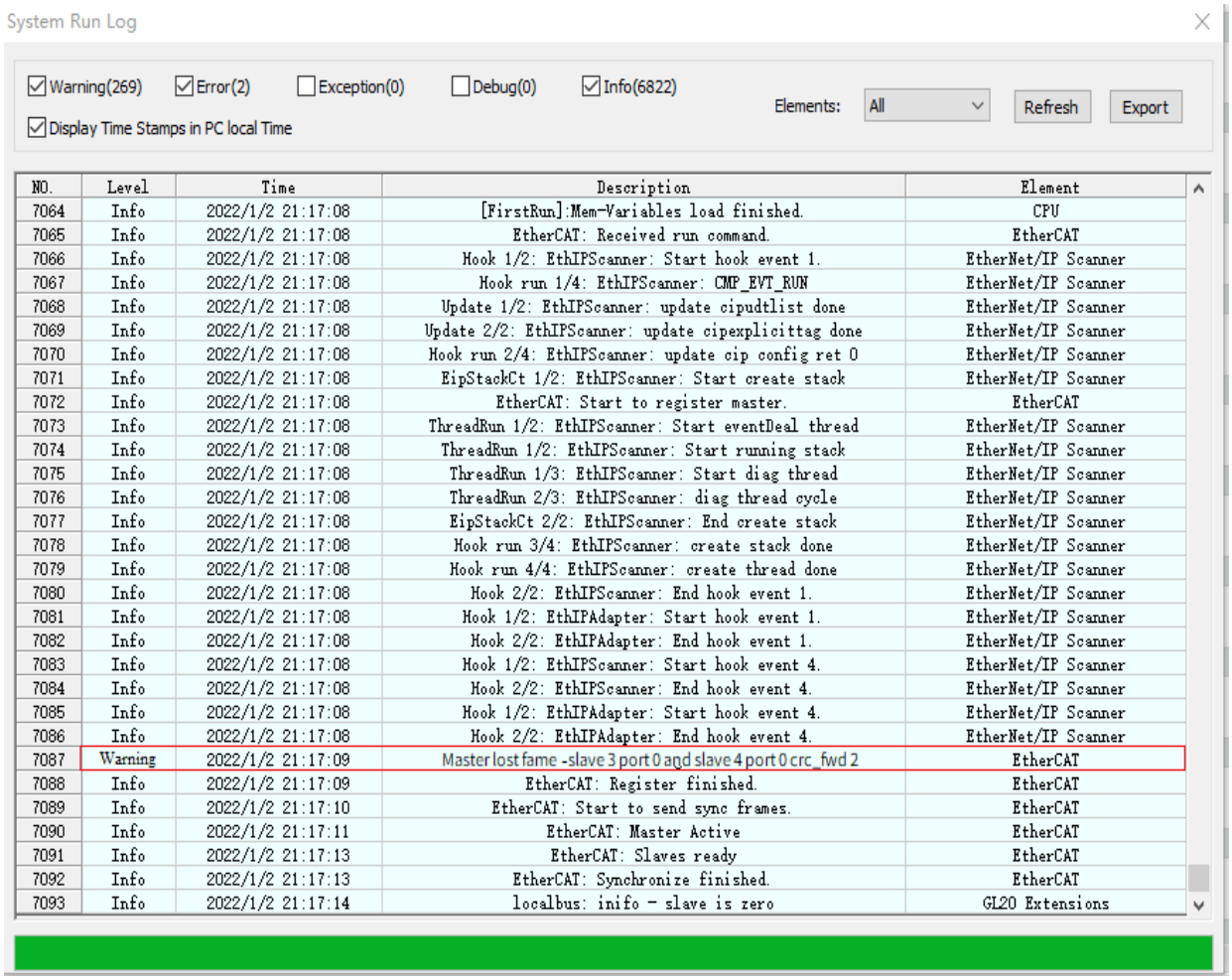
11.4.3 Fault Diagnosis for Occasional Frame Loss

Preconditions

- Applicable model: Easy523
- Firmware version: V6.4.0.0 and later
- Software version: AutoShop V4.10.2.0 and later

Symptoms

The operation logs contain information about slaves having occasional frame loss, port, and the number of lost frames, as shown in the following figure.



NO.	Level	Time	Description	Element
7064	Info	2022/1/2 21:17:08	[FirstRun]:Mem-Variables load finished.	CPU
7065	Info	2022/1/2 21:17:08	EtherCAT: Received run command.	EtherCAT
7066	Info	2022/1/2 21:17:08	Hook 1/2: EthIPScanner: Start hook event 1.	EtherNet/IP Scanner
7067	Info	2022/1/2 21:17:08	Hook run 1/4: EthIPScanner: CMP_EVT_RUN	EtherNet/IP Scanner
7068	Info	2022/1/2 21:17:08	Update 1/2: EthIPScanner: update cipudtlist done	EtherNet/IP Scanner
7069	Info	2022/1/2 21:17:08	Update 2/2: EthIPScanner: update cipexplicittag done	EtherNet/IP Scanner
7070	Info	2022/1/2 21:17:08	Hook run 2/4: EthIPScanner: update cip config ret 0	EtherNet/IP Scanner
7071	Info	2022/1/2 21:17:08	EipStackCt 1/2: EthIPScanner: Start create stack	EtherNet/IP Scanner
7072	Info	2022/1/2 21:17:08	EtherCAT: Start to register master.	EtherCAT
7073	Info	2022/1/2 21:17:08	ThreadRun 1/2: EthIPScanner: Start eventDeal thread	EtherNet/IP Scanner
7074	Info	2022/1/2 21:17:08	ThreadRun 1/2: EthIPScanner: Start running stack	EtherNet/IP Scanner
7075	Info	2022/1/2 21:17:08	ThreadRun 1/3: EthIPScanner: Start diag thread	EtherNet/IP Scanner
7076	Info	2022/1/2 21:17:08	ThreadRun 2/3: EthIPScanner: diag thread cycle	EtherNet/IP Scanner
7077	Info	2022/1/2 21:17:08	EipStackCt 2/2: EthIPScanner: End create stack	EtherNet/IP Scanner
7078	Info	2022/1/2 21:17:08	Hook run 3/4: EthIPScanner: create stack done	EtherNet/IP Scanner
7079	Info	2022/1/2 21:17:08	Hook run 4/4: EthIPScanner: create thread done	EtherNet/IP Scanner
7080	Info	2022/1/2 21:17:08	Hook 2/2: EthIPScanner: End hook event 1.	EtherNet/IP Scanner
7081	Info	2022/1/2 21:17:08	Hook 1/2: EthIPAdapter: Start hook event 1.	EtherNet/IP Scanner
7082	Info	2022/1/2 21:17:08	Hook 2/2: EthIPAdapter: End hook event 1.	EtherNet/IP Scanner
7083	Info	2022/1/2 21:17:08	Hook 1/2: EthIPScanner: Start hook event 4.	EtherNet/IP Scanner
7084	Info	2022/1/2 21:17:08	Hook 2/2: EthIPScanner: End hook event 4.	EtherNet/IP Scanner
7085	Info	2022/1/2 21:17:08	Hook 1/2: EthIPAdapter: Start hook event 4.	EtherNet/IP Scanner
7086	Info	2022/1/2 21:17:08	Hook 2/2: EthIPAdapter: End hook event 4.	EtherNet/IP Scanner
7087	Warning	2022/1/2 21:17:09	Master lost fame -slave 3 port 0 and slave 4 port 0 crc_fwd 2	EtherCAT
7088	Info	2022/1/2 21:17:09	EtherCAT: Register finished.	EtherCAT
7089	Info	2022/1/2 21:17:10	EtherCAT: Start to send sync frames.	EtherCAT
7090	Info	2022/1/2 21:17:11	EtherCAT: Master Active	EtherCAT
7091	Info	2022/1/2 21:17:13	EtherCAT: Slaves ready	EtherCAT
7092	Info	2022/1/2 21:17:13	EtherCAT: Synchronize finished.	EtherCAT
7093	Info	2022/1/2 21:17:14	localbus: inifo - slave is zero	GL20 Extensions

Possible cause

Occasional frame loss occurs in the configuration link.

Solution

As shown in the preceding figure, occasional frame loss occurs between port 0 of slave 3 and port 0 of slave 4, and two frames are lost. In this case, check the network and port connection between port 0 of slave 3 and port 0 of slave 4.

11.5 Troubleshooting

11.5.1 Fault Indication

The BF indicator indicates the fault status of the EtherCAT bus. The following table lists the statuses and solutions.

LED Indicator	Description	Solution
OFF	No fault	/
Flashing	The EtherCAT bus is abnormal.	Troubleshoot the problem based on the fault code displayed. For details, see Fault Codes.
ON	Failed to request for the master station.	Troubleshoot the problem based on the fault code displayed. For details, see Fault Codes.

You can obtain the code of an EtherCAT instruction fault based on the ErrorID in the instruction.

You can view the EtherCAT bus faults on the "Fault Diagnosis" page.

11.5.2 Troubleshooting

When an EtherCAT bus slave fails, check whether this slave is supported by the current software version.

- The H5U firmware of 6.1.0.1 and later versions support the GR10-EC-6SW branch module and simple slaves (such as Inovance GR20T-ECT-1616EMN and Inovance GR20T-ECT-0808EMN).
- The Easy firmware of 6.3.0.0 and later versions support the GR10-EC-6SW branch module and simple slaves (such as Inovance GR20T-ECT-1616EMN and Inovance GR20T-ECT-0808EMN).

11.5.3 Fault Codes

Fault code	Cause	Solution
8001	Failed to request the master.	• The board version does not match the software version (upgrade the software to the correct version). • Restart the PLC.
8002	Failed to obtain slave configuration parameters.	Check whether the board software and software tool versions match.
8003	Master station startup timed out.	Check the network connection.

Fault code	Cause	Solution
8004	Failed to request the master.	• Restart the PLC. • An error occurred while loading ECAT. Upgrade the firmware to the correct version.
8005	EtherCAT configuration data format inconsistent	Upgrade the software or board to the correct version.
8006	EtherCAT initialization failure	Restart the PLC.
...	-	-
8200	Failed to write startup parameters.	• Check whether there is an object dictionary that is not supported by the slave in the startup parameter list. • Check whether the value of the object dictionary is out of range.
8201	The slave is lost during operation.	• Check whether the network with the slave station is disconnected. • Check whether the slave station is powered off.
8202	• The slave station enters a non-OP state during running.	• Check whether the network with the slave station is disconnected. • Check whether the slave station is powered off.
8203	Reserved	-
8204	The slave type is incorrect.	• Check whether the network cable is reversely connected. • Check whether the connected device matches the configuration.
8205	PDO address error	• Check whether the memory runs out. • Check whether the background and board software versions match. • Check whether the background and board software versions match. • Restart the drive.
8206	Incorrect PDO length	Check whether the background and board software versions match.

Fault code	Cause	Solution
8301	Failed to switch to INIT state.	Check whether the slave station machine supports state transition.
8302	Failed to switch to PerOP state.	Check whether the slave supports the CoE protocol.
8304	Failed to switch to SafeOP state.	Check whether the PDO communication configuration is correct.
8308	Failed to switch to OP state.	• Check the network communication quality. • Check whether the EtherCAT task cycle is appropriate.
8310	FMMU unit configuration error	Check whether the slave supports the FMMU unit.
8311	Email configuration error	Check whether the slave supports the SM unit.
8400	ECTA configuration error	Check whether the configured expansion module is consistent with the actually connected expansion module.
8401	ECTA hardware error	• Check whether the connection between the ECTA and the expansion module is loose. • Replace the ECTA.
8402	Incorrect expansion module mounted to ECTA	Check the fault type of the expansion module based on the ECTA guide.
8403	ECTA expansion module slot error	• Locate the expansion module with the ERR indicator on. • Check the ECAT-related error message (error location and fault code) in the log information • Determine the fault type based on description of the diagnosis object dictionary of the expansion module in the ECTA application guide and eliminate the fault.
1280	The master is not found.	Check whether EtherCAT bus communication is enabled.
1281	The slave is not found.	Check whether the slave exists in configuration.
1282	The length of the SDO to read or write to is 0 or greater than 4.	Check whether the length specified in the SDO read or write function is correct.
1283	The master is not found.	Check whether the master configuration parameters are correct.
1284	• Reading or writing fails.	• Check whether the SDO operation is allowed by the slave state machine. • Check whether the SDO to read or write to exists.

Fault code	Cause	Solution
	• The SDO read/write timed out. • SDO does not exist. • SDO read/write is not supported due to the status of the slave station. • The SDO length to be read or written is incorrect.	• Check whether the length of the SDO to read or write to is correct.
1285	Failed to request the memory.	• Check whether the PLC memory runs out. • Contact the manufacturer.
1286	The master is in Stopping state.	Do not call this instruction when the master is in Stopping state.

When an EtherCAT bus fault occurs, the fault status indicator in the lower-right corner of AutoShop turns to yellow. If you double-click this area, the “Fault Diagnosis” dialog box is displayed, in which you can view the configuration of the failed slave station, the fault code, and the fault details.

12 Ethernet-IP Communication

12.1 Overview

CAUTION

- This function is supported by H5U series with AutoShop of V4.4.0.5 and later versions.
- This function is supported by Easy series with firmware of V5.66.0.0 and later versions and AutoShop of V4.8.0.0 and later versions.

The open connected end is the EtherNet/IP master station, and the opened end is the EtherNet/IP slave station, as shown in the following figure.

EtherNet/IP communication specifications:

- H5U and Easy320/Easy52X series PLCs support one EtherNet/IP master station.
- The minimum requested packet interval (RPI) is 5 ms.
- The maximum data volume of single connection communication is 1444 bytes for Easy, and 500 bytes for H5U.
- A maximum of 32 connections (consumer tags + server tags + slave connections) are supported.
- A maximum of 32 tags (producer tags + server tags) can be created.

NOTE

- When EtherCAT and EtherNet/IP networks both exist in a networking project, the real-time communication performance of EtherNet/IP network is reduced because EtherCAT has the highest priority by default.
- For protocol details, see the official standard documents EIP-CIP-V1-1.0 and EIP-CIP-V2-1.0.

12.2 Technical Specifications

12.2.1 Ethernet-IP Transmission Specifications

The following table lists the transmission specifications.

Item	Technical Specifications	
	10BASE-T	100BASE-TX
Transmission rate	10 Mbps	100 Mbps
Transmission media	STP or UTP above Cat3[1]	STP or UTP above Cat5[1]
Max. cable length[2]	100 m	100 m

NOTE

Meet the standard IEEE802.3.

- [1]: STP: shielded twisted pair. UTP: unshielded twisted pair.
- [2]: The maximum cable length is the distance between an EtherNet/IP unit and a network device.

12.2.2 Ethernet-IP Communication Specifications

Item				H5U series	Easy32x series	Easy52x series
CIP service	Implicit message communication	Number of I/O connections at the originator[1]		32	32	32
		Number of I/O connections at the target end[2]		32	32	32
		RPI (communication cycle)		5 ms to 50,000 ms		
		Bandwidth allowable for implicit (I/O) messaging	(@4 bytes)	12800 pps[3]	12800 pps[3]	12800 pps[3]
			(@250 bytes)	6400 pps[3]	12800 pps[3]	12800 pps[3]
			(@500 bytes)	3200 pps[3]	12800 pps[3]	12800 pps[3]
			(@1400 bytes)	-	8000 pps[3]	12800 pps[3]
		Maximum data size per connection[4]		500 bytes	1444 bytes	1444 bytes
		Multicast filter[5]		Supported (IGMP client function)		
	Explicit messaging	Number of Class3 tags at the originator[6]		32	32	32
		Number of Class-3 tags at the target end[6]		32	32	32
		Number of UCMM tags at the initiator[6]		Number of simultaneous executions: 32	Number of simultaneous executions: 32	Number of simultaneous executions: 32
		Number of UCMM tags at the target end[6]		Number of simultaneous executions: 32	Number of simultaneous executions: 32	Number of simultaneous executions: 32

Item		H5U series	Easy32x series	Easy52x series
	Number of connections[7]	48	64	

[1] I/O connections at the originator include:

- Consumer tags: Connections requested by the originator with the name of the producer tag as the connection path.
- Originator generic I/O connections: Connections requested by the originator with the instance ID of the generic I/O connections of the target end as the connection path.

[2] I/O connections at the target end include:

- Consumer tags: Responses from the target end to the connection requests with the name of the producer tag as the connection path.
- Target end generic I/O connections: Responses from the target end to connection requests with the instance ID of the generic I/O connections of the target end as the connection path.

[3] pps refers to Packet Per Second. pps is the unit of network throughput rate. Here, it means the sum of the number of grouping packets sent and received that can be processed in one second. It is calculated according to the following formula: Communication bandwidth pps = 1000 ms/RPI x Number of connections x 2.

[4] Data simultaneity within a connection is guaranteed. The device used supports Large Forward Open (CIP option specification) when the data size is greater than 509 bytes.

[5] The EtherNet/IP unit supports the IGMP client function, so the use of an Ethernet switch that supports IGMP Snooping allows filtering out unattended multicast packets.

[6] The number of tags is as follows:

- Initiator tags include the consumer tags, initiator generic I/O connections, Class-3 tags at the originator, and UCMM tags at the initiator. The maximum number is 32.
- Target end tags include the producer tags, target end generic I/O connections, Class-3 tags at the target end, and UCMM tags at the target end. The maximum number is 32. The maximum number of target end generic I/O connections is 16.

[7] Total number of connections = Number of I/O connections at the originator + Number of I/O connections at the target end + Number of Class-3 tags at the originator + Number of Class-3 tags at the target end

12.2.3 Ethernet-IP Quick Reference Table

The quick reference table of EtherNet/IP solutions lists typical values generated based on the bandwidth allowable for implicit (I/O) message communication provided in EtherNet/IP Communication Specifications. It strictly adheres to the relevant communication bandwidth regulations and is designed to help customers to quickly find a solution. When user requirements do not match the solution quick reference table, the bandwidth allowable for implicit (I/O) message communication provided in EtherNet/IP Communication Specifications shall be used as a design constraint.

The following table lists the solution quick reference table for the H5U series.

Specification No.	Data Size	Number of Connections	RPI	Communication Bandwidth
SPEC.001	(0, 4] bytes	32	5 ms	12800 pps

Specification No.	Data Size	Number of Connections	RPI	Communication Bandwidth
SPEC.002	(4, 250] bytes	32	10 ms	6400 pps
SPEC.003	(4, 250] bytes	16	5 ms	6400 pps
SPEC.004	(250, 500] bytes	32	20 ms	3200 pps
SPEC.005	(250, 500] bytes	16	10 ms	3200 pps
SPEC.006	(250, 500] bytes	8	5 ms	3200 pps

1. You are recommended to set the user program scan cycle to RPI x 2.

2. "(a, b] bytes" means an integer value greater than a but equal to or less than b.

3. When user requirements match the data in the table, but the connection number and scan cycle do not match, balance them to meet the communication bandwidth requirements.

For example, when the data size of user requirement REQ.X does not match the data size of SPEC.Y in the table, REQ.X must meet the following condition:

REQ.X RPI/REQ.X connection number = SPEC.Y RPI/SPEC.Y connection number

The following table lists the solution quick reference table for the Easy32X series.

Specification No.	Data Size	Number of Connections	RPI	Communication Bandwidth
SPEC.001	(0, 4] bytes	32	5 ms	128000 pps
SPEC.002	(4, 250] bytes	32	5 ms	128000 pps
SPEC.003	(250, 500] bytes	32	5 ms	128000 pps
SPEC.004	(500, 1400] bytes	32	8 ms	8000 pps

1. You are recommended to set the user program scan cycle to RPI x 2.

2. "(a, b] bytes" means an integer value greater than a but equal to or less than b.

3. When user requirements match the data in the table, but the connection number and scan cycle do not match, balance them to meet the communication bandwidth requirements.

For example, when the data size of user requirement REQ.X does not match the data size of SPEC.Y in the table, REQ.X must meet the following condition:

REQ.X RPI/REQ.X connection number = SPEC.Y RPI/SPEC.Y connection number

The following table lists the solution quick reference table for the Easy52X series.

Specification No.	Data Size	Number of Connections	RPI	Communication Bandwidth
SPEC.001	(0, 4] bytes	32	5 ms	128000 pps
SPEC.002	(4, 250] bytes	32	5 ms	128000 pps
SPEC.003	(250, 500] bytes	32	5 ms	128000 pps

Specification No.	Data Size	Number of Connections	RPI	Communication Bandwidth
SPEC.004	(500, 1444] bytes	32	5 ms	128000 pps

1. You are recommended to set the user program scan cycle to RPI x 2.

2. "(a, b] bytes" means an integer value greater than a but equal to or less than b.

3. When user requirements match the data in the table, but the connection number and scan cycle do not match, balance them to meet the communication bandwidth requirements.

For example, when the data size of user requirement REQ.X does not match the data size of SPEC.Y in the table, REQ.X must meet the following condition:

REQ.X RPI/REQ.X connection number = SPEC.Y RPI/SPEC.Y connection number

12.2.4 Ethernet-IP Solution Selection Example

Taking REQ.A as an example to describe how to use the solution quick reference table

User requirement REQ.A

One H5U connects to 32 EtherNet/IP devices.

The data size of each connection is 500 bytes for both input data and output data.

RPI is 25 ms.

Solution

1. Use the data size and connection number of REQ.A as indexes and search for the specification SPEC.004 in the "Solution quick reference table for the H5U series" of the section Quick Reference Table of EtherNet/IP Solutions.
2. The scan cycle of REQ.A is 25 ms, greater than 20 ms of SPEC.004. In this case, you do not need to reduce the connection number and the recommended user program scan cycle is 50 ms.
3. Options:
   REQ.A communication bandwidth: 1000 ms/RPI x Connection number x 2 = 1000 ms/25 ms x 32 x 2 = 2560 pps.

The communication bandwidth provided in the "Solution quick reference table for the H5U series" of the section Reference Table of EtherNet/IP Solutions is 3200 pps, greater than 2560 pps. Therefore, the solution is appropriate.

Taking REQ.B as an example to describe how to use the solution quick reference table

User requirement REQ.B

One H5U connects to 32 EtherNet/IP devices.

The data size of each connection is 300 bytes for both input data and output data.

RPI is 10 ms.

Solution 1

1. Use the data size and connection number of REQ.B as indexes and search for the specification SPEC.004 in the "Solution quick reference table for the H5U series" of the section Quick Reference Table of EtherNet/IP Solutions.

2. RPI of REQ.B is 10 ms, less than 20 ms of SPEC.004.
3. Solution: Increase the RPI

RPI of REQ.B must be equal to or greater than 20 ms of SPEC.004. Therefore, you are recommended to set RPI to 20 ms.

Solution 2

1. Use the data size and RPI of REQ.B as indexes and search for the specification SPEC.005 in the "Solution quick reference table for the H5U series" of the section Quick Reference Table of EtherNet/IP Solutions.
2. The connection number of REQ.B is 32, greater than 16 of SPEC.005.
3. Solution: Reduce the connection number

The connection number of REQ.B must be equal to or less than 16 of SPEC.005. Therefore, you are recommended to set the connection number to 16.

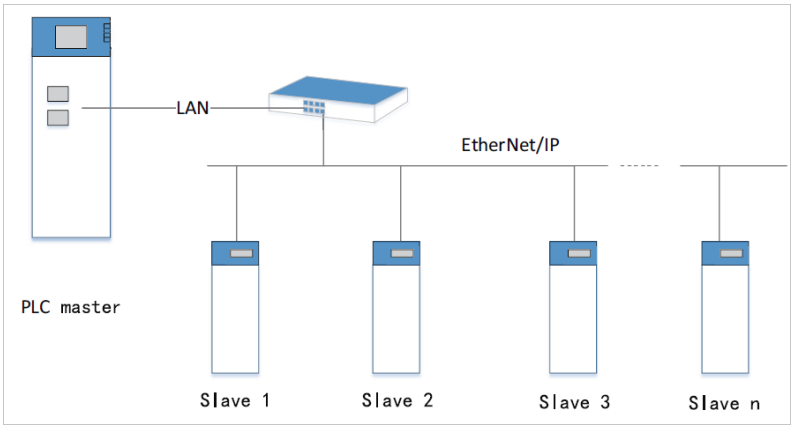
In general, both solution 1 and solution 2 are appropriate for REQ.B after adjustment.

12.3 Class 1 Communication

12.3.1 Planning IP Addresses for Ethernet-IP Devices

EtherNet/IP supports bus topology, star topology, and bus-star topology. In a star topology, all nodes are connected to the network hub, and nodes are easily added, deleted, and maintained. Such topology is often used due to its cost-effectiveness, easy connection, and availability of required devices.

IP addresses of all devices must be unique and in the same EtherNet/IP network segment. The following figure shows a star topology.



IP address of the EtherNet/IP master: 192.168.1.100

IP address of the EtherNet/IP slave 1: 192.168.1.101

IP address of the EtherNet/IP slave 2: 192.168.1.102

……

IP address of the EtherNet/IP slave n: 192.168.1.XXX

12.3.2 Master Configuration

12.3.2.1 Adding an Ethernet-IP Slave

1. Connect to the target PLC. For details, see 4.2.3 USB Connection.

2. Import the EDS description file of the slave.
   Create a master station project. Right-click "EtherNet/IP Devices" under "Toolbox". In the menu displayed, click "Import EDS". In the dialog box displayed, select the EDS file and then click "Open". The following figure takes importing the EDS file of KEYENCE N-L20 EtherNet/IP as an example.

   > **NOTE**
   >
   > To synchronize with the latest EDS file of a slave when there is any change to the EDS file, expand the "EtherNet/IP" menu, right-click the slave, and select "Update EDS" in the shortcut menu. In this process, select the option that the EDS file exported from the slave overwrites the original file in the directory. For details, see 12.3.2.2 Exporting an EDS File.

3. Create the configuration. You can create a configuration in AutoShop in two ways.
   ①: Locate the device file you want to import, and double-click the file. The device is added to the network configuration. You can add multiple EtherNet/IP slave stations one by one.

   ②: Scan a device to add it to the configuration. When AutoShop and PLC are connected, you can click "Auto Scan" to scan EtherNet/IP devices connected to the PLC. The operation steps are as follows.

   a. Right-click "EtherNet/IP" in the project tree. In the menu displayed, click "Auto Scan".

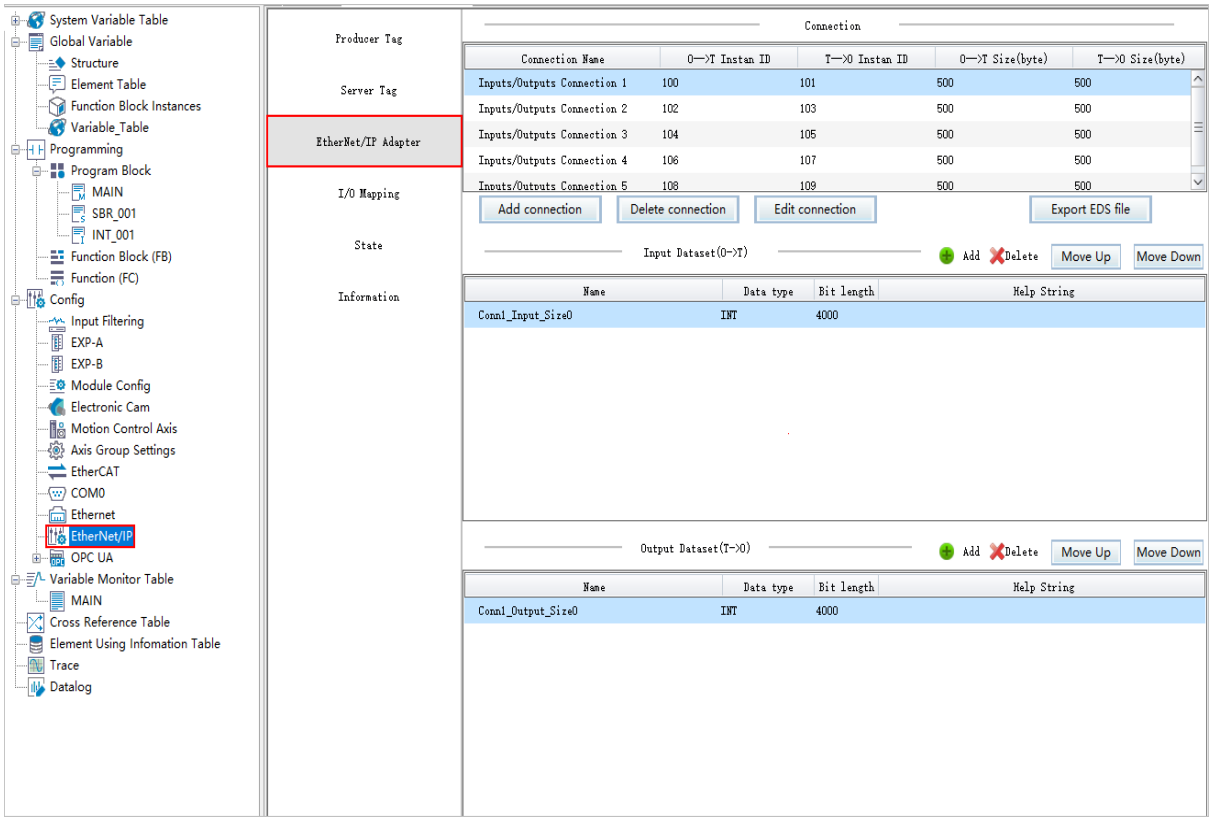
   b. In the "Auto Scan" dialog box displayed, slaves configured for the current project are displayed on the left. The names of the slaves detected by the PLC after you click "Auto Scan" are displayed on the right. The slaves in red are devices for which no matching EDS file is found in AutoShop, while configurations of slaves in black can be updated.

   c. Click "Update Config". The system asks you to confirm whether to save the current configuration. After you select "Yes", slaves whose configurations can be updated are added to the current configuration. If you select "No", all EtherNet/IP configurations are deleted and the slaves whose configurations can be updated are added.

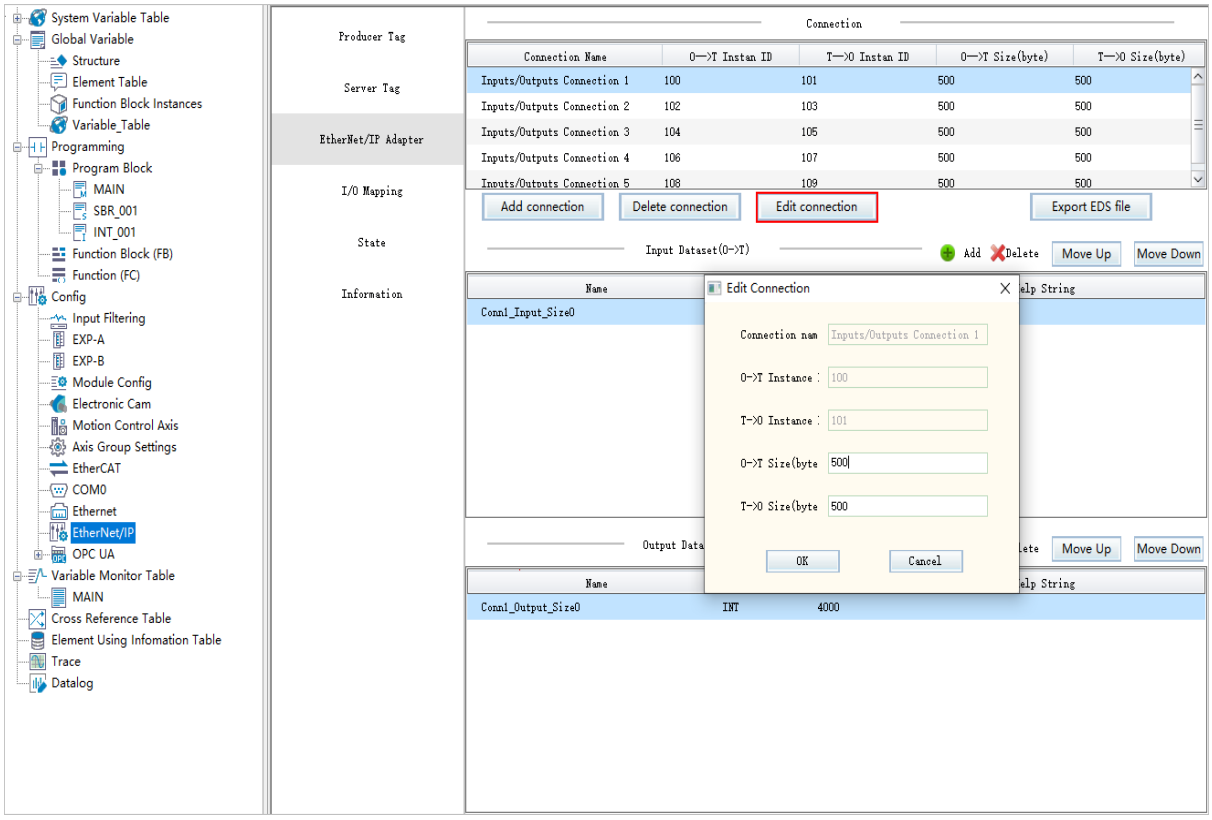
12.3.2.2 Exporting an EDS File

The way to export EDS files is basically the same for Easy and H5U. The main difference is that Easy supports a maximum of 1444 bytes while H5U supports a maximum of 500 bytes.

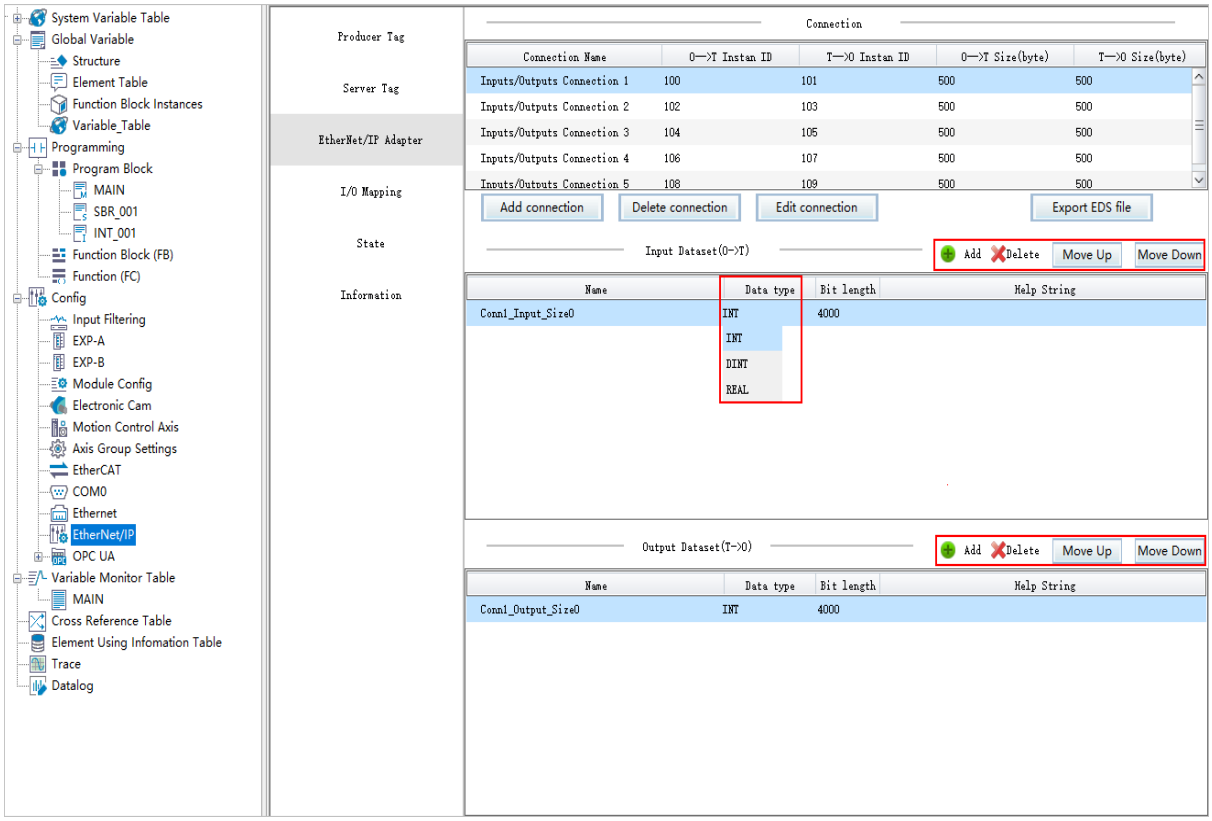
1. Double-click "EtherNet/IP", and then select "EIP Adapter". On the EIP slave page displayed, you can add or delete connections (up to 16 connections are supported), as shown in the following figure.



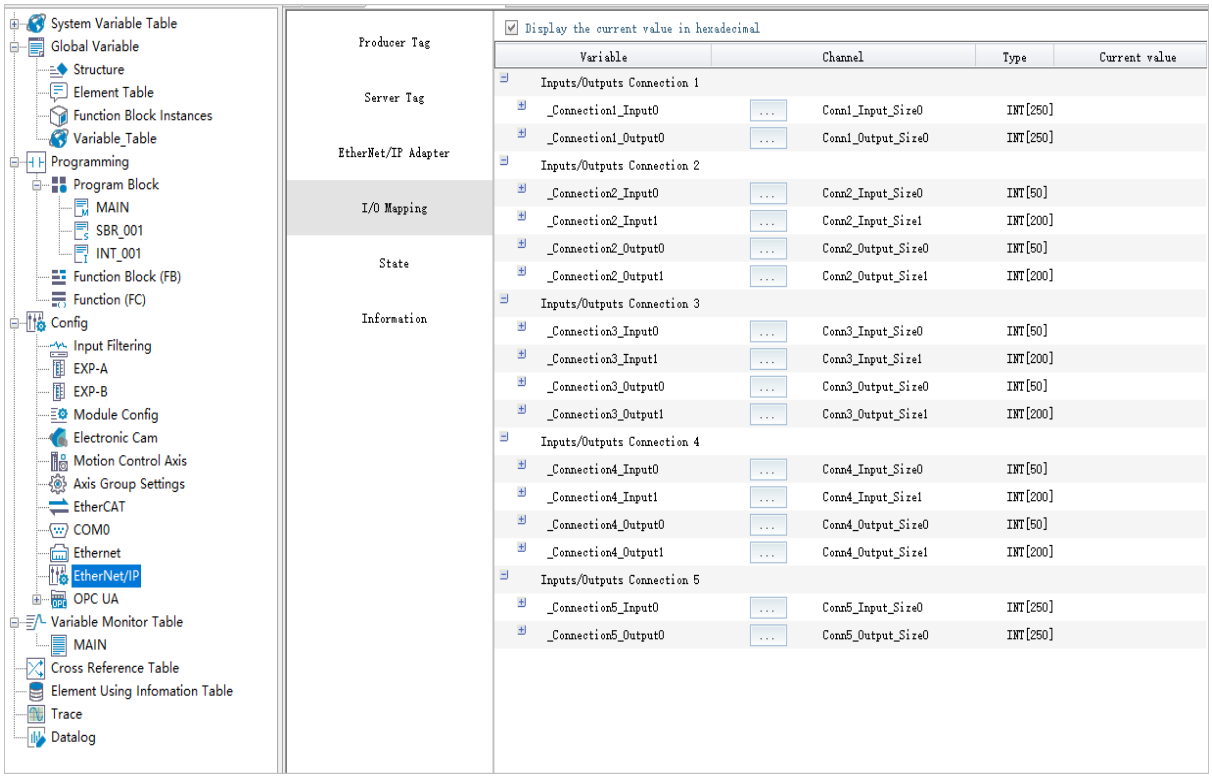
2. Click "Edit connection" to edit the O->T and T->O sizes for connections. The default value is 500 bytes.



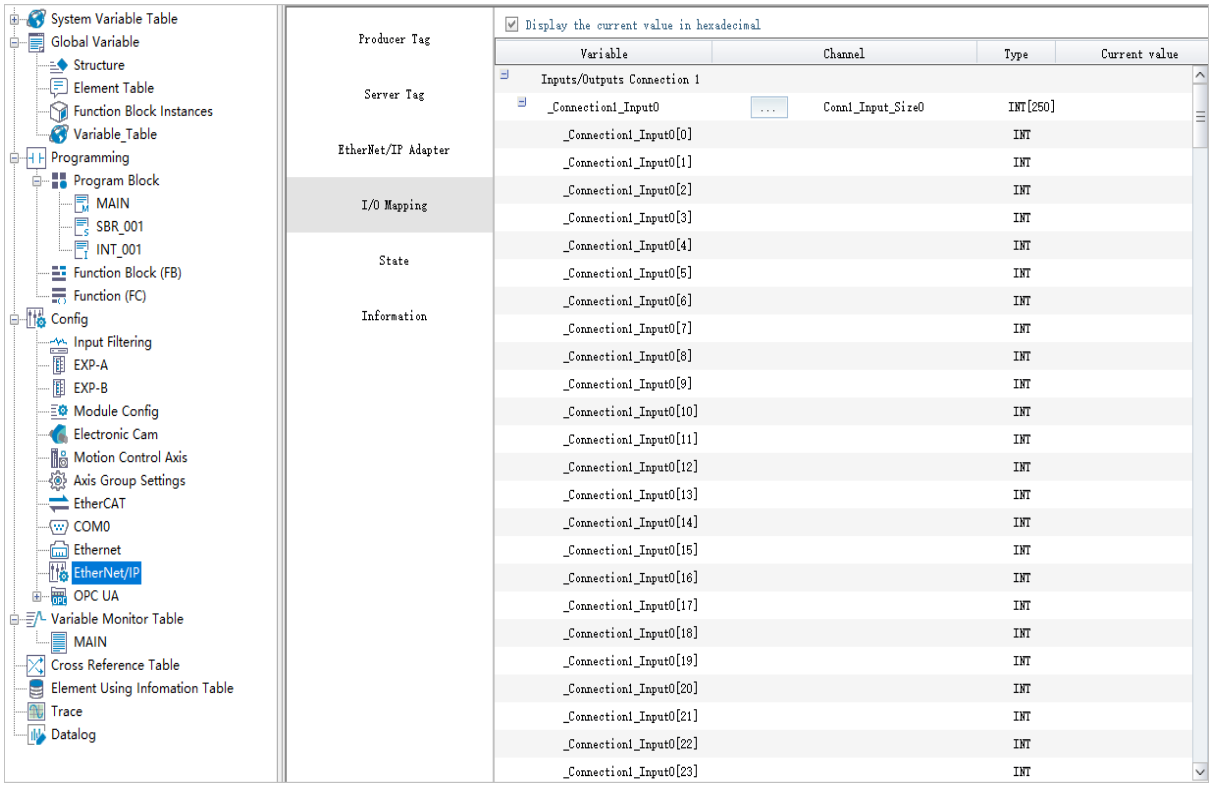
3. You can also split the size of bytes according to the data set of O->T and T->O, and split the total bytes into multiple variables of the required byte size for mapping. You can add or delete a data set to adjust the byte size, or move a data set up or down.



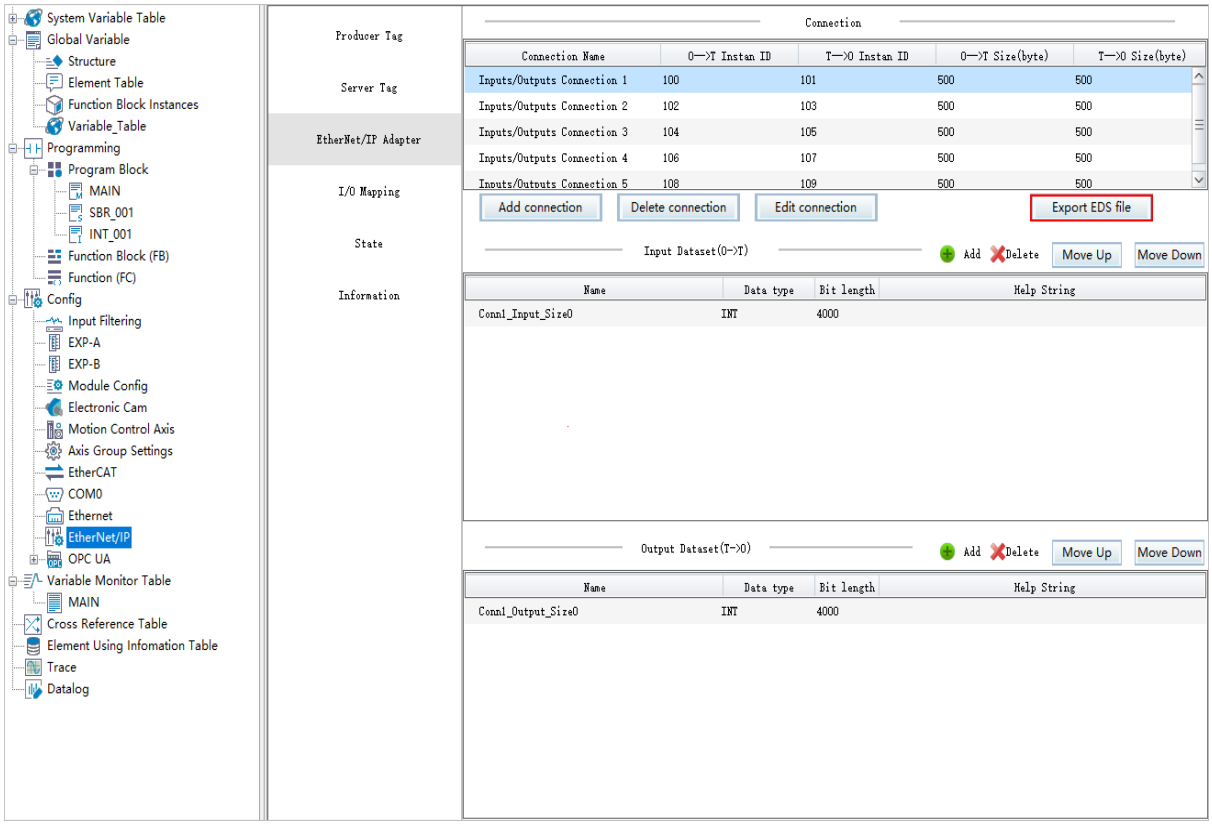
4. Split bytes will generate multiple corresponding variable which can be viewed on the "IO Mapping" page. The corresponding bytes will generate the corresponding variable numbers, and the byte size will be that of the connection.



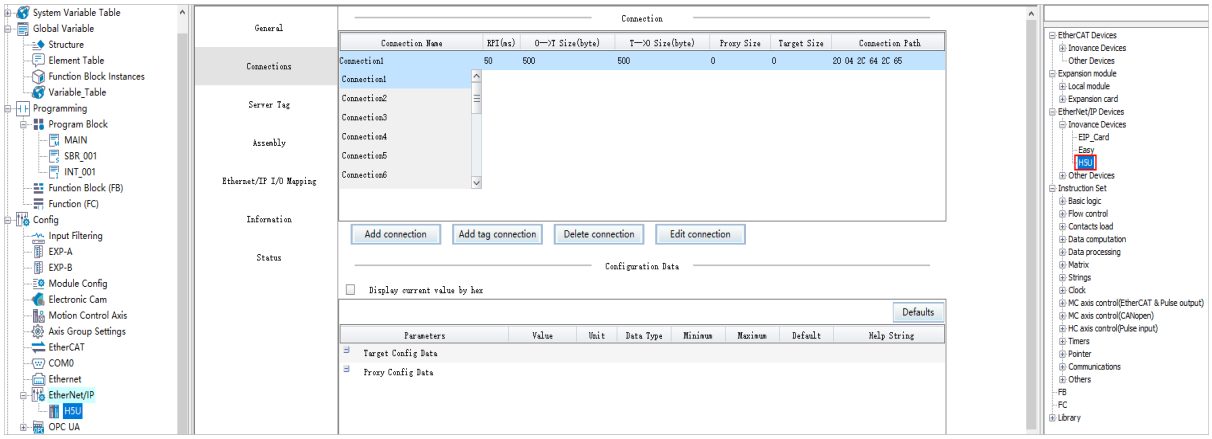
5. You can expand and view an array.



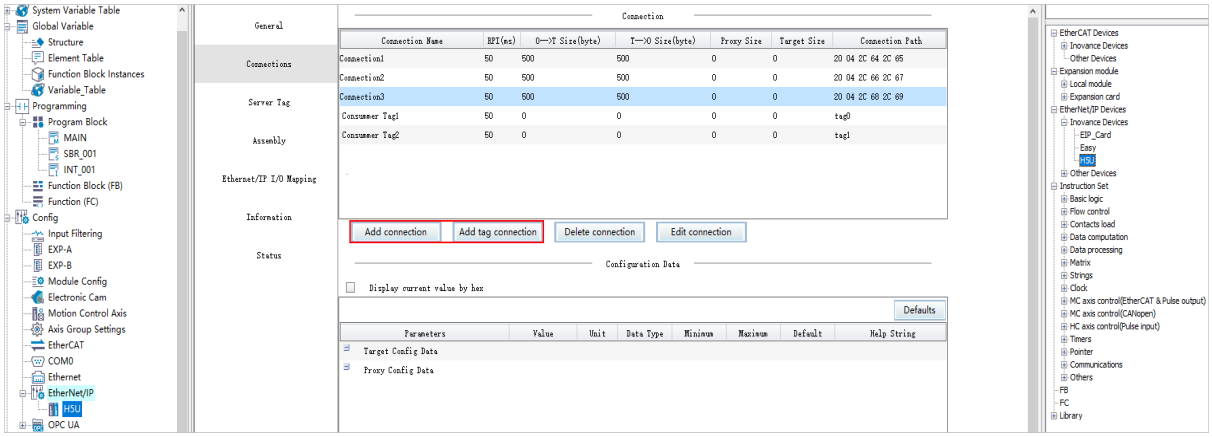
6. Select parameters, switch to the "EtherNet/IP" page, and click "Export EDS file" to export the EDS file to any folder.



7. To import the exported EDS file to the background, choose "EtherNet/IP Devices" under "Toolbox", and double-click the H5U.eds file. On the following page displayed, select the I/O connection from the drop-down list of "Connection Name".



8. You can click "Add connection" or "Add tag connection" to add the required connection. "Add connection" is used to add an I/O connection while "Add tag connection" is used to add a consumer connection.

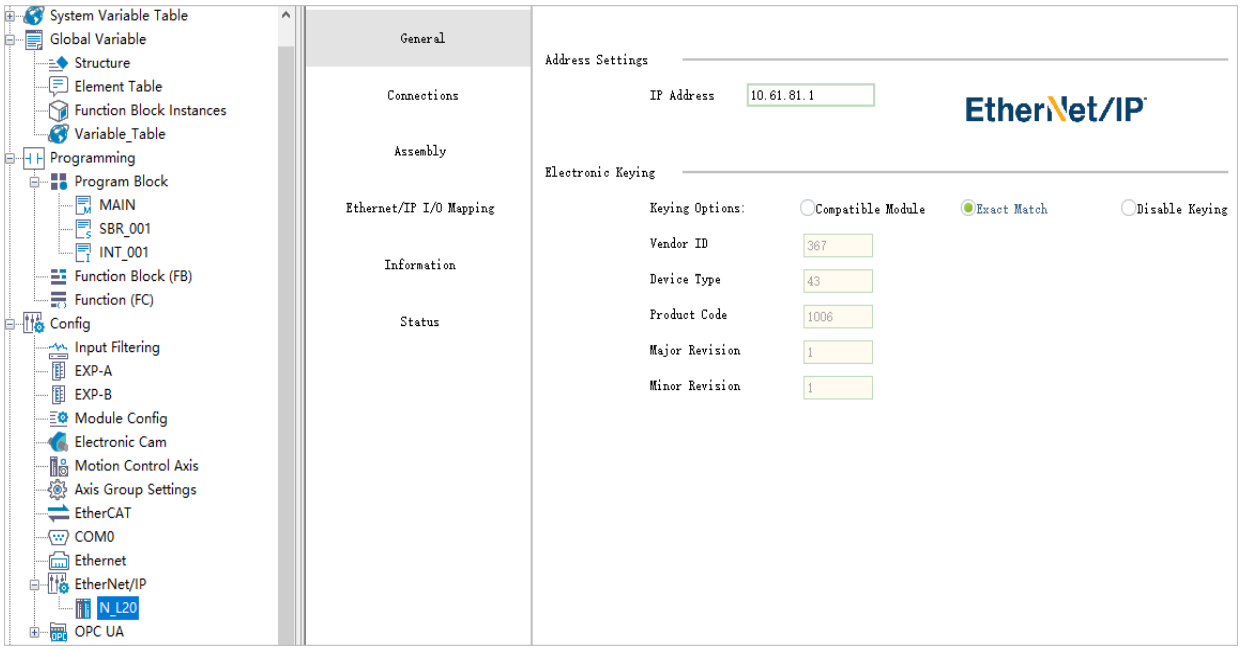


12.3.3 Slave Configuration

12.3.3.1 General Settings

Modifying the configuration of a single slave

Double-click the EtherNet/IP slave station "N-L20" you want to set. On the page displayed, set the IP address and matching options for the slave station.



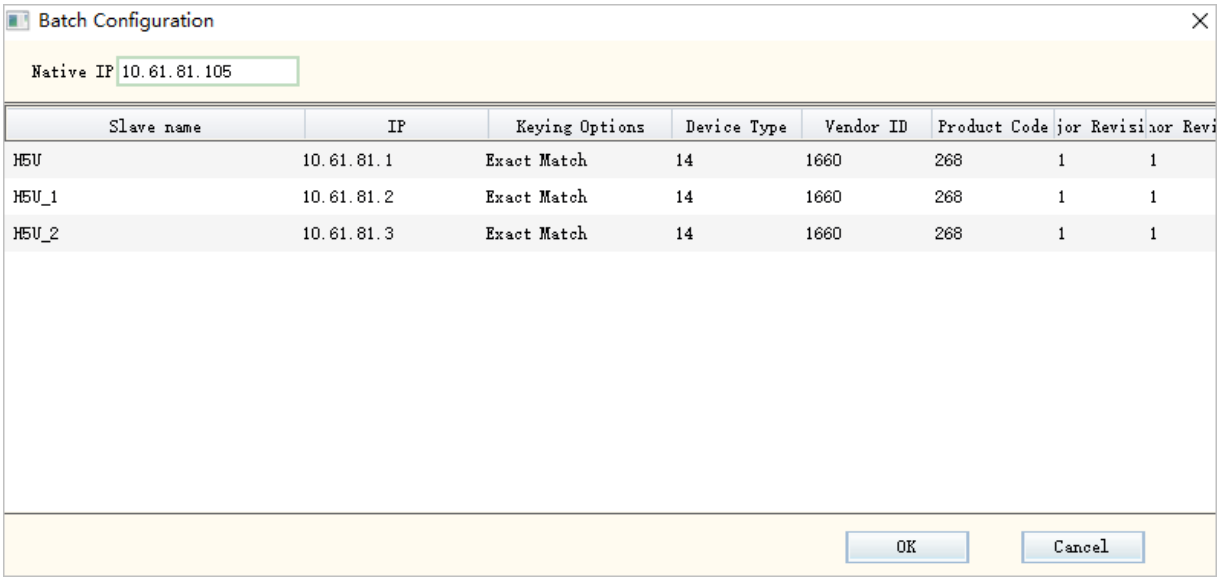
Electronic Keying: The EtherNet/IP master station checks whether the following fields of the EtherNet/IP slave station match those in the EDS file: supplier code, device type, product code, major version, and minor version. The following table lists the matching options.

Parameter Name	Description
Compatible Module	Conditional matching. The communication is established only when the vendor code, device type, and product code are identical and the major version and minor version of the slave station are equal to or later than those in the EDS file (the version of the slave station is ≥ the version in the EDS file, and the slave station is compatible with EDS files of earlier versions).
Exact Match	Communication can be established between EtherNet/IP master and slave stations only when all of such fields are exactly matched.
Disable Keying[1]	Communication is directly established between EtherNet/IP master and slave stations without checking whether such fields are matched.

[1]: Exercise special caution when selecting "Disable Keying". Improper use of this option may cause physical injury, property damage, or economic loss. It is strongly advised against using the "Disable Keying" option.

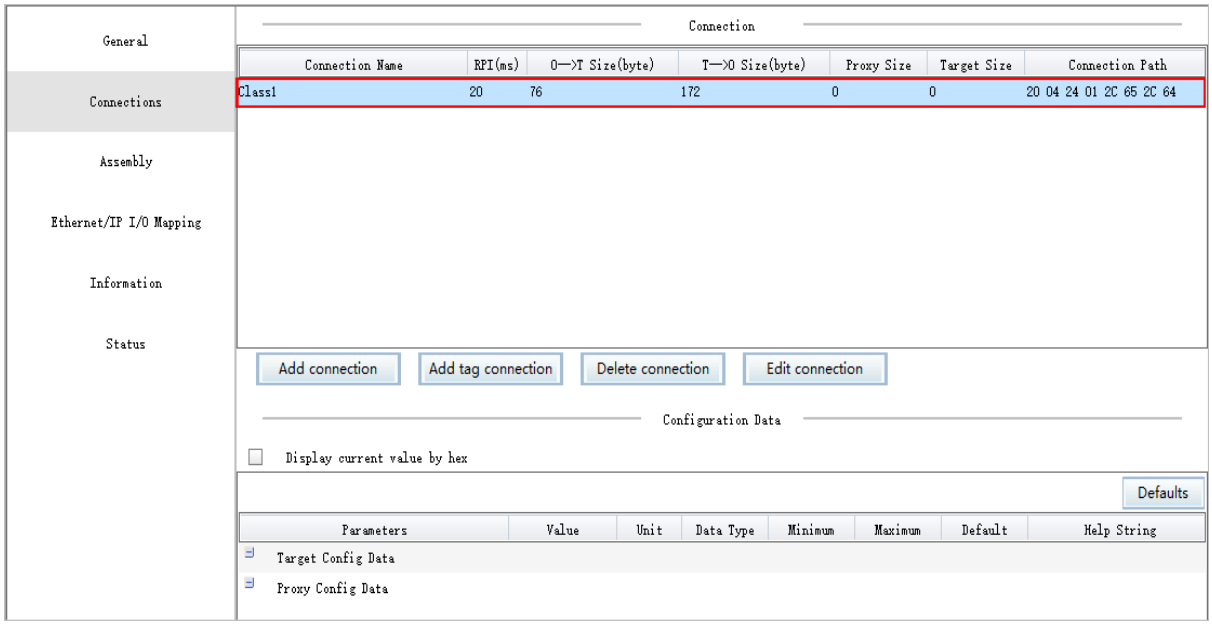
Modifying configurations of multiple slave stations

When multiple slave stations are configured, right-click "EtherNet/IP" in the "Project Manager" section. In the menu displayed, select "Batch Configuration". In the "Batch Configuration" dialog box displayed, modify the IP addresses and matching options of slave stations in a batch.

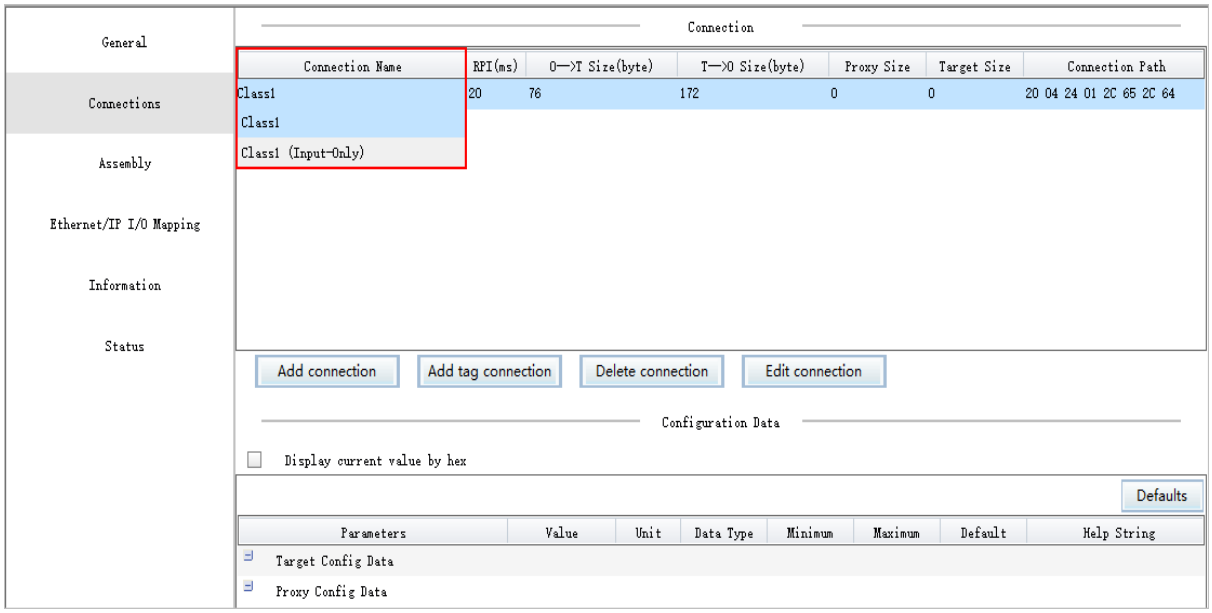


12.3.3.2 Connection Settings

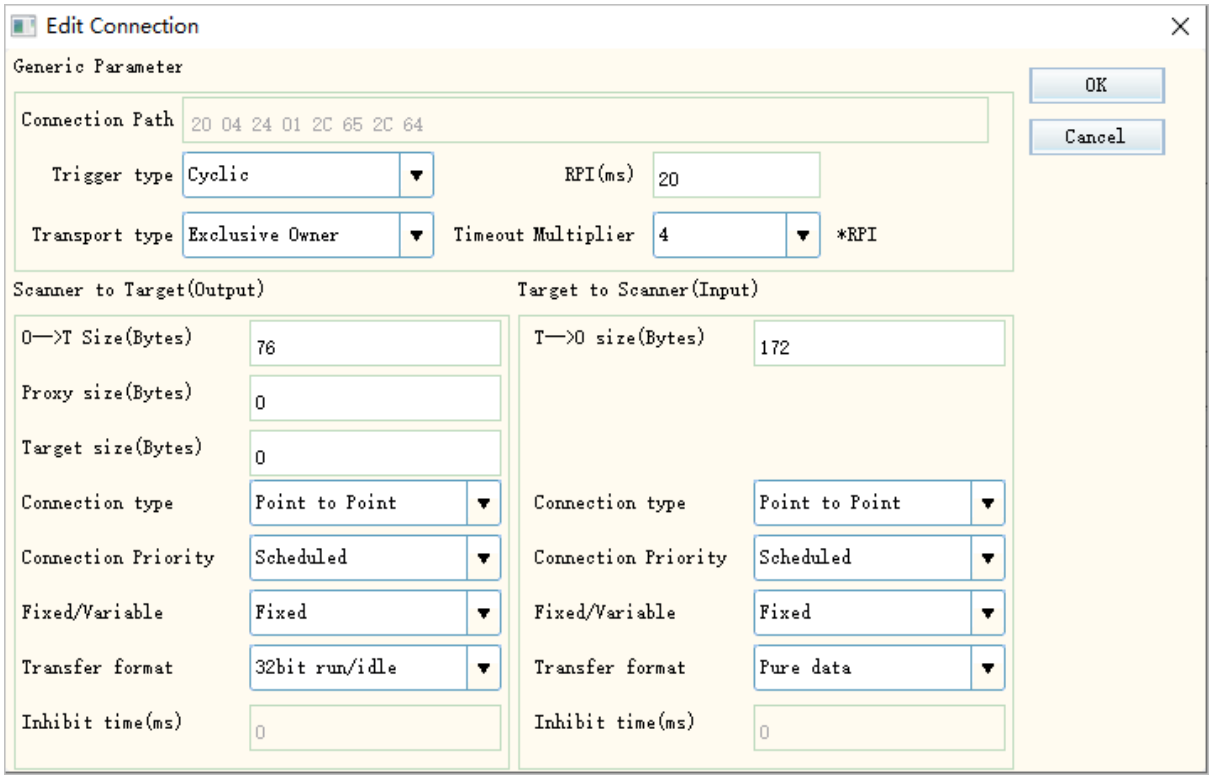
1. Add a connection.
   The EDS description file of an EtherNet/IP slave station contains a default connection path. After the EtherNet/IP network configuration is added, the background connection page loads the default connection path, as shown in the following figure.



You can also click the connection name and set the connection path pre-defined in the EDS file, See the figure below.



On the preceding page, click "Edit connection" to enter the connection setting page. Generally, all parameters other than RPI (communication cycle) of the connection must use the default values.



2. Configure the general parameters.

- Connection Path: Specifies the format and connection instance of a byte stream frame.
  For example, 20 04 2C C6 2C 68 (for details, see EIP-CIP-V1-1.0, Appendix C: Data Management).

  20: Logical Segment, ClassID, and 8-bit logical address

  04: Assembly Object (04H)

  2C: Logical Segment, Connection Point, and 8-bit logical address

  C6: ID-C6H of the Assembly Object instance

  2C: Logical Segment, Connection Point, and 8-bit logical address

  68: ID-68H of the Assembly Object instance

  Note: The connection path must be configured based on the guide of the specific slave station. The connection path varies with the manufacturer.

- Trigger type: When "Change-Of-State" is selected, data is circularly sent according to the RPI, and the sending operation is triggered immediately when the data value changes in each scan cycle of the PLC.

- RPI (ms): Requested Packet Interval. It indicates the communication transmission interval in ms. The RPI of each node can be set individually without affecting each other.

3. Scanner to Target.

- Transmission byte size
  1) O->T Size (Bytes): Indicates the amount of data transferred from the producer (scanner) to the consumer (target device), in bytes.

  2) T->O Size (Bytes): Indicates the amount of data transferred from the consumer (target device) to the producer (scanner), in bytes.

- Transport type
  Exclusive Owner: Allows users to set both data sending from the initiator to the target device and data receiving from the target device to the initiator.

Redundant Owner: Allows multiple initiators to create independent and identical connections to the same target device.

Input Only: This connection can only be used to set data receiving from the target device to the initiator.

Listen Only: EtherNet/IP devices apply this type of connections to listen to multicast data without providing configuration or scheduling information.

- Trigger type
  Cyclic: Periodically triggers data transmission.

  Change-Of-State: Transmits data when a change in the state of the application object is detected.

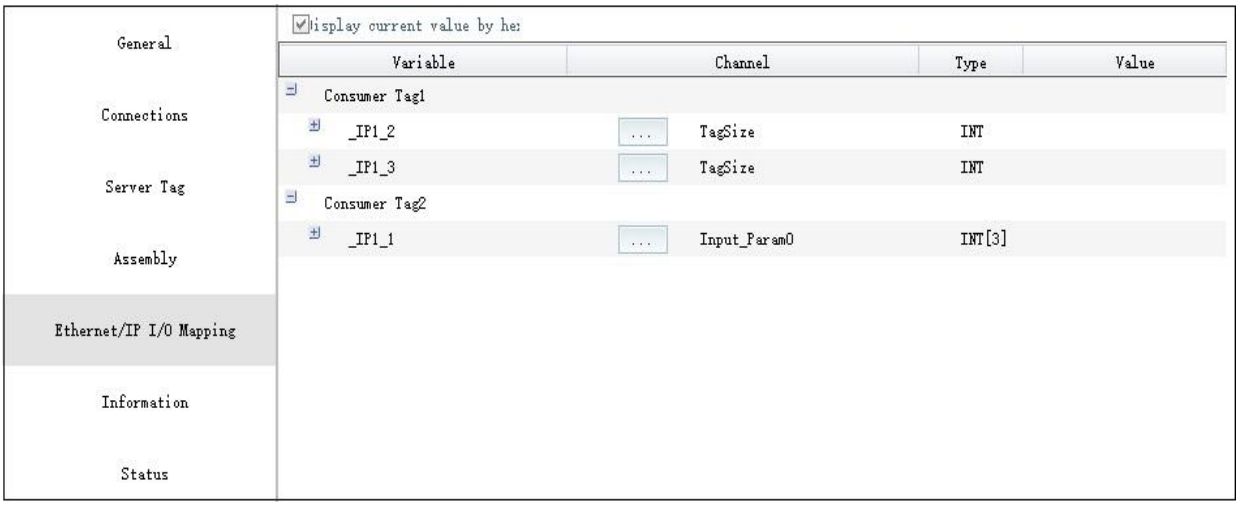
  Application Object: Transmits data when the application object is triggered.
- Connection type
  Multicast: Multiple scanners receive data from one target device at the same time.

  Point-to-Point: One scanner can receive data from only one target device.
- Inhibit time (ms): Indicates the minimum data refresh interval. It can be set when the trigger type is set to "Change-Of-State". The value range is 0 ms to 255 ms and the default value is 0 ms, indicating no inhibit time is set.

> **NOTE**
>
> Click "Add connection" to open the connection setting page and select "General connections" to customize a connection as needed. This requires knowledge of the CIP protocol. For users without such knowledge, it is recommended to use the default connection configuration.

12.3.3.3 Configuring IO Variable Mapping

You can expand an array when modifying a variable mapping. For example, to create a connection, of which the O->T size of the consumer tag is 10 bytes, the number of server tags is 5, and _IP2_0 and _IP2_1 are variables of the default mapping, click "..." and input the variables at the variable name position to modify the mapping.



Only three data types are displayed: INT, DINT, and REAL. If you have modified the data type of the consumer tag through the data set, the data type is converted to any of these three types. The correspondence is as follows.

INT: BYTE, INT, UINT, SINT, and USINT are all displayed as INT.

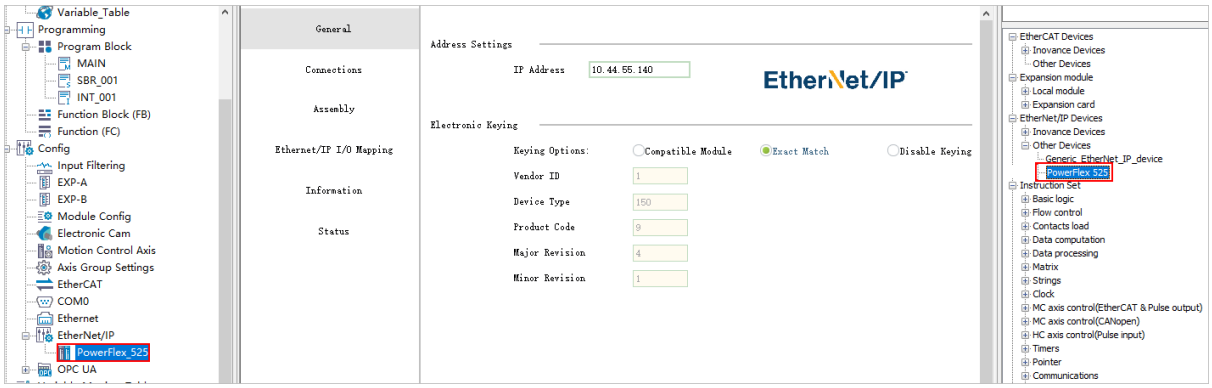
DINT: DINT, UDINT, LINT, ULINT, LWORD, DWORD, and WORD are all displayed as DINT.

REAL: REAL and LREAL are displayed as REAL.

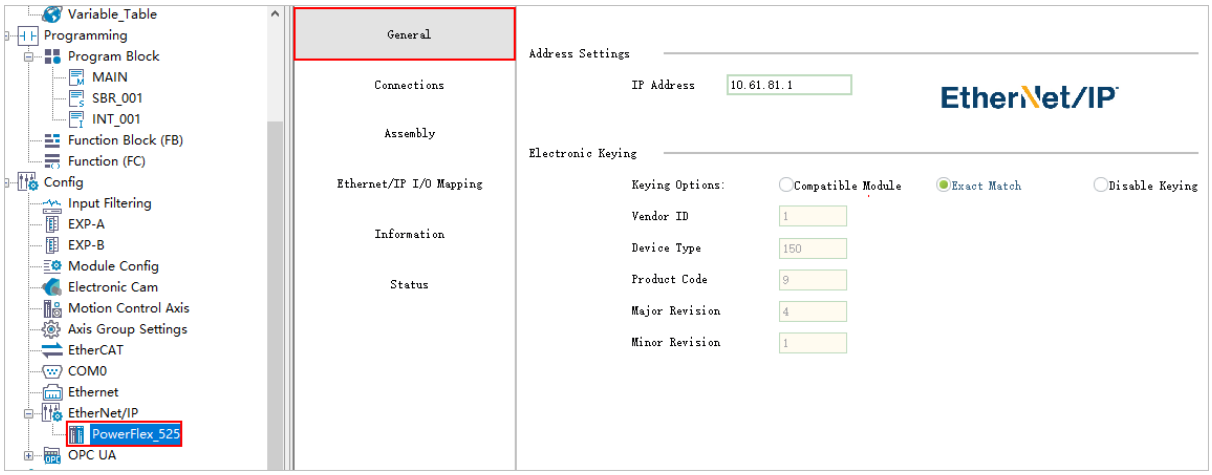
12.3.4 Ethernet-IP Master Application Example

This project describes how to create an EtherNet/IP network in which H5U serves as the master station. In this example, the slave station is the Rockwell PowerFlex 525 AC drive.

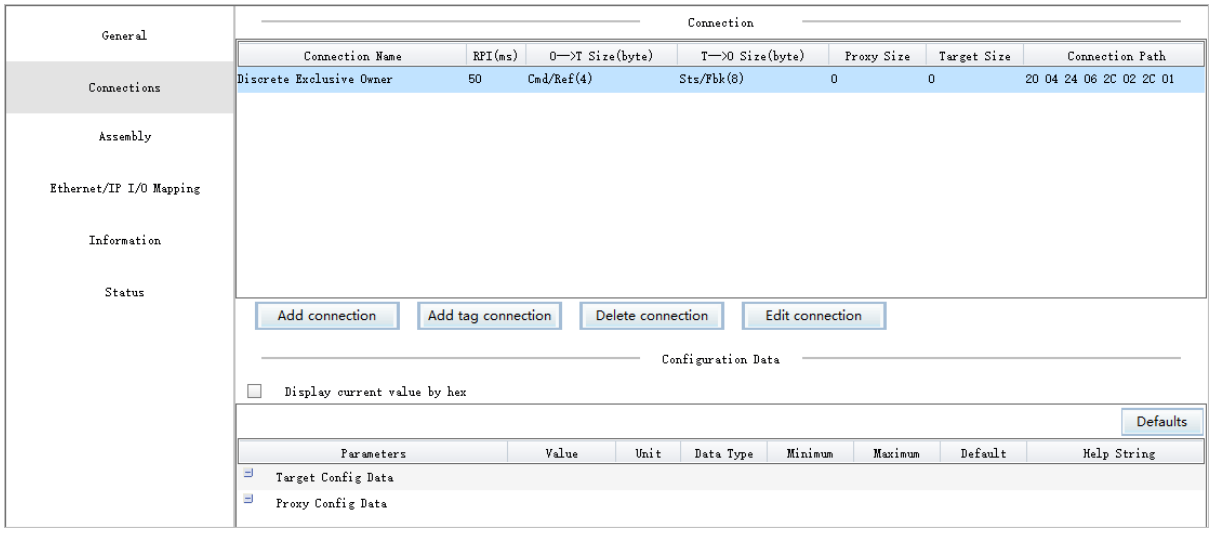
1. Under "EtherNet/IP Devices" of "Toolbox", select the corresponding slave station for I/O communication.



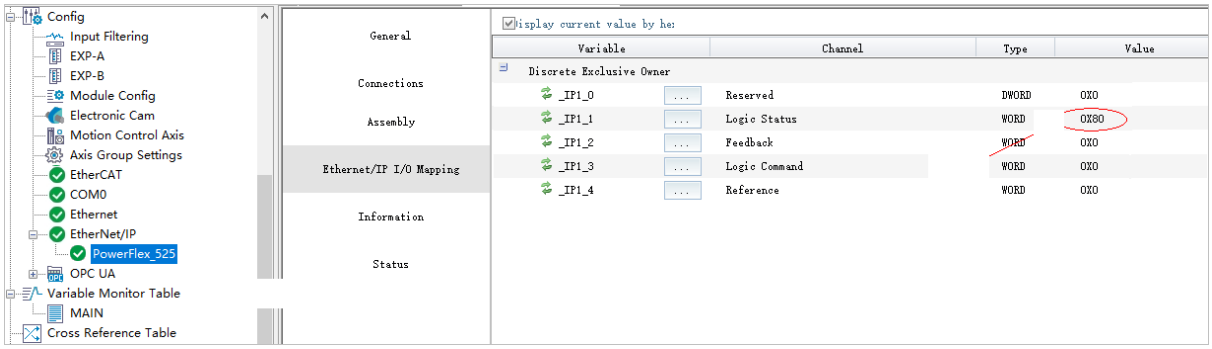
2. Set the IP address of the slave station you want to connect.



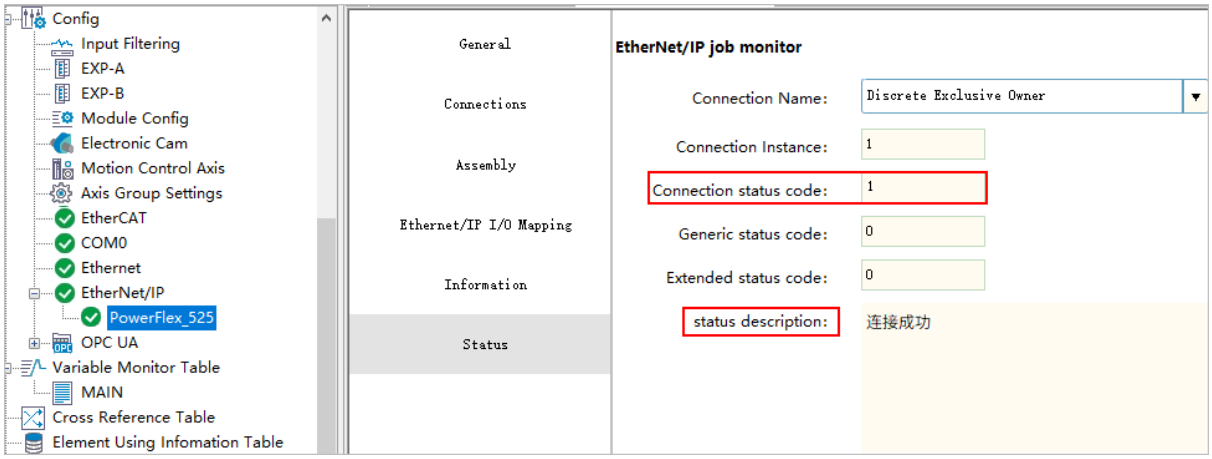
3. Set the connection parameters. The PowerFlex 525 AC drive supports only the default connection and therefore only one connection is listed under "Connection Name". You can directly change the RPI. In this case, RPI is 50 ms and cannot be modified.



4. After making the configuration, download the program, run the PLC, and monitor the communication status.



The communication details can be viewed on the status page.



The following table lists the status parameters.

Parameter Name	Description
Connection name	Name of the current connection
Connection instance	A unique instance ID allocated for each connection.
Connection status code	ID of the connection status
General status code	Used in combination with extension error code. See the following for details.
Extension status code	Used in combination with general error code. See the following for details.
Status description	See the description below

- Connection status codes and description

Connection Status Code	Status Code Description	
	English	Chinese
1	New connection successfully established	New connection successfully established
2	Connection establishment failed. Target is offline	Connection establishment failed. Target is offline
3	Connection establishment failed. Target is online but did not respond to the connection request	Connection establishment failed. Target is online but did not respond to the connection requestConnection establishment failed. Target is offline

Connection Status Code	Status Code Description	
	English	Chinese
4	Connection establishment failed. Target responded with an error response	Connection establishment failed. Target responded with an error response
5	Active connection timed out	Active connection timed out
6	Connection closed based either on the local or remote request	Connection closed based either on the local or remote request
16	Connection establishment failed. (D)TLS handshake could not be completed	Connection establishment failed. (D)TLS handshake could not be completed
101	Unable to connect to the path specified in the request	Unable to connect to the path specified in the request
102	Request timed out - response never received	Request timed out - response never received

- General status codes, extension status codes, and description

General Status Code	Extension Status Code	Status Description	
		English	Chinese
1	256	Connection in use or Duplicate Forward Open	Connection in use or Duplicate Forward Open
1	259	Transport Class and Trigger combination not supported	Transport class and trigger mode not supported
1	261	A connection related service failed along the connection path.	Ownership conflict or OUNID mismatch. The configuration is already owned by another initiator.
1	262	Connection ownership conflict	Ownership conflict. Other connections have exclusively occupied some resources of this connection. For example, two PLCs control the output point of IO module at the same time.
1	263	Connection not found at target application	The target connection was not found, and the connection that Forward_Close requested to close was not found in the target node.
1	264	Invalid connection type or connection priority	Invalid network connection parameters. The connection type, connection priority, redundancy owner or fixed/variable value device are not supported.
1	265	Invalid connection size	Invalid connection size
1	272	Device not configured	

General Status Code	Extension Status Code	Status Description	
		English	Chinese
			The connection target is not configured, and the connection request does not contain a data segment for configuration.
1	273	RPI not supported or there is a problem with connection time-out multiplier or production inhibit time	RPI not supported or there is a problem with connection time-out multiplier or production inhibit time
1	274	RPI not acceptable.	The RPI value is not acceptable, the RPI value in the Forward_Open request is out of the range required by the target device, or the target device is being produced in different cycles.
1	275	Connection Manager cannot support any more connections	The connection limit is exceeded, and the target cannot support more connections.
1	276	Either the Vendor Id or the Product Code in the key segment did not match the device	Vendor ID or product code does not match.
1	277	Product Type in the key segment did not match the device	Product type mismatch
1	278	Major or Minor revision information in the key segment did not match the device	Version mismatch
1	279	Invalid connection point specified in the connection path	Invalid production or consumption path
1	280	Invalid configuration format	Invalid or inconsistent configuration path
1	281	Connection request fails since there is no controlling connection currently open	The non-listening-only connection is not enabled, the non-listening-only connection that does not use the same path for the listening-only connection has been enabled, or the current and enabled connection types are not multicast.
1	282	Target application cannot support any more connections	The target object exceeds the connection limit.
1	283	RPI is smaller than the Production Inhibit time	RPI is smaller than the Production Inhibit time

General Status Code	Extension Status Code	Status Description	
		English	Chinese
1	284	The transport class requested in the Transport Type/Trigger parameter is not supported	The transport type not supported.
1	285	The production trigger requested in the Transport Type/Trigger parameter is not supported	The T->O trigger mode is not supported.
1	286	The direction requested in the Transport Type/Trigger parameter is not supported	The direction requested in the Transport Type/Trigger parameter is not supported.
1	287	O->T fixed/variable flag is not supported	O->T fixed/variable flag is not supported.
1	288	T->O fixed/variable flag is not supported	T->O fixed/variable flag is not supported.
1	289	O->T priority code is not supported	O->T priority is not supported.
1	290	T->O priority code is not supported	T->O priority is not supported.
1	291	O->T connection type is not supported	O->T connection type is not supported.
1	292	T->O connection type is not supported	T->O connection type is not supported.
1	293	O->T Redundant Owner flag is not supported	O->T Redundant Owner flag is not supported.
1	294	The data segment provided in the Connection_Path parameter did not contain an acceptable number of 16-bit words for the configuration application path requested	Configuration length is not supported.
1	295	Cannot support the size requested in the O->T Network Connection Parameter	O->T connection length is not supported.
1	296	Cannot support the size requested in the T->O Network Connection Parameter	T->O connection length is not supported.

General Status Code	Extension Status Code	Status Description	
		English	Chinese
1	297	The configuration application path specified does not correspond to a valid configuration application path	The configuration path is invalid.
1	298	The consumed application path specified does not correspond to a valid consumed application path	Invalid consumption path (O->T)
1	299	The produced application path specified does not correspond to a valid produced application path	Invalid production path (T->O)
1	300	Configuration Symbol does not exist	Configuration symbol does not exist.
1	301	Consuming Symbol does not exist	Consuming Symbol does not exist (O->T).
1	302	Producing Symbol does not exist	Producing symbol does not exist (T->O).
1	303	The combination of configuration and/or consume and/or produce application paths specified are inconsistent with each other	The combination of configuration, consumption and production application paths is not supported.
1	304	Information in the data segment is not consistent with the format of the consumed data	Information in the data segment is not consistent with the format of the consumed data.
1	305	Information in the data segment is not consistent with the format of the produced data	Information in the data segment is not consistent with the format of the producer data.
1	306	The target does not support the function requested by the Null Forward Open	The target does not support the function requested by the Null Forward Open.
1	307	Connection timeout multiplier value is reserved or produces a timeout value that is too large to support in the device	Unacceptable timeout multiplier

General Status Code	Extension Status Code	Status Description	
		English	Chinese
1	308	Mismatched connection size for existing multicast connection	T->O connection size does not match the existing multicast connection.
1	309	Mismatched fixed/variable connection type for existing multicast connection	Do not match T->O fixed/variable connection type of existing multicast connection.
1	310	Mismatched fixed/variable connection priority for existing multicast connection	Do not match T->O connection priority of existing multicast connection.
1	311	Mismatched transport class for existing multicast connection	Do not match the transmission type of the existing multicast connection.
1	312	Mismatched production trigger for existing multicast connection	Do not match the T->O production trigger mode of the existing multicast connection.
1	313	Mismatched production inhibit timer for existing multicast connection	Do not match T->O production prohibition time of existing multicast connection.
1	515	Connection timed out	Connection timeout
1	516	Unconnected Send timed out waiting for a response	Unconnected Send timed out waiting for a response
1	517	Parameter error in Unconnected Send service	Parameter error in Unconnected Send service
1	518	Message too large for Unconnected message service	Message too large for Unconnected message service
1	519	Unconnected acknowledge without reply	The Unconnected consumer service has been acknowledged but has not received a response.
1	769	No buffer memory available	No buffer memory available
1	770	Network bandwidth not available for data	Network bandwidth not available for data
1	771	No tag filters available	No consumed connection ID filter is available.
1	772	Not configured to send real-time data	Not configured to send plan priority data.
1	785	Port specified in port segment not available	The port is unavailable.

General Status Code	Extension Status Code	Status Description	
		English	Chinese
1	786	Link Address specified in port segment not available	Invalid link address
1	789	Invalid Segment Type or Segment Value in Path	Invalid segment exists in the connection path.
1	791	Either segment not present or encoded value in Network segment is invalid	Plan is not specified.
1	792	Link Address to Self invalid	The link address to self is invalid.
1	793	Resources on Secondary unavailable	Standby resource is not available.
1	794	Connection already established	Rack connection established
1	795	Direct connection already established	Module connection established
1	796	Miscellaneous	Miscellaneous. No other extension status code to indicate a connection error.
1	797	Redundant connection mismatch	Redundant connection mismatch
1	798	No more consumer resources available in the producing object	No more user-configurable linked consumer resources available in the production module
1	799	No connection resources exist for target path	No consumer resources available in the producing object
2	65535	Resources needed for the object to perform the requested service were unavailable	Resource unavailable, internal processing buffer unavailable
3	65535	Invalid Parameter value	Invalid parameter value
4	65535	The path segment identifier or the segment syntax was not understood by the processing node	Path format error
5	65535	The path is referencing an object class, instance or structure element that is not known or is not contained in the processing node	The target path is unknown. Check whether the requested path is correct.
6	65535	Only part of the expected data was transferred	Only part of the expected data was transferred.

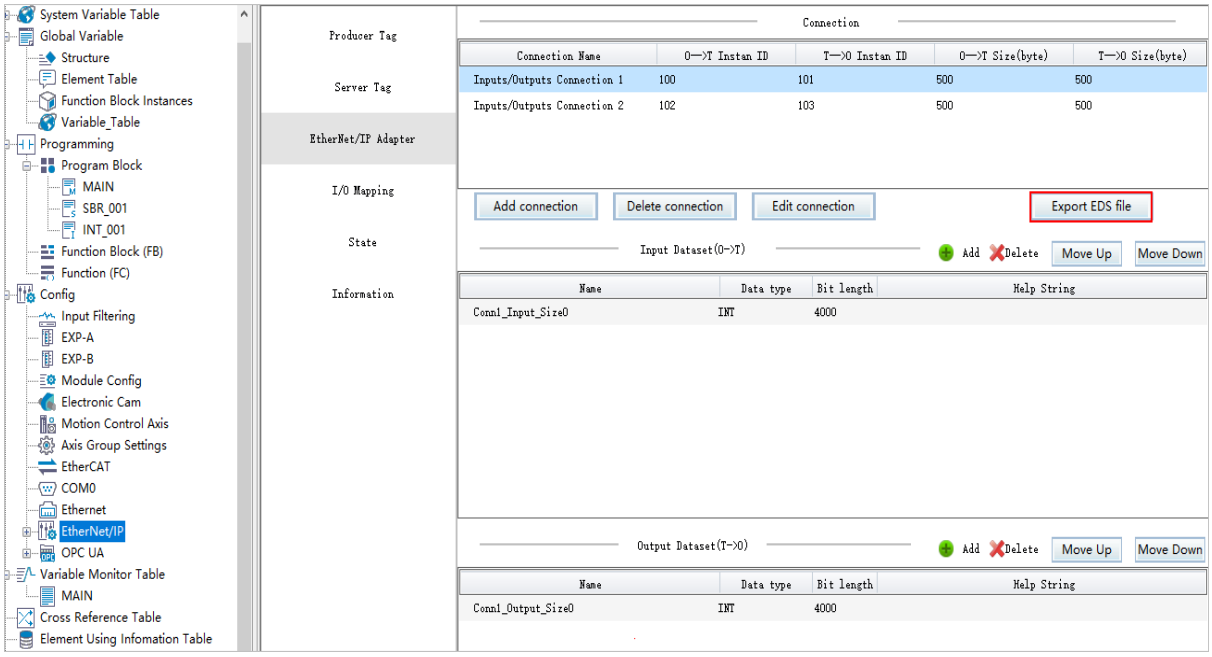
General Status Code	Extension Status Code	Status Description	
		English	Chinese
7	65535	The messaging connection was lost	Connection disconnected
8	65535	The requested service was not implemented or was not defined for this object class /instance	Service not supported
9	65535	Invalid attribute data detected	Invalid attribute data
10	65535	An attribute in the Get_Attribute_List or Set_Attribute_List response has a non-zero status	An attribute in the Get_Attribute_List or Set_Attribute_List response has a non-zero status.
11	65535	The object is already in the mode/state being requested by the service	The object is already in the state being requested by the service.
12	65535	The object cannot perform the requested service in its current mode/state	The object cannot perform the requested service in its current mode/state.
13	65535	The requested instance of object to be created already exists	The requested instance of object to be created already exists.
14	65535	A request to modify a non-modifiable attribute was received	The attribute is not settable.
15	65535	A permission/privilege check failed	A permission/privilege check failed.
16	65535	The device's current mode/ state prohibits the execution of the requested service	The device's current mode/state prohibits the execution of the requested service.
17	65535	The data to be transmitted in the response buffer is larger than the allocated response buffer	Response data is too large.
18	65535	The service specified an operation that is going to fragment a primitive data value, i.e. half a REAL data type	The underlying type is fragmented, and the service fragments the underlying data value, that is, half a REAL data type.
19	65535	The service did not supply enough data to perform the specified operation	The service did not supply enough data to perform the specified operation.

General Status Code	Extension Status Code	Status Description	
		English	Chinese
20	65535	The attribute specified in the request is not supported	The attribute is not supported.
21	65535	The service supplied more data than was expected	The service supplied more data than was expected.
22	65535	The object specified does not exist in the device	The object instance does not exist.
23	65535	The fragmentation sequence for this service is not currently active for this data	The fragmentation sequence for this service is not currently active for this data.
24	65535	The attribute data of this object was not saved prior to the requested service	The attribute value was not saved prior to the requested service.
25	65535	The attribute data of this object was not saved due to a failure during the attempt	The attribute value was not saved due to a failure during the attempt.
26	65535	The service request packet was too large for transmission on a network in the path to the destination	Routing error. The request packet is too large.
27	65535	The service response packet was too large for transmission on a network in the path to the destination	Routing error. The response packet is too large.
28	65535	The service did not supply an attribute in a list of attributes that was needed by the service to perform the requested behavior	Attribute list data is missing
29	65535	The service is returning the list of attributes supplied with status information for those attributes that were invalid	Invalid attribute value list
30	65535	An embedded service resulted in an error	Embedded error. Multi-service response contains an error.
31	65535	Vendor specific error	Vendor specific error. See the target equipment guide.
32	65535		

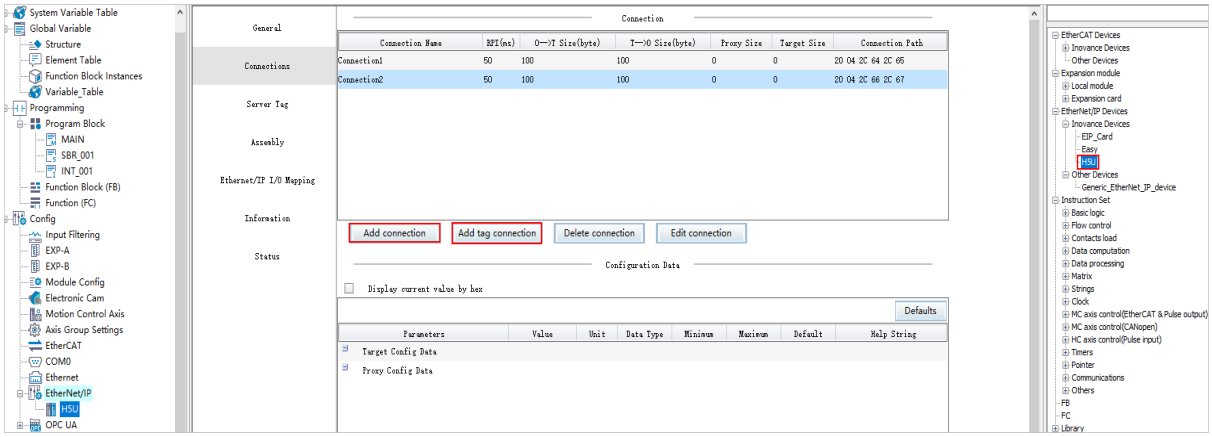
General Status Code	Extension Status Code	Status Description	
		English	Chinese
		A parameter associated with the request was invalid	Invalid parameter. The parameter does not meet specification requirements.
33	65535	An attempt was made to write to a write-once medium that has already been written, or to modify a value that cannot be changed once established	An attempt was made to write to a write-once medium that has already been written (such as the WORM driver and PROM), or to modify a value that cannot be changed once established.
34	65535	An invalid reply is received (e.g. reply service code does not match the request service code, or reply message is shorter than the minimum expected reply size).	An invalid reply is received.
37	65535	The Key segment that was included as the first segment in the path does not match the destination module	Error in electronic match in path
38	65535	The size of the path which was sent with the Service Request is either not large enough to allow the request to be routed or too much routing data was included	Invalid path size
39	65535	An attempt was made to set an attribute that is not able to be set at this time	An attempt was made to set an attribute that is not able to be set at this time.
40	65535	The Member ID specified in the request does not exist in the specified Class/ Instance/Attribute	Invalid member ID
41	65535	A request to modify a non-modifiable member was received	Member is not settable.
43	65535	An unknown Modbus error occurred	Unknown Modbus error
106	65535	An error in the encapsulation header	An error in the encapsulation header

12.3.5 Ethernet-IP Slave Application Example

1. Add two connections, and modify their O->T size to 500 bytes and T->O size to 500 bytes. Export the EDS file, set the IP address of the connected PLC to 10.45.124.150, compile the project, and then download and run the project.

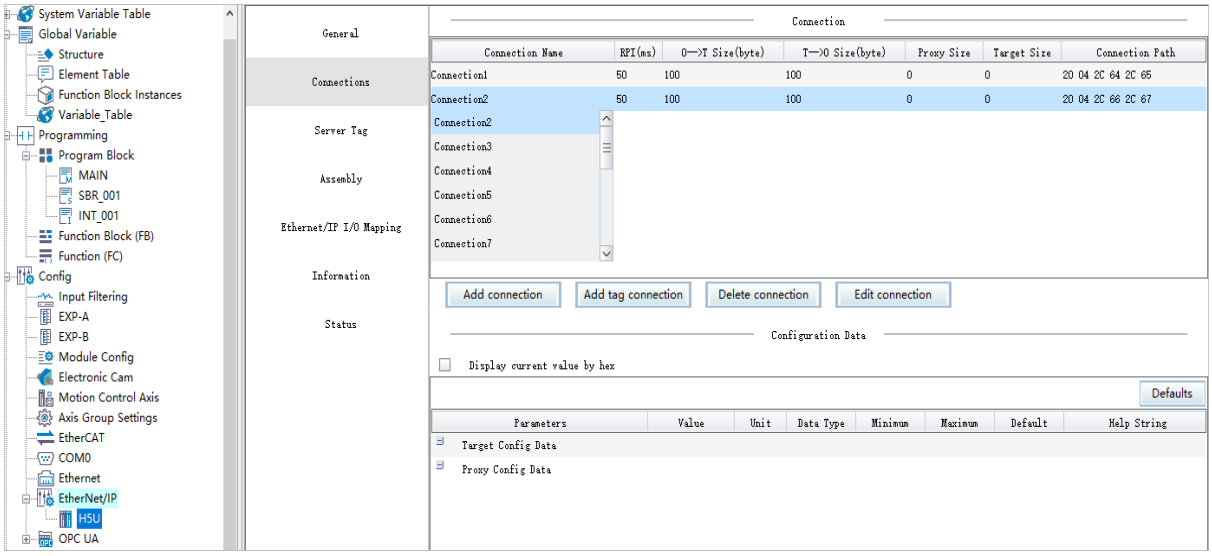


2. Import the exported EDS file. Double-click "H5U" under "EtherNet/IP" in the "Toolbox" section, add a slave station, and set the IP address of the connected PLC to 10.45.124.77.

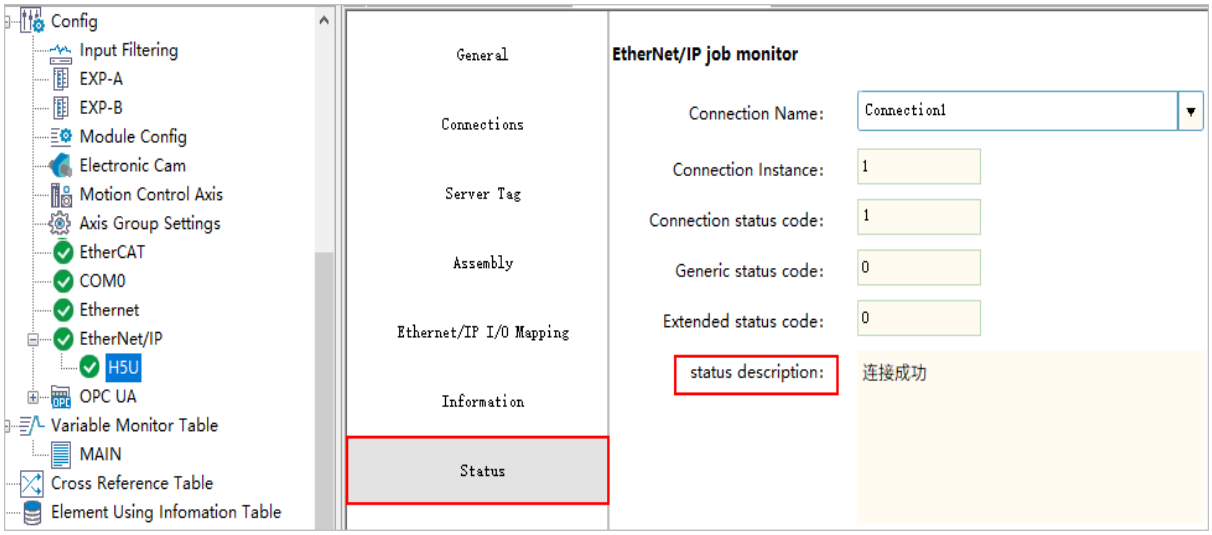


3. After the file is imported, "Connection1" is selected by default, whose O->T and T->O sizes are those you set when creating the connection.

   You can click "Add connection" to add a custom I/O connection or click "Add tag connection" to add a consumer tag connection. You can click "Connection Name" to unfold the drop-down list which displays connections defined in the EDS file. You can switch, edit, and delete connections in the drop-down list.



4. On the "General" tab page, set the IP address to 10.45.124.150, and then export and run the project.



5. If a green icon is displayed for the H5U node in the project tree, the node is connected. On the status page on which the connection status of "Connection1" is successful.

12.3.6 Tag Communication

12.3.6.1 Configuring Producer Tag Data

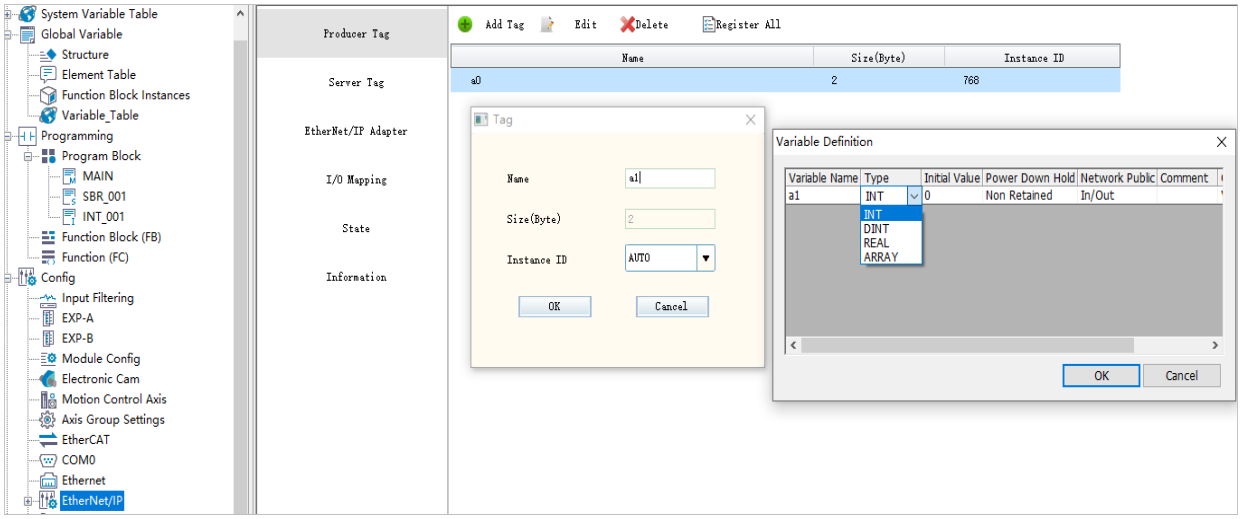
Introduction

Producer tags are used for tag communication based on the Class 1 communication mode. For details, see EtherNet/IP Communication Specifications.

Configuring the producer tag data on the "Producer Tag" page

1. In the "Project Manager" section, unfold "Config", and double-click "EtherNet/IP" to open the "Producer Tag" page. Supported producer tag types are INT, DINT, REAL, and array.
2. To create a producer tag, create a variable in the variable table. On the "Producer Tag" page, click "Add Tag", input the created variable, and then click "OK".

For data of the type INT, DINT, or REAL, you can input a name to create a tag on the "Producer Tag" page. Then, AutoShop automatically displays the variable creation dialog box, on which you can create the variable, as shown in the following figure.



Directly configuring the tag in the variable table

In the variable table, modify the "Network Public" property of the variable to "IN/OUT". Then, the variable is configured as a producer tag.

NO.	Variable...	Data Type	Initial Value	Power Down Hold	Network Public	Comment	Element Addr.	Length
1	intarr	INT[10]	...	Non Retained	Private			nBitLen:160

NOTE

To configure producer tags through the variable table in AutoShop V4.10.1.0 and later versions, you must register the variables on the producer tag page first. In the "Project Manager" tree, choose "EtherNET/IP" > "Producer Tag" > "Register All", select the variable, and then click "Register".

12.3.6.2 Ethernet-IP Consumer Tag Connection

Introduction

Consumer tags are used for tag communication based on the Class 1 communication mode. For details, see EtherNet/IP Communication Specifications.

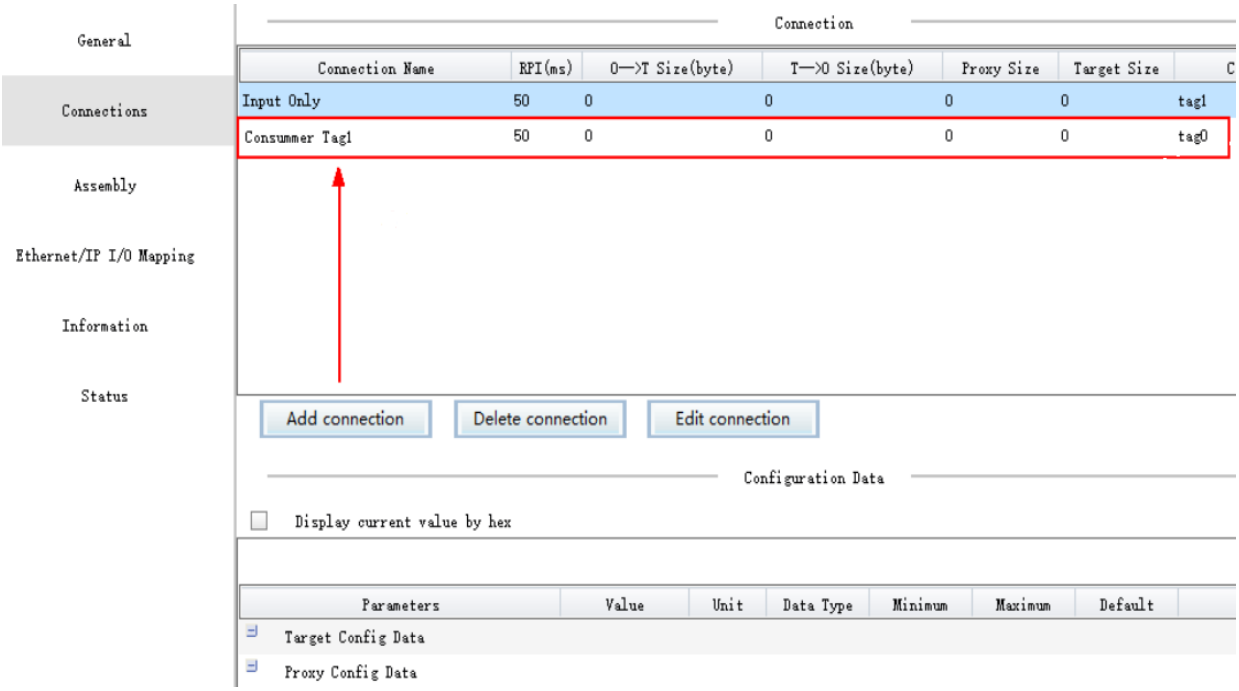
Configuring an EtherNet/IP consumer tag connection

NOTE

Before configuring a consumer or producer, add a slave and select a tag from the slave.

Take the EDS file of the Inovance slave as an example. In the toolbox, import the "AM600_400 Series PLC EIP Adapter.eds" file, add the configuration, modify the IP address, and then access the "Connection" page, on which a connection is loaded from the EDS file by default.

If the connection is a consumer tag connection, the "Add connection" button is activated and can be used to add a custom consumer tag connection. You can also click "Edit connection" to modify the RPI, T->O size, and connection path, wherein the connection path is the consumer tag name.



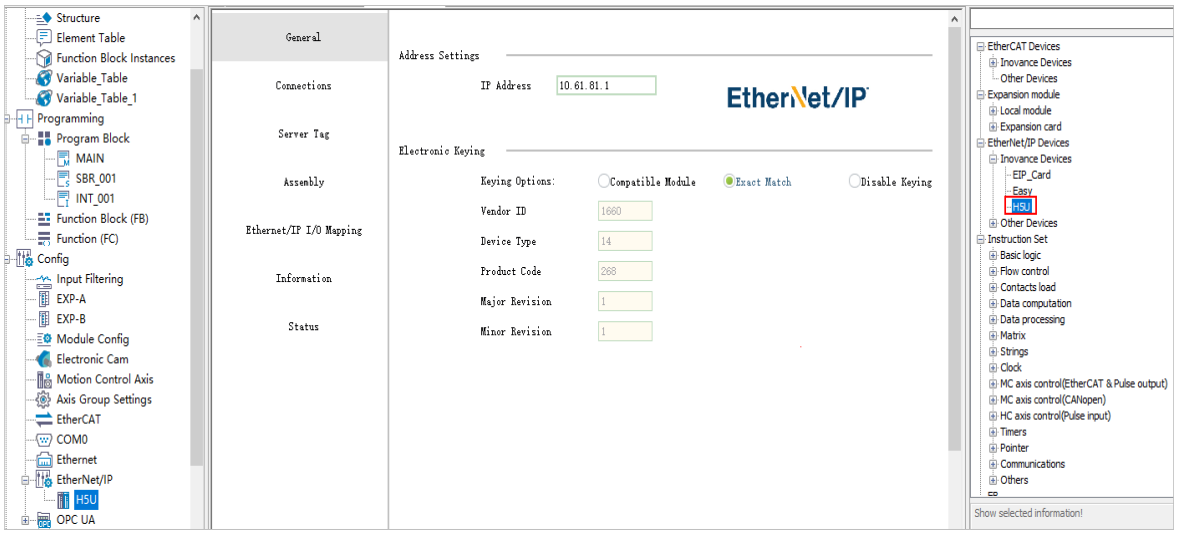
Methods for adding a slave

1. Import the EDS file.

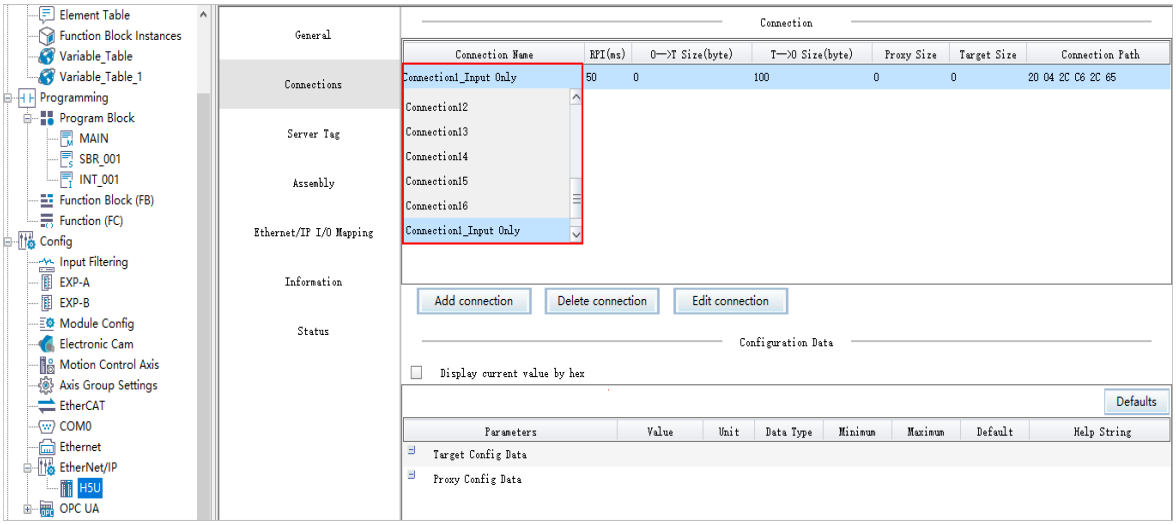
The following describes how to add a slave by taking the EDS file of the Inovance slave as an example.

a. Right-click "EtherNet/IP Devices" in the "Toolbox" section, click "Import EDS", and select the corresponding EDS file.

b. After the EDS file is imported, double-click the EDS file, and add the slave corresponding to EIP.

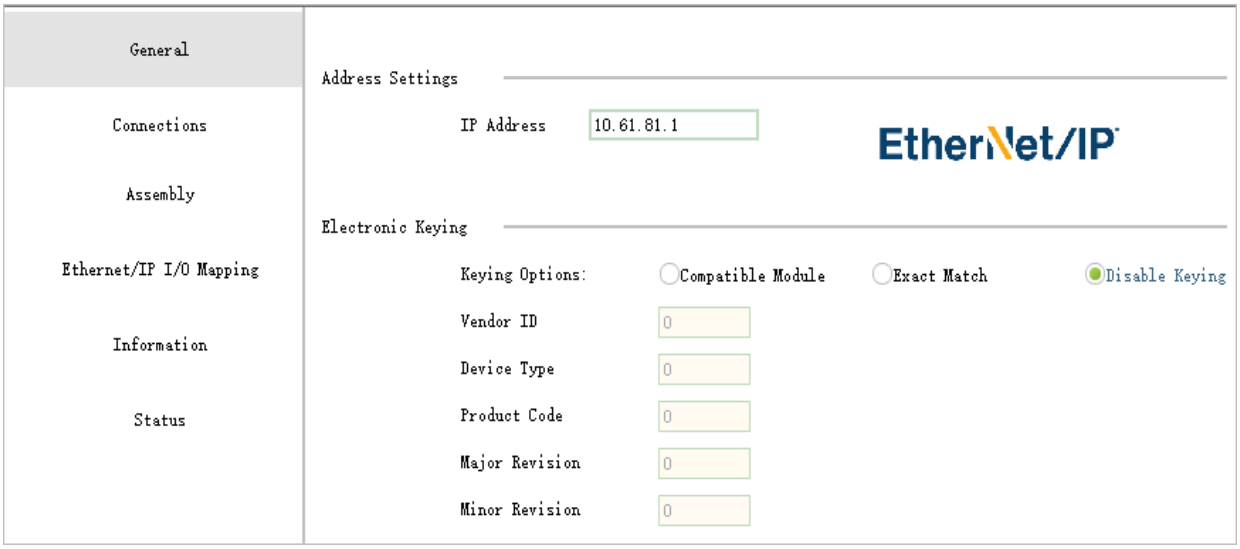


c. Select "Connection" and click the corresponding connection.



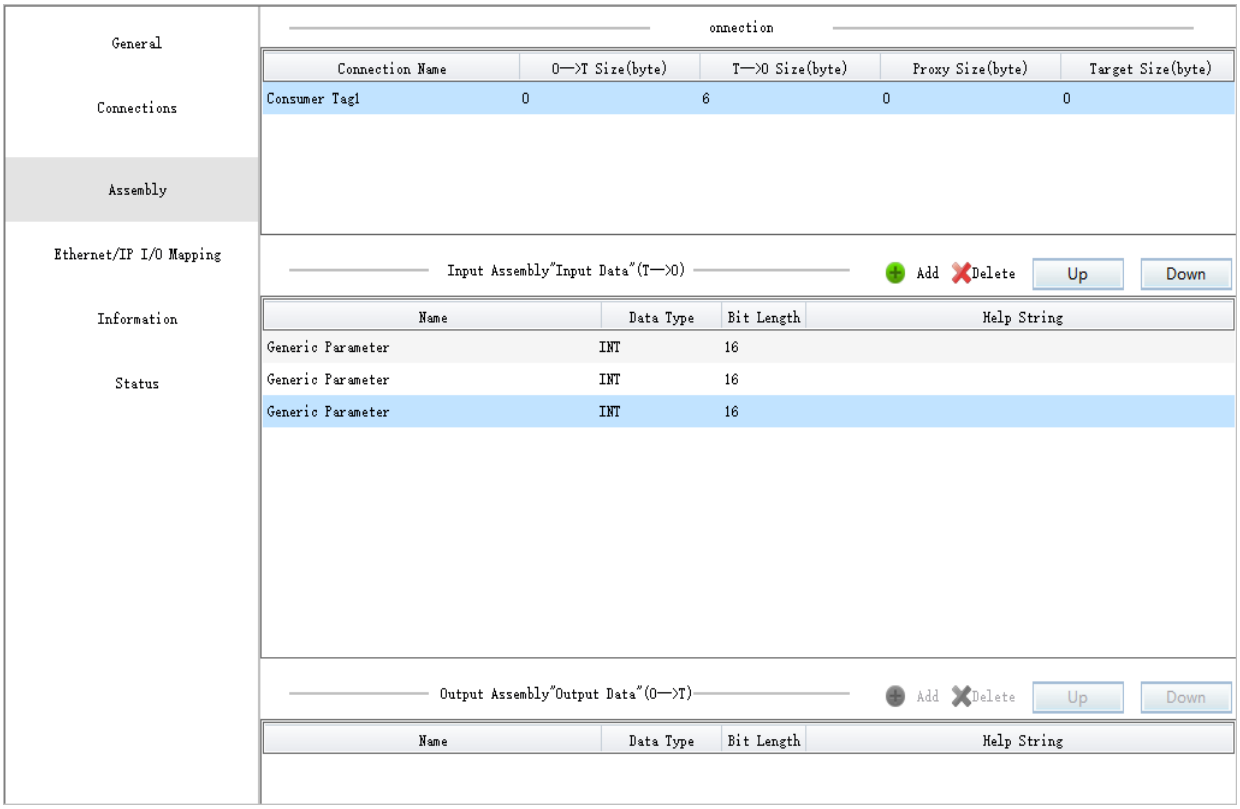
2. Use the generic template Generic_EtherNet_IP_device.

When no EDS file is provided for third-party device tag communication or the product ID of the current device does not match that in the EDS file, you can add this device and manually configure the device. On the "General" page, set all items of electronic matching to 0 and select "Disable Keying", as shown in the following figure.



12.3.6.3 Setting Tag Data Set

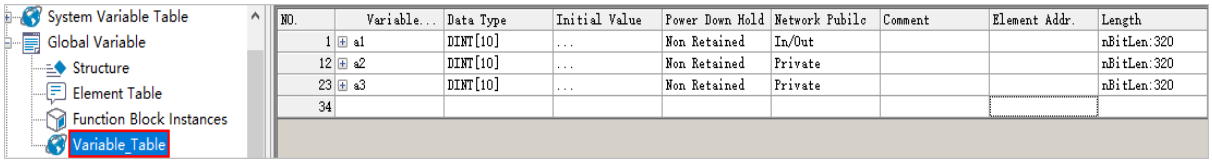
In the data area of tag communication, the parameter data types can be classified based on the T->O size. You can combine structure variable members to generate the corresponding I/O mapping.



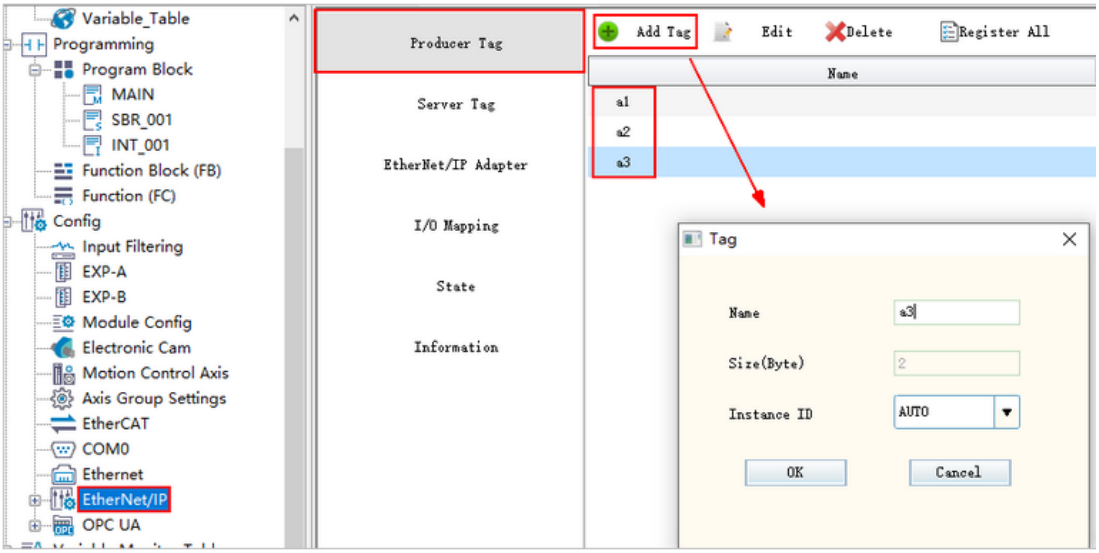
12.3.6.4 Tag Communication Application Example

The following takes the example of one H5U consuming tags a1, a2, and a3 created by another H5U to describe how to make the configuration.

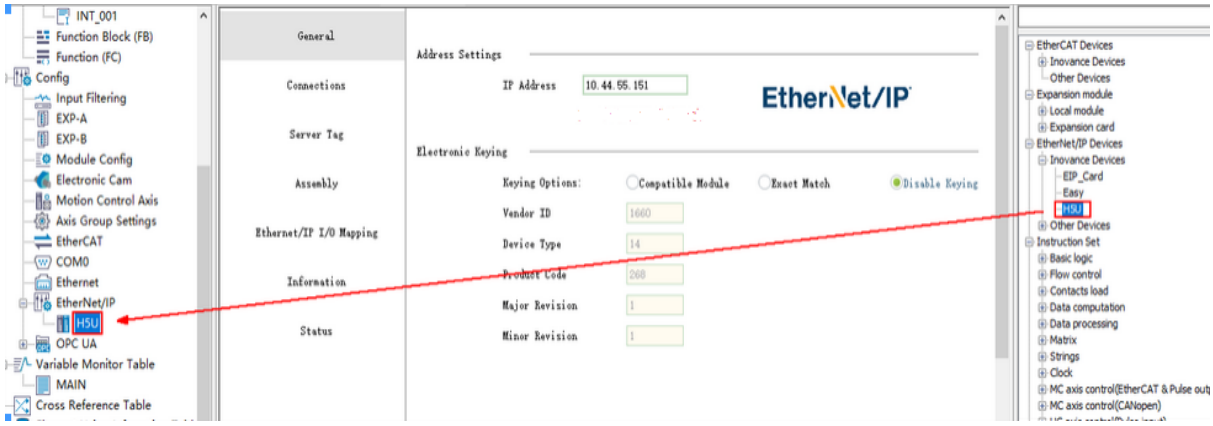
1. In the variable table, create the variable (producer). The configuration page is as follows.



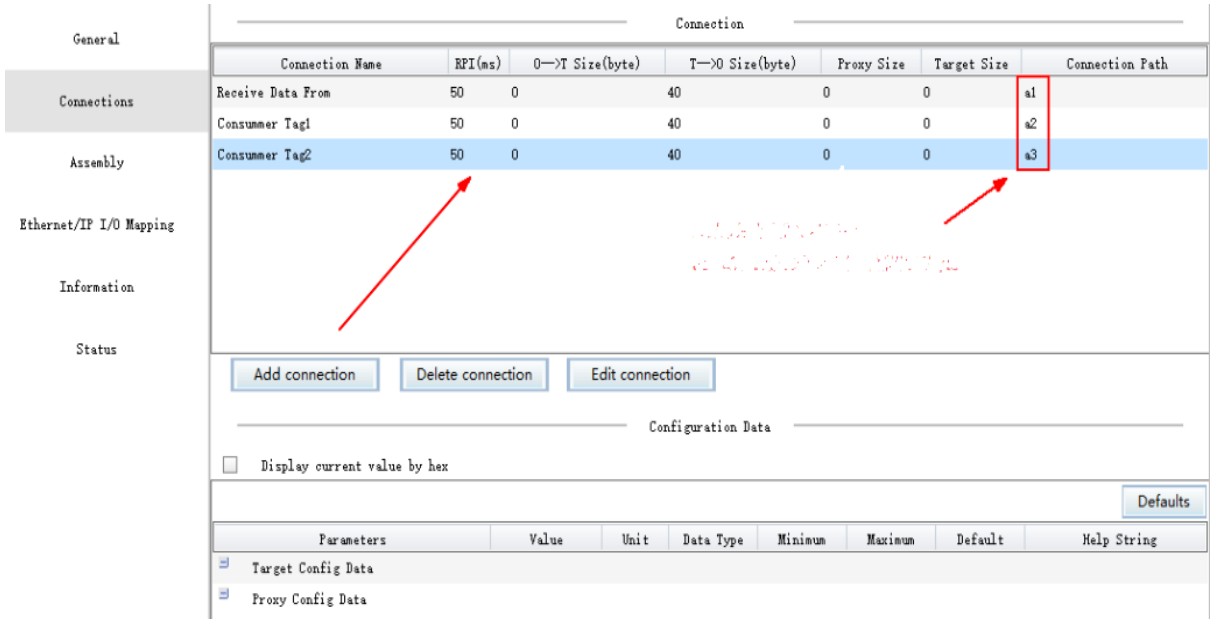
2. Associate the variable with the tag (producer).



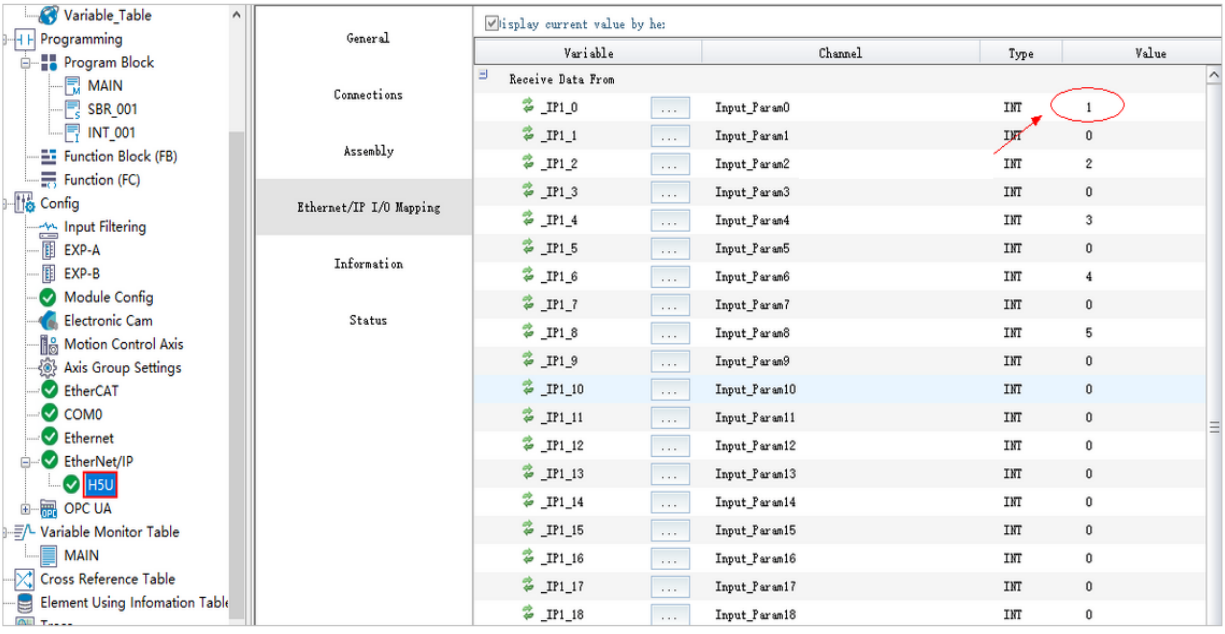
3. Access the configuration page (consumer).



4. Add connections to consumer tags corresponding to the tags a1, a2, and a3 of H5U.



After configuration, download the projects to the PLC to monitor their communication states.



12.4 Service Message Tag Communication

12.4.1 Service Message Tag: Server Configuration

Introduction

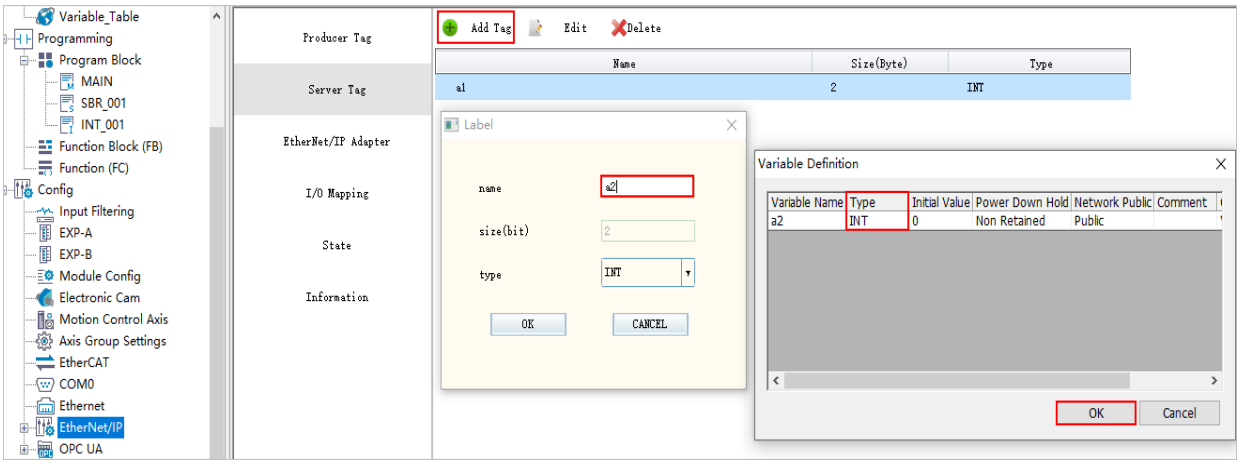
Service message tags are used for tag communication based on the Class 3 communication mode.

Procedure

Configuration on the "Server Tag" page

1. In the "Project Manager" section, unfold "Config", and double-click "EtherNet/IP" to open the "Server Tag" page. Data types supported by service message tags are INT, DINT, REAL, BOOL, BYTE, STRING, Stru, and array.
2. To create a service message tag, you can create a variable in the variable table. On the "Server Tag" page, click "Add Tag", input the created variable, and then click "OK".

You can also click "Add Tag" and input a name to create a tag on the "Server Tag" page. Then, AutoShop automatically displays the variable creation dialog box, as shown in the following figure. (*If you modify the variable of a created server tag in the variable table, an error is reported when you compile the variable directly. In this case, you can click "Edit" to refresh the data type of the variable associated with the tag.*)



Directly configuring the tag in the variable table

In the "Project Manager" tree, unfold "Global Variable" and then select "Variable_Table". In the variable table, modify the "Network Public" property of the variable to "Public". Then, the variable is configured as a service message tag.

NO.	Variable...	Data Type	Initial Value	Power Down Hold	Network Public	Comment	Element Addr.	Length
1	intarr	INT[10]	...	Non Retained	Public			nBitLen:160

12.4.2 Service Message Tag: Client Configuration

Introduction

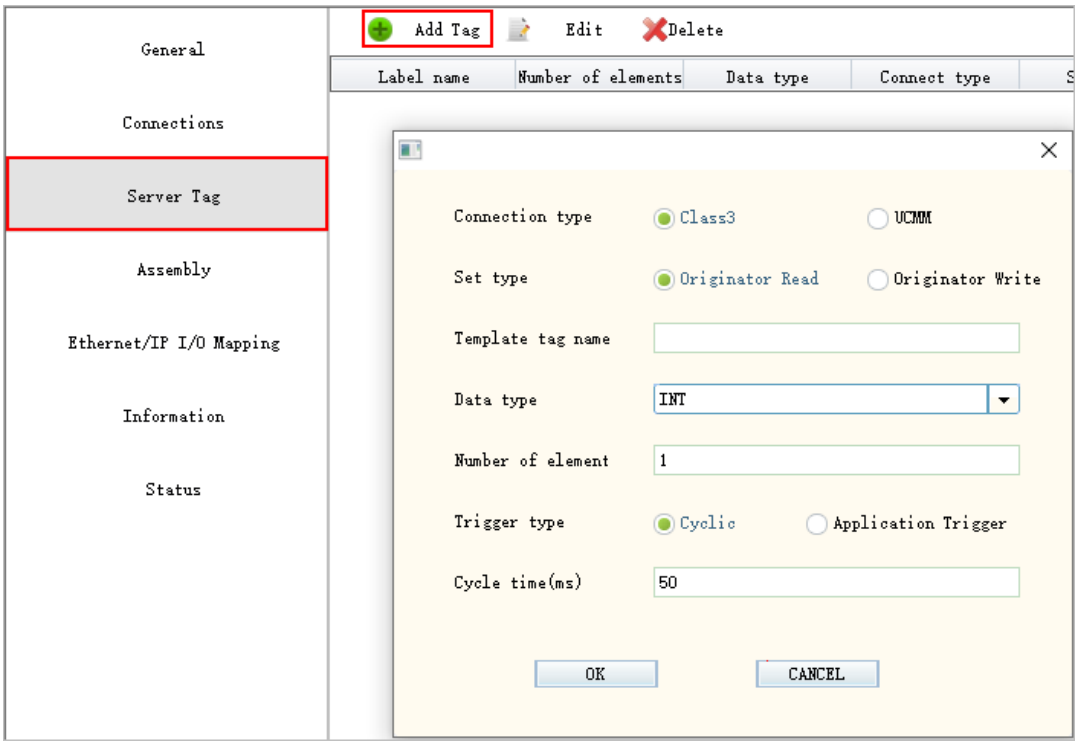
Service message tags are used for tag communication based on the Class 3 communication mode.

Procedure

The following takes service message tag communication between Inovance H5Us as an example to describe how to configure service message tags.

1. In the "Toolbox" section, double-click the EDS file of H5U, add the configuration, modify the IP address, and access the "Server Tag" page.

2. Click "Add Tag", and specify the name and data size of the service message tag you want to request, that is, the response service message tag created by the peer device to be connected.



You can click "Edit" to modify an added service message tag or double-click a tag in the list to edit the tag.

You can click "Delete" to delete an added service message tag.

The following section describes terms on the page of adding service message tag configuration.

- Connection type
  Class3 means a CIP connection is required for communication, that is, a connection-type tag communication.

  UCMM means a CIP connection is not required for communication, that is, connection-free tag communication.

- Set type
  Originator Read means reading the data of the target tag.

  Originator Write means writing data to the target tag.

- Template tag name: Indicates the name of the service message tag created by the requested device.
- Data type: Indicates the data type of the requested service message tag. Supported data types are INT, DINT, REAL, BOOL, BYTE, STRING, and UDT.
- Number of element: Indicates the number of arrays of the requested service message tag. When the number of elements is N (N is greater than 1), an array whose size is N and type is basic data is created.
- Trigger type
  Cyclic means data is requested periodically. The valid range of "Cycle time" is 5 ms to 1000 ms, and the default value is 50 ms.

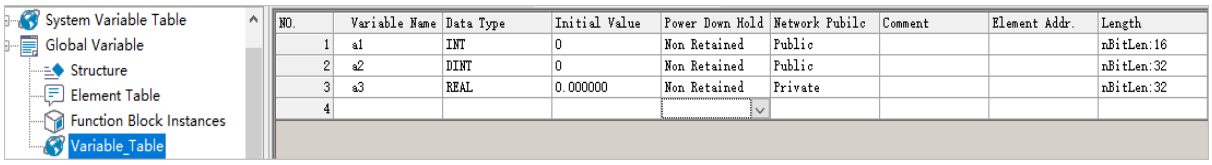
  Application Trigger indicates the status trigger and corresponds to the input variable. The request is triggered by the BOOL-type variables.

> **NOTE**
>
> Tags created in H5U and Easy series PLCs are of the array type. When the requested element number is 1, a subscript must be added to the target tag name to indicate which element is requested. When the number of requested element N is greater than 1, the subscript is K or no subscript is added, that is, requesting the Nth element counting from the subscript K or (0) by default.

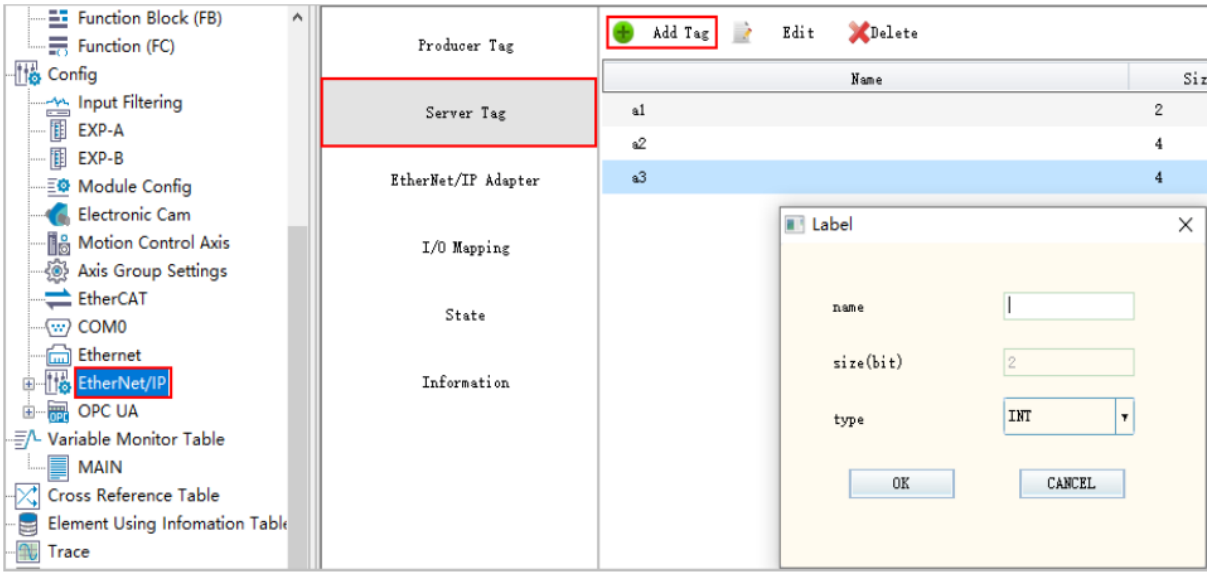
12.4.3 Application Example

The following takes the example of H5U-1 consuming tags a1, a2, and a3 created by H5U-2.

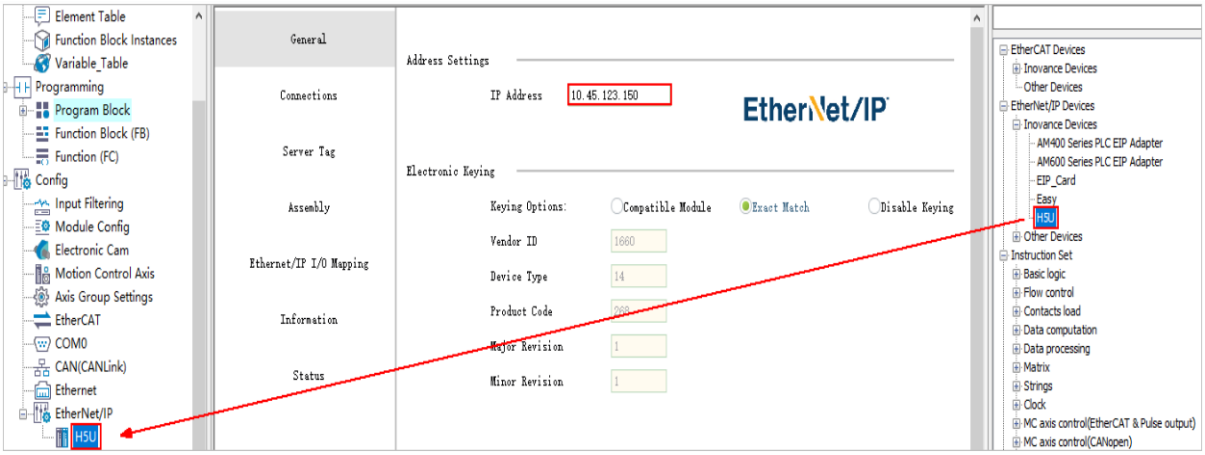
1. Response service message tag: In the variable table, create the variable (producer). The H5U-2 configuration page is as follows.

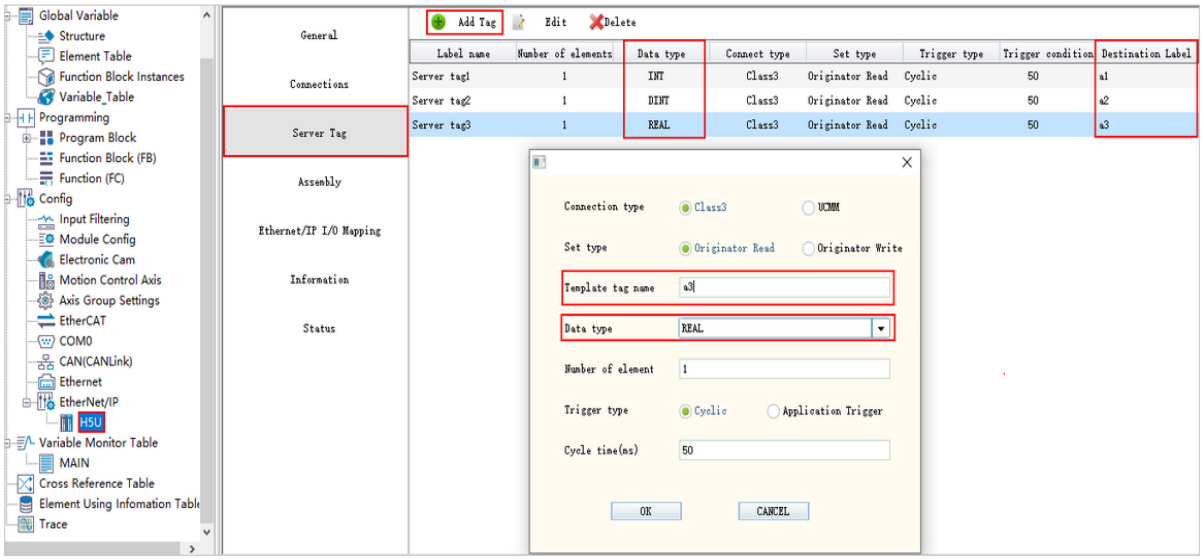


2. Response service message tag: In the H5U-2 project, associate the variable with the tag.

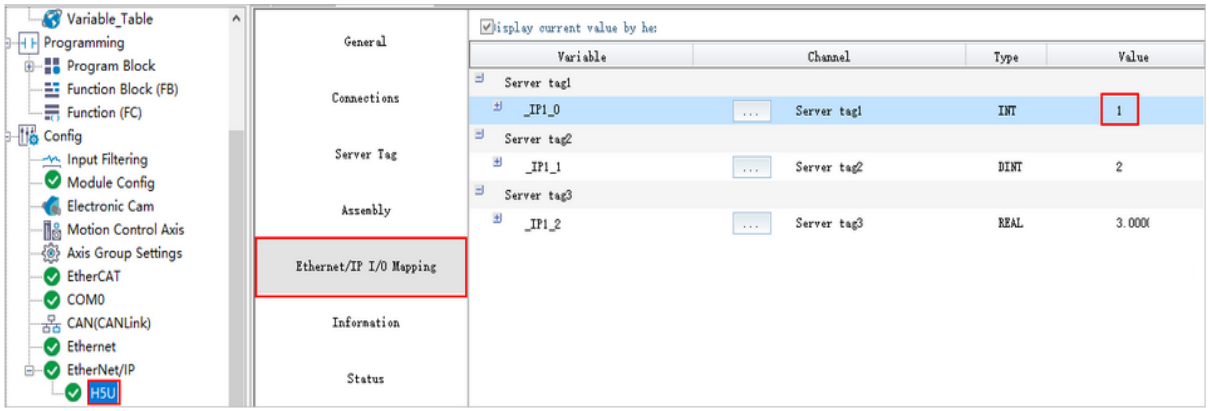


3. Request service message tag: In the H5U-1 project, configure the service message tag.





4. After configuration, download the projects to the PLC to monitor their communication states, and monitor the request data in the H5U-1 project. On the Ethernet/IP I/O mapping page, members of variables of UDT type cannot be unfolded. To view their monitored values, see the quick monitor table or the variable monitor table.



12.5 Fault Diagnosis

12.5.1 Loop Detection

Introduction

When loop detection is enabled, after a device is connected to the network, the system checks whether the device is in a loop in the current network. When a loop is detected, you can view and print the diagnosis and

log information in the "System Diagnosis" and "System Run Log" windows, which shows that a network loop exists and the loop connection must be disconnected.

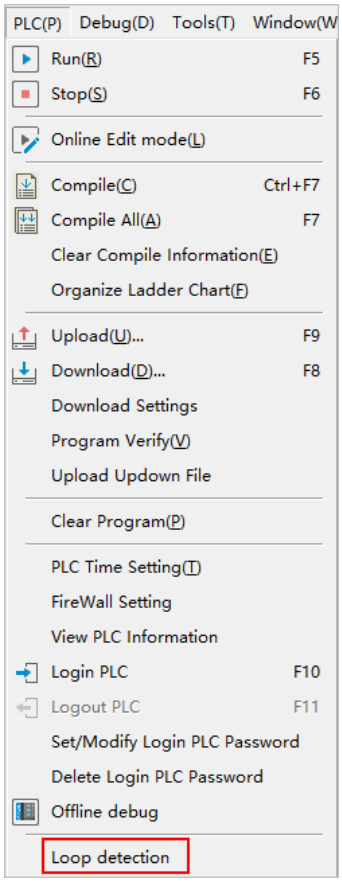
Preconditions

- Applicable model: Easy523
- Firmware version: V6.4.0.0 and later
- Software version: AutoShop V4.10.2.0 and later

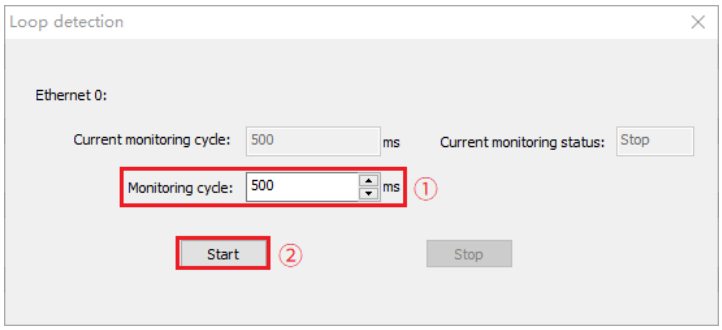
Procedure

Function configuration

1. In the menu bar, choose "PLC(P)" > "Loop detection". The "Loop detection" configuration page is displayed, as shown in the following figure.

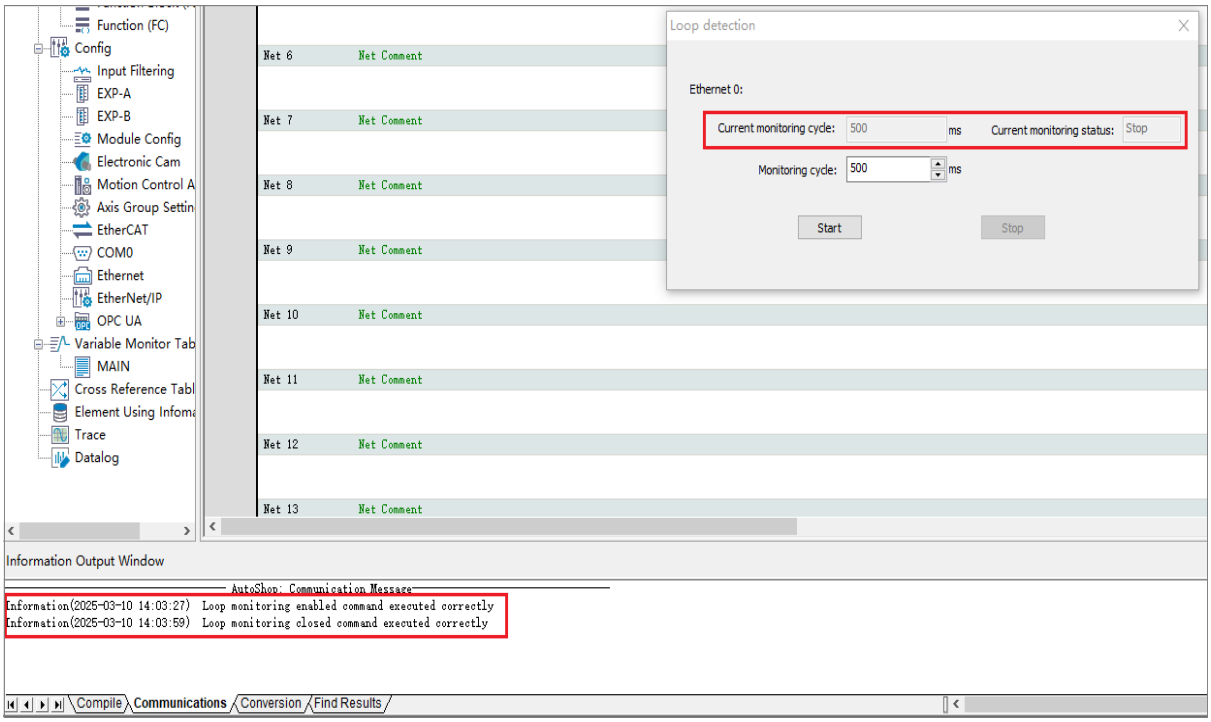


2. On this page, "Current monitoring cycle" and "Current monitoring status (ON/OFF)" are automatically identified and displayed. You can set "Monitoring cycle" to 500 ms at ①, and click "Start" at ② to enable loop detection, as shown in the following figure.

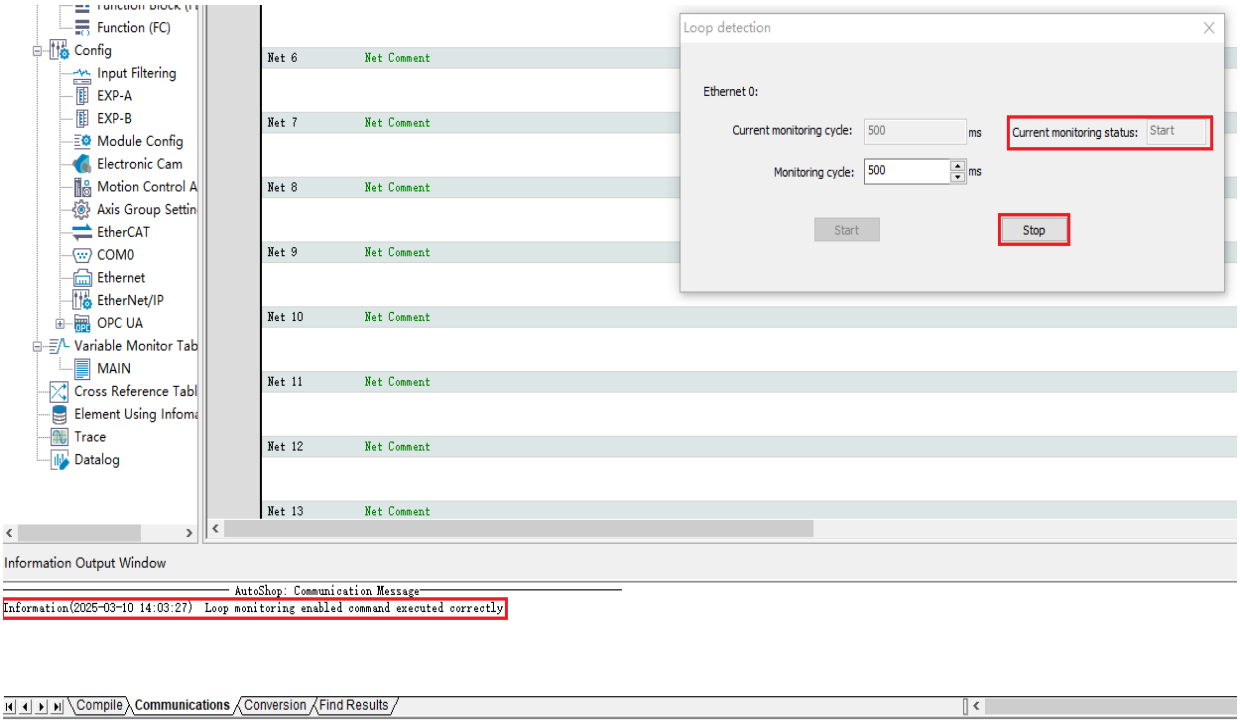


3. In the "Information Output Window" area, the prompt "Loop monitoring enabled command executed correctly". The current monitoring cycle in the "Loop detection" window is updated to the setpoint and the

current monitoring status is updated to "ON", as shown in the following figure.



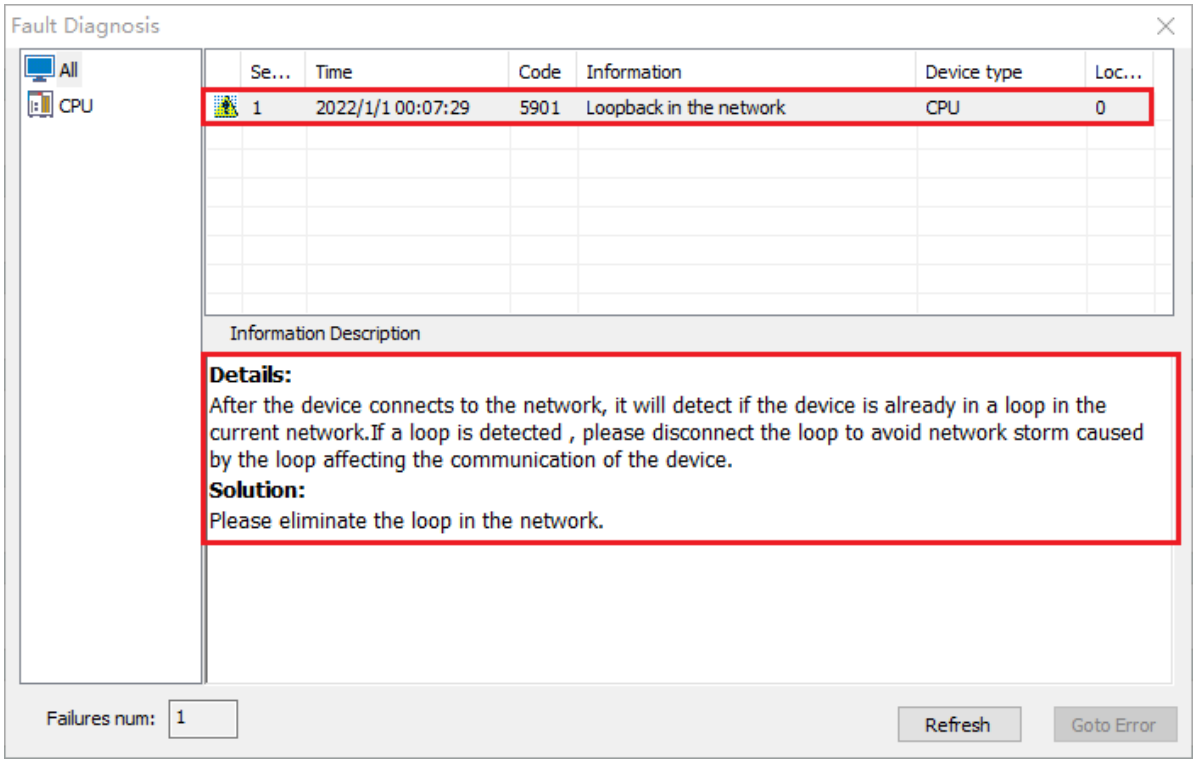
To disable the loop detection function, click "Stop" in the "Loop detection" window. Then, the prompt "Loop monitoring closed command executed correctly" is displayed in the "Information Output Window" area and the current monitoring status in the "Loop detection" window is updated to "OFF", as shown in the following figure.



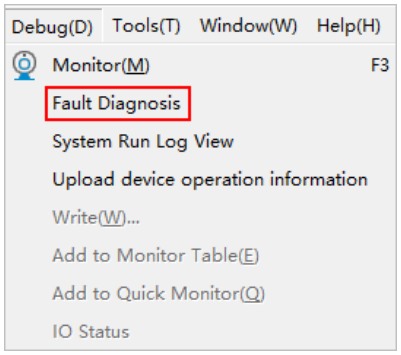
Viewing the alarm

You can view the loop alarm in any of the following two ways.

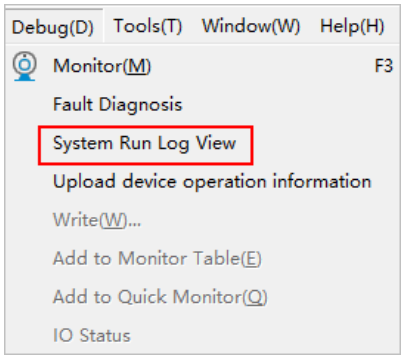
- In the menu bar, choose "Debug (D)" > "Fault Diagnosis" to view the loop alarm information, as shown in the following figure.

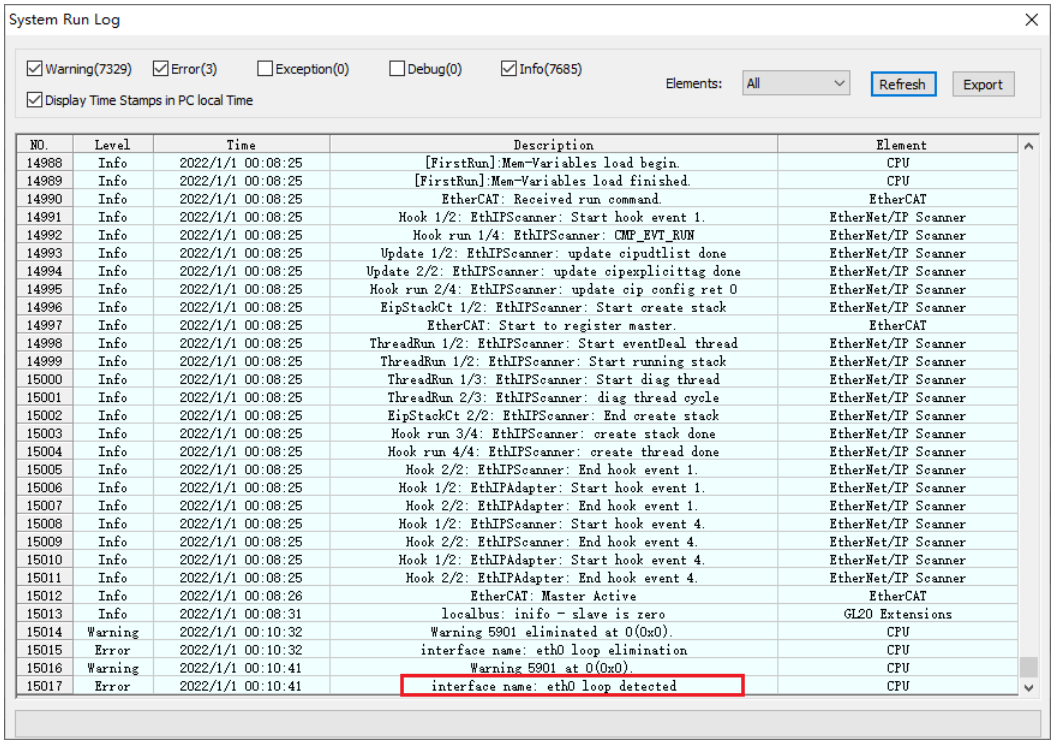


Click a diagnosis information item to view the details and solutions of a loop detection alarm, as shown in the following figure.



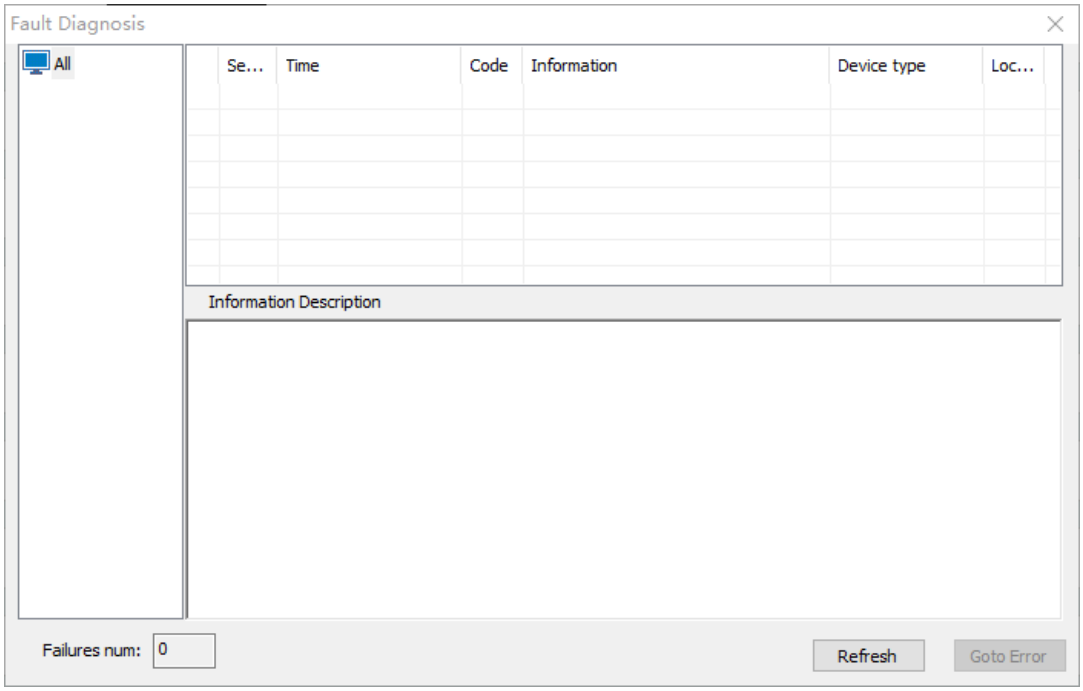
- In the menu bar, choose "Debug (D)" > "System Run Log View" to view the loop alarm information. When the system network has a loop, the alarm log indicating that Ethernet0 has a loop is printed, as shown in the following figure.

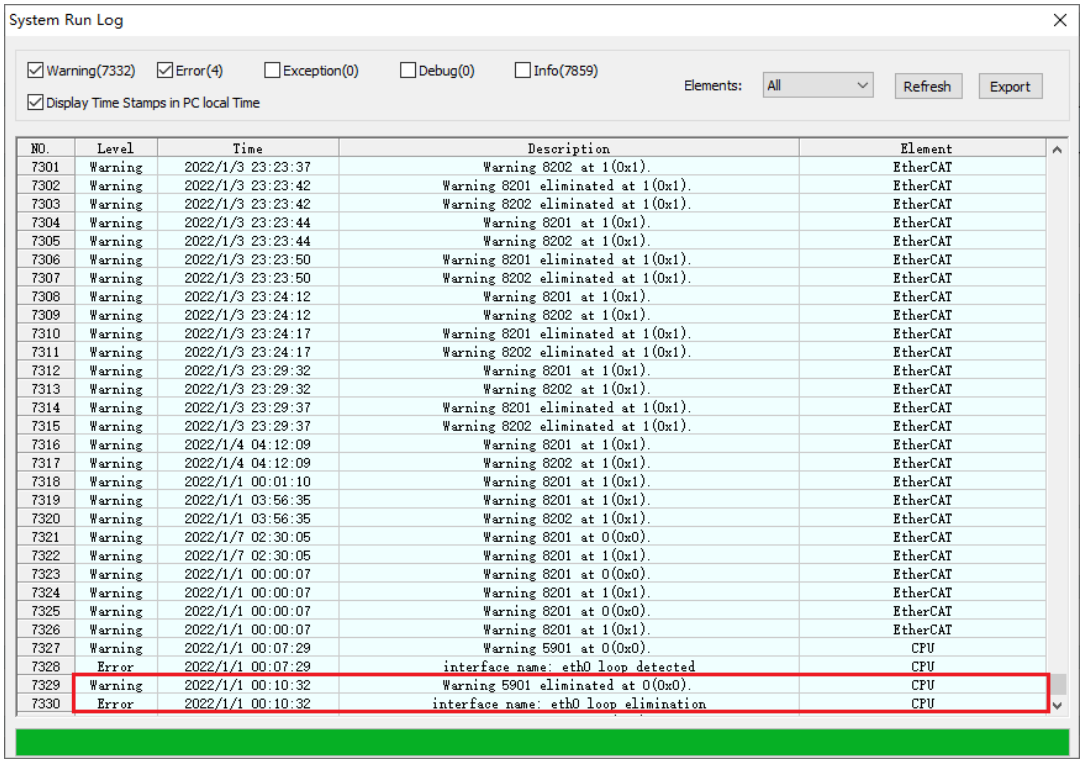




NO.	Level	Time	Description	Element
14988	Info	2022/1/1 00:08:25	[FirstRun]:Mem-Variables load begin.	CPU
14989	Info	2022/1/1 00:08:25	[FirstRun]:Mem-Variables load finished.	CPU
14990	Info	2022/1/1 00:08:25	EtherCAT: Received run command.	EtherCAT
14991	Info	2022/1/1 00:08:25	Hook 1/2: EthIPScanner: Start hook event 1.	EtherNet/IP Scanner
14992	Info	2022/1/1 00:08:25	Hook run 1/4: EthIPScanner: CMP_EVT_RUN	EtherNet/IP Scanner
14993	Info	2022/1/1 00:08:25	Update 1/2: EthIPScanner: update cipudtlist done	EtherNet/IP Scanner
14994	Info	2022/1/1 00:08:25	Update 2/2: EthIPScanner: update cipexplicittag done	EtherNet/IP Scanner
14995	Info	2022/1/1 00:08:25	Hook run 2/4: EthIPScanner: update cip config ret 0	EtherNet/IP Scanner
14996	Info	2022/1/1 00:08:25	EipStackCt 1/2: EthIPScanner: Start create stack	EtherNet/IP Scanner
14997	Info	2022/1/1 00:08:25	EtherCAT: Start to register master.	EtherCAT
14998	Info	2022/1/1 00:08:25	ThreadRun 1/2: EthIPScanner: Start eventDeal thread	EtherNet/IP Scanner
14999	Info	2022/1/1 00:08:25	ThreadRun 1/2: EthIPScanner: Start running stack	EtherNet/IP Scanner
15000	Info	2022/1/1 00:08:25	ThreadRun 1/3: EthIPScanner: Start diag thread	EtherNet/IP Scanner
15001	Info	2022/1/1 00:08:25	ThreadRun 2/3: EthIPScanner: diag thread cycle	EtherNet/IP Scanner
15002	Info	2022/1/1 00:08:25	EipStackCt 2/2: EthIPScanner: End create stack	EtherNet/IP Scanner
15003	Info	2022/1/1 00:08:25	Hook run 3/4: EthIPScanner: create stack done	EtherNet/IP Scanner
15004	Info	2022/1/1 00:08:25	Hook run 4/4: EthIPScanner: create thread done	EtherNet/IP Scanner
15005	Info	2022/1/1 00:08:25	Hook 2/2: EthIPScanner: End hook event 1.	EtherNet/IP Scanner
15006	Info	2022/1/1 00:08:25	Hook 1/2: EthIPAdapter: Start hook event 1.	EtherNet/IP Scanner
15007	Info	2022/1/1 00:08:25	Hook 2/2: EthIPAdapter: End hook event 1.	EtherNet/IP Scanner
15008	Info	2022/1/1 00:08:25	Hook 1/2: EthIPScanner: Start hook event 4.	EtherNet/IP Scanner
15009	Info	2022/1/1 00:08:25	Hook 2/2: EthIPScanner: End hook event 4.	EtherNet/IP Scanner
15010	Info	2022/1/1 00:08:25	Hook 1/2: EthIPAdapter: Start hook event 4.	EtherNet/IP Scanner
15011	Info	2022/1/1 00:08:25	Hook 2/2: EthIPAdapter: End hook event 4.	EtherNet/IP Scanner
15012	Info	2022/1/1 00:08:26	EtherCAT: Master Active	EtherCAT
15013	Info	2022/1/1 00:08:31	localbus: inifo - slave is zero	GL20 Extensions
15014	Warning	2022/1/1 00:10:32	Warning 5901 eliminated at 0(0x0).	CPU
15015	Error	2022/1/1 00:10:32	interface name: eth0 loop elimination	CPU
15016	Warning	2022/1/1 00:10:41	Warning 5901 at 0(0x0).	CPU
15017	Error	2022/1/1 00:10:41	interface name: eth0 loop detected	CPU

Disconnect the network with the loop. The network loop diagnosis information disappears and "eth0 loop elimination" is displayed in the system operation log, as shown in the following figure.





NO.	Level	Time	Description	Element
7301	Warning	2022/1/3 23:23:37	Warning 8202 at 1(0x1).	EtherCAT
7302	Warning	2022/1/3 23:23:42	Warning 8201 eliminated at 1(0x1).	EtherCAT
7303	Warning	2022/1/3 23:23:42	Warning 8202 eliminated at 1(0x1).	EtherCAT
7304	Warning	2022/1/3 23:23:44	Warning 8201 at 1(0x1).	EtherCAT
7305	Warning	2022/1/3 23:23:44	Warning 8202 at 1(0x1).	EtherCAT
7306	Warning	2022/1/3 23:23:50	Warning 8201 eliminated at 1(0x1).	EtherCAT
7307	Warning	2022/1/3 23:23:50	Warning 8202 eliminated at 1(0x1).	EtherCAT
7308	Warning	2022/1/3 23:24:12	Warning 8201 at 1(0x1).	EtherCAT
7309	Warning	2022/1/3 23:24:12	Warning 8202 at 1(0x1).	EtherCAT
7310	Warning	2022/1/3 23:24:17	Warning 8201 eliminated at 1(0x1).	EtherCAT
7311	Warning	2022/1/3 23:24:17	Warning 8202 eliminated at 1(0x1).	EtherCAT
7312	Warning	2022/1/3 23:29:32	Warning 8201 at 1(0x1).	EtherCAT
7313	Warning	2022/1/3 23:29:32	Warning 8202 at 1(0x1).	EtherCAT
7314	Warning	2022/1/3 23:29:37	Warning 8201 eliminated at 1(0x1).	EtherCAT
7315	Warning	2022/1/3 23:29:37	Warning 8202 eliminated at 1(0x1).	EtherCAT
7316	Warning	2022/1/4 04:12:09	Warning 8201 at 1(0x1).	EtherCAT
7317	Warning	2022/1/4 04:12:09	Warning 8202 at 1(0x1).	EtherCAT
7318	Warning	2022/1/1 00:01:10	Warning 8201 at 1(0x1).	EtherCAT
7319	Warning	2022/1/1 03:56:35	Warning 8201 at 1(0x1).	EtherCAT
7320	Warning	2022/1/1 03:56:35	Warning 8202 at 1(0x1).	EtherCAT
7321	Warning	2022/1/7 02:30:05	Warning 8201 at 0(0x0).	EtherCAT
7322	Warning	2022/1/7 02:30:05	Warning 8201 at 1(0x1).	EtherCAT
7323	Warning	2022/1/1 00:00:07	Warning 8201 at 0(0x0).	EtherCAT
7324	Warning	2022/1/1 00:00:07	Warning 8201 at 1(0x1).	EtherCAT
7325	Warning	2022/1/1 00:00:07	Warning 8201 at 0(0x0).	EtherCAT
7326	Warning	2022/1/1 00:00:07	Warning 8201 at 1(0x1).	EtherCAT
7327	Warning	2022/1/1 00:07:29	Warning 5901 at 0(0x0).	CPU
7328	Error	2022/1/1 00:07:29	interface name: eth0 loop detected	CPU
7329	Warning	2022/1/1 00:10:32	Warning 5901 eliminated at 0(0x0).	CPU
7330	Error	2022/1/1 00:10:32	interface name: eth0 loop elimination	CPU

12.5.2 Address Conflict Detection

Introduction

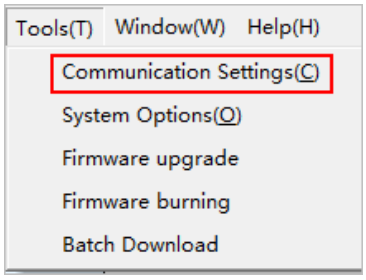
The address conflict detection function is used to check whether the IP address of the network adapter of a device is occupied by other devices in the current network after this device is connected to the network, its IP address is modified, or the Ethernet port cable is removed. If the IP address has been occupied, check the alarm information through troubleshooting or by referring to the system operation log page.

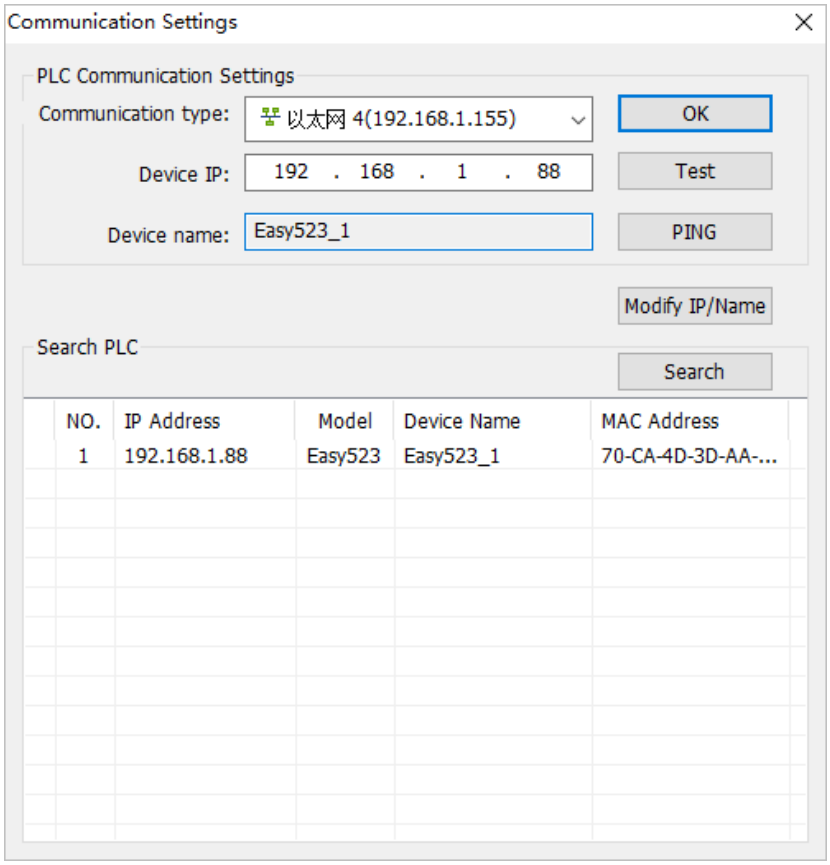
Preconditions

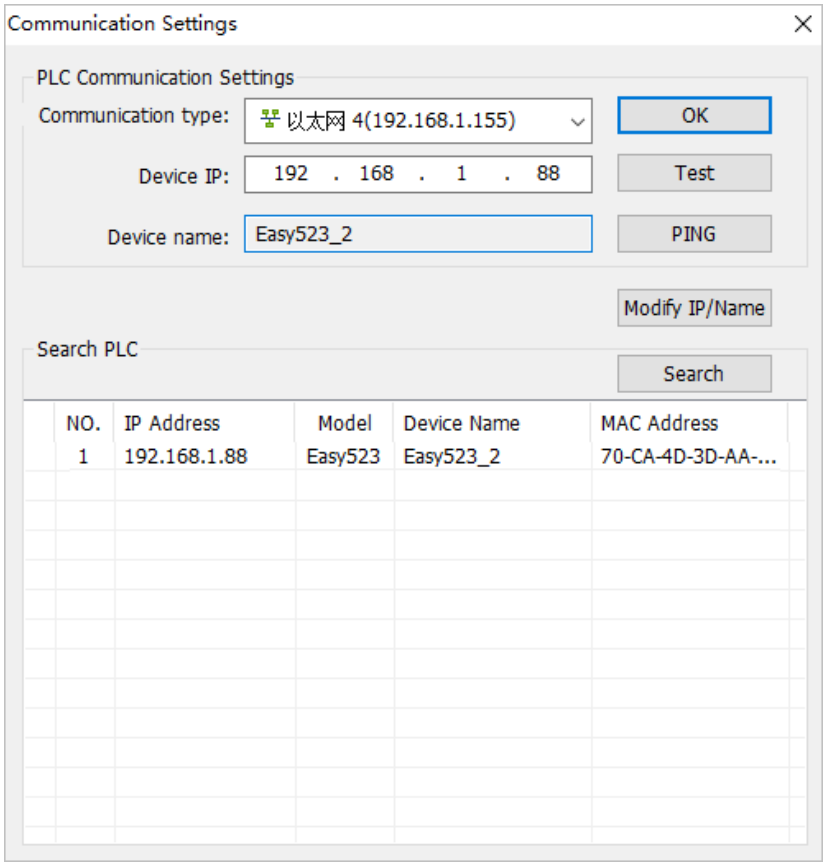
- Applicable model: Easy523
- Firmware version: V6.4.0.0 and later
- Software version: AutoShop V4.10.2.0 and later

Procedure

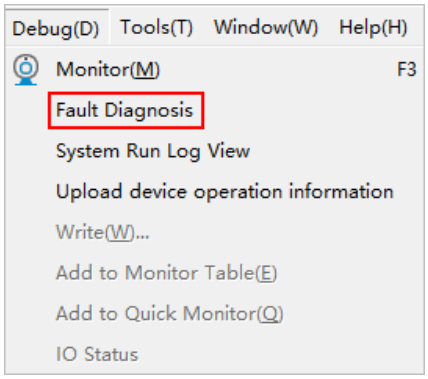
1. In the menu bar, choose "Tools (T)" > "Communication Settings". On the page displayed, modify the IP addresses of two Easy523 PLCS to 192.168.1.88. Then, only one device with the IP address 192.168.1.88 can be detected in the Ethernet, as shown in the following figure.

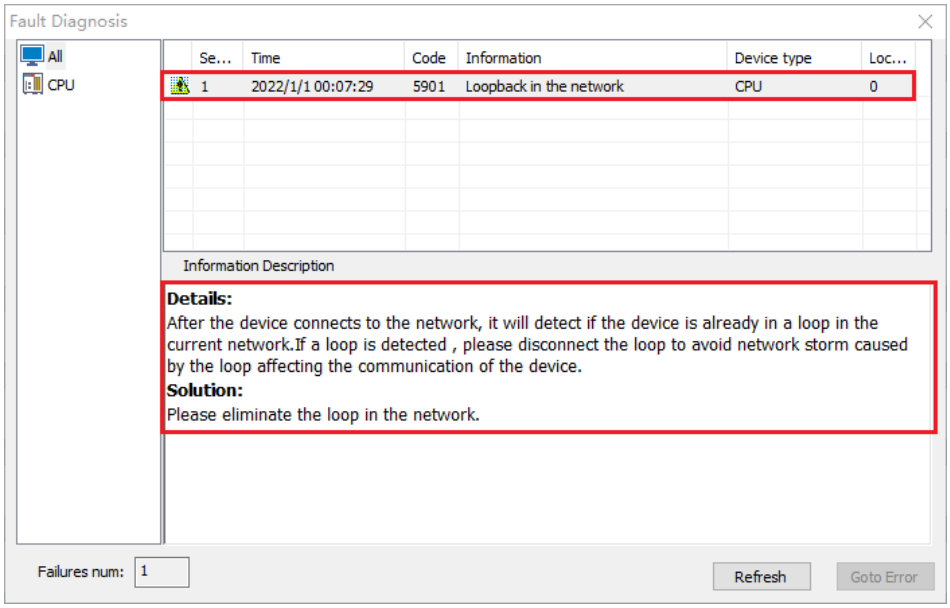




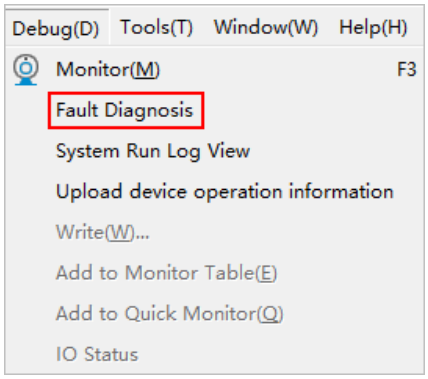


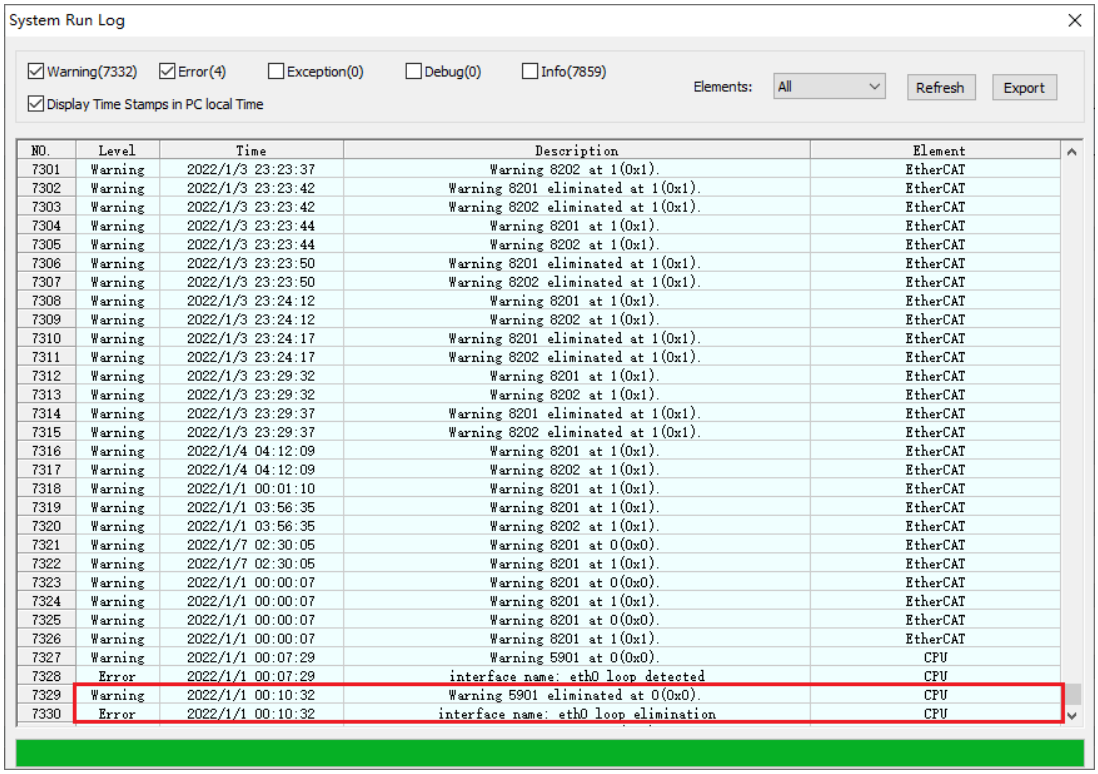
2. You can view the diagnosis information of "Ethernet IP address conflict" in any of the following two ways. In the menu bar, choose "Debug (D)" > "Fault Diagnosis" to view the diagnosis information of "Ethernet IP address conflict", as shown in the following figure.





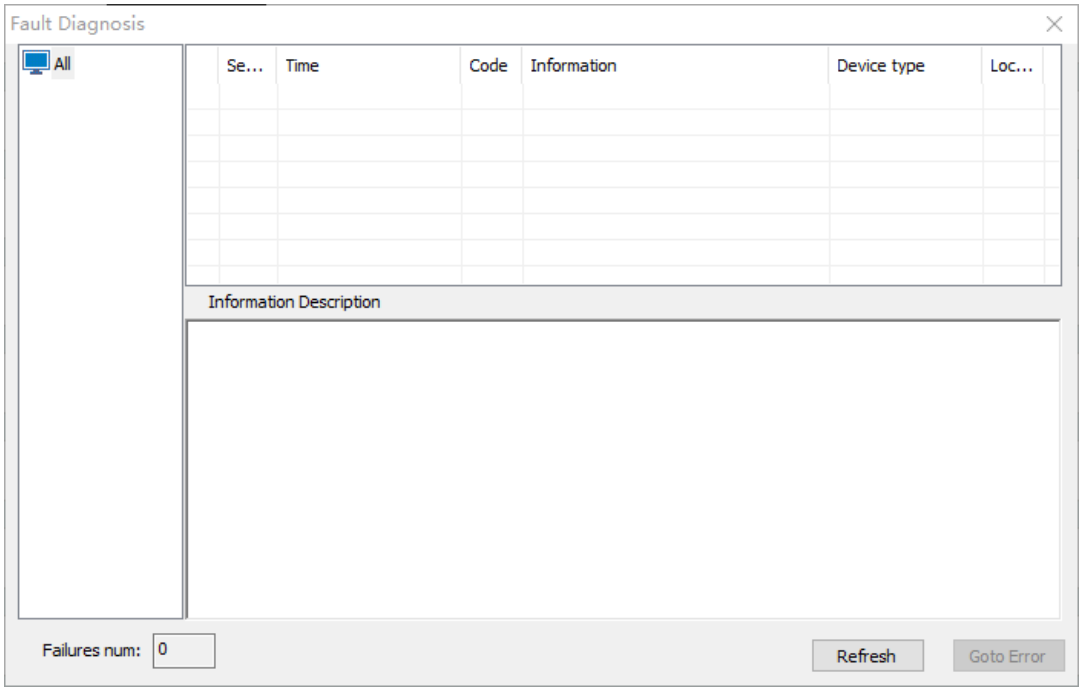
In the software, choose "Debug (D)" > "System Run Log View" in the menu bar at the top to view the diagnosis information of "Ethernet IP address conflict". Conflicting IP address, network adapter name, and MAC address are displayed in the logs, as shown in the following figure.





NO.	Level	Time	Description	Element
7301	Warning	2022/1/3 23:23:37	Warning 8202 at 1(0x1).	EtherCAT
7302	Warning	2022/1/3 23:23:42	Warning 8201 eliminated at 1(0x1).	EtherCAT
7303	Warning	2022/1/3 23:23:42	Warning 8202 eliminated at 1(0x1).	EtherCAT
7304	Warning	2022/1/3 23:23:44	Warning 8201 at 1(0x1).	EtherCAT
7305	Warning	2022/1/3 23:23:44	Warning 8202 at 1(0x1).	EtherCAT
7306	Warning	2022/1/3 23:23:50	Warning 8201 eliminated at 1(0x1).	EtherCAT
7307	Warning	2022/1/3 23:23:50	Warning 8202 eliminated at 1(0x1).	EtherCAT
7308	Warning	2022/1/3 23:24:12	Warning 8201 at 1(0x1).	EtherCAT
7309	Warning	2022/1/3 23:24:12	Warning 8202 at 1(0x1).	EtherCAT
7310	Warning	2022/1/3 23:24:17	Warning 8201 eliminated at 1(0x1).	EtherCAT
7311	Warning	2022/1/3 23:24:17	Warning 8202 eliminated at 1(0x1).	EtherCAT
7312	Warning	2022/1/3 23:29:32	Warning 8201 at 1(0x1).	EtherCAT
7313	Warning	2022/1/3 23:29:32	Warning 8202 at 1(0x1).	EtherCAT
7314	Warning	2022/1/3 23:29:37	Warning 8201 eliminated at 1(0x1).	EtherCAT
7315	Warning	2022/1/3 23:29:37	Warning 8202 eliminated at 1(0x1).	EtherCAT
7316	Warning	2022/1/4 04:12:09	Warning 8201 at 1(0x1).	EtherCAT
7317	Warning	2022/1/4 04:12:09	Warning 8202 at 1(0x1).	EtherCAT
7318	Warning	2022/1/1 00:01:10	Warning 8201 at 1(0x1).	EtherCAT
7319	Warning	2022/1/1 03:56:35	Warning 8201 at 1(0x1).	EtherCAT
7320	Warning	2022/1/1 03:56:35	Warning 8202 at 1(0x1).	EtherCAT
7321	Warning	2022/1/7 02:30:05	Warning 8201 at 0(0x0).	EtherCAT
7322	Warning	2022/1/7 02:30:05	Warning 8201 at 1(0x1).	EtherCAT
7323	Warning	2022/1/1 00:00:07	Warning 8201 at 0(0x0).	EtherCAT
7324	Warning	2022/1/1 00:00:07	Warning 8201 at 1(0x1).	EtherCAT
7325	Warning	2022/1/1 00:00:07	Warning 8201 at 0(0x0).	EtherCAT
7326	Warning	2022/1/1 00:00:07	Warning 8201 at 1(0x1).	EtherCAT
7327	Warning	2022/1/1 00:07:29	Warning 5901 at 0(0x0).	CPU
7328	Error	2022/1/1 00:07:29	interface name: eth0 loop detected	CPU
7329	Warning	2022/1/1 00:10:32	Warning 5901 eliminated at 0(0x0).	CPU
7330	Error	2022/1/1 00:10:32	interface name: eth0 loop elimination	CPU

3. Correct the conflicting IP address. Then, the diagnosis information of "Ethernet IP address conflict" disappears from the "Fault Diagnosis" window and the "System Run Log" window, a shown in the following figure.



System Run Log

Warning(7329) Error(3) Exception(0) Debug(0) Info(7685)

Display Time Stamps in PC local Time

Elements: All Refresh Export

NO.	Level	Time	Description	Element
14988	Info	2022/1/1 00:08:25	[FirstRun]:Mem-Variables load begin.	CPU
14989	Info	2022/1/1 00:08:25	[FirstRun]:Mem-Variables load finished.	CPU
14990	Info	2022/1/1 00:08:25	EtherCAT: Received run command.	EtherCAT
14991	Info	2022/1/1 00:08:25	Hook 1/2: EthIPScanner: Start hook event 1.	EtherNet/IP Scanner
14992	Info	2022/1/1 00:08:25	Hook run 1/4: EthIPScanner: CMP_EVT_RUN	EtherNet/IP Scanner
14993	Info	2022/1/1 00:08:25	Update 1/2: EthIPScanner: update cipudtlist done	EtherNet/IP Scanner
14994	Info	2022/1/1 00:08:25	Update 2/2: EthIPScanner: update cipexplicittag done	EtherNet/IP Scanner
14995	Info	2022/1/1 00:08:25	Hook run 2/4: EthIPScanner: update cip config ret 0	EtherNet/IP Scanner
14996	Info	2022/1/1 00:08:25	EipStackCt 1/2: EthIPScanner: Start create stack	EtherNet/IP Scanner
14997	Info	2022/1/1 00:08:25	EtherCAT: Start to register master.	EtherCAT
14998	Info	2022/1/1 00:08:25	ThreadRun 1/2: EthIPScanner: Start eventDeal thread	EtherNet/IP Scanner
14999	Info	2022/1/1 00:08:25	ThreadRun 1/2: EthIPScanner: Start running stack	EtherNet/IP Scanner
15000	Info	2022/1/1 00:08:25	ThreadRun 1/3: EthIPScanner: Start diag thread	EtherNet/IP Scanner
15001	Info	2022/1/1 00:08:25	ThreadRun 2/3: EthIPScanner: diag thread cycle	EtherNet/IP Scanner
15002	Info	2022/1/1 00:08:25	EipStackCt 2/2: EthIPScanner: End create stack	EtherNet/IP Scanner
15003	Info	2022/1/1 00:08:25	Hook run 3/4: EthIPScanner: create stack done	EtherNet/IP Scanner
15004	Info	2022/1/1 00:08:25	Hook run 4/4: EthIPScanner: create thread done	EtherNet/IP Scanner
15005	Info	2022/1/1 00:08:25	Hook 2/2: EthIPScanner: End hook event 1.	EtherNet/IP Scanner
15006	Info	2022/1/1 00:08:25	Hook 1/2: EthIPAdapter: Start hook event 1.	EtherNet/IP Scanner
15007	Info	2022/1/1 00:08:25	Hook 2/2: EthIPAdapter: End hook event 1.	EtherNet/IP Scanner
15008	Info	2022/1/1 00:08:25	Hook 1/2: EthIPScanner: Start hook event 4.	EtherNet/IP Scanner
15009	Info	2022/1/1 00:08:25	Hook 2/2: EthIPScanner: End hook event 4.	EtherNet/IP Scanner
15010	Info	2022/1/1 00:08:25	Hook 1/2: EthIPAdapter: Start hook event 4.	EtherNet/IP Scanner
15011	Info	2022/1/1 00:08:25	Hook 2/2: EthIPAdapter: End hook event 4.	EtherNet/IP Scanner
15012	Info	2022/1/1 00:08:26	EtherCAT: Master Active	EtherCAT
15013	Info	2022/1/1 00:08:31	localbus: inifo - slave is zero	GL20 Extensions
15014	Warning	2022/1/1 00:10:32	Warning 5901 eliminated at 0(0x0).	CPU
15015	Error	2022/1/1 00:10:32	interface name: eth0 loop elimination	CPU
15016	Warning	2022/1/1 00:10:41	Warning 5901 at 0(0x0).	CPU
15017	Error	2022/1/1 00:10:41	interface name: eth0 loop detected	CPU

13 PROFINET Communication

13.1 Overview

About PROFINET

PROFINET, launched by PROFIBUS International (PI), is a new generation of automation bus standards based on the industrial Ethernet technology.

PROFINET provides a complete network solution for automation communication, encompassing hot topics in automation such as real-time Ethernet, motion control, distributed automation, fail-safe and network security. As a cross-supplier technology, it is fully compatible with industrial Ethernet and existing fieldbus (such as PROFIBUS) technologies.

Extension modules supported by H5U and Easy series PLCs

H5U and Easy series PLCs support connection to Siemens PLCs through the PROFINET protocol. Extension modules of H5U and Easy series PLCs are operated through the TIA Portal (used as an example in this section), STEP7 MicroWIN, and other software and data is mapped to H5U and Easy series PLCs. H5U and Easy series devices can only be used as PROFINET slaves, and only support the reading and writing of process data of extension modules.

Easy301 and Easy302 do not support the PROFINET communication protocol.

- H5U devices with AutoShop of V4.8.2.0 and later versions and PLC firmware of V6.1.0.0 and later versions support the PROFINET protocol.
- H5U and Easy devices with AutoShop of V4.10.0.0 and later versions and PLC firmware of V6.2.0.0 and later versions support the PROFINET protocol.

- The following table lists extension modules supported by H5U devices.

Module Name	Input (Bytes)	Output (Bytes)	Description
0016ETN	-	2	A module with 16 DO terminals (transistor NPN output)
0016ETP	-	2	A module with 16 DO terminals (transistor PNP output)
0016ER	-	2	A module with 16 DO terminals (relay output)
1600END	2	-	A module with 16 DI terminals
3200END	4	-	An input module with 32 channels

Module Name	Input (Bytes)	Output (Bytes)	Description
0032ETN	-	4	An input module with 32 channels
4DA	-	8	A module with 4 DA terminals, supporting voltage and current output
4AD	8	-	A module with 4 AD terminals, supporting voltage and current input
8TC	16	-	8-channel temperature module
4TC	8	-	4-channel temperature module
4PT	8	-	4-channel temperature module
Share IN2 BOOL	2	-	Remote module input data area
Share IN32 DINT	32	-	Remote module input data area
Share IN32 INT	32	-	Remote module input data area
Share IN32 REAL	32	-	Remote module input data area
Share IN64 DINT	64	-	Remote module input data area
Share IN64 INT	64	-	Remote module input data area
Share IN64 REAL	64	-	Remote module input data area
Share OUT2 BOOL	-	2	Remote module output data area
Share OUT32 DINT	-	32	Remote module output data area
Share OUT32 INT	-	32	Remote module output data area
Share OUT32 REAL	-	32	Remote module output data area
Share OUT64 DINT	-	64	Remote module output data area
Share OUT64 INT	-	64	Remote module output data area
Share OUT64 REAL	-	64	Remote module output data area

The physical extension modules supported by H5U are the GL10 series modules.

- The following table lists extension modules supported by Easy devices.

Module Name	Input (Bytes)	Output (Bytes)	Description
0016ETN	-	2	

Module Name	Input (Bytes)	Output (Bytes)	Description
			A module with 16 DO terminals (transistor NPN output)
0016ETP	-	2	A module with 16 DO terminals (transistor PNP output)
1600END	2	-	A module with 16 DI terminals
3200END	4	-	An input module with 32 channels
0032ETN	-	4	An input module with 32 channels
4DA	-	8	A module with 4 DA terminals, supporting voltage and current output
4AD	8	-	A module with 4 AD terminals, supporting voltage and current input
4TC	8	-	4-channel temperature module
4PT	8	-	4-channel temperature module
0008ETN	-	1	A module with 8 DO terminals (transistor NPN output)
0008ETP	-	1	A module with 8 DO terminals (transistor PNP output)
0008ER	-	1	A module with 8 DO terminals (relay output)
0800END	1	-	A module with 8 DI terminals
0808ETN	1	1	A module with 8 DI terminals and 8 DO terminals
3232ETN	4	4	A module with 32 DI terminals and 32 DO terminals
Share IN2 BOOL	2	-	Remote module input data area
Share IN32 DINT	32	-	Remote module input data area
Share IN32 INT	32	-	Remote module input data area
Share IN32 REAL	32	-	Remote module input data area
Share IN64 DINT	64	-	Remote module input data area
Share IN64 INT	64	-	Remote module input data area
Share IN64 REAL	64	-	Remote module input data area
Share OUT2 BOOL	-	2	Remote module output data area

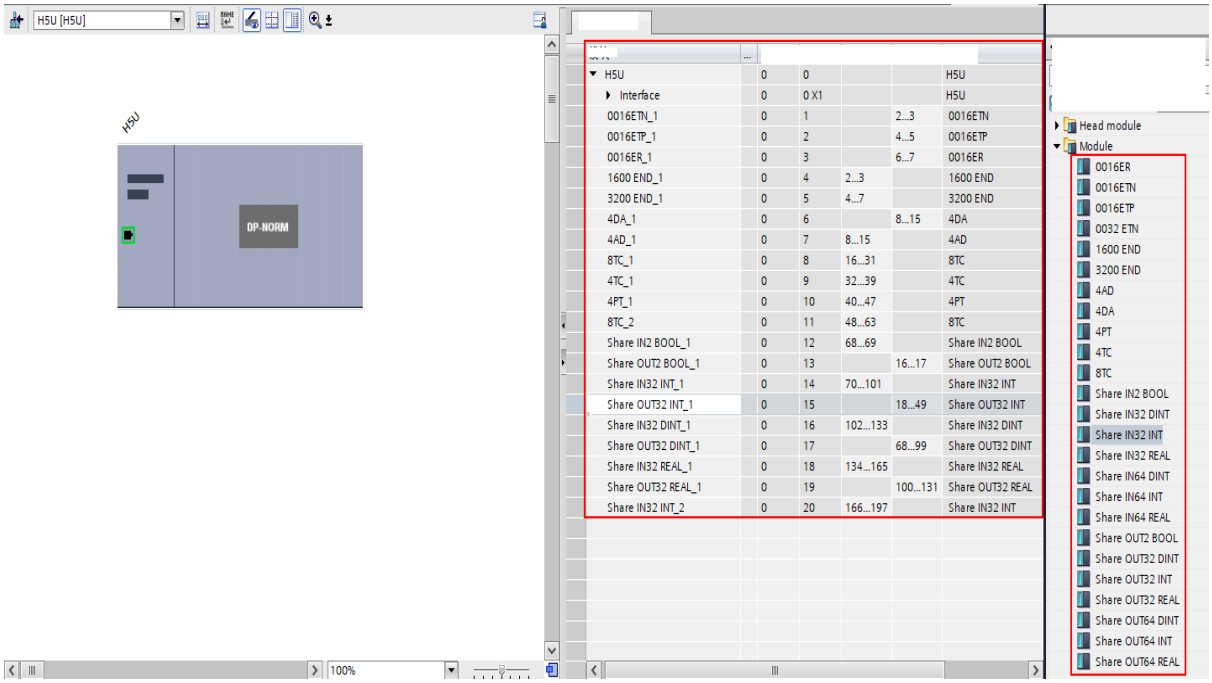
Module Name	Input (Bytes)	Output (Bytes)	Description
Share OUT32 DINT	-	32	Remote module output data area
Share OUT32 INT	-	32	Remote module output data area
Share OUT32 REAL	-	32	Remote module output data area
Share OUT64 DINT	-	64	Remote module output data area
Share OUT64 INT	-	64	Remote module output data area
Share OUT64 REAL	-	64	Remote module output data area

NOTE

The physical extension modules supported by Easy are the GL20 series modules.

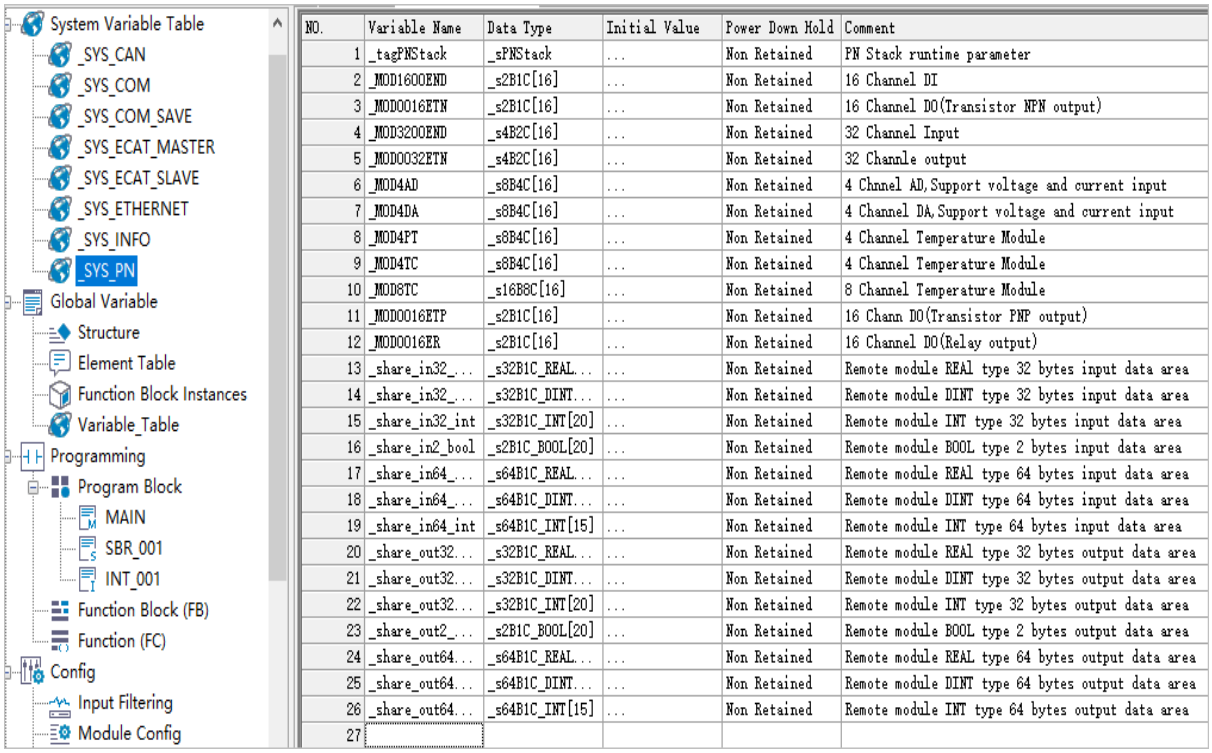
Data mapping relationship

- The following figure shows the TIA Portal page of an H5U device.



The directory tree "Module" on the right shows all currently supported extension modules, as well as shared modules Share IN XX and Share OUT XX. All modules can be added to the slots under the H5U or Easy device in the "Device Overview" area in any order. A maximum of 16 modules can be added to the 16 slots (except Share INXX/Share OUTXX, a maximum of 15 Share IN64XX/Share OUT64XX modules or a maximum of 20 other modules can be added). In the TIA Portal software, you can operate the I or Q address of each module, map the final data to the H5U or Easy device, and view the data of the module in AutoShop.

- The following figure shows the AutoShop page of an H5U device.



NO.	Variable Name	Data Type	Initial Value	Power Down Hold	Comment
1	_tagPNStack	_sPNStack	...	Non Retained	PN Stack runtime parameter
2	_MOD1600END	_s2B1C[16]	...	Non Retained	16 Channel DI
3	_MOD0016ETN	_s2B1C[16]	...	Non Retained	16 Channel DO(Transistor NPN output)
4	_MOD3200END	_s4B2C[16]	...	Non Retained	32 Channel Input
5	_MOD0032ETN	_s4B2C[16]	...	Non Retained	32 Channle output
6	_MOD4AD	_s8B4C[16]	...	Non Retained	4 Chnnel AD,Support voltage and current input
7	_MOD4DA	_s8B4C[16]	...	Non Retained	4 Channel DA,Support voltage and current input
8	_MOD4PT	_s8B4C[16]	...	Non Retained	4 Channel Temperature Module
9	_MOD4TC	_s8B4C[16]	...	Non Retained	4 Channel Temperature Module
10	_MOD8TC	_s16B8C[16]	...	Non Retained	8 Channel Temperature Module
11	_MOD0016ETP	_s2B1C[16]	...	Non Retained	16 Chann DO(Transistor PNP output)
12	_MOD0016ER	_s2B1C[16]	...	Non Retained	16 Channel DO(Relay output)
13	_share_in32_...	_s32B1C_REAL...	...	Non Retained	Remote module REAl type 32 bytes input data area
14	_share_in32_...	_s32B1C_DINT...	...	Non Retained	Remote module DINT type 32 bytes input data area
15	_share_in32_int	_s32B1C_INT[20]	...	Non Retained	Remote module INT type 32 bytes input data area
16	_share_in2_bool	_s2B1C_BOOL[20]	...	Non Retained	Remote module BOOL type 2 bytes input data area
17	_share_in64_...	_s64B1C_REAL...	...	Non Retained	Remote module REAl type 64 bytes input data area
18	_share_in64_...	_s64B1C_DINT...	...	Non Retained	Remote module DINT type 64 bytes input data area
19	_share_in64_int	_s64B1C_INT[15]	...	Non Retained	Remote module INT type 64 bytes input data area
20	_share_out32...	_s32B1C_REAL...	...	Non Retained	Remote module REAl type 32 bytes output data area
21	_share_out32...	_s32B1C_DINT...	...	Non Retained	Remote module DINT type 32 bytes output data area
22	_share_out32...	_s32B1C_INT[20]	...	Non Retained	Remote module INT type 32 bytes output data area
23	_share_out2_...	_s2B1C_BOOL[20]	...	Non Retained	Remote module BOOL type 2 bytes output data area
24	_share_out64...	_s64B1C_REAL...	...	Non Retained	Remote module REAL type 64 bytes output data area
25	_share_out64...	_s64B1C_DINT...	...	Non Retained	Remote module DINT type 64 bytes output data area
26	_share_out64...	_s64B1C_INT[15]	...	Non Retained	Remote module INT type 64 bytes output data area
27					

AutoShop uses the system variable "_SYS_PN" to obtain mapped data in the TIA Portal software.

In the TIA Portal software, data areas of all H5U and Easy modules are mapped to corresponding variables to the variable table "_SYS_PN" in AutoShop. The following table lists the mapping relationship.

- Mapping relationship of physical extension modules

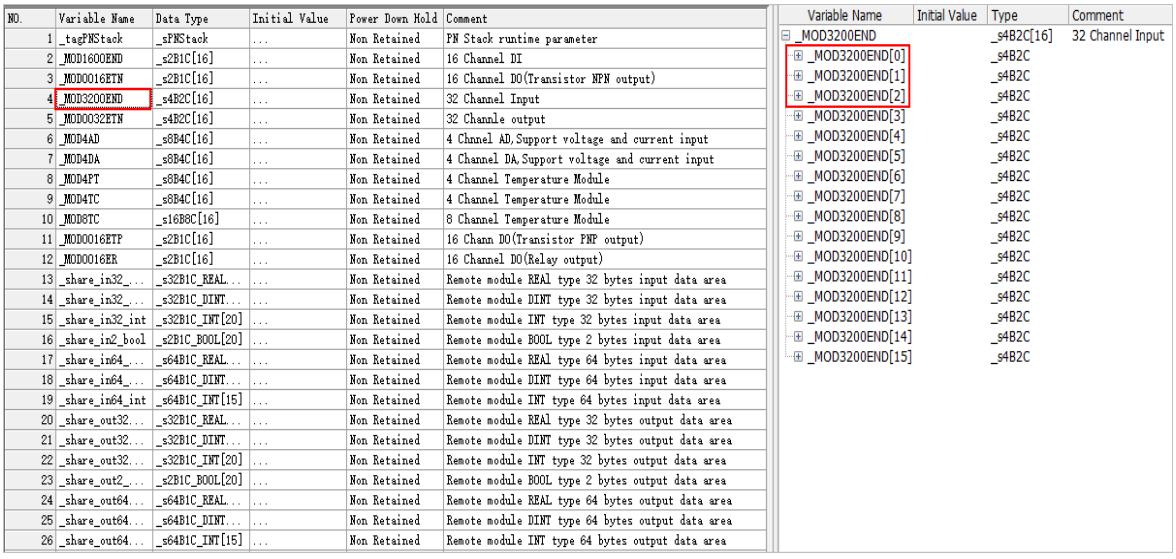
  The following table lists mapping relationship of physical extension modules of H5U.

TIA Portal Module Name	AutoShop Variable Name
0016ETN	MOD0016ETN
0016ETP	MOD0016ETP
0016ER	MOD0016ER
1600END	MOD1600END
3200END	MOD3200END
0032ETN	MOD0032ETN
4DA	MOD4DA
4AD	MOD4AD
8TC	MOD8TC
4TC	MOD4TC
4PT	MOD4PT

  The following table lists mapping relationship of physical extension modules of Easy.

TIA Portal Module Name	AutoShop Variable Name
0016ETN	MOD0016ETN
0016ETP	MOD0016ETP
1600END	MOD1600END
3200END	MOD3200END
0032ETN	MOD0032ETN
4DA	MOD4DA
4AD	MOD4AD
4TC	MOD4TC
4PT	MOD4PT
0008ETN	MOD0008ETN
0008ETP	MOD0008ETP
0008ER	MOD0008ER
0800END	MOD0800END
0808ETN	MOD0808ETN
3232ETN	MOD3232ETN

When there are multiple identical modules, these modules will be mapped to the data areas of the corresponding indexes in the order they are added in the TIA Portal software. For example, three 3200END modules are added in the Portal software, and they are mapped to "MOD3200[0]", "MOD3200[1]", and "MOD3200[2]", respectively.



NO.	Variable Name	Data Type	Initial Value	Power Down Hold	Comment
1	_tagPNStack	_sPNStack	...	Non Retained	PN Stack runtime parameter
2	_MOD1600END	_s2B1C[16]	...	Non Retained	16 Channel DI
3	_MOD0016ETN	_s2B1C[16]	...	Non Retained	16 Channel DO(Transistor NPN output)
4	_MOD3200END	_s4B2C[16]	...	Non Retained	32 Channel Input
5	_MOD0032ETN	_s4B2C[16]	...	Non Retained	32 Channle output
6	_MOD4AD	_s8B4C[16]	...	Non Retained	4 Chnnel AD,Support voltage and current input
7	_MOD4DA	_s8B4C[16]	...	Non Retained	4 Channel DA,Support voltage and current input
8	_MOD4PT	_s8B4C[16]	...	Non Retained	4 Channel Temperature Module
9	_MOD4TC	_s8B4C[16]	...	Non Retained	4 Channel Temperature Module
10	_MOD8TC	_s16B8C[16]	...	Non Retained	8 Channel Temperature Module
11	_MOD0016ETP	_s2B1C[16]	...	Non Retained	16 Chann DO(Transistor PNP output)
12	_MOD0016ER	_s2B1C[16]	...	Non Retained	16 Channel DO(Relay output)
13	_share_in32_...	_s32B1C_REAL...	...	Non Retained	Remote module REAl type 32 bytes input data area
14	_share_in32_...	_s32B1C_DINT...	...	Non Retained	Remote module DINT type 32 bytes input data area
15	_share_in32_int	_s32B1C_INT[20]	...	Non Retained	Remote module INT type 32 bytes input data area
16	_share_in2_bool	_s2B1C_BOOL[20]	...	Non Retained	Remote module BOOL type 2 bytes input data area
17	_share_in64_...	_s64B1C_REAL...	...	Non Retained	Remote module REAl type 64 bytes input data area
18	_share_in64_...	_s64B1C_DINT...	...	Non Retained	Remote module DINT type 64 bytes input data area
19	_share_in64_int	_s64B1C_INT[15]	...	Non Retained	Remote module INT type 64 bytes input data area
20	_share_out32...	_s32B1C_REAL...	...	Non Retained	Remote module REAl type 32 bytes output data area
21	_share_out32...	_s32B1C_DINT...	...	Non Retained	Remote module DINT type 32 bytes output data area
22	_share_out32...	_s32B1C_INT[20]	...	Non Retained	Remote module INT type 32 bytes output data area
23	_share_out2_...	_s2B1C_BOOL[20]	...	Non Retained	Remote module BOOL type 2 bytes output data area
24	_share_out64...	_s64B1C_REAL...	...	Non Retained	Remote module REAL type 64 bytes output data area
25	_share_out64...	_s64B1C_DINT...	...	Non Retained	Remote module DINT type 64 bytes output data area
26	_share_out64...	_s64B1C_INT[15]	...	Non Retained	Remote module INT type 64 bytes output data area

Variable Name	Initial Value	Type	Comment
_MOD3200END		_s4B2C[16]	32 Channel Input
_MOD3200END[0]		_s4B2C	
_MOD3200END[1]		_s4B2C	
_MOD3200END[2]		_s4B2C	
_MOD3200END[3]		_s4B2C	
_MOD3200END[4]		_s4B2C	
_MOD3200END[5]		_s4B2C	
_MOD3200END[6]		_s4B2C	
_MOD3200END[7]		_s4B2C	
_MOD3200END[8]		_s4B2C	
_MOD3200END[9]		_s4B2C	
_MOD3200END[10]		_s4B2C	
_MOD3200END[11]		_s4B2C	
_MOD3200END[12]		_s4B2C	
_MOD3200END[13]		_s4B2C	
_MOD3200END[14]		_s4B2C	
_MOD3200END[15]		_s4B2C	

- "Share IN XX" and "Share OUT XX" modules are mapped to variables "_share_in xx" and "_share_out xx", respectively. These modules can be used as data mapping variables for remote modules (Share IN XX mapped to _share_in xx, and Share OUT XX mapped to _share_out xx).

13.2 Configuration Process

13.2.1 TIA Portal Configuration

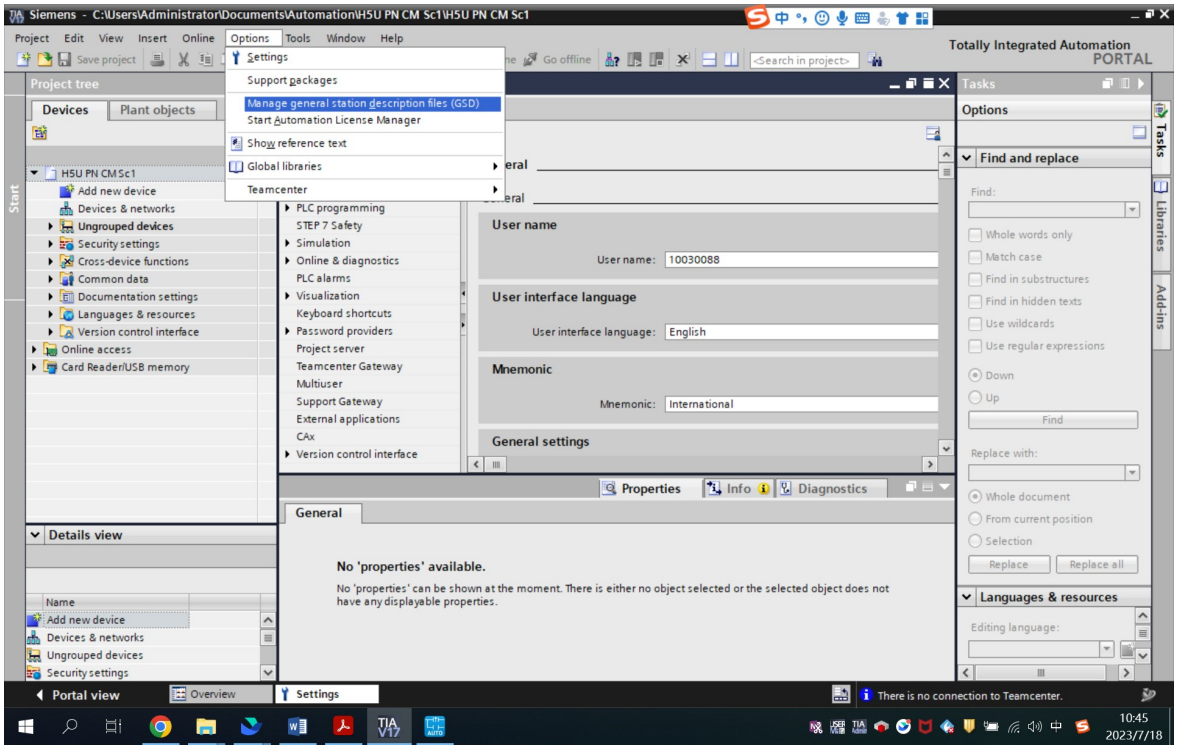
The configuration procedure of the H5U is similar to that of the Easy series. The following takes H5U as an example to describe how to make configuration.

Preconditions

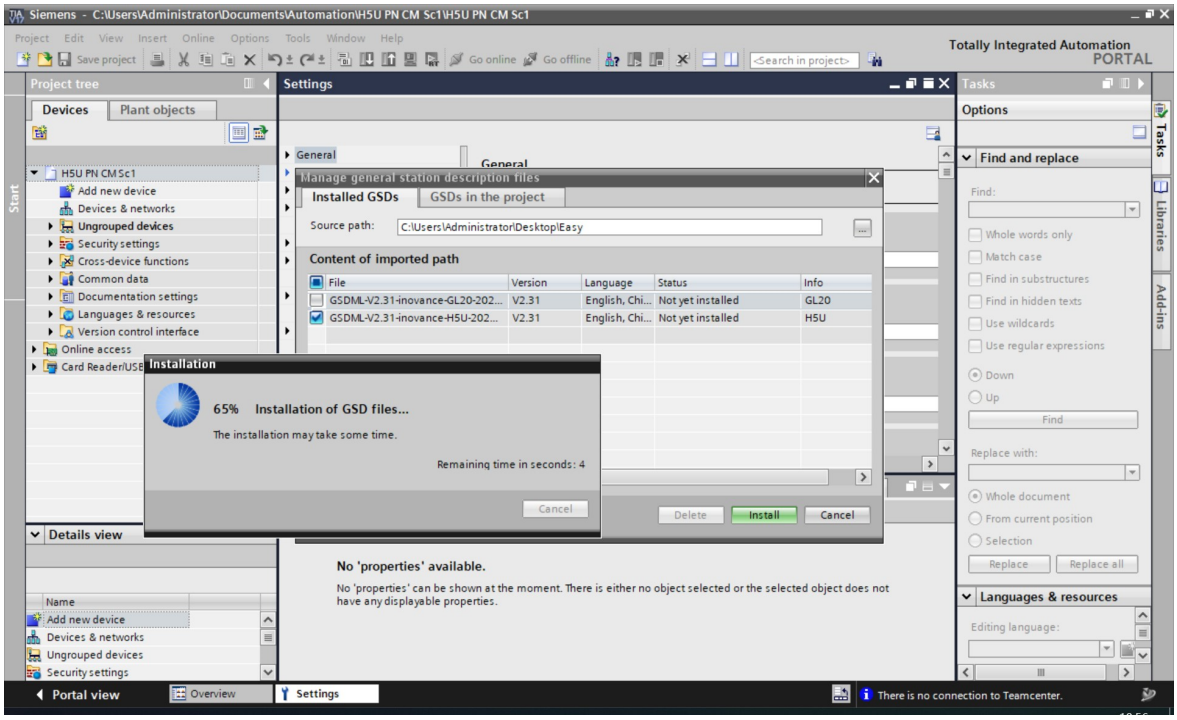
You have downloaded the H5U GSD file of Easy GSD file from Service and Support > Downloads in Inovance official website (www.inovance.com).

Procedure

1. Import and install the GSD file.
   a. In the menu bar, choose "Option" > "Manage general station description files (GSD)", as shown in the following figure.

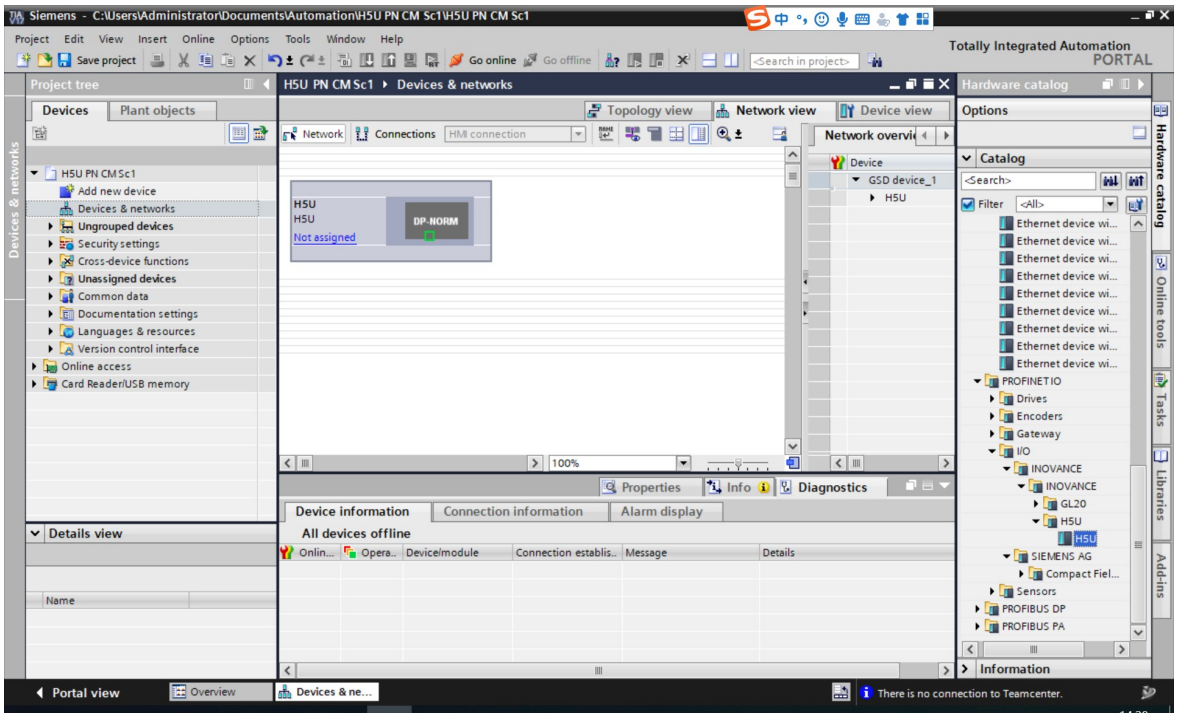


   b. In the dialog box displayed, click ..., locate the GSD file (the software automatically indexes the file), select the GSD file, and then click "Install".

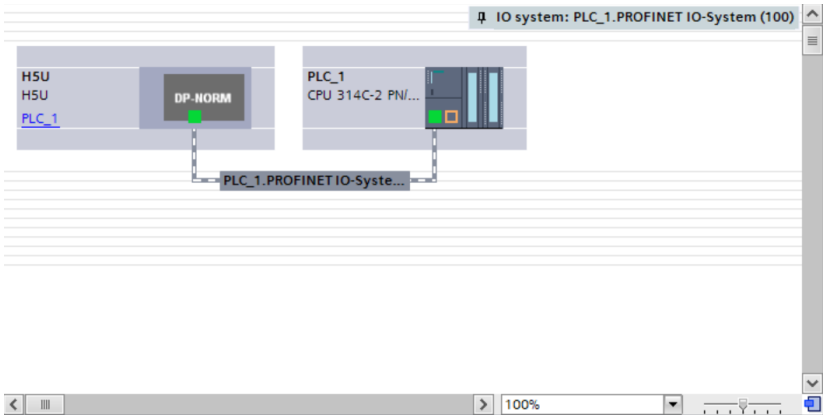


2. Configure H5U.

a. Double-click "H5U" in the hardware directory or drag "H5U" to the "Network" view section, as shown in the following figure.



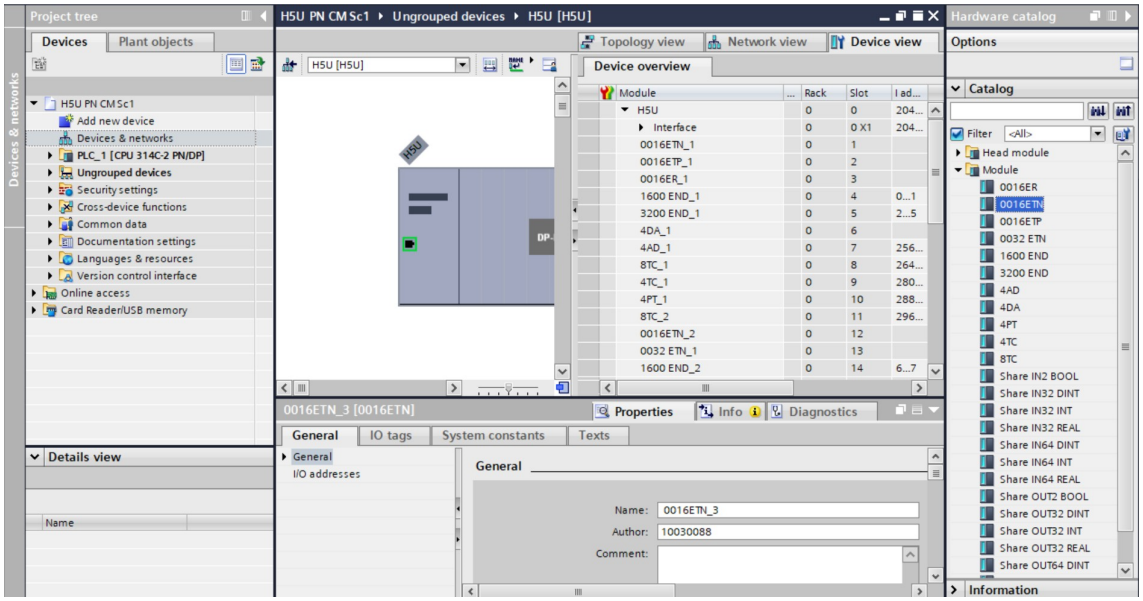
b. Hold down the left mouse button and drag the PLC green box to the H5U green box to connect the PLC and H5U, as shown in the following figure.



NOTE

When H5U servers as a PROFINET slave, the PROFINET bus in the TIA Portal software automatically allocates an IP address for IP, which needs to be modified to one different from the actual IP address of H5U; otherwise, the actual IP address of H5U will become invalid.

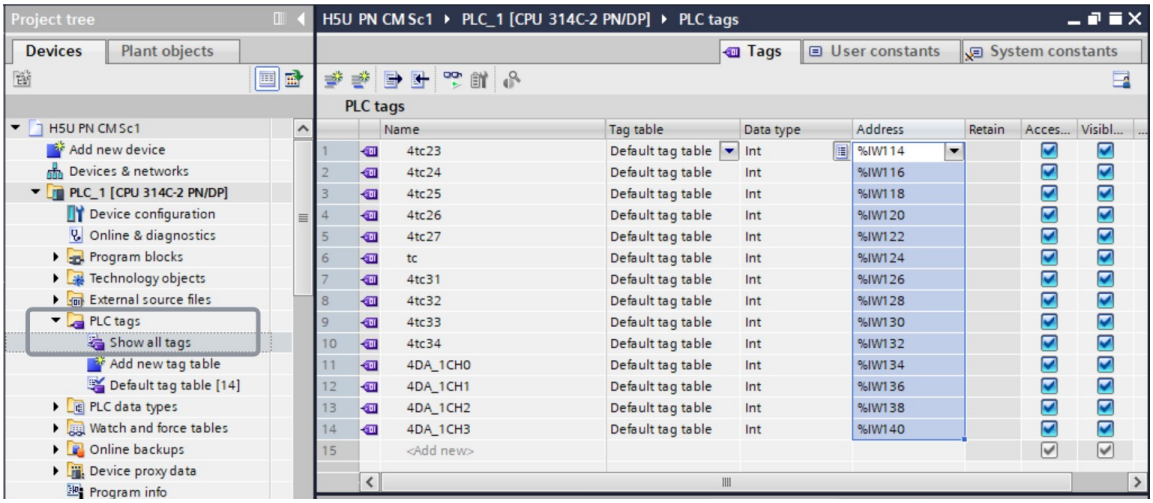
c. Double-click a module in the hardware directory tree on the right to add the module and assign the address, as shown in the following figure.



NOTE

It is recommended to keep the order of added modules and slots consistent with the AutoShop interface.

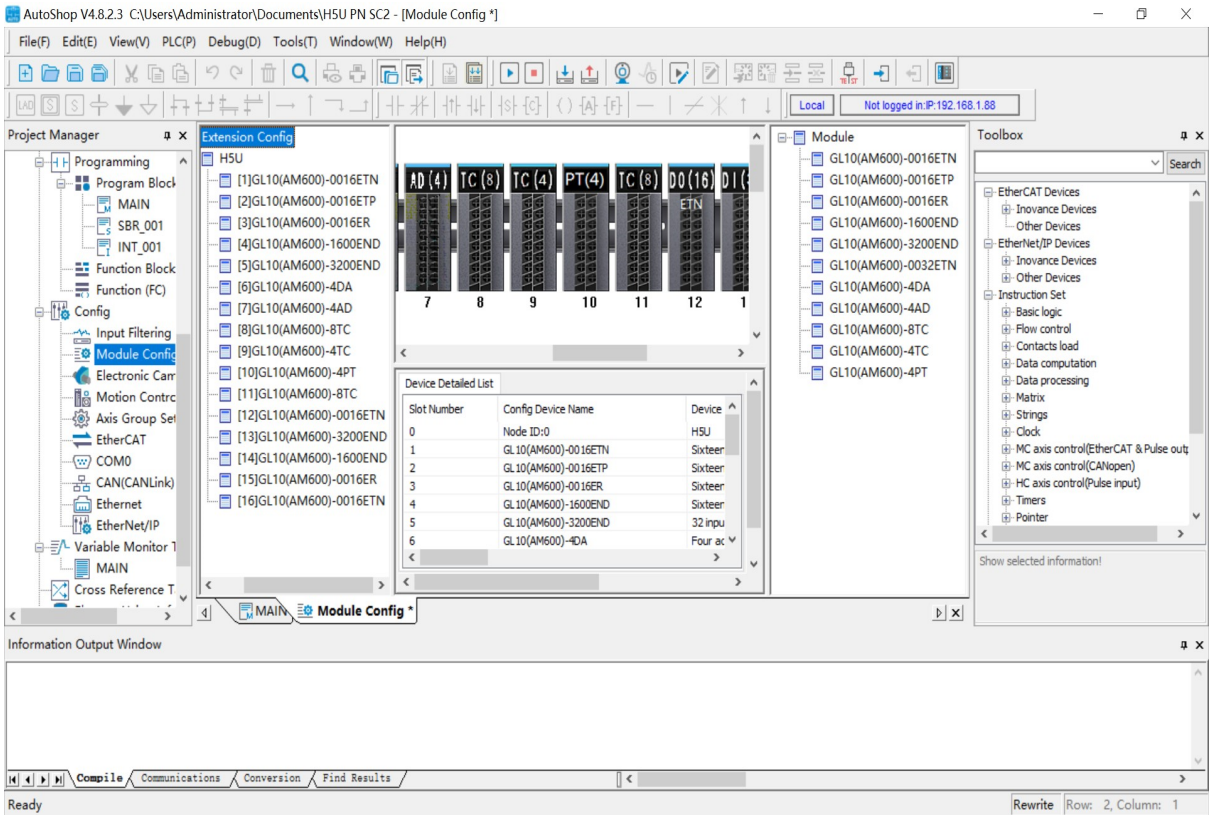
d. Click "Show all tags". On the "PLC tags" page displayed, define the specific address variable of the module, as shown in the following figure.



13.2.2 AutoShop Configuration

The configuration of H5U series PLC is similar to that of Easy series. The following section takes H5U series as an example.

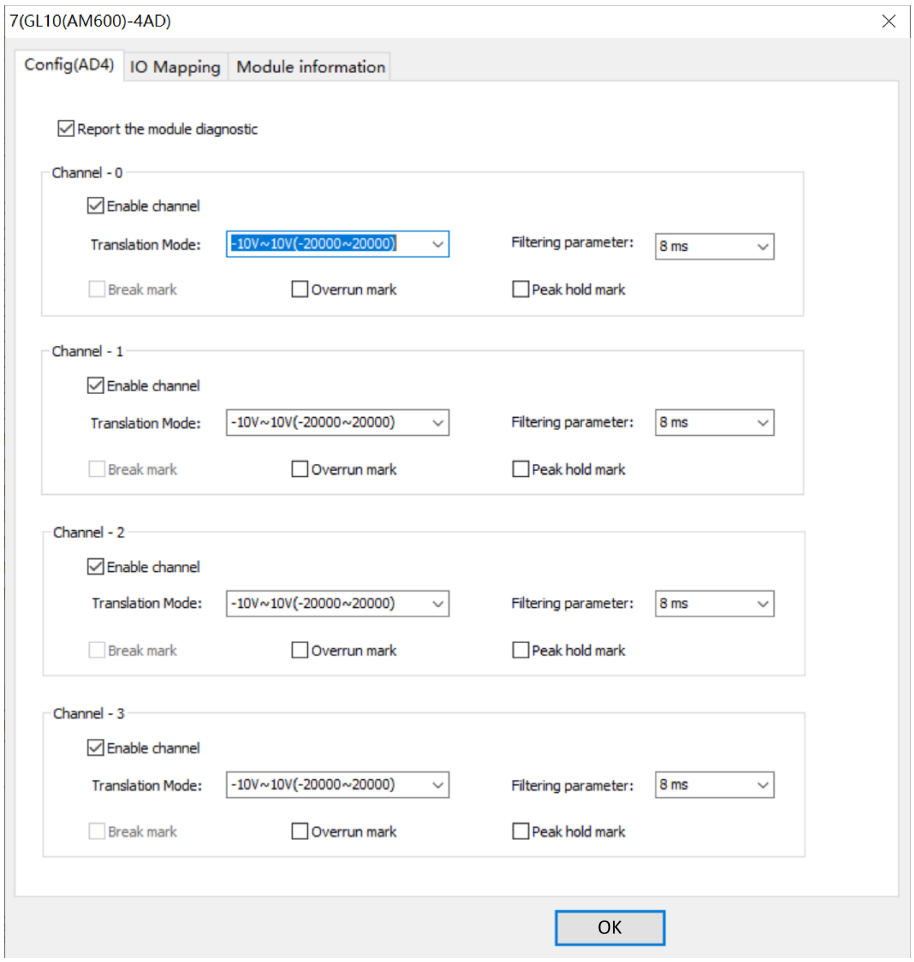
1. Under “Config” in the “Project Manager” section, double-click “Module Config”. In the “Module” section, double-click the module you want to add, as shown in the following figure.



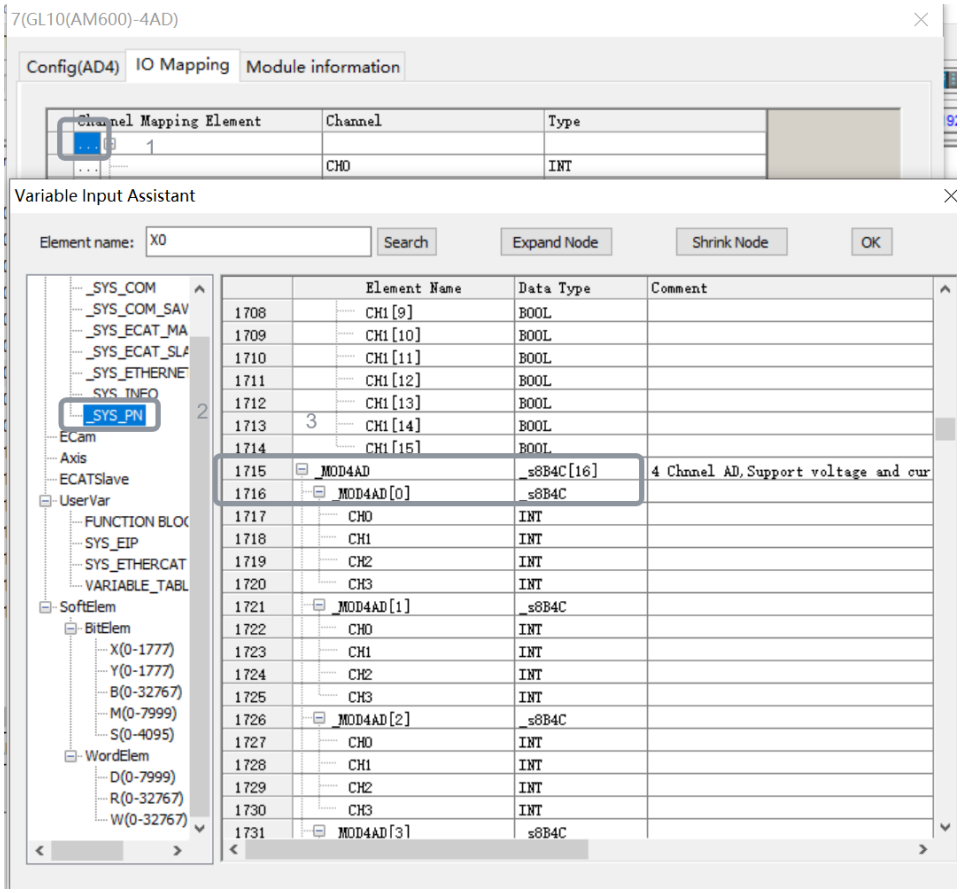
NOTE

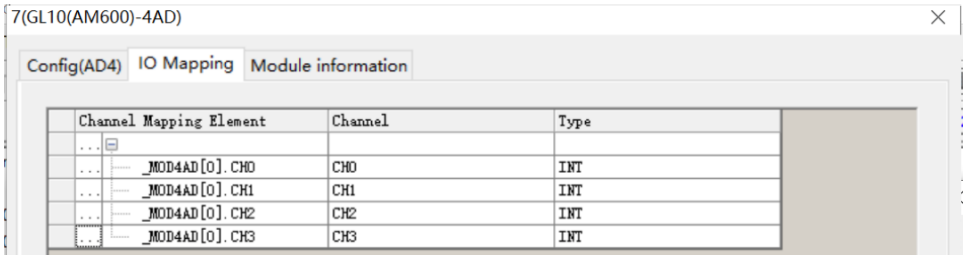
It is recommended to keep the order of added modules and slot numbers consistent with the TIA Portal interface.

2. In the “Device Detailed List” section, double-click the module (taking GL10-4AD as an example) for which you want to configure parameters, configure the module parameters, and then click “OK”, as shown in the following figure.



3. Map the module I/O to the corresponding variable in the system variable “_SYS_PN”.
   a. In the “Device Detailed List” section, double-click the module (taking GL10-4AD as an example) for which you want to perform I/O mapping, as shown in the preceding figure.
   b. Click “IO Mapping”, and map the module to the variable “MOD4AD[0]”, as shown in the following figure.





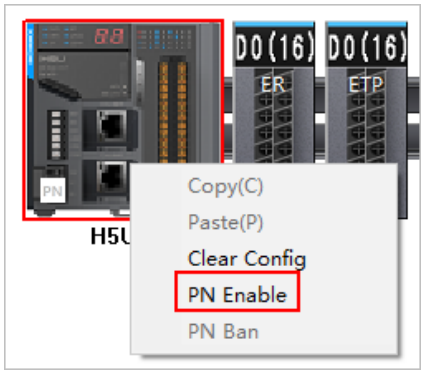
c. Click “OK”, and then “OK”.

13.3 Enable and Disable

PROFINET is disabled for H5U by default. To enable this function, click the "Enable" function. The configuration procedure of the H5U is similar to that of the Easy series. The following takes H5U as an example to describe how to enable or disable the protocol.

Enabled

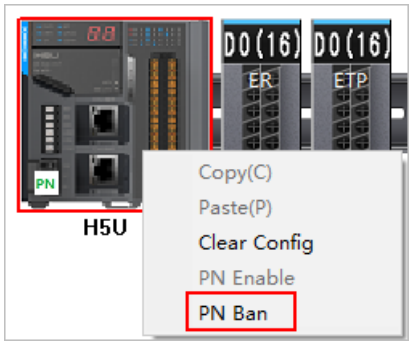
1. In the "Project Manager" tree, click "Config". On the "Module Config" page, right-click "H5U", and select "PN Enable".



2. Download the program to H5U.
3. Restart H5U. Then, the function is disabled.

Disabled

1. In the "Project Manager" tree, click "Config". On the "Module Config" page, right-click "H5U", and select "PN Ban".



2. Download the program to H5U.
3. Restart H5U. Then, the function is disabled.

14 OPC UA Communication

14.1 Overview

Introduction

Open Platform Communications Unified Architecture (OPC UA) is an industrial communication protocol to realize safe and reliable data exchange between devices, systems, and applications of different manufacturers and platforms. The OPC UA protocol is formulated by OPC Foundation. Developed from classic OPC protocols (such as OPC DA, OPC HDA, and OPC A&E), it solves the problem that traditional OPC relies on the Microsoft COM/DCOM technology and is limited to Windows environment.

Key features

- Cross-platform: OPC UA is a cross-platform protocol that supports a variety of operating systems including but not limited to Windows, Linux, Mac OS, and mobile devices.
- Unified architecture: It integrates multiple functions of data access, alarm, and events into a unified architecture, without the need of a separate interface.
- Service orientation: Based on service-oriented architecture (SOA), it supports flexible service-oriented communication.
- Security: It provides security mechanisms such as encryption, authentication, and authorization to ensure the security of data communication.
- Vendor independence: It is independent of specific vendor technology, which helps to achieve interoperability between devices from different vendors.
- Information model: It uses an object-oriented information modeling method, including variables, events and methods, which are interrelated by reference, and supports the transmission of complex data types.
- Rich communication modes: It supports client/server (C/S) mode and publish/subscribe (Pub/Sub) mode to adapt to different real-time and bandwidth requirements.
- Historical data and time series: it not only supports real-time data access, but also can store and retrieve historical data and events.
- Extensibility: It allows custom extension services and data types to meet the ever-developing industrial automation and Internet of Things application requirements.
- Network independence: It can be deployed in local area network, wide area network or Internet, and is suitable for Industrial 4.0 and industrial Internet scenarios.

The OPC UA protocol is an important infrastructure of modern industrial automation and industrial Internet, which promotes seamless data integration and exchange in different levels and fields (from sensors to the cloud).

Description

The OPC UA protocol is divided into two parts: OPC UA Server (server) and client.

The OPC UA server provides cross-platform and cross-system standardized data access services, ensuring safe and reliable data exchange from sensors to the cloud.

The OPC UA server provides the following core features.

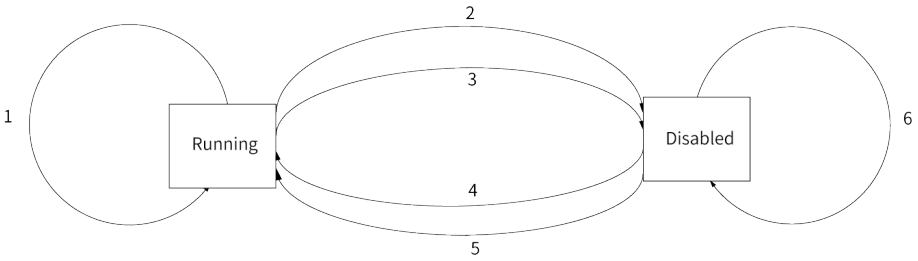
- It provides a unified data model that supports complex information structure description and can transmit real-time data, historical data, and rich metadata.
- It implements a secure communication mechanism and supports multiple authentication and encryption modes, including anonymous access, username and password authentication, and certificate authentication.
- It is widely applied and suitable for industrial automation, Internet of Things (IoT), intelligent manufacturing, and other fields. It can be used for the interoperability between different levels of equipment, such as the integration of PLC, SCADA system, MES system, and ERP system.
- It allows any device as a data provider, so that products from different manufacturers can share data seamlessly, and the transparency and efficiency of the whole production process are improved.
- The OPC UA server acts as the core hub of industrial data by being built in or attached to devices, controllers or running on a separate computer, which promotes the process of enterprise informatization and intelligence.

14.2 OPC UA Configuration

Background

NOTE

- When the OPC UA server is running, modification to the PLC IP address through background or system variables can take effect on the OPC UA server after you stop, run, or re-power on the PLC.
- When the OPC UA server runs on the PLC for the first time, or when the OPC UA server runs on the PLC for the first time after the IP address is modified, it is necessary to regenerate the OPC UA server certificate file, which takes 1s to 2s.OPC UA server start/stop logic



1. After the PLC is switched to the Stop state, the OPC UA server keeps running. When you stop, run, or re-power on the PLC with the PLC project not changed, the running state of the OPC UA server will not change.
2. When the OPC UA server is running, modify the PLC project, disable the OPC UA server, and then download the project to the PLC. After stopping/running or re-powering on the PLC, the OPC UA server is shut down.
3. When the OPC UA server is running, run the OPCUA_Server_Shutdown command to shut down the OPC UA server.
4. When the OPC UA server is shut down, but the OPC UA server configuration is enabled in the project of the PLC, the OPC UA server will be turned on after you stop, run, or re-power on the PLC.
5. When the OPC UA server is shut down, modify the PLC project, disable the OPC UA server, and then download the project to the PLC. After stopping/running or re-powering on the PLC, the OPC UA server is turned on.
6. When the OPC UA server is shut down and the OPC UA server configuration is disabled in the project of the PLC, the server status remains unchanged after you stop, run, or re-power on the PLC.

- Specifications of OPC UA server variables
  - Data type of variables registered to the OPC UA server
    - Basic variable type

      The following table lists the basic data type of PLC corresponds to the data type of OPC UA.

Controller Data Type	OPC UA Data Type	OPC UA Data Type Description
BOOL	BOOLEAN	Value indicating two states (8-bit)
BYTE	BYTE	8-bit unsigned integer
INT	INT16	16-bit signed integer
DINT	INT32	32-bit signed integer

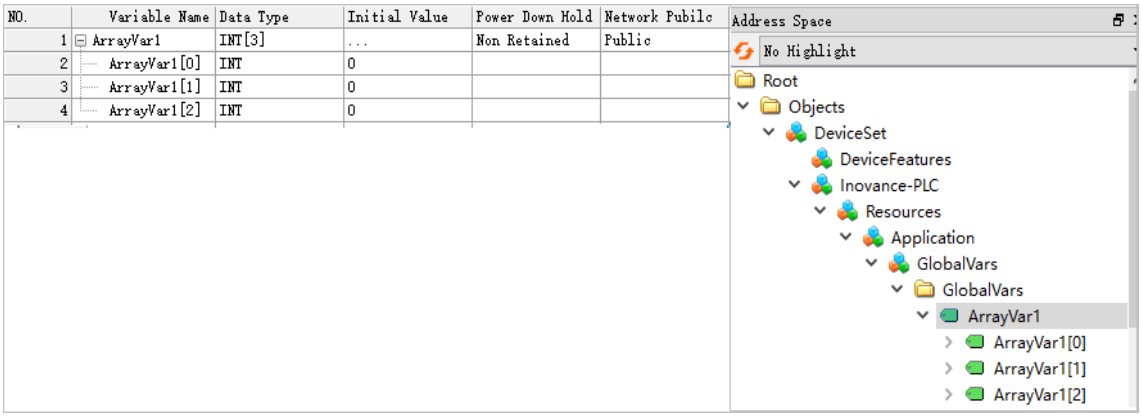
Controller Data Type	OPC UA Data Type	OPC UA Data Type Description
REAL	FLOAT	IEEE-754 single precision floating point value
STRING	STRING	UTF-8 character string ended with NULL

Conform to the specifications of PLCOpen OPC UA information model 1.00.

- Array

  For the array variable of the PLC, the entire array variable is registered as a node (basic variable type), and each array variable member is registered as a node independently.

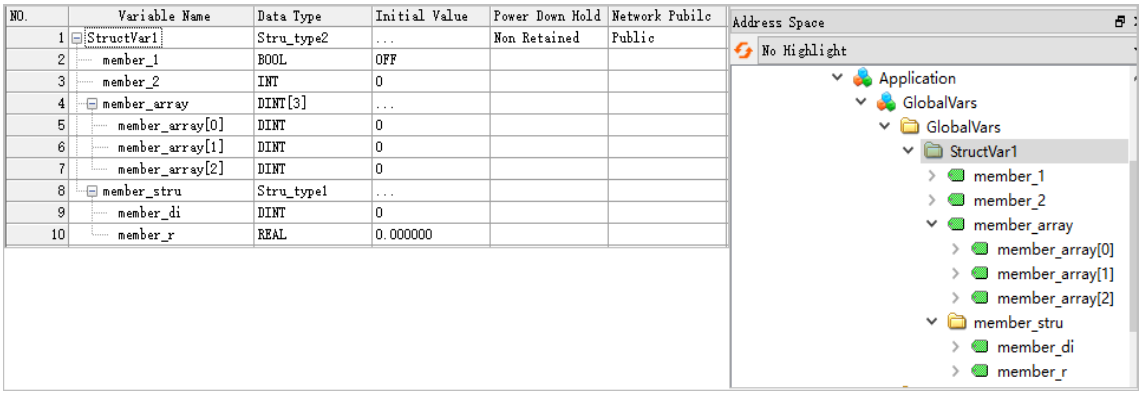
  As shown in the following figure, ArrayVar1 is an array variable of the INT type.



  - The OPC UA client displays each element in the array.
  - You can read and write the entire array variable (basic variable type) and each member of the array.
  - For the structure array variable, the entire array variable is registered as a path node, and the array members are registered according to the structure variable rules.

- Structure

  For the structure variables of the PLC, the entire structure is registered as a path node, and each variable member of the structure is registered separately as a node.



  - The OPC UA client displays each element in the structure.
  - Each member in the structure variable can be read and written.

◦ Restrictions of variables registered to the OPC UA server

Variables registered to the OPC UA server are subject to the following restrictions. Not all variables set with the network public (public or input/output) property can be registered to the OPC UA server. If there are variables that are not registered to the OPC UA server, it is necessary to check the variable configuration in the project to ensure that they are within the following restrictions.

Item	Restriction	Description
Public variable restriction	Maximum number of public variables	2000
	Maximum number of public variable elements	10000
Array specification	Maximum array variable size	2048 bytes
	Maximum number of array variable elements	10000
	Array dimension	Only one-dimensional arrays are supported.
Structure specification	Maximum structure variable size	2048 bytes
	Maximum number of structure variable elements	10000
	Maximum number of nested layers of structure	Nested layers cannot exceed 3.

The number of public variable elements is calculated as follows.

Number of public variable elements = (Number of variables of basic type) + (Number of members of array variables) + (Number of members of structure variables)

When calculating the number of members of a structure or an array, the members of basic variable types shall be counted.

- Network public property of variables registered to the OPC UA server
  Whether a global variable can be registered to the OPC UA server is determined by the network public property of the global variable, as shown in the following table.

Network Public Property	Registered to OPC UA Server	Read/Write Permission of OPC UA Client
Private	No	It cannot be read or written.
Public	Yes	It can be read and written.
Input/Output	Yes	It can be read and written.

Adding, deleting and modifying variables of public and input/output properties under online modification is not supported.

Preconditions

- Easy320 and Easy52X series PLCs support the OPC UA server.
- Easy (firmware version 6.3.0.0 and later) supports the OPC UA server in AutoShop software version 4.10.1.0 and later.

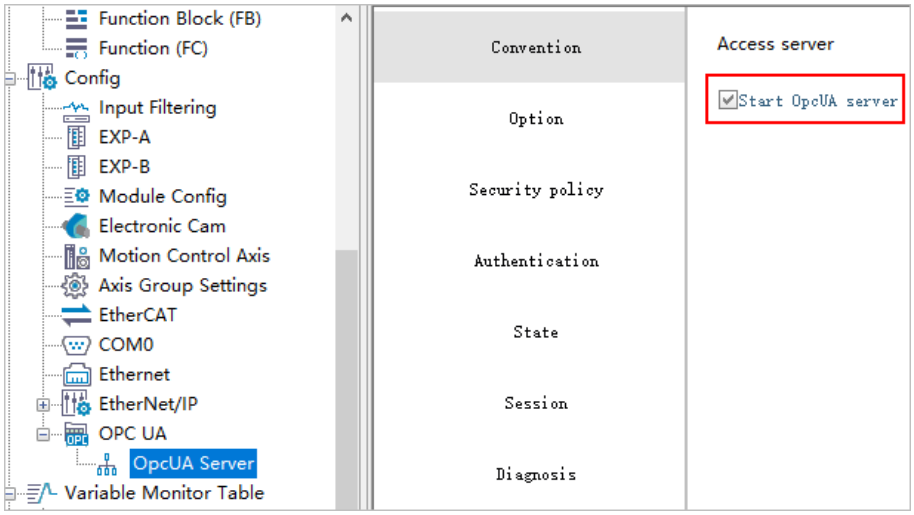
Precaution

After the OPC UA server is started, the data variables of the "input/output" and "public" properties to be monitored are the same as those in EtherNet/IP and need to be created in the variable table. In the project where EtherNet/IP and OPC UA networking exist at the same time, both communications can read and write the same variable.

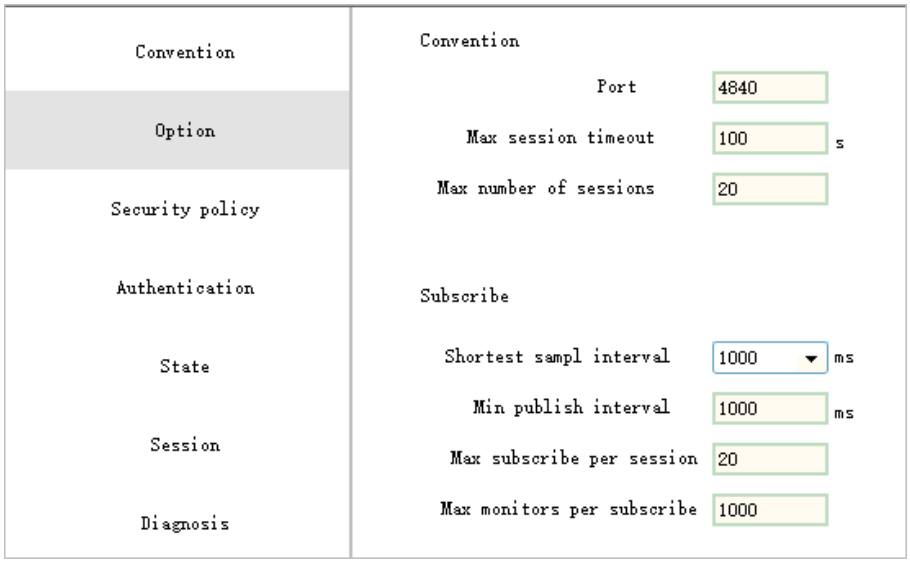
Procedure

1. Configure the OPC UA server.
   a. In the "Project Manager" tree, choose "Config" > "OPC UA" > "OpcUA Server". In the section on the right, click "Convention".

   b. Select "Start OPC UA Server" to enable the OPC UA server.



   c. Click "Option" and configure parameters of the OPC UA server.



Parameter Name	Description	Value
Port	Set the port number of the OPC UA server.	Default: 4840 Value range: 1024 to 49151 The following special ports cannot be set: • Modbus-TCP: 502

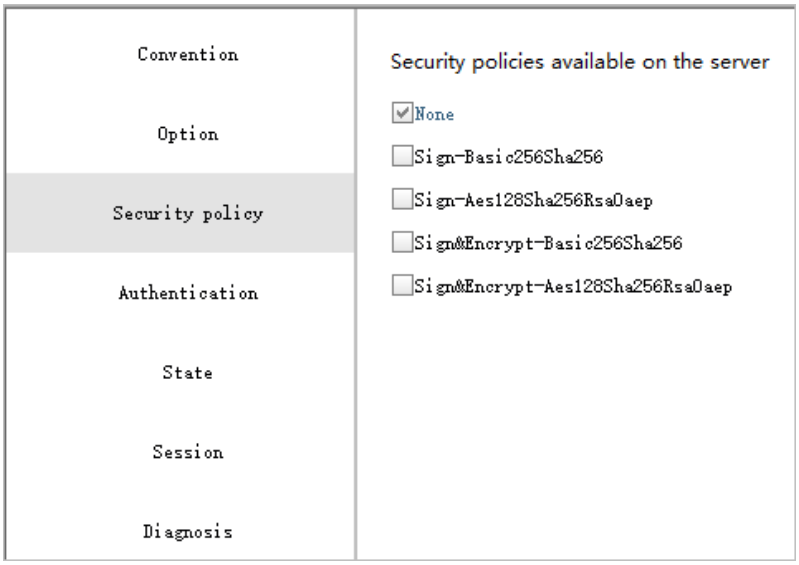
Parameter Name	Description	Value
		• FINS: 9600 • EtherNet/IP: 44818 and 2222 • EtherCAT: 34980 • Tag communication: 12939 and 12940
Maximum session timeout time	The maximum duration before the OPC UA server closes the session without data exchange.	Default: 50s Value range: 5s to 600000s
Maximum number of sessions	The maximum number of sessions that the OPC UA server starts and operates at the same time, which depends on the performance of the PLC.	The maximum number of subscriptions is 100. • It is fixed to 20 for Easy52x series PLCs. • It is fixed to 10 for Easy320 series PLCs.
Minimum sampling interval	The time interval when the OPC UA server records the PLC variable value and compares it with the previous value to check whether there is any change.	Default: 1000 ms Value range: 100 ms, 250 ms, 500 ms, 1000 ms, 5000 ms, and 10000 ms
Minimum release interval	Time interval when the server sends a message to the client through the new value when the variable value changes.	Default: 1000 ms Value range: 500 ms to 20000000 ms
Maximum number of subscriptions per session	The maximum number of subscriptions allowed per session.	It is fixed to 20.
Maximum number of monitoring items per subscription	The maximum number of monitoring elements that the server assembles into the notification package in each subscription. Monitoring will occupy resources, and the maximum number of monitoring elements depends on the performance of the PLC.	It is fixed to 1000.

The total maximum number of monitoring items for all subscriptions of the server is 1000. The number of monitoring items is affected by the sampling interval. Under the same number of monitoring items, the smaller the sampling interval is, the greater the system load is. To ensure the system function and execution efficiency, it is recommended to control the number of monitoring items within the following number according to the sampling interval.

Sampling Interval	Maximum Number of Monitoring Items
100 ms	100

Sampling Interval	Maximum Number of Monitoring Items
250 ms	500
≥500 ms	1000

d. Click "Security policy" and configure an encryption policy for the OPC UA server.

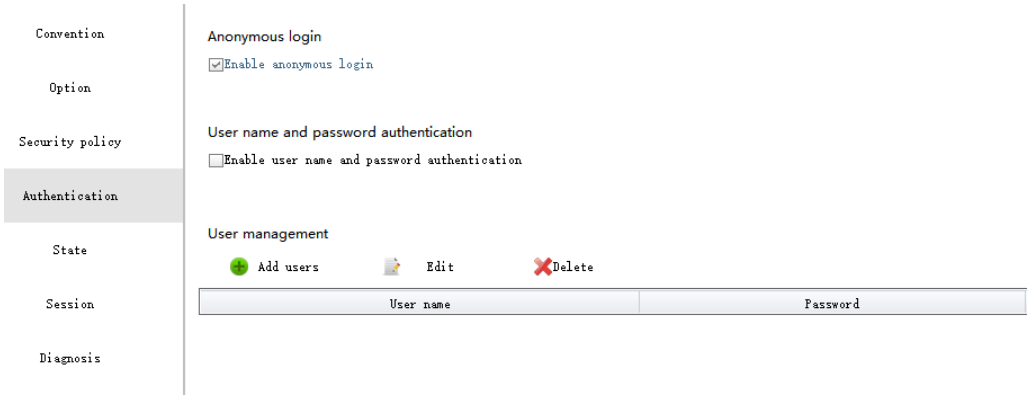


The security policy is the message security mechanism allowed when the OPC UA server communicates with the client, which is realized by setting the signature of the message, the encryption of the message, and the algorithm of signature and encryption. You can choose "None", single or multiple security policies. The specific security policy information is shown in the following table.

Parameter Name	Definition
None	No security settings
Sing-Basic256Sha256	Use signatures to ensure data integrity (to prevent tampering). Use Basic256Sha256 as the signature algorithm.
Sing-Aes128Sha256RsaOaep	Use signatures to ensure data integrity (to prevent tampering). Use Aes128Sha256RsaOaep as the signature algorithm.
Sing&Encrypt-Basic256Sha256	Use signature and encryption to ensure data integrity (to prevent tampering) and reliability (to prevent eavesdropping) Use Basic256Sha256 as the signature and encryption algorithm.
Sing&Encrypt-Aes128Sha256RsaOaep	Use signature and encryption to ensure data integrity (to prevent tampering) and reliability (to prevent eavesdropping) Use Aes128Sha256RsaOaep as the signature and encryption algorithm.

It is recommended to set a security policy at the initial stage of commissioning. After commissioning, select a security policy compatible with the safety concept of equipment or factory, and disable other security policies.

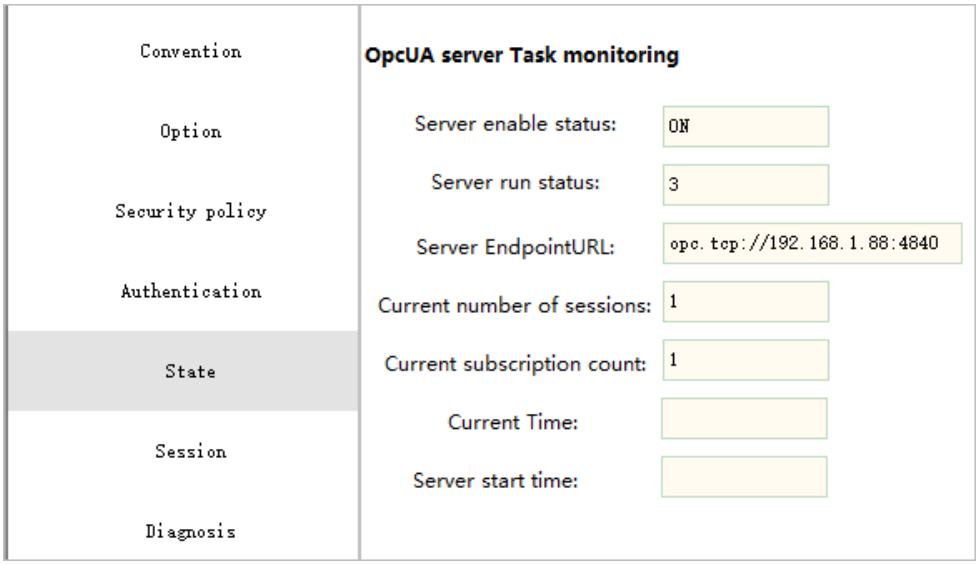
e. Click "Authentication" and set an authentication mode for users on the OPC UA client to access the server.



Parameter Name	Description	Value
Enable anonymous login	Users do not need to prove their identity (anonymous access), and the OPC UA server will not check the authorization of client users.	Select "Enable anonymous login" is selected by default.
Enable user name and password authentication	Users must prove their identity (non-anonymous access), and the OPC UA server will check whether the client user has the permission to access the server, and authenticate the user with the user name and correct password.	Select Edit the user information in the "User management" list below.
Add users	Add a user.	Up to 20 users can be added.
Edit	Edit the user name, such as modifying the user name or password.	User name and password can contain up to 20 characters, which can be numbers, uppercase and lowercase letters or English symbols.
Delete	Delete the selected user.	-

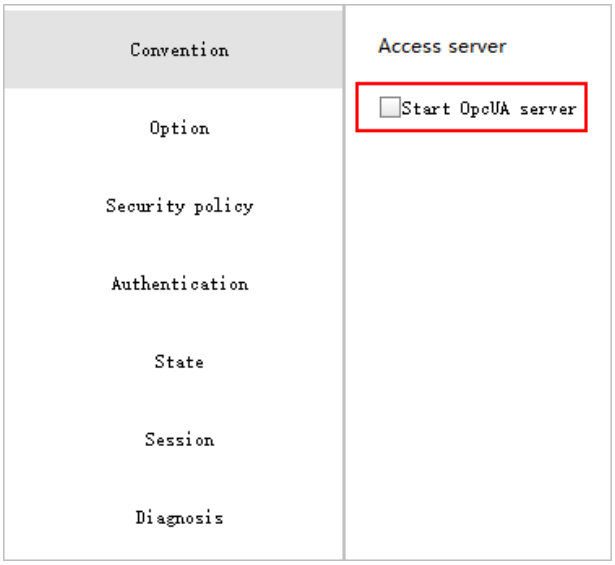
Select "Enable anonymous login" only at the initial stage of commissioning. and enable "Enable user name and password authentication" after commissioning to ensure communication security.

f. Click "State" to view the running state of the OPC UA server.



Parameter Name	Description
Server enable status	Display whether the OPC UA server is enabled.
Server run status	Display the running state of the OPC UA server. The definitions of status codes are as follows: 0: Server shut down 1: Server initializing 2: Server preparing 3: Server running 4: Server stopped due to error 5: Server shutting down
Server EndpointURL	The server address used for the client to access the server Server address format: opc:tcp://server IP address: server port number
Current number of sessions	The number of sessions currently connected to the server
Current subscription count	The number of current subscriptions

2. Set the communication variables of the OPC UA server.
   a. In the "Project Manager" tree, unfold "Global Variable" and then select "Variable_Table". Create a global variable, as shown in the following figure.
   b. Modify the "Network Public" property of the variables to "Public" or "IN/OUT" to register the variable to the OPC UA server.
3. Shut down the OPC UA server.
   - Modify the user project.
     In the "Project Manager" tree, choose "Config" > "OPC UA" > "OpcUA Server". In the section on the right, click "Convention", deselect "Start OpcUA server", compile and download the project. After you run or stop the PLC again, the OPC UA server is shut down.



- Run the OPCUA_Server_Shutdown instruction. For operation procedure, see the relevant instruction description in H5U and Easy Series Programmable Logic Controller Instructions Guide.

14.3 Application Example

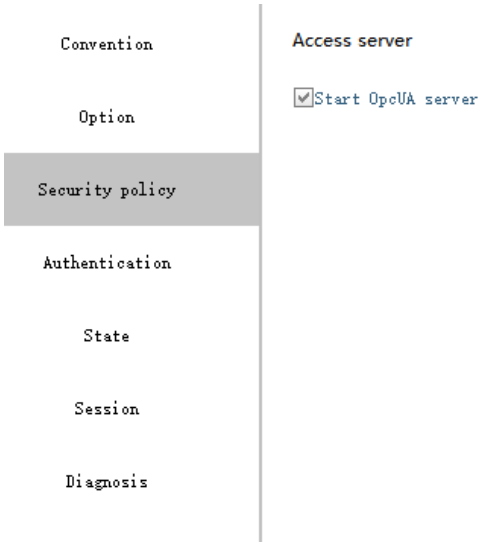
The OPC UA client "UaExpert" provided by Unified Automation is used as an example.

Configuring the OPC UA server

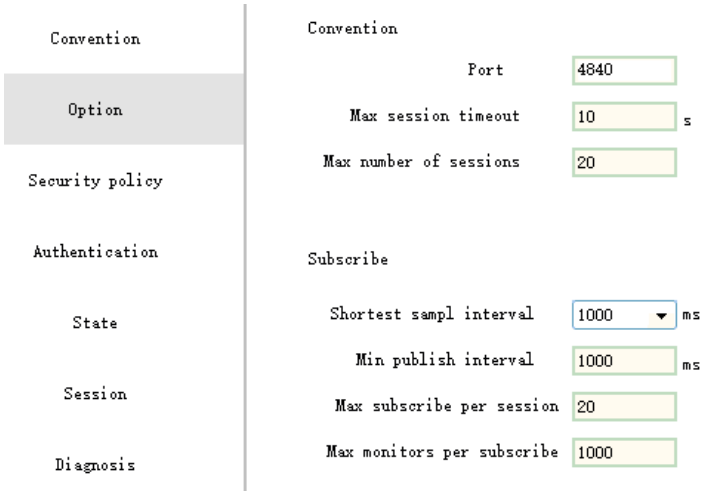
Enable the OPC UA server.

1. In the "Project Manager" tree, choose "Config" > "OPC UA" > "OpcUA Server". In the section on the right, click "Convention".

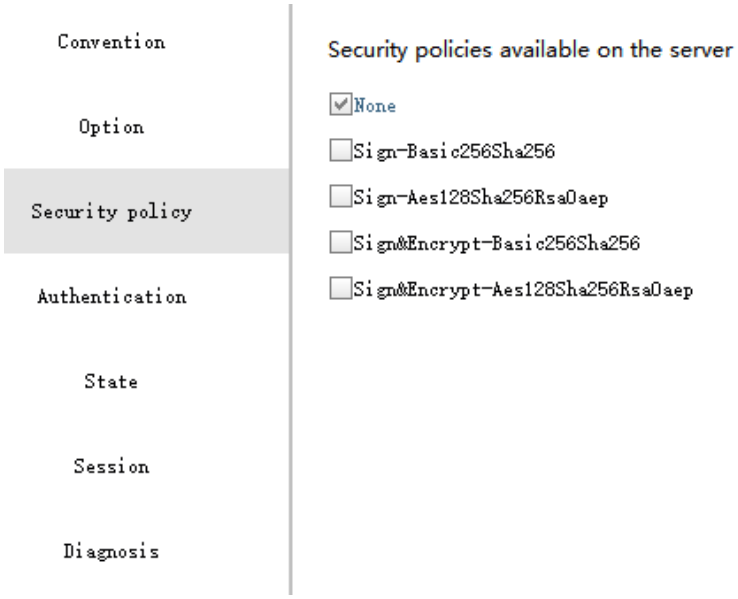
2. Select "Start OPC UA Server" to enable the OPC UA server.



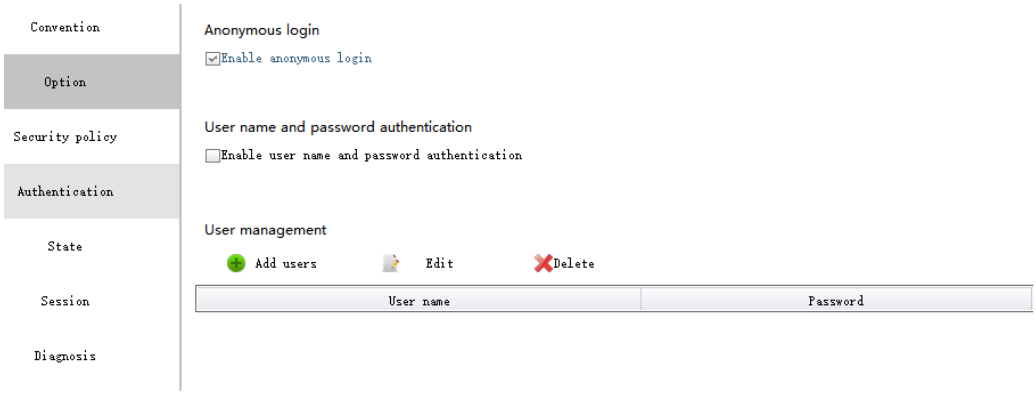
3. Click "Option" and configure parameters of the OPC UA server. In this example, all parameters are defaulted.



4. Click "Security policy" and configure an encryption policy for the OPC UA server. "None" is selected in this example.



5. Click "Authentication" and set an authentication mode for users on the OPC UA client to access the server. In this example, "Enable anonymous login" is selected.



6. In the "Project Manager" tree, unfold "Global Variable" and then select "Variable_Table". Create a variable table. In the following figure, global variables IntVar1, ArrayVar1, and StructVar1 are created.

NO.	Variable Name	Data Type	Initial Value	Power Down F	Network Publ
1	IntVar1	INT	0	Non Retained	Private
2	⊞ ArrayVar1	DINT[3]	...	Non Retained	Private
6	⊞ StructVar1	Stru_type2	...	Non Retained	Private
16					

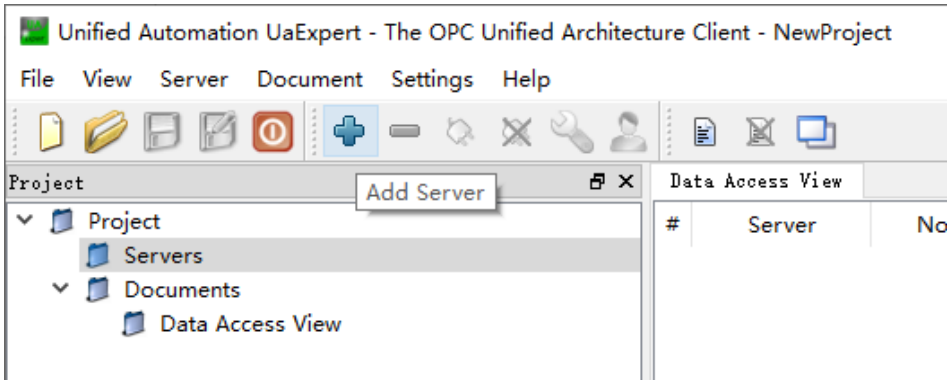
7. Modify the "Network Public" property of the variables to "Public" or "IN/OUT".

NO.	Variable Name	Data Type	Initial Value	Powert Down H	Network Public	Comment	Element Addr.
1	IntVar1	INT	0	Non Retained	Public		
2	⊟ ArrayVar1	DINT[3]	...	Non Retained	In/Out		
3	ArrayVar1[0]	DINT	0				
4	ArrayVar1[1]	DINT	0				
5	ArrayVar1[2]	DINT	0				
6	⊟ StructVar1	Stru_type2	...	Non Retained	Public		
7	member_1	BOOL	OFF				
8	member_2	INT	0				
9	⊟ member_array	DINT[3]	...				
10	member_array[0]	DINT	0				
11	member_array[1]	DINT	0				
12	member_array[2]	DINT	0				
13	⊟ member_stru	Stru_type1	...				
14	member_di	DINT	0				
15	member_r	REAL	0.000000				
16							

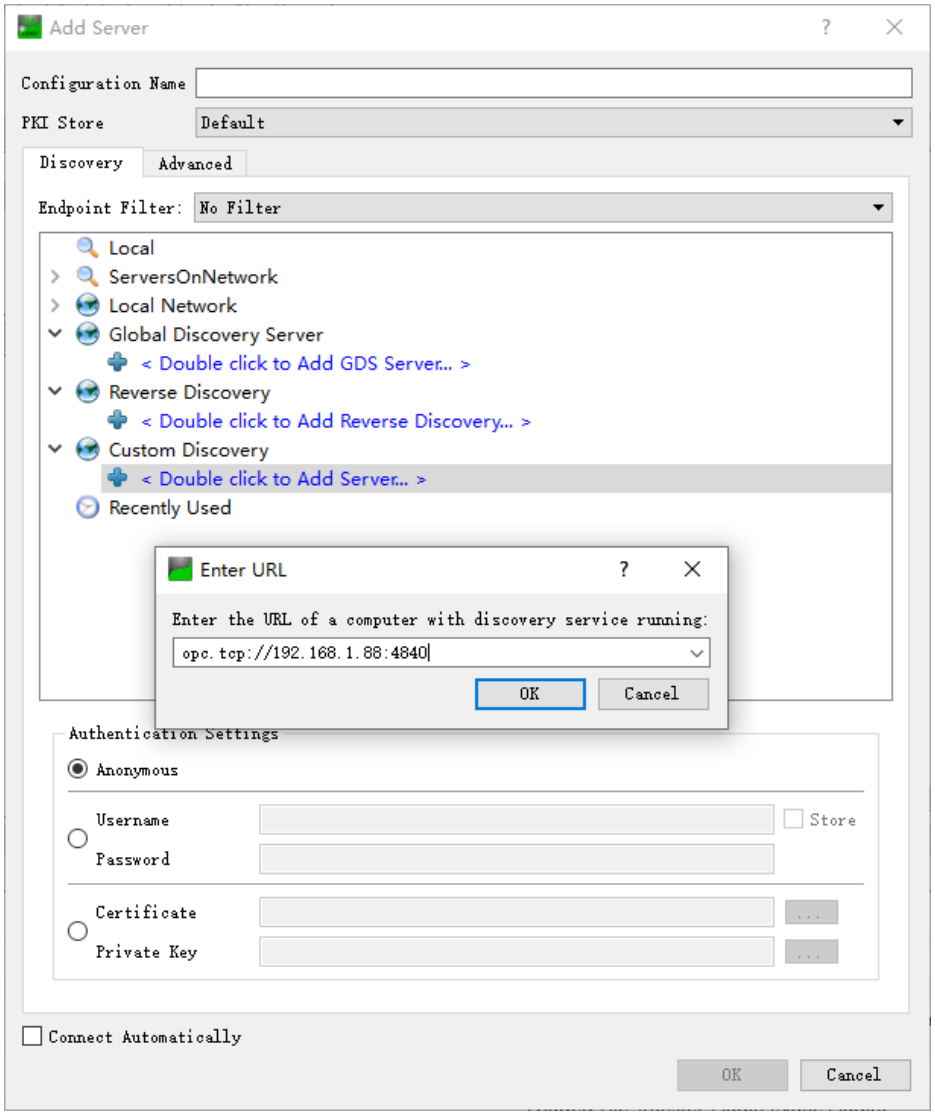
8. Compile and download the project to the PLC and then run the PLC.

Configuring the OPC UA client

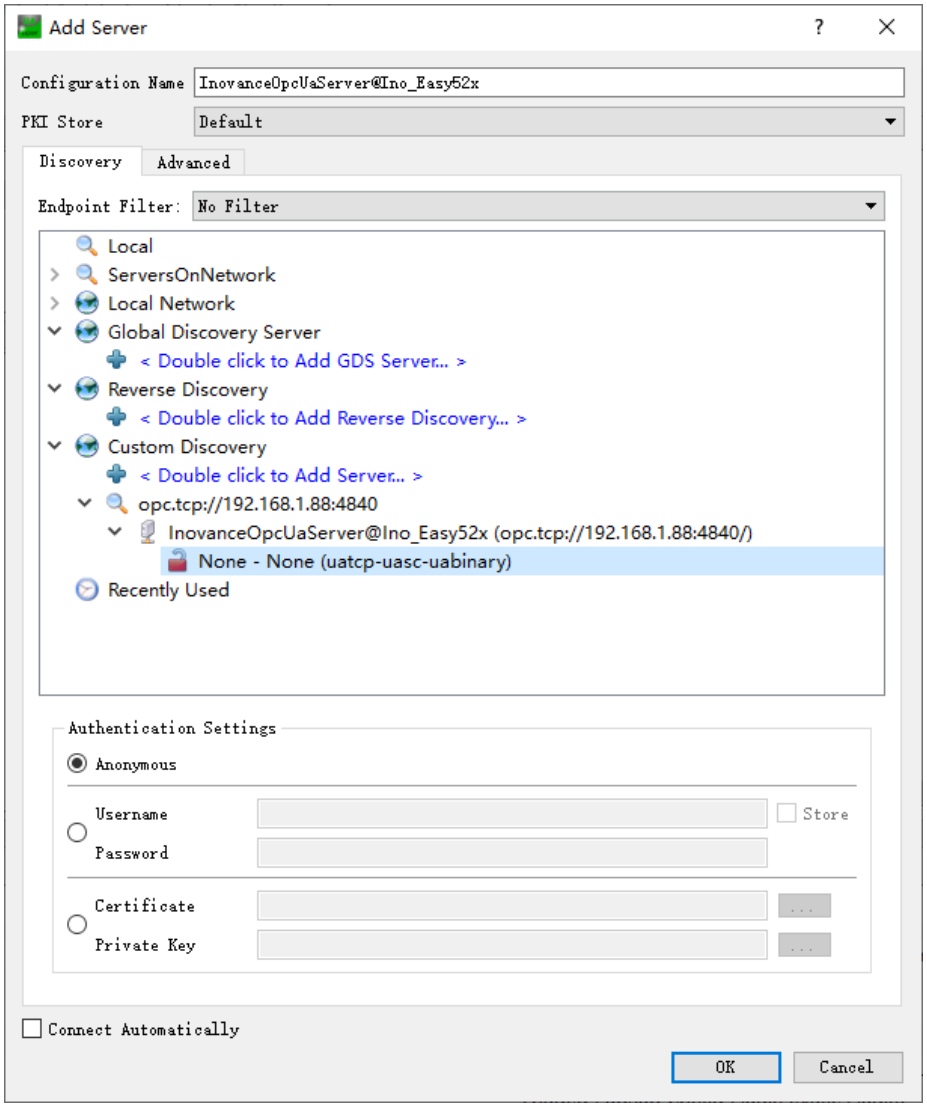
1. Add an OPC UA server station.
   a. Run the UaExpert software and click in the menu bar.



   b. In the "Add Server" dialog box displayed, click "Double click to Add Server" below "Custom Discovery". In the "Enter URL" dialog box displayed, enter the address of the OPC UA server of the Easy PLC.

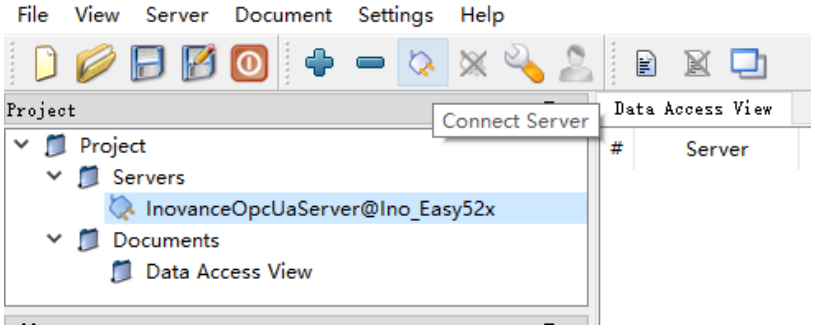


c. After the station is added, information of the OPC UA server of the Easy PLC is displayed below "Add Server". Then, you can select a safety policy and authentication mode, and then click "OK". In this example, "None-None(uatcp-uasc-uabinary)" and "Anonymous" are selected.

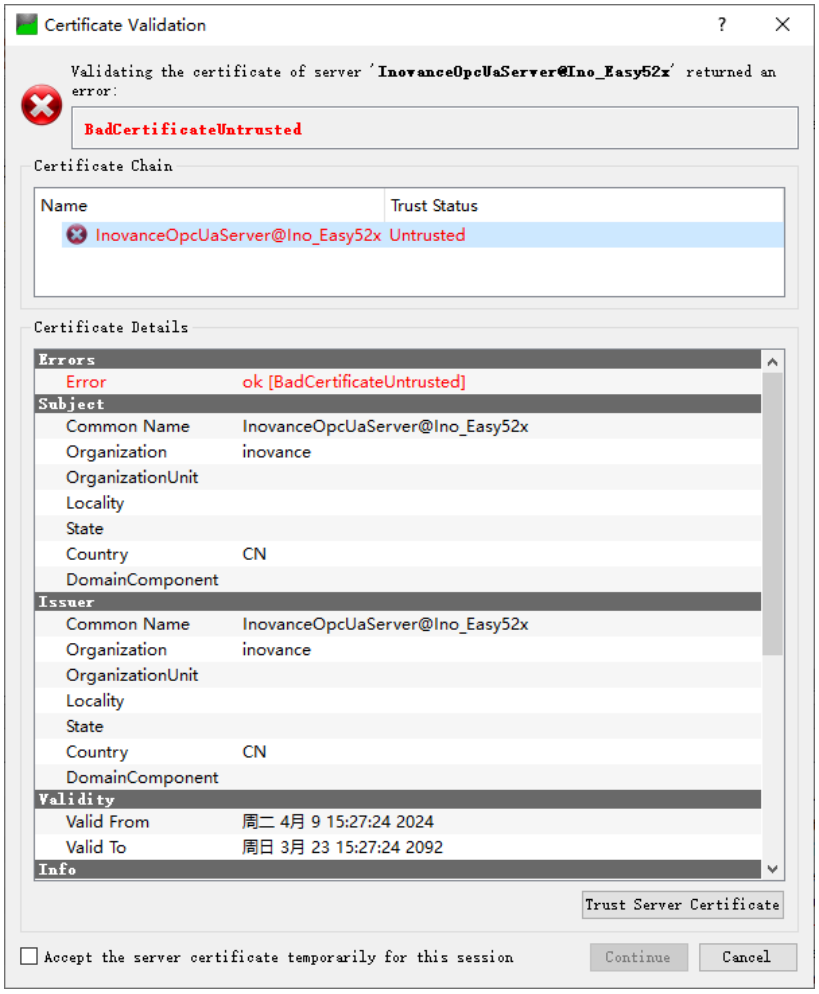


2. Connect the OPC UA server station.

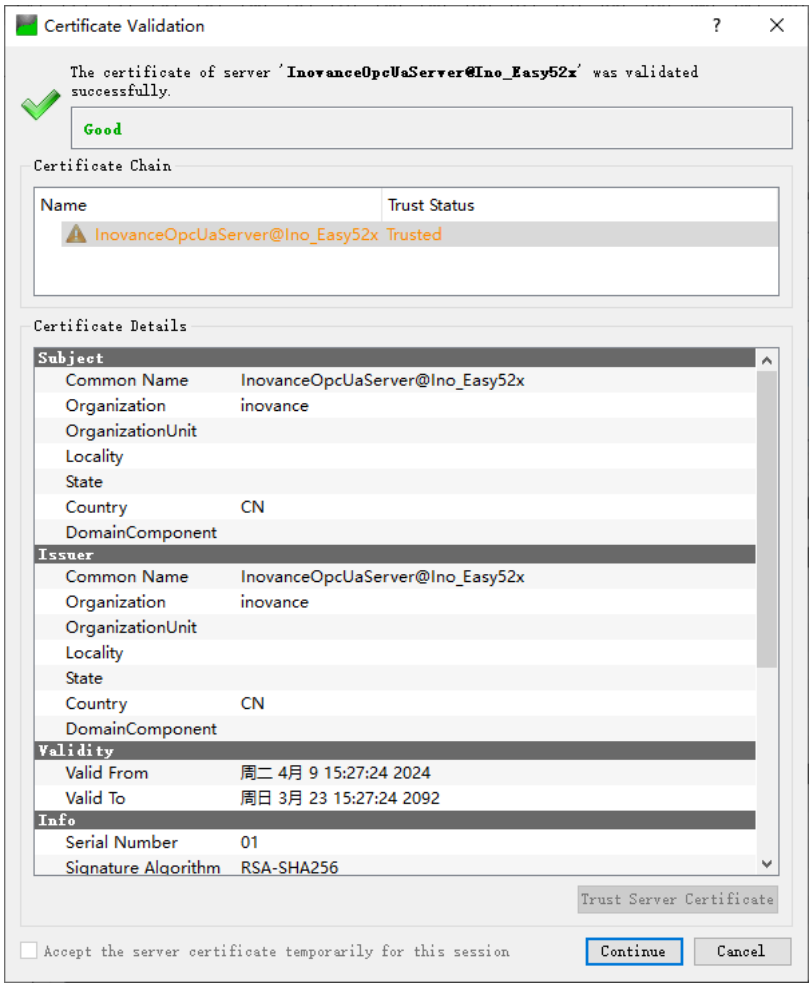
   a. Select the OPC UA server, and click in the menu bar.



   b. When the server is connected for the first time, UaExpert pops up a dialog box, requiring to verify the certificates of the communication parties.

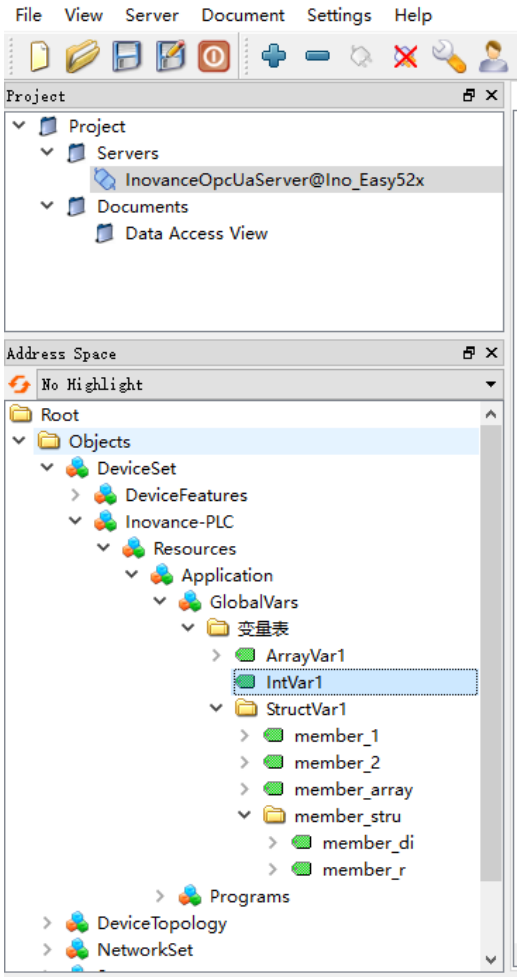


c. Click "Trust Server Certificate".

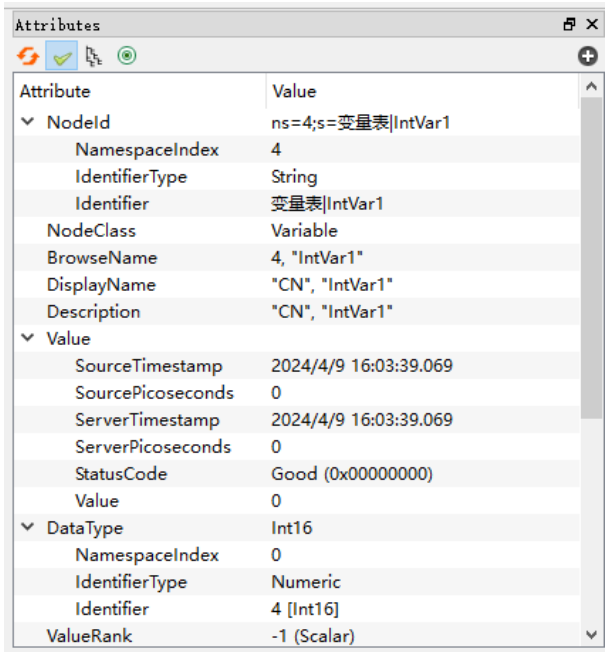


3. Access data of the OPC UA server.

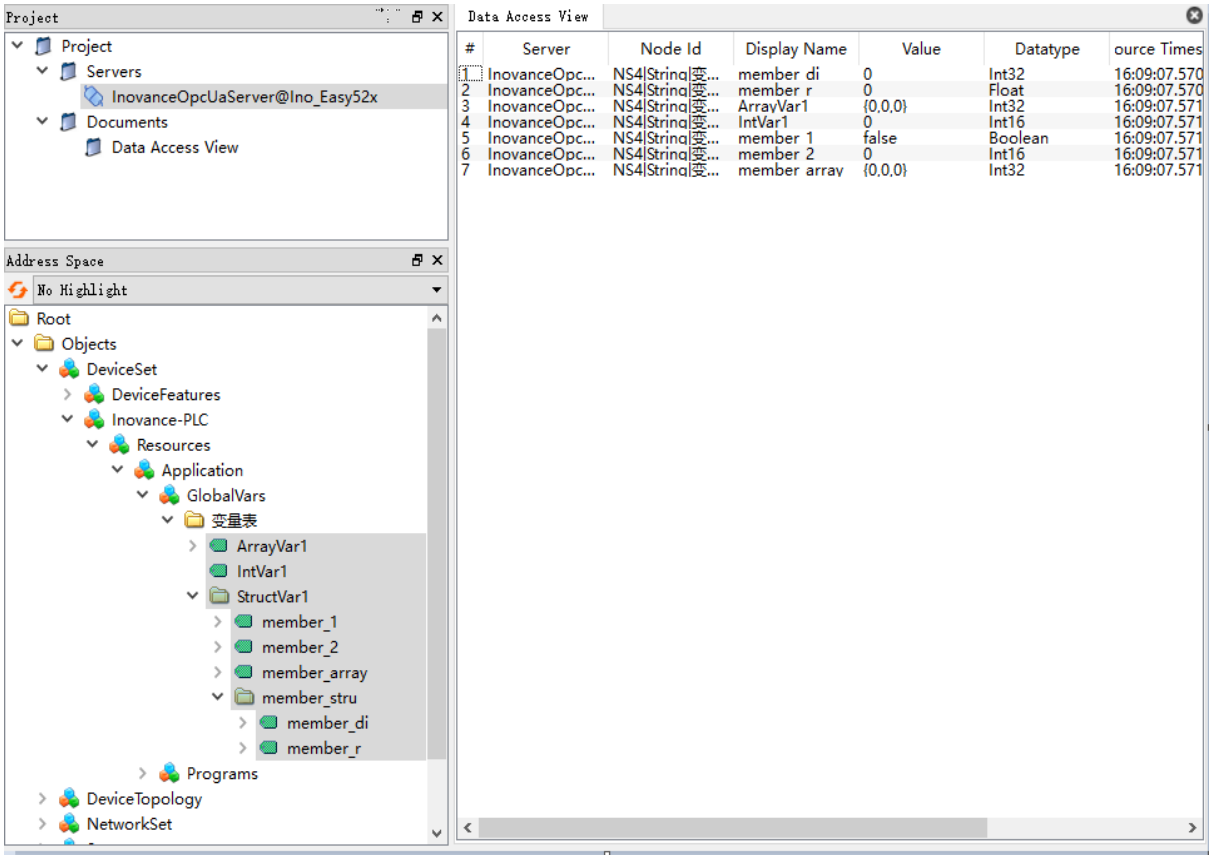
   After certificate is verified, you can view the information about the OPC UA server of the Easy PLC in the "Address Space" window.



Select a node and then view its properties in the "Attribute" window.



Drag a variable from the "Address Space" window to "Data Access View" to subscribe to or read/write this variable, as shown in the following figure.



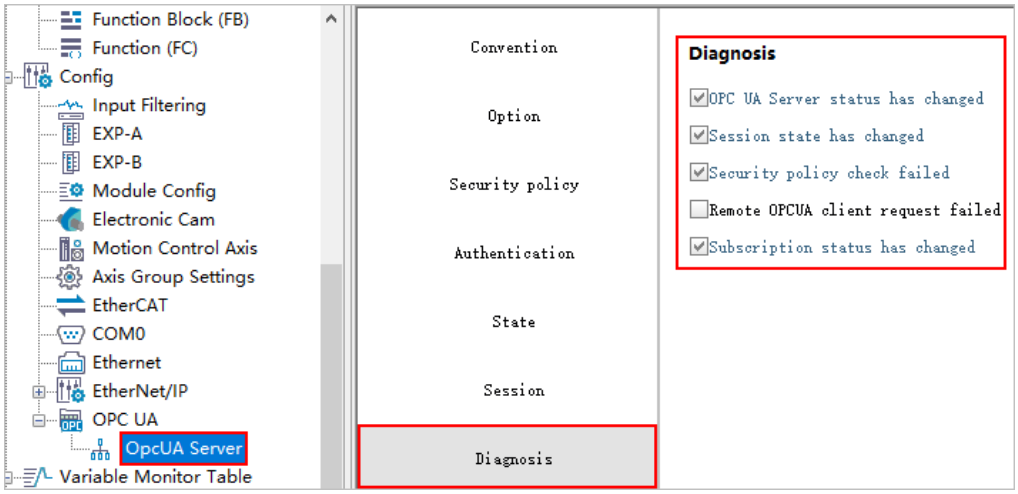
14.4 Configuring Fault Diagnosis Function

Preconditions

- Firmware version: V6.4.0.0 and later
- Software version: AutoShop V4.10.2.0 and later

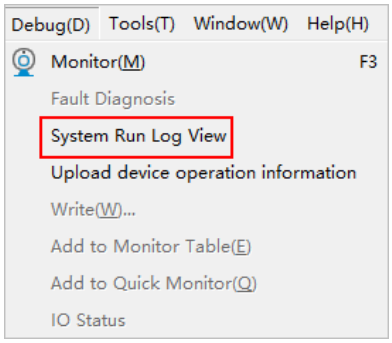
Procedure

1. Configure the fault diagnosis page. In the "Project Manager" tree, choose "Config" > "OPC UA" > "OpcUA Server". In the section on the right, click "Diagnosis" and select the desired diagnosis functions, as shown in the following figure.

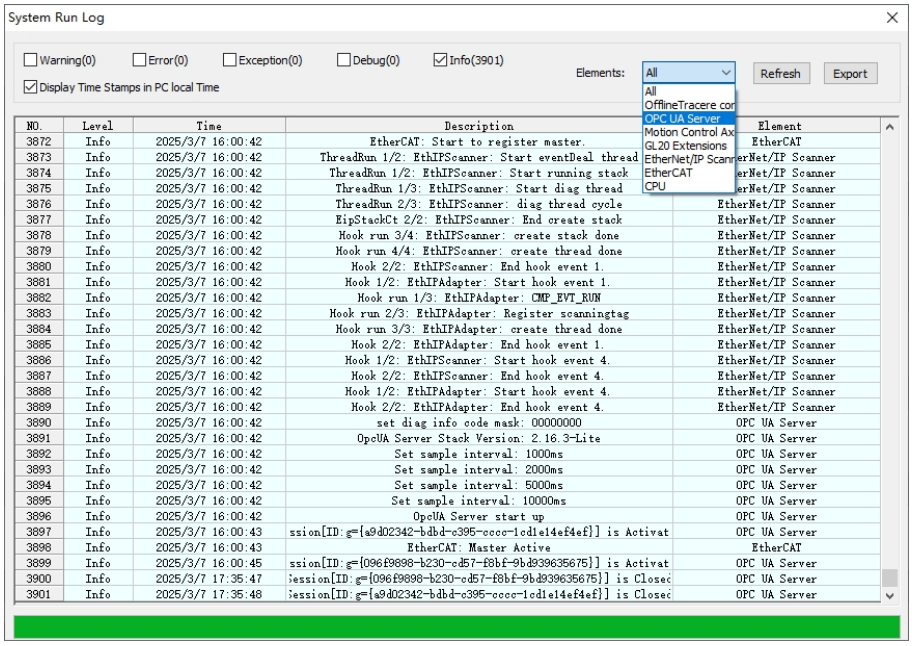


Name	Description
OPC UA Server status has changed	Detect the server running, stop, timeout status switchover and report the diagnosis result through the diagnosis code.
Session state has changed	Detect the status of created sessions and report the diagnosis result through the diagnosis code, such as creation, activation, timeout, and shutdown.
Security policy check failed	Report security policy check failure due to reasons such as incorrect account password, connection over not-supported security policy, and certification expiration. The diagnosis result is reported through diagnosis code and the number of rejected requests on the session tab is increased.
Remote OPCUA client request failed	Detect request failures of the OPC UA client. The diagnosis result is reported through the diagnosis code and the number of rejected requests on the session tab is increased.
Subscription status has changed	Detect the status of subscriptions created for the session between the client and the server, and the diagnosis result is reported through the diagnosis code.

2. View the diagnosis code. In the menu bar, choose "Debug" > "System Run Log View", as shown in the following figure.



3. In the "System Run Log" dialog box displayed, select "OPC UA Server" from the drop-down list of "Elements". Status codes of configured diagnosis report functions are displayed, as shown in the following figure.



The following table describes definitions of diagnosis codes.

Diagnosis Code	Diagnosis Description	Possible Cause	Solution
0x0001	Insufficient memory	Insufficient memory	Check whether any memory is available.
0x0002	Socket opening failure	Insufficient permission, insufficient system resources, and port occupied	1. Check that the network connection is normal. 2. View the log to check for specific error information.
0x0003	IP address unspecified	No IP address is specified when the socket is bound.	Check whether the configured IP address is correct.
0x0004	Socket binding failure	Insufficient permission, insufficient system resources, and IP address or port occupied	• View the log to check for specific error information. • Check whether the IP address is correct. • Check whether the port is occupied.
0x0006	hostname too long	The length of hostname exceeds 512 bytes.	Check whether the length of the server URL identifier configured exceeds 512 bytes.
0x0008	EndpointUrl invalid	The server endpoint URL configured is invalid.	Check whether the server URL identifier is normal. If the device name is used, check whether it is consistent with the actual device name. If the IP address is used, check whether the IP address configured is correct.
0x0009	Security policy empty	The security policy configuration is incorrect.	Check whether no security policy option is selected on the configuration page.
0x000A			

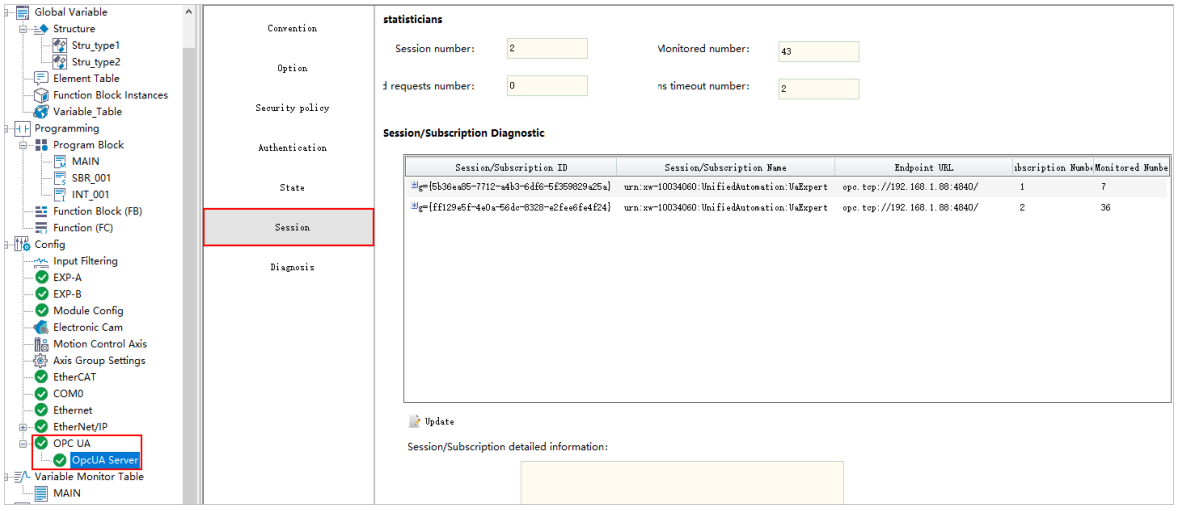
Diagnosis Code	Diagnosis Description	Possible Cause	Solution
	Failed to set security policy	The security policy configuration is incorrect.	Check whether the security policy is correctly configured on the configuration page.
0x000B	Mismatch between security mode and security protocol	When the security mode is not None, the security protocol is set to None (it cannot be None).	Check whether the security policy is correctly configured on the configuration page and takes effect.
0x000E	Parameter ports configured exceeding the limit	The network port of OPC UA is disconnected or the remote network is not reachable.	Check whether the number of ports set in IDE exceeds the limit.
0x000F	Maximum session timeout time exceeding the limit	The number of ports configured exceeds the limit.	Check whether the number of ports set in IDE exceeds the limit.
0x0010	Maximum number of sessions exceeding the limit	The maximum number of sessions configured exceeds the limit.	Check whether the number of ports set in IDE exceeds the limit.
0x0011	Sampling interval exceeding the limit	The sampling interval configured exceeds the limit.	Check whether the number of ports set in IDE exceeds the limit.
0x0012	Release interval exceeding the limit	The release interval configured exceeds the limit.	Check whether the number of ports set in IDE exceeds the limit.
0x0013	Maximum number of subscriptions per session exceeding the limit	The maximum number of subscriptions per session configured exceeds the limit.	Check whether the number of ports set in IDE exceeds the limit.
0x0014	Maximum number of monitoring items per subscription exceeding the limit	The maximum number of monitoring items per subscription configured exceeds the limit.	Check whether the setting exceeds the limit.
0x0015	Configuration file not exist	The INI configuration file is missing.	Refresh the device or upgrade the firmware if necessary.
0x0201	(Address space, node management) Memory used up	Insufficient memory	Check whether any memory is available.
0x0401	(Session service) memory used up	Insufficient memory	Check whether any memory is available.
0x0402	Too many sessions		

Diagnosis Code	Diagnosis Description	Possible Cause	Solution
		The number of session connections exceeds the limit.	Check whether the number of sessions connected to the server exceeds the limit.
0x0404	Session state changed to Created	Session created	Prompt information
0x0405	Session state changed to Activated	Session activated	Prompt information
0x0406	Session state changed to Timeout	Session timed out	Prompt information
0x0407	Session state changed to Closed	Session closed	Prompt information
0x0601	(Read/write, property) memory used up	Insufficient memory	Check whether any memory is available.
0x0602	Too many read/write operations	The number of read/write requests in a single frame exceeds the upper limit.	Check whether the number of nodes read/written in a single request at the client exceeds the maximum number of nodes that an be read/written in a single request at the server.
0x0603	Node value property unreadable	The node permission is unreadable when the variable value property is read.	Check whether "Public Variable" (OPC UA variable) is selected for the variable node on the configuration page.
0x0605	indexRange no value	No data is read for the specified indexRange during node property reading.	• Check whether the value variable in the server is an array, whether the array size is matching, or whether any value is null. • Check whether the read request of the client is within the array range.
0x0606	Node value property not writable.	The node permission is not writable when the variable value property is written.	• Check whether "Public Variable" (OPC UA variable) is selected on the configuration page. • Check whether the variable is writable.
0x0801	(Subscription, monitoring) memory used up	Insufficient memory	Check whether any memory is available.
0x0802	Too many subscriptions	The number of subscriptions in a single session exceeds	• Check whether the number of session subscriptions of the server exceeds the

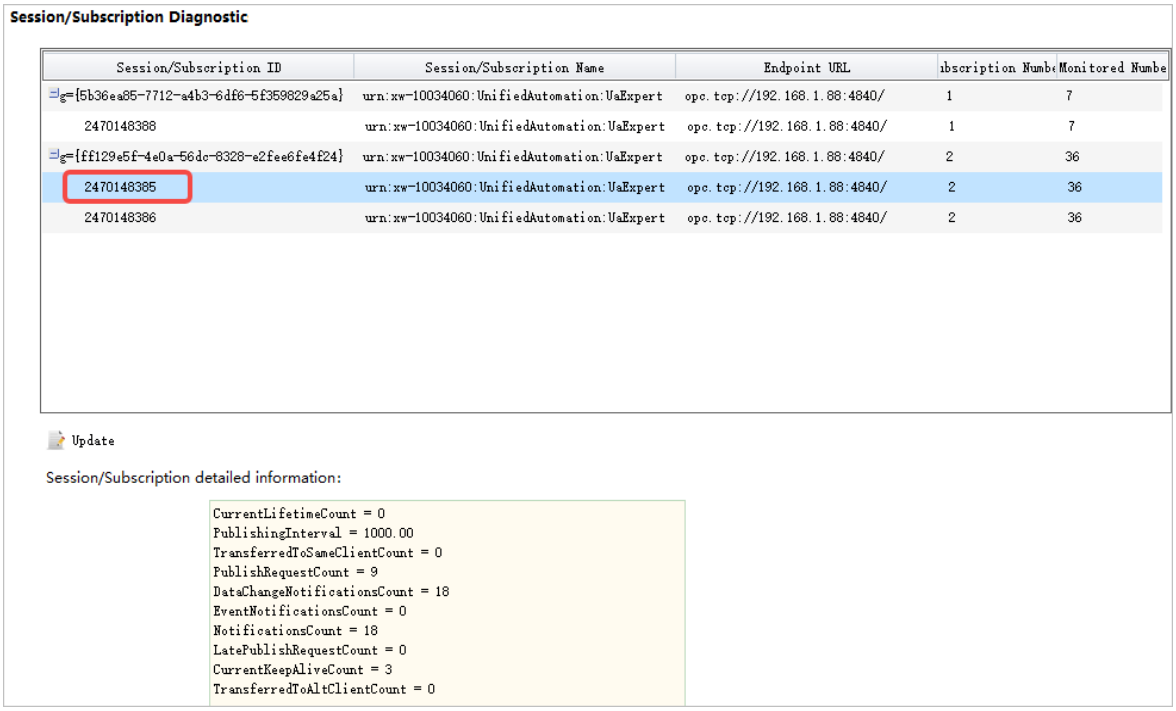
Diagnosis Code	Diagnosis Description	Possible Cause	Solution
		the upper limit during subscription creation or transfer.	maximum number. Check whether it matches the product specifications. • Check whether the total subscription count reaches the product specifications.
0x0803	Subscription state changed to Creating	Subscription creation	Prompt information
0x0804	Subscription state changed to Normal	Subscription creation	Prompt information
0x0805	Subscription state changed to Late	Subscription creation	Prompt information
0x0806	Subscription state changed to Closed	Subscription closed	Prompt information
0x0C01	(Browse service) memory used up	Insufficient memory	Check whether any memory is available.

4. View the server self-diagnosis information.
   You can only view the server self-diagnosis information on the page, including statisticians, session/subscription diagnosis information, and session/subscription details.

   a. In the "Project Manager" tree, choose "Config" > "OPC UA" > "OpcUA Server". In the section on the right, click "Session", as shown in the following figure.



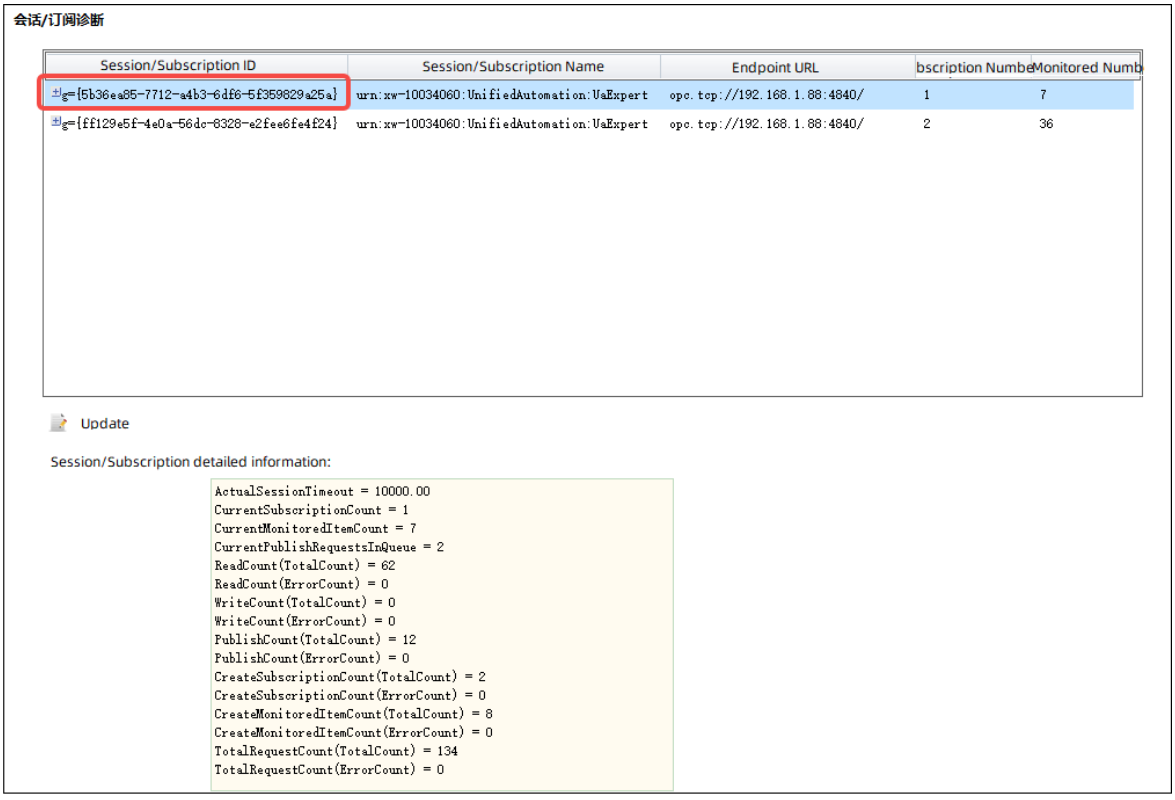
   b. Click a session ID. The details of all subscriptions to the current session are displayed in the "Session/Subscription details information" section, as shown in the following figure.



Name	Description
ActualSessionTimeout	The request session specified by the client in the open session call timed out.
CurrentSubscriptionsCount	Number of subscriptions to the current session
CurrentMonitoredItemsCount	Number of monitoring items of the current session
CurrentPublishRequestsInQueue	Number of current publish requests in the session queue
ReadCount(TotalCount)	The counter of the current session reading service, which identifies the total number of requests for this service received in the session
ReadCount(ErrorCount)	The error request counter of the current session reading service, which identifies the number of errors of the service request received in the session.
WriteCount(TotalCount)	The counter of the current session writing service, which identifies the total number of requests for this service received in the session
WriteCount(ErrorCount)	The error request counter of the current session writing service, which identifies the number of errors of the service request received in the session.
PublishCount(TotalCount)	The counter of the current session publishing service, which identifies the total number of requests for this service received in the session
PublishCount(ErrorCount)	The error request counter of the current session publishing service, which identifies the number of errors of this service request received in the session
CreateSubscriptionCount(Total Count)	The counter of subscriptions created by the current session, which identifies the total number of requests for the service received in the session

Name	Description
CreateSubscriptionCount(ErrorCount)	The error request counter of subscriptions created by the current session, which identifies the number of errors of this service request received in the session
CreateMonitoredItemsCount(TotalCount)	The counter of monitoring service created by the current session, which identifies the total number of requests for the service received in the session
CreateMonitoredItemsCount(ErrorCount)	The error request counter of monitoring service created by the current session, which identifies the number of errors of this service request received in the session
TotalRequestCount(TotalCount)	The counter of all service request of the current session, which identifies the total number of any service request received in the session
TotalRequestCount(ErrorCount)	The error counter of all service requests of the current session, which identifies the number of errors of any service request received in the session

c. Click "+" on the left of a session to unfold the session details. Click a subscription to view the subscription ID, as shown in the following figure.



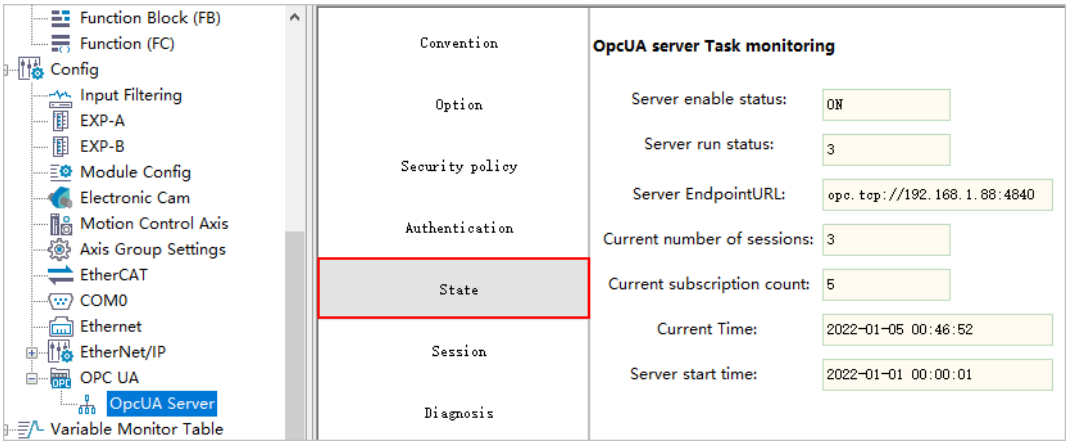
Name	Description
CurrentLifetimeCount	The current lifetime count of the current subscription
PublishingInterval	The publishing interval of the current subscription, in ms
TransferredToSameClientCount:	The number of times the current subscription transferred to a standby session of the same client
PublishRequestCount	The publishing interval of the current subscription, in ms

Name	Description
DataChangeNotificationsCount	Number of data change notifications sent by the subscription
EventNotificationsCount	Number of event notifications sent by the subscription
NotificationsCount	Number of notifications sent by the subscription
LatePublishRequestCount	Number of times the subscription entered the LATE state, that is, the number of times the publishing timer expired and the notification was not sent
CurrentKeepAliveCount	Number of times the subscription entered the KeepAlive state
TransferredToAltClientCount	Number of times the subscription was transferred to an alternate client

d. Click "Update". The latest diagnosis information is displayed, with session and subscription monitoring added, as shown in the following figure.

5. Monitor the server state. The OPC UA server state can be monitored. You can view the server enabling state, number of current sessions, number of current subscriptions, endpoint URL, current time, and server start time.

a. In the "Project Manager" tree, choose "Config" > "OPC UA" > "OpcUA Server". In the section on the right, click "State", as shown in the following figure.



b. To update the state, click

in the menu bar or press F3 to enable the monitoring function.

15 MC Communication

15.1 Overview

Preconditions

- PLC: Easy523, Easy522, Easy521, Easy320, and Easy302
- Firmware version: 6.4.0.0 and later
- Software version: 4.10.0.0 and later

Introduction

The Mitsubishi MC Protocol is a communication protocol developed by Mitsubishi Electric for industrial automation equipment, particularly applied within its PLC systems. This protocol is commonly used to enable communication between PLCs over network environments such as Ethernet, RS232, and RS485. It also supports connections between PLCs and external devices (such as computers, touchscreens, and human-machine interfaces).

Communication specifications:

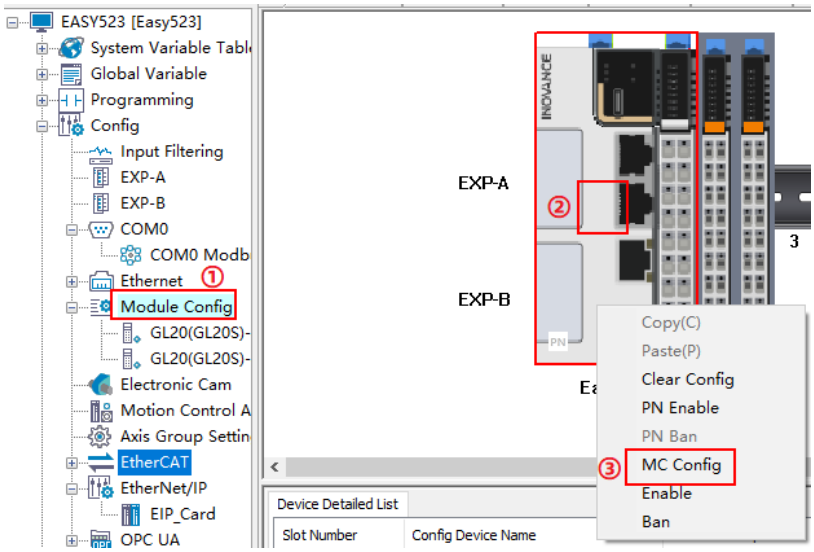
- Support 3E and 4E frames.
- Support two data encoding types: ASCII and binary.
- Support MC communication protocols based on TCP and UDP.

> NOTE
>
> In TCP communication, a maximum of 32 master connections are supported (limited by the PLC's own TCP protocol).

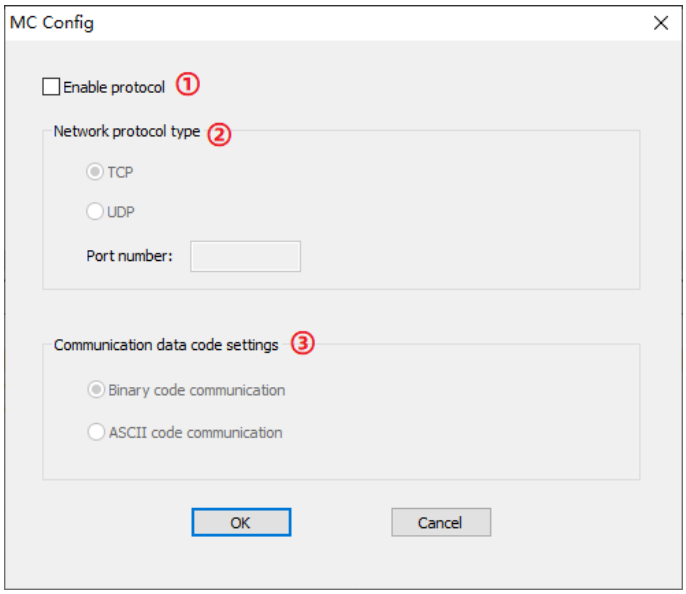
15.2 Configuration

Procedure

1. In the "Project Manager" tree on the left, choose "Config" > "Module Config". On the left area of the "Module Config" page, click the picture of a PLC. In the menu displayed, select "MC Config".



2. On the "MC Config" page displayed, select "Enable protocol" to enable the MC server for the PLC. In the "Network protocol type" section, set the communication protocol and port. In the "Communication data code settings" section, set the encoding format of the MC data and click "OK".



The following table lists the speed compensation parameters.

Parameter Name		Description
Enable MC protocol		Enable or disable the MC server.
Network protocol type	TCP	Set the communication protocol to TCP.
	UDP	Set the communication protocol to UDP.
	Port number	Set the port for the MC server. The value ranges from 1025 to 4999 and 5010 to 49151. The following ports cannot be set: • FINS: 9600 • EtherNet/IP: 44818 and 2222 • EtherCAT: 34980 • Tag communication: 12939 and 12940
Communication data code settings	Binary code communication	Set the communication data encoding format to binary.
	ASCII code communication	Set the communication data encoding format to ASCII.

3. Then, click to download the program to the PLC. The MC server is enabled for the PLC.

15.3 List of System Variables

Name	Description
McSlvEn	MC server enable status 0: Not started 1: Started
McNetworkType	Network communication type 0: TCP 1: UDP
McEncodingType	Encoding mode 0: Binary code 1: ASCII code
McSlvPort	MC service port The value ranges from 1025 to 4999 and 5010 to 49151. The following ports cannot be set: • FINS: 9600 • EtherNet/IP: 44818 and 2222 • EtherCAT: 34980 • Tag communication: 12939 and 12940
connected	Connection state 0: No TCP connection 1: TCP connection available
connections	Number of connected TCP clients
IPAddress	List of TCP client IP addresses

15.4 Supported Specifications

Supported instruction codes and access points

Operation	Instruction	Sub-instruction		Number of Access Points	
		Q/L Series	iQ-R Series	ASCII	Binary
Batch read (word)	0401	0000	0002	1 to 960 (word)	
Batch read (bit)	0401	0001	0003	1 to 3584 (bit)	1 to 7168 (bit)
Batch write (word)	1401	0000	0002	1 to 960 (word)	

Operation	Instruction	Sub-instruction		Number of Access Points	
		Q/L Series	iQ-R Series	ASCII	Binary
Batch write (bit)	1401	0001	0003	1 to 3584 (bit)	1 to 7168 (bit)
Random read (word)	0403	0000	0002	1 ≤ (word access points + double word access points) ≤ 96 (word)	
Random write (word)	1402	0000	0002	1 ≤ (word access points × 12 + double word access points × 14) ≤ 960 (word)	
Random write (bit)	1402	0001	0003	1 to 94 (bit)	

Supported soft elements

Inovance Soft Elements	Type	Range	Number of Points	Corresponding Mitsubishi Soft Elements
X	Bit	X0 to X1777	1024, octal	X
Y	Bit	Y0 to Y1777	1024, octal	Y
M	Bit	M0 to M7999	8000	M
S	Bit	S0 to S4095	4096	L
B	Bit	B0 to B32767	32768	B
D	Word	D0 to D7999	8000	D
R	Word	R0 to R32767	32768	D
W	Word	W0 to W32767	32767	W

NOTE

- The number of access points refers to the number of soft elements that can be operated by a single instruction.
- To access the R soft element of an Inovance PLC over the MC protocol, construct the MC request frame as if accessing a D soft element. The PLC automatically handles the unified addressing of D and R soft elements. For example, to access the soft element R0, you need to access D8000.
- To access the S soft element of an Inovance PLC over the MC protocol, construct the MC request frame as if accessing an L soft element.
- Note the base conversion between Mitsubishi soft element codes and Inovance soft element codes.

15.5 Fault Codes

Fault Code	Description	Remarks
0000	The read/write operation is normal.	-
4031	The number of read/written soft elements exceeds the allowed range.	-
	The number of batch write operations exceeds the allowed range.	-
	The start address of the target device is outside the readable/writable range.	-
C050	The received soft element code is invalid.	ASCII mode only
	The write value is abnormal for bit-unit operation.	-
C051	The operation length exceeds the maximum number of points per transaction (batch read/write - word unit).	-
C052	The operation length exceeds the maximum number of points per transaction (batch read/write - bit unit).	-
C053	The operation length exceeds the maximum number of points per transaction (random write - bit unit).	-
C054	The operation length exceeds the maximum number of points per transaction (batch read/write - word unit).	-
C056	The soft element code in the batch read instruction is invalid.	Binary mode only
C058	The write data length does not match the request data length in the frame structure (total frame length).	ASCII mode only
C059	Unrecognized instruction/sub-instruction	-
	Illegal bit-level operation is performed on a word soft element.	-
C05B	The received soft element code is invalid.	Binary mode only
C05C	Illegal bit-level operation is performed on a word soft element.	-
C060	The write value is abnormal for bit-unit operation.	Binary mode only
C061		-

Fault Code	Description	Remarks
	The write data length does not match the request data length in the frame structure (total frame length).	
	The write data length does not match the soft element operation length in the frame structure (total frame length).	-
No response	The received sub-header is not a valid sub-header for 3E/4E frames.	-
	The write data length does not match the request data length in the frame structure (total frame length).	-
	The write data length does not match the soft element operation length in the frame structure (total frame length).	-

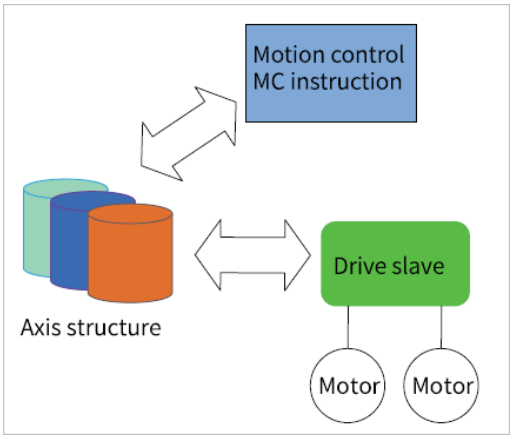
16 Motion Control

16.1 Introduction of Motion Control Axis

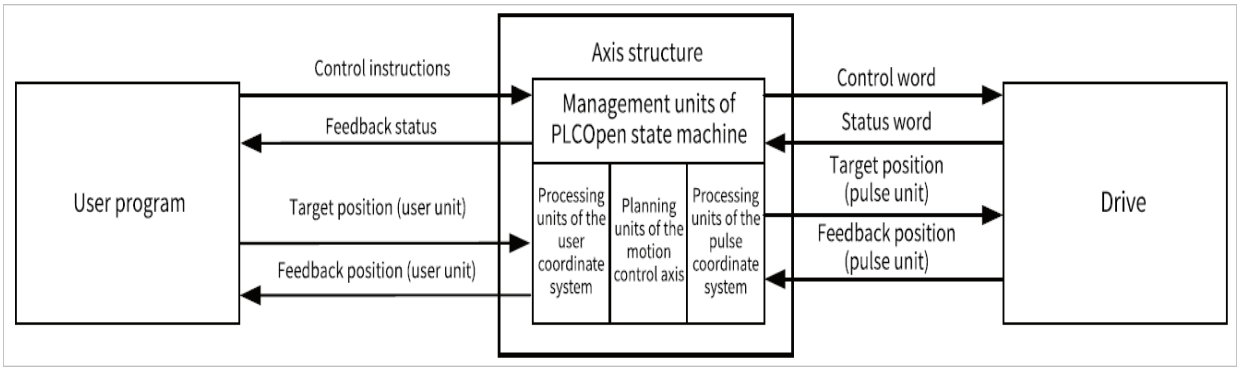
16.1.1 Overview

Basic composition and control logic

In a motion control system, the objects of motion control are called axes. The axes are the bridge between a drive and the PLC instructions. The motion control axes are used to control EtherCAT bus drive that is compliant with the CiA 402 protocol, as well as local high-speed pulse outputs and high-speed pulse inputs.



The following figure shows the basic composition and processing logic of an axis in the PLC.



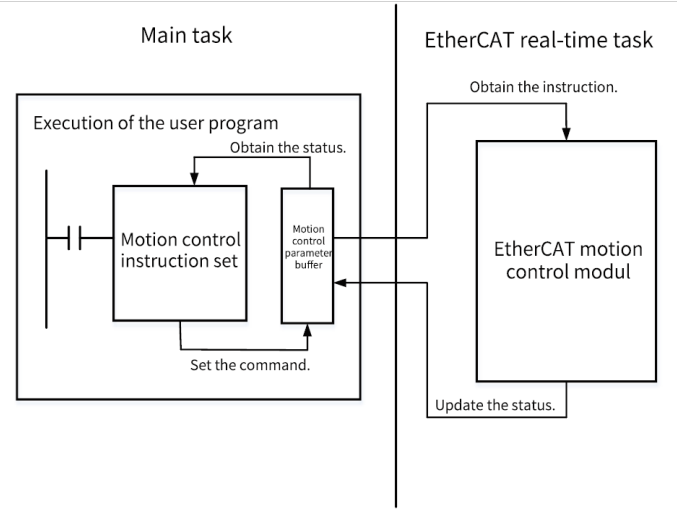
Scheduling mechanism of motion control instructions

Main programs, subprograms, and interrupt subprograms are provided for users to write programs. However, motion control instructions can only be called in the main programs or subprograms, not in the interrupt subprograms.

The EtherCAT tasks are hidden tasks that are not open to users, so programming of the EtherCAT tasks is not supported.

The following figure shows the execution process of EtherCAT instructions. In a main program, the PLC scans all the motion control instructions written in the program in turn, and stores the final result in the motion control parameter buffer according to the interrupt rules of the program. The PLC updates the motion control instruction when a EtherCAT task is executed. After the execution is completed, the execution result is put into the buffer, and the motion control instruction in the main program updates the instruction status according to the execution result.

For example, there are two MC_MoveAbsolute instructions in the program. The target position of the first instruction is 100, the target position of the second instruction is 200, and both instructions are triggered by the soft element M1000 at the same time. When the PLC scans the program, the PLC first scans the first absolute positioning instruction, obtaining the target position 100, and then scans the second instruction, updating the target position to 200. At the end of the main task, the PLC finally writes the target position 200 to the motion control buffer and implements the instruction according to the second absolute positioning parameter, interrupting the first instruction. After obtaining the target position 200, the EtherCAT task starts to execute the absolute positioning algorithm, and sets the completion flag when the positioning is completed. After the second absolute positioning instruction in the main program obtains the completion flag, the Done signal is activated.



Axis Type

Supported axis types include bus servo axis, local pulse axis, bus encoder axis, and local encoder axis.

Axis Type		Content
Bus servo axis		• Bus servo axis is controlled using EtherCAT slave servo drives. • When virtual axis mode is disabled, the bus servo axis is assigned to the actual servo drive for use. • The bus servo axis supports the control of several basic modes such as torque, point, velocity, and homing modes.
Local pulse axis	Easy302 Easy320 Easy501/502 Easy521/522/523	• Local pulse axis is controlled using a pulse drive controlled by local high-speed I/O. • Five local pulse axes can be set: Y0/Y1, Y2/Y3, Y4/Y5, Y6/Y7, Y0 and Y1. • For Y0/Y1, Y2/Y3, Y4/Y5, and Y6/Y7, each pulse output channel can be set to pulse+direction, CW/CCW, or AB phase. For Y0 and Y1, the pulse output channel can be set to pulse. • For Y0/Y1, Y2/Y3, Y4/Y5, and Y6/Y7, up to two probe terminals can be set per pulse output channel. In single pulse output mode, Y0 and Y1 do not support probes.

Axis Type		Content
		• The local pulse axis supports control in several basic modes such as point, velocity, and homing modes. Torque mode is not supported.
	Easy301	• Local pulse axis is controlled using a pulse drive controlled by local high-speed I/O. • Four local pulse axes can be set: Y0/Y1, Y2/Y3, Y0, Y1, Y2, and Y3. • For Y0/Y1 and Y2/Y3, each pulse output channel can be set to pulse+direction, CW/CCW, or AB phase. For Y0, Y1, Y2, and Y3, the pulse output channel can be set to pulse. • For Y0/Y1, Y2/Y3, Y0, Y1, Y2, and Y3, up to two probe terminals can be set per pulse output channel. • The local pulse axis supports control in several basic modes such as point, velocity, and homing modes. Torque mode is not supported.
	H5U	• Local pulse axis is controlled using a pulse drive controlled by local high-speed I/O. • Four local pulse axes can be set: Y0/Y1, Y2/Y3, Y4/Y5, and Y6/Y7. • Each pulse output channel can be set to pulse+direction, CW/CCW, or AB phase. • Up to two probe terminals can be set per pulse output channel. • The local pulse axis supports control in several basic modes such as point, velocity, and homing modes. Torque mode is not supported.
Bus encoder axis		Reserved
Local encoder axis		See Introduction to High-Speed Counter Axes.

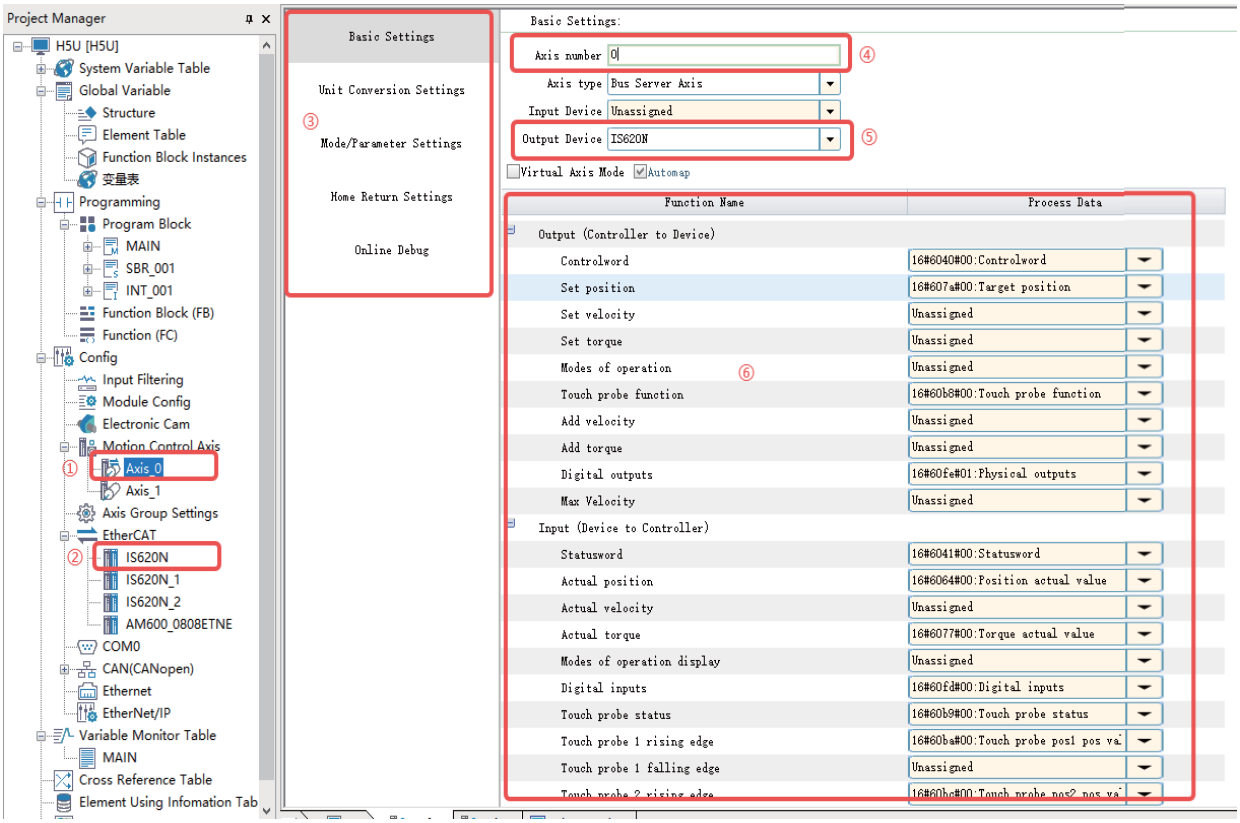
To fully describe the attributes of an axis, monitor the axis status, and control the axis motion, each axis is divided into three parts.

Axis Structure	Function
Axis configuration parameter	Configures parameters of an axis, such as gear ratio, homing type, and encoder mode.
Axis system variable	Monitors the operating status and abnormal information of an axis, such as the current position and axis error code.

Axis Structure	Function
Axis control instruction	In a user program, axis motion control is performed using MC motion control instructions. Axis control instructions are divided into management (such as MC_Power), motion (such as MC_Jog), and status (MC_ReadStatus).

Configuration page

In the project, the axis configuration page is as follows:

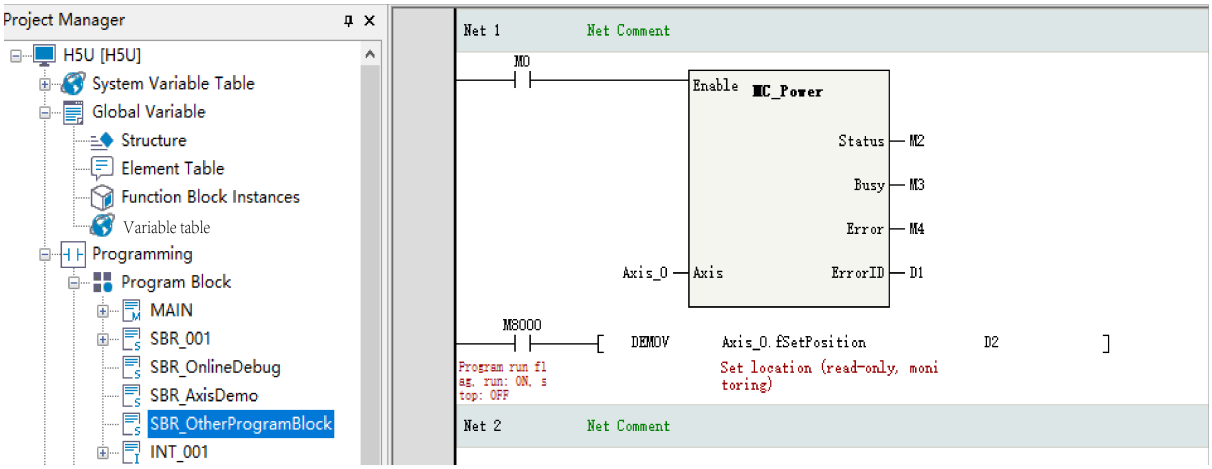


No.	Name
①	Motion Control Axis
②	EtherCAT Bus Drive
③	List of axis configuration and monitoring options
④	Axis number (unique access ID for an axis)
⑤	Associated physical drive
⑥	Detailed parameter settings

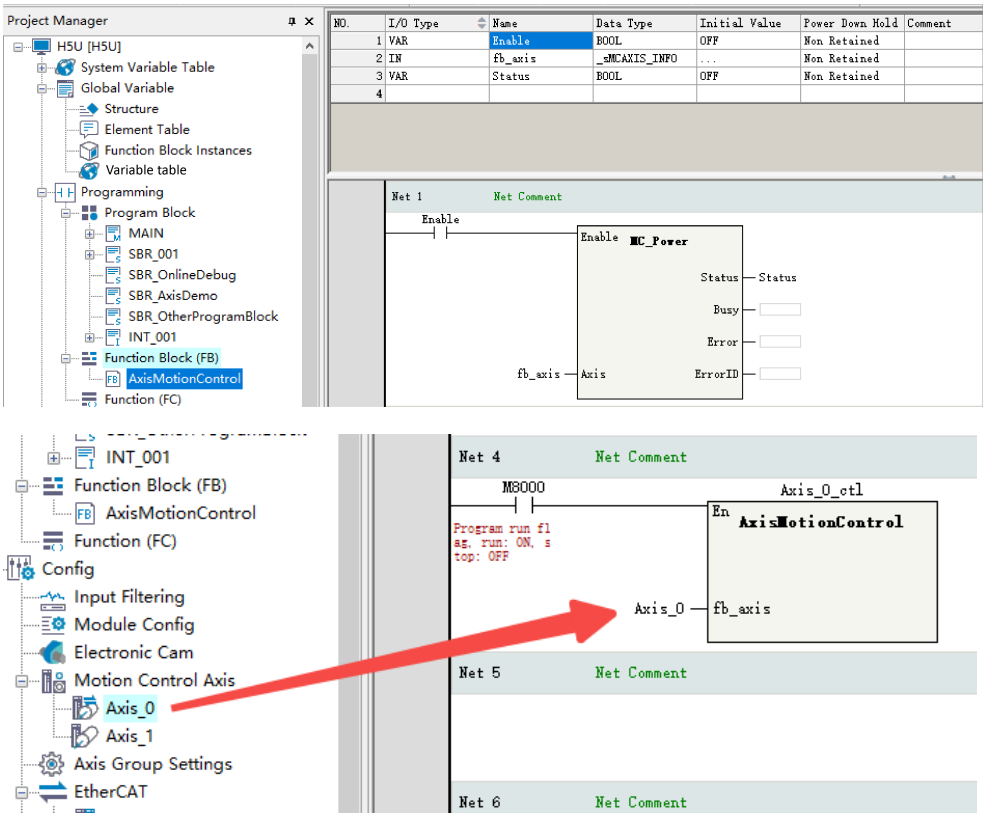
Motion control axis access modes

In the PLC program, a motion control axis can be accessed in two ways: motion control instructions and system variables.

- In AutoShop of version V4.0.0.0 or later versions in combination with PCB software of version V3.0.0.0 or later versions, axis instructions and system variables can be accessed by using axis names. Axis names can also be introduced into FB as parameters, for example:



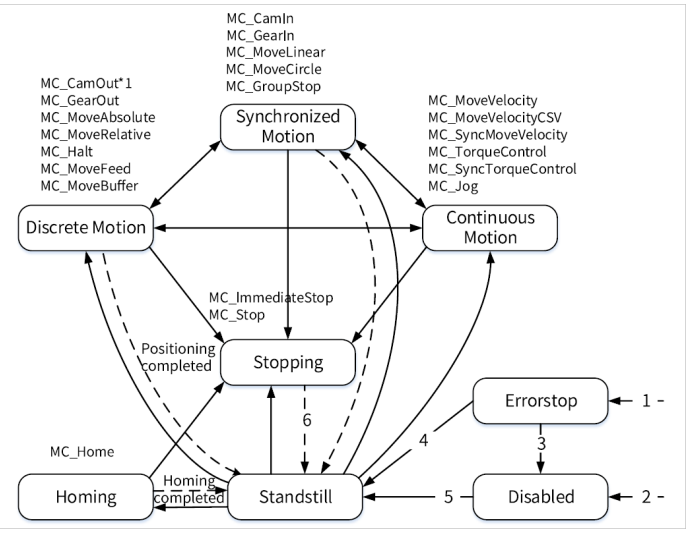
Axes can also be introduced into FB as parameters.



16.1.2 PLCopen State Machine

The PLCOpen state machine allows you to manage axis status and motion and complete different functions in different states .

The status transition diagram is as follows.



The detailed description is as follows.

Status value	Status	Description
0	Disabled	Disabled
1	ErrorStop	Fault reaction
2	Stopping	Stopping
3	StandStill	Enabled
4	Discrete motion	Discrete motion
5	Continuous motion	Continuous motion
7	Homing	Homing
8	Synchronized motion	Synchronized motion

The following table summarizes the status transition conditions.

Transition	Transition condition
1	The fault detection logic of the axis has detected a fault.
2	The axis is free of faults and MC_Power.Enable=FALSE
3	MC_Reset is called to reset the axis fault and MC_Power.Status=FASLE.
4	MC_Reset is called to reset the axis fault and MC_Power.Status=TRUE.
5	MC_Power.Enable=TRUE and MC_Power.Status=TRUE.
6	MC_Stop(MC_ImmediateStop).Done=TRUE and MC_Stop(MC_ImmediateStop).Execute=FALSE.

16.1.3 Axis Units

Two units are used in the axis structure: user unit and pulse unit.

User unit

It refers to the measurement units used in instructions, such as millimeters, centimeters, and angles, which are called user units.

The user coordinate systems are divided into linear coordinate system and rotary coordinate system according to working conditions.

A linear coordinate system typically includes a zero point. An axis is in forward motion when the target position increases, or in reverse motion when the target position decreases. The linear coordinate system can set positive and negative software limits.

The rotary coordinate system includes a zero point and a rotation cycle in which CW motion occurs when the target position increases and CCW motion occurs when the target position decreases.

Pulse unit

It refers to the units measured in pulses and used on the drive, which are usually represented by pulse.

The drive usually contains two parameters: the pulse zero point and the number of pulses of an encoder per revolution of the motor.

16.1.4 Axis Configuration Parameters

Attributes of motion control axes can be set as needed. The following table summarizes the axis configuration parameters.

Type	Description	Bus servo axis	Local pulse axis
Basic settings	Axis number	√	√
	Axis type	√	√
	Input device	×	×
	Output device	√	√
	Automatic mapping	√	×
	Virtual axis mode	√	✓
	Profile mode	√	×
	PDO	√	×
Unit conversion settings	Reverse direction	√	√
	The number of instruction pulses per revolution of the motor/encoder	√	√
	The distance per revolution of the worktable in the background	√	√
	Defines the numerator of the gear ratio.	√	√

Type	Description	Bus servo axis	Local pulse axis
	Defines the denominator of the gear ratio.	√	√
Mode/Parameter settings	Encoder mode	√	×
	Linear/rotary mode settings	√	√
	Software limits	√	√
	Software error response	√	√
	Tracking error	√	√
	Axis speed settings	√	√
	Torque limit	√	×
	Probe settings	×	√
	Output settings	×	√
	Hardware limit logic	√	×
	Not entering ErrorStop state upon a limit activation	√	√
Homing settings	Home signal	√	√
	Positive limit	√	√
	Negative limit	√	√
	Z signal	√	×
	Homing direction	√	√
	Home input detection direction	√	√
	Homing list	√	√
	Homing velocity	√	√
	Homing approaching speed	√	√
	Homing acceleration	√	√
	Homing timeout time	√[1]	√
	Negative limit terminal settings	×	√
	Positive limit terminal settings	×	√

Type	Description	Bus servo axis	Local pulse axis
	Home signal settings	×	√
Online commissioning	Monitoring list	√	√
	Motion commissioning	√	√

NOTE

[1]: This function is only available to Inovance servo drives.

16.1.5 Axis System Variables

In the program, you can monitor the current status of an axis through its system variables. The system variables of a bus servo axis/local pulse axis are shown in the following table.

Name	Data type	Function
bPowerState	BOOL	Monitoring parameter, enabled or disabled status of the axis, read-only
bDebugState	BOOL	Monitoring parameter, commissioning status of the axis, read-only
fSetPosition	REAL	Monitoring parameter, position reference of the axis, user unit, read-only
fSetVelocity	REAL	Monitoring parameter, velocity reference of the axis (that is, the change rate of the position reference), user unit, read-only
fSet_Acc_Dec	REAL	Monitoring parameter, acceleration reference of the axis (that is, the change rate of the velocity reference), user unit, read-only
fSetTorque	REAL	Monitoring parameter, torque reference of the axis, user unit, read-only
fActPosition	REAL	Monitoring parameter, feedback position of the axis, user unit, read-only
fActVelocity	REAL	Monitoring parameter, the current velocity of the axis (that is, the change rate of the feedback position), user unit, read-only
fActAcc_Dec	REAL	Monitoring parameter, the current acceleration of the axis (that is, the change rate of the feedback velocity), user unit, read-only
fActTorque	REAL	Monitoring parameter, feedback torque of the axis, user unit, read-only
wPLCOpenState	INT	

Name	Data type	Function
		Monitoring parameter, PLCOpen state machine for the axis, read-only 0: PowerOff 1: ErrorStop 2: Stopping 3: StandStill 4: Discrete motion 5: Continuous motion 7: Homing 8: Synchronized motion
wConfigState	INT	Monitoring parameter, configuration status of the axis, read-only 0: Init (axis in the initialization state) 1: Configure finish (reading of configuration data completed) 2: Sync finish (synchronized with EtherCAT tasks) 3: Wait communication (communication with the servo drive established) 4: Slave ready (initialization completed for the servo drive controlled by axes) 5: Axis ready (communication established)
wAxisError	INT	Monitoring parameter, fault code for the axis in the ErrorStop state, read-only
wServoError	INT	Monitoring parameter, fault code for a drive or local axis, displaying 0x603f values, read-only
bEnterDebug	BOOL	Commissioning parameter, entering the commissioning mode when the variable is valid
bPowerOn	BOOL	Commissioning parameter, axis enable instruction
bStop	BOOL	Commissioning parameter, axis stop instruction
bReset	BOOL	Commissioning parameter, axis reset instruction
bJogP	BOOL	Commissioning parameter, forward jog instruction
bJogN	BOOL	Commissioning parameter, reverse jog instruction
bHome	BOOL	Commissioning parameter, homing instruction
bSetPos	BOOL	Commissioning parameter, current position setting instruction

Name	Data type	Function
bAbsPos	BOOL	Commissioning parameter, absolute positioning instruction
bRevPos	BOOL	Commissioning parameter, relative positioning instruction
bRelPos	BOOL	Commissioning parameter, reciprocating motion instruction
bVelocity	BOOL	Commissioning parameter, continuous motion instruction
bTorque	BOOL	Commissioning parameter, torque instruction
wDebugMotionType	INT	Commissioning parameter, commissioning mode 0: Idle 1: Relative positioning control 2: Absolute positioning control 3: Continuous motion control 5: Reciprocating motion control 6: Torque control
fJogVelocity	REAL	Commissioning parameter, jogging velocity
fPositionOffser	REAL	Commissioning parameter, homing offset
fPresetPosition	REAL	Commissioning parameter, preset positions
fTarPosition1	REAL	Commissioning parameter, target position
fTarVelocity1	REAL	Commissioning parameter, target velocity
fTarAcceleration1	REAL	Commissioning parameter, target acceleration
fTarDeceleration1	REAL	Commissioning parameter, target deceleration
wCurveType1	REAL	Commissioning parameter, curve type
fTarPosition2	REAL	Commissioning parameter, target position 2
fTarVelocity2	REAL	Commissioning parameter, target velocity 2
fTarAcceleration2	REAL	Commissioning parameter, target acceleration 2
fTarDeceleration2	REAL	Commissioning parameter, target deceleration 2
wCurveType2	INT	Commissioning parameter, curve type 2
dUnused		Commissioning parameter, reserved
fTarTorque	REAL	Commissioning parameter, target torque
fTarTorqueSlop	REAL	Commissioning parameter, torque slope

Name	Data type	Function
fLimitVelocity	REAL	Commissioning parameter, velocity limit in torque mode
bSingleSyncMode	BOOL	Commissioning parameter, single-axis synchronous control mode
wControlWord	INT	Loop variable, control word 0x6040, read-only
WStatusWord	INT	Loop variable, control word 0x6041, read-only
dSetPosition	DINT	Loop variable, target position 0x607a, read-only
dActPosition	DINT	Loop variable, feedback position 0x6064, read-only
dSetVelocity	DINT	Loop variable, velocity reference 0x60ff, read-only
dActVelocity	DINT	Loop variable, feedback velocity 0x606c, read-only
dSetTorque	DINT	Loop variable, torque reference 0x6071, read-only
dActTorque	DINT	Loop variable, feedback torque 0x6077, read-only
dDO	DINT	Loop variable, DO 0x60fe:1, read-only
dDI	DINT	Loop variable, DI 0x60fd, read-only
wModesOfOperation	INT	Loop variable, control mode 0x6060, read-only
wModesOfOperationDisplay	INT	Loop variable, current control mode 0x6061, read-only
wTouchFunction	INT	Loop variable, probe function settings 0x60b8, read-only
wTouchStatus	INT	Loop variable, probe status 0x60b9, read-only
dTouch1Ppos	DINT	Loop variable, probe 1 position on the rising edge 0x60ba, read-only
dTouch2Ppos	DINT	Loop variable, probe 2 position on the rising edge 0x60bb, read-only
dTouch1Npos	DINT	Loop variable, probe 1 position on the falling edge 0x60bc, read-only
dTouch2Npos	DINT	Loop variable, probe 2 position on the falling edge 0x60bd, read-only
dMaxVelocity	DINT	Loop variable, maximum velocity 0x607f, maximum velocity
wErrorCode	INT	Loop variable, drive fault code 0x603f, read-only
dProfielVelocity	DINT	Profile velocity (Read-only, PDO data, 0x6081)

Name	Data type	Function
dProfielAcceleration	DINT	Profile acceleration (Read-only, PDO data, 0x6083)
dProfielDecelerationi on	DINT	Profile deceleration (Read-only, PDO data, 0x6084)
wAxisRingPos	INT	Configuration parameter, axis configuration position, power-on initialization
WAxisID	INT	Configuration parameter, axis ID, power-on initialization
fUnits	REAL	Configuration parameter, axis gear ratio, power-on initialization
fFilter	REAL[3]	Setting the filter coefficient of the master axis, initialized to 1.0,.0.0,0.0 upon power-on, and modifiable
bMotionState	BOOL	Monitoring parameter, motion status, indicating whether the axis is in motion, read-only
bphlimit	BOOL	Monitoring parameter, hardware positive limit input status, read-only
bnhlimit	BOOL	Monitoring parameter, hardware negative limit input status, read-only
bhomestate	BOOL	Monitoring parameter, hardware home switch input status, read-only
bpslimit	BOOL	Monitoring parameter, indicating whether the software positive limit is reached, read-only
bnslimit	BOOL	Monitoring parameter, indicating whether the software negative limit is reached, read-only
bTardir	BOOL	Axis target motion direction, read-only OFF: Forward ON: Negative
dLocialAxisSetPos	DINT	Monitoring parameter, position reference of the local pulse axis, read-only
fFollowPos	REAL	Following error, read-only

16.1.6 List of Axis Control Instructions

The following table lists single-axis control instructions. For details about how to use these instructions, see H5U and Easy Series Programmable Logic Controllers Instructions Guide.

Instruction	Name
MC_Power	Enable control
MC_Reset	Fault reset
MC_ReadStatus	Axis state read
MC_ReadAxisError	Axis error read
MC_ReadDigitalInput	Digital input read
MC_ReadActualPosition	Current position read
MC_ReadActualVelocity	Current velocity read
MC_ReadActualTorque	Current torque read
MC_SetPosition	Current position settings
MC_SetAxisConfigPara	Instruction for setting axis parameters
MC_TouchProbe	Probe
MC_MoveRelative	Relative positioning reference
MC_MoveAbsolute	Absolute positioning reference
MC_MoveVelocity	Speed reference
MC_Jog	Jogging
MC_JogP	Profile-based jog control
MC_TorqueControl	Torque control instruction
MC_Home	ZRN
MC_Stop	Stop
MC_Halt	Halt
MC_ImmediateStop	Immediate stop
MC_MoveAbsRelPP	Positioning based on profile position mode
MC_MoveVelocityPV	Profile velocity instruction
MC_MoveFeed	Interrupt positioning
MC_MoveBuffer	Multi-position positioning
MC_MoveSuperImposed	Motion superimposition

Instruction	Name
MC_MoveVelocityCSV	CSV-based velocity control with adjustable pulse width
MC_SyncMoveVelocity	CSV-based synchronous velocity control with adjustable pulse width
MC_SyncTorqueControl	Synchronous torque control

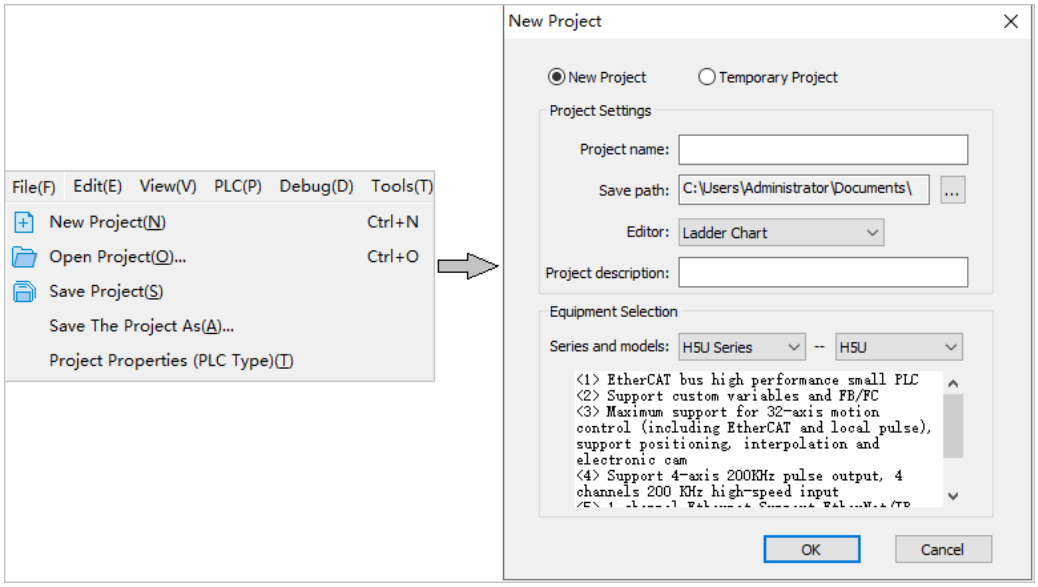
16.2 Motion Control Axis Settings

16.2.1 Creating a Project

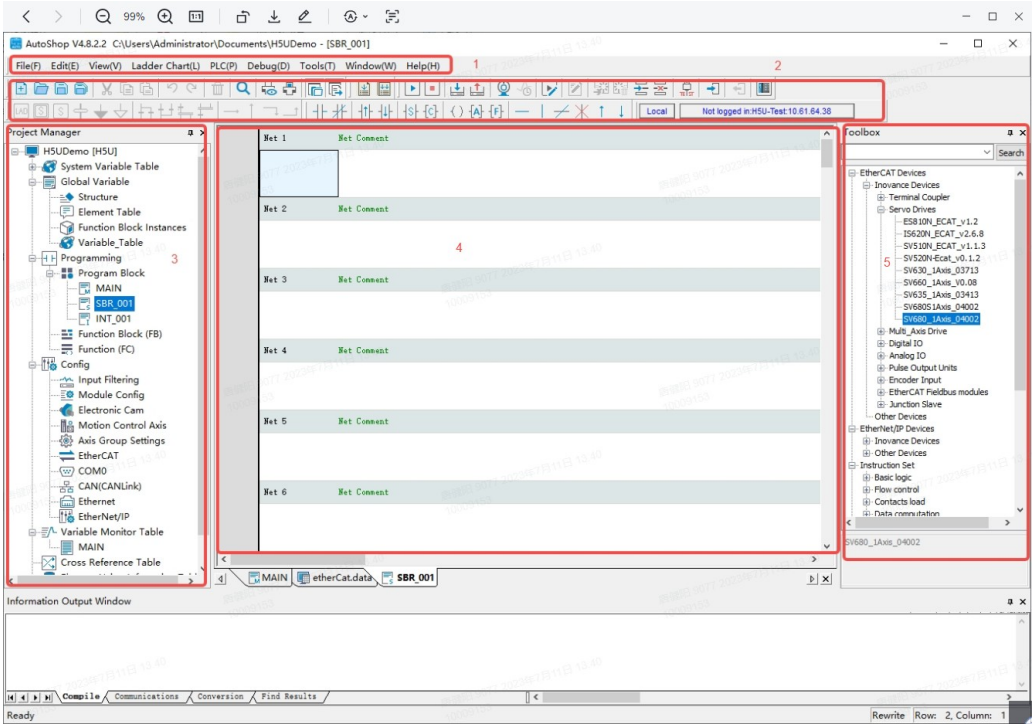
Follow these steps to accurately control an axis: First, create a configuration based on the needs of your project. Then, set relevant parameters based on the working conditions, download the project, and carry out simple operations through online commissioning to determine whether the parameter settings are reasonable and whether the hardware connection is reliable. Lastly, write a PLC program to complete the overall control logic. Here is an example.

This routine creates a new bus servo axis and a local pulse axis, and implements simple motion through two ways: online commissioning interface and instructions.

1. Open the Autoshop and create a project. Select “H5U” in the “PLC type” pane.



2. Access the main interface, as shown in the following figure.



No.	Name
1	Menu bar
2	Toolbar
3	Project manager
4	Program editing section
5	Tool kit

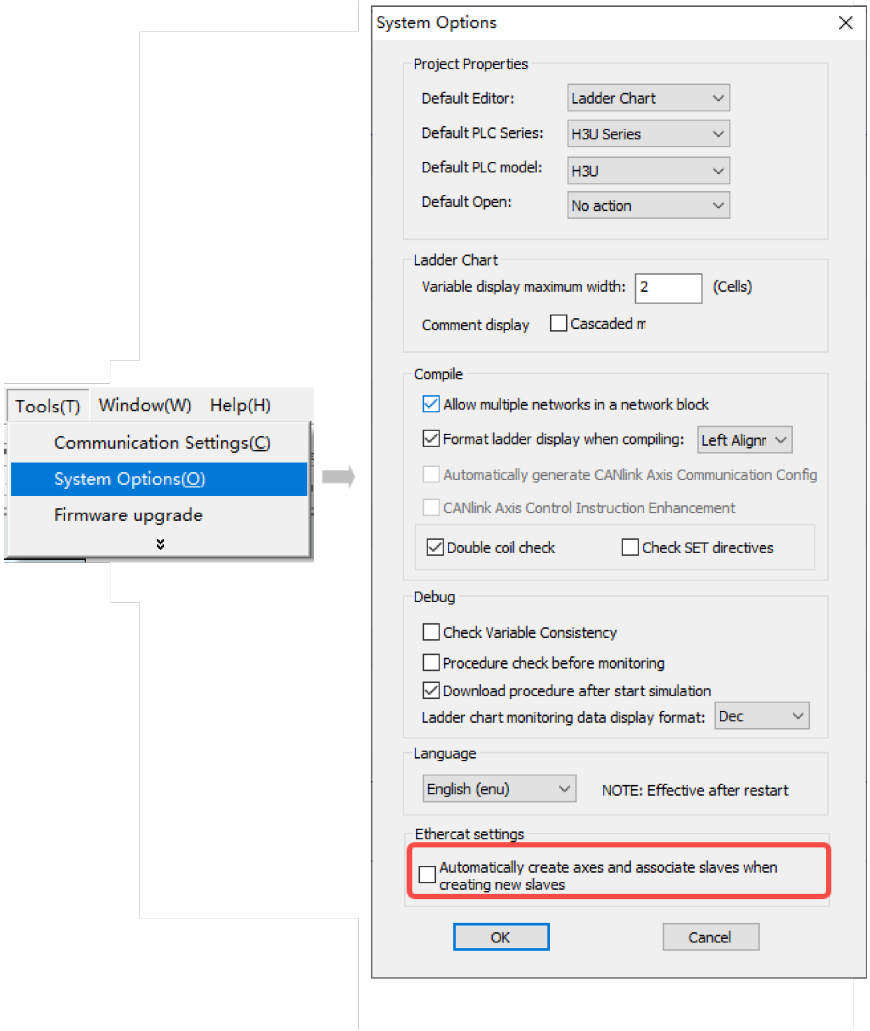
16.2.2 Project Configuration

To control the IS620N motion, you can configure a servo drive and a bus servo axis and link them together. You can add the connection through automatic scan or manual addition.

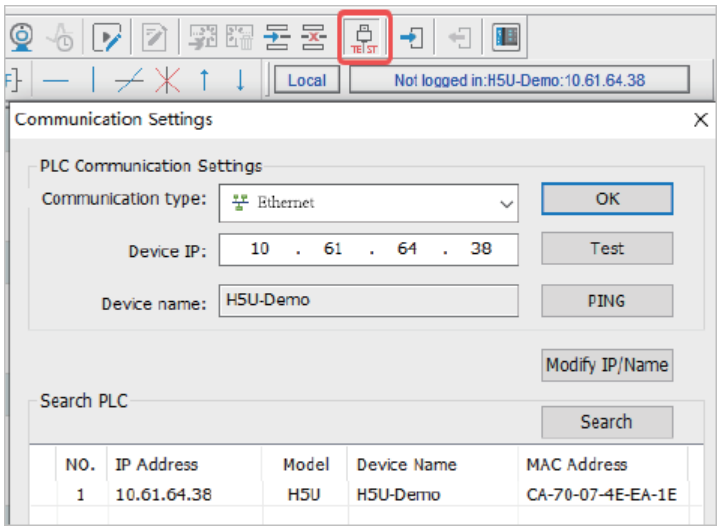
In the automatic scan mode, bus servo axes are automatically added, while local pulse axes need to be added manually.

Automatic scan

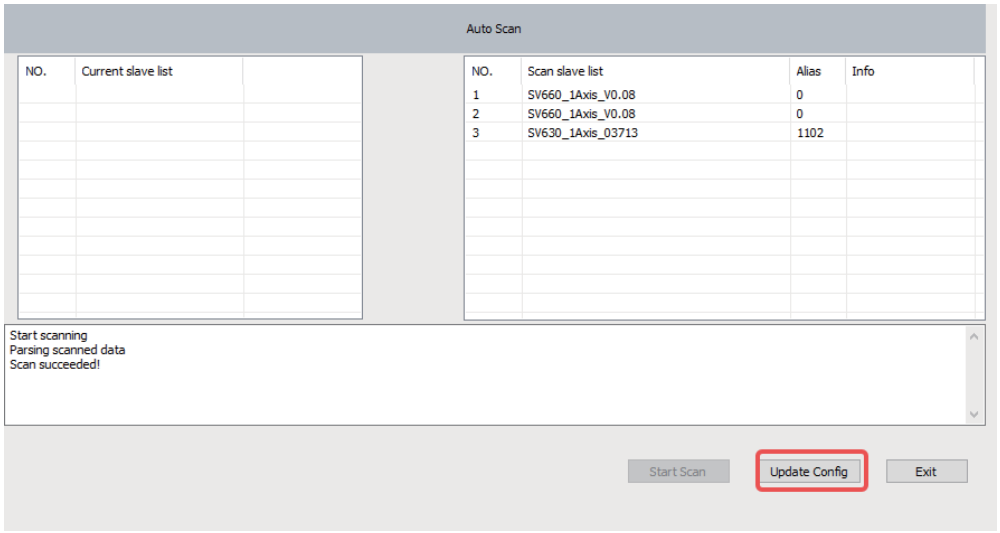
1. In "System Options", check whether the option "Automatically create axes and associate slaves when creating new slaves" is selected. If not, select it.



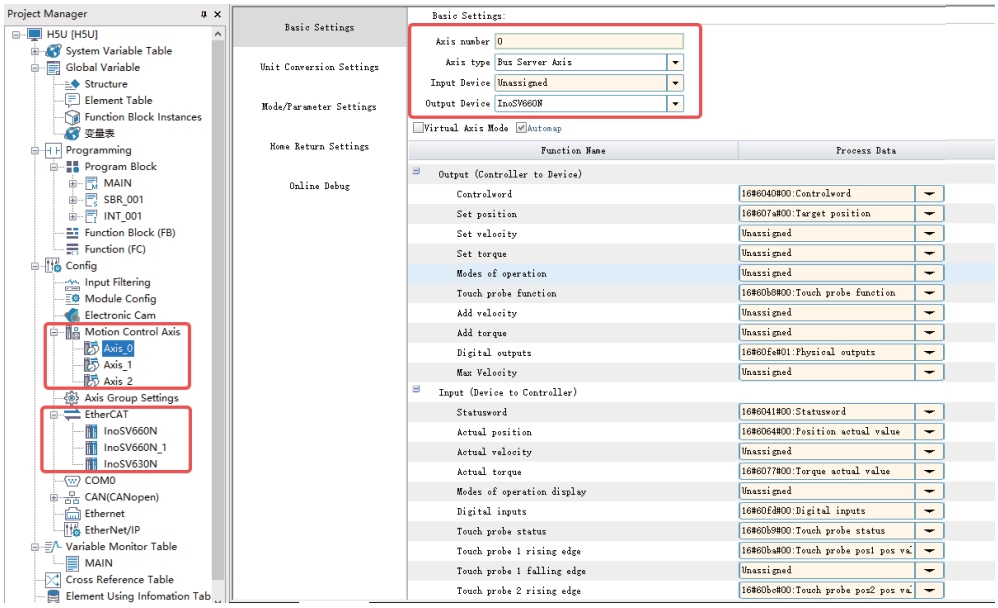
2. Check whether the computer host is normally connected to a PLC and whether the EtherCAT network port of the PLC is normally connected to a servo drive.
   The way to test the connection of PLC to the computer host is as follows.



3. In the toolbox, check whether the EtherCAT device list includes the IS620N. If not, add the corresponding XML file.
4. Select the master station, right-click it, select "Auto Scan", and the "Auto Scan" dialog box will pop up.
5. Click "Start Scan". After the scan is completed, click "Update Config" to complete the creation of a bus servo axis, as shown in the following figure.

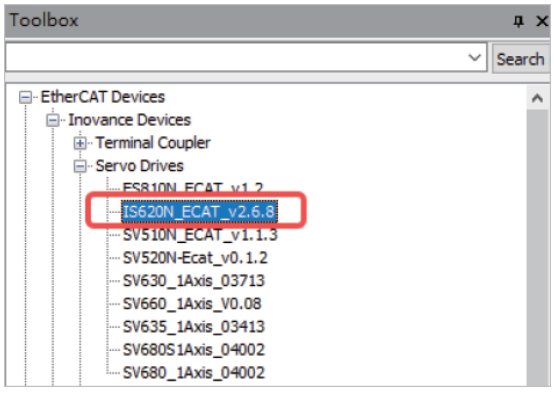


6. After the scan is completed, you can see the servo drive and bus servo axis in the device tree, as shown in the following figure.



Adding a device manually

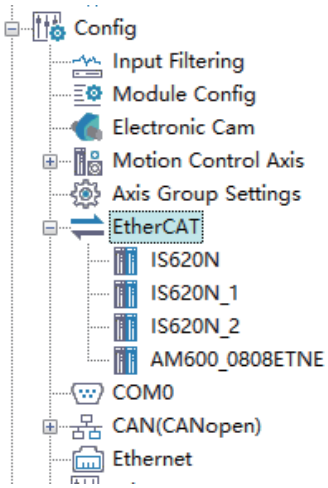
1. Open the toolbox and locate the IS620N. If it is unavailable, you can add the device profile ESI for IS620N to the toolbox by importing the file, as shown in the following figure.



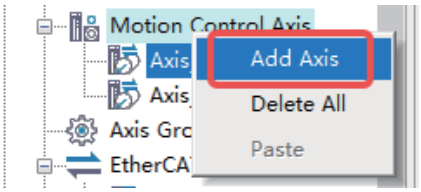
2. Double-click the IS620N in the toolbox to add an IS620N to the device tree EtherCAT configuration, as shown in the following figure.

   If you have ticked "Automatically create axes and associate slaves when creating new slaves" in the "System Options", a bus servo axis will be added when you add the IS620N. Otherwise, the bus servo axis

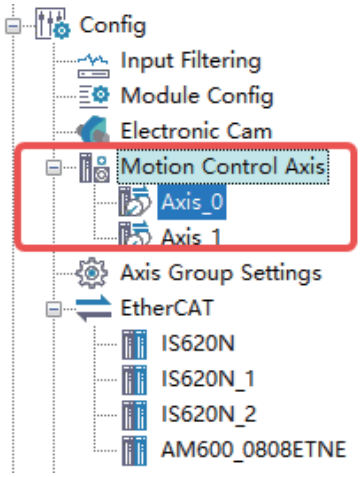
will not be added automatically. The assumption in this routine is that "Automatically create axes and associate slaves when creating new slaves" is not ticked.



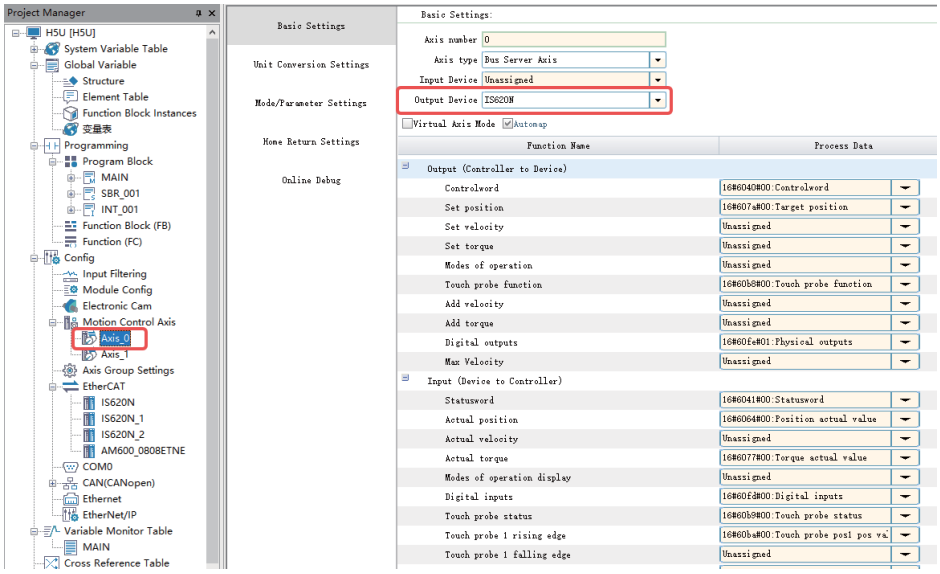
3. You can add a motion control axis by choosing "Motion Control Axis" in the device tree, and right-clicking the "Add Axis". You can establish two axes by repeating this operation twice, as shown in the following figure.



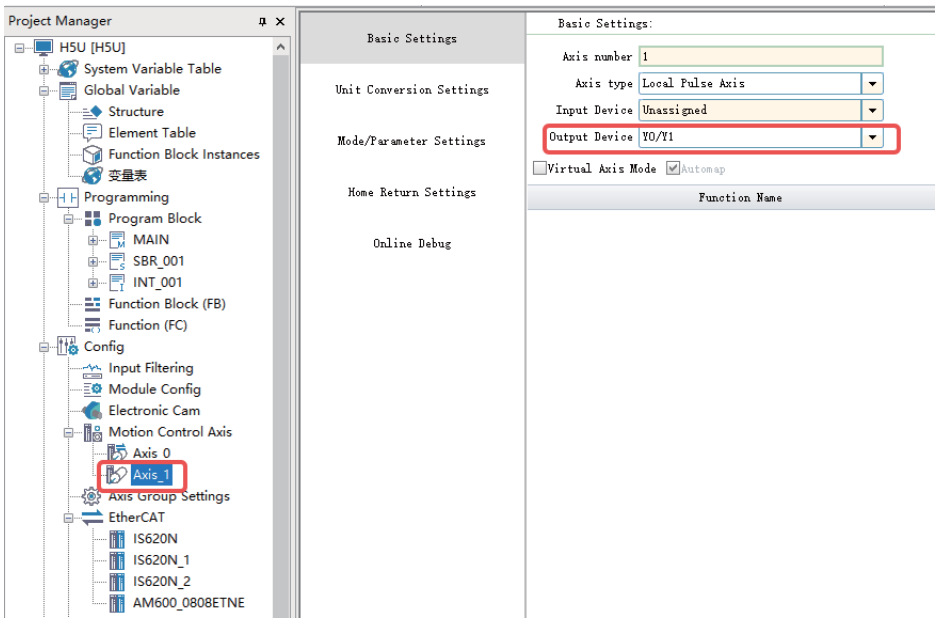
The following figure shows the configuration after adding two axes.



4. Set the first axis as the bus servo axis and associate it with IS620N. Set the second axis to be the local pulse axis and associate it with the Y0/Y1 channel.
   The following figure shows the effect of adding the bus servo drive.



The following figure shows a local output axis.



Profile mode

> **CAUTION**
>
> This mode is supported in AutoShop of V4.10.0.0 and later versions and Easy firmware of V6.2.0.0 and later versions.

The profile mode is mainly applicable for the point-to-point positioning application. AutoShop defines the target position, operating speed, and acceleration/deceleration rate. The position profile generator inside the servo drive generates the position profile based on settings, and the servo drive executes position control, speed control, and torque control.

The following describes the profile mode.

- Easy series models supported: Easy501, Easy502, Easy521, Easy522, and Easy523
- Servo drive models supported: SV630N, SV635N, SV660N, SV670N, SV680N, IS810N, SV660ND, SV630ND, and STD60N

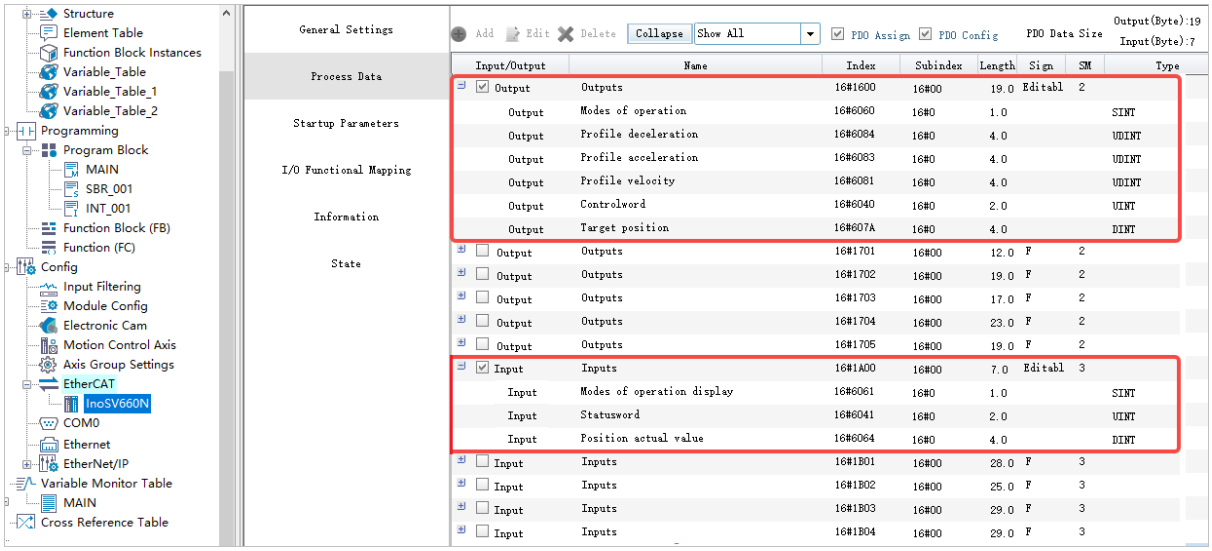
- Instructions supported: MC_Power, MC_Reset, MC_ReadStatus, MC_ReadAxisError, MC_ReadDigitalInput, MC_ReadActualPosition, MC_ReadActualVelocity, MC_ReadActualTorque, MC_SetPosition, MC_Home, MC_Stop, MC_Halt, MC_ImmediateStop, MC_SetAxisConfigPara, MC_MoveAbsRelPP, MC_MoveVelocityPV, and MC_JogP
- Number of supported profile axes: Up to 32 for Easy523, up to 16 for Easy502 and Easy522, and up to 8 for Easy501 and Easy521
- Profile position mode: AutoShop sets parameters through the bus, and the servo drive generates position references and performs positioning control.
- Profile velocity mode: AutoShop sets parameters through the bus, and the servo drive generates speed references and performs speed control.

The following describes the profile mode with the MC_MoveAbsRelPP instruction as an example.

1. Add a bus servo axis (set "Axis type" to "Bus Servo Axis"), ad select "Profile Mode" to enable the profile mode.

> **CAUTION**
>
> - This mode cannot be enabled together with the virtual axis mode.
> - In this mode, the jogging, positioning, and continuous motion functions on the "Online Commissioning" page are not available.

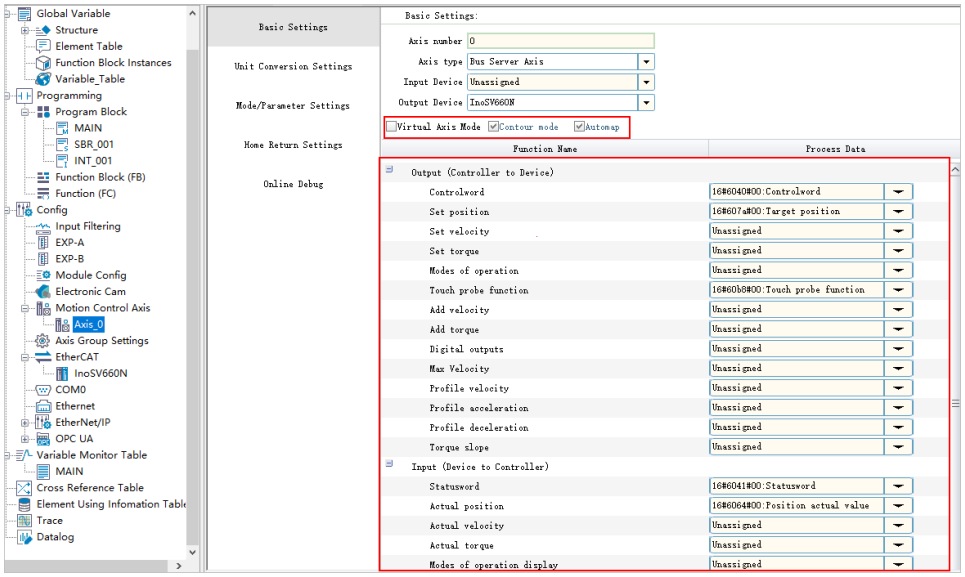
2. In the device tree on the left, double-click "InoSV660N". On the page displayed, click "Process Data" and configure the PDO parameters, as shown in the following figure. The following table lists the definitions of indexes.



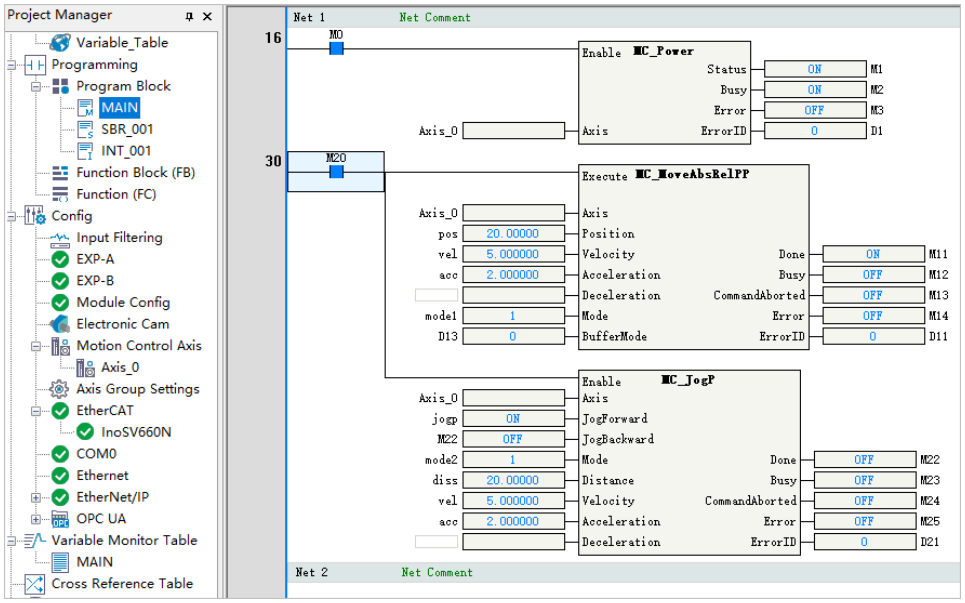
RPDO (Output)	TPDO (Input)
6040h: Control word	6041h: Status word
607Ah: Target position	6064h: Position feedback
6081h: Profile velocity	-
6083h: Profile acceleration	-
6084h: Profile deceleration	-

RPDO (Output)	TPDO (Input)
6060h: Mode selection	6061h: Operation mode display

3. Double-click "Axis_0". On the "Basic Settings" page, select "Automap", and the input/output process data is automatically configured.



4. Compile the program and set parameter values.



NOTE

The MC_MoveAbsRelPP instruction implements the absolute or relative positioning function in profile position mode for the EtherCAT bus axis. This instruction can be called only when the profile mode option is selected in the basic axis configuration. It is active on the rising edge. Other instructions can be executed only after the MC_Power instruction is executed to enable the axis.

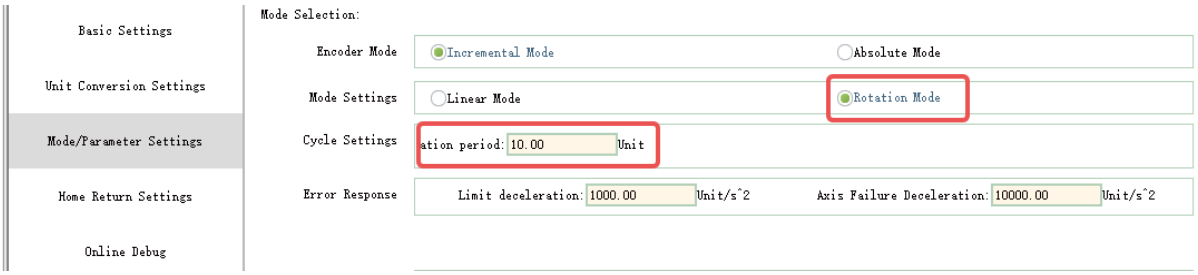
5. Then, the servo runs based on the parameter settings, and "_80rn" is displayed on the panel of the servo drive (for details, see the SV660N series servo guides), indicating that the servo drive is running normally.

16.2.3 Setting Axis Parameters

16.2.3.1 Bus Servo Axis

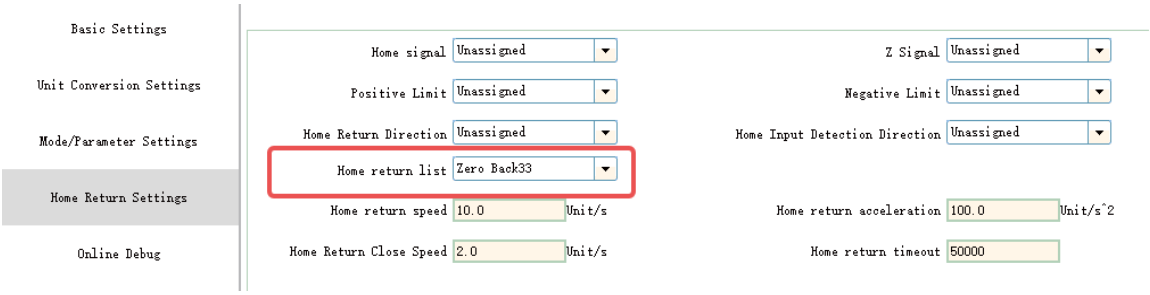
You can set the relevant parameters of an axis based on actual working conditions and requirements. The settings in this routine are as follows.

1. Set the mode to rotary and the rotation cycle to 10.



> **NOTE**
>
> The absolute rotary mode is not supported in the H5U or EasyX series PLCs with a servo.

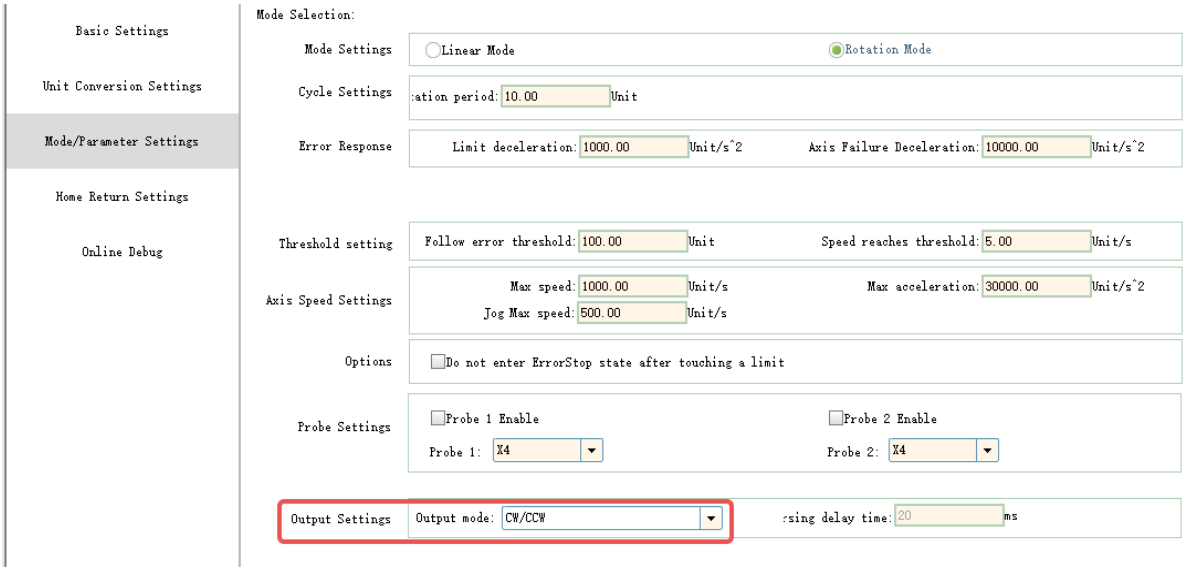
2. Set the homing mode to 33.



16.2.3.2 Local Pulse Axis

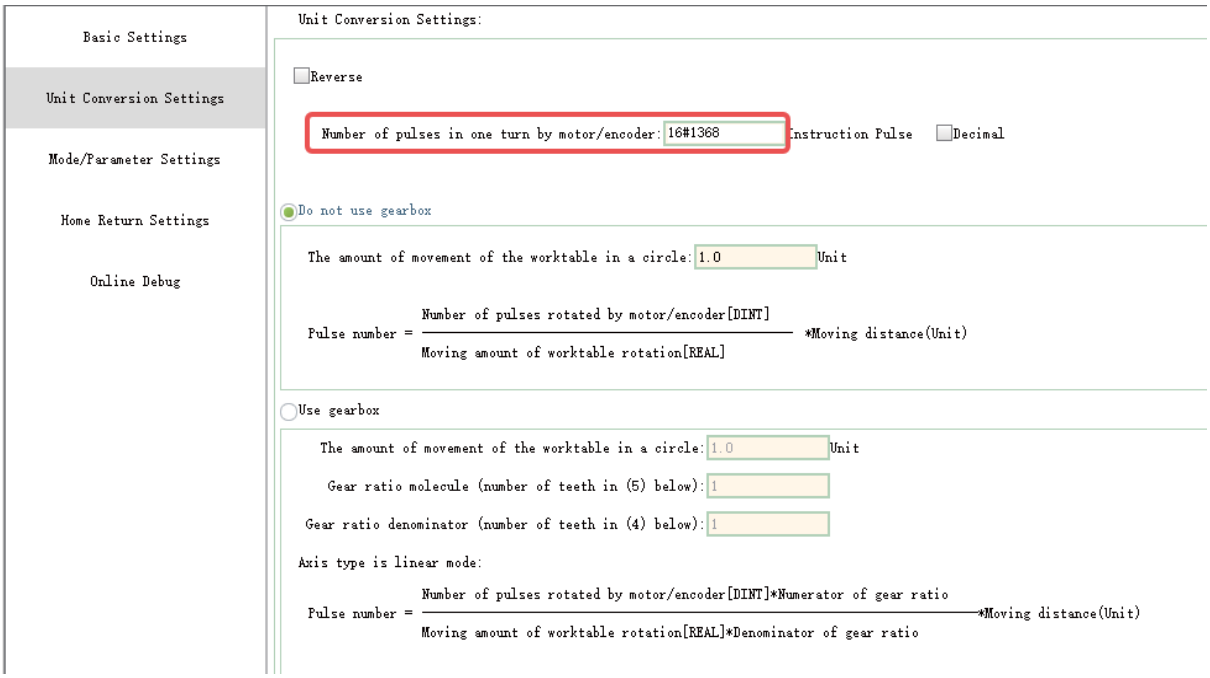
You can set the relevant parameters of an axis based on actual working conditions and requirements. The settings in this routine are as follows.

1. Set the pulse output mode to CW/CCW.

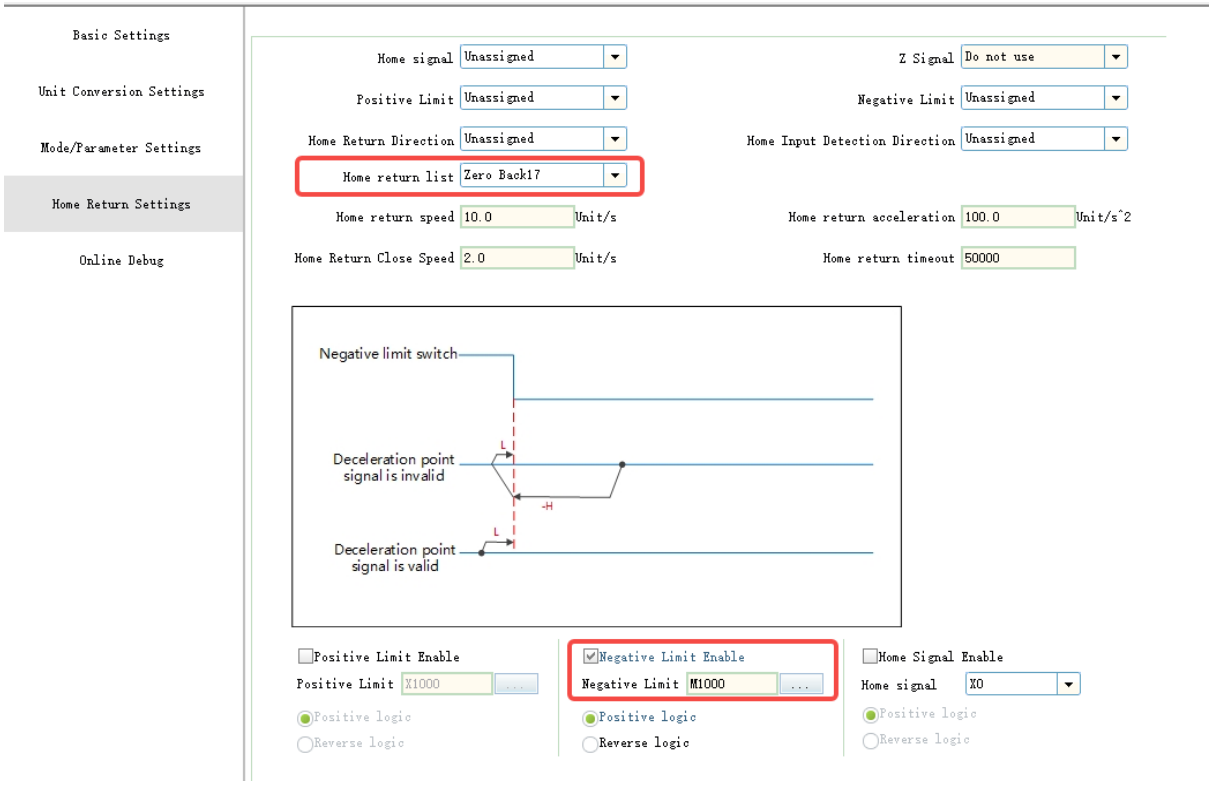


> **NOTE**
>
> The absolute rotary mode is not supported in the H5U or EasyX series PLCs with a servo.

2. Set the "Number of pulses in one turn by motor/encoder" to 5000 (16#1388).

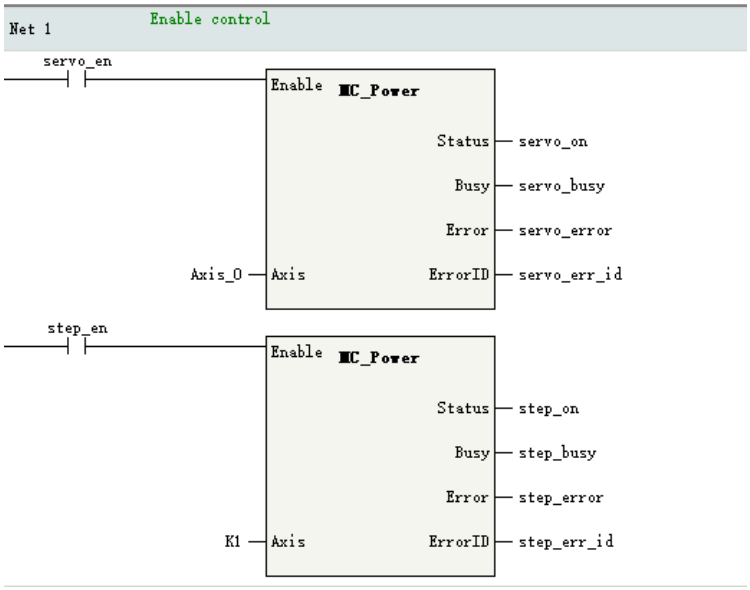


3. Set the homing mode to 17 and the negative limit to M1000.

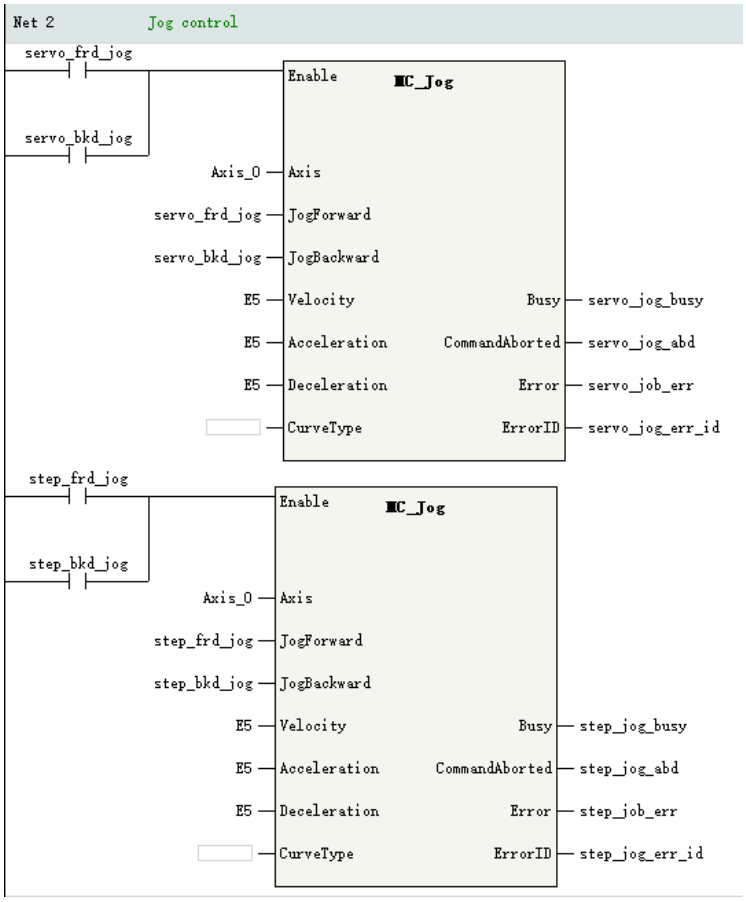


16.2.4 Programming

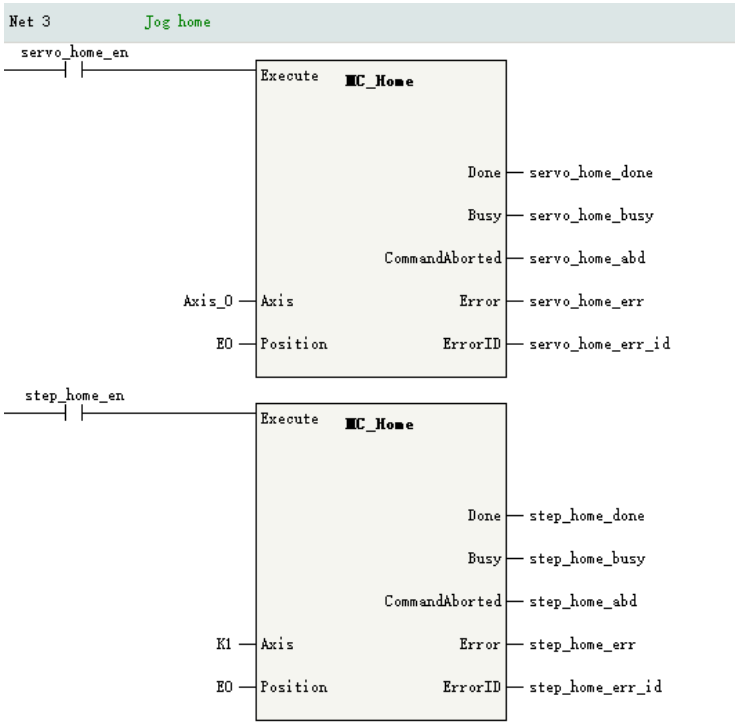
1. The function block MC_Power is used to control the enabling of an axis. MC_Power is an instruction and therefore does not need to be instantiated. This also applies to the following instructions.



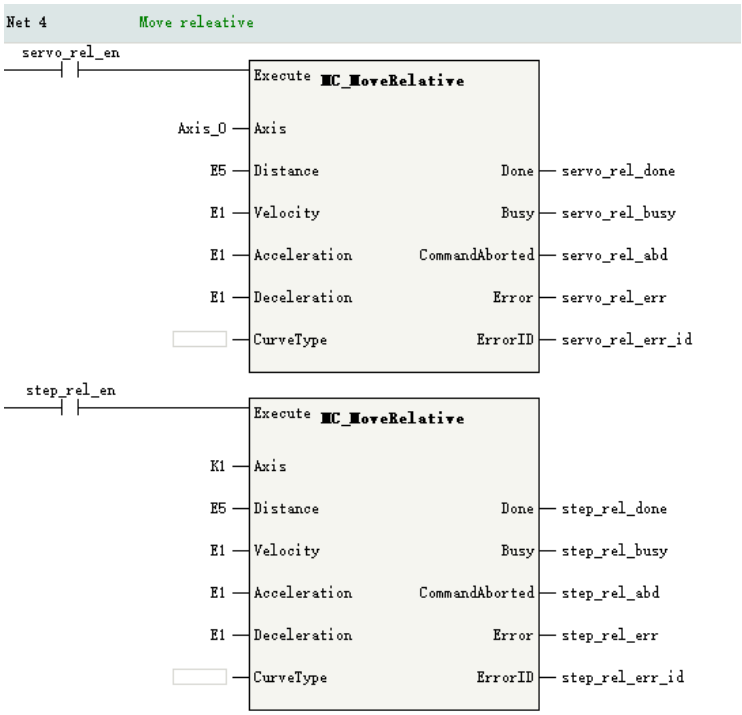
2. Call instruction MC_Jog for a test.



3. Call instruction MC_Home for a test.

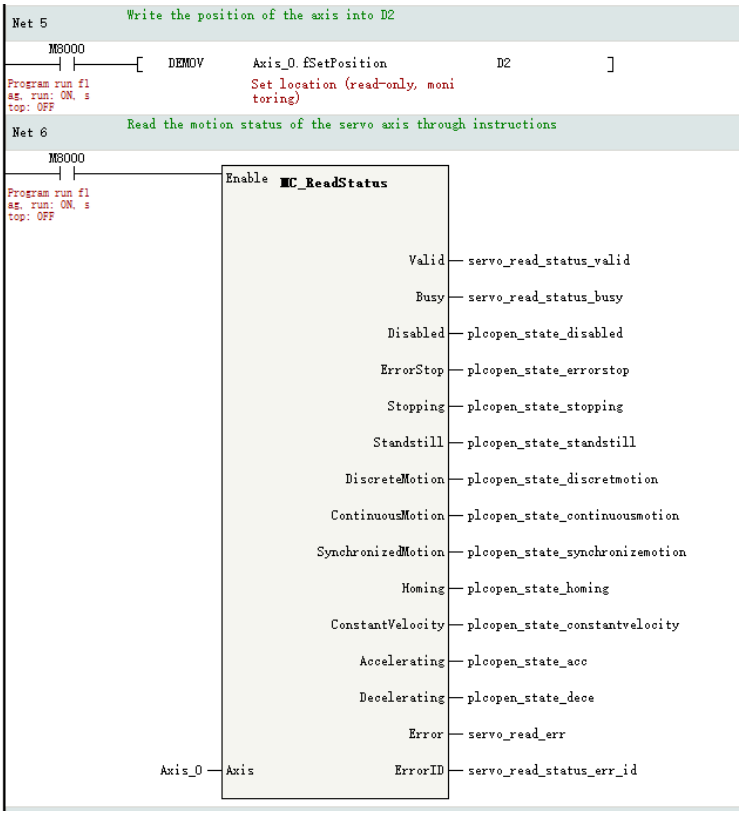


4. Call instruction MC_MoveRelative to test discrete motion.



5. Monitoring axis status

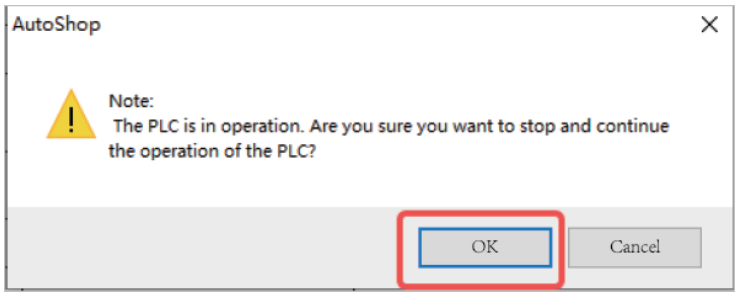
   In the PLC program, the status of the axes can be monitored through function blocks or axis system variables.



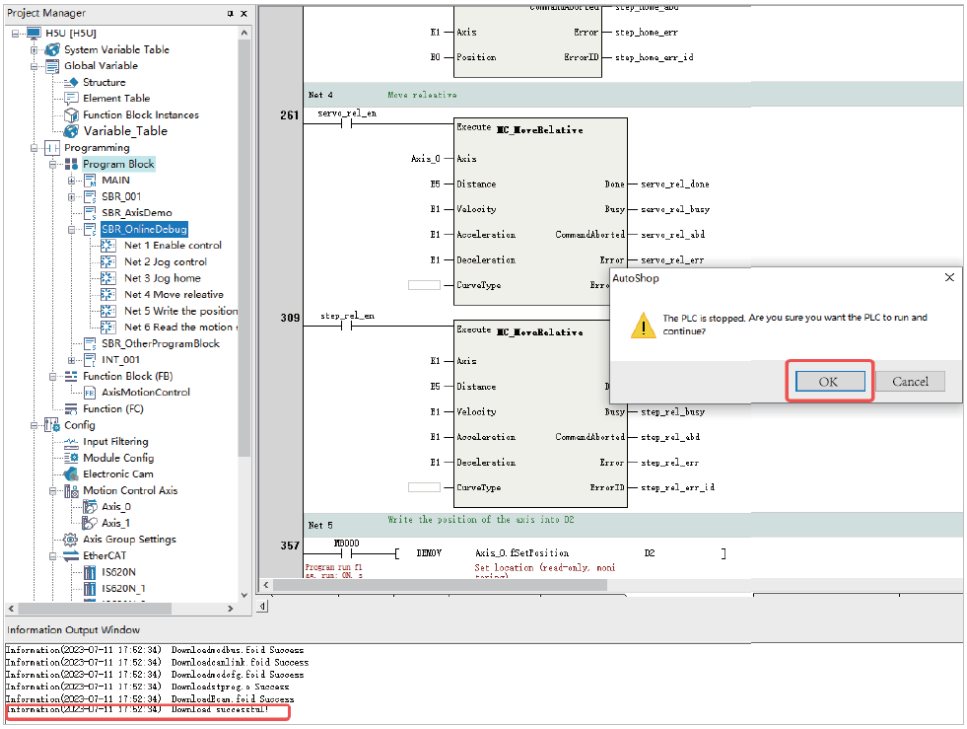
16.2.5 Downloading the Project

After completing the programming and project setup, perform the download operation as follows.

1. Click the download button , in which case the compilation operation is performed first.
2. After the compilation is completed, if the PLC is in the running state, the following dialog box will pop up. Select "OK" to go to step 3. If the PLC is in the stopped state, download the project directly and go to step 4.



3. If the PLC is in the running state before you download the program, you can see the download completion prompt after the download is completed. In the dialog box that pops up, select "OK" to switch the PLC to the running state.



4. If the PLC is in a stopped state before you download the program, the information output window will pop up the download completion prompt after the download is completed. Manually click the start button to switch the PLC to the running state.

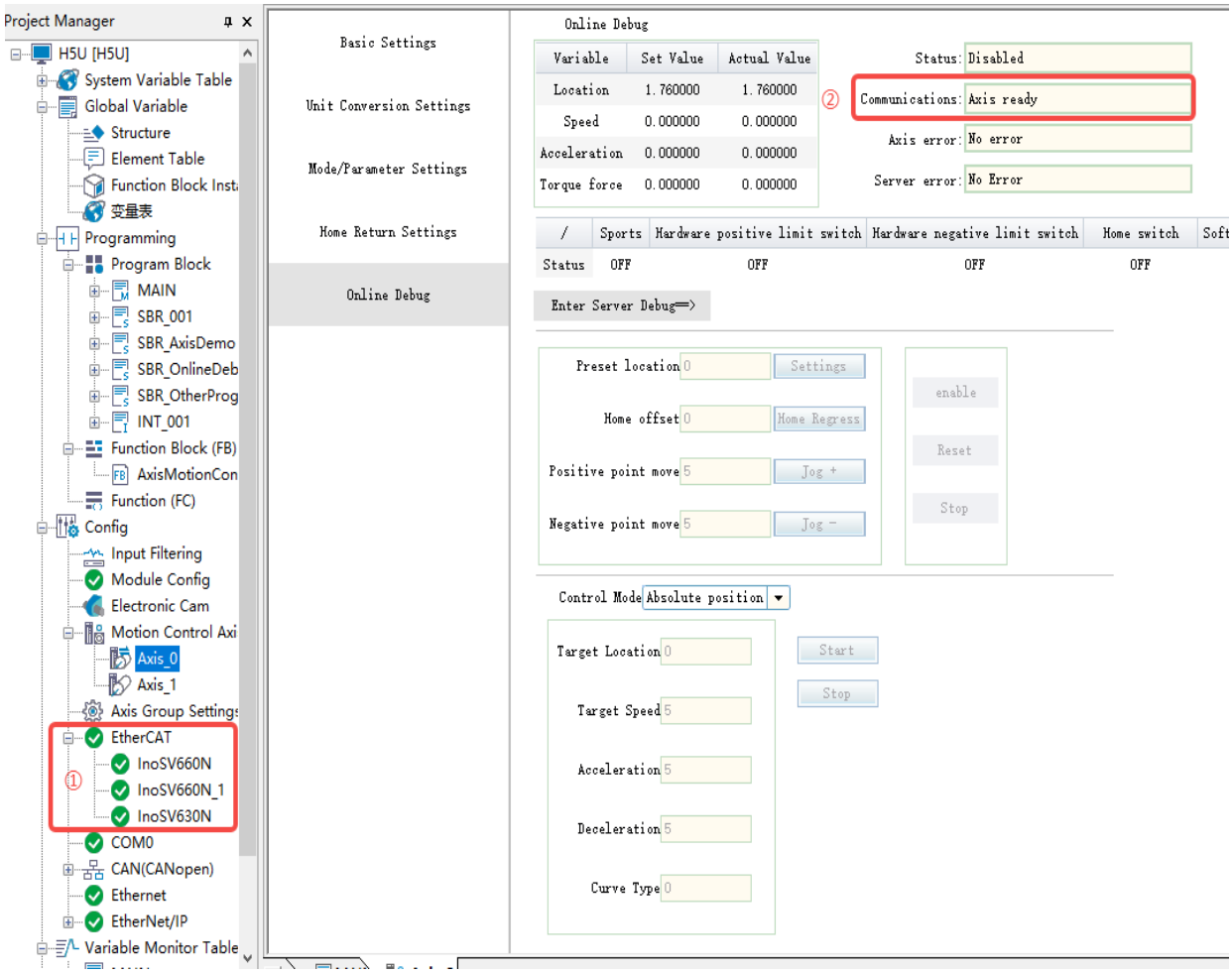
```
Information(2023-07-11 17:52:34)  Downloadmodbus.foid Success
Information(2023-07-11 17:52:34)  Downloadcanlink.foid Success
Information(2023-07-11 17:52:34)  Downloadmodcfg.foid Success
Information(2023-07-11 17:52:34)  Downloadstprog.o Success
Information(2023-07-11 17:52:34)  DownloadEcam.foid Success
Information(2023-07-11 17:52:34)  Download successful!
```

16.2.6 Basic Motion

16.2.6.1 Preconditions

To complete the basic action in this routine, enter the monitoring mode, and click the monitoring button to enter the monitoring mode.

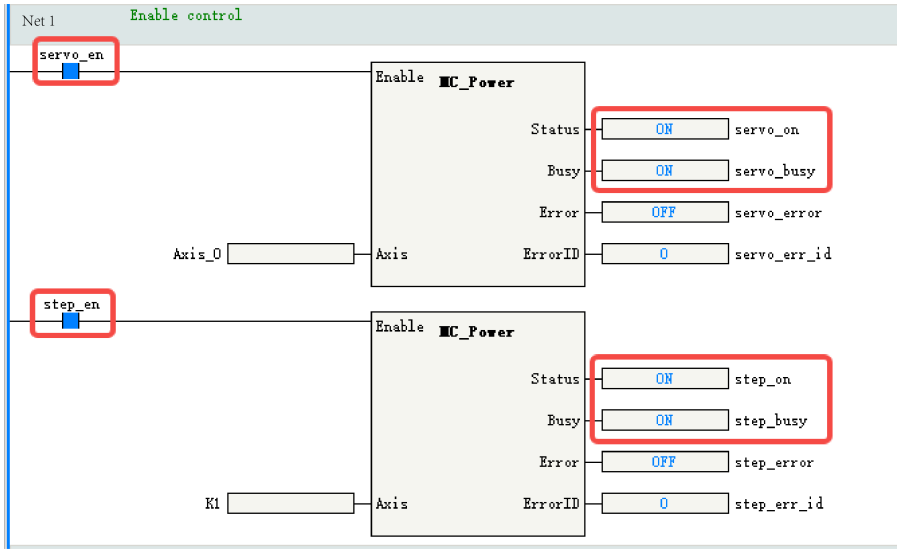
After entering the monitoring mode, you can see that the EtherCAT bus startup is completed and the servo axis initialization is completed.



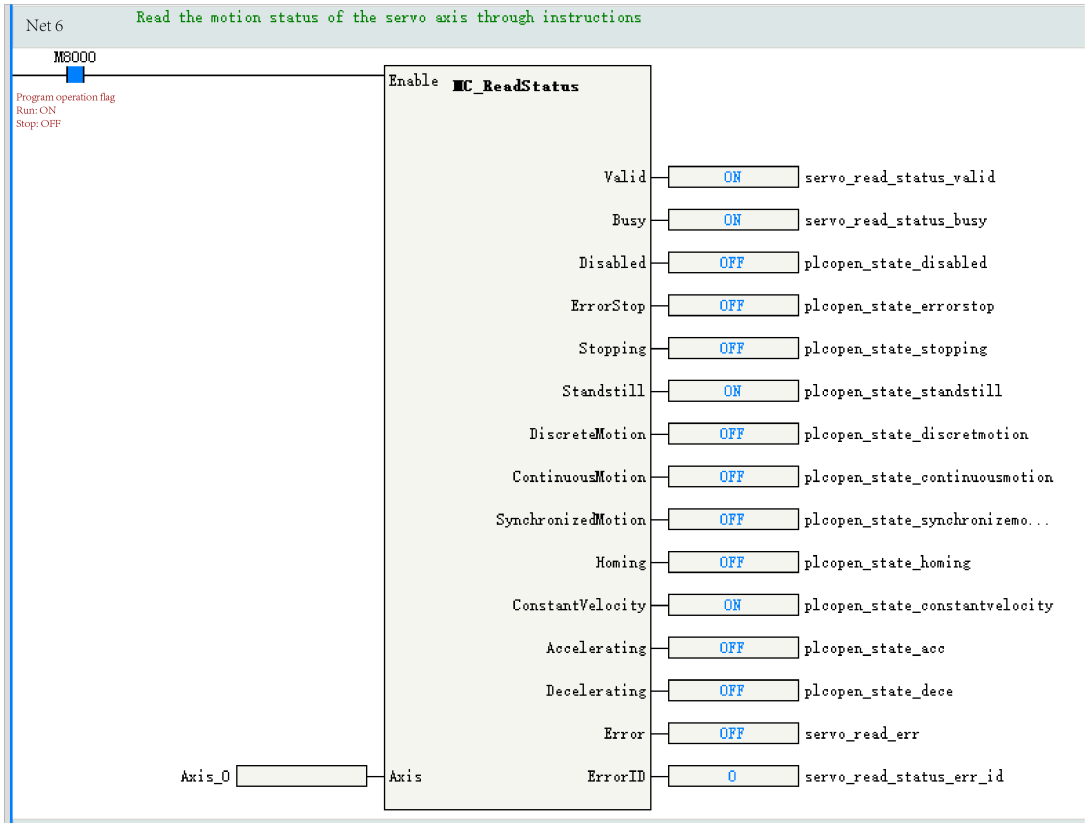
No.	Name
①	EtherCAT initialization completed
②	Axis initialization completed

16.2.6.2 PLC Program Control

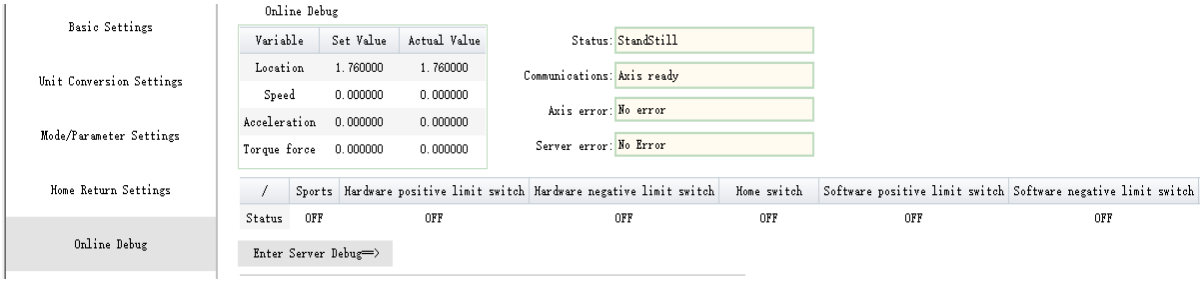
1. Enable control: Set the two BOOL variables of servo_en and step_en to TRUE, and the output of servo_on and step_on will be valid after the bus servo axis and local pulse axis are enabled.



Call the MC_ReadStatus instruction to view the status of the bus servo axis.

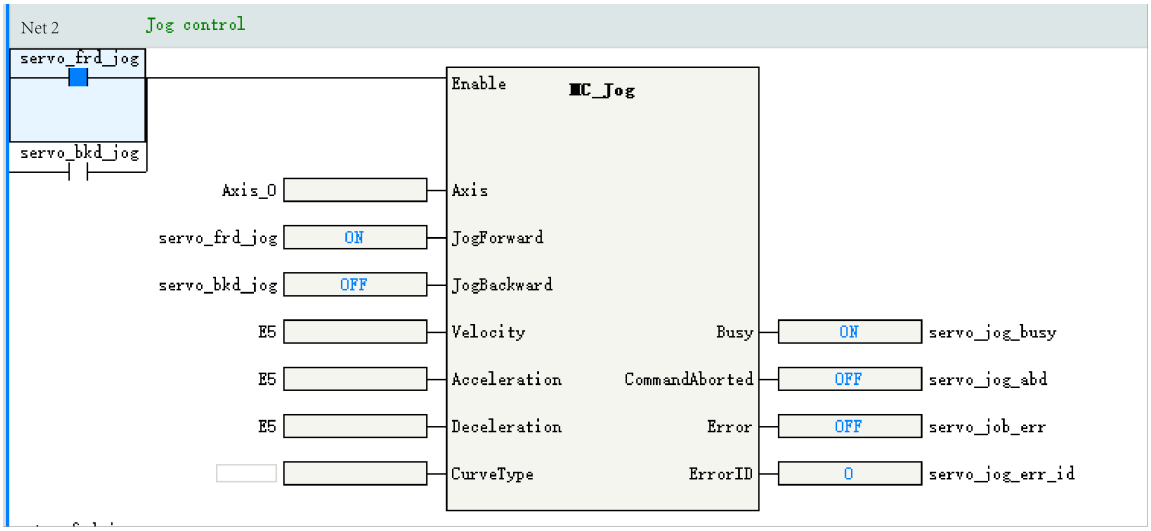


Alternatively, access the online commissioning page to view the axis status, which is standstill in this example.



2. Jogging operation of the bus servo axis

- When the variable servo_frd_jog is set to TRUE, the bus servo axis starts to run forward at the speed reference, and the actual axis driven by the servo drive runs forward at 5 rpm/s.

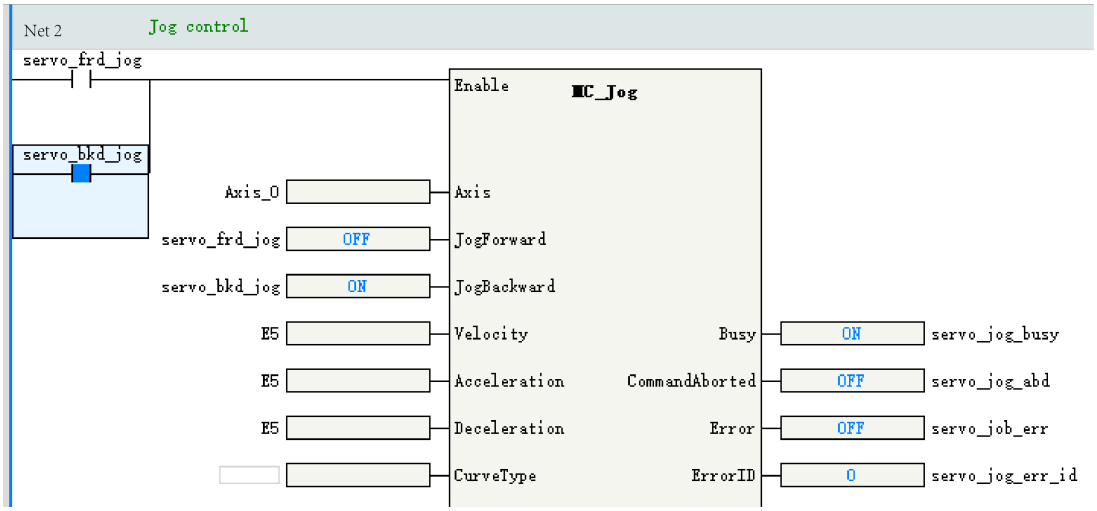


Online Debug

Variable	Set Value	Actual Value
Location	5.120000	5.099999
Speed	5.000000	5.000000
Acceleration	0.000000	0.000000
Torque force	0.000000	0.000000

Status: Continuous Motion
Communications: Axis ready
Axis error: No error
Server error: No Error

- When the variable servo_bkd_jog is set to TRUE, the bus servo axis starts to run in reverse at the speed reference, and the actual axis driven by the servo drive runs in reverse at 5 rpm/s.

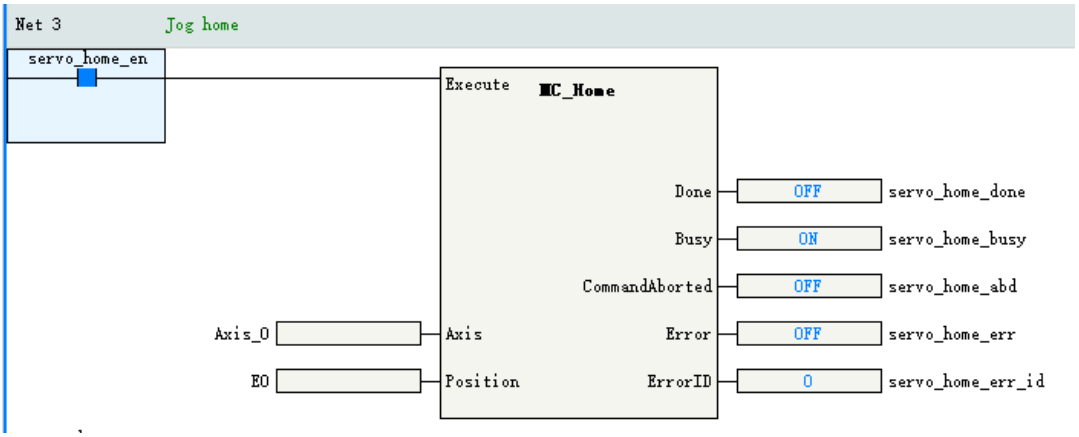


Online Debug

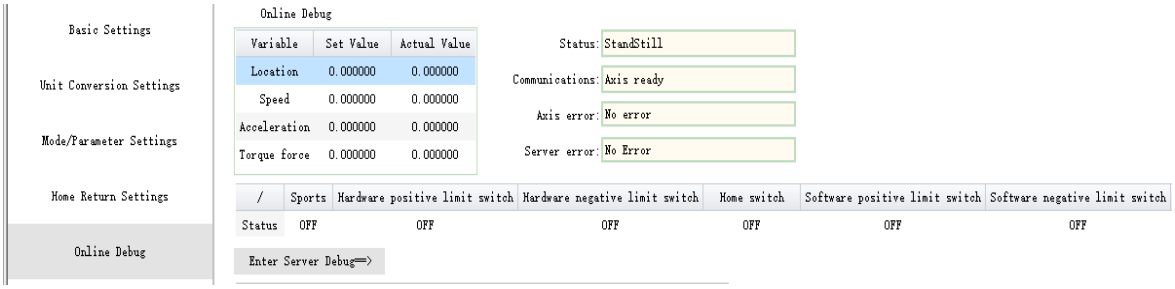
Variable	Set Value	Actual Value
Location	5.980000	6.000000
Speed	-5.000000	-5.000000
Acceleration	0.000000	0.000000
Torque force	0.000000	0.000000

Status: Continuous Motion
Communications: Axis ready
Axis error: No error
Server error: No Error

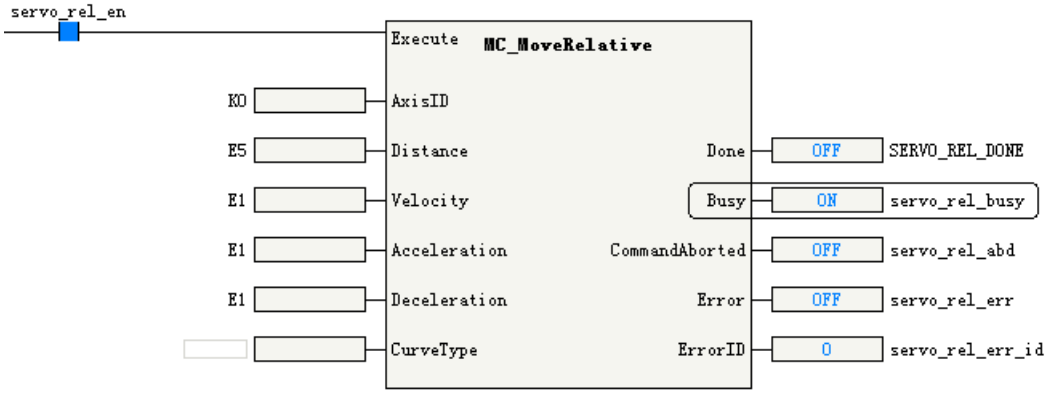
3. Homing test of the bus servo axis
   - When the servo_home_en is set to ON, the servo_home_busy will automatically set to ON.
   - When the servo motor encounters the Z signal, the homing is automatically completed, in which case the servo_home_done is set to ON, and the servo_home_busy is automatically set to OFF.



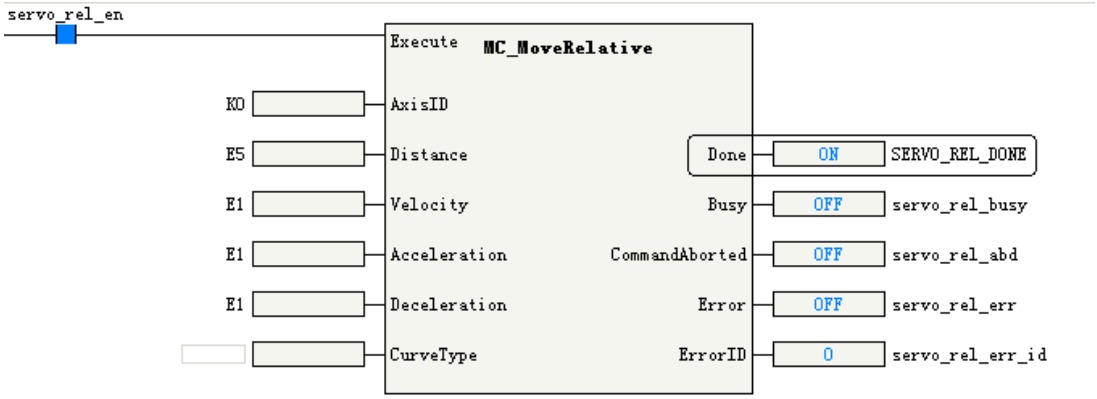
4. Relative positioning test of the bus servo axis
   - The current position of the bus servo axis is as follows:



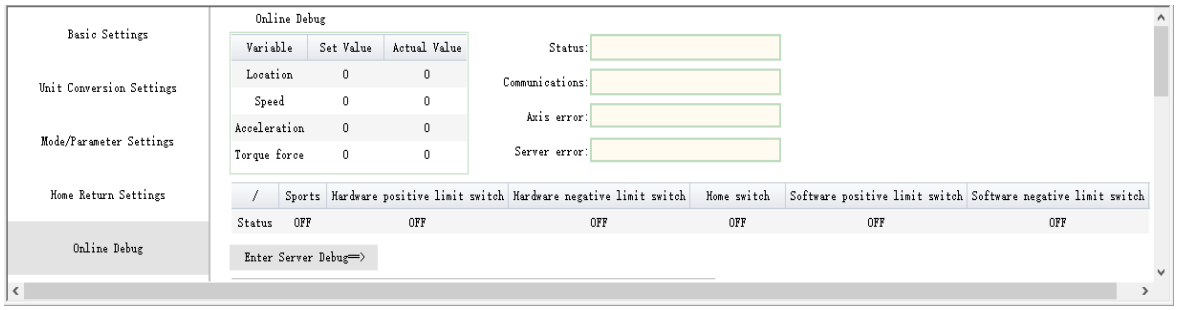
- When the variable servo_rel_en is set to TRUE, the output of variable servo_rel_busy by the function block is TRUE, in which case the bus servo axis starts to run.



- After the positioning is completed, the output of variable servo_rel_done is TRUE.

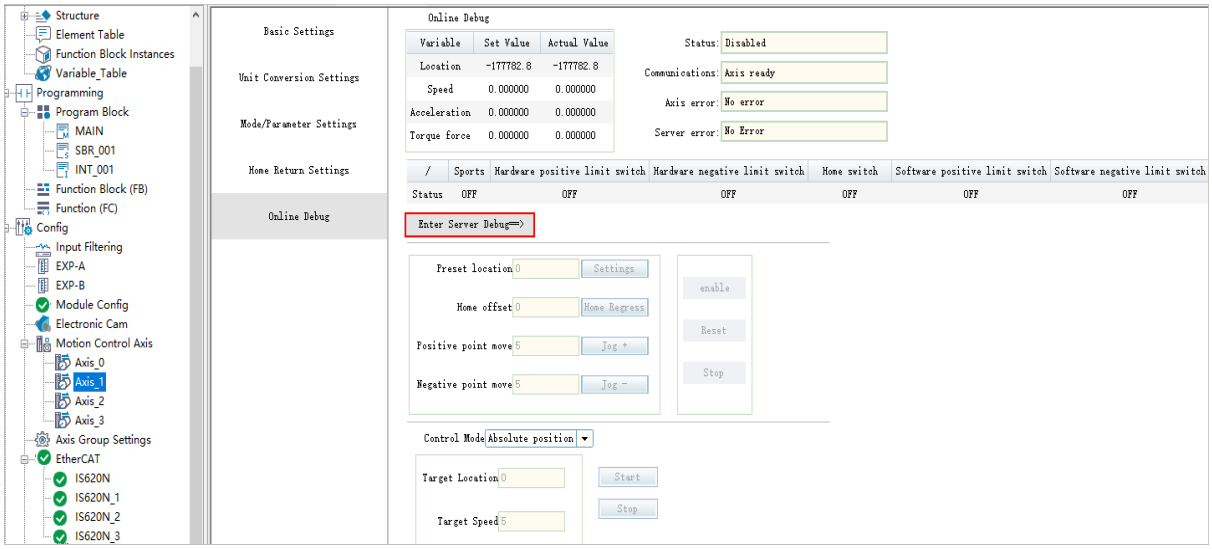


- Through the online commissioning page, you can see that the set position of the servo at this time has increased by 5 user units compared with the original.

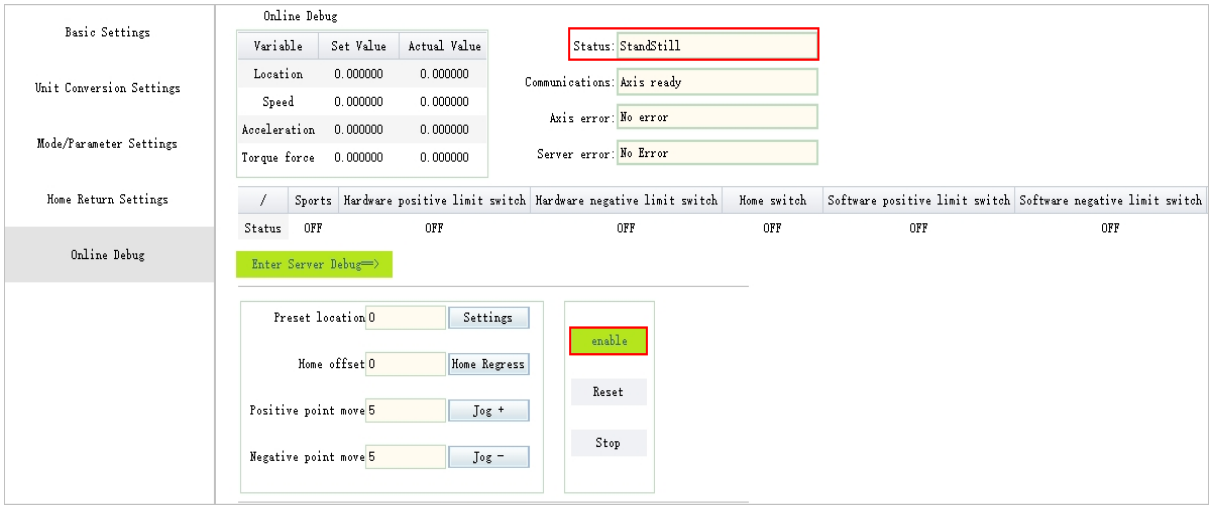


16.2.6.3 Online Commissioning

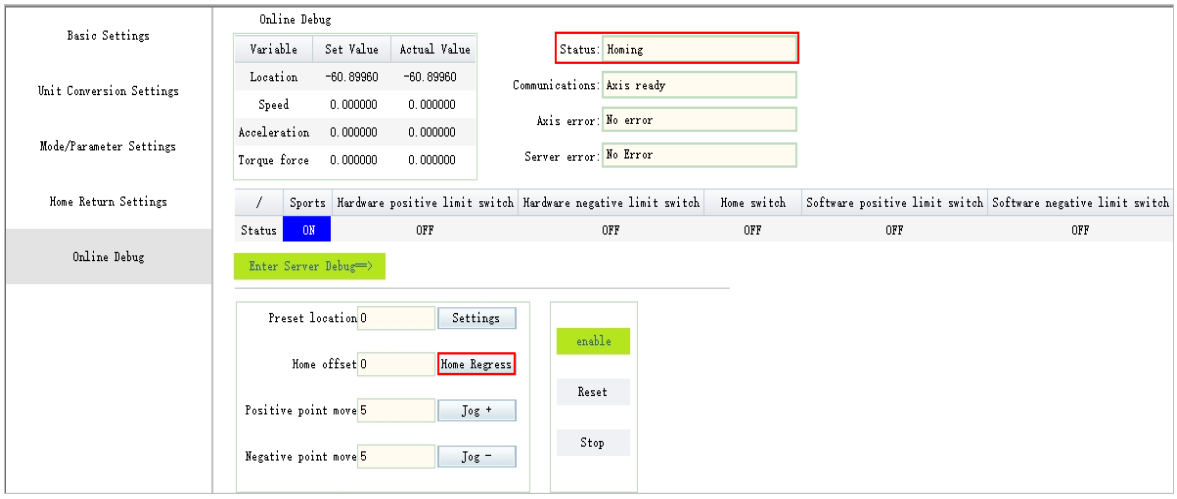
1. Open the online commissioning interface of the local pulse axis. Click "Enter Server Debug" to enter the commissioning mode.



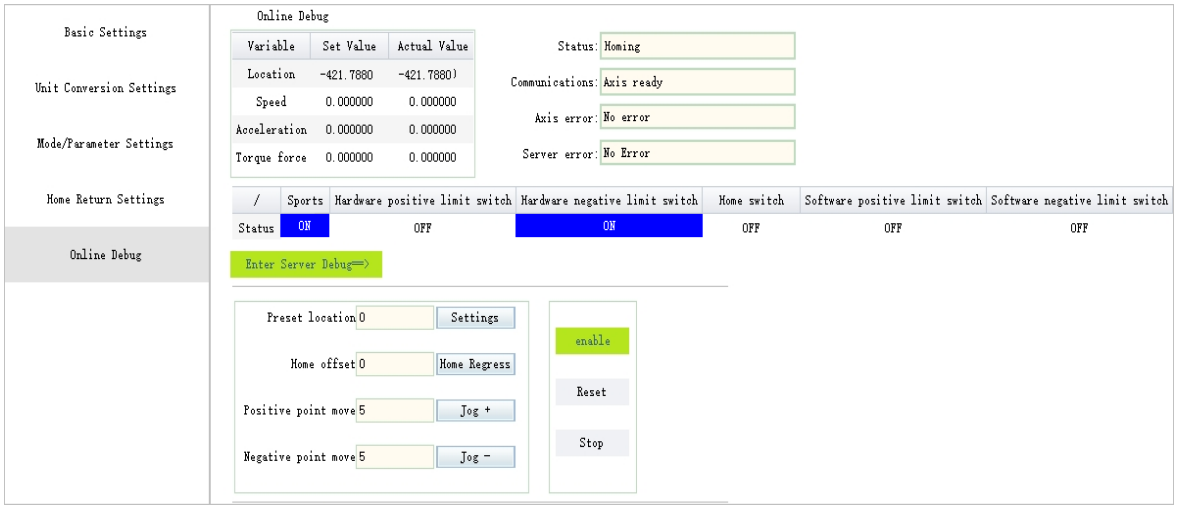
2. Click "Enable" to enable the servo drive.



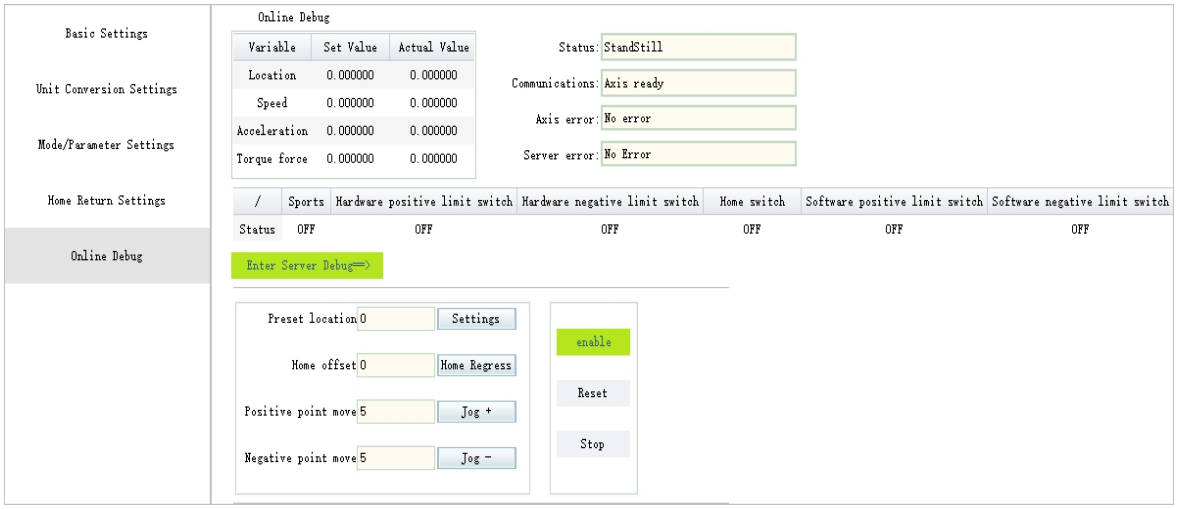
3. Click "Home Regress" to start homing. Correctly operate the negative limit variable M1000 to complete the homing action.
   - When M1000 is set to FALSE, homing begins and the axis returns to home through reverse motion.



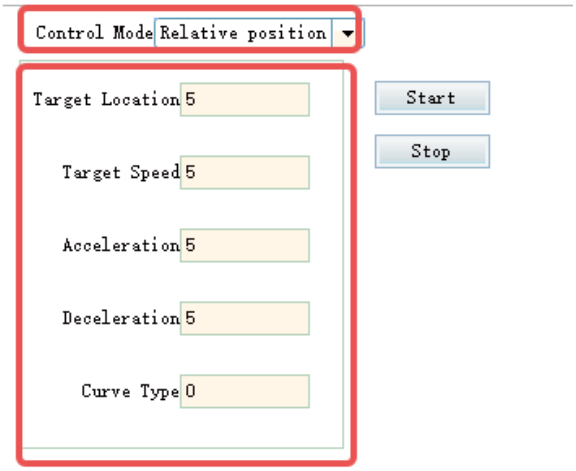
   - When M1000 is set to TRUE, the axis decelerates to stop after triggering the negative limit, and then moves forward at a low speed.

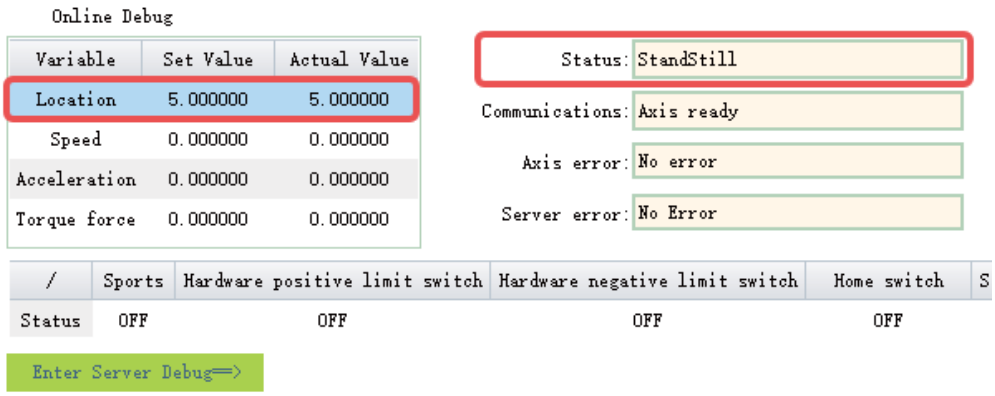


- Set M1000 to FALSE and complete homing.



4. Open the "Mode/Parameter Settings" tab page of the local axis pulse, select "Relative position" for "Control Mode", set the parameters as follows, and click "Start" to complete the relative positioning test.





16.3 Motion Control Axis Configuration

16.3.1 Comparison of Bus Servo Axis and Local Pulse Axis

The local pulse output and the EtherCAT drive are controlled with the same set of instructions, and share the same axis structure in design. The main differences between them are listed below.

Item	Local Pulse Output	EtherCAT Bus Drive
Different axis types	A local pulse axis needs to be selected.	A bus servo axis needs to be selected.
Different output devices	Local IO terminals need to be set up, with every two of them forming a group, that is, Y0/Y1, Y2/Y3, Y4/Y5, and Y6/Y7 groups. In the Y0/Y1 group, Y0 is the pulse terminal while Y1 is the direction terminal.	The PDO needs to be configured and mapped into the loop variable of the axis.
Pulse output forms	The two supported pulse forms are pulse+direction and CW/CCW, which can be selected in the "Mode/Parameter Settings" > "Output setting".	No setup is required.
Probe function	Two probes are supported, and each probe terminal can be selected from X0 to X7. This function needs to be selected in the "Mode /Parameter Settings" > "Probe setting".	The probe terminals need to be configured according to the application guide for the EtherCAT drive.
Homing settings	The supported homing modes (except for the Z signal) are specified in the CiA 402 protocol. The limit signal and home signal of the local pulse output axis can be selected through the page "Set the homing".	The homing modes No 1 to No 35 specified in the CiA 402 protocol are supported in setting, while the limits and home signals need to be set on the drive side.

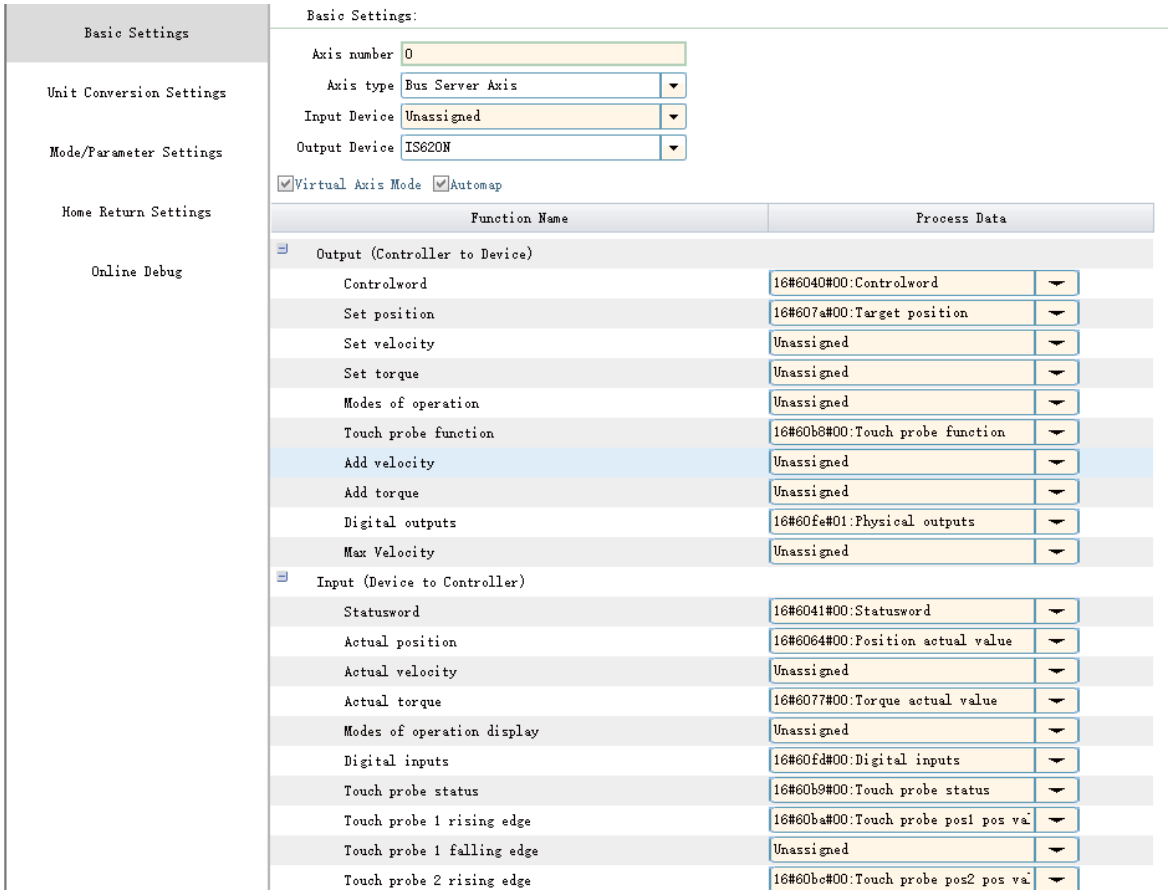
16.3.2 Basic Settings

NOTE

The software of AutoShop V4.11.0.0 and later and firmware of V6.5.0.0 and later support the Manufacturer Error code bus servo axis loop variable.

The "Basic Settings" page of an axis is used to set the type of the axis, and select a physical drive.

The following figure shows the "Basic Settings" page.



Name	Description
Axis number	Each axis is assigned a separate number in the range of 0 to 36, which cannot be modified manually. An axis number can be used as a unique input parameter of the MC instruction to access the axis.
Axis type	Optional axis types are bus servo axis, local pulse axis, bus encoder axis (not supported before version 4.2.0.0), and local encoder axis.
Input Device	It is used only for bus encoder axes and local encoder axes.
Output device	It is active only in bus servo axis and local pulse axis modes. In case of a bus servo axis, it is used to select EtherCAT servo drives; in case of a local pulse axis, it is used to select the local high-speed output terminals. Four groups of high-speed output terminals including Y0/Y1, Y2/Y3, Y4/Y5, and Y6/Y7 are available for selection.
Virtual Axis Mode	It is active only in bus servo axis and local pulse axis modes. When the virtual axis mode is ticked, the axis will no longer control the drive selected by the output device (high-speed output terminal), but will execute motion control instructions internally on a virtual servo axis.
Profile mode	It is active only in bus servo axis mode and mutually exclusive with the virtual axis mode. After the profile mode is selected, the host controller sets parameters through the bus, and the servo drive generates speed or torque references and performs speed or torque control.
Loop Variable	It is active only in bus encoder axis and bus servo axis modes. The EtherCAT slave communicates periodically based on the PDO and an axis is connected to the object

Name	Description
	dictionary of the EtherCAT slave through a loop variable. When automatic mapping is selected, the mapping process is assigned automatically and cannot be configured manually.
Automap	When this option is selected, the process data of the motion control axis is automatically mapped to the process data (input/output) of the servo. This function is selected by default after a motion control axis is added. When this option is selected, the input/output mapping of the axis cannot be modified on the "Basic Settings" page. When this operation is deselected, after the process data of a slave is modified, the input/output mapping of the axis is automatically refreshed.

Loop variables of the bus servo axis

The following table lists variables. For the detailed meaning of the object dictionary, see the standard CiA 402 protocol.

Loop Variable	Object Dictionary	Function
Controlword	0x6040	Control word
Set position	0x607a	Corresponding to the target position in servo drive CSP mode
Set velocity	0x60ff	Reserved
Set torque	0x6071	Corresponding to the target torque in CST mode of the servo drive.
Modes of operation	0x6060	Control mode, with a range of setting as follows. 6: Homing mode 8: Cyclic synchronous position (CSP) mode 9: Cyclic synchronous velocity (CSV) mode 10: Cyclic synchronous torque (CST) mode
Touch probe function	0x60b8	Probe control word
Add velocity	Reserved	Reserved
Add torque	Reserved	Reserved
Digital outputs	0x60fe:1	Digital output (DO)
Max Velocity	0x60FF	Maximum velocity
Profile velocity	6081h	Profile velocity
Profile acceleration	6083h	Profile acceleration
Profile deceleration	6084h	Profile deceleration

Loop Variable	Object Dictionary	Function
Torque slope	6087h	Torque slope
Statusword	0x6041	Status word
Actual position	0x6064	Feedback position
Actual velocity	0x606c	Feedback speed
Actual torque	0x6077	Feedback torque
Modes of operation display	0x6061	Current control mode
Digital inputs	0x60fd	DI terminal status, with functions as follows: Bit2: Home switch Bit1: Positive limit switch Bit0: Negative limit switch
Touch probe status	0x60b9	Touch probe status
Touch probe 1 rising edge	0x60ba	Probe 1 position on the rising edge
Touch probe 1 falling edge	0x60bb	Probe 1 position on the falling edge
Touch probe 2 rising edge	0x60bc	Probe 2 position on the rising edge
Touch probe 2 falling edge	0x60bd	Probe 2 position on the falling edge
Errorcode	0x603f	Drive error code
Manufacturor Error code	203Fh	Fault subcode **Note**: If PDO adds this map, it will be automatically mapped, and the fault code of the servo can be read.

The following table lists parameters that need to be set for unit conversion.

Parameter Name	Function
The number of pulses per revolution of the motor/ encoder	Set the number of pulses required for the motor to rotate one turn according to the encoder resolution.
Gear change mechanisms in use or not	Specify whether gear change mechanisms are in use or not.
Amount of movement per revolution of the motor/ encoder	The workpiece movement amount per revolution of the motor when no gear change mechanism is in use

Parameter Name	Function
Amount of movement per revolution of the worktable	The distance per revolution on the workpiece side when gear change mechanisms are in use
Gear ratio on the workpiece side	Set a gear ratio on the workpiece side.
Gear ratio on the motor side	Set a gear ratio on the motor side.

The bus drives (local pulse axes) use pulse units when controlling a motor, and use common measurement units for motion control instructions, such as millimeters, degrees, and inches, which are called user units. According to the configuration parameters, the two units are converted to each other within an axis. The conversion between the two units is divided into the following modes. The following takes the user unit "degree" as an example.

- Do not use gearbox.
  When the gear change mechanisms are not in use, the conversion equation from user unit to pulse unit is as follows.

  $$\text{Number of pulses (unit: pulse)} = \frac{\text{Number of pulses per revolution of the motor/encoder [DINT]}}{\text{Distance per revolution of the workbench [REAL]}} \times \text{Distance (unit: Unit)}$$

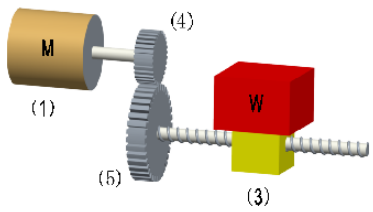
  Take the Inovance 20-bit encoder as an example. The set parameters are as follows.

  Number of pulses per revolution of the motor/encoder = 1048576

  Amount of movement per revolution of the motor/encoder = 1

  Then, when the target displacement given by the relative positioning instruction is 10, the actual number of pulses sent by the motion control axis is 10485760, then the motor rotates by 10 revolutions.

- With gear change mechanisms
  Typical working condition in linear mode is shown in the following figure.

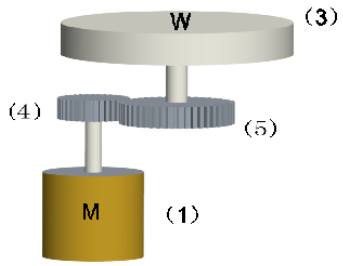
  

  In the figure, (1) is the servo motor, (3) is the workpiece, (4) is the gear ratio denominator, and (5) is the gear ratio numerator.

  The calculation equation from user unit to pulse unit is as follows.

  $$\text{Number of pulses (unit: pulse)} = \frac{\text{Number of pulses per revolution of the motor/encoder [DINT]} \times \text{Gear ratio numerator [DINT]}}{\text{Distance per revolution of the workbench [REAL]} \times \text{Gear ratio denominator [DINT]}} \times \text{Distance (unit: Unit)}$$

  Typical working condition in rotary mode is shown in the following figure.



In the figure, (1) is the servo motor, (3) is the workpiece, (4) is the gear ratio denominator, and (5) is the gear ratio numerator.

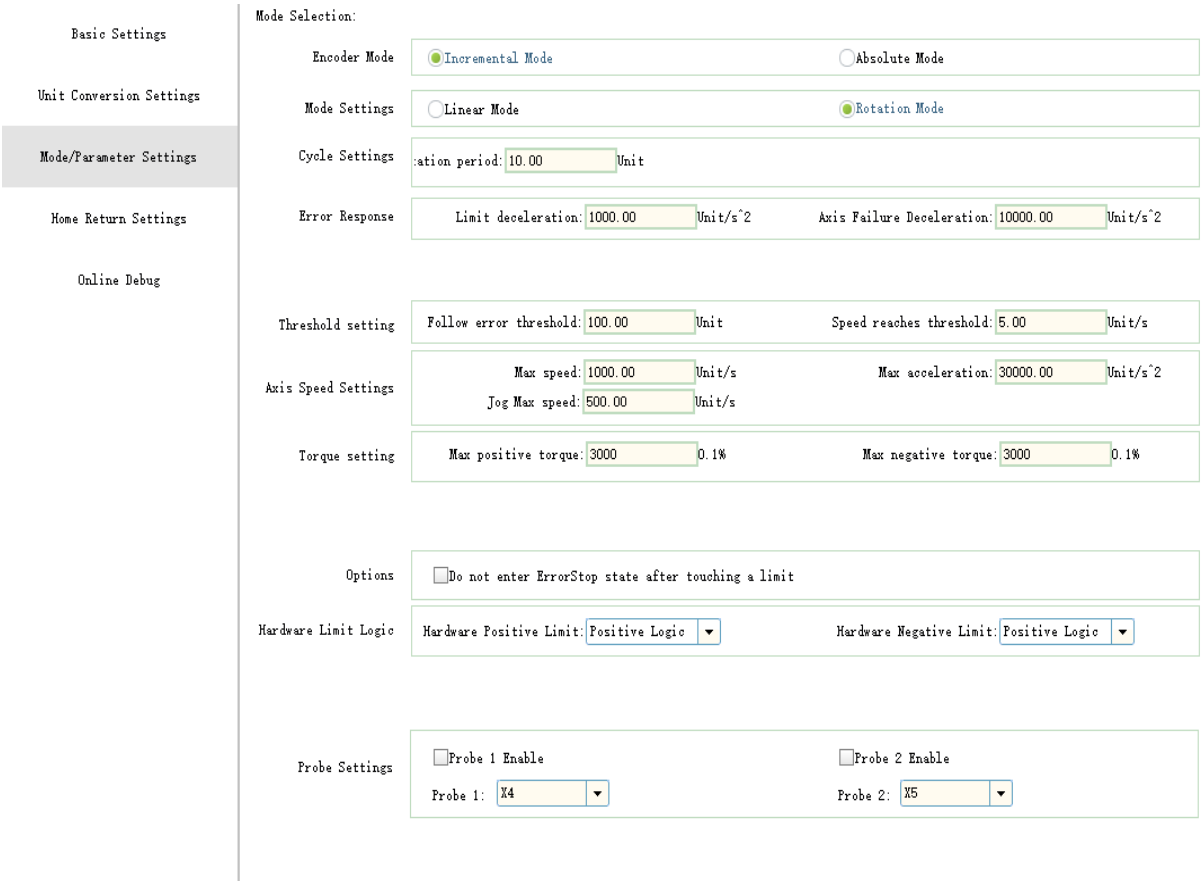
The calculation equation from user unit to pulse unit is as follows.

$$\text{Number of pulses (unit: pulse)} = \frac{\text{Number of pulses per revolution of the motor/encoder [DINT]} \times \text{Gear ratio numerator [DINT]}}{\text{Distance per revolution of the workbench [REAL]} \times \text{Gear ratio denominator [DINT]}} \times \text{Distance (unit: Unit)}$$

16.3.3 Mode/Parameter Setting

16.3.3.1 Configuration Interface

The following figure shows the "Basic Settings" page. The parameter list varies with the selected axis type.



16.3.3.2 Encoder Mode

The encoder mode is only valid in the bus servo axis mode and is used with incremental encoder servo drives and absolute encoder servos. Choose the mode according to the type of servo drive actually used . The processing on the PLC side is as follows.

Incremental mode

The increase in the number of revolutions caused by the overflow of the 32-bit counter of servo drive encoder is not taken into account on the PLC side. The PLC does not save the current position of the encoder when the power is turned off and on again. After the second power-on, the current position of the axis is calculated only according to the position of a single revolution given by the servo drive.

Absolute mode

The increase in the number of revolutions caused by the overflow of the 32-bit counter of servo drive encoder is taken into account on the PLC side. The PLC saves the current position given the encoder when the power is turned off and on again. After the second power-on, the encoder position of the axis saved inside the PLC and the position given the servo drive are read during the initialization to calculate the current absolute position of the axis.

> **NOTE**
>
> If an Inovance servo is used and process parameters 200B.3Bh (low-order 32 bits of mechanical absolute position) and 200B.3Dh (high-order 32 bits of mechanical absolute position) are configured, the axis calculates its current absolute position during initialization. The calculation is based on the 64-bit mechanical absolute position feedback from the servo drive and the position offset saved in the PLC. This can effectively prevent data overflow in the 32-bit counter.

16.3.3.3 Mode Settings

Set the motion control axis to linear mode or ring mode based on actual working conditions.

Linear mode

Linear mode is usually used for devices that move within the range of mechanical actions in the X-Y linear coordinate system.

Linear mode typically involves a zero point.

An increase in the feedback position during motion indicates a forward motion, while a decrease indicates a reverse motion.

Forward and reverse software limits can be set. After the software limit is enabled, an axis can only move within the limit range.

Absolute positioning mode: When the target position is greater than the start position, the axis moves forward over a distance equal to the result of the target position minus the start position. When the target position is less than the start position, the axis moves reversely over a distance equal to the result of the start position minus the target position.

Relative positioning mode: When the target displacement is greater than 0, the axis moves forward over a distance equal to the target displacement. When the target displacement is less than 0, the axis moves reversely over a distance equal to the target displacement.

Processing mode of velocity instructions in linear mode: If the target velocity is greater than 0, a forward motion is performed. If the target velocity is less than 0, a reverse motion is performed.

Ring mode

The ring mode is a mode in the form of a ring counter that is capable of infinitely repeated counting within a set range. It is usually used in revolving stages or rolls.

The ring mode usually involves a zero point and a rotation cycle. The feedback position of a ring counter is ≤ 0 and less than one rotation cycle.

In ring mode, an increase in feedback position indicates a CW motion, and a decrease in feedback position indicates a CCW motion.

Software limits are not applicable in ring mode.

- Processing mode of relative positioning: When the target displacement is greater than 0, the axis moves clockwise over a distance equal to the target displacement. When the target displacement is less than 0, the axis moves counter-clockwise over a distance equal to the target displacement.
- Processing mode of absolute positioning:
  Forward: First, take the modulus of the target position divided by the rotation cycle. Then, move the axis in a CW manner from the start position to the target position.

  Reverse: First, take the modulus of the target position divided by the rotation cycle. Then, move the axis from the start position to the target position in a CCW manner.

  The shortest distance: First, take the modulus of the target position divided by the rotation cycle to obtain the target position. Then, calculate the displacement from the start point in a CW manner to the target position. If the displacement is less than or equal to half cycle, a CW motion is performed; otherwise, a CCW motion is performed to the target position.

  Current direction: A motion is performed to the target position in the latest axis motion direction, or a forward motion is performed to the target position in case of initial power-on.

  Processing mode of velocity instructions in ring mode: If the target velocity is greater than 0, a CW motion is performed; if the target velocity is less than 0, a CCW motion is performed.

16.3.3.4 Software Limits

Software limits can be set in linear mode .

When software limits are valid, the system constantly detects the absolute position of an axis that decelerates from the current velocity to 0 at the set limit deceleration rate according to T-type deceleration mode. If the absolute position of the axis is beyond the limit range, the axis will execute the software limit deceleration algorithm and the positioning or velocity instruction being executed will be aborted.

Software limits are invalid in homing or torque mode.

16.3.3.5 Deceleration Upon Axis Fault

During the operation of an axis, if the axis must switch to the ErrorStop state due to a logic failure of the motion instruction itself, the axis will do T-type deceleration according to the setting of deceleration upon axis fault. The axis will not enter the ErrorStop state until it decelerates to 0.

16.3.3.6 Following Error

During the execution of positioning and velocity instructions, the servo drive actually works in cyclic synchronous position (CSP) mode, and the planning of the position curve is done on the PLC. The PLC sends the target position to the servo drive through 0x607A, the servo drive drives the servo motor to move, and the position of the motor encoder is sent back to the PLC through 0x6064. Due to the inherent features of the servo drive and the motor, a difference between 0x607A and 0x6064 is generated. This difference is called

the following error when converted to user units. You can set a following error limit. If the absolute value of the following error of an axis exceeds the limit, the axis reports an excessive following error and enters the ErrorStop state.

The following error calculation method of local pulse axis is the same as that of the bus servo axis, that is, calculating the difference between 0x607A and 0x6064, and then converting this difference to user units.

16.3.3.7 Axis Speed Setting

You can set three parameters: maximum velocity, maximum acceleration, and maximum jogging velocity. When a parameter, such as the target velocity, acceleration, and deceleration, in the positioning instruction or velocity instruction exceeds the velocity limit, the relevant instruction reports an error and the axis enters the ErrorStop state.

For the bus servo axis, the maximum velocity is converted into pulse units and written into the object dictionary 0x607f of the servo drive in the form of startup parameters.

16.3.3.8 Torque Setting

The torque setting is only applicable to the bus servo axis.

If the target torque exceeds the maximum torque in the torque instruction, the instruction reports an error and the axis enters the ErrorStop state.

The forward torque limit will be written into the object dictionary 0x60e0 of the servo drive in the form of startup parameters, and the reverse torque limit will be written into the object dictionary 0x60e1 of the servo drive in the form of startup parameters.

16.3.3.9 Probe Setting

Probe terminals can be enabled for local pulse axes through probe setting.

Each local pulse axis can be configured with up to two probe terminals. The probe terminals can be selected from X0 to X0-X7.

When a probe terminal is enabled, the local pulse axis can use probe instructions and interrupt positioning instructions.

16.3.3.10 Output Setting

You can select Y0/Y1, Y2/Y3, Y4/Y5, and Y6/Y7 as four local pulse axes.

The H5U series and Easy301 support 4-axis pulse output, and other Easy series support 5-axis pulse output.

The output of local pulse axes can be set as pulses in the format of pulse+direction or CW/CCW.

For channels that are set as pulse axes, when pulse+direction is selected, Y0, Y2, Y4, and Y6 are the pulse terminals, and Y1, Y3, Y5, and Y7 are the direction terminals. When CW/CCW is selected, Y0, Y2, Y4, and Y6 are CW pulse terminals, and Y1, Y3, Y5, and Y7 are CCW terminals.

16.3.3.11 Options

This function is only available for AutoShop 4.0.0.0 matching PCB software of version 3.0.0.0 or later.

- When this option is selected, the motion mode is synchronized, and only the axis control instruction reports an error when the motion control axis reaches the limit, but the motion control axis does not enter the ErrorStop state.
- When this option is not selected, if the motion control axis reaches the limit after a motion control instruction other than MC_Jog is called, the motion control axis enters the ErrorStop state, and the instruction reports an error.

16.3.3.12 Hardware Limit Logic

This function is only available for AutoShop 4.0.0.0 matching PCB software of version 3.0.0.0 or later.

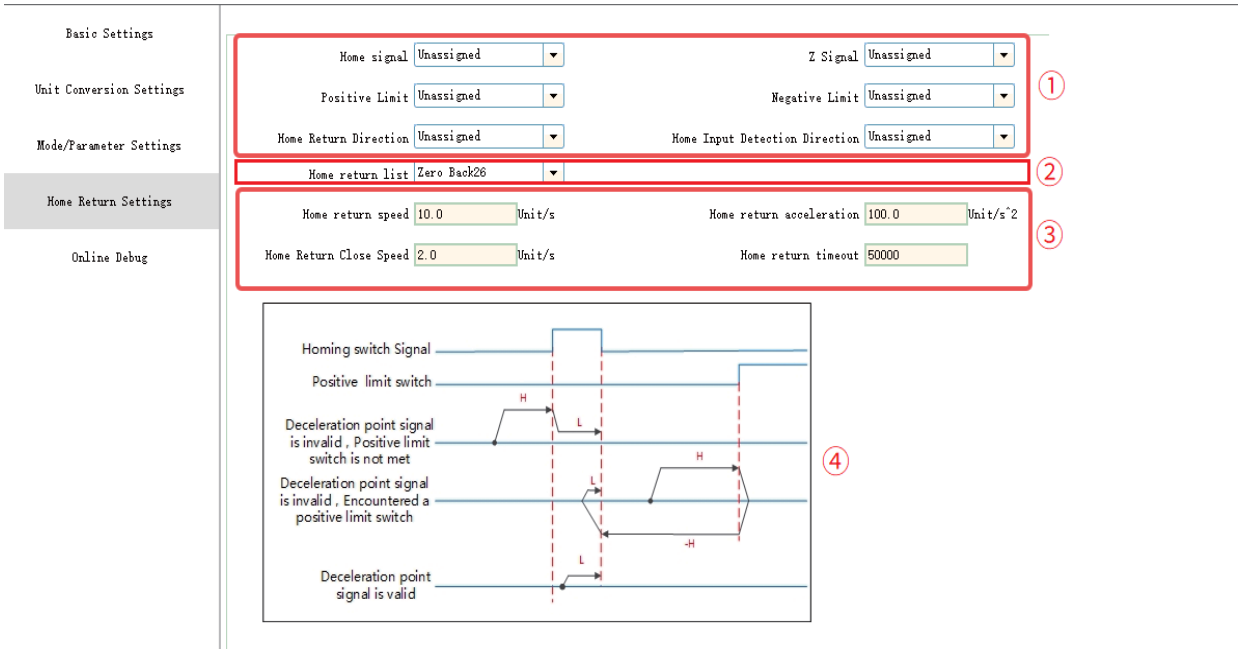
Among axis system variables, bphlimit and bnhlimit indicate the hardware positive limit and hardware negative limit, respectively. When the hardware limit logic is positive, they correspond to the values of bit1 and bit0 in the object dictionary 0x60fd, respectively. When the hardware limit logic is negative, they are inverted values of bit1 and bit0 in the object dictionary 0x60fd, respectively.

The hardware limit logic setting is only reflected in the above variables and has no impact on the limit stop processing when the servo reaches the limit.

16.3.4 Homing

The homing modes No. 1 to No. 35 specified in the CiA 402 protocol are supported.

The homing settings page is as follows.



No.	Description
①	Set the filter condition.
②	Select a homing mode.
③	Set the homing parameters.

No.	Description
④	Illustration

The following table lists the parameters that need to be set.

Setting parameters	Description
Home signal	It is used to choose whether to use a home signal. • When "Unassigned" is selected, it is not used as a mandatory filter. • When "Do not use" is selected, the homing modes that must use a home signal are removed. • When "Use" is selected, the homing modes that do not support a home signal are removed.
Negative Limit	It is used to select whether to use a hardware left limit signal. • When "Unassigned" is selected, it is not used as a mandatory filter. • When "Do not use" is selected, the homing modes that must use a negative limit signal are removed. • When "Use" is selected, the homing modes that do not support a negative limit signal are removed.
Positive Limit	It is used to choose whether to use a hardware right limit signal. • When "Unassigned" is selected, it is not used as a mandatory filter. • When "Do not use" is selected, the homing modes that must use a positive limit signal are removed. • When "Use" is selected, the homing modes that do not support a negative limit signal are removed.
Z Signal	It is used to choose whether to use a motor Z signal. • When "Unassigned" is selected, it is not used as a mandatory filter. • When "Do not use" is selected, the homing modes that must use a Z signal are removed. • When "Use" is selected, the homing modes that do not support a Z signal are removed.
Home Return Direction	It is used to set the direction of motion at the beginning of homing. • Forward: The direction of motion is forward when the limit (home) signal input is inactive, otherwise it is the opposite. • Reverse: The direction of motion is reverse when the limit (home) signal input is inactive, otherwise it is the opposite.
Home Input Detection Direction	The direction of motion when the home signal is reached • Forward: stopping at the edge of the home signal during forward motion.

Setting parameters	Description
	• Reverse: stopping at the edge of the home signal during the reverse motion.
Home return list	The homing modes, with a setting range of 1 to 35, are written into the object dictionary 0x6098 in the form of startup parameters.
Homing mode	It is used to set the relative mode or absolute mode in homing mode No. 35, which are written into the object dictionary 0x60e6 in the form of startup parameters.
Home return speed	The homing velocity is written into the sub-index No. 1 of the object dictionary 0x6099 in the form of startup parameters after the user units are converted into pulse units.
Home Return Close Speed	The homing closing velocity is written into the sub-index No. 2 of the object dictionary 0x6099 in the form of startup parameters after the user units are converted into pulse units.
Home return acceleration	The homing acceleration is written into the object dictionary 0x609A in the form of startup parameters after the user units are converted into pulse units.
Home return timeout	This parameter is exclusive to Inovance drives.

In actual use, the homing mode is defined by several parameters, such as home signal, positive limit, negative limit, Z signal, homing direction, and homing detection direction, and then selected from the options of the homing list as needed.

Note that multiple homing modes are still left after the conditions for homing are filtered, in which case you can choose an appropriate homing mode from the homing list. For example, if the parameters are set as in the following table, you can filter out two homing modes.

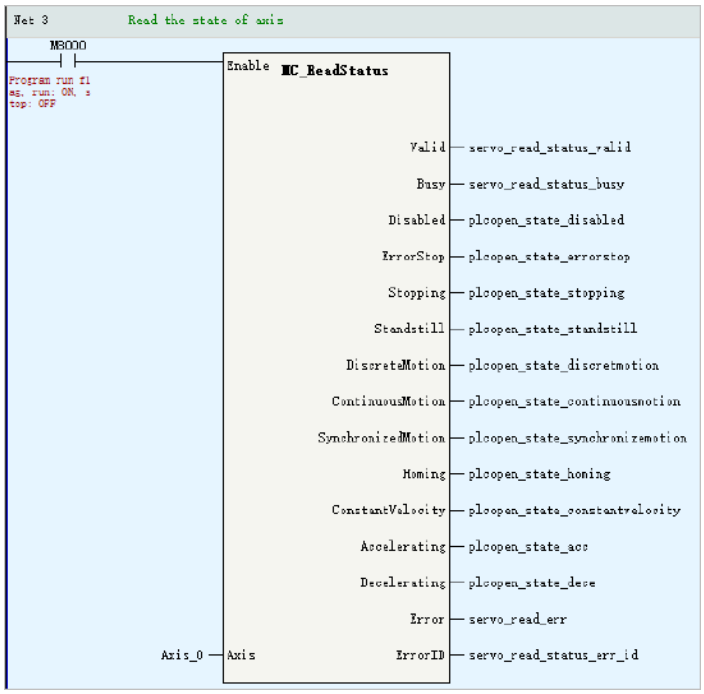
Signal	Value
Home signal	Use
Negative limit	Do not use
Positive limit	Use
Z signal	Use
Homing direction	Forward
Homing detection direction	Forward

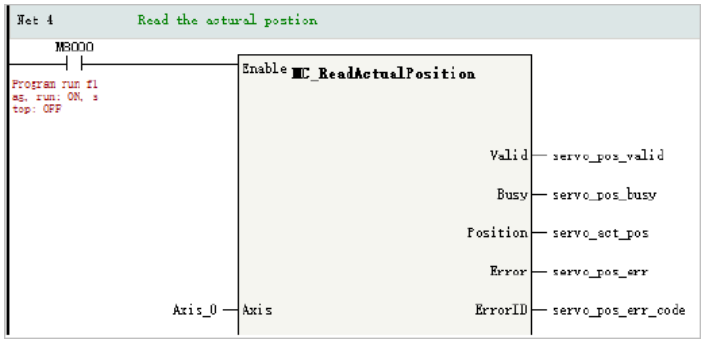
According to the above settings, the two homing modes No. 8 and No. 10 can be filtered out, and the difference between the two is whether the home input signal is active when the homing is completed.

16.4 Online Monitoring

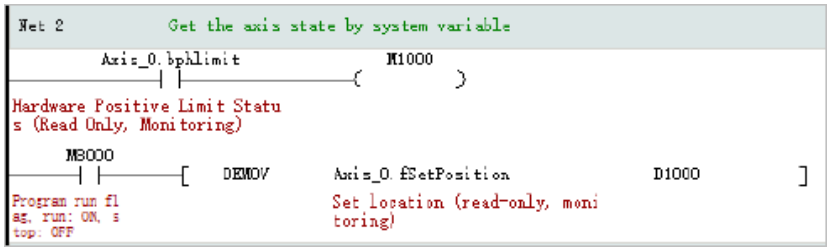
You can obtain the status of an axis through PLC instructions, system variables of the axis, and the online commissioning interface in the background.

1. Obtaining the axis status through instructions
   You can obtain the status of an axis through status instructions such as MC_ReadStatus and MC_ReadPosition.

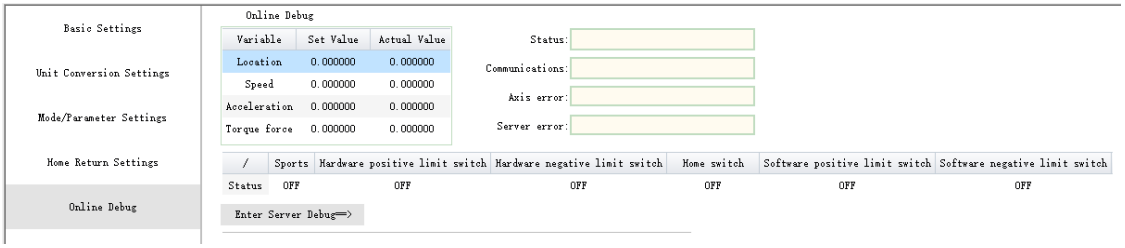




2. Obtaining the status of an axis through system variables
   With system variables, you can view the motion status of each axis in real time.



3. Obtaining the axis status through the online commissioning interface
   Through the online commissioning interface, you can view the motion status of each axis in real time. The online monitoring interface of an axis is shown in the following figure.



The following table shows the data that can be monitored.

Target	System variable	Function
Position reference	fSetPosition	Target position (user unit) when PLC performs path planning
Velocity reference	fSetVelocity	Target velocity (user unit) when PLC performs path planning
Acceleration reference	fSet_Acc_Dec	Target acceleration (user unit) when PLC performs path planning
Torque reference	fSetTorque	Target torque when PLC performs torque planning (%)
Actual position	fActPosition	Current position of drive feedback (user unit)
Actual velocity	fActVelocity	Velocity calculated from actual position (user unit)
Actual acceleration	fAct_Acc_Dec	Acceleration calculated from actual velocity (user unit)
Actual torque	fActTorque	Actual torque of drive feedback (%)
Status	wPLCOpenState	Status of PLCopen state machine 0: Poweroff 1: ErrorStop 2: Stop 3: StandStill 4: DiscreteMotion 5: ContinuousMotion 7: Homing 8: Synchronized motion
Communication	wConfigState	The status of data communication between the motion control axis and the drive 0: Init (axis in the initialization state) 1: Configure finish (reading of configuration data completed) 2: Sync finish (synchronized with EtherCAT tasks)

Target	System variable	Function
		3: Wait communication (communication with the servo drive established) 4: Slave ready (initialization completed for the servo drive controlled by axes) 5: Axis ready (communication established)
Axis error	wAxisError	The internal error of a motion control axis
Servo error	wServoError	For a local pulse axis, it is the fault code of the axis; for a bus servo axis, it is the corresponding 0x603F value. See the guide for the drive used.
Motion	bMotionState	Indicates whether a motion control axis is currently in motion.
Hardware positive limit	bphlimit	Indicates whether the hardware positive limit input for a motion control axis is valid.
Hardware negative limit	bnhlimit	Indicates whether the hardware negative limit input for a motion control axis is valid.
Home switch	bhomestate	Indicates whether the home switch input of a motion control axis is valid.
Software positive limit	bpslimit	Indicates whether a motion control axis has reached the software positive limit.
Software negative limit	bnslimit	Indicates whether a motion control axis has reached the software negative limit.

16.5 Axis Control Functions

16.5.1 Overview

Basic servo control can be realized through the online commissioning function, such as enable, stop, jogging, and point-to-point control. After the basic actions are confirmed to be normal, complex logic control can be realized through motion control instructions. Online commissioning and PLC instruction control cannot be used at the same time. The restrictions are as follows:

- When the MC_Stop instruction is called to make an axis enter the Stopping state, the axis cannot enter the online commissioning mode through the background.
- The MC_Power instruction and the enable function in online commissioning are in a OR relationship, that is, when either of them is valid, the axis can be enabled.
- Motion instructions, such as MC_MoveAbsolute, that have a lower priority than the online commissioning are invalid when an axis is in the online commissioning mode. In this case, the instructions will report an error if called, but the axis will not enter the fault state.

16.5.2 Online Commissioning

The functions that can be achieved by online commissioning are as follows:

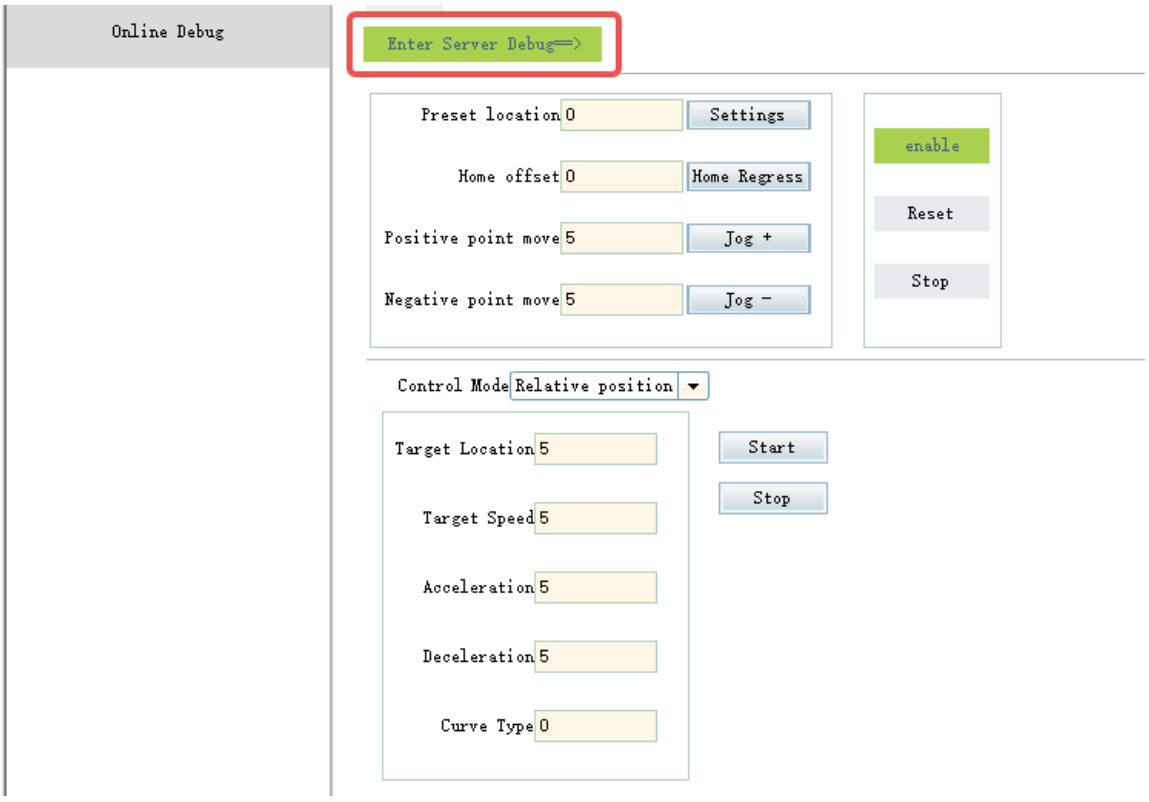
Function point	System variable	Description
Entering the online commissioning mode	bEnterDebug	Makes an axis enter the online commissioning mode, after which the PLC will not continue to execute motion control instructions.
Enable	bPowerOn	Similar to the way to call the MC_Power instruction to put an axis in an enabled or disabled state.
Reset	bReset	Resets a fault that occurs on an axis. This instruction is equivalent to the MC_Reset instruction.
Stop	bStop	Stops the motion of an axis. This instruction is equivalent to the MC_Stop instruction.
Homing	bHome	Performs a homing action. This instruction is equivalent to the MC_Home instruction.
Set the current position.	bSetPos	Sets the current position of an axis. This instruction is equivalent to the MC_SetPosition instruction.
Jog	bJogP/bJogN	Implements the jog function. This instruction is equivalent to the MC_Jog instruction.
Motion type	bDebugMotionType	Selects a motion mode from the following options: 1: Absolute positioning 2: Relative positioning 3: Continuous motion 5: Reciprocating motion 6: Torque mode
Relative positioning	bRelPos	Implements the relative positioning. This instruction is equivalent to the MC_MoveRelative instruction.
Absolute positioning	bAbsPos	Implements the absolute positioning. This instruction is equivalent to the MC_MoveAbsolute instruction.
Continuous motion	bVelocity	

Function point	System variable	Description
		Implements continuous motion at a certain velocity. This instruction is equivalent to the MC_MoveVelocity instruction.
Reciprocating motion	bRevPos	Implements the reciprocating motion between two absolute positions. This instruction is equivalent to two MC_MoveAbsolute instructions in a loop.
Torque control	bTorque	Implements the torque control. This instruction is equivalent to the MC_MoveTorque instruction.

Procedure

1. Entering the online commissioning mode

   As shown in the following figure, click “Enter Server Debug” to put an axis in the online commissioning mode.

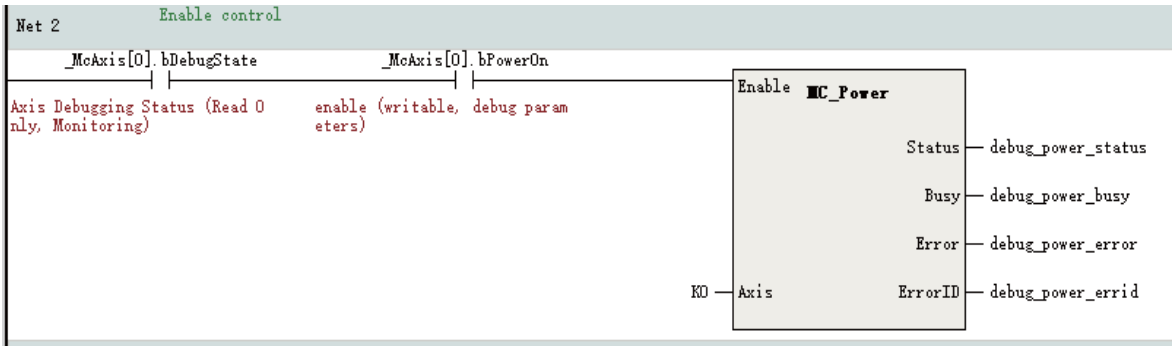


   After receiving the instruction to enter the online commissioning mode, the background will do the following checks:

   - If the current axis is in the Stopping state, the axis cannot enter online commissioning mode.
   - If the current axis is already in motion, the user will be asked whether to enter the online commissioning mode. If the axis is forced to enter the online commissioning mode, the original motion state will be interrupted and the axis motion will stop.
   - If the axis is already enabled when entering the online commissioning mode, the axis remains enabled after entering the mode.

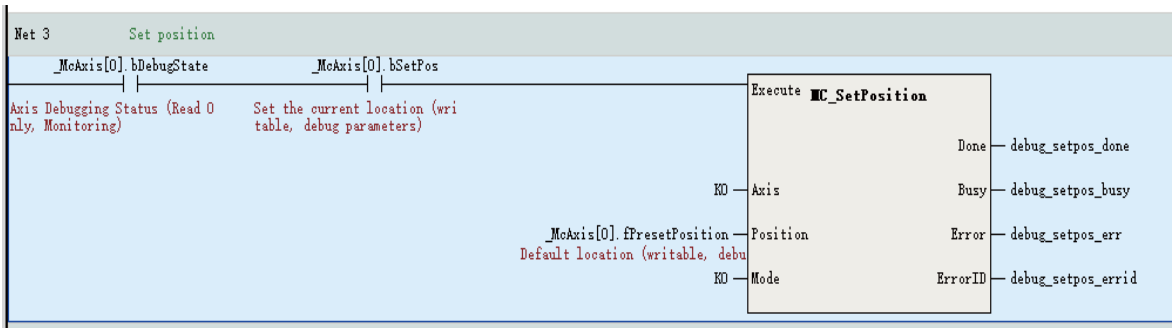
2. Basic operations

- Enable: After an axis enters the online commissioning mode, click “Enable” to make the axis enter the enabled state, and the execution effect is equivalent to the following instructions.

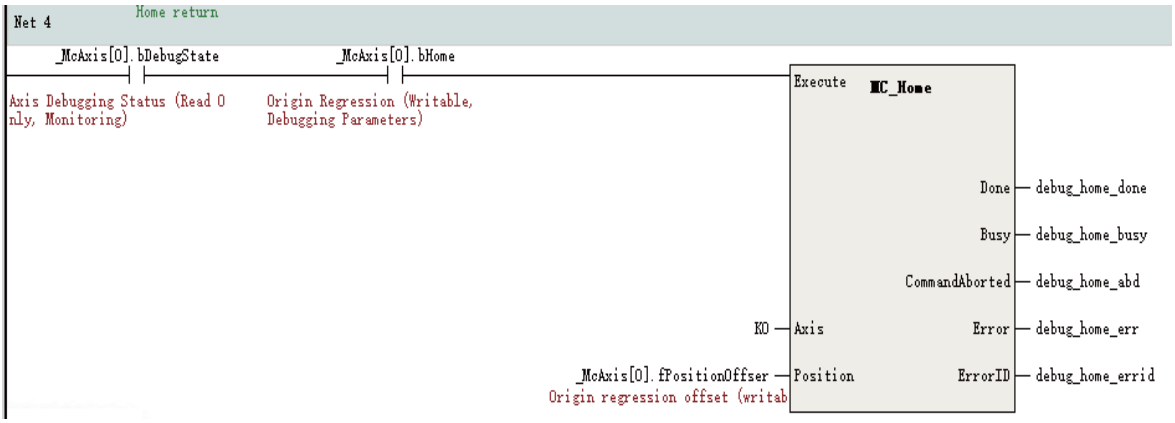


> **NOTE**
>
> The enable-control in online commissioning and the MC_Power instruction called in the PLC program jointly control the enable-state of the axis. When either of the inputs is active, the axis is enabled.

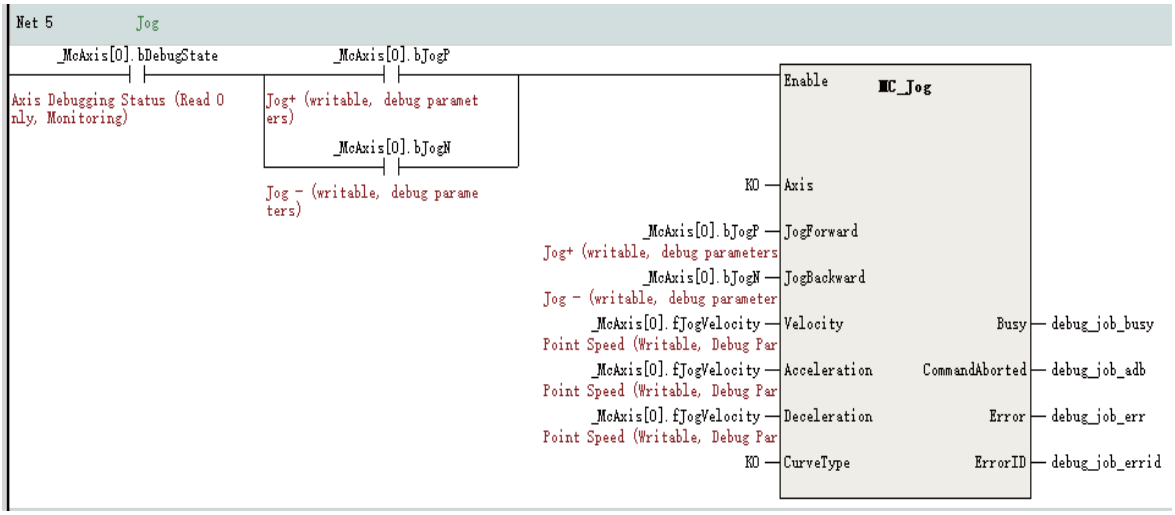
- Preset position: When an axis is in non-motion mode, click “Settings” to write the value in the preset position field to the axis. The execution logic is as follows:



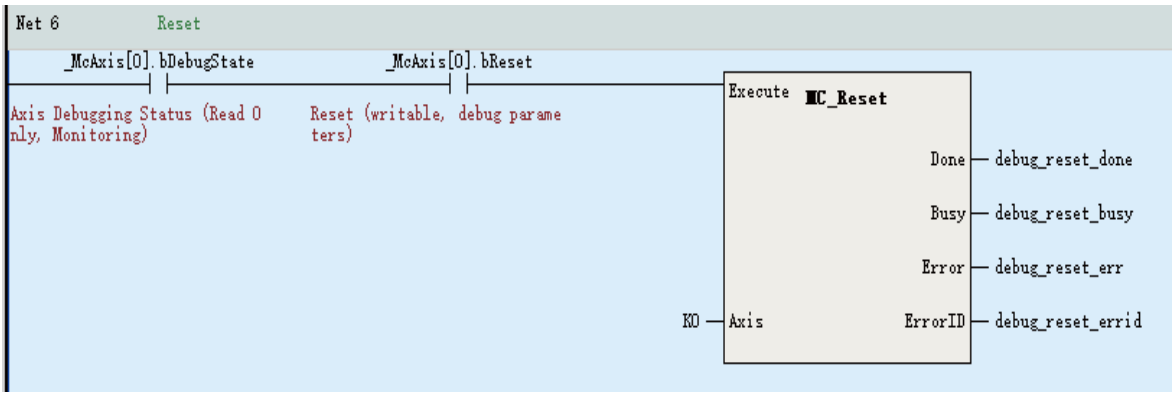
- Homing control: When an axis is in the standstill state, click Home return, the axis will control the servo drive to perform the homing operation. When the homing is completed, the value set in Home offset will be written into the servo drive. The execution logic is as follows:



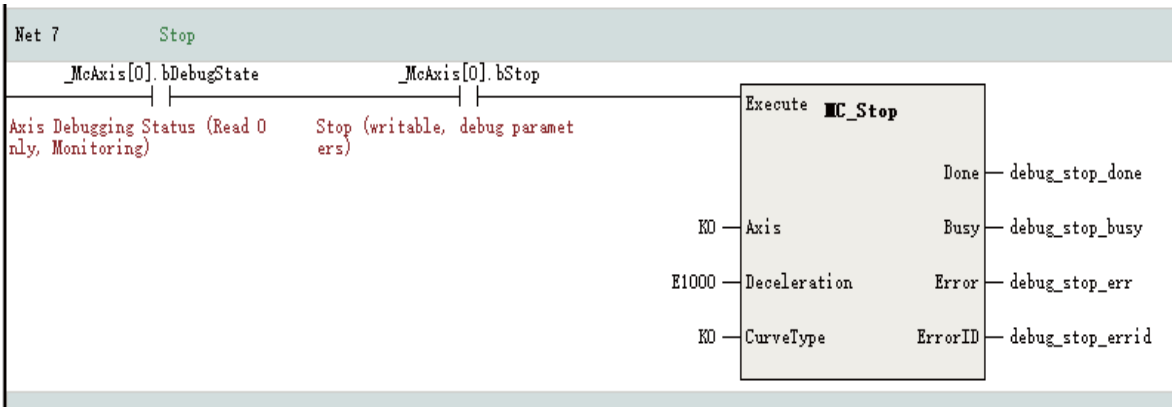
- Jog: When an axis is in the Standstill state, click “Jog+” to move the axis in the forward direction according to the target velocity set in the “Positive point move” field. Click “Jog-” to move the axis in the opposite direction according to the target velocity set in the “Negative point move” field. The execution logic is as follows:



- Reset: When an axis is in Errorstop state, click “Reset” in attempt to reset the axis faults. If the servo drive fault cannot be reset, the reset may fail. The execution logic is as follows:



- Stop: There are two stop buttons in the interface, both with the same function, and both for stopping the motion of an axis. The deceleration used is the limit deceleration in the mode/parameter settings of an axis. The execution logic is as follows:

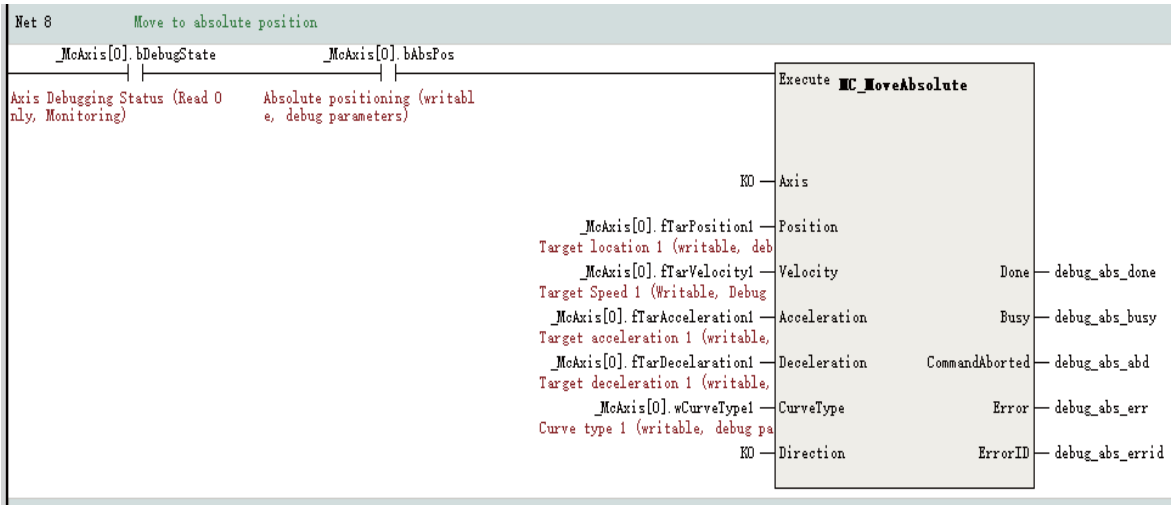


3. Motion mode

   A motion mode can only be set when an axis is enabled.

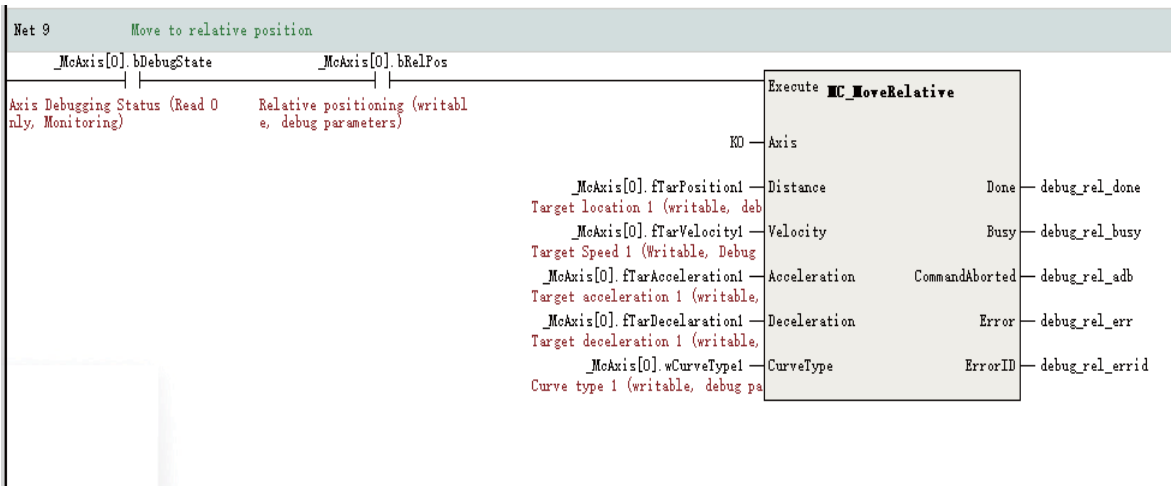
- Absolute positioning

  After the control mode is set to absolute positioning, you can set five parameters: target position, target velocity, acceleration, deceleration, and curve type. Click the start button to start absolute positioning according to the above parameters. The execution logic is as follows:



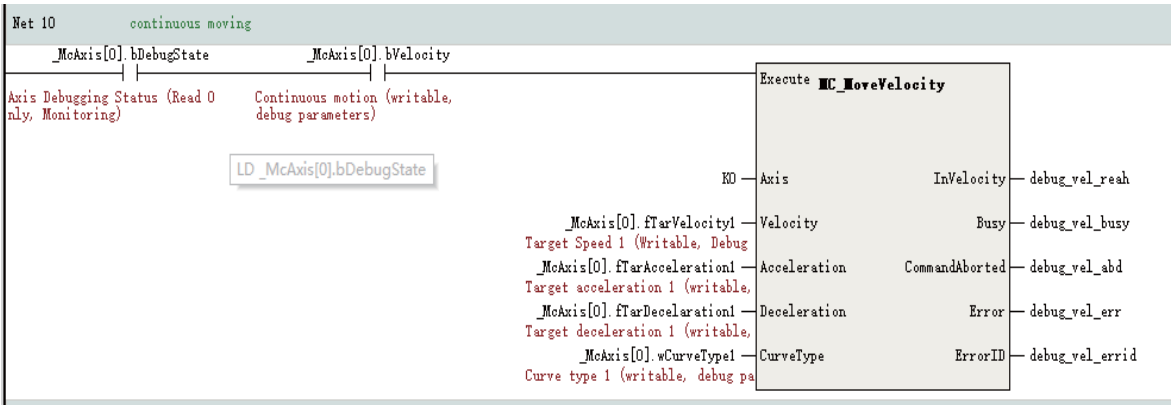
- Relative positioning

  After the control mode is set to relative positioning, you can set five parameters: target position, target velocity, acceleration, deceleration, and curve type. Click the start button to start relative positioning according to the above parameters. The execution logic is as follows:



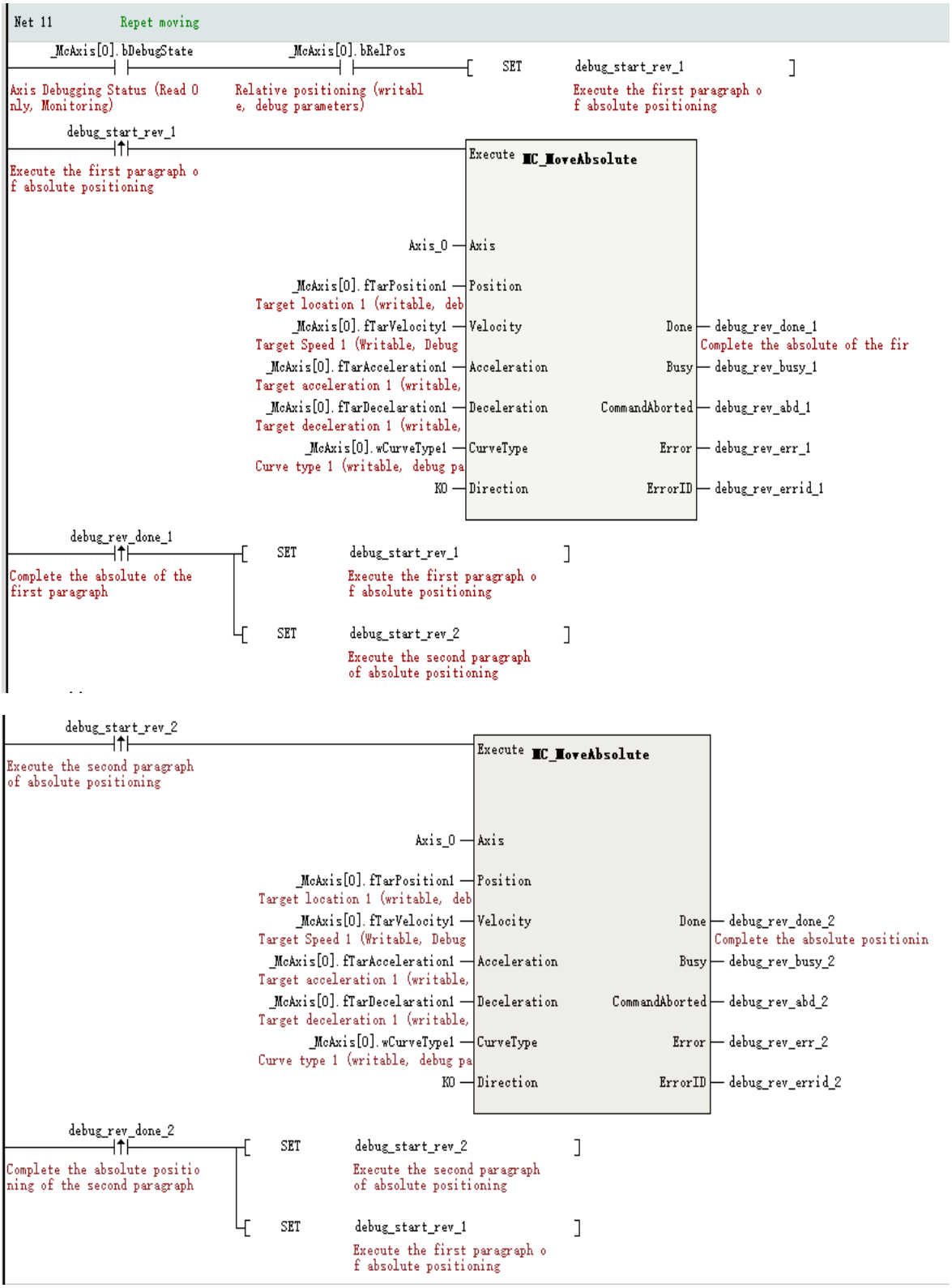
- Continuous motion

  After the control mode is set to continuous motion, you can set four parameters: target velocity, acceleration, deceleration, and curve type. Click the start button to start absolute positioning according to the above parameters. The execution logic is as follows:



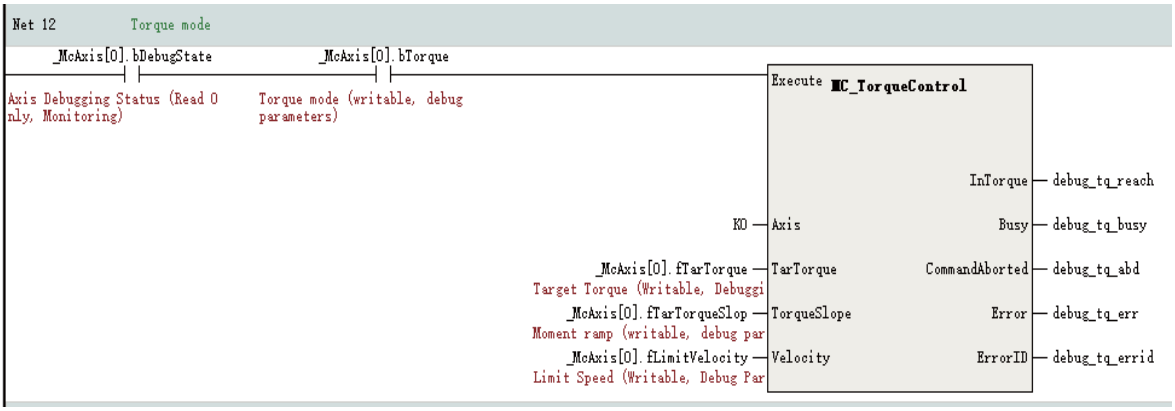
- Reciprocating motion

  After the control mode is set to the reciprocating motion, you can set two sets of target positions, target velocities, acceleration rates, deceleration rates, and curve types. Click the start button, and the axis will first locate the target position 1 according to the set parameters, and then locate the target position 2, followed by reciprocating. The execution logic is as follows:



- Torque mode

  After the control mode is set to the torque mode, you can set three parameters: target torque, torque slope, and limit velocity. Click the start button to start absolute positioning according to the above parameters. The execution logic is as follows:



16.5.3 Rules for Calling Instructions

Motion of axes can be controlled in the PLC through instructions. The rules for calling instructions are as follows.

- Instructions do not need to be instantiated.
- The axis number in the instruction is the unique identifier to access the axis.
- The priority of motion instructions is generally lower than the priority of online commissioning mode.
- Floating-point parameters in the instructions must meet the precision range of floating-point numbers, which generally contain 7 valid digits and can be set to 9999999 at the maximum.

16.5.4 Limit Processing

Two limit detection methods are supported: software limit detection and hardware limit detection.

- To ensure correct processing of hardware limit signals, 0x60Fd must be configured in the process data.
- Software limit processing is only valid for calling position and velocity instructions in linear mode. It is invalid for calling homing and torque instructions.
- If the absolute target position of the called position instruction is within the software limit, the instruction can be executed. If the absolute target position exceeds the software limit, the execution of the positioning instruction is interrupted and the axis stops at the software limit.
- When a velocity instruction is called, if the current velocity of the axis exceeds the software limit, the execution of the velocity instruction is interrupted and the axis stops at the software limit.
- If the axis position has exceeded the positive (negative) limit, the axis can only run in the negative (positive) direction.

The following table lists system variables of limits.

Variable	Function description
bphlimit	Positive limit input status of hardware
bnhlimit	Negative limit input status of hardware
bpslimit	Software positive limit status
bnslimit	Software negative limit status

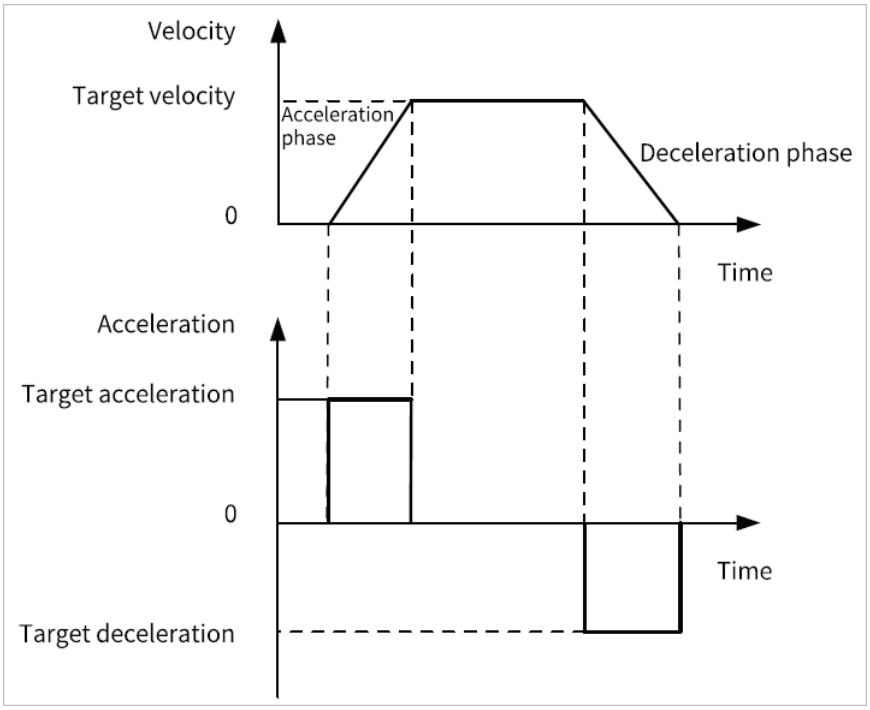
16.5.5 Positioning Curve

Two types of velocity curves are supported: T-shape acceleration/deceleration curve, and 5-segment S-shape acceleration/deceleration curve. The curve type is determined by the CuriveType parameter in the instructions.

When the axis reaches the limit or enters the ErrorStop state and decelerates to stop upon a fault, the axis decelerates to stop according to the T-shape curve.

T-shape velocity curve

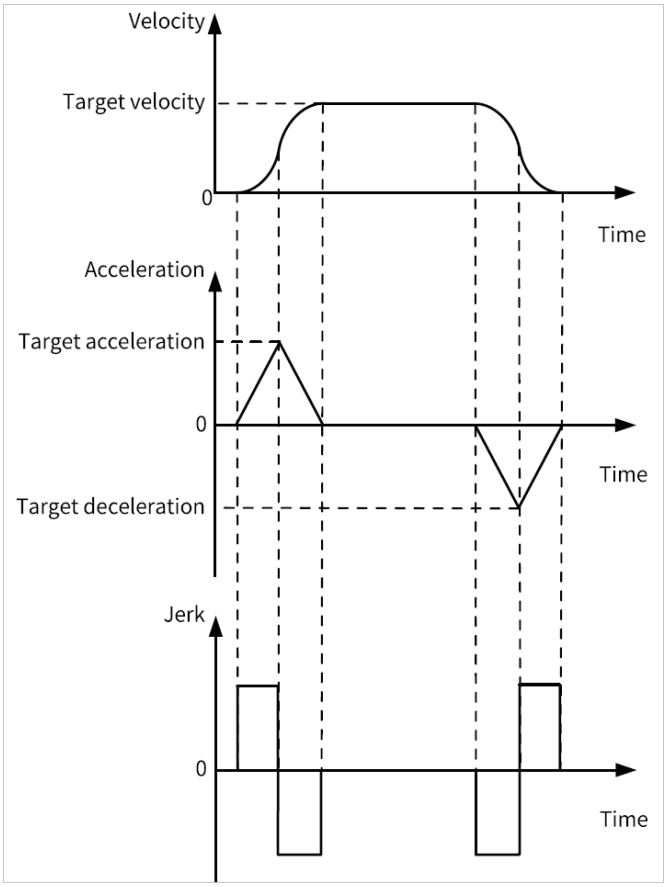
When CuriveType in the instruction is 0, the axis accelerates or decelerates according to the T-shape curve. In the T-shape velocity curve, the axis plans the curve according to the target position, target velocity, target acceleration rate, and target deceleration rate. The actual acceleration/deceleration rate is fixed during acceleration/deceleration. The following figure shows a positioning curve.



- Target position: Indicates the final position of the axis in the absolute positioning instruction, in a user unit, such as degree.
- Target velocity: Indicates the maximum velocity of the axis, in the unit of degree/s (user unit/second).
- Target acceleration rate: Indicates the amount of change in velocity per second when the axis accelerates, in the unit of degree/t2.
- Target deceleration rate: Indicates the amount of change in velocity per second when the axis decelerates, in the unit of degreet2.
- Assume that the axis initial velocity is Vs, target velocity is Vt, and target acceleration rate is Acc, then the acceleration time is:
  Tacc = (Vt – Vs)/Acc
- Assume that the axis initial velocity is Vs, target velocity is Ve, and target deceleration rate is Dec, then the deceleration time is:
  Tdec = (Vs – Ve)/Dec

S-shape velocity curve

When CuriveType in the instruction is 1, the axis accelerates or decelerates according to the S-shape curve. In the 5-segment S-shape curve, the axis plans the curve according to the target position, target velocity, target acceleration rate, and target deceleration rate, of which the target acceleration rate and target deceleration rate refer to the maximum accelerate rate and deceleration rate the axis can reach. The following figure shows a positioning curve.



The 5-segment S-shape velocity curve is divided into 5 stages based on the acceleration rate state: increasing acceleration, decreasing acceleration, constant speed, increasing deceleration, and decreasing deceleration. The constant acceleration and constant deceleration stages do not exist. During variable acceleration stages such as increasing acceleration or increasing deceleration, the actual Jerk is calculated by the PLC and cannot be set by users.

- Target position: Indicates the final position of the axis in the absolute positioning instruction, in a user unit, such as degree.
- Target velocity: Indicates the maximum velocity of the axis, in the unit of degree/s (user unit/second).
- Target acceleration rate: Indicates the amount of change in velocity per second when the axis accelerates at variable velocities, in the unit of degree/2 In the velocity curve, the acceleration rate at the moment (t2) when the stage changes from increasing acceleration to decreasing acceleration is certainly the target acceleration rate.
- Target deceleration rate: Indicates the amount of change in velocity per second when the axis decelerates at variable velocities, in the unit of degree/t2. In the velocity curve, the deceleration rate at the moment (t5) when the stage changes from decreasing acceleration to decreasing deceleration is certainly the target deceleration rate.
- Assume that the axis initial velocity is Vs, target velocity is Vt, and target acceleration rate is Acc, then the acceleration time is:
  Tacc = 2 x (Vt – Vs)/Acc

- Assume that the axis initial velocity is Vs, target velocity is Ve, and target deceleration rate is Dec, then the deceleration time is:
  Tdec = 2 x (Vs – Ve)/Dec

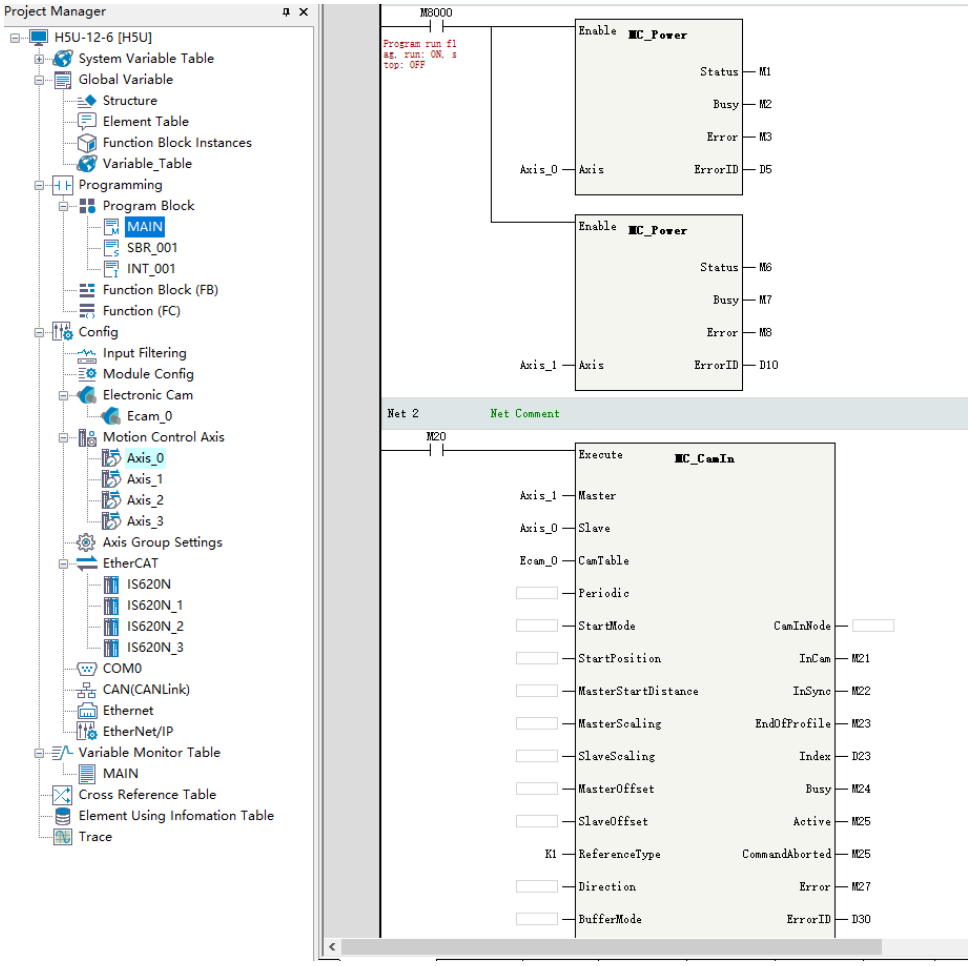
16.6 Dragging a Motion Control Axis

You can directly drag motion control axes to adjust their configuration sequences.

> **NOTE**
>
> This function is available in AutoShop V4.4.0.0 and PCB software V5.0.0.0 or later versions.

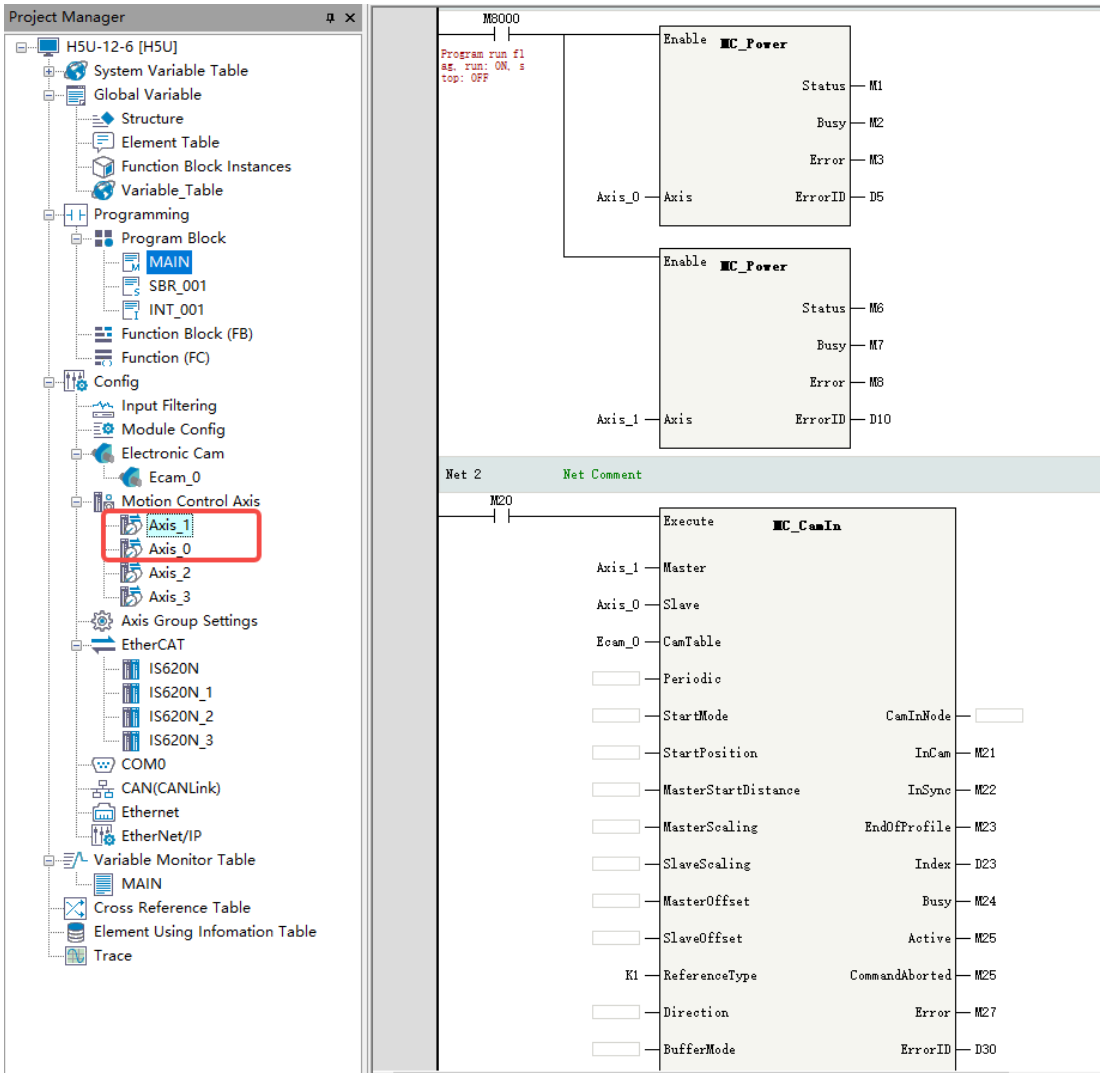
Example

Create a project containing four bus servo axes with Axis_1 as the cam master axis and Axis_0 as the cam slave axis. When the cam position type is 1 (Position set in the current period), an error is reported after the MC_CamIn instruction is called. The fault code is 9251, indicating that the configured master axis address is equal to or greater than the slave axis address. In this case, you can drag motion control axes to adjust their sequences to eliminate the error as follows.



Hover the pointer over Axis_0, hold down the left mouse button, drag Axis_0 to cover Axis_1, and then release the mouse. The configuration positions of Axis_0 and Axis_1 are changed. The servos bound with Axis_0 and Axis_1 remain unchanged.

Download the adjusted configuration to the PLC and call the MC_CamIn instruction again. Then, no error will be reported.



16.7 Modifying Axis Configuration Parameters

You can modify configuration parameters of an axis in the PLC program to meet different application requirements, such as software limit and rotation cycle in ring mode.

Function description

- To modify configuration parameters of an axis in the PLC program, you need to use the system structure variable _sCfgAxis of the axis configuration to set parameters. The setting of this structure is not retentive upon power failure.
- After the PLC is powered on, this structure variable is initialized based on the axis configuration in the software background.
- After modifying the value of the initialized variable based on application requirements, call the MC_SetAxisConfigPara instruction. Then, the setting takes effect.

> **NOTE**
>
> The axis configuration parameters are variables that are not retentive at power failure. After the PLC is restarted, the previously set values are lost. Therefore, set the values every time the PLC is started.

The following table shows the definitions of _sCfgAxis structure variables.

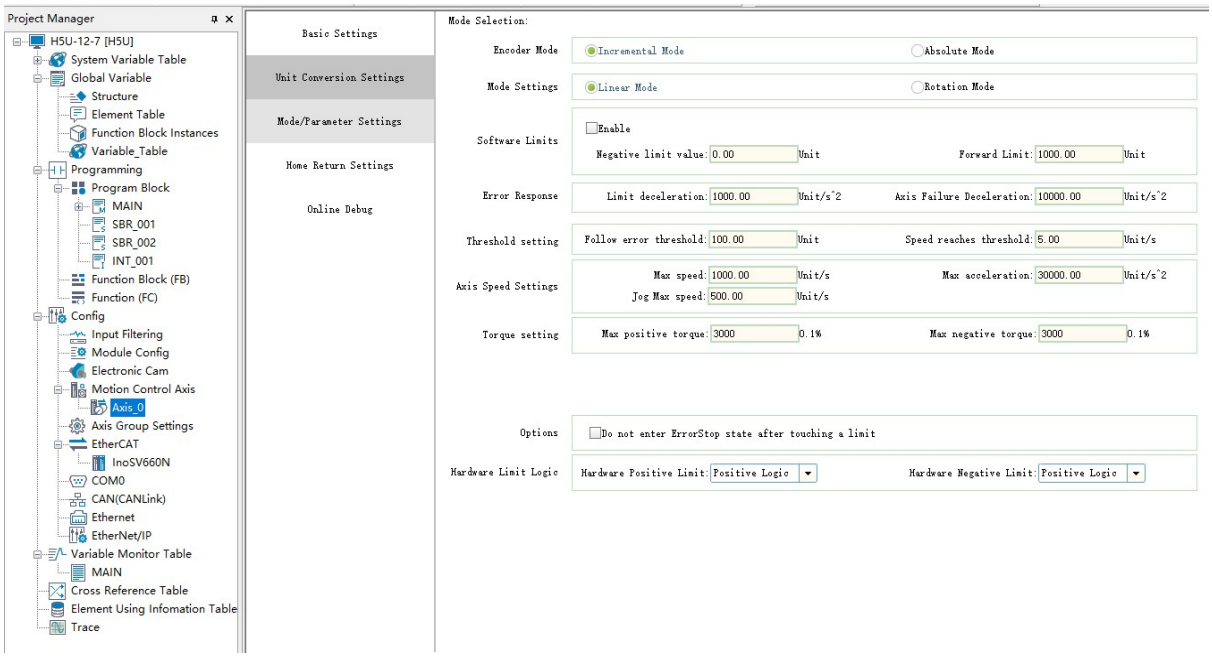
Variable name	Data type	Description
dPlusePreCycle	DINT	Pulses per revolution of the motor/encoder
fDistancePreCycle	REAL	Distance per revolution of the rotary table
dNumerator	DINT	Defines the numerator of the gear ratio.
dDenominator	DINT	Defines the denominator of the gear ratio.
bDirection	BOOL	Direction ON: Forward OFF: Reverse
bVirtualMode	BOOL	Virtual axis mode OFF: Disabled ON: Enabled
bSoftLimitEnable	BOOL	Software limit enable OFF: Disabled ON: Enabled
bEnterErrorStop	BOOL	Not entering ErrorStop state upon an axis fault OFF: Disabled ON: Enabled
bPLimitTerminalPolarity	BOOL	Positive limit polarity OFF: Positive logic ON: Negative logic
bNLimitTerminalPolarity	BOOL	Negative limit polarity OFF: Positive logic ON: Negative logic
bHomeTerminaPolarity	BOOL	Home signal polarity OFF: Positive logic ON: Negative logic
iEncoderMode	INT	Encoder mode (valid for the bus servo axis) 0: Absolute mode 1: Incremental mode
iLineRotateMode	INT	Linear/Rotary mode 0: Linear mode 1: Rotary mode

Variable name	Data type	Description
fPLimit	REAL	Positive limit in linear mode
fNLimit	REAL	Negative limit in linear mode
fRotation	REAL	Rotation period in rotary mode
fLimitDeceleraion	REAL	Limit deceleration
fErrorStopDeceleration	REAL	Deceleration upon axis fault
fFollowErrorWindow	REAL	Following error window
fInVelocityWindow	REAL	Velocity window
fMaxVelocity	REAL	Max. velocity
fMaxJogVelocity	REAL	Maximum jogging velocity
fMaxAcc	REAL	Max. acceleration
fMaxPTorque	REAL	Max. positive torque (bus servo axis) Only used for instruction parameter inspection, not delivered to servo
fMaxNTorque	REAL	Max. negative torque (bus servo axis) Only used for instruction parameter inspection, not delivered to servo
iHomeMethod	INT	Homing mode, local pulse axis Options: 17 to 30 and 35
fHomeVelocity	REAL	Homing velocity, valid for local pulse axis
fHomeApproachVelocity	REAL	Homing approaching velocity, valid for local pulse axis
fHomeAcceleration	REAL	Homing acceleration rate, valid for local pulse axis
dHomeTimeOut	DINT	Homing timeout time, valid for local pulse axis
iHomePositionMode	INT	Homing position mode selection, local pulse axis 0: Absolute homing 1: Relative homing
dTouchProbeID1	DINT	ID of the probe terminal 1 (Modbus address) -2: Disable -1: Reserved
dTouchProbeID2	DINT	ID of the probe terminal 2 (Modbus address)

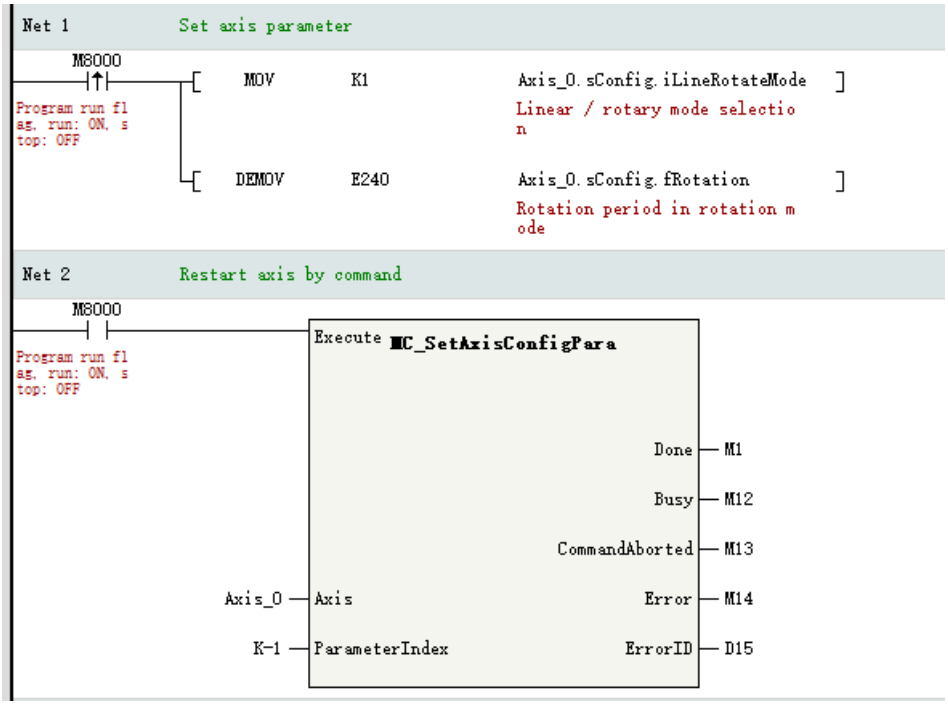
Variable name	Data type	Description
		-2: Disable -1: Reserved
iPluseMethod	INT	Pulse output mode (valid for the local pulse axis) 3: A/B phase 4: Pulse + Direction 5: CW/CCW
dPLimitTerminalID	DINT	ID of the positive limit signal (Modbus address) -2: Disable -1: Reserved
dNLimitTerminalID	DINT	ID of the negative limit signal (Modbus address) -2: Disable -1: Reserved
dHomeTerminalID	DINT	ID of the home signal (Modbus address) Only X0 to X0-X7 are supported. -2: Disable -1: Reserved

Example

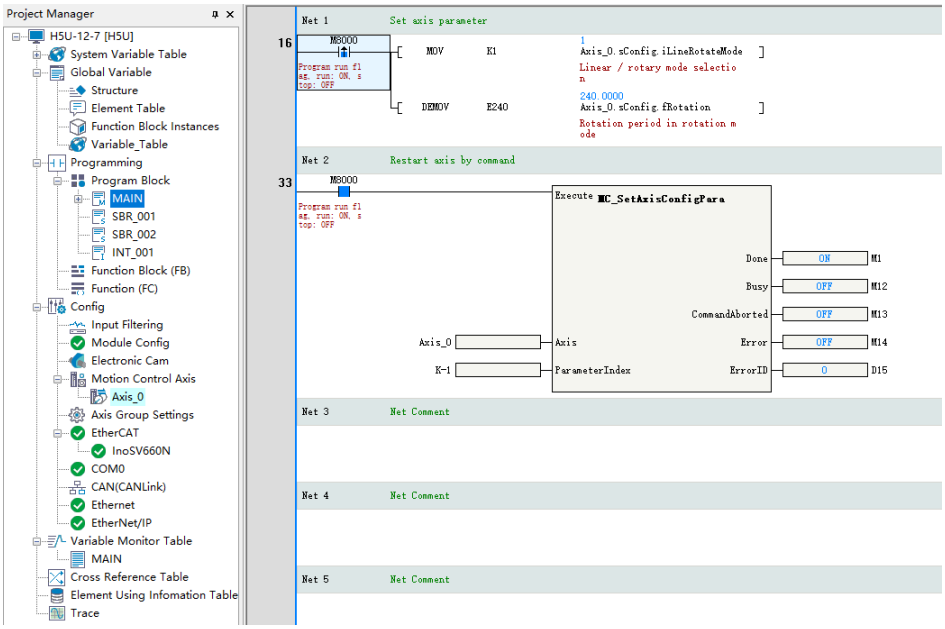
1. Create a network consisting of H5Us and IS620Ns, enable EtherCAT communication, and add a motion control axis Axis_0 with default parameters.



2. Compile the PLC program, modify the mode of Axis_0 to rotary, set the rotation period to 240, and then call the MC_SetAxisConfigPara instruction to make the settings take effect.



3. Connect one IS620N to the EtherCAT network port of the controller, and compile and download the program. The following figure shows the running effect.



16.8 Fault Type

Axis faults are divided into instruction faults, axis faults, and drive faults.

- Instruction faults are faults of MC axis control instructions due to reasons such as inappropriate instruction parameter settings or change to the PLCOpen state machine during axis running. You can obtain the fault codes based on ErrorID in the failed instructions.
- Axis faults are faults of axes, such as excessive following error. You can view axis fault codes in four ways. Method 1: On the “Online Debug” page of the background, view the fault codes in the “Axis error” column. Method 2: Obtain fault codes based on AxisErrorID in the MC_ReadAxisError instruction. Method 3: Obtain fault codes based on wAxisError in the axis system variables. Method 4: View fault codes on the “Fault Diagnosis” page.
- Drive faults are faults of the EtherCAT bus drive or local pulse output axis. To view the fault codes of the EtherCAT bus drive, you must configure 0x603F in the PDO mapping and associate it with the axis. You

can view drive faults in three ways. Method 1: On the “Online Debug” page of the background, view the fault codes in the “Server error” column. Method 2: Obtain fault codes based on ServoErrorID in the MC_ReadAxisError instruction. Method 3: Obtain fault codes based on wServoError in the axis system variables.

17 High-Speed Counter

17.1 About High-Speed Counter

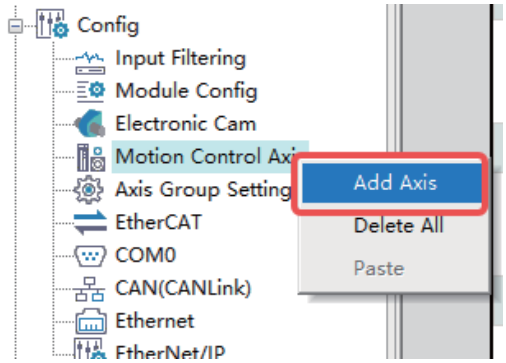
In AutoShop software and engineering applications, a counter is used as an encoder axis to enable management, and the counter associated with an axis is known as a counter axis.

AutoShop supports 4-axis 32-bit high-speed counters, which can realize phase AB frequency multiplication by 1/2/4, CW/CCW, pulse+direction, and single-phase counting. Counting signal sources are external pulse input or internal 1 ms/1 μs clock counting, which can be used to preset and latch the counter in combination with other input signals.

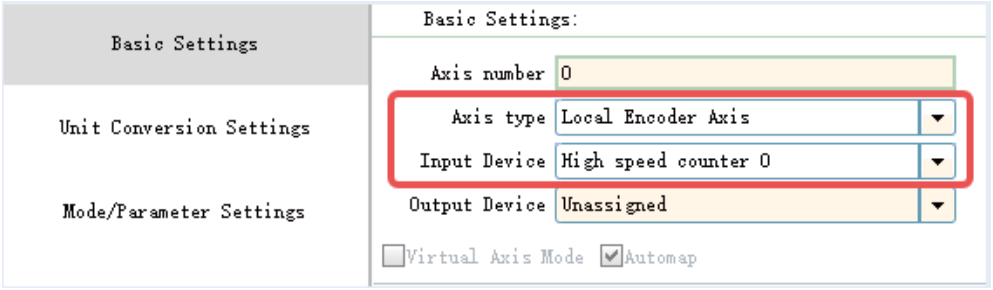
17.2 Creating a Counter Axis

Before using the counter in AutoShop, associate the counter with the axis first. .

1. In the Project Manager, go to Config, right-click Motion Control Axis, and select Add axis to create a motion control axis.

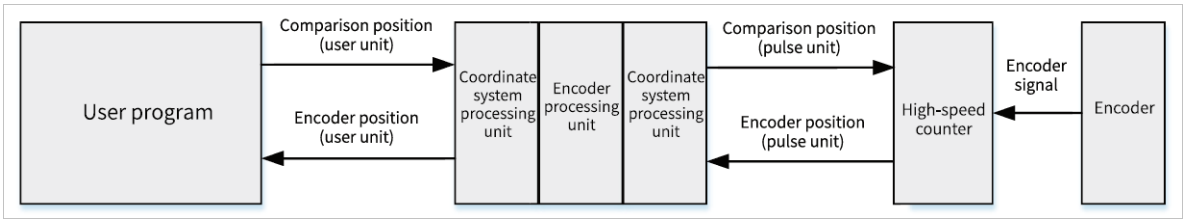


2. Double-click the newly added axis (see Figure Axis_0) to open the setting page. In the Basic Settings interface, select the local encoder axis as the axis type and high-speed counter as the input device to associate the axis with the counter. The axis number is used as the axis ID in the program to control corresponding counter axis.



17.3 User Units and Unit Conversion of Counter Axis

High-speed counters use pulse units when decoding encoder signals, and use common measurement units for counter instructions such as millimeters, degrees, or inches, which are called user units. Through unit conversion, the number of pulses can be converted into user units, which can be defined as equipment-related units (such as millimeters and revolutions) according to actual applications.



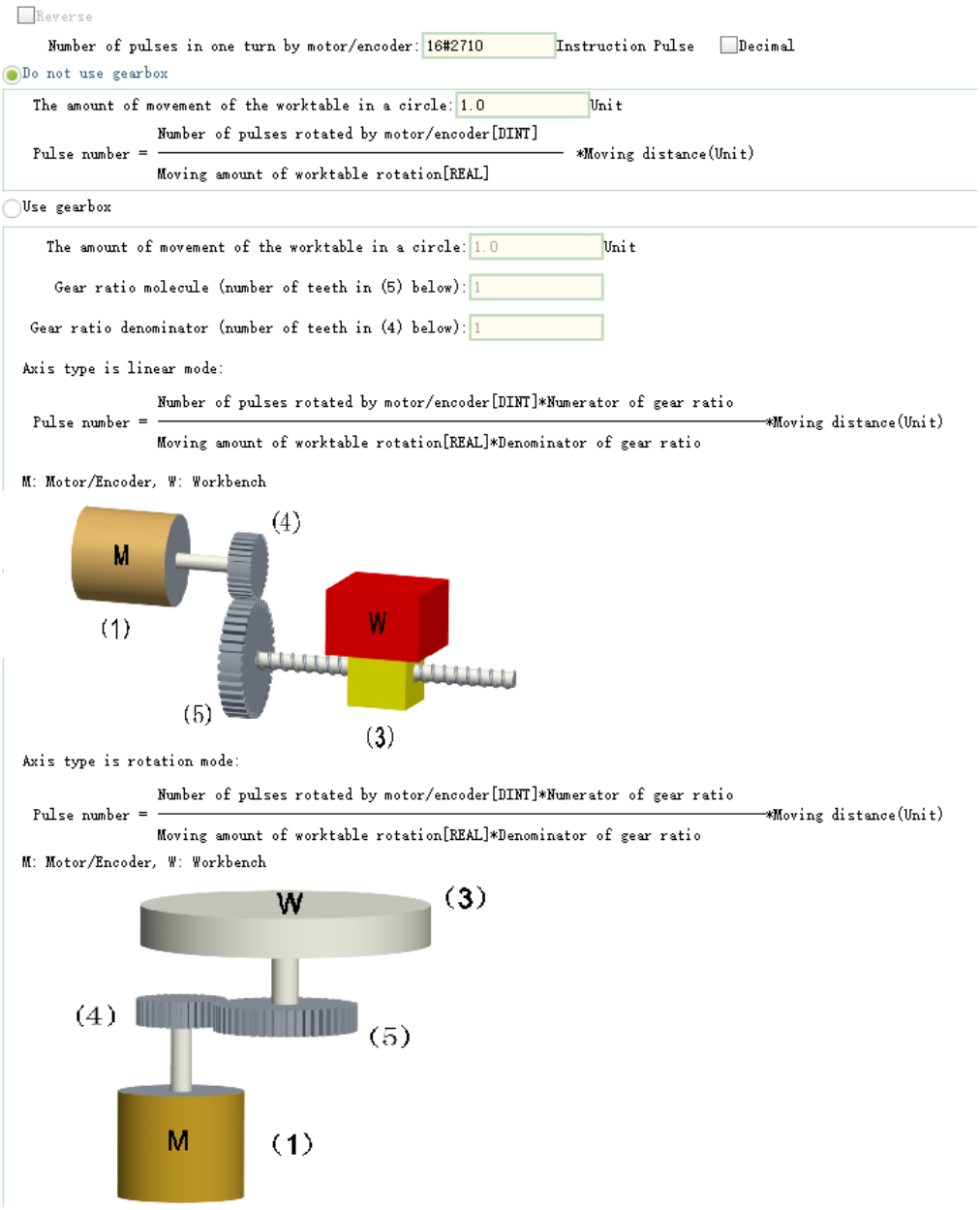
The following table lists parameters that need to be set for unit conversion.

Parameter Name	Function
Number of pulses per revolution of the motor/ encoder	Set the number of pulses required for the motor to rotate one turn according to the encoder resolution.
Gear change mechanisms in use or not	Specify whether gear change mechanisms are in use or not.
Amount of movement per revolution of the motor/ encoder	The workpiece movement amount per revolution of the motor when no gear change mechanism is in use
Amount of movement per revolution of the worktable	The workpiece movement amount per revolution of the worktable when gear change mechanisms are in use
Gear ratio numerator	Set a gear ratio on the workpiece side.
Gear ratio denominator	Set a gear ratio on the motor side.

When a servo motor is connected to a screw rod through a reducer to drive the worktable and gives feedback about the worktable position to an encoder counter through the PLC, if the counter counts the encoder in pulses, the number of pulses shall be the unit, and if the counter axis is used to represent the worktable position, then millimeter shall be the unit.

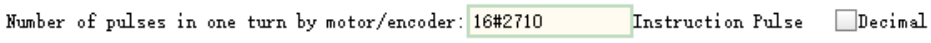
Therefore, in the program, the user unit "degree" is used uniformly as the user unit of the counter axis.

The conversion rule between the user unit "degree" and the pulse is as follows:



In the unit conversion setting, set the relevant parameters according to the actual device.

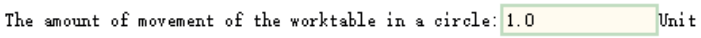
1. Number of pulses per revolution of the motor/encoder: "16#" in the input box indicates that hexadecimal numbers are used.
   If the number of pulses per revolution of the encoder is 10000, whose hexadecimal equivalent is 2710, input 16#2710.

2. Working stroke setting: The working stroke can be set with or without gear change mechanisms.
   - Do not use gearbox.
     When the gear change mechanisms are not in use, the conversion equation from user unit "degree" to pulse unit is as follows.

     $$\text{Number of pulses (unit: pulse)} = \frac{\text{Number of pulses per revolution of the motor/encoder [DINT]}}{\text{Distance per revolution of the workbench [REAL]}} \times \text{Distance (unit: Unit)}$$

     If one revolution of the encoder corresponds to one revolution of the working axis and the user unit is degree, then the working stroke of the motor/encoder per revolution shall be set to 1.



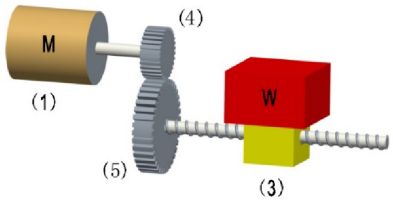
Take the Inovance 20-bit encoder as an example. The set parameters are as follows.

Number of pulses per revolution of the motor/encoder = 1048576

Amount of movement per revolution of the motor/encoder = 1

Then, when the target displacement given by the relative positioning instruction is 10, the actual number of pulses sent by the motion control axis is 10485760, then the motor rotates by 10 revolutions.

- With gear change mechanisms
  Typical working condition in linear mode is shown in the following figure.

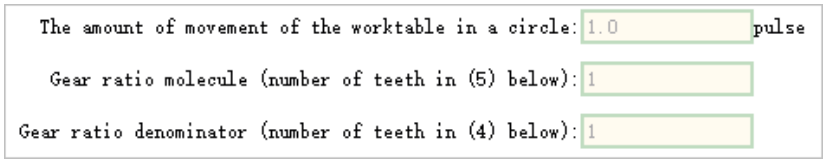


In the figure, (1) is the servo motor, (3) is the workpiece, (4) is the gear ratio numerator, and (5) is the gear ratio denominator.

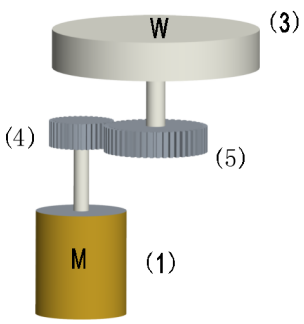
The calculation equation from user unit "degree" to pulse unit is as follows.

$$\text{Number of pulses (unit: pulse)} = \frac{\text{Number of pulses per revolution of the motor/encoder [DINT]} \times \text{Gear ratio numerator [DINT]}}{\text{Distance per revolution of the workbench [REAL]} \times \text{Gear ratio denominator [DINT]}} \times \text{Distance (unit: Unit)}$$

If a servo motor is connected to a screw rod through a reducer to drive the worktable, the working stroke per revolution of the screw rod is 1 degree, and the reduction ratio is 1:1. The setting is as follows.



Typical working condition in rotary mode is shown in the following figure.



In the figure, (1) is the servo motor, (3) is the workpiece, (4) is the gear ratio numerator, and (5) is the gear ratio denominator.

The calculation equation from user unit "degree" to pulse unit is as follows.

$$\text{Number of pulses (unit: pulse)} = \frac{\text{Number of pulses per revolution of the motor/encoder [DINT]} \times \text{Gear ratio numerator [DINT]}}{\text{Distance per revolution of the workbench [REAL]} \times \text{Gear ratio denominator [DINT]}} \times \text{Distance (unit: Unit)}$$

17.4 Work Mode Setting

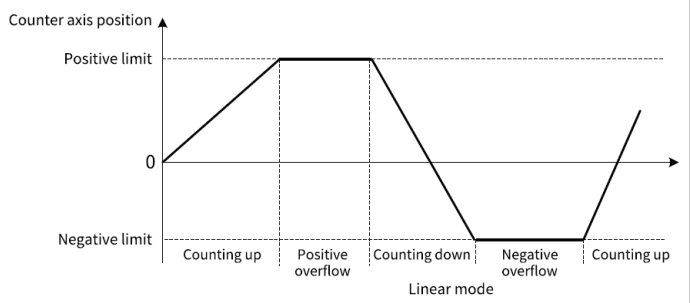
17.4.1 Linear Mode

The counter axis moves between the negative and positive limits. After the counter axis reaches the limits, pulses in the same direction are still input; the counter axis reports an overflow while the counter axis position remains unchanged. After the counter axis reports an overflow, input the reverse pulses. The counter axis counts in reverse, and the overflow error is removed.

In linear mode, you can set the negative and positive position limits of the counter axis in the interface, with user unit (Unit) as the position unit. The negative limit must be less than or equal to 0, and the positive limit must be greater than or equal to 0.

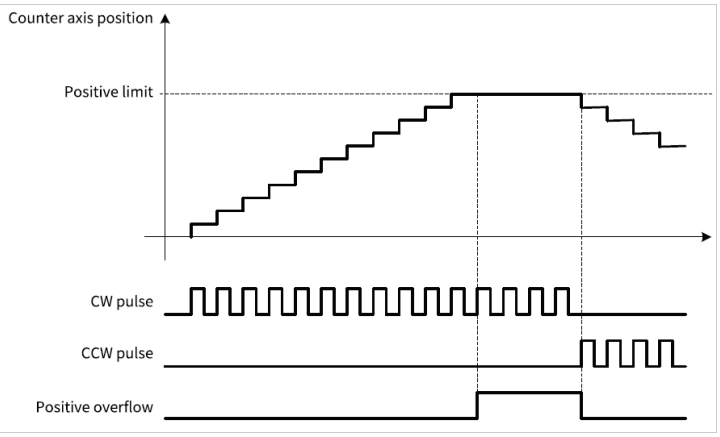
Since the high-speed counter is a 32-bit counter, the negative and positive limits must be within the 32-bit integer range [-2147483648, +2147483647] after being converted to pulse units.

In linear mode, the high-speed counter operates in the closed interval of [negative limit, positive limit]. When the direction is negative, the count value decreases in the negative direction. After the negative limit is reached, the count value no longer decreases. When the direction is positive, the count value increases in the positive direction. After the positive limit is reached, the count value no longer increases.



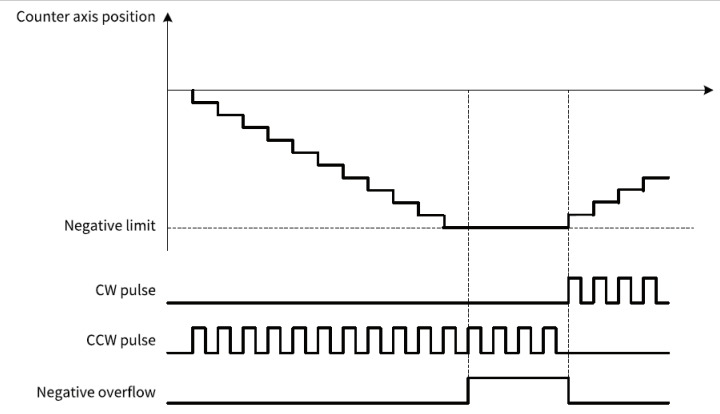
CW pulse counting

In linear mode, input CW pulses. After the incremental count of the counter axis position reaches the limit, keep inputting CW pulses. The counter axis reports a positive overflow error, and the counter axis position value remains unchanged. Input CCW pulses. The counter axis position counts down, and positive overflow error is removed.



CCW pulse counting

In linear mode, input CCW pulses. After the decremental count of the counter axis position reaches the limit, keep inputting CCW pulses. The counter axis reports a negative overflow error, and the counter axis position value remains unchanged. Input CW pulses. The counter axis position counts up, and the negative overflow error is removed.

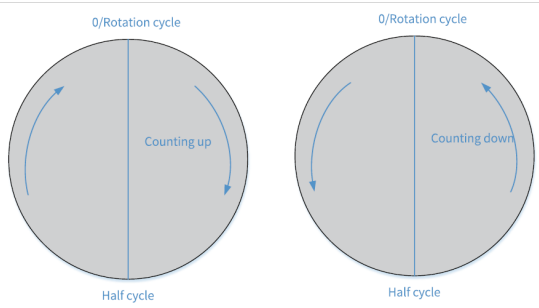


17.4.2 Rotary Mode

The position of the counter axis changes cyclically during the rotation cycle. In case of incremental count, the counter axis position turns 0 after reaching the maximum value in the rotation cycle; in case of decremental count, the counter axis position decreases from the maximum value in the rotation cycle after turning 0.

In rotary mode, you can set the rotation cycle of the counter axis in the interface, with user unit (Unit) as the cycle unit.

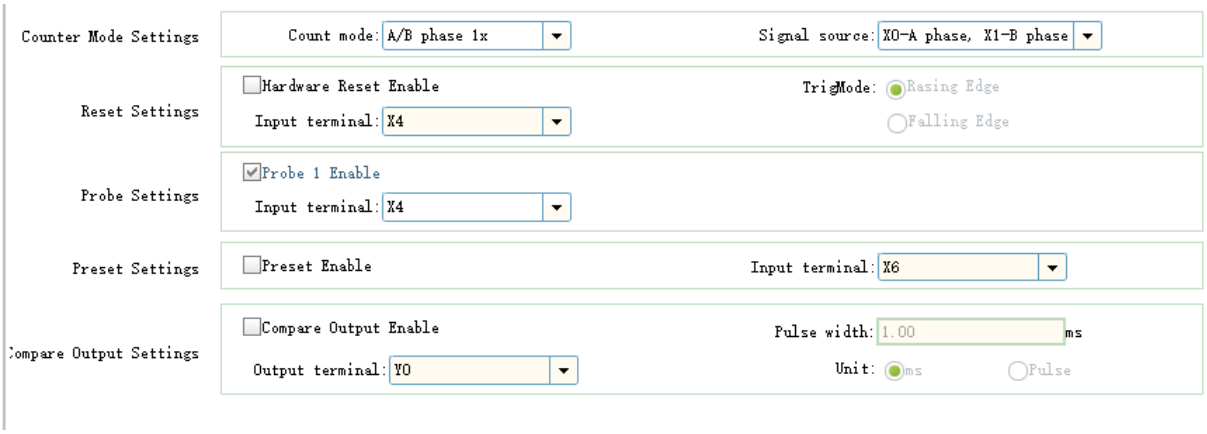
Since the high-speed counter is a 32-bit counter, the rotation cycle must be within the 32-bit integer range [-2147483648, +2147483647] after being converted to pulse units.



17.5 Counter Parameter Setting

17.5.1 Overview

Parameter settings mainly involve the count mode, resetting, probes, presetting, and output comparison features.



17.5.2 Counting Modes

17.5.2.1 Overview

Local encoder axes support multiple signal counting modes, including phase A/B (frequency multiplication by 1/2/4), CW/CCW, pulse+direction, and single-phase counting.

Signal sources: Different signal sources can be selected depending on the counting mode.

The supported input ports of signal sources in various counting modes are shown in the following table. One input port of signal source can be selected for different local encoder axes.

Port Mode	X0	X1	X2	X3	Internal 1 ms	Internal 1 μs
Phase A/B	Phase A	Phase B	Phase A	Phase B	×	×
CW/CCW	CW	CCW	CW	CCW	×	×
Pulse+direction	Pulse	Direction	Pulse	Direction	×	×
Single-phase counting	Pulse	Pulse	Pulse	Pulse	Pulse	Pulse

NOTE

When two input signals are required in the selected working mode, X0 and X1 make up one group of input signals, while X2 and X3 make up the other group.

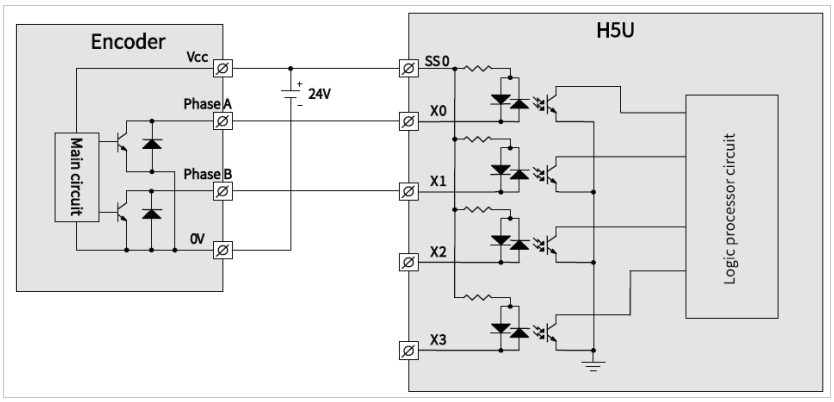
The above counting modes and signal sources can be arbitrarily selected for the four counters, and repeatedly used for different counters.

17.5.2.2 Phase A/B Mode

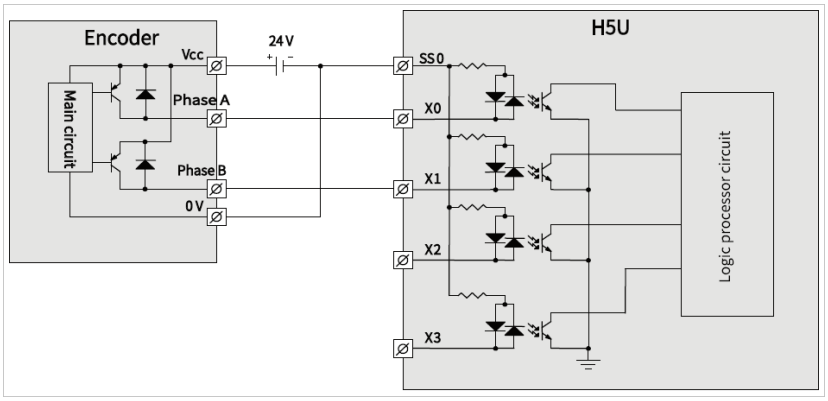
In phase A/B mode, the encoder generates two orthogonal phase pulse signals with a phase difference of 90°, that is, phase A signal and phase B signal. When the phase A signal leads the phase B signal, the counter counts up; when the phase B signal leads the phase A signal, the counter counts down.

Phase A/B encoder wiring diagram

Sink input wiring

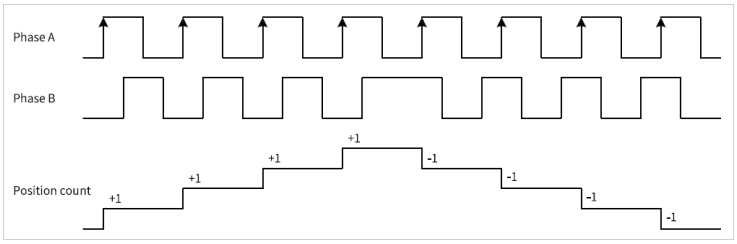


Source input wiring

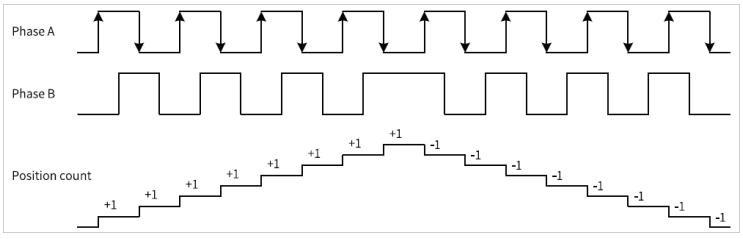


The phase A/B pulse can be set to operate in frequency multiplication by 1, 2, or 4.

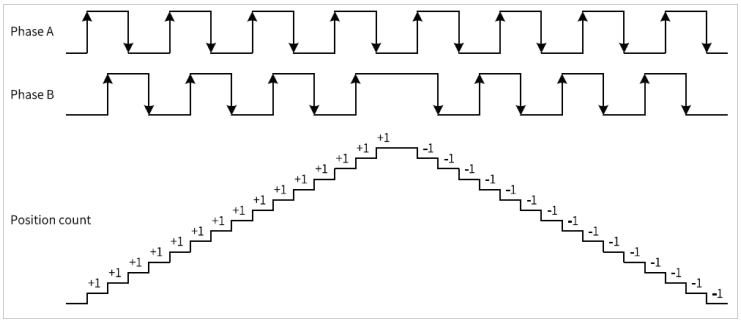
- In the mode of phase A/B frequency multiplication by 1, only the rising edge of phase A pulse is counted as shown in the following figure.



- In the mode of phase A/B frequency multiplication by 2, the rising and falling edges of phase A pulse are counted as shown in the following figure.

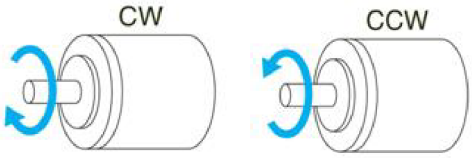


- In the mode of phase A/B frequency multiplication by 4, the rising and falling edges of the phase A pulse and the phase B pulse are counted as shown in the following figure.

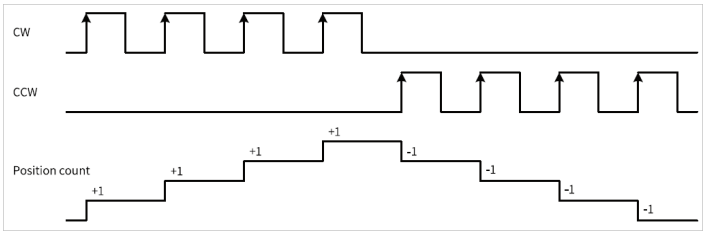


17.5.2.3 CW/CCW Mode

Clock Wise (CW) is the forward pulse signal, and Counter Clock Wise (CCW) is the reverse pulse signal. When the encoder is forward running, CW pulse signals are output; when the encoder is reverse running, CCW pulse signals are output.

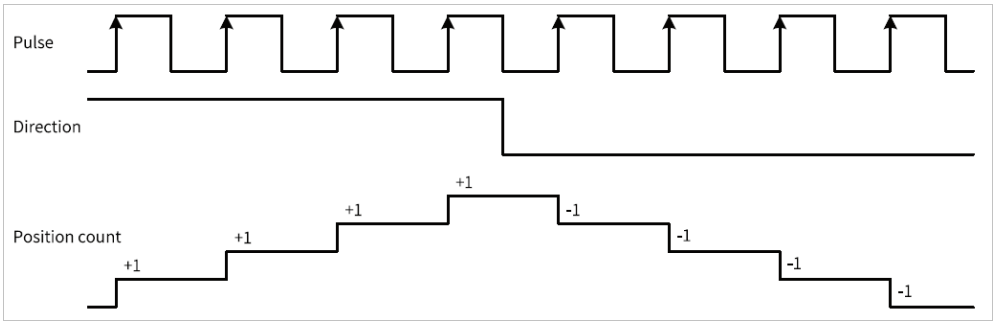


When the local encoder axis operates in this count mode, the high-speed counter counts up the CW signals and counts down the CCW signals, as shown in the following figure.



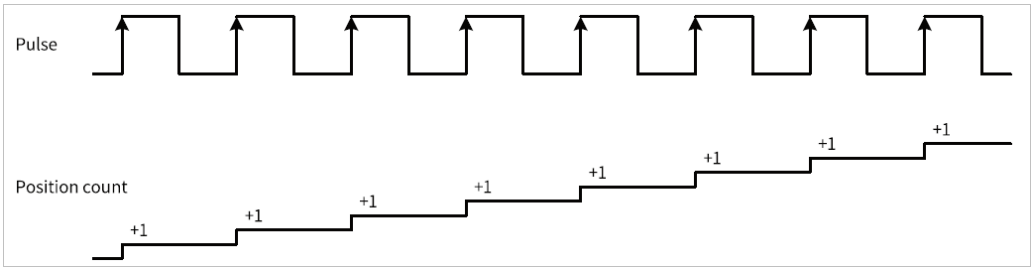
17.5.2.4 Pulse & Direction Mode

In this mode, when the direction signal is ON, the high-speed counter counts up the pulse signals; when the direction signal is OFF, the high-speed counter counts down the pulse signals, as shown in the following figure.



17.5.2.5 Single Phase Counting

In this mode, the high-speed counter counts up the pulse signals. The position count is added with 1 when the rising edge of a pulse is input.



17.5.3 Reset Setting

Enable the reset encoder axis position of external inputs by ticking "Hardware Reset Enable". The input terminal can be any of X0 to X7 inputs, with rising edge or falling edge as the trigger condition.

After the reset feature is enabled, the encoder axis position can be reset to 0 through external inputs.

17.5.4 Probe Terminal Setting

Each counter supports two external inputs to latch the current value of the counter to realize the probe function. Enable the counter axis position latch of the external inputs by ticking the "Touch probe enable". The input terminal can be any of X0 to X7 inputs.

When the probe is enabled, read the probe position of the counter axis through the function block instruction HC_TouchProbe.

NOTE

For the probe functions supported by the Easy series models, see the specific model for more information.

Series	Local encoder axis	Local pulse axis
Easy301 series	Each axis has only two probes.	Dual-axis mode: 2 probes Single-axis mode: Y0, Y1, Y2, and Y3 support two probes
Other models of the Easy series	Each axis has only one probe.	Dual-axis mode: 2 probes Single-axis mode: Probes are not supported

17.5.5 Preset Terminal Setting

Enable the preset counter value of external inputs by ticking "Preset Enable". The input terminal can be any of X0 to X7 inputs, with rising edge or falling edge as the trigger mode.

When the preset function is enabled, the encoder axis position is preset by external inputs through the function block instruction HC_Preset.

17.5.6 Compare Output Terminal Setting

After "Compare Output Enable" is selected, the hardware output can be realized when the positions are equal in comparison without software processing, featuring high real-time performance and microsecond-level output responses.

- After the comparison output function is enabled, in combination with function block instructions, the output controlled by the hardware circuit when the positions are equal in comparison is ON. The output terminal can be any of Y0 to Y3, and the pulse width when the output is ON can be measured in time units or user units.
- Each local encoder axis is equipped with one comparison output function, and the input terminal and output pulse width can be configured as needed.
- After configuration, the axis position comparison output is realized through the function block instructions HC_Compare, HC_ArrayCompare, and HC_StepCompare.
- When "ms" is selected as the unit, the time range for setting is 0.1 ms to 6553.5 ms. When the unit is pulse, the set value must fall between 1 and 65535 after being converted into pulse units.

The comparison output is directly output through the hardware control port, instead of software processing. Therefore, status of comparison output is not available in the Y element in the program. The Y soft element and the comparison output control the output port in an OR relationship. If the Y element is continuously controlled to ON, the actual port output remains ON.

17.6 Counter Axis Instruction Application (H5U)

17.6.1 Overview

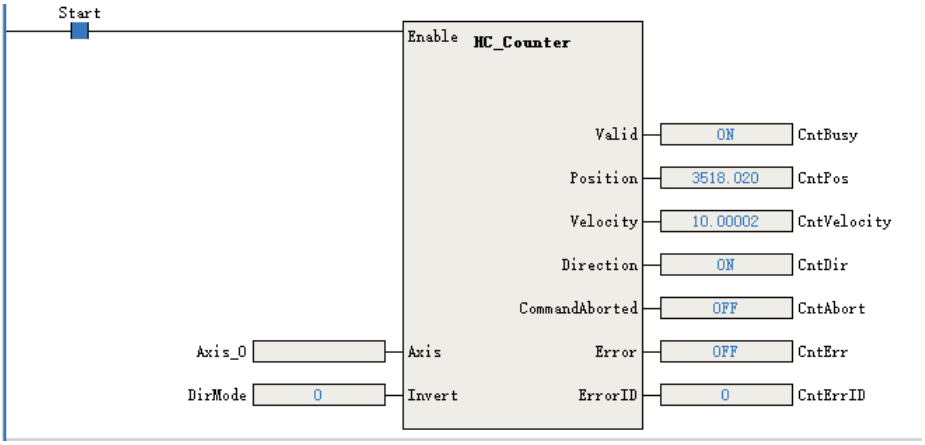
After a counter axis is set in AutoShop, the counter axis can be used in combination with function block instructions to implement features such as axis position counting/speed measurement, axis position presetting, and axis position latching and comparison.

17.6.2 Instruction for Axis Position Counting and Velocity Measurement

The HC_Counter instruction implements position counting and velocity measurement of the counter axis.

The counter axis position varies within a certain range based on the mode setting, with user unit (Unit) as the position unit.

The counter axis velocity is the current real-time velocity (unit: Unit/s). The minimum velocity that can be measured by the counter axis is the velocity corresponding to one pulse of the counter within 1s. If one pulse of the counter corresponds to 0.01 Unit, the minimum velocity that can be measured is 0.01 Unit/s.



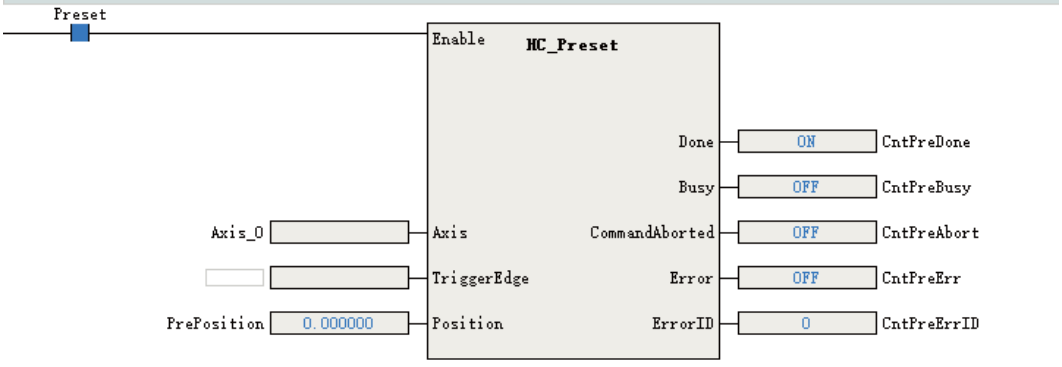
The parameter Invert in the instruction can be set to change the count direction. Modification on Invert takes effect only after this function block instruction is enabled again.

The following table shows the relationship between the Invert setting and the counting direction.

Invert	Phase A/B	Pulse + Direction	CW/CCW	Single-phase counter
0	Phase A leading phase B, counting up Phase B leading phase A, counting down	Direction signal = OFF, counting down Direction signal = ON, counting up	Phase A, counting up Phase B, counting down	Counting up
1	Phase A leading phase B, counting down Phase B leading phase A, counting up	Direction signal = OFF, counting up Direction signal = ON, counting down	Phase A, counting down Phase B, counting up	Counting down

17.6.3 Axis Position Preset Instruction

The HC_Preset instruction assigns the counter axis position according to the preset condition.



The preset condition TriggerType can be set to the trigger by the rising edge of the instruction or by external X input.

TriggerType	Definition
0	Triggered on the rising edge of the instruction flow

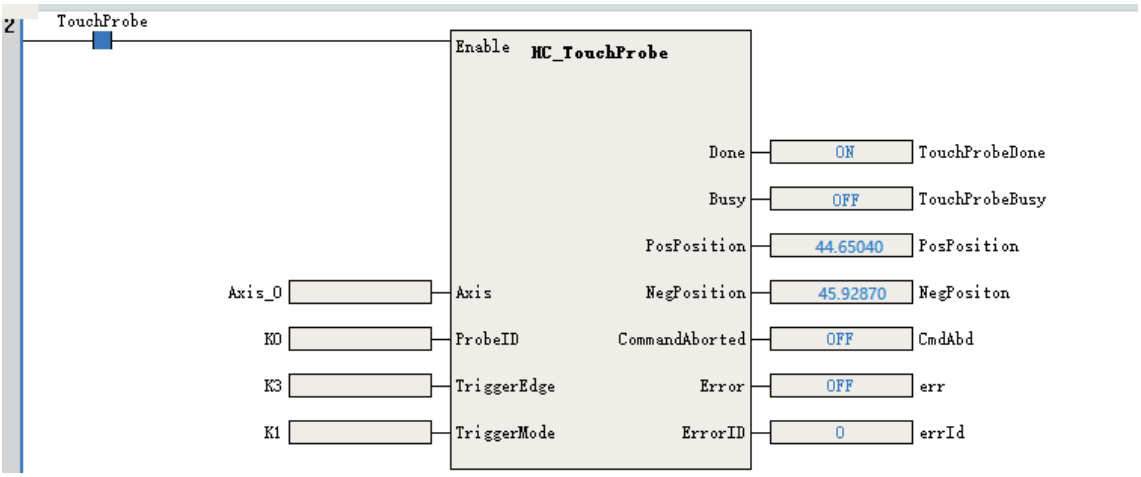
TriggerType	Definition
1	Triggered on the rising edge of the external input X
2	Triggered on the falling edge of the external input X
3	Triggered on the rising or falling edge of the external input X

When the preset condition is set to the trigger by external X input, you need to tick the preset function in counter parameter settings and select the input terminal and trigger mode. The input terminal can be any of X0 to X7, with rising edge or falling edge as the optional trigger mode.

17.6.4 Probe Instruction

The HC_TouchProbe instruction can latch the counter axis position value when the external input trigger mode is active.

Each counter axis supports two probes. During use, you need to tick the corresponding probe function in counter parameter settings and select the input terminal and trigger mode. The input terminal can be any of X0 to X7.



The parameter ProbeID specifies the number of the probe used by the counter.

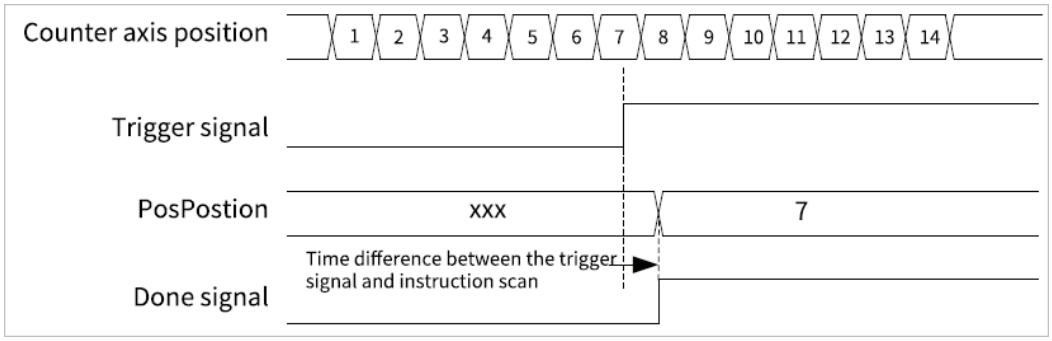
ProbeID	Definition
0	Indicates that probe 1 is used.
1	Indicates that probe 2 is used.

TriggerEdge specifies the probe trigger edge. The rising edge trigger position is latched in the output parameter PosPosition, and the falling edge trigger position is latched in the output parameter NegPosition.

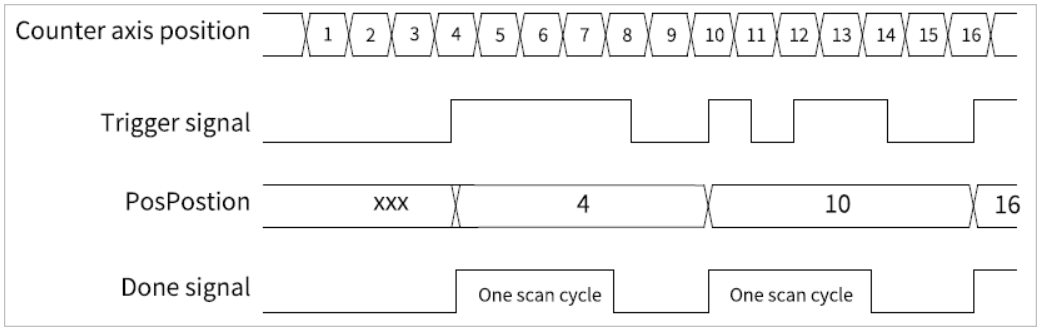
TriggerEdge	Definition
1	Triggered on the rising edge of the external input X
2	Triggered on the falling edge of the external input X
3	Triggered on the rising or falling edge of the external input X

TriggerMode in the instruction can be set to the single trigger or continuous trigger mode.

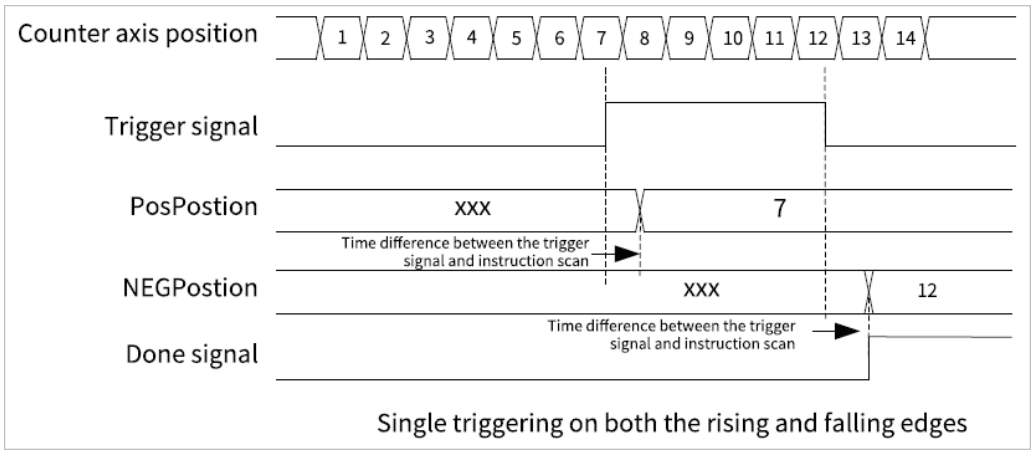
- If the single trigger mode is used, when the function block instruction flow and the external input trigger mode are active, the counter axis position is latched once, and the Done signal is output.
  The counter axis position is latched in real time based on the trigger edge, which is not affected by program execution. During instruction execution, affected by the scan cycle, when the program scans and executes to the latched instruction, it updates the latched position to the output parameter of the instruction.

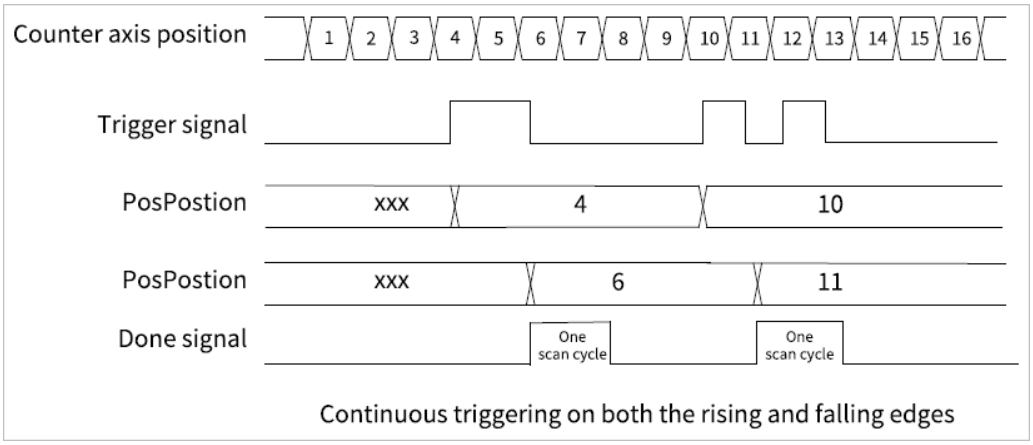


- If the continuous trigger mode is used, when the function block instruction flow and the external input trigger mode are active, the counter axis position is latched, and the Done signal that is active for one scan cycle is output. When the Done signal becomes OFF and the external input trigger mode is active, the counter axis position continues to be latched and the Done signal that is active for one scan cycle is output. During the scan cycle in which the Done signal is active, if the external input trigger mode is active, the counter axis position is not latched at this time.



- When the dual-edge trigger mode is used, the Done signal is output after the instruction is triggered on both the rising and falling edges to complete the latch. In single trigger mode, the Done signal remains active until the instruction execution is completed; in continuous trigger mode, the Done signal is active for one scan cycle, and the latch signal is not responded within the scan cycle when the Done signal is active.



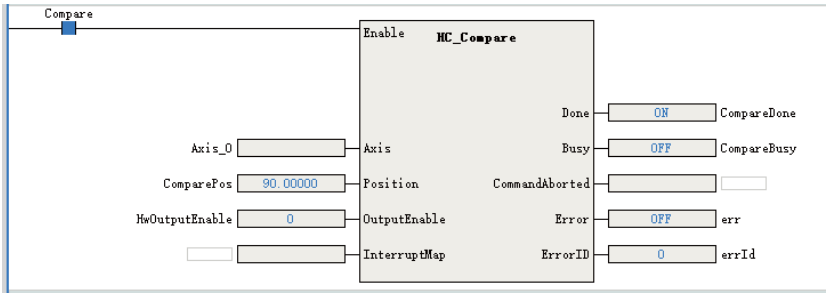


17.6.5 Comparison Instructions

The instructions HC_Compare, HC_StepCompare, and HC_ArrayCompare can compare the counter axis position with a single position, equally-spaced positions continuously, or multiple positions continuously.

HC_Compare

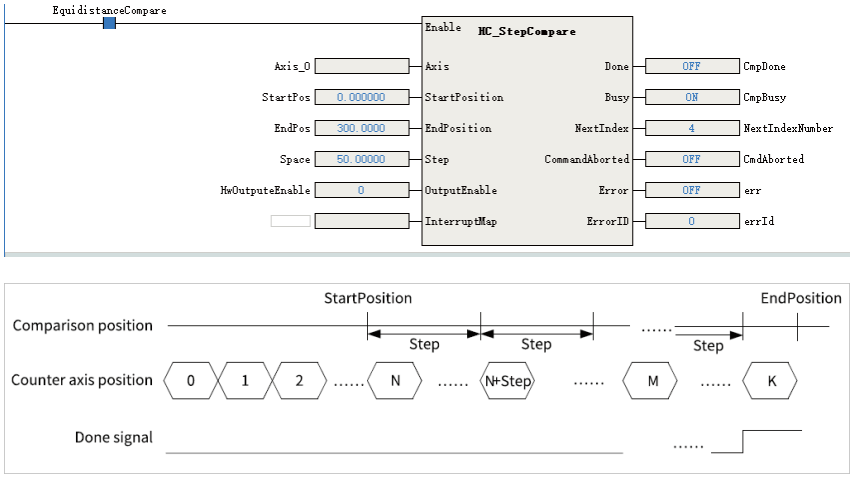
The instruction compares the counter axis with a single position. When the instruction flow is active, the Done signal is output after the counter axis position reaches the comparison position.



HC_StepCompare

This instruction compares the counter axis position with equally-spaced positions continuously. When the instruction flow is active, the counter axis position is compared with the position specified by StartPosition. When they are equal, the comparison position increases or decreases by a value specified by Step and then is compared with the counter axis position. The Done signal of one cycle is not output after they are equal in each comparison, but after the last comparison position is compared.

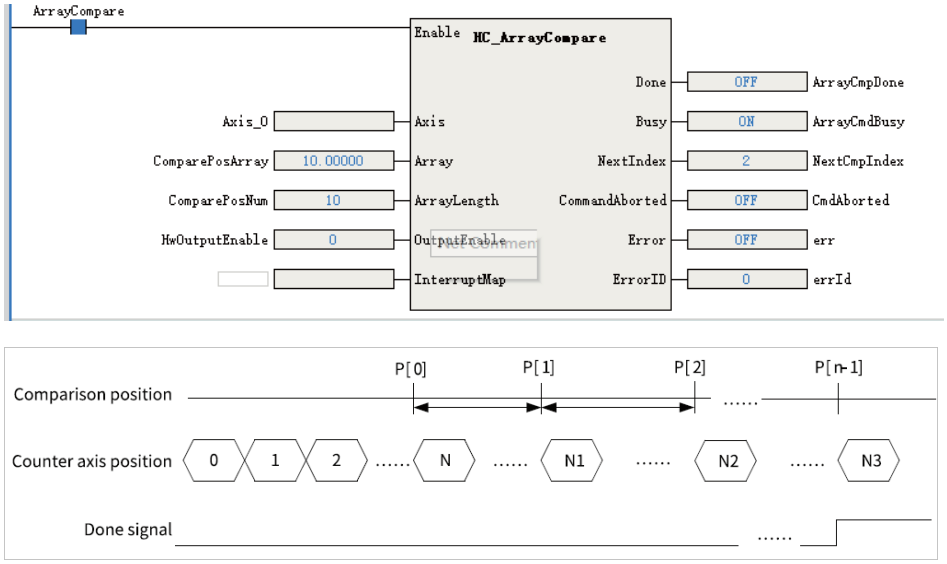
- If the StartPosition is less than the EndPostion, when the positions are equal in each comparison, the comparison position increases by a value specified by Step. When the current comparison position added with the value specified by Step is greater than the EndPosition, the current comparison position is the last one.
- If the StartPosition is greater than the EndPostion, when the positions are equal in each comparison, the comparison position decreases by a value specified by Step. When the current comparison position minus the value specified by Step is less than the EndPosition, the current comparison position is the last one.
- The output parameter NextIndex indicates the index of the next comparison point, that is, the number of completed comparison points.



HC_ArrayCompare

This instruction compares the counter axis position with multiple positions in an array continuously. When the instruction flow is active, the counter axis position is compared with the first position in the array. If they are equal, the counter axis position is compared with the next position value in the array. After the last comparison position is compared, the Done signal is output.

- ArrayLength in the instruction specifies the array length. After all the positions in the array are compared, the Done signal is continuously output and the comparison with multiple positions is completed.
- The output parameter NextIndex indicates the index of the next comparison point, that is, the number of completed comparison points.



17.6.6 High-Speed Hardware Comparison Output

The counter axis can realize the position comparison hardware output. When the counter axis and the comparison position are equal, the output directly controlled by the hardware circuit is ON, and the output delay is less than 1 μs.

Setting the comparison output function of the counter axis

On the parameter setting page of the counter axis, select "Compare Output Enable".

The output terminal can be any of Y0 to Y3. The output width is set as the pulse width when the output is ON, for which the unit can be time unit or user unit.

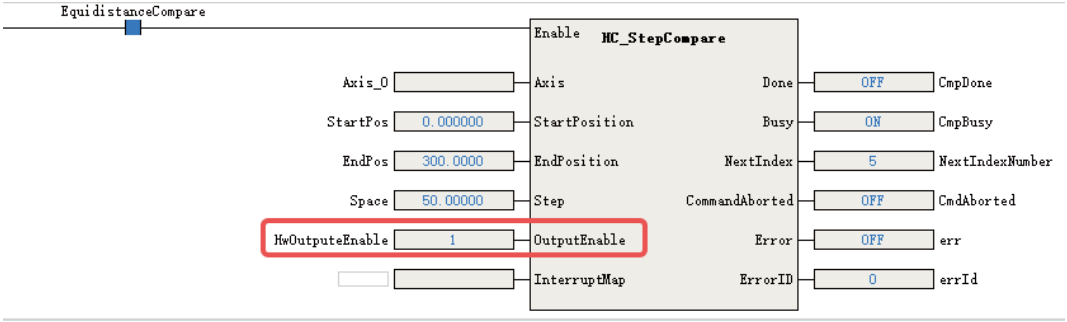
NOTE

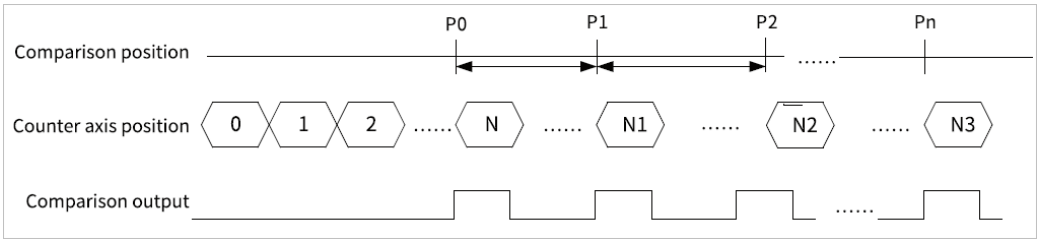
When the unit of pulse output width is set to time, the time accuracy of pulse output width is 100 μs, and the maximum output width is 6500 ms. When the unit of pulse output width is set to pulse unit, the maximum output width is equivalent to 65535 pulses.

Enabling the OutputEnable parameter in the comparison instructions

Use HC_Compare, HC_StepCompare, and HC_ArrayCompare comparison instructions to set OutputEnable to 1, that is, associate hardware outputs when the instruction executes the comparison for equality.

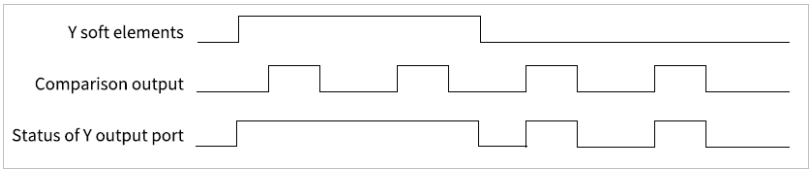
When the instruction executes the comparison for equality, the output terminal set directly through the control of hardware circuit is ON, and the output turns OFF after the continuous width output.





NOTE

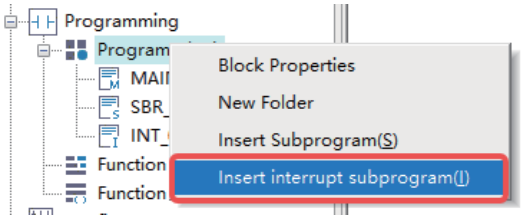
High-speed comparison hardware output is directly output through the hardware control port. Therefore, status of comparison output is not available in the Y element in the program. The Y element and the comparison output control the output port in an OR relationship. If the Y element is continuously controlled as ON, the actual port output remains ON.



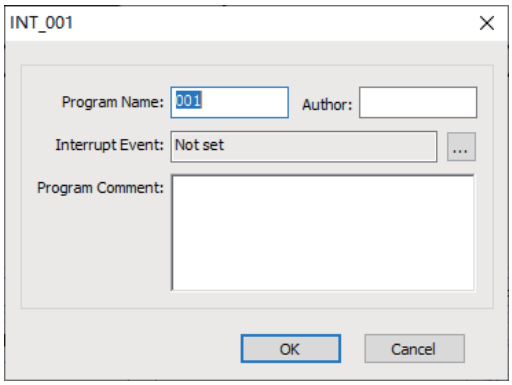
17.6.7 Comparison Interrupt

When the counter axes are compared for equality, comparison interrupt can be associated and the interrupt subprogram can be executed. The operation steps are as follows.

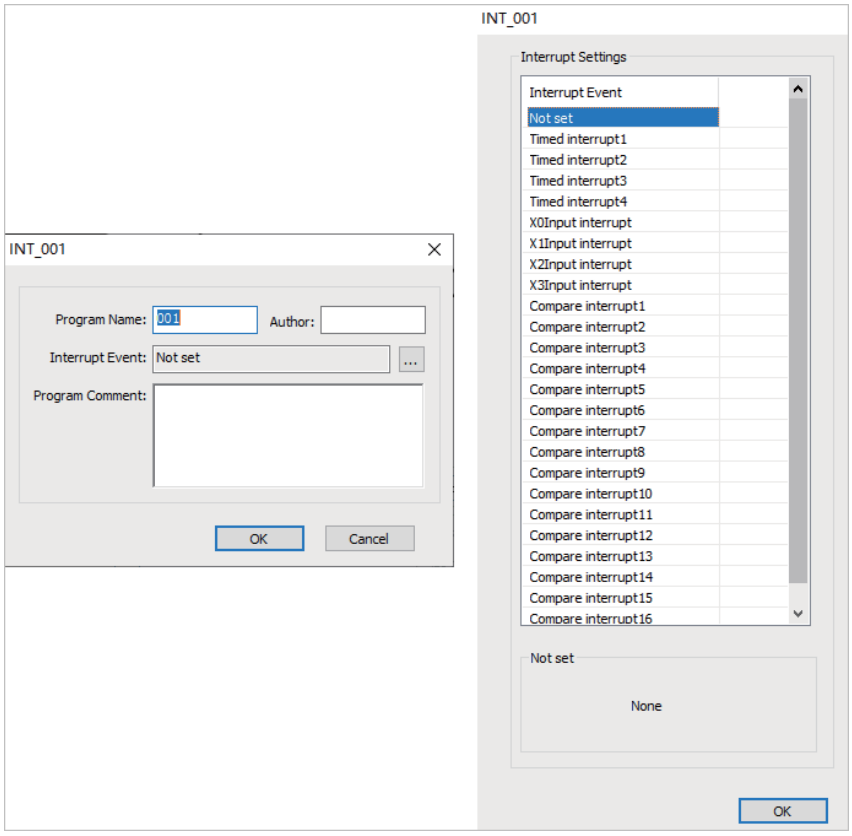
1. Under the item "Programming" in "Project Manager", right-click the "Program Block", and select "Insert interrupt subprogram".



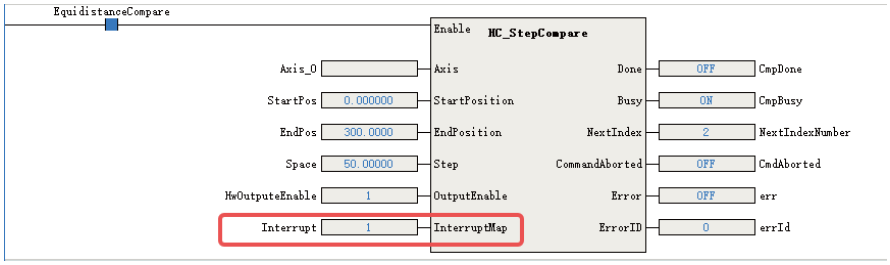
2. Right-click the inserted interrupt subprogram (such as INT_001 in the figure above) and select "Properties" to open the interrupt subprogram settings page as shown in the following figure.



3. Click the icon after the "Interrupt Event" field, select a comparison interrupt program and then write interrupt subprograms in INT_001.



4. Call HC_Compare, HC_StepCompare, and HC_ArrayCompare instructions in the main program or subprogram to associate the parameter InterruptMap with the comparison interruption number, that is, setting the parameter InterruptMap as the comparison interruption number. EI is enabled in the program, and when the instruction executes the comparison for equality, the comparison of corresponding interrupt subprograms will be triggered.



17.7 Counter Axis Instruction Application (Easy)

17.7.1 Overview

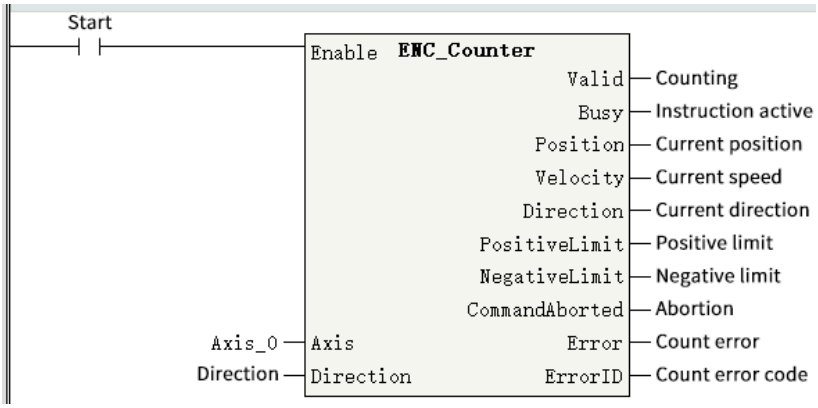
After a counter axis is set in AutoShop, the counter axis can be used in combination with function block instructions to implement features such as axis position counting/speed measurement, axis position presetting, and axis position latching and comparison.

17.7.2 Instruction for Axis Position Counting and Velocity Measurement

The instruction ENC_Counter implements position counting and velocity measurement of the counter axis.

The counter axis position varies within a certain range based on the mode setting, with user unit (Unit) as the position unit.

The counter axis velocity is the current real-time velocity (unit: Unit/s). The minimum velocity that can be measured by the counter axis is the velocity corresponding to one pulse of the counter within 1s. If one pulse of the counter corresponds to 0.01 Unit, the minimum velocity that can be measured is 0.01 Unit/s.



The parameter Direction in the instruction can be set to change the count direction. The modification on Direction takes effect only after this function block instruction is enabled again.

The following table shows the relationship between the Direction setting and the counting direction.

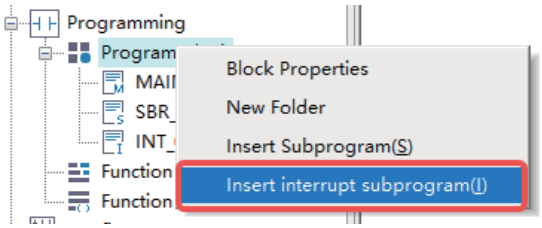
Direction	Phase A/B	Pulse + Direction	CW/CCW	Single-phase counter
0	Phase A leading phase B, counting up Phase B leading phase A, counting down	Direction signal = OFF, counting down Direction signal = ON, counting up	Phase A, counting up Phase B, counting down	Counting up
1	Phase A leading phase B, counting down	Direction signal = OFF, counting up	Phase A, counting down	Counting down

Direction	Phase A/B	Pulse + Direction	CW/CCW	Single-phase counter
	Phase B leading phase A, counting up	Direction signal = ON, counting down	Phase B, counting up	

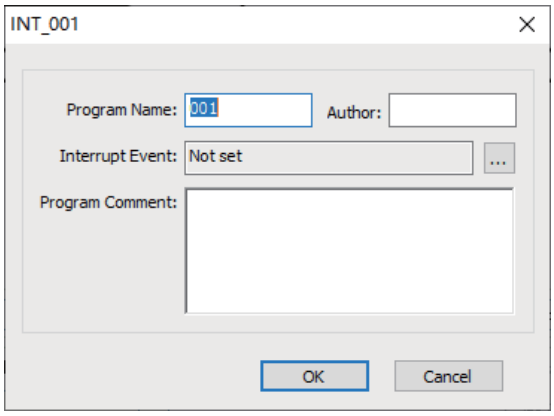
17.7.3 Axis Position Preset Instruction

When the counter axes are compared for equality, comparison interrupt can be associated and the interrupt subprogram can be executed. The operation steps are as follows.

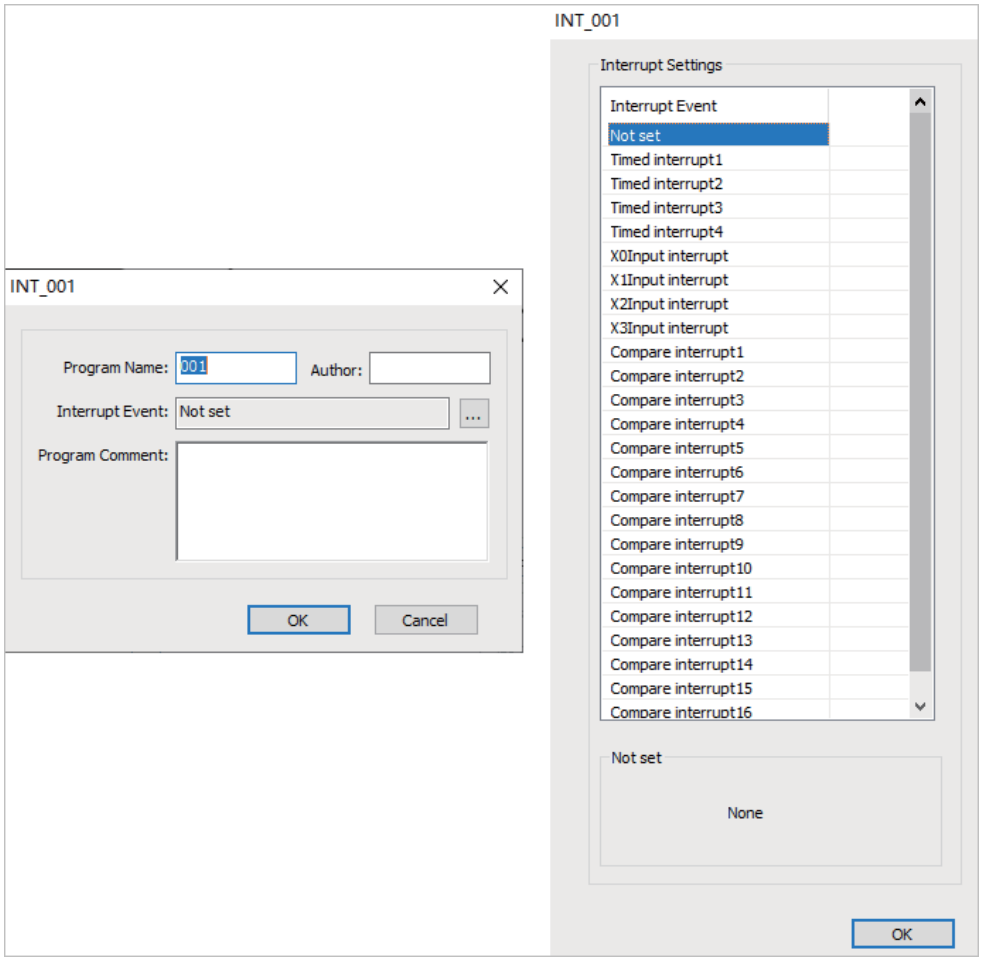
1. Under the item "Programming" in "Project Manager", right-click the "Program Block", and select "Insert interrupt subprogram".



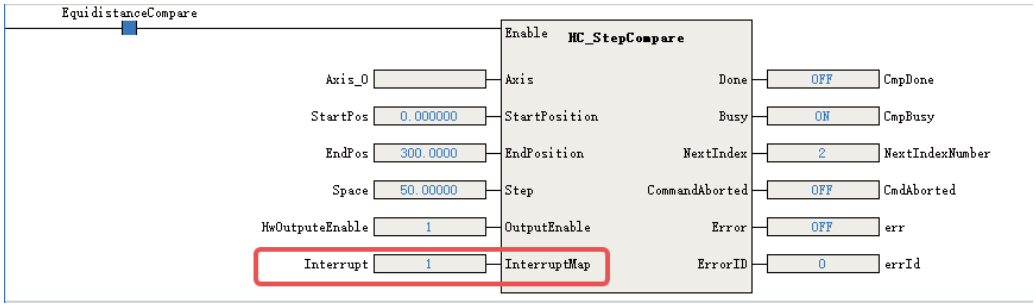
2. Right-click the inserted interrupt subprogram (such as INT_001 in the figure above) and select "Properties" to open the interrupt subprogram settings page as shown in the following figure.



3. Click the icon after the "Interrupt Event" field, select a comparison interrupt program and then write interrupt subprograms in INT_001.



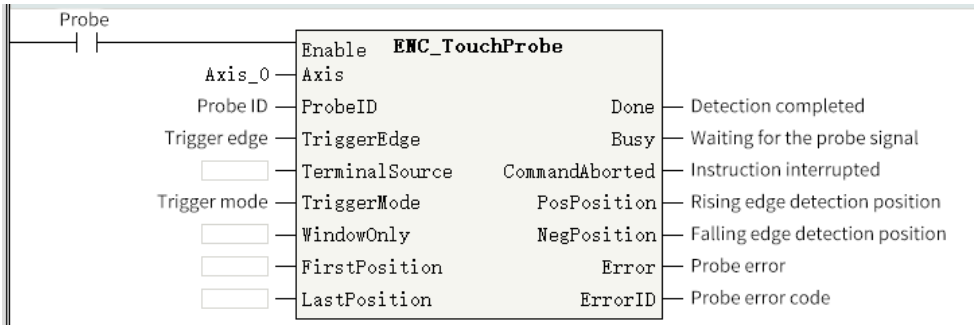
4. Call HC_Compare, HC_StepCompare, and HC_ArrayCompare instructions in the main program or subprogram to associate the parameter InterruptMap with the comparison interruption number, that is, setting the parameter InterruptMap as the comparison interruption number. EI is enabled in the program, and when the instruction executes the comparison for equality, the comparison of corresponding interrupt subprograms will be triggered.



17.7.4 Probe Instruction

The function block instruction ENC_TouchProbe can latch the counter axis position value when the external input trigger condition is valid.

Each counter axis supports two probes. During use, you need to tick the corresponding probe function in counter parameter settings and select "Input Terminal" and "Trigger Condition". The input terminal can be any of X0 to X7.



The parameter ProbeID specifies the number of the probe used by the counter.

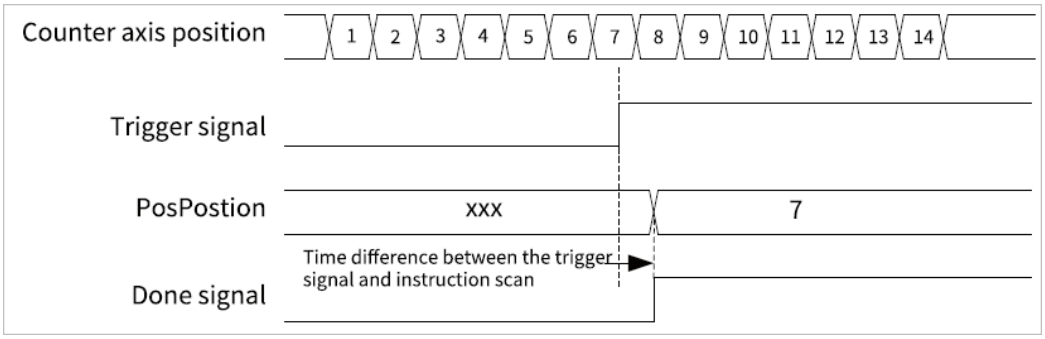
ProbeID	Definition
0	Indicates that probe 1 is used.

The parameter TriggerEdge specifies the probe trigger edge. The rising edge trigger position is latched in the output parameter PosPosition, and the falling edge trigger position is latched in the output parameter NegPosition.

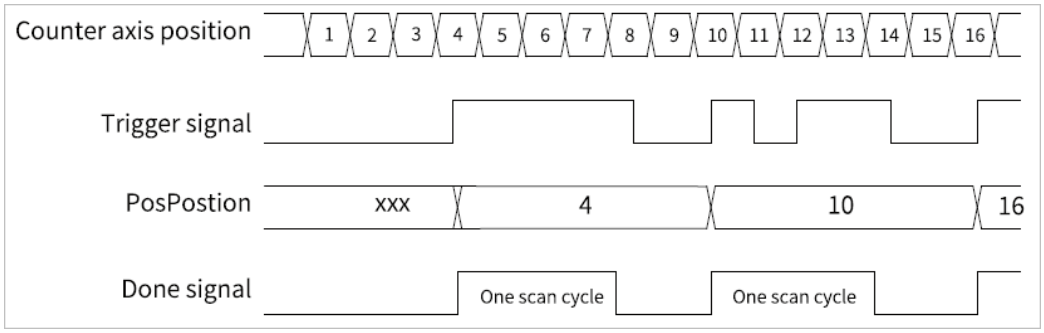
TriggerEdge	Definition
0	Trigger by the rising edge of the external X
1	Trigger by the falling edge of the external X
2	Trigger by the rising or falling edge of the external X

TriggerMode in the instruction can be set to the single trigger or continuous trigger.

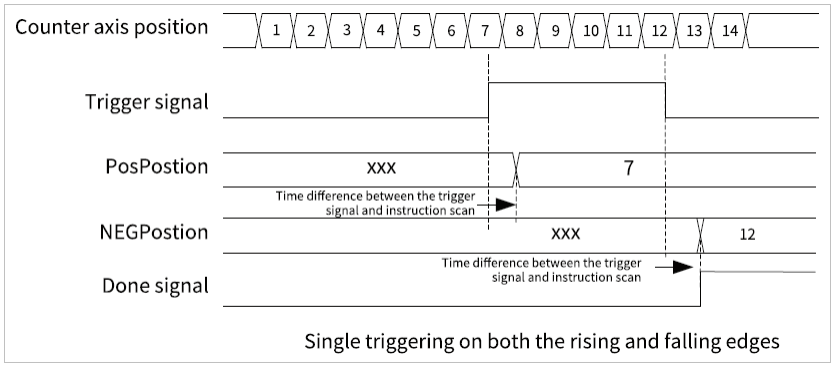
- If the single trigger mode is used, when the function block instruction flow and the external input trigger condition are valid, the counter axis position is latched once, and the Done signal is output.
  The counter axis position is latched in real time based on the trigger edge, which is not affected by program execution. During instruction execution, affected by the scan cycle, when the program scans and runs to the latched instruction, it updates the latched position to the output parameter of the instruction.



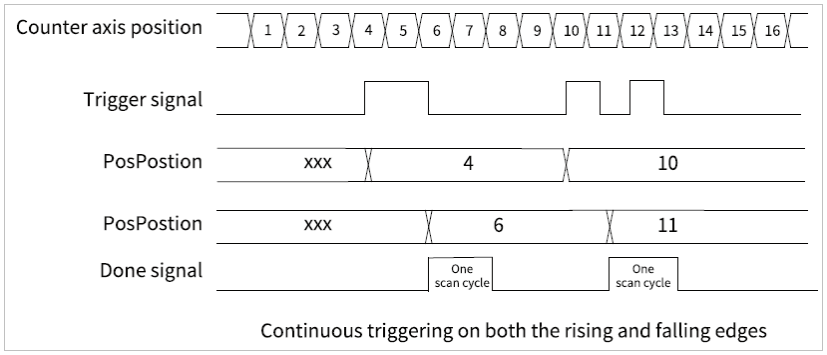
- If the continuous trigger mode is used, when the function block instruction flow and the external input trigger condition are valid, the counter axis position is latched, and the Done signal that is active for one scan cycle is output. When the Done signal becomes OFF and the external input trigger condition is valid, the counter axis position continues to be latched and the Done signal that is active for one scan cycle is output. During the scan cycle in which the Done signal is active, if the external input trigger condition is valid, the counter axis position is not latched at this time.



- When the dual-edge trigger mode is used, the Done signal is output after the instruction is triggered on both the rising and falling edges to complete the latch. In single trigger mode, the Done signal remains active until the instruction execution is completed; in continuous trigger mode, the Done signal is active for one scan cycle, and the latch signal is not responded within the scan cycle when the Done signal is active.



Single triggering on both the rising and falling edges



Continuous triggering on both the rising and falling edges

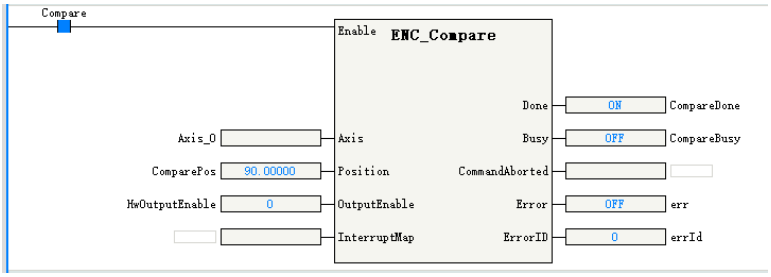
When WindowOnly is set to ON, the window setting is enabled and the probe is only active if the encoder axis is in the window confined by FirstPosition and LastPosition.

17.7.5 Comparison Instructions

The instructions ENC_Compare, ENC_StepCompare, and ENC_ArrayCompare can compare the counter axis position with a single position, equally-spaced positions continuously, or multiple positions continuously.

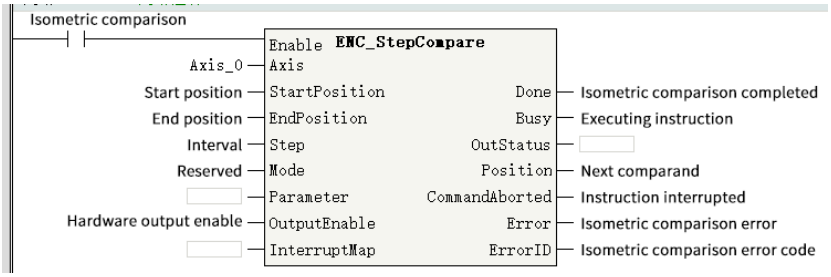
ENC_Compare

The instruction compares the counter axis with a single position. When the instruction flow is active, the Done signal is output after the counter axis position reaches the comparison position.

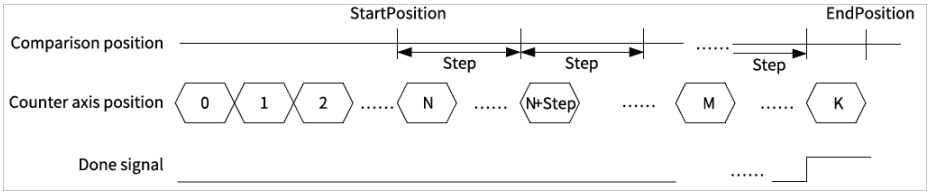


ENC_StepCompare

This instruction compares the counter axis position with equally-spaced positions continuously. When the instruction flow is active, the counter axis position is compared with the position specified by StartPosition. When they are equal, the comparison position increases or decreases by a value specified by Step and then is compared with the counter axis position. The Done signal of one cycle is not output after they are equal in each comparison, but after the last comparison position is compared.



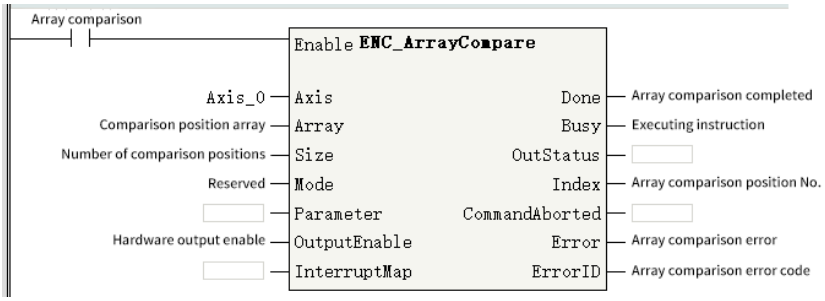
- If the StartPosition is less than the EndPostion, when the positions are equal in each comparison, the comparison position increases by a value specified by Step. When the current comparison position added with the value specified by Step is greater than the EndPosition, the current comparison position is the last one.
- If the StartPosition is greater than the EndPostion, when the positions are equal in each comparison, the comparison position decreases by a value specified by Step. When the current comparison position minus the value specified by Step is less than the EndPosition, the current comparison position is the last one.

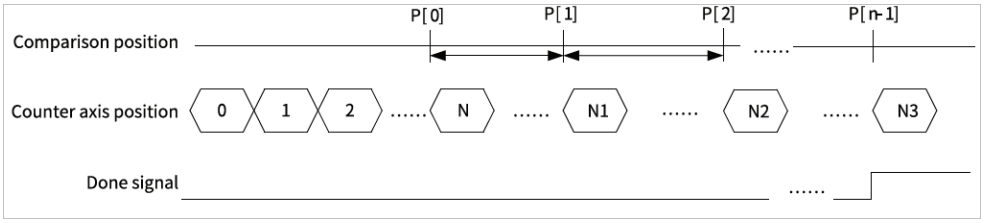


ENC_ArrayCompare

This instruction compares the counter axis position with multiple positions in an array continuously. When the instruction flow is active, the counter axis position is compared with the first position in the array. If they are equal, the counter axis position is compared with the next position value in the array. After the last comparison position is compared, the Done signal is output.

- Size in the instruction specifies the array length. After all the positions in the array are compared, the Done signal is continuously output and the comparison with multiple positions is completed.
- The output parameter Index indicates the index of the next comparison point, that is, the number of completed comparison points.





17.7.6 High-Speed Hardware Comparison Output

The counter axis can realize the position comparison hardware output. When the counter axis and the comparison position are equal, the output directly controlled by the hardware circuit is ON, and the output delay is less than 1 μs.

Setting the comparison output function of the counter axis

On the parameter setting page of the counter axis, select "Compare Output Enable".

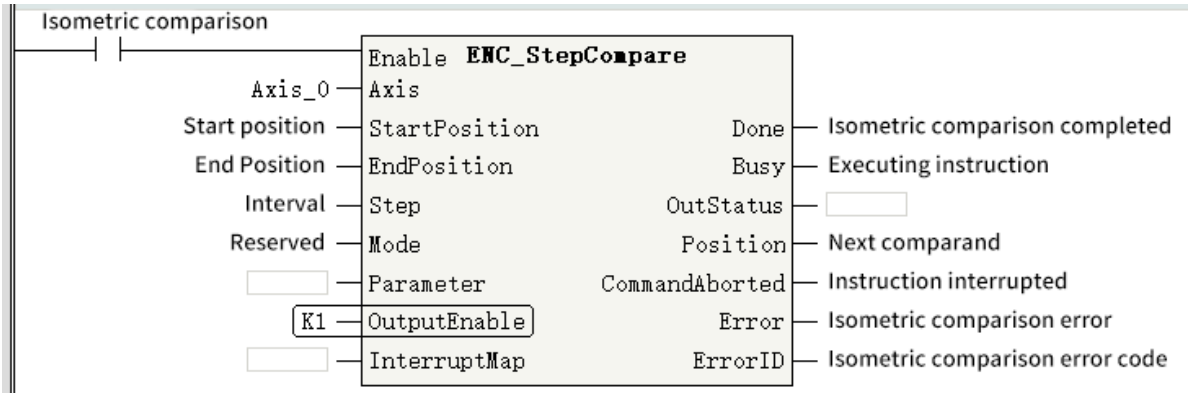
The output terminal can be any of Y0 to Y3. The output width is set as the pulse width when the output is ON, for which the unit can be time unit or user unit.

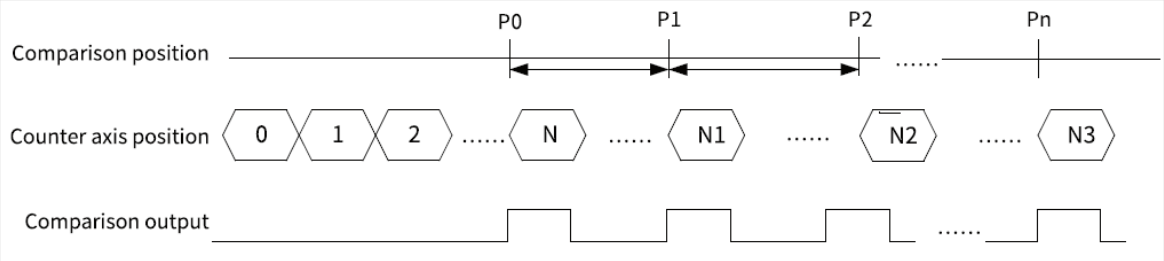
> **NOTE**
>
> When the unit of pulse output width is set to time, the time accuracy of pulse output width is 100 μs, and the maximum output width is 6500 ms. When the unit of pulse output width is set to pulse unit, the maximum output width is equivalent to 65535 pulses.

Enabling the OutputEnable parameter in the comparison instructions

Use ENC_Compare, ENC_StepCompare, and ENC_ArrayCompare comparison instructions to set OutputEnable to 1, that is, associate hardware outputs when the instruction executes the comparison for equality.

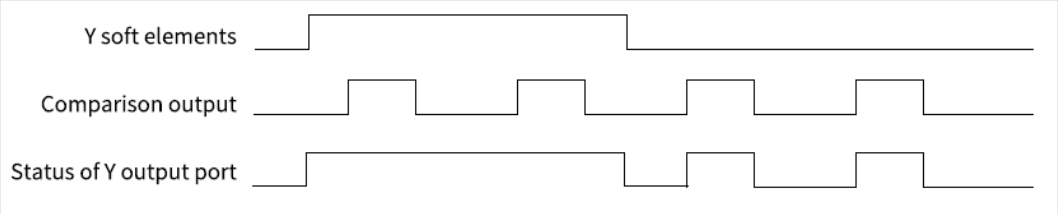
When the instruction executes the comparison for equality, the output terminal set directly through the control of hardware circuit is ON, and the output turns OFF after the continuous width output.





NOTE

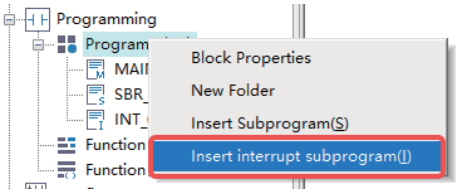
High-speed comparison hardware output is directly output through the hardware control port. Therefore, status of comparison output is not available in the Y element in the program. The Y element and the comparison output control the output port in an OR relationship. If the Y element is continuously controlled as ON, the actual port output remains ON.



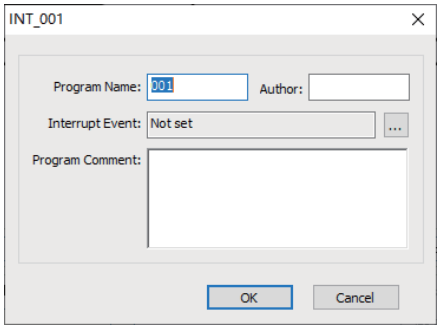
17.7.7 Comparison Interrupt

When the counter axes are compared for equality, comparison interrupt can be associated and the interrupt subprogram can be executed. The operation steps are as follows.

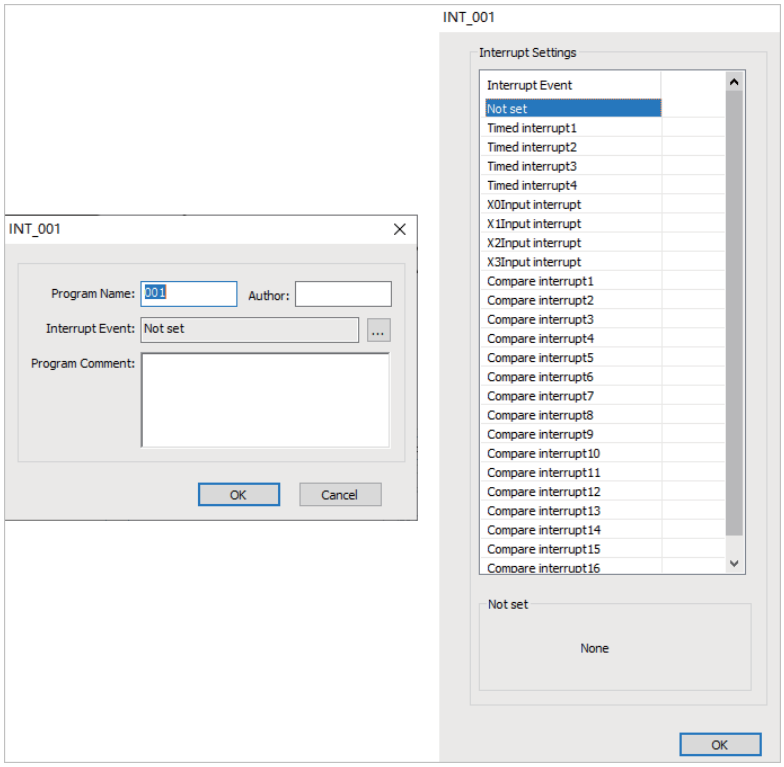
1. Under the item "Programming" in "Project Manager", right-click the "Program Block", and select "Insert interrupt subprogram".



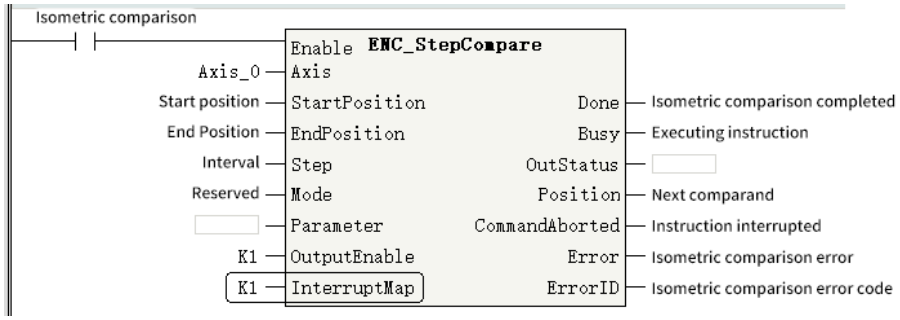
2. Right-click the inserted interrupt subprogram (such as INT_001 in the figure above) and select "Properties" to open the interrupt subprogram settings page as shown in the following figure.



3. Click the icon after the "Interrupt Event" field, select a compare interrupt program, and then write interrupt subprograms in INT_001.



4. Call ENC_Compare, ENC_StepCompare, and ENC_ArrayCompare instructions in the main program or subprogram to associate the parameter InterruptMap with the comparison interruption number, that is, setting the parameter InterruptMap as the comparison interruption number. EI is enabled in the program, and when the instruction executes the comparison for equality, the comparison of corresponding interrupt subprograms will be triggered.



17.7.8 Axis Gear Ratio Setting

The PLC reconfigures the gear ratio of the local encoder axis before the local encoder axis enables count after power-on, program download, or a RUN/STOP operation.

The gear ratio of a local encoder axis is calculated as follows:

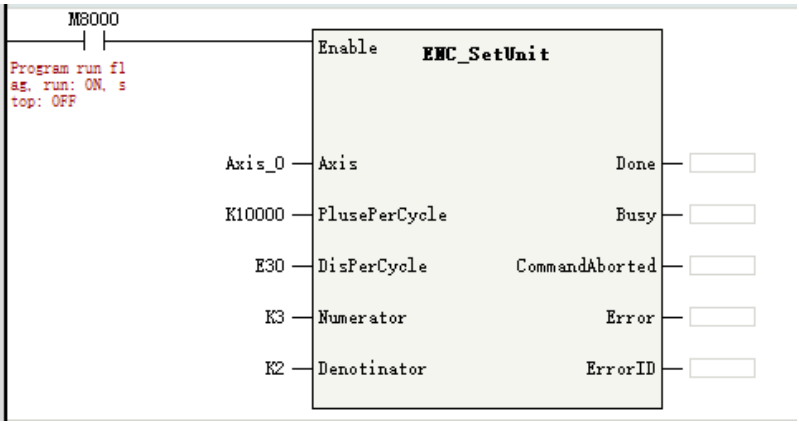
$$\text{Gear ratio} = \frac{PlusePreCycle \times Numerator}{DisPerCycle \times Denotinator}$$

This example describes how to set the local encoder axis parameters as follows automatically after power-on of the PLC.

Parameter	Value
Number of pulses per revolution of the encoder	10000
Distance per revolution of the rotary table	30

Parameter	Value
Gear ratio numerator	3
Gear ratio denominator	2

You are recommended to add the following program to the PLC to trigger this instruction with M8000.



NOTE

If this instruction is called during program execution, the local encoder axis will be re-initialized, causing a sudden change in its feedback position.

17.7.9 Axis Linear/Rotary Mode Setting

The PLC reconfigures the linear/rotary mode of the local encoder axis before the local encoder axis enables count after power-on, program download, or a RUN/STOP operation.

- When LineRotateMode is set to 0, the local encoder axis is in linear mode.
  In linear mode, when SoftLimitEnable is set to OFF, limiting is disabled. When SoftLimitEnable is set to ON, limiting is enabled. In this case, PLimit represents the positive limit, and NLimit represents the negative limit.
- When LineRotateMode is set to 1, the local encoder axis is in rotary mode. In this case, Rotation represents the rotation cycle.

For example, to realize that the local encoder axis automatically switches to the linear mode after the PLC is powered on, and the limit is enabled with a positive limit of +100 and a negative limit of -10, the program is as follows.

18 Interpolation Functions

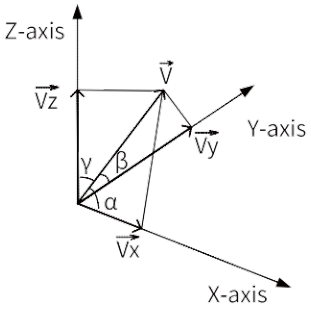
18.1 Introduction

18.1.1 Overview

The space rectangular coordinate system is adopted for interpolation, which supports linear interpolation, circular interpolation, and elliptical interpolation, and is performed in the form of axis groups.

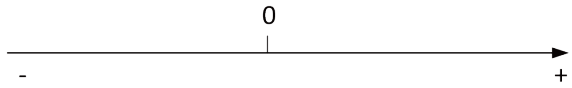
- Each axis group can control up to four motion control axes (bus servo axes or local pulse axes), including three coordinate axes X, Y, and Z, and one auxiliary axis.
- The H5U supports up to eight axis groups, each of which can contain two axes (X and Y), three axes (X, Y, and Z), or four axes (X, Y, Z, and auxiliary).
- The Easy series models support different number of axes. See the user guide of a specific model for the axis group supported. Each axis group needs at least two single axis. For example, Easy501 supports up to eight axes and supports four axis groups.
- Linear, circular, and elliptical interpolation support BufferMode. Each axis group allows up to eight curves to be buffered, and the transition mode between curves can be set separately (see Interrupt+No Transition for more information on buffering and transition).

The following figure shows the space rectangular coordinate system.



Vx, Vy, and Vz represent the velocities of the three coordinate axes, respectively, which are also the actual running velocities of the servo axes. V represents the real-time speed of the interpolation curve.α , β, and γ represent the angles between the speed of the interpolation curve and the coordinate axis, respectively.

The following figure shows the rectangular coordinate system of the auxiliary axis.



- During linear interpolation, the motion control axes representing the three coordinate axes of X, Y, and Z move along the coordinate axes, while the auxiliary axis moves along a straight line from the start point to the end point.
- During circular interpolation, you can choose one of the X-Y, Y-Z, and X-Z axis planes for circular interpolation, in which case if the axis group contains other axes, they will move along a straight line from the start point to the end point.
- During elliptical interpolation, you can choose one of the X-Y, Y-Z, and X-Z axis planes for elliptical interpolation, in which case if the axis group contains other axes, they will move along a straight line from the start point to the end point.

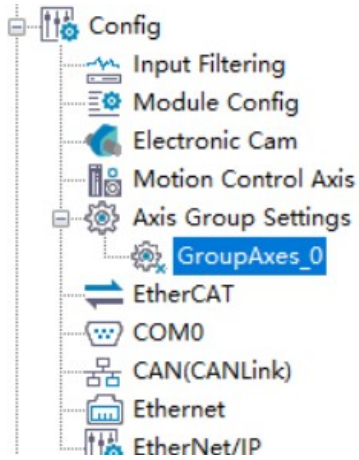
18.1.2 List of Axis Group Instructions

The following table lists axis group control instructions. See H5U and Easy Series Programmable Logic Controllers Instructions Guide for detailed usage of related instructions.

Instruction	Name
MC_MoveLinear	Linear interpolation
MC_MoveCircular	Circular interpolation
MC_MoveEllipse	Elliptical interpolation
MC_GroupStop	Stop axis group operation
MC_GroupPause	Pause axis group operation

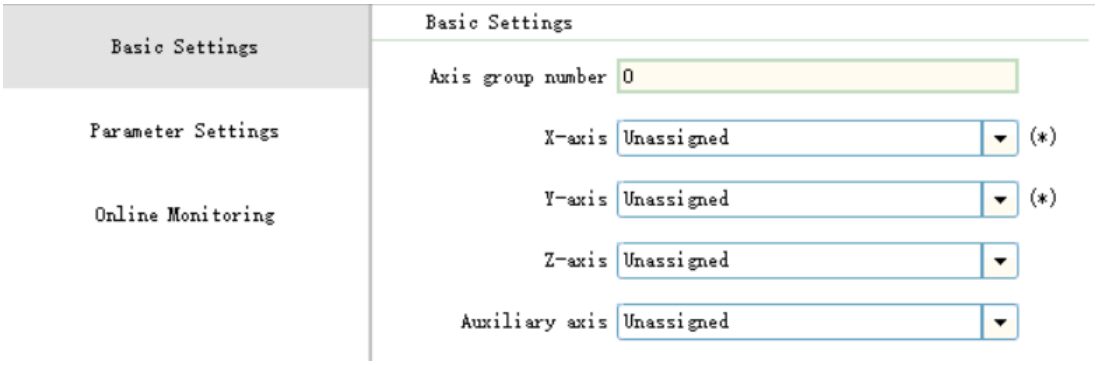
18.1.3 Configuration Interface

The menu "Axis Group Settings" is located under the node "Config". After creating an axis group, double-click the axis group to open the configuration interface of the axis group.



The interface of "Axis Group Settings" comprises three parts: "Basic Settings", "Parameter Settings", and "Online Monitoring".

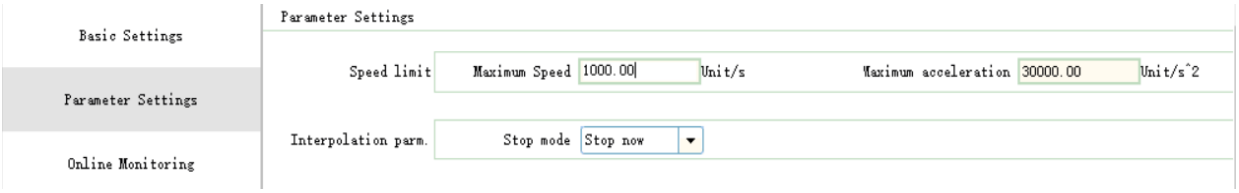
Basic Settings



- "Axis group number": Specifies the number of an axis group.

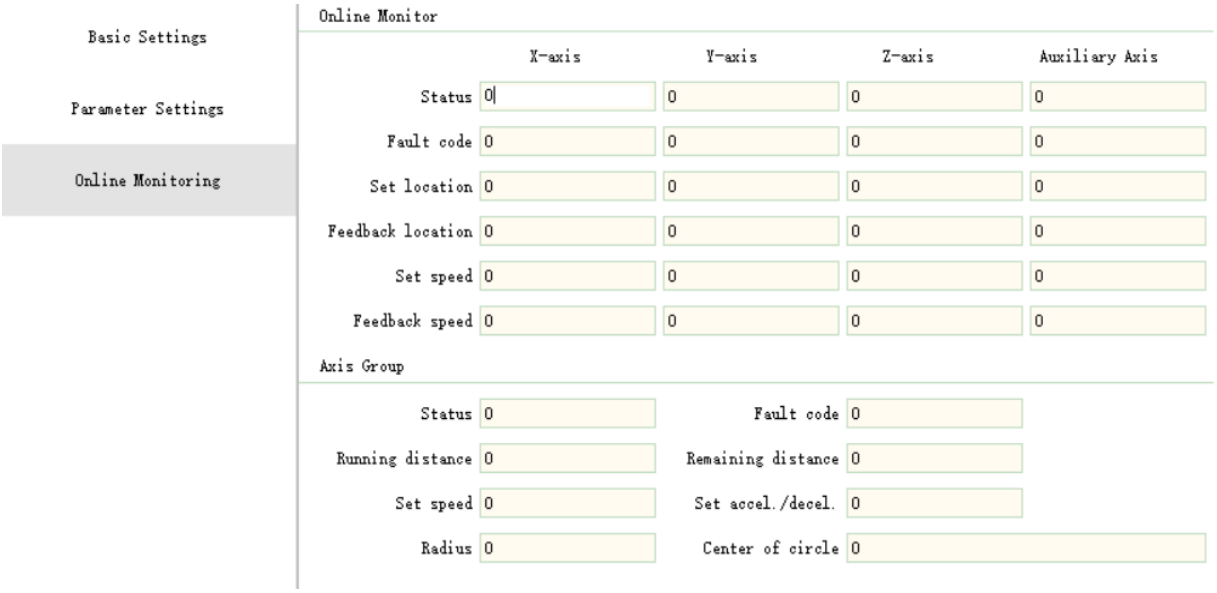
- Selection of coordinate axes: You can select coordinate axes from the corresponding drop-down lists, in which the X-axis and Y-axis are required, and the Z-axis and auxiliary axis are optional. One axis can exist in different axis groups.

Parameter Settings



- Maximum Speed: Specifies the maximum interpolation linear velocity of a space straight line in the linear interpolation mode, or the maximum linear velocity of a circular arc in the circular interpolation mode, or the maximum linear velocity of a circular arc in the circular interpolation mode.
- Maximum Acceleration: Specifies the maximum interpolation linear acceleration rate of a space straight line in the linear interpolation mode, or the maximum linear acceleration rate of a circular arc in the circular interpolation mode, or the maximum linear acceleration rate of a circular arc in the circular interpolation mode.
- Stop mode: Specifies the stop mode when the axis group encounters an error.

Online Monitoring



The following table lists online monitoring parameters.

Parameter Name	Description
Status	The status of a single-axis PLCOpen state machine 0: PowerOff 1: ErrorStop 2: Stopping 3: StandStill 4: DiscreteMotion 5: ContinuousMotion

Parameter Name	Description
	7: Homing 8: SynchronizedMotion
Fault code	The fault code when a single axis is in the ErrorStop state
Set location	Real-time target position for a single axis
Feedback location	Real-time feedback position for a single axis
Set speed	Real-time velocity reference for a single axis
Feedback speed	Real-time feedback velocity for a single axis

The following table lists the axis group monitoring parameters.

Name	Description
Status	The status of an axis group 0: Init The axis configuration in the axis group is not completed. 1: Disabled Not all axes in the axis group are enabled. 2: Single Stop An axis in the axis group calls the instruction MC_Gtop. 3: Single Homing An axis in the axis group calls the instruction MC_Home. 4: Single motion An axis in the axis group calls single-axis motion instructions such as MC_MoveAbsolute. 5: ErrorStop An axis in the axis group is in a fault state. 6: StandStill All axes in the axis group are in the StandStill state. 7: Stopping The instruction MC_GroupStop is called. 8: Synchronous Motion A linear interpolation, circular interpolation, or elliptical interpolation instruction is called.
Fault code	The fault code when the axis group fails due to error of a single axis
Running distance	In linear interpolation mode, it indicates the distance at which a space straight line moves after the instruction is executed.

Name	Description
	In circular interpolation mode, it indicates the length of a circular arc in which the circular arc moves after the instruction is executed. In elliptical interpolation mode, it indicates the length of an elliptical arc in which the elliptical arc moves after the instruction is executed.
Remaining distance	In linear interpolation mode, it indicates the left distance for this section of a space straight line after the instruction is executed. In circular interpolation mode, it indicates the length of a space circular arc left after the instruction is executed. In elliptical interpolation mode, it indicates the length of an elliptical arc left after the instruction is executed.
Set speed	In linear interpolation mode, it indicates the interpolation velocity of a space straight line. In circular interpolation mode, it indicates the linear velocity of a circular arc. In elliptical interpolation mode, it indicates the linear velocity of an oval.
Set accel./decel.	The change rate of the velocity reference
Radius	The radius of a circular arc during circular interpolation
Center of circle	The center of a circular arc during circular interpolation

18.2 Interpolation Operation

18.2.1 Overview

To properly execute an interpolation instruction, you need to first create an axis group and enable axes in the axis group. The following figure shows the basic process.

NOTE

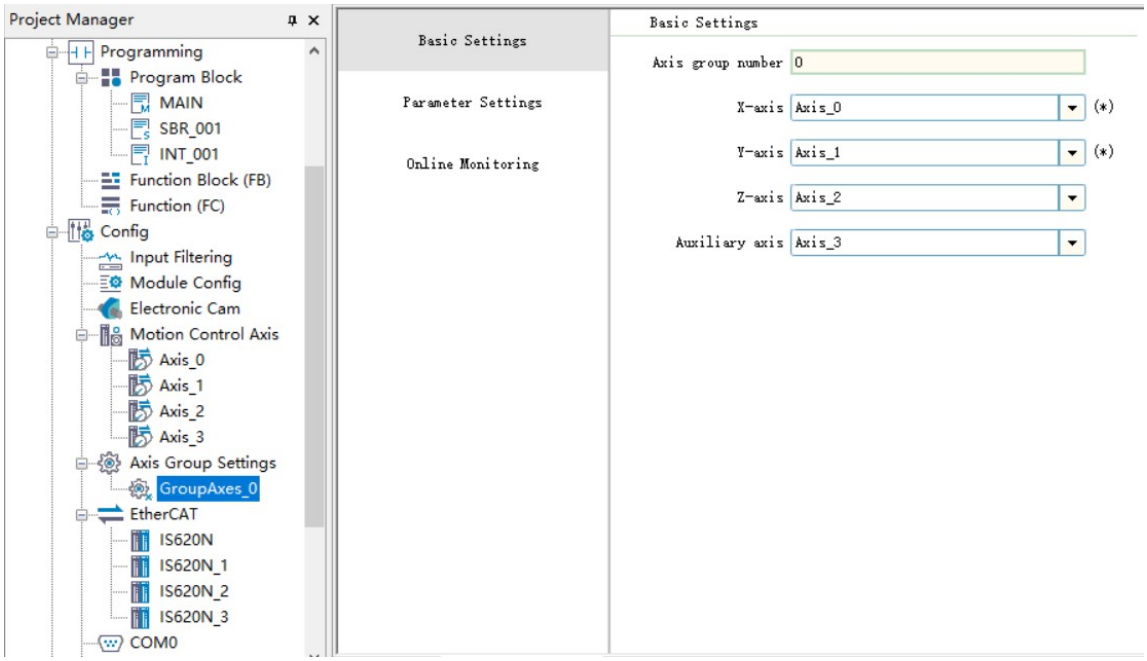
After an axis group is created, axes in the axis group can still execute single-axis motion and control instructions. However, motion instructions for a single axis and interpolation instructions for an axis group are mutually exclusive. These instructions cannot be activated at the same time or interrupt each other.

This section describes the basic interpolation procedures based on a routine that combines Axis_0, Axis_1, Axis_2, and Axis_3 into an axis group to perform related actions.

For details about the configuration of motion control axes, see Configuring Motion Control Axes.

18.2.2 Creating an Axis Group

Right-click "Axis Group Settings", and select "Add Axis Group". After creating an axis group, you can choose the coordinate axes and auxiliary axis and set relevant parameters.



18.2.3 Enabling the Axis Group

Each single axis in the axis group is enabled and disabled through the instruction MC_Power. The axis group control instruction can only be executed if all axes in the axis group are enabled.

18.2.4 Linear Interpolation

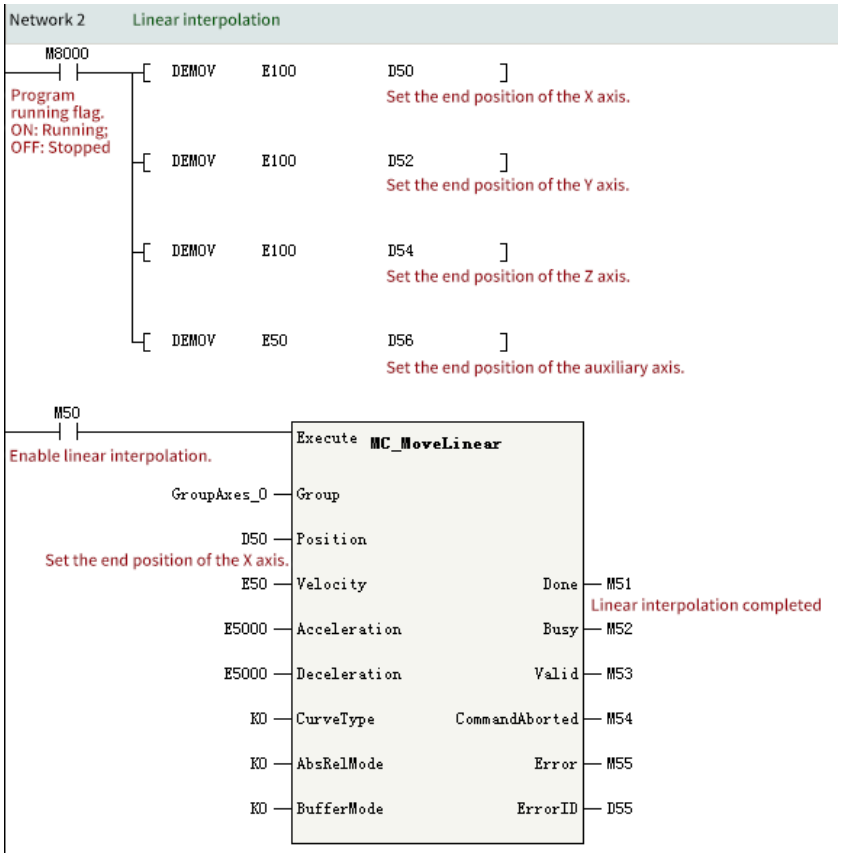
The linear interpolation of an axis group is implemented by the instruction MC_MoveLinear .

When all axes in the axis group are in the StandStill state, the Execute is triggered, the axis group starts to implement linear interpolation, and all axes in the axis group switch to the Synchronized Motion state. In this case, the single-axis motion instructions such as MC_MoveAbsolute and MC_Stop must not be executed.

After the linear interpolation is completed, all axes in the axis group return to the StandStill state, in which case single-axis motion instructions such as MC_MoveAbsolute and MC_Stop can be executed again.

Example

This routine uses absolute positioning to position the X-axis, Y-axis, and Z-axis to the position (100,100,100), and the auxiliary axis to the position 50.

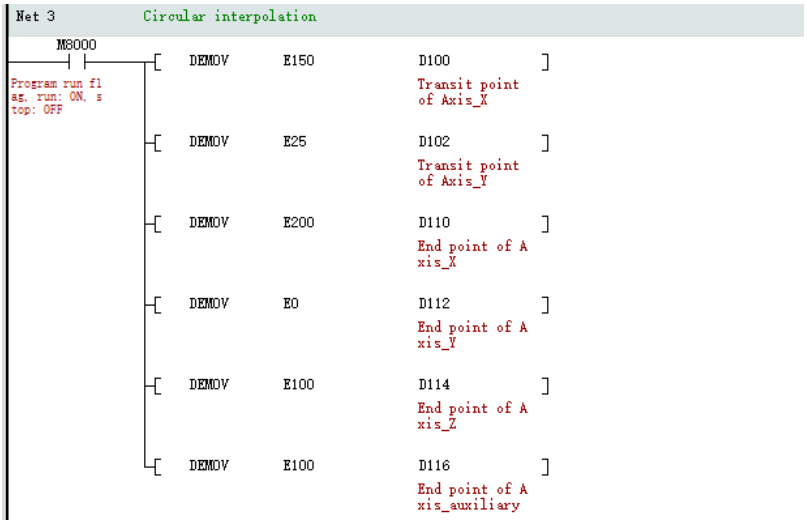


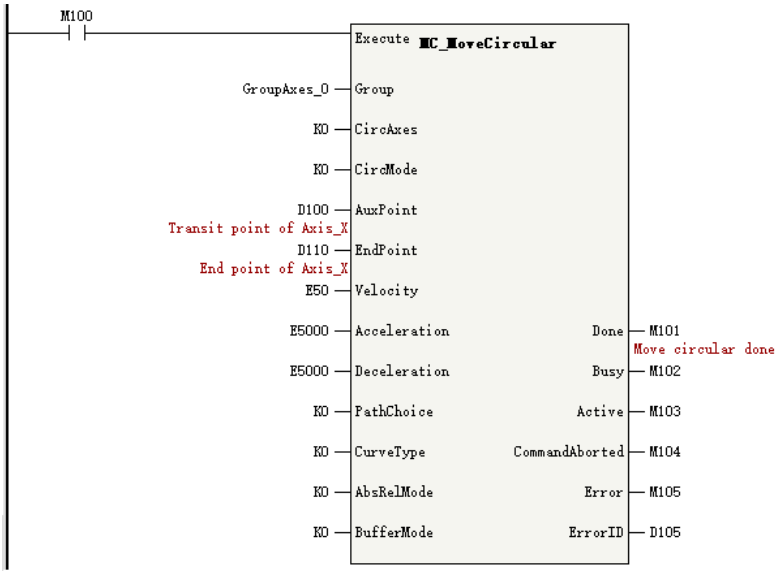
18.2.5 Circular Interpolation

Circular interpolation of an axis group is implemented by the instruction MC_MoveCircular. The conversion rules for PLCOpen state machines are the same as those for linear interpolation.

This routine implements circular interpolation of the X-Y axis plane while the Z-axis and auxiliary axis making synchronous linear motion. The circular interpolation is implemented in the border point mode, that is, with absolute positioning, first passing through the border point (150,25) and then reaching the position (200,0). The Z-axis and auxiliary axis reach the position 100.

For specific parameters related to the circular arc instructions, see H5U and Easy Series Programmable Logic Controllers Instructions Guide.



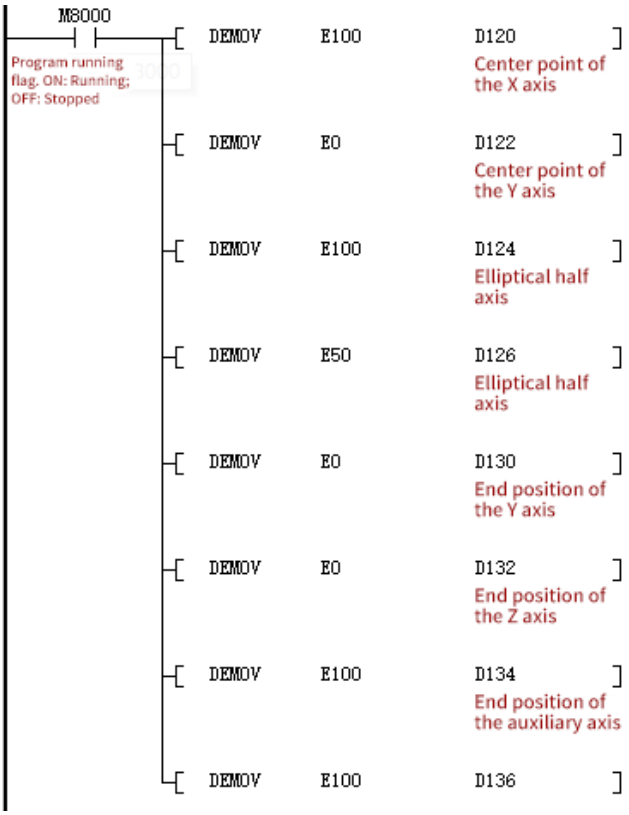


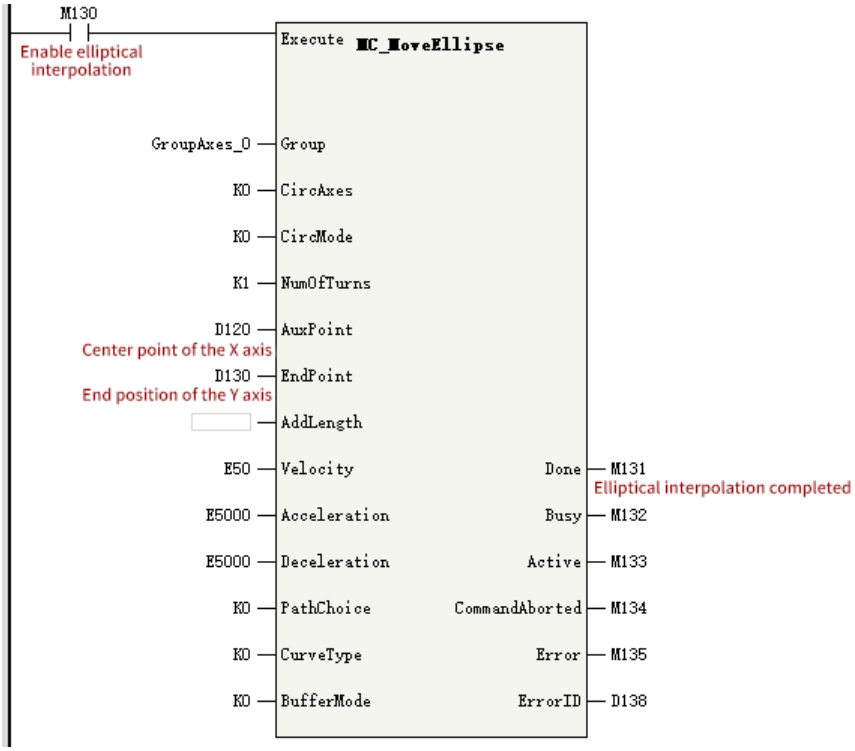
18.2.6 Elliptical Interpolation

Elliptical interpolation of an axis group is implemented by the instruction MC_MoveEllipse. The conversion rules for PLCOpen state machines are the same as those for linear interpolation.

This routine implements elliptical interpolation of the X-Y axis plane while the Z-axis and auxiliary axis making synchronous linear motion. Elliptical interpolation is done in run completed ellipse mode, running clockwise, with the ellipse start position at (0,0), center point at (100,0), and end position at (0,0), with the long half axis set to 100 and the short half axis set to 50. The Z-axis and auxiliary axis reach the position 100.

For specific parameters related to the ellipse instructions, see H5U and Easy Series Programmable Logic Controllers Instructions Guide.





18.2.7 Axis Group Stop

Stop the execution of the interpolation curve by the instruction MC_GroupStop .

The execution of the interpolation instruction is interrupted on the rising edge of Execute, and the CommandAborted output of the interpolation instruction is valid.

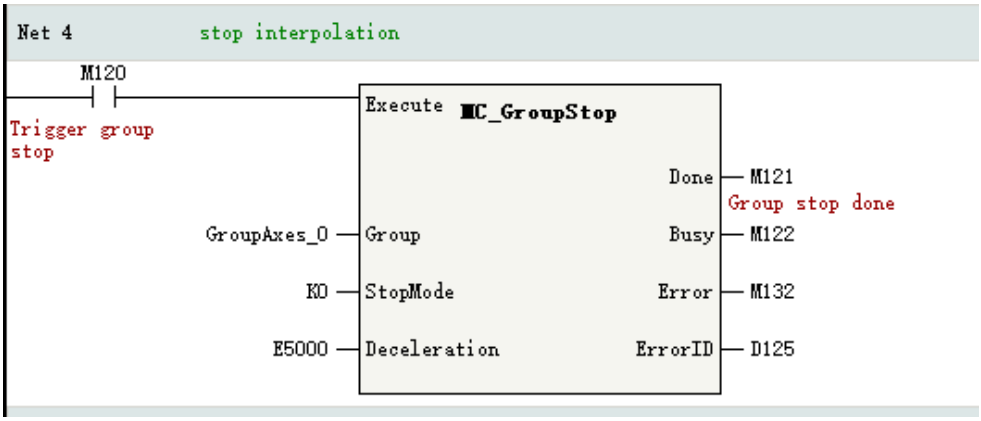
Interpolation instructions cannot be triggered when Execute is TRUE. Execute must be set to False to re-execute a new interpolation instruction.

This instruction can only be called when all axes in the axis group are in the StandStill or Synchronized Motion state. Axes are in the Synchronized Motion state while the Execute of the instruction is true.

MC_GroupStop can only stop the operation of an interpolation curve, rather than that of single-axis motion instructions such as MC_MoveAbsolute.

Example

In this routine, the following instruction is called during linear interpolation or circular interpolation to stop the operation of an interpolation curve. With the decelerate-to-stop mode in use, the deceleration to stop is 5000.



18.2.8 Axis Group Pause

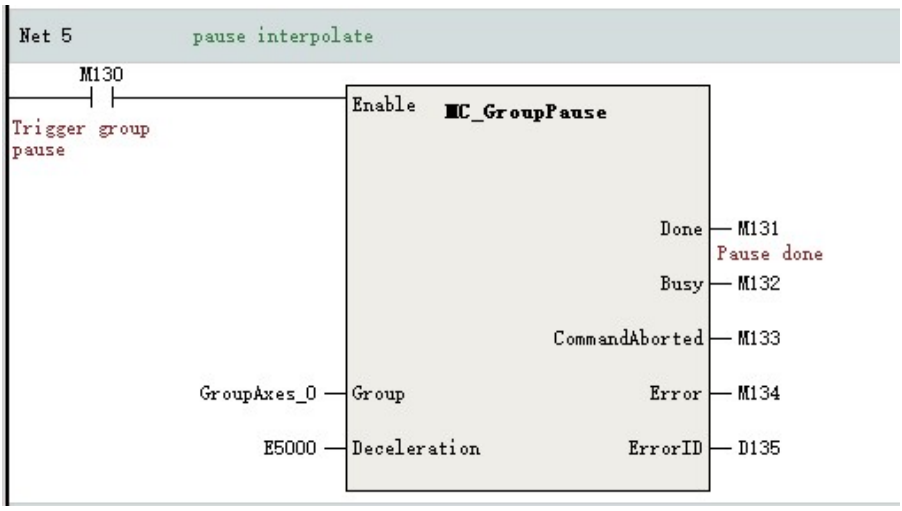
The interpolation curve is paused by the instruction MC_GroupPause .

Pause the interpolation curve when Enable is TRUE and resume the execution when Enable is False.

MC_GroupPause can only pause the operation of an interpolation curve, rather than that of single-axis motion instructions such as MC_MoveAbsolute.

Example

In this routine, the following instruction is called during line or circular interpolation to pause the operation of the interpolation curve, and the deceleration to pause is 5000.

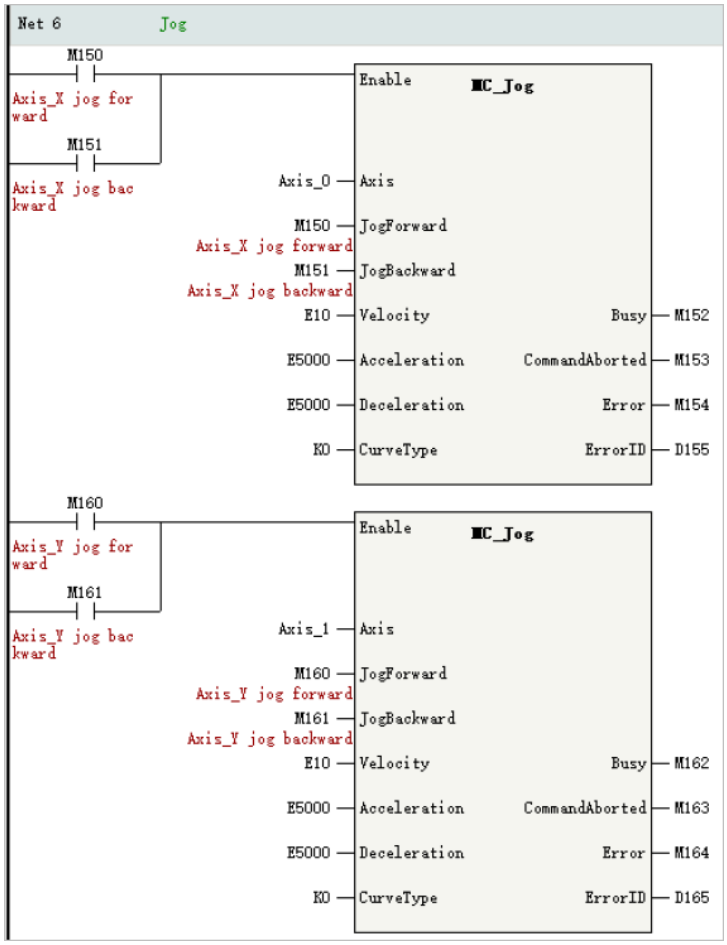


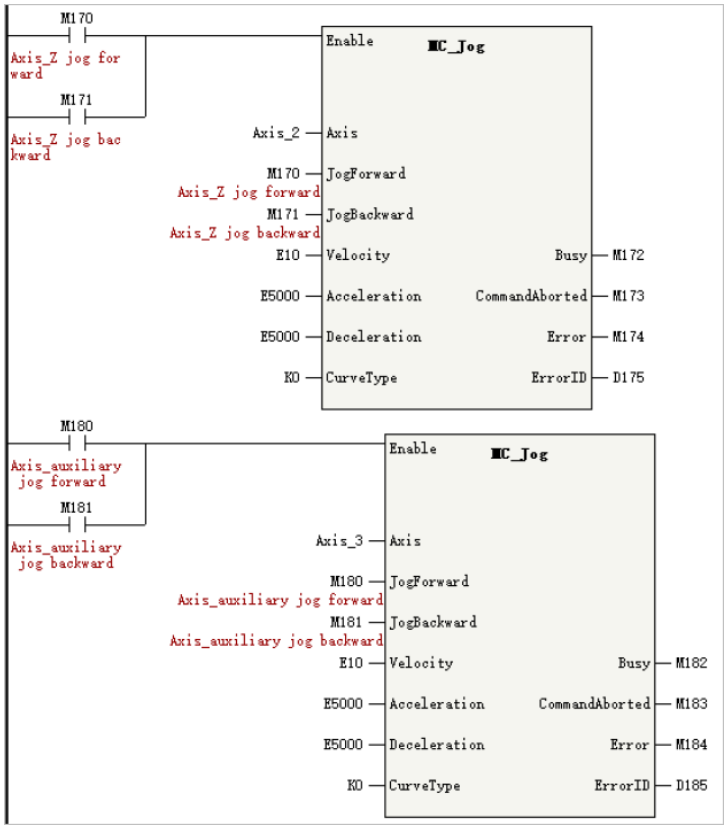
18.2.9 Single-Axis Motion

When the single-axis PLCOpen state machine is in the Synchronized Motion state due to the execution of the interpolation action, the single-axis instructions such as MC_MoveAbsolute and MC_Stop must not be executed. When the axis is in the StandStill state, the single-axis motion instructions, such as MC_MoveAbsolute, MC_MoveRelative, and MC_Jog, can be called to control the single-axis operation.

Example

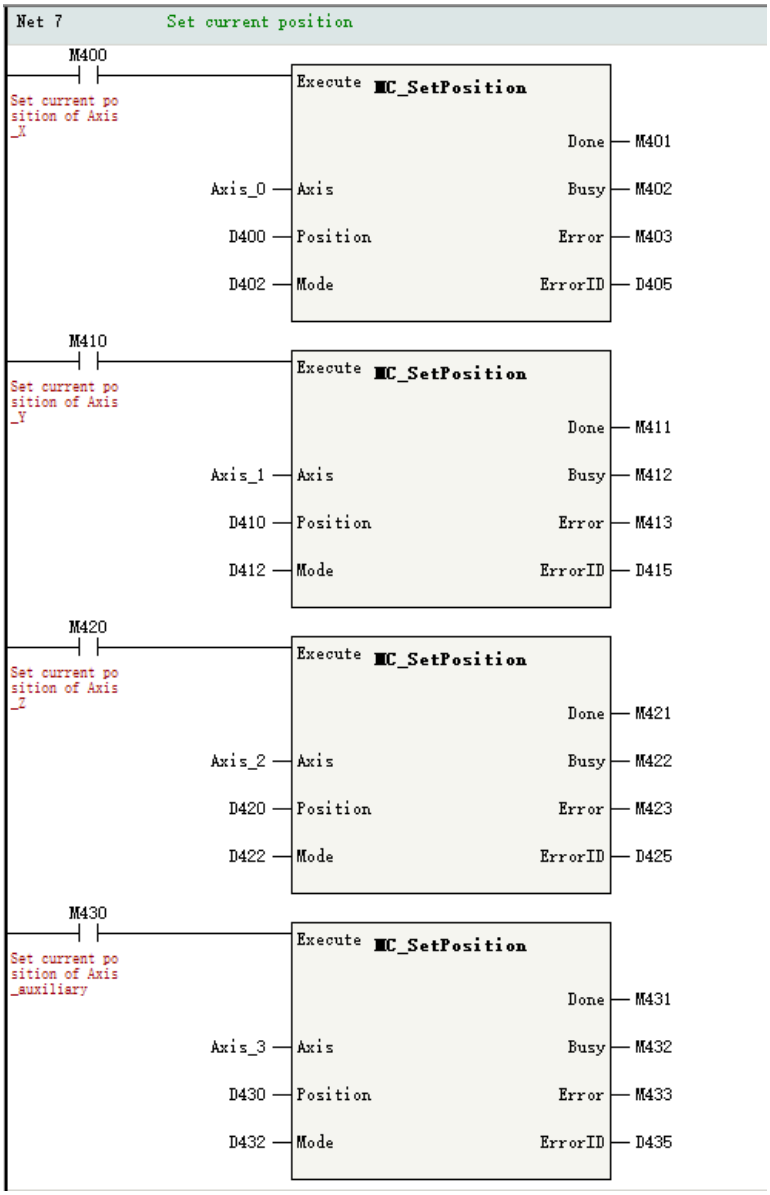
In this routine, when no interpolation instruction is executed for the axis group, the jogging of the single axis is realized through the instruction MC_Jog.





18.2.10 Setting the Current Position

Set the current position through the instruction MC_SetPosition.



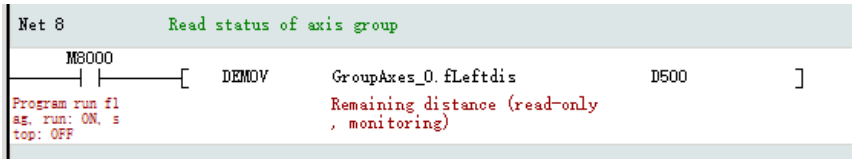
18.2.11 Reading the Current Status

Single-axis status

The status, feedback position, feedback velocity, and feedback torque of the PLCOpen state machine of a single axis are obtained through instructions MC_ReadStatus, MC_ReadActPosition, MC_ReadActVelocity, and MC_ReadActTorque. The status of a single axis can also be accessed through the system variables of the axis. See Online Monitoring Online Monitoring for details.

The status of an axis group

Access the state of an axis group through its system variables.

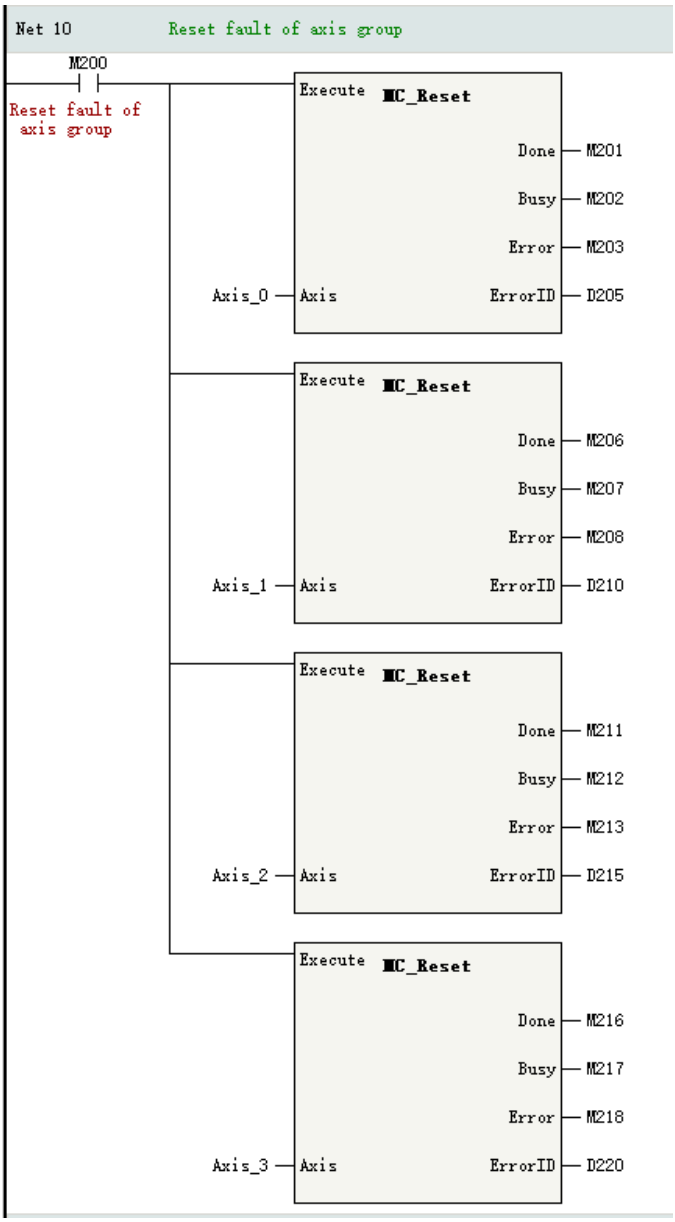


18.2.12 Resetting an Axis Group Fault

When a single axis enters the fault state, the fault of the axis can be reset through the instruction MC_Reset, and only by resetting the single-axis fault can the fault of the entire axis group be removed.

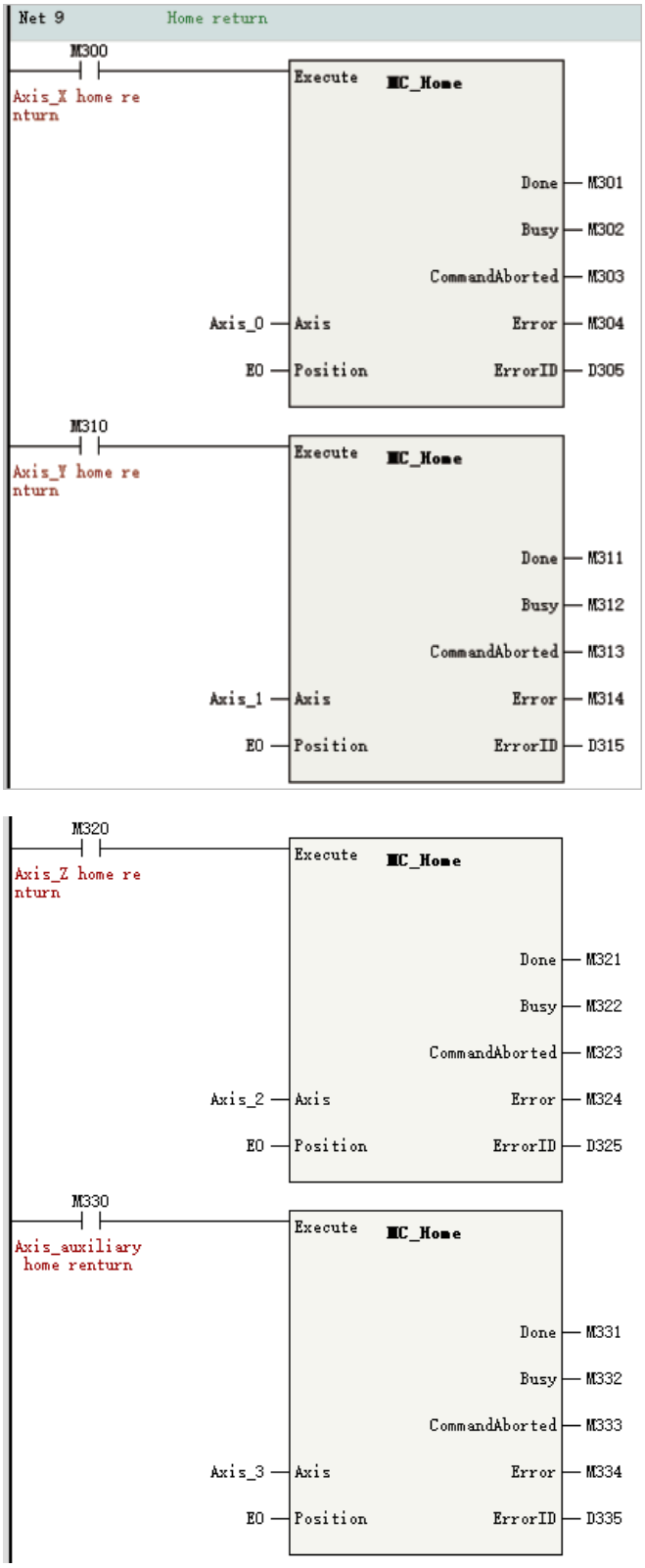
Example

In this routine, M200 is triggered to reset the fault of the axis group.



18.2.13 Homing

The homing for an axis group can be realized through single homing. The following is the single homing implemented by the instruction MC_Home.



18.3 Buffer and Transition

18.3.1 Overview

The buffer mode refers to the process of executing instructions when multiple interpolation instructions are started at the same time.

The transition mode refers to the way when multiple curves switch between each other.

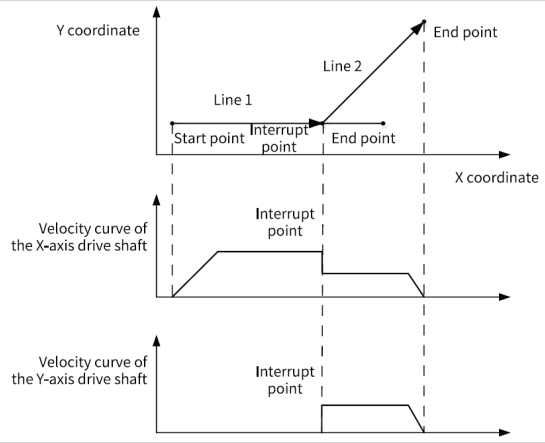
The following four combined buffer and transition modes are supported.

No.	Buffer mode	Description
0	Aborting + No transition	Immediately switch to the next function block. There is no transition curve.
1	Buffered + No transition	Execute the buffered function block after the first segment of deceleration is completed. There is no transition curve.
2	Previous velocity + No transition	Move to the end of the first segment at the current velocity and start the second segment at the rate of the first segment.
3	Superimpose corners	Add acceleration of the second segment when deceleration starts in the first segment. There is a transition curve.

18.3.2 Interrupt+No Transition

The first interpolation instruction is executed first, and the second interpolation instruction is triggered before the first straight line is completed. If the BufferMode of the second interpolation instruction is set to "Interrupt+No Transition", the second interpolation instruction immediately interrupts the first interpolation instruction and starts implementing a new interpolation curve.

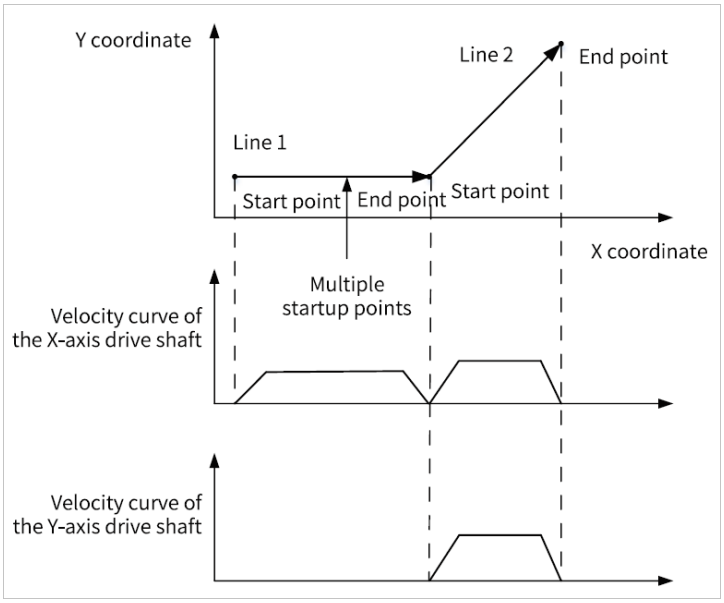
At the interrupt point, the new curve remains at the same velocity rate, and the velocities of the X-axis, Y-axis are re-decomposed, as shown in the following figure.



18.3.3 Buffer+No Transition

The first interpolation instruction is executed first, and the second interpolation instruction is triggered before the first straight line is completed. If the BufferMode of the second interpolation instruction is set to "Buffer+No Transition", the interpolator will continue to execute the first interpolation instruction.

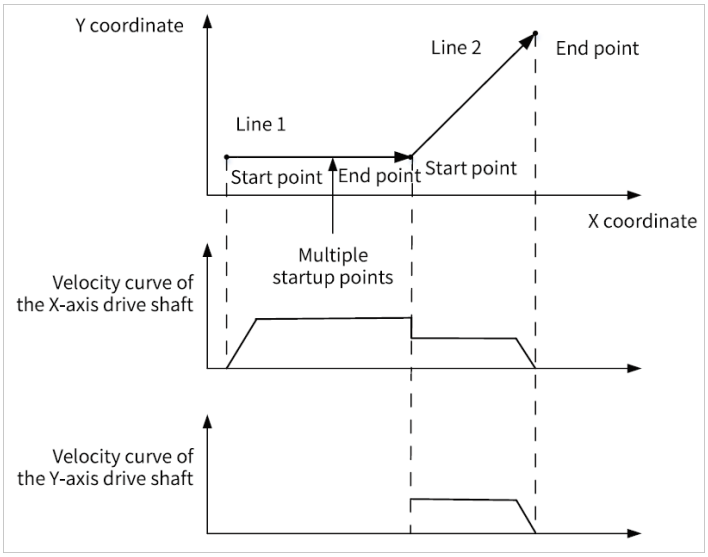
After the first interpolation instruction is executed and an active Done signal is output, the execution of the second interpolation instruction will begin, as shown in the following figure.



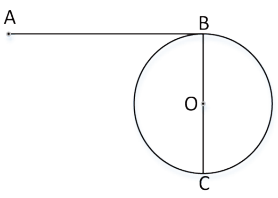
18.3.4 Previous Velocity+No Transition

The first interpolation instruction is executed first, and the second interpolation instruction is triggered before the first straight line is completed. If the BufferMode of the second interpolation instruction is set to "Previous Velocity+No Transition", the interpolator will attempt to maintain the target velocity of the first instruction to implement a full straight line.

After the first interpolation instruction is executed and an active Done signal is output, the execution of the second interpolation instruction will begin. The velocity rate will remain unchanged at the switching point, and the velocities of the coordinate axes will be redistributed. See the following figure.

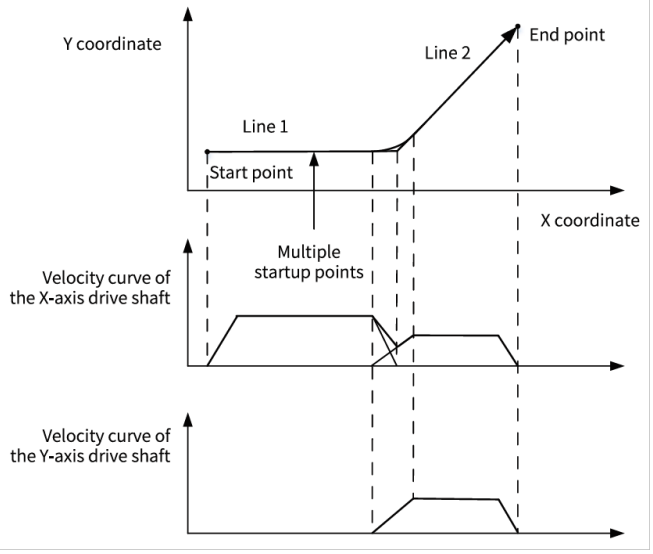


This mode is particularly suitable for switching between straight lines and circular arcs with the straight lines on the tangent line of the arcs, and can be used to maintain the constant velocity of an interpolation curve.



18.3.5 Additional Angle Mode

The first interpolation instruction is executed first, and the second interpolation instruction is triggered before the first straight line is completed. If the BufferMode of the second interpolation instruction is set to the additional angle mode, the interpolator will initiate the second interpolation instruction when it detects that the first straight line has begun to perform deceleration, and the final velocity of each coordinate axis is the sum of the velocities of the two instructions, as shown in the following figure.



18.4 Handling Methods of Single-Axis Configuration Parameters in Interpolation

Some of the configuration parameters in the interpolation are different from the single-axis configuration parameters, as shown in the following table.

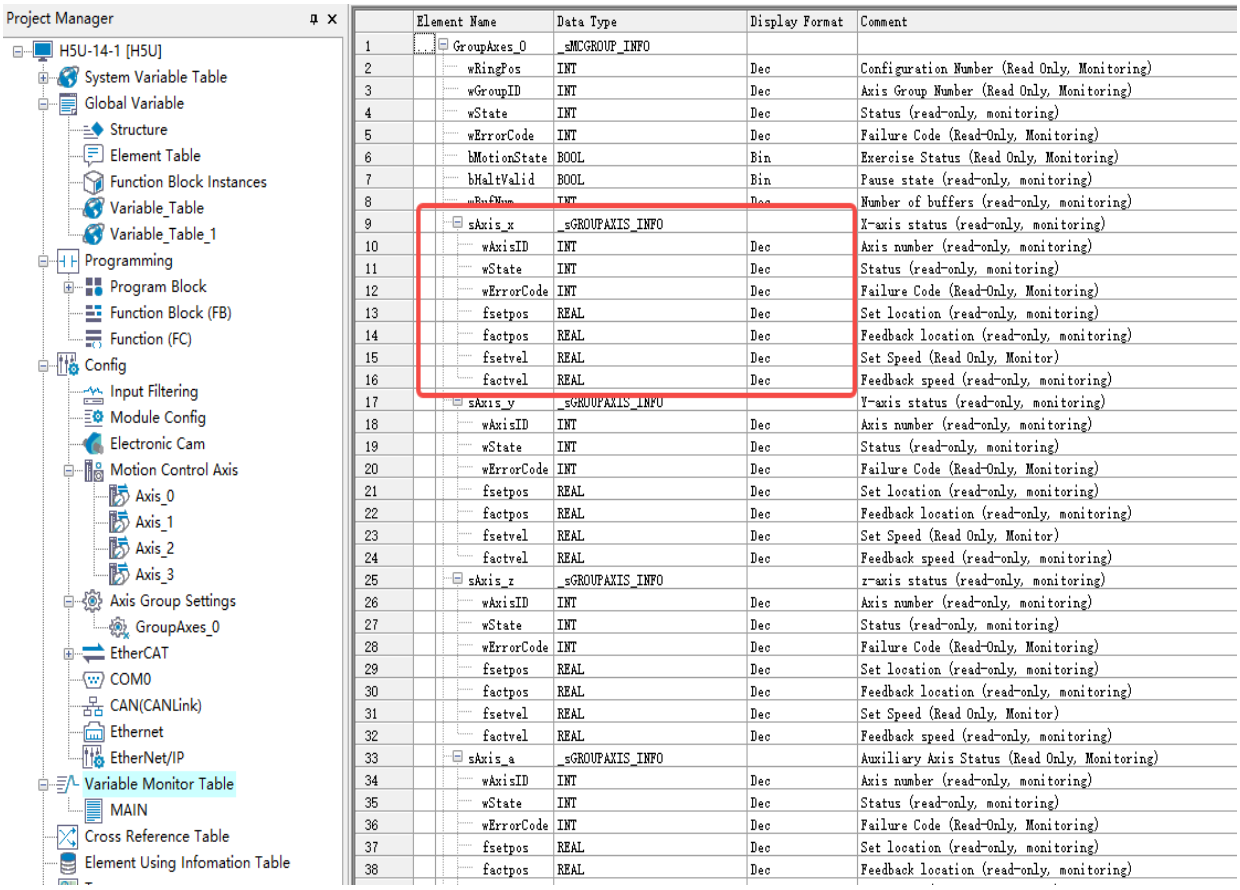
Single-axis configuration parameters	Handling methods in interpolation
Gear ratio setting	The gear ratio of a single axis in an axis group is set on the interface “Unit Conversion Settings” of the single axis.
Encoder mode selection	The encoder mode of the drive can be set to incremental or absolute. Set this parameter in the interface “Mode/Parameter Settings” of a single axis.
Mode	Modes of axes in an axis group are divided into linear mode and ring mode according to the working conditions, and the interpolation instructions only support the linear mode.
Limit Handling	Axes in the interpolation instructions support both software limits and hardware limits.
Tracking error	Axes in interpolation instructions support following errors.
Speed limit	The velocity in an interpolation instruction is limited by the maximum velocity of a single axis, not by the maximum acceleration rate.
Torque limit	Not involved.

18.5 System Variables

18.5.1 Axis Status of Axis Group _sGROUPAXIS_INFO

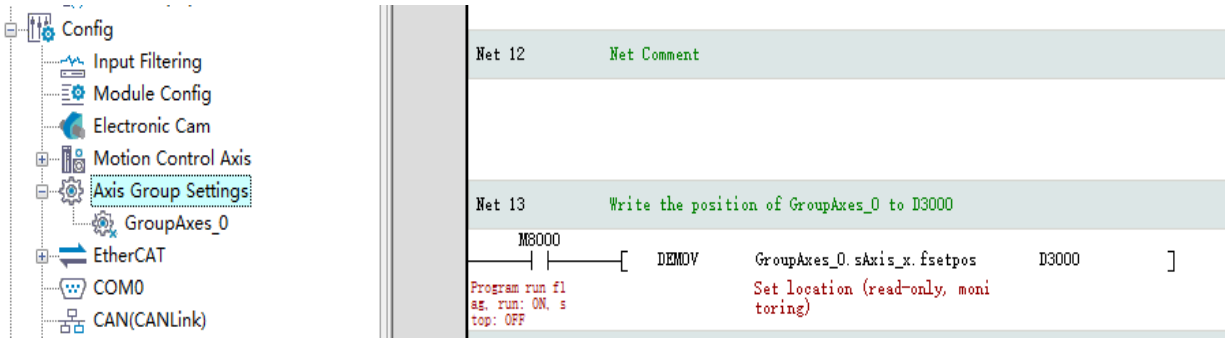
Name	Type	Description
wAxisID	INT16	Axis ID
wState	INT16	The status of an axis' PLCOpen state machine 0: PowerOff 1: ErrorStop 2: Stopping 3: StandStill 4: DiscreteMotion 5: ContinuousMotion 7: Homing 8: SynchronizedMotion
wErrorCode	INT16	Axis fault code
fsetpos	REAL	Position reference
factpos	REAL	Feedback position
fsetvel	REAL	Speed reference
factvel	REAL	Feedback speed

This system variable exists in the axis group _sMCGROUP_INFO and is used to represent the state of individual axes within the axis group, as shown in the following figure.



	Element Name	Data Type	Display Format	Comment
1	GroupAxes_0	_sMCGROUP_INFO		
2	wRingPos	INT	Dec	Configuration Number (Read Only, Monitoring)
3	wGroupID	INT	Dec	Axis Group Number (Read Only, Monitoring)
4	wState	INT	Dec	Status (read-only, monitoring)
5	wErrorCode	INT	Dec	Failure Code (Read-Only, Monitoring)
6	bMotionState	BOOL	Bin	Exercise Status (Read Only, Monitoring)
7	bHaltValid	BOOL	Bin	Pause state (read-only, monitoring)
8	wBufNum	INT	Dec	Number of buffers (read-only, monitoring)
9	sAxis_x	_sGROUPAXIS_INFO		X-axis status (read-only, monitoring)
10	wAxisID	INT	Dec	Axis number (read-only, monitoring)
11	wState	INT	Dec	Status (read-only, monitoring)
12	wErrorCode	INT	Dec	Failure Code (Read-Only, Monitoring)
13	fsetpos	REAL	Dec	Set location (read-only, monitoring)
14	factpos	REAL	Dec	Feedback location (read-only, monitoring)
15	fsetvel	REAL	Dec	Set Speed (Read Only, Monitor)
16	factvel	REAL	Dec	Feedback speed (read-only, monitoring)
17	sAxis_y	_sGROUPAXIS_INFO		Y-axis status (read-only, monitoring)
18	wAxisID	INT	Dec	Axis number (read-only, monitoring)
19	wState	INT	Dec	Status (read-only, monitoring)
20	wErrorCode	INT	Dec	Failure Code (Read-Only, Monitoring)
21	fsetpos	REAL	Dec	Set location (read-only, monitoring)
22	factpos	REAL	Dec	Feedback location (read-only, monitoring)
23	fsetvel	REAL	Dec	Set Speed (Read Only, Monitor)
24	factvel	REAL	Dec	Feedback speed (read-only, monitoring)
25	sAxis_z	_sGROUPAXIS_INFO		z-axis status (read-only, monitoring)
26	wAxisID	INT	Dec	Axis number (read-only, monitoring)
27	wState	INT	Dec	Status (read-only, monitoring)
28	wErrorCode	INT	Dec	Failure Code (Read-Only, Monitoring)
29	fsetpos	REAL	Dec	Set location (read-only, monitoring)
30	factpos	REAL	Dec	Feedback location (read-only, monitoring)
31	fsetvel	REAL	Dec	Set Speed (Read Only, Monitor)
32	factvel	REAL	Dec	Feedback speed (read-only, monitoring)
33	sAxis_a	_sGROUPAXIS_INFO		Auxiliary Axis Status (Read Only, Monitoring)
34	wAxisID	INT	Dec	Axis number (read-only, monitoring)
35	wState	INT	Dec	Status (read-only, monitoring)
36	wErrorCode	INT	Dec	Failure Code (Read-Only, Monitoring)
37	fsetpos	REAL	Dec	Set location (read-only, monitoring)
38	factpos	REAL	Dec	Feedback location (read-only, monitoring)

For example, write the position reference of the X-axis into the D3000 in the PLC, as shown in the following figure.



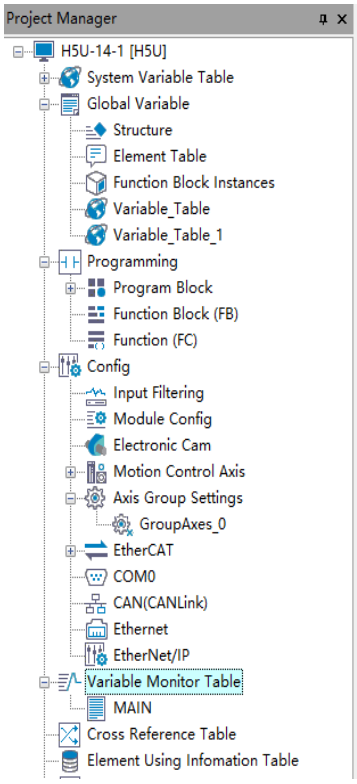
18.5.2 Status of Axis Group _sMCGROUP_INFO

Name	Type	Description
wRingPos	INT16	Axis group number
wGroupID	INT16	Axis number
wState	INT16	Axis group status • 0: Init. The axis configuration in the axis group is not completed. • 1: Disabled. Not all axes in the axis group are enabled. • 2: Single Stop. An axis in the axis group calls the instruction MC_Gtop.

Name	Type	Description
		• 3: Single Homing. An axis in the axis group calls the instruction MC_Home. • 4: Single Motion. An axis in the axis group calls single-axis motion instructions such as MC_MoveAbsolute. • 5: ErrorStop. An axis in the axis group is in a fault state. • 6: StandStill. All axes in the axis group are in the StandStill state. • 7: Stopping. The MC_GroupStop instruction is called repeatedly. • 8: Synchronous Motion. A linear interpolation or circular interpolation instruction is called.
wErrorCode	INT16	Axis fault code
bMotionState	BOOL	Motion status TRUE: Moving FALSE: Not moving
bHaltValid	BOOL	Halt status TRUE: Halt applied FALSE: Halt not applied
wBufNum	INT16	The number of buffered curves
sAxis_x	_sGROUPAXIS_INFO	The status of the X-axis
sAxis_y	_sGROUPAXIS_INFO	The status of the Y-axis
sAxis_z	_sGROUPAXIS_INFO	The status of the Z-axis
sAxis_a	_sGROUPAXIS_INFO	The status of the auxiliary axis
Setvel	REAL	Speed reference In linear interpolation mode, it indicates the interpolation speed of a space straight line. In circular interpolation mode, it indicates the linear speed of a circular arc.
fSetacc_dec	REAL	Acceleration/deceleration rate reference It indicates the change rate of setvel.
fSetvel_buf	REAL	The speed reference of a buffered curve linear interpolation mode, it indicates the interpolation speed of a space straight line. In circular interpolation mode, it indicates the linear speed of a circular arc.
fSetacc_dec_buf	REAL	The acceleration/deceleration reference of a buffered curve It indicates the change rate of fSetvel_buf.
fSetdis	REAL	Distance reference In linear interpolation mode, it indicates the distance at which a space straight line moves after the instruction is executed.

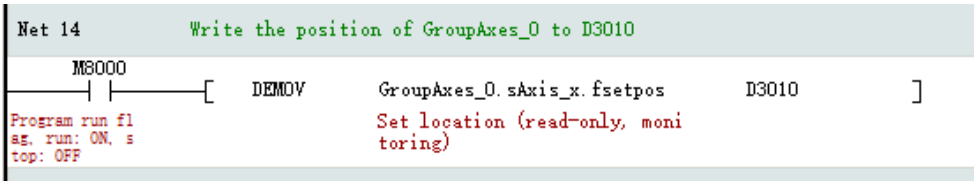
Name	Type	Description
		In circular interpolation mode, it indicates the length of a circular arc in which the circular arc moves after the instruction is executed.
fLeftdis	REAL	Left distance In linear interpolation mode, it indicates the left distance for this section of a space straight line after the instruction is executed. In circular interpolation mode, it indicates the length of a space circular arc left after the instruction is executed.
fCenter_x	REAL	The coordinates of point X at the center of a circular arc during circular interpolation
fCenter_y	REAL	The coordinates of point Y at the center of a circular arc during circular interpolation
fCenter_z	REAL	The coordinates of point Z at the center of a circular arc during circular interpolation
fRadius	REAL	The radius of a circular arc during circular interpolation
fStartAng	REAL	The start angle during circular interpolation
fSetAng	REAL	The motion angle during circular interpolation

This system variable is used to indicate the status of the entire axis group, as shown in the following figure.



	Element Name	Data Type	Display Format	Comment
1	GroupAxes_0	_sMCGROUP_INFO		
2	wRingPos	INT	Dec	Configuration Number (Read Only, Monitoring)
3	wGroupID	INT	Dec	Axis Group Number (Read Only, Monitoring)
4	wState	INT	Dec	Status (read-only, monitoring)
5	wErrorCode	INT	Dec	Failure Code (Read-Only, Monitoring)
6	bMotionState	BOOL	Bin	Exercise Status (Read Only, Monitoring)
7	bHaltValid	BOOL	Bin	Pause state (read-only, monitoring)
8	wBufNum	INT	Dec	Number of buffers (read-only, monitoring)
9	sAxis_x	_sGROUPAXIS_INFO		X-axis status (read-only, monitoring)
10	wAxisID	INT	Dec	Axis number (read-only, monitoring)
11	wState	INT	Dec	Status (read-only, monitoring)
12	wErrorCode	INT	Dec	Failure Code (Read-Only, Monitoring)
13	fsetpos	REAL	Dec	Set location (read-only, monitoring)
14	factpos	REAL	Dec	Feedback location (read-only, monitoring)
15	fsetvel	REAL	Dec	Set Speed (Read Only, Monitor)
16	factvel	REAL	Dec	Feedback speed (read-only, monitoring)
17	sAxis_y	_sGROUPAXIS_INFO		Y-axis status (read-only, monitoring)
18	wAxisID	INT	Dec	Axis number (read-only, monitoring)
19	wState	INT	Dec	Status (read-only, monitoring)
20	wErrorCode	INT	Dec	Failure Code (Read-Only, Monitoring)
21	fsetpos	REAL	Dec	Set location (read-only, monitoring)
22	factpos	REAL	Dec	Feedback location (read-only, monitoring)
23	fsetvel	REAL	Dec	Set Speed (Read Only, Monitor)
24	factvel	REAL	Dec	Feedback speed (read-only, monitoring)
25	sAxis_z	_sGROUPAXIS_INFO		z-axis status (read-only, monitoring)
26	wAxisID	INT	Dec	Axis number (read-only, monitoring)
27	wState	INT	Dec	Status (read-only, monitoring)
28	wErrorCode	INT	Dec	Failure Code (Read-Only, Monitoring)
29	fsetpos	REAL	Dec	Set location (read-only, monitoring)
30	factpos	REAL	Dec	Feedback location (read-only, monitoring)
31	fsetvel	REAL	Dec	Set Speed (Read Only, Monitor)
32	factvel	REAL	Dec	Feedback speed (read-only, monitoring)
33	sAxis_a	_sGROUPAXIS_INFO		Auxiliary Axis Status (Read Only, Monitoring)

For example, write the X-axis coordinates of the center of an axis group to D3010, as shown in the following figure.

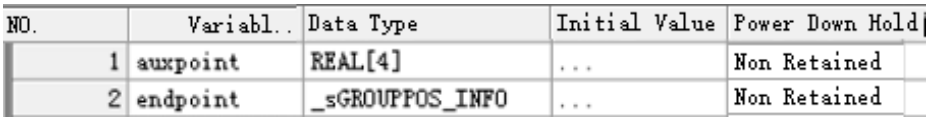


18.5.3 Axis Target Position within Axis Group_sGROUPPOS_INFO

Name	Type	Description
px	REAL	The position of the X-axis
py	REAL	The position of the Y-axis
pz	REAL	The position of the Z-axis
pa	REAL	The position of the auxiliary axis

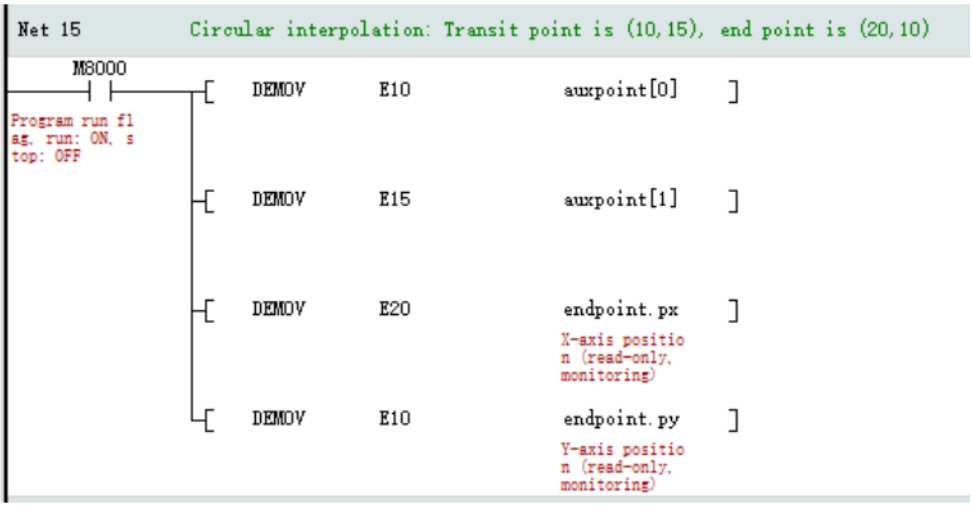
This structure sets the target position of a circular arc as an input parameter to MC_MoveCircular.

1. Create a global variable, as shown in the following figure.

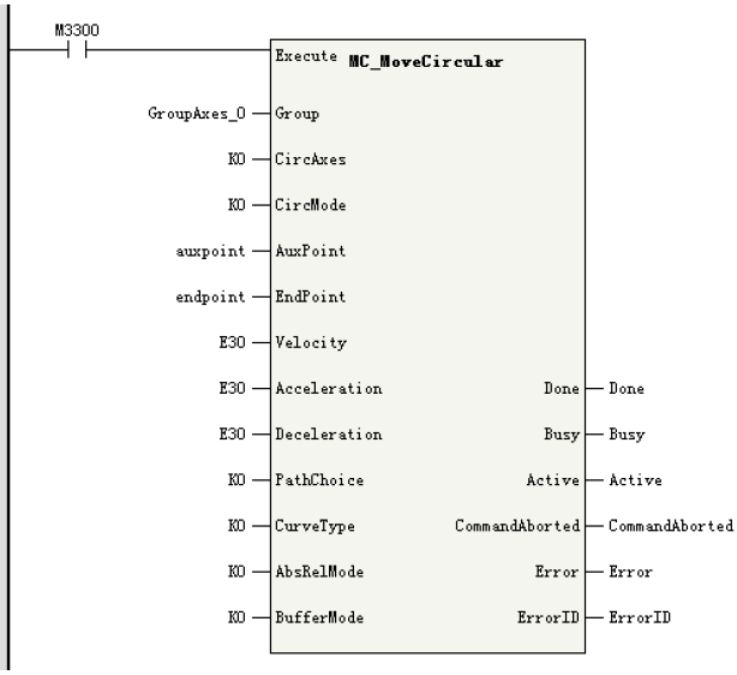


NO.	Variabl..	Data Type	Initial Value	Power Down Hold
1	auxpoint	REAL[4]	...	Non Retained
2	endpoint	_sGROUPPOS_INFO	...	Non Retained

2. Assign values to the global variable, as shown in the following figure.



3. Call the instruction MC_MoveCircular, as shown in the following figure.



18.6 Fault Codes

When a fault occurs during use of the interpolation functions, refer to the fault codes listed in the following table for troubleshooting .

Fault code	Description	Solution
9400	Maximum axis group quantity exceeded	Check whether the number of axis groups is greater than 8.
9401	An axis in the axis group is faulty.	Check whether an axis in the axis group has entered the ErrorStop state. Locate the fault based on the fault code of each axis.
9402	The number of buffered interpolation instructions is greater than 8.	Check whether the number of buffered interpolation instructions is greater than 8.
9403	Axis reused	Locate the reused axis and replace it with an unused axis.
9404	Failed to create axis group	The x-axis and y-axis cannot be empty. Check whether the x-axis or y-axis does not exist or is not specified.
9405	The specified z-axis does not exist.	Check whether the axis specified by AxisID_z exists.
9406	Specified auxiliary axis non-existent	Check whether the axis specified by AxisID_a exists.
9407	Axis group ID duplicated	Check whether GroupID is duplicate.

Fault code	Description	Solution
9408	Failed to configure the axis.	Check whether any axis in the axis group fails to be configured. If yes, check whether the board software and the background match.
9409	The axis ID is less than 0.	Check whether the ID of an axis in the axis group is less than 0.
9410	The axis group is not released because the same MC_SetAxesGroup instruction is triggered repeatedly in a short time period.	Do not re-trigger the MC_SetAxesGroup instruction while its Busy signal output is still active.
9411	The MC_GroupStop instruction is aborted.	Check whether an instruction with higher priority is called while the MC_GroupStop instruction is still active.
9412	The value of CircAxes of the circular interpolation instruction is out of range.	Check whether the value of CircAxes of the circular interpolation instruction is out of range.
9413	The value of CircMode of the circular interpolation instruction is out of range.	Check whether the value of CircMode of the circular interpolation instruction is out of range.
9414	The value of PathChoice of the circular interpolation instruction is out of range.	Check whether the value of PathChoice of the circular interpolation instruction is out of range.
9415	The value of StopMode of the stop instruction is out of range.	Check whether the value of StopMode of the stop instruction is out of range.
9416	The x-axis is set to ring mode.	Do not set the motion control axis to the ring mode in an interpolation instruction.
9417	The y-axis is set to ring mode.	Do not set the motion control axis to the ring mode in an interpolation instruction.
9418	The z-axis is set to ring mode.	Do not set the motion control axis to the ring mode in an interpolation instruction.
9419	The auxiliary axis is set to ring mode.	Do not set the motion control axis to the ring mode in an interpolation instruction.
9420	The circular interpolation instruction is triggered repeatedly.	Do not re-trigger the same circular interpolation instruction while its Busy signal output is still active.

Fault code	Description	Solution
9421	The linear interpolation instruction is triggered repeatedly.	Do not re-trigger the same linear interpolation instruction while its Busy signal output is still active.
9422	Failed to obtain the axis group.	Check whether the axis group specified by GroupID has been created by calling MC_SetAxesGroup.
9423	Failed to configure the axis.	Check whether an instruction is triggered when axis configuration is not completed. Check whether communication state of all axes in the axis group is Axis ready.
9424	An axis is disabled.	Do not call the interpolation instruction when any axis is in Disabled state.
9425	Axis in execution of single-axis motion instruction	Do not call the interpolation instruction when any axis is executing single-axis motion instructions and not in StandStill state.
9426	An axis is in Stopping state.	Do not call the interpolation instruction when any axis is in Stopping state after executing the MC_Stop instruction.
9427	The axis group is in Stopping state.	Do not call the interpolation instruction while the MC_GroupStop instruction is still active.
9428	An axis is in Homing state.	Do not call the interpolation instruction when any axis is in Homing state after executing the MC_Home instruction.
9429	An axis is executing the position setting instruction.	Do not call the interpolation instruction when any axis is setting the current position by executing the MC_SetPosition instruction.
9430	An axis is in commissioning state.	Do not call the interpolation instruction when any axis is in commissioning state.
9431	An axis enters the commissioning state during interpolation, which aborts instruction execution of other axes.	Check whether any axis enters the commissioning state during interpolation.
9432	Failed to request the memory.	Check whether the memory runs out. Contact the manufacturer.
9433	The target velocity is 0 or less than 0.	Ensure that the target velocity of the instruction is greater than 0.
9434		

Fault code	Description	Solution
	The target acceleration is 0 or less than 0.	Ensure that the target acceleration of the instruction is greater than 0.
9435	The target deceleration is 0 or less than 0.	Ensure that the target deceleration of the instruction is greater than 0.
9436	The curve type setting is out of range.	Check whether the curve type is set to a value other than the T-shaped curve for the interpolation instruction.
9437	AbsRelMode is set incorrectly.	Check whether the parameter is set to a value other than the absolute positioning and relative positioning modes.
9438	BufferMode is set incorrectly.	Check whether the value of BufferMode is out of range.
9439	InsertMode is set incorrectly.	Check whether the value of InsertMode is proper.
9440	An axis stops due to a fault.	Locate the faulty axis and rectify the fault based on the fault code.
9441	The MC_GroupStop instruction is called repeatedly.	Do not re-trigger an MC_GroupStop instruction or call other MC_GroupStop instructions while an MC_GroupStop instruction is still active.
9442	Data buffer area not empty	Contact Inovance for technical support.
9443	Not a circle	-
9444	The start, end, and border points in the circular interpolation instruction are the same point, and no circle can be drawn.	Check the input parameters of the circular interpolation instruction and ensure that the start, end, and border points can form a circle.
9445	The instruction buffer area is full.	Contact Inovance for technical support.
9446	The velocity of the x-axis exceeds the maximum allowable velocity.	Ensure that the target velocity of the x-axis is not greater than the maximum allowable velocity.
9447	The velocity of the y-axis exceeds the maximum allowable velocity.	Ensure that the target velocity of the y-axis is not greater than the maximum allowable velocity.
9448	The velocity of the z-axis exceeds the maximum allowable velocity.	Ensure that the target velocity of the z-axis is not greater than the maximum allowable velocity.

Fault code	Description	Solution
9449	The velocity of the auxiliary axis exceeds the maximum allowable velocity.	Ensure that the target velocity of the auxiliary axis is not greater than the maximum allowable velocity.
9450	Failed to obtain the number of axis groups.	Update the software tool to the latest version.
9451	Internal fault	Contact the manufacturer.
9452	The instruction is called when the axis is in StandStill state.	Do not call this instruction when the axis is StandStill state.
9453	Maximum velocity exceeded	Ensure that the target velocity of the instruction is not greater than the maximum velocity specified on the axis group configuration interface.
9454	The maximum allowable acceleration (deceleration) is exceeded.	Ensure that the target acceleration (deceleration) of the instruction is not greater than the maximum acceleration (deceleration) specified on the axis group configuration interface.
9455	The axis group becomes faulty due to an error reported by the linear interpolation instruction.	Identify the first linear interpolation instruction that reports the error and troubleshoot the fault based on the fault code.
9456	The axis group becomes faulty due to an error reported by the circular interpolation instruction.	Identify the first circular interpolation instruction that reports the error and troubleshoot the fault based on the fault code.
9457	The axis group becomes faulty due to an error reported by the axis group stop instruction.	Identify the first axis group stop instruction that reports the error and troubleshoot the fault based on the fault code.
9458	The axis group becomes faulty due to an error reported by the axis group pause instruction.	Identify the first axis group pause instruction that reports the error and troubleshoot the fault based on the fault code.

19 Bus Encoder Axis

19.1 Introduction

Bus encoder axes support a maximum input pulse frequency of 200 kHz is supported, use the GR10-2HCE module as the driver, and can count three forms of pulses: phase A/B, pulse+direction, and CW/CCW.

Up to eight bus encoder axes can be configured. One GR10-2HCE module has two channels, each of which can be assigned for one axis.

Bus encoder axes can also work with the DI and DO terminals of the GR10-2HCE module to implement position presetting, probe, gating, and comparison output.

The feedback position of the GR10-2HCE module in pulses is sent to the PLC through PDOs, and the feedback position within the PLC is finally provided to users in the form of REAL data type through the conversion of gear ratios, which can also be used as the master axis of a cam or gear.

NOTE

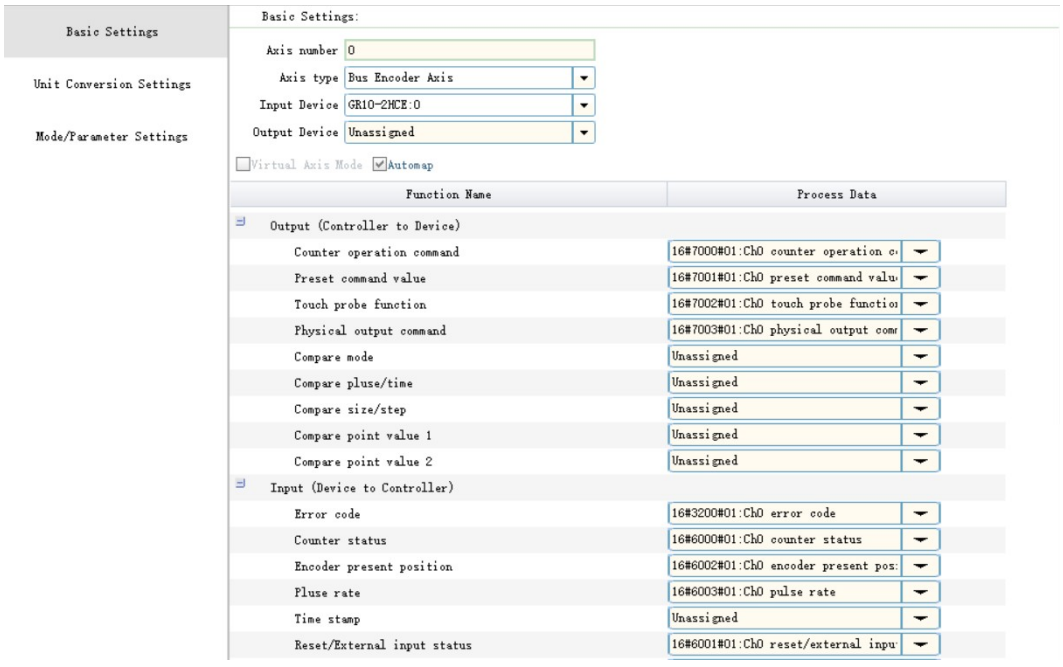
The software of the GR10-2HCE module must be version V2.2.0.0 or later versions.

19.2 Software Configuration

19.2.1 Basic Settings

Bus encoder axes represent a category of motion control axes that require the user to enable EtherCAT communication and add the GR10-2HCE module before enabling a bus encoder axis .

The configuration interface of a bus encoder axis is shown in the following figure.



- Axis No: Each bus encoder axis has an axis number, which is the unique identifier of the encoder axis and is automatically assigned when the configuration is established.
- Axis type: Bus encoder axis.

- Input device: H5U series PLC an example, a bus encoder axis needs to be used with the GR10-2HCE module. Each GR10-2HCE module supports two counting channels. You can select a channel as required during device selection.
- Output device: Not supported
- Automap: After "Automap" is ticked, the I/O variable of an axis and the PDO of the GR10-2HCE module will be automatically associated. If this option is not ticked, you can manually configure the mapping.

Tag mapping

The correspondence between the I/O variables of a bus encoder axis and the PDO of the GR10-2HCE module is shown in the following table.

For details on output variable mapping, see the following table.

Variable	Channel 0 Object dictionary	Channel 1 Object dictionary	Description
Counter operation command	7000h:1	7000h:2	Encoder control word
Preset command value	7001h:1	7001h:2	Preset position
Touch probe function	7002h:1	7002h:2	Probe control word
Physical output command	7003h:1	7003h:2	DO terminal control word
Compare mode	7003h:3	7003h:11	Comparison output mode
Compare pulse/time	7003h:5	7003h:13	Comparison output pulses/time
Compare size/step	7003h:6	7003h:14	Comparison output array length/step
Compare point value 1	7008h:1	7009h:1	Comparison output value 1
Compare point value 2	7008h:2	7009h:2	Comparison output value 2

For details on input variable mapping, see the following table.

Variable	Channel 0 Object dictionary	Channel 1 Object dictionary	Description
Error code	3200h:1	3200h:2	Fault code
Counter status	6000h:1	6000h:2	Counter status
Encoder present position	6002h:1	6002h:2	

Variable	Channel 0 Object dictionary	Channel 1 Object dictionary	Description
			Encoder feedback position
Pulse rate	6003h:1	6003h:2	Pulse frequency
Time stamp	6009h:1	6009h:2	Time stamp
Physical input status	6001h:1	6001h:2	DI terminal status
Touch probe status	6004h:1	6004h:2	Probe status
Touch probe pos 1 pos value	6005h:1	6005h:2	Touch probe 1 positive edge
Touch probe pos 1 neg value	6006h:1	6006h:2	Touch probe 1 falling edge
Touch probe pos 2 pos value	6007h:1	6007h:2	Touch probe 2 positive edge
Touch probe pos 2 neg value	6008h:1	6008h:2	Touch probe 2 falling edge
Physical output status	600Eh:1	600Eh:2	DO terminal status
Compare error code	6003h:3	6003h:5	Comparison output error code
Current compare number/ position	6003h:7	6003h:9	Comparison output current group subscript/ position

19.2.2 Unit Conversion

The following table lists parameters that need to be set for unit conversion.

Parameter Name	Function
The number of pulses per revolution of the motor/ encoder	Set the number of pulses required for the motor to rotate one turn according to the encoder resolution.
With gear change mechanisms	Specify whether gear change mechanisms are in use or not.
Amount of movement per revolution of the motor/ encoder	The workpiece movement amount per revolution of the motor when no gear change mechanism is in use

Parameter Name	Function
Amount of movement per revolution of the worktable	The workpiece movement amount per revolution of the worktable when gear change mechanisms are in use
Gear ratio on the motor/encoder side	Set a gear ratio on the motor/encoder side.
Gear ratio on the worktable side	Set a gear ratio on the worktable side.

The module GR10-2HCE counts in pulses, while the encoder axis instructions use common measurement units in operation, such as millimeters, degrees, and inches, which are known as user units. In this example, the unit "degree" is used. The conversion between the two units is divided into the following modes:

1. Do not use gearbox.
   When the gear change mechanisms are not in use, the conversion equation from user unit to pulse unit is as follows.

$$\text{Number of pulses (unit: pulse)} = \frac{\text{Number of pulses per revolution of the motor/encoder [DINT]}}{\text{Distance per revolution of the workbench [REAL]}} \times \text{Distance (unit: Unit)}$$

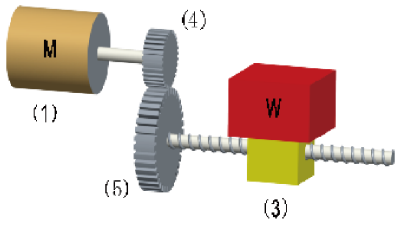
   Take the Inovance 23-bit encoder as an example. The set parameters are as follows.

   Number of pulses per revolution of the motor/encode = 8388608

   Workpiece movement amount per revolution of the motor/encoder = 1

   When the motor rotates by 10 revolutions, the number of pulses counted by the 2HCE module is 83886080, and the encoder axis instruction counts to the position increment of 10 degrees.

2. With gear change mechanisms
   - Typical working condition in linear mode is shown in the following figure.

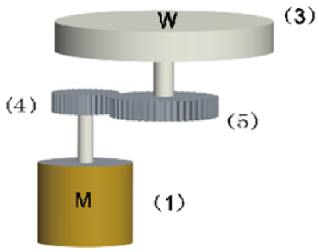


     Where, (1) is the motor/encoder; (3) is the worktable; (4) is the gear ratio denominator; (5) is the gear ratio numerator.

     The calculation equation from user unit to pulse unit is as follows.

$$\text{Number of pulses (unit: pulse)} = \frac{\text{Number of pulses per revolution of the motor/encoder [DINT]} \times \text{Gear ratio numerator [DINT]}}{\text{Distance per revolution of the workbench [REAL]} \times \text{Gear ratio denominator [DINT]}} \times \text{Distance (unit: Unit)}$$

   - Typical working condition in rotary mode is shown in the following figure.



Where, (1) is the motor/encoder; (3) is the worktable; (4) is the gear ratio denominator; (5) is the gear ratio numerator.

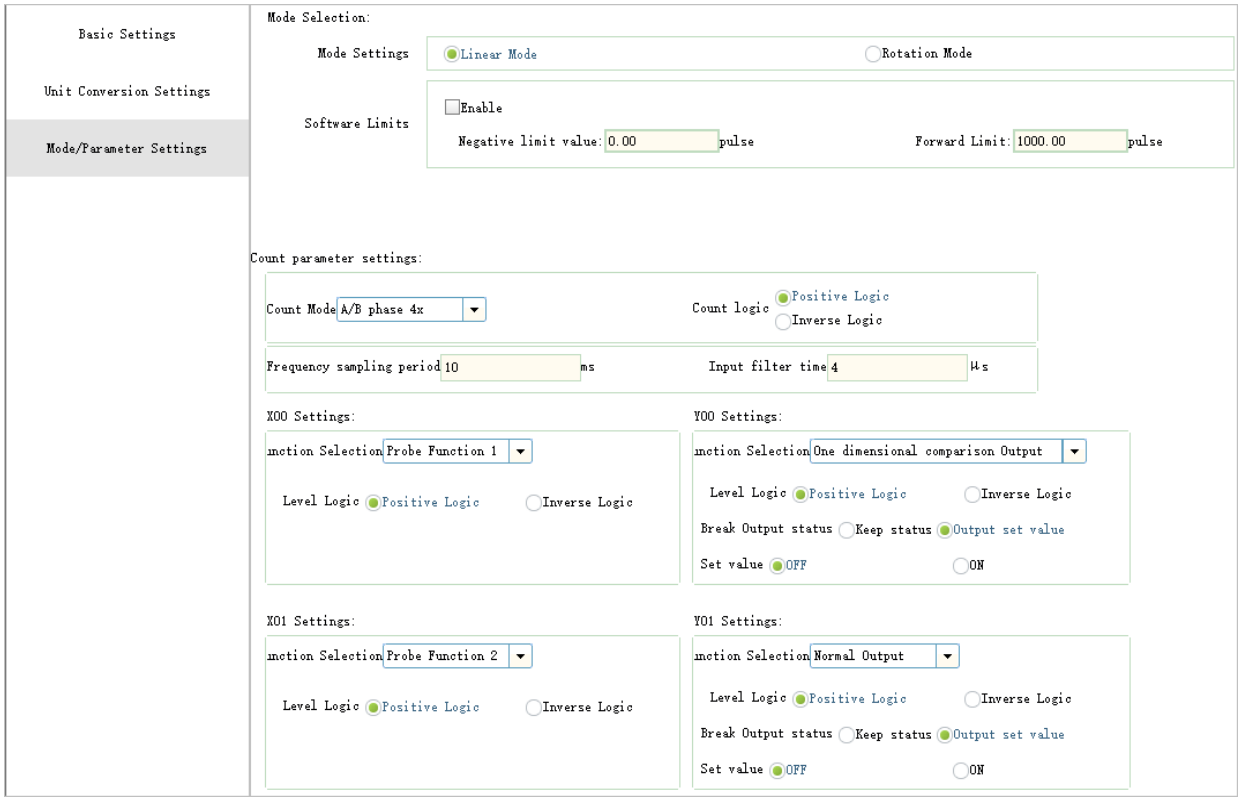
The calculation equation from user unit to pulse unit is as follows.

$$\text{Number of pulses (unit: pulse)} = \frac{\text{Number of pulses per revolution of the motor/encoder [DINT] x Gear ratio numerator [DINT]}}{\text{Distance per revolution of the workbench [REAL] x Gear ratio denominator [DINT]}} \text{ x Distance (unit: Unit)}$$

19.2.3 Mode/Parameter Setting

19.2.3.1 Configuration Interface

The following figure shows the mode and parameter configuration page.

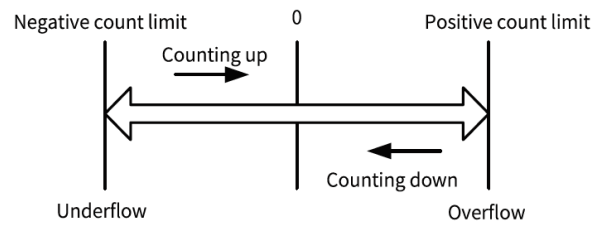


19.2.3.2 Linear/Rotary Mode Selection

Linear mode

You can enable or disable software limits.

When the software limit is enabled, positive and negative limits can be set, and the counter will stop counting when reaching the limits, and display an out-of-limit mark. When the positive limit is reached, the output of the encoder axis system variable bPLimit is valid. When the negative limit is reached, the output of the encoder axis system variable bNLimit is valid.



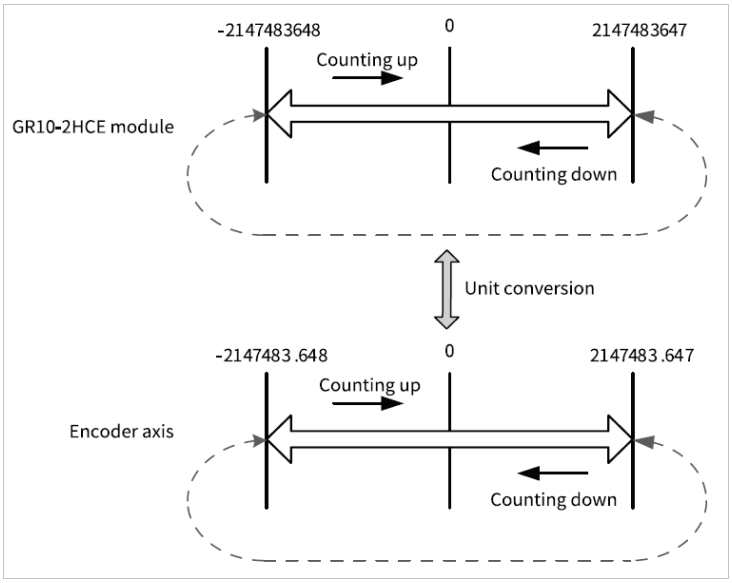
If the software limit is not enabled, the count value of GR10-2HCE ranges between -2147483648 and +2147483647, jumps to -2147483648 when the positive count reaches +2147483647, and jumps to +2147483647 when the negative count reaches -2147483648.

In this case, the range that the bus encoder axis can count needs to be calculated according to the scheme described in Unit Conversion.

Example

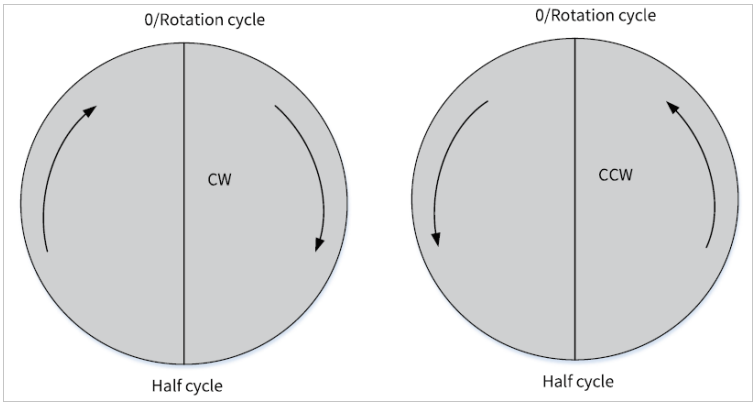
Number of pulses in one turn by motor/encoder = 1000

Amount of movement in one turn by motor/encoder = 1



Rotary mode

In rotary mode, you can set the ring period. The counter counts in a reciprocating cycle between 0 and the ring period.

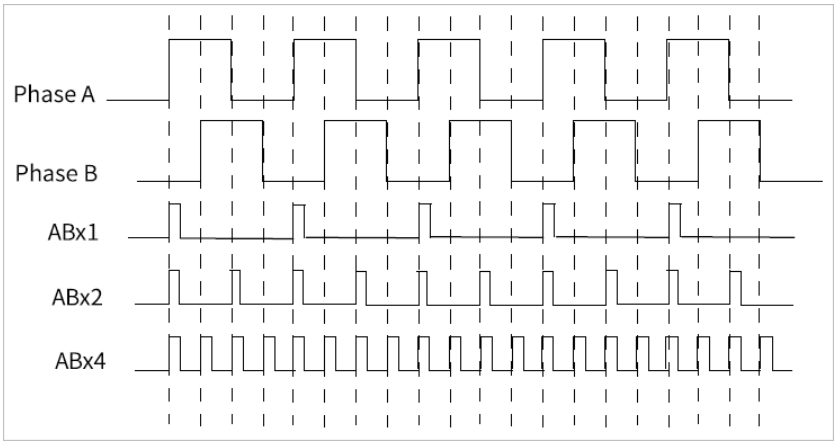


19.2.3.3 Counter Mode Selection

Count mode

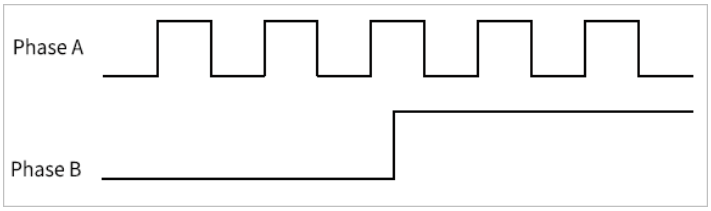
Five count modes are supported: phase A/B 1-frequency multiplication, phase A/B 2-frequency multiplication, phase A/B 4-frequency multiplication, pulse+direction, and CW/CCW.

1. Phase A/B pulse



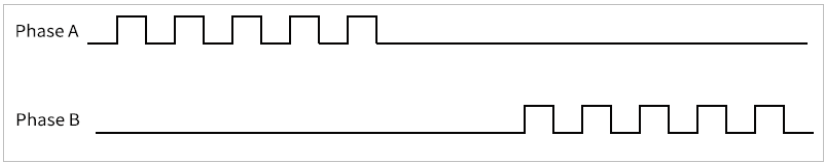
2. Pulse + Direction

   Phase A is used as the counter's pulse input, and phase B is used as the counter's count direction control input.



3. CW/CCW mode

   For CW/CCW mode, under positive logic, the counter counts up on the rising edge that inputs pulse A and counts down on the rising edge that inputs pulse B.



Count logic

Count logic is used to set the logic of the count direction. The logic is as follows.

Pulse mode	Positive logic	Negative logic
Phase A/B	Incremental count if phase A leads phase B Decremental count if phase B leads phase A	Decremental count if phase A leads phase B Incremental count if phase B leads phase A
Pulse + Direction	Incremental count if phase B is at a high level Decremental count if phase B is at a low level	Decremental count if phase B is at a high level Incremental count if phase B is at a low level
CW/CCW	Incremental count if it is phase A pulse input	Decremental count if it is phase A pulse input

Pulse mode	Positive logic	Negative logic
	Decremental count if it is phase B pulse input	Incremental count if it is phase B pulse input

19.2.3.4 Frequency Sampling Period

It is used to set the calculation period of the pulse frequency.

19.2.3.5 Input Filter Time

It is used to set the filter time of DI terminals and ABZ input signals.

19.2.3.6 Input Terminal Function Selection

Each counting channel can be independently configured with four DI terminals. The functions that can be allocated for each terminal are shown in the following table.

Terminal	Optional Function	Default
Xn0	0: Common input 1: Probe function 1 3: Reset 4: Preset 5: Gating	Probe function 1
Xn1	0: Common input 2: Probe function 2 3: Reset 4: Preset 5: Gating	Probe function 2
Xn2	0: Common input 3: Reset 4: Preset 5: Gating	Common input
Xn3	0: Common input 3: Reset 4: Preset 5: Gating	Common input

- Common input
  When a terminal is allocated with the common input function, its status can be obtained through the system variable iDIStatus or the instruction ENC_ReadStatus.

- Probe function
  When a terminal is allocated with the probe function, you need to call the instruction ENC_TouchProbe to implement the probe function. For details, see the explanation of this instruction.
- Preset function
  When a terminal is allocated with the preset function, you need to call the instruction ENC_Preset to implement the counter preset function. When the input signal of the DI terminal is active, the value of the counter will be set to the preset value of parameter Position of the instruction ENC_Preset.

  NOTE

  If multiple terminals are configured with a preset function, the preset function is activated when one of the input terminals receives an active signal.

- Gating function
  If the DI terminal signal is set as a gating signal, the gating function is enabled, in which case the counter is enabled to start counting only after the parameter Enable of ENC_Counter is ON and the gating input signal is active. If the DI terminal signal is not set as a gating signal, and the gating function is not enabled, then the counter starts counting after the parameter Enable of ENC_Counter is ON.

  NOTE

  If multiple terminal input signals are used as gating signals, these gating signals must be active at the same time for the counter to start counting.

19.2.3.7 Output Terminal Function Selection

Each counting channel can be independently configured with three DO terminals. The functions that can be allocated for each terminal are shown in the following table.

Terminal	Optional Function	Default
Yn0	0: Common output 3: One-dimensional comparison output 4: Two-dimensional comparison output (only supported by channel 0)	3: One-dimensional comparison output
Yn1	0: Common output	0: Common output
Yn2	0: Common output	0: Common output

- Common output
  When a DO terminal is set to a common output terminal, its status can be controlled by the instruction ENC_DigitalOutput. See description of the instruction ENC_DigitalOutput for detailed usage.
- One-dimensional comparison output mode
  When a terminal is allocated with the one-dimensional comparison output function, the step comparison output or array comparison output functions can be performed. See instructions ENC_StepCompare and ENC_ArrayCompare for detailed guidance.
- Two-dimensional comparison output mode

When a bus encoder axis is bound to channel 0 of the GR10-2HCE module, the output terminal Y00 can be allocated with the two-dimensional comparison output function. See the instruction ENC_GroupArrayCompare for detailed usage.

19.3 System Variables

_sPoint2D: Structure of coordinate points in a two-dimensional coordinate system, as shown in the following table

Variable	Type	Function description
px	REAL	Coordinate points on X-axis
py	REAL	Coordinate points on Y-axis

_sENC_PDO: PDO of the bus encoder axis, as shown in the following table

Variable	Type	Function description
iControlWord	INT	Control word
iStatusWord	INT	Status word
dPresetPosition	DINT	Preset position
dActualPositionValue	DINT	Feedback position
dActualVelocityValue	DINT	Feedback speed
dTimeStamp	DINT	Time stamp
iDOControlWord	INT	DO terminal control word
iDOStatusWord	INT	DO terminal status word
iCompareMode	INT	Comparison output mode
dCompare_Pluse_Time	DINT	Comparison output pulses/time
dCompare_Size_Step	DINT	Comparison output array length/step
dCompareValue_1	DINT	Comparison output comparison value 1
dCompareValue_2	DINT	Comparison output comparison value 2
dCompareAct_Num_Pos	DINT	Next position/comparison value of comparison output
iCompareErrorCode	INT	Fault code of comparison output
iDIStatus	INT	DI terminal status

Variable	Type	Function description
iTouchProbeFunc	INT	Probe control word
iTouchProbeStatus	INT	Probe status word
dPos1PosValue	DINT	Probe 1 position latched on the rising edge
dPos2PosValue	DINT	Probe 2 position latched on the rising edge
dPos1NegValue	DINT	Probe 1 position latched on the falling edge
dPos2NegValue	DINT	Probe 2 position latched on the falling edge
iErrorCode	INT	Fault code

_sENC_CONFIG: General configuration parameters for encoder axes, as shown in the following table

Variable	Type	Function description
bLimitEnable	BOOL	Limit enable
fUnits	REAL	Gear ratio
iRingMode	INT	Linear/ring mode 0: Linear mode 1: Ring mode
fNegLimitPos	REAL	Negative limit in linear mode
fForLimitPos	REAL	Positive limit in linear mode
fRotationPeriod	REAL	Rotation cycle in rotary mode

_sENC_EXT_AXIS: System variables of the bus encoder axis, as shown in the following table

Variable	Type	Function description
bEnable	BOOL	Encoder counting state OFF: Stop counting ON: Enable counting
bActDir	BOOL	Counting direction OFF: Forward (incremental count) ON: Reverse (decremental count)
bPLimit	BOOL	Reaching the positive limit

Variable	Type	Function description
		OFF: Invalid ON: Valid
bNlimit	BOOL	Reaching the negative limit OFF: Invalid ON: Valid
iConfigAddress	INT	Configuration address
iAxisID	INT	Axis number
fPosition	REAL	Feedback position (user unit)
fVelocity	REAL	Feedback velocity (user unit)
dPosition	DINT	Feedback position (pulse unit)
dFrequency	DINT	Pulse frequency (pulse unit)
iAxisState	INT	Statuses of the axis 0: Stop counting 1: Fault state of the axis 5: Enable counting
iConfigState	INT	Configuration status of the axis 1: Init (axis in the initialization state) 2: Configure finish (reading of configuration data completed) 3: Sync finish (synchronized with EtherCAT tasks) 4: Wait communication (communication with the drive established) 5: Slave ready (initialization completed for the servo drive controlled by axes) 6: Axis ready (communication established)
iAxisError	INT	Axis fault code
iSlaveAxisError	INT	Drive fault code
sCounter	_sENC_CNT	Reservation for compatibility with the local encoder axis

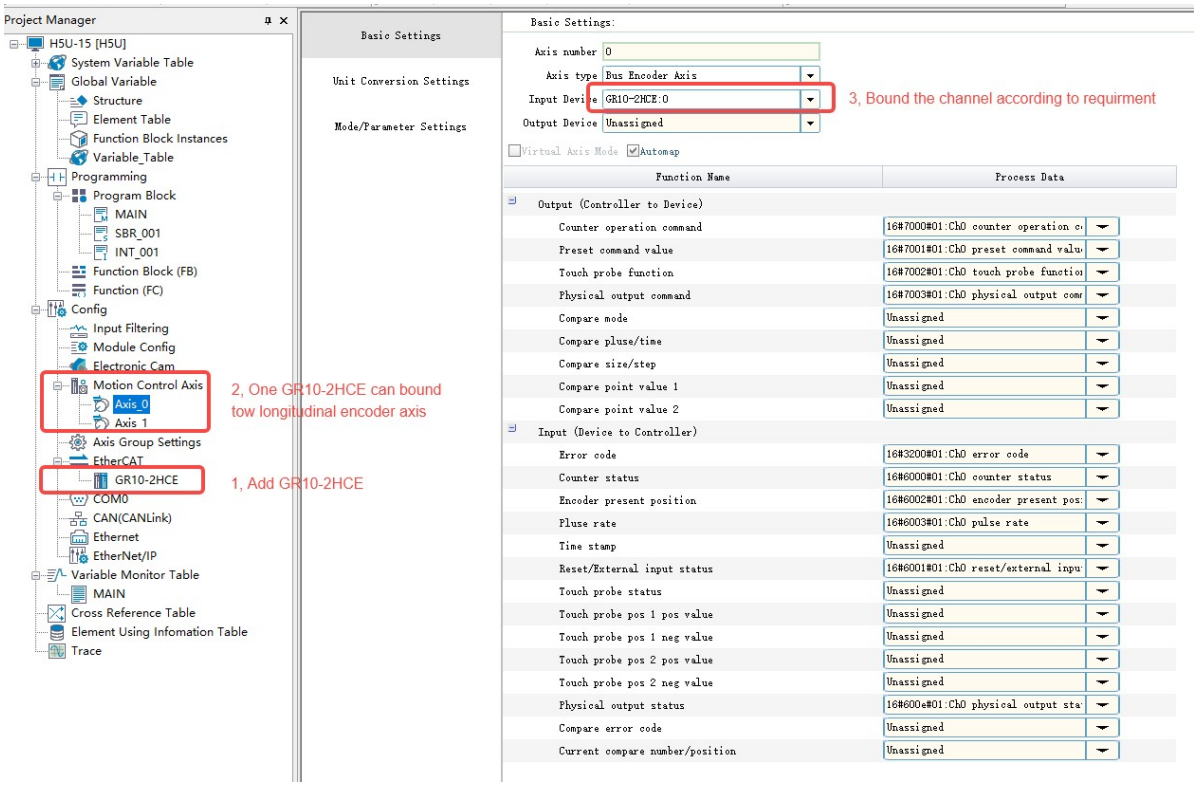
Variable	Type	Function description
sReset	_sENC_RST	Reservation for compatibility with the local encoder axis
sPreset	_sENC_PRESET	Reservation for compatibility with the local encoder axis
sProbe	_sENC_PROBE[2]	Reservation for compatibility with the local encoder axis
sMatch	_sENC_MATCH[2]	Reservation for compatibility with the local encoder axis
sPDO	_sENC_PDO	Parameter value area of PDOs
sConfigure	_sENC_CONFIG	Parameter value area of configuration

19.4 Function Examples

19.4.1 Configuration Setup

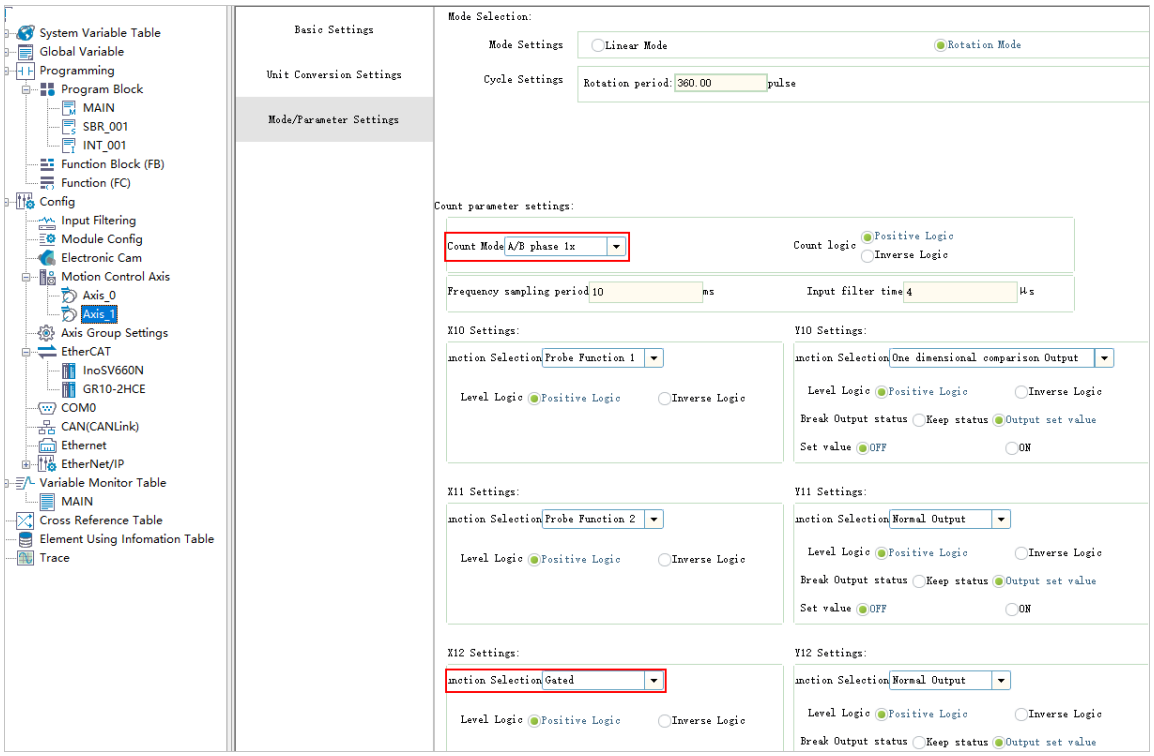
In this example, the hardware required is as follows: one H5U and one GR10-2HCE module with its channels CH0 and CH1 connected to the external phase A/B pulse of fixed frequency.

Create a project and add the GR10-2HCE module to the EtherCAT configuration. Then two bus encoder axes are automatically added. Among them, Axis_0 is automatically bound to channel CH0 of the GR10-2HCE module, while Axis_1 is automatically bound to channel CH1 of the GR10-2HCE module.

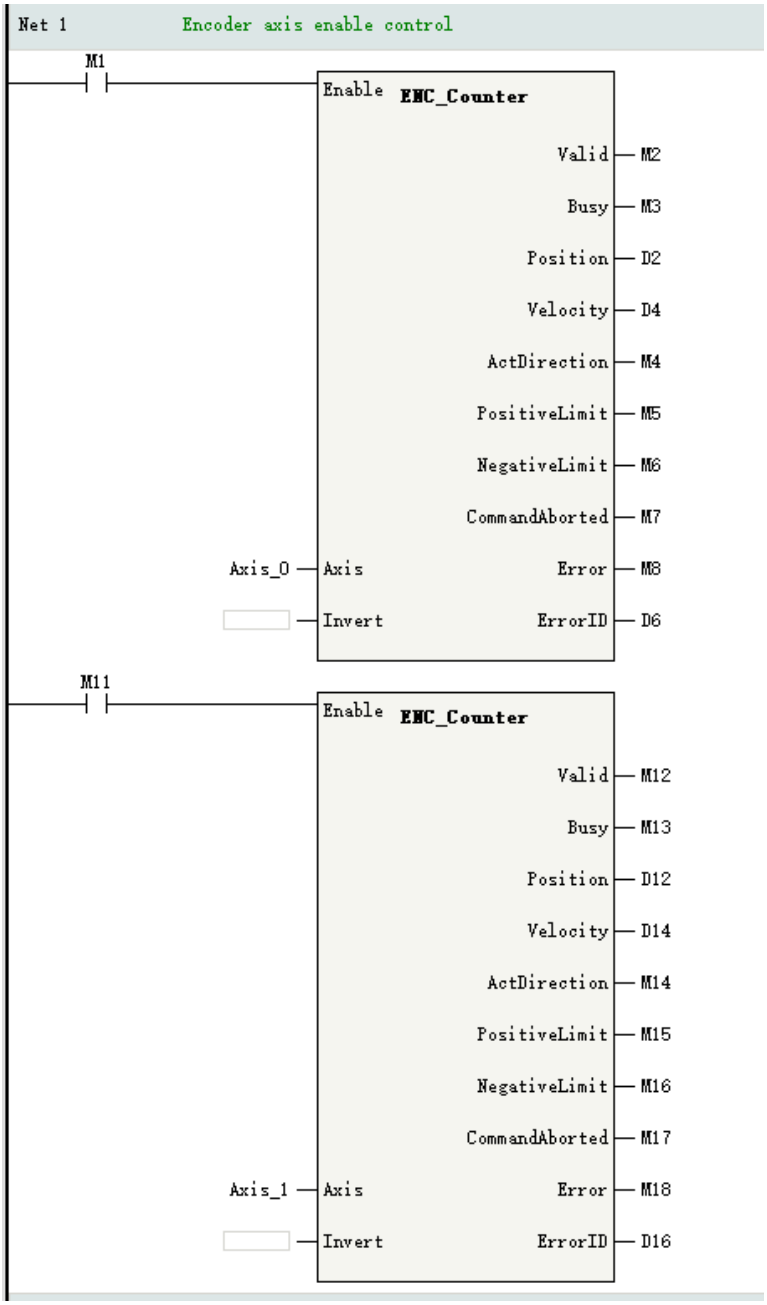


19.4.2 Counter Enable

Set the "Count Mode" to "A/B phase 1x" for both Axis_0 and Axis_1, and select "Gated" in "X12 Settings" for Axis_1.



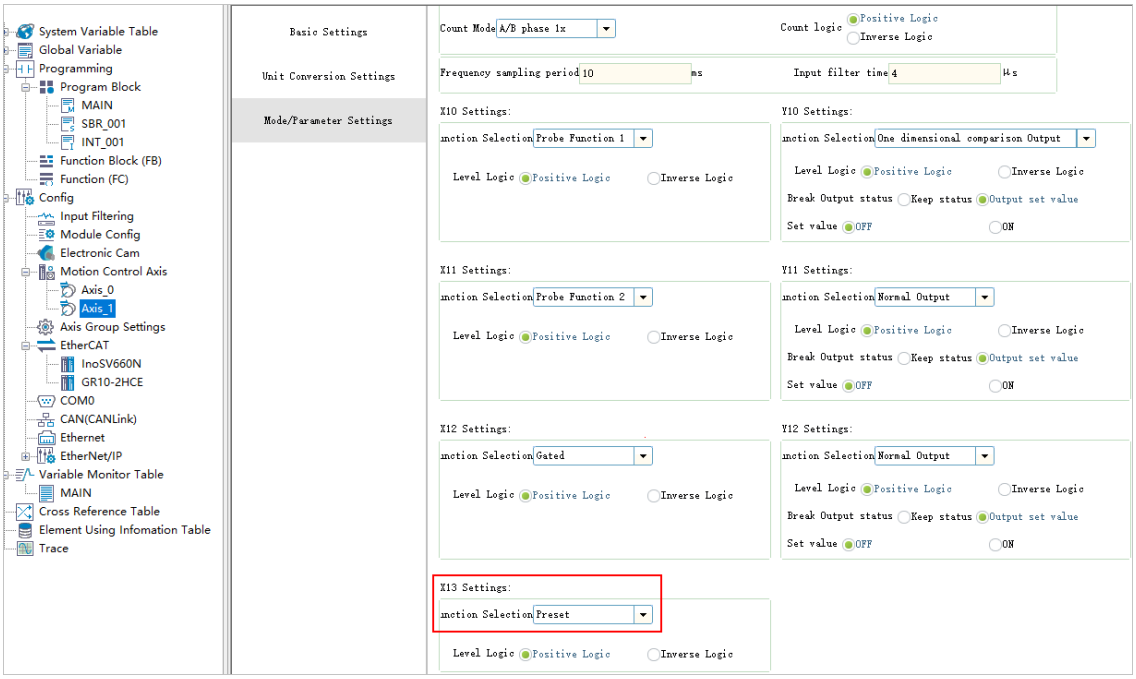
Call the instruction ENC_Counter to enable both bus encoders.

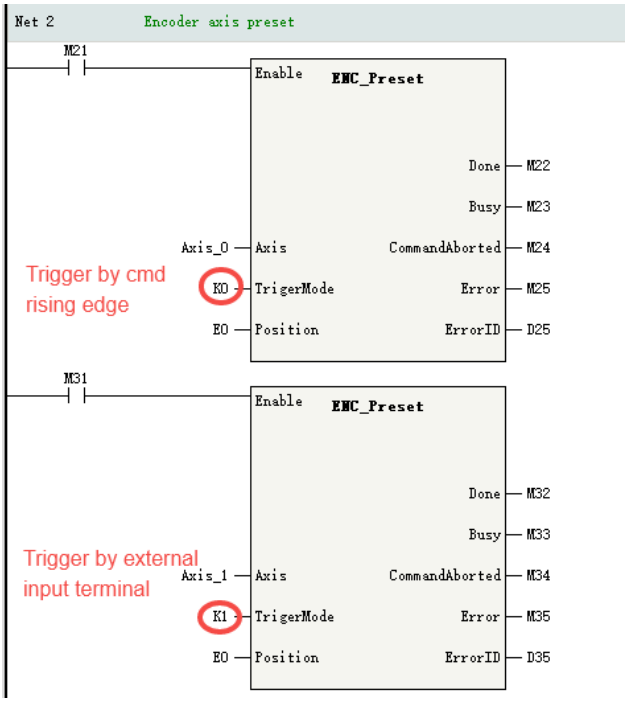


- After M1 is set to ON, M2 and M3 outputs are ON, D2 displays the current position, and the bus encoder axis Axis_0 starts counting.
- After M11 is set to ON, if the X12 input of the gating terminal is OFF, the M13 output of the instruction is ON, the M12 output is OFF, and D12 shows the current position, then the encoder axis Axis_1 does not count. If the X12 input of the gating terminal is ON and the M12 output of the instruction is ON, then the encoder axis Axis_1 starts counting.

19.4.3 Counter Preset

Set X13 as the preset function and use the EMC_Preset instruction to set the axis_1 encoder count value.



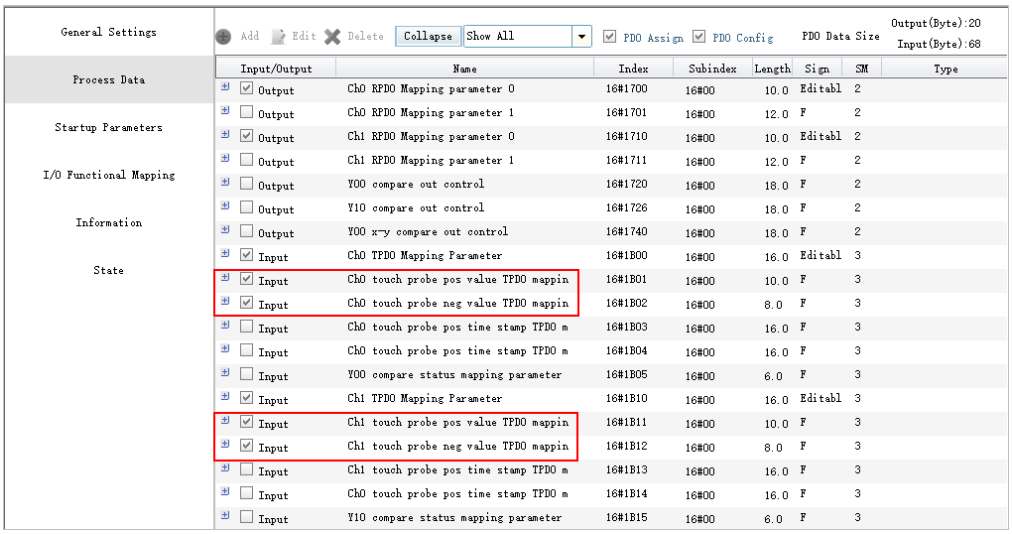


- Set M21 to ON, and the current count value of Axis_0 is set to 0. After the setting is complete, M22 output is ON.
- Set M31 to ON and M33 output to ON. When terminal X13 of GR10-2HCE turns from OFF to ON, the current count value of Axis_1 is set to 0. After the setting is completed, the M32 output is ON.

19.4.4 Probe Functions

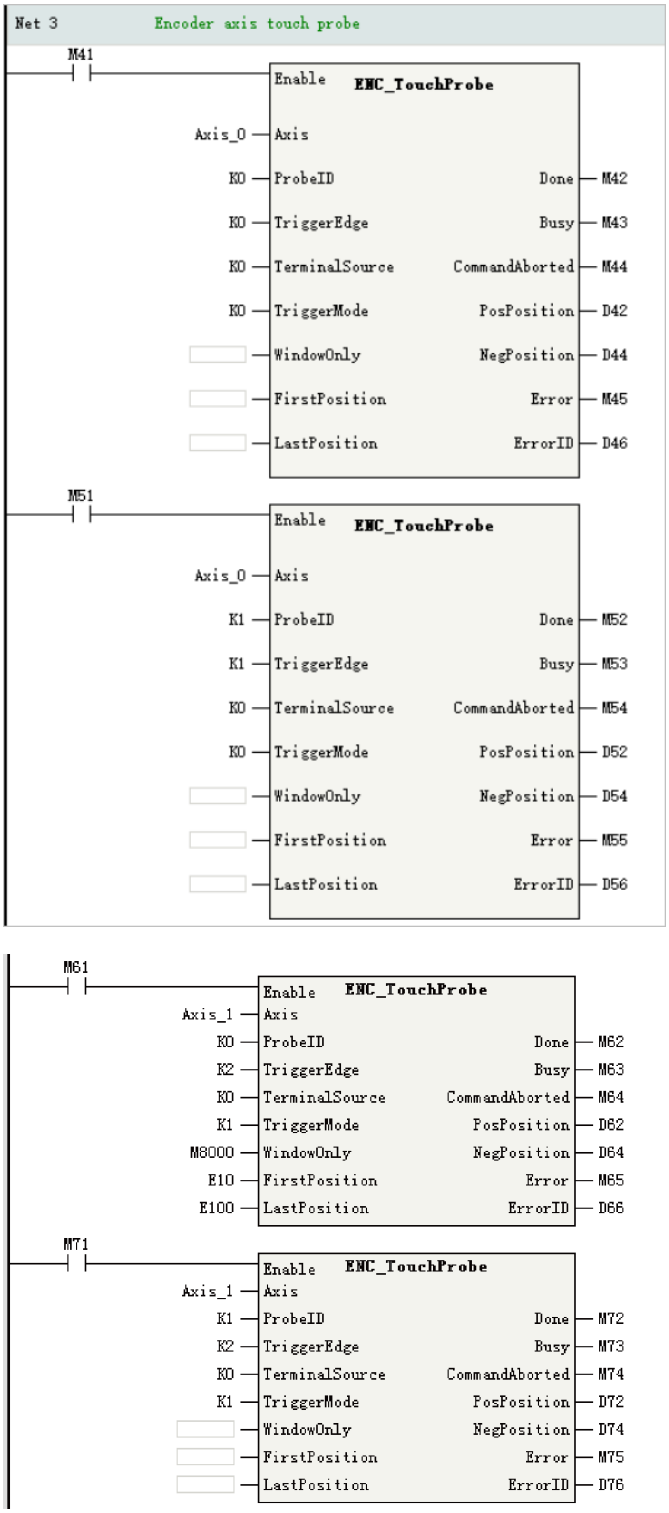
Allocate the probe function 1 and probe function 2 for X00 and X01 in the encoder axis Axis_0 and for X10 and X11 in the encoder axis Axis_1.

Tick the following PDO data on the "Process Data" tab page of GR10-2HCE.



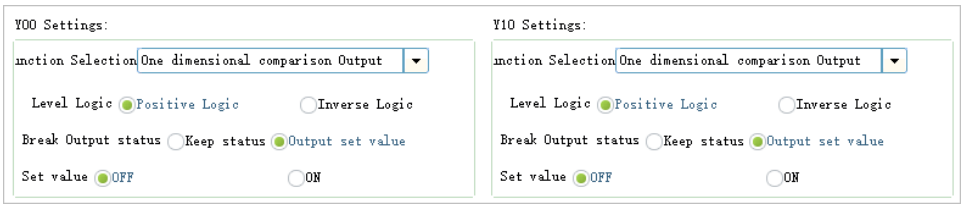
Call the instruction ENC_TouchProbe to control the probe function of the encoder axis. The functions of the four probes are set as follows.

Parameter	Axis_0 probe 1	Axis_0 probe 2	Axis_1 probe 1	Axis_1 probe 2
Probe ID	Probe 1	Probe 2	Probe 1	Probe 2
Trigger edge	Trigger by the rising edge only	Trigger by the falling edge only	Trigger by both rising and falling edges	Trigger by both rising and falling edges
Terminal	DI terminal	DI terminal	DI terminal	DI terminal
Trigger mode	Single trigger	Single trigger	Continuous trigger	Continuous trigger
Window limit	Disabled	Disabled	Enabled	Disabled
Start position	-	-	10	-
End position	-	-	100	-

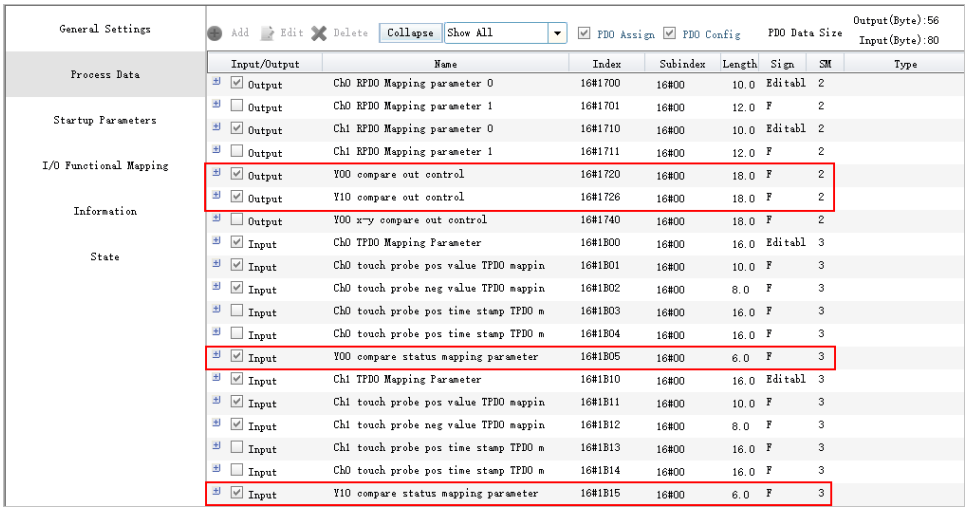


19.4.5 One-Dimensional Comparison Output

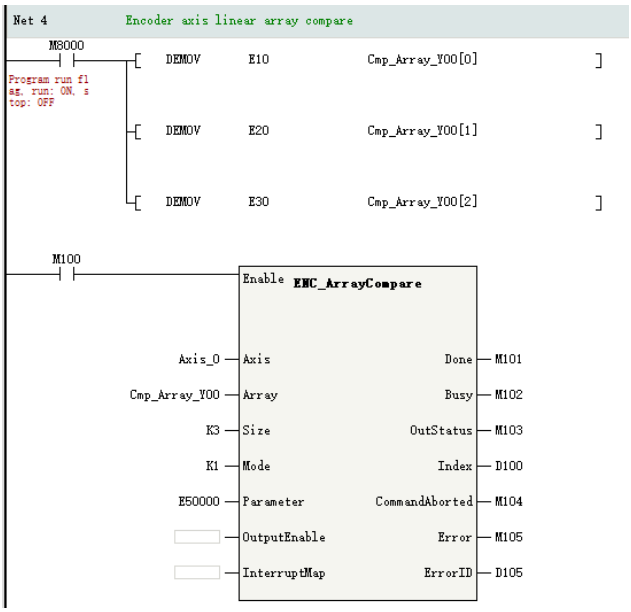
Allocate the one-dimensional comparison output function for Y00 in the encoder axis Axis_0 and Y10 in the encoder axis Axis_1.



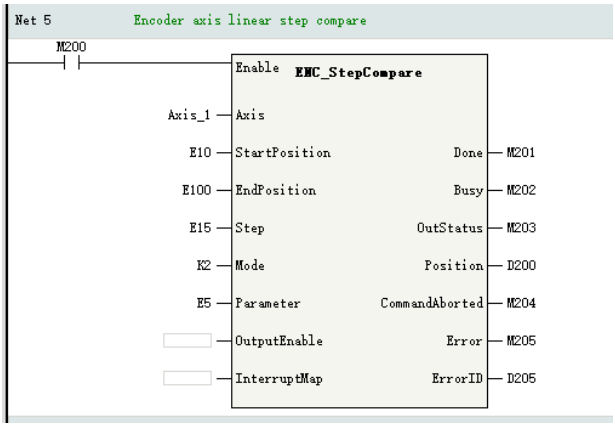
Tick the following PDO data on the "Process Data" tab page of GR10-2HCE-ZG.



Call the instruction ENC_ArrayCompare to implement the array comparison of terminal Y00, and output 50 ms at 10, 20, and 30 points, respectively.

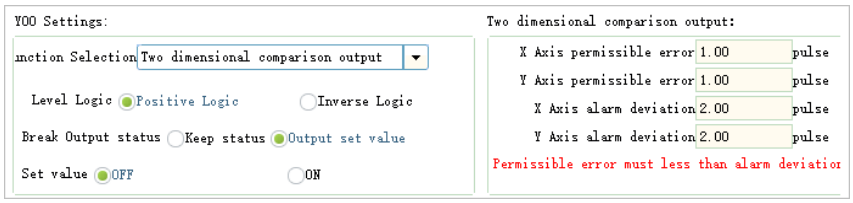


Call the instruction ENC_StepCompare to implement the step comparison of terminal Y10, with the start point at 10, end point at 100, and step size as 15. The comparison output is performed in level control mode with a level length of 5 Unit.

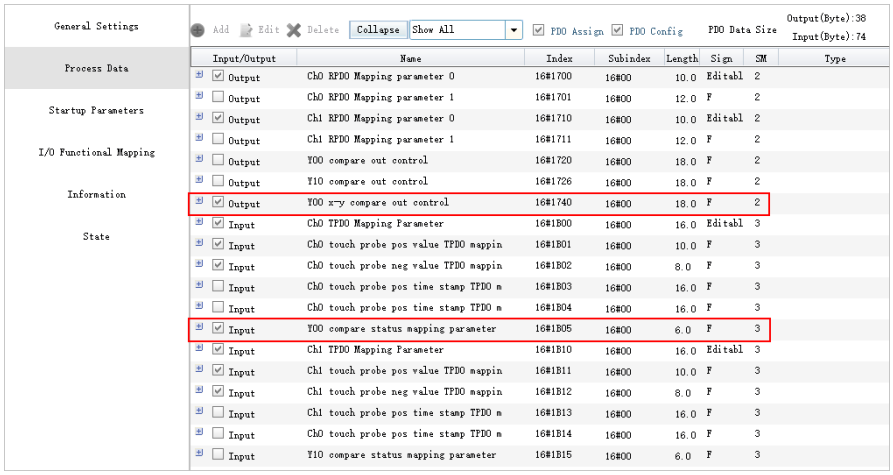


19.4.6 Two-Dimensional Comparison Output

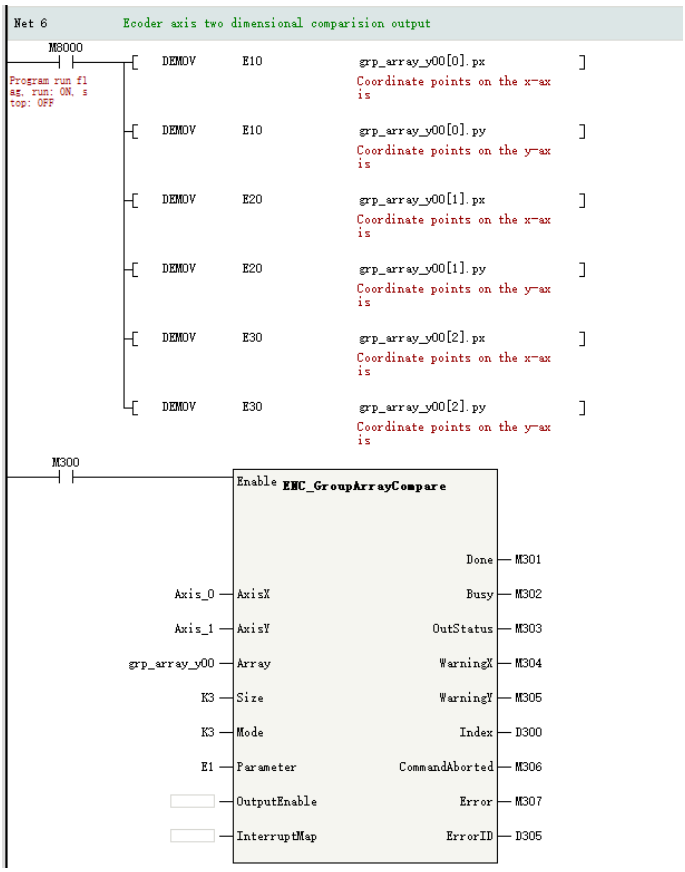
The GR10-2HCE module supports the two-dimensional comparison output function only at terminal Y00. Therefore, only the terminal Y00 of the encoder axis Axis_0 bound to channel CH0 can be allocated with the two-dimensional comparison output function.



Tick the following PDO data on the "Process Data" tab page of GR10-2HCE.



Call the instruction ENC_GroupArrayCompare to realize the comparison output function, and set three comparison points (10,10), (20,20), and (30,30). The comparison output is performed in level control mode with a high initial level.



19.4.7 Digital Output (DO) Terminal Control

If the DO terminal of the GR10-2HCE module is used as a common DO terminal, call the instruction ENC_DigitalOutput in the program to control the terminal. The address mapping table of the DO terminals is as follows.

Address	Controlled Terminal
D400.0	Y00
D400.1	Y01
D400.2	Y02
D410.0	Y10
D410.1	Y11
D410.2	Y12

19.4.8 Reading Axis Status

Call the instruction ENC_ReadStatus to obtain the fault code of an axis and the status of the DI terminal. The address mapping table of the DI terminals is as follows.

Address	Mapped Terminal
D502.0	X00
D502.1	X01
D502.2	X02
D502.3	X03
D512.0	X10
D512.1	X11
D512.2	X12
D512.3	X13

Through the system variables, write the feedback position of the pulse unit of Axis_0 to D508 and the feedback frequency of the pulse unit of Axis_1 to D518.

20 Electronic Cam

20.1 Introduction

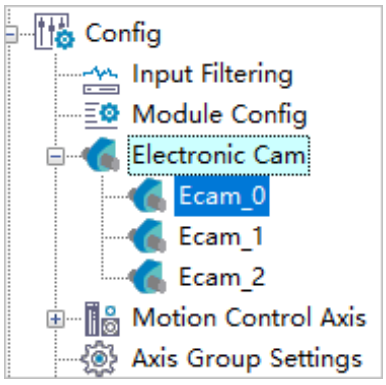
Electronic cam essentially involves the motion of the slave axis following the master axis. The motion relationship between the master axis and the slave axis can be expressed in a cam table data or electronic gear ratio approach .

- In the cam table data approach, up to 361 key points can be created. The electronic gear ratio of master to slave axis is fixed.
- If electronic gears are used, you only need to set the numerator and denominator without setting the cam table. If electronic cams are used, you need to set the electronic cam table first.
- The programming software can be configured with 16 cam tables, each with up to 361 key points. Up to 8 electronic cams can be used simultaneously in the program.
- During cam execution, it is allowed to add, delete, and modify key points of a cam table, and the modified cam table takes effect in the next cam cycle.

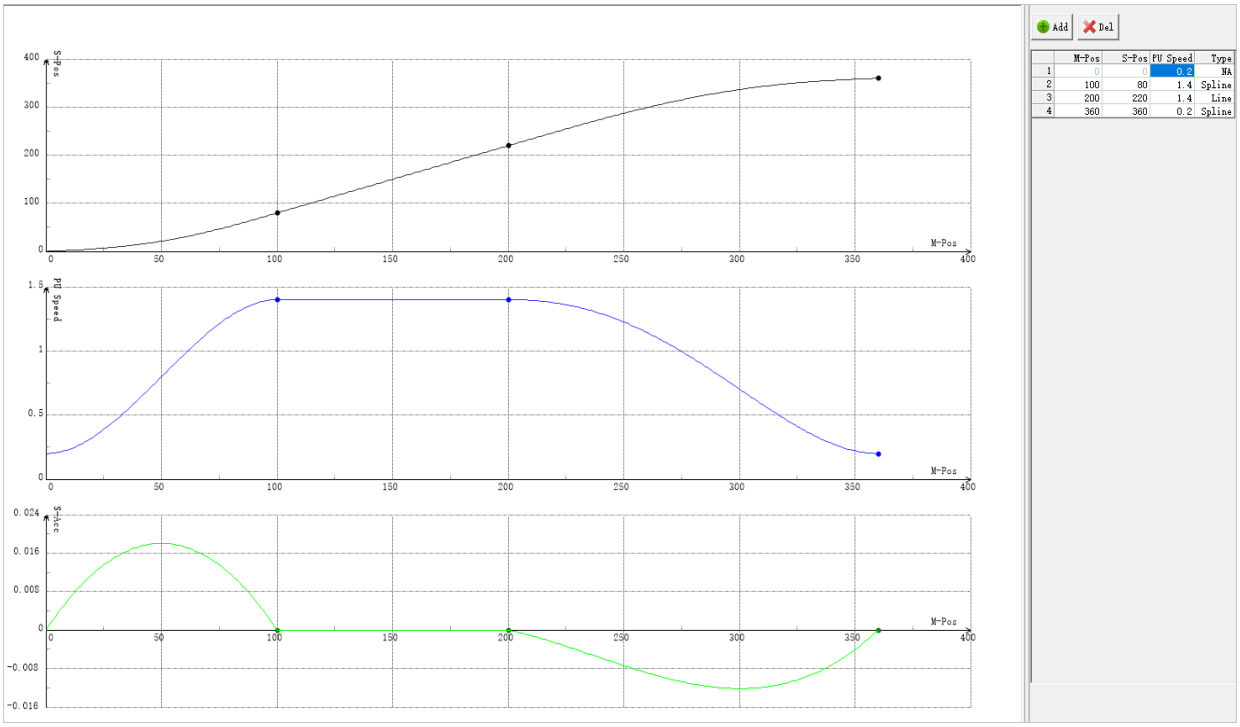
20.2 Software Configuration

20.2.1 Overview

In the "Project Manager" tree, expand "Config", and double-click "Electronic Cam" to open the relevant configuration page.



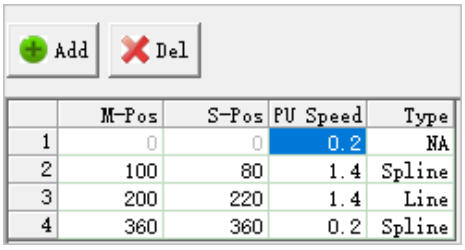
The cam table page contains a graphic editing area on the left and a parameter point editing area on the right.



20.2.2 Cam Node Setting

You can set cam nodes in the parameter point editing area.

Click "Add" to add a cam node data row and edit the relevant data. To delete a node, select the corresponding node data row and click "Del".



	M-Pos	S-Pos	PU Speed	Type
1	0	0	0.2	NA
2	100	80	1.4	Spline
3	200	220	1.4	Line
4	360	360	0.2	Spline

The following table lists definitions of cam node parameters.

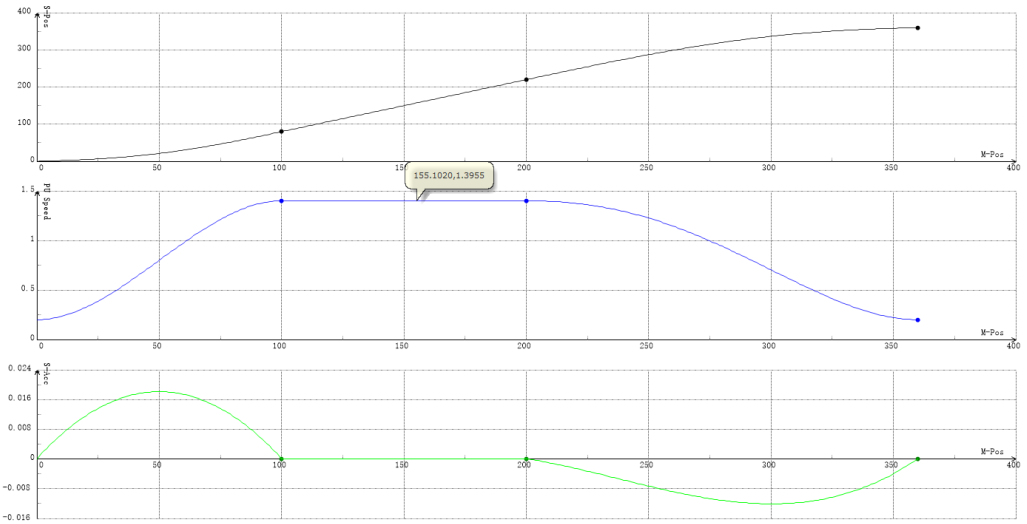
Parameter	Function
M-Pos	Master axis phase Sets the phase of the master axis (relative mode)
S-Pos	Slave axis displacement Sets the offset of the slave axis (relative mode)
PU-Speed	Connection speed Automatically generated when the curve type is set to straight line, or manually set when the curve type is set to quintic curve
Type	Sets the curve type Line: Straight line Spline: Quintic curve

NOTE

- The master axis phase and slave axis offset at the first point are defaulted to 0, which cannot be modified.
- The master axis phases are arranged in ascending order.
- The last point of the master axis determines the cycle size of the master axis, and the cycle does not be defined separately.

20.2.3 Cam Curve Setting

In the graphic editing area, you can set cam curves, including position, speed ratio, and acceleration ratio curves.



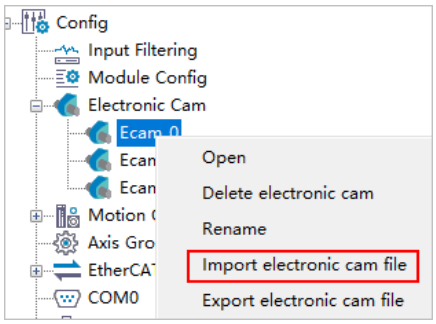
NOTE

- Cam key points on the position curve can be moved up or down and left or right. The speed ratio curve can only be moved up or down. The acceleration ratio curve does not allow modification.
- The last point can only be dragged up and down. To change the value of the last point leftwards or rightwards, manually modify the data in the toolbar on the right.
- Hover the mouse cursor over a point in the coordinate system, and the specific coordinate information will be displayed.
- Right-click to insert or delete a key point.
- Click a line segment between two key points in any coordinate system, and the line segments between the two key points in all the three coordinate systems will become bold.

20.2.4 Importing/Exporting an Electronic Cam

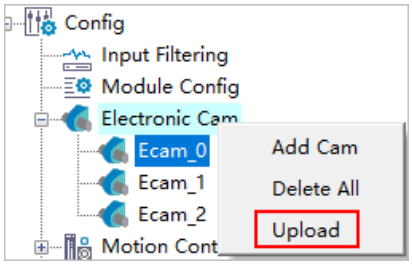
You can export or import each individual cam table.

Select the electronic cam you want to export or import and right-click it to export/import the electronic cam to/from a CSV file.



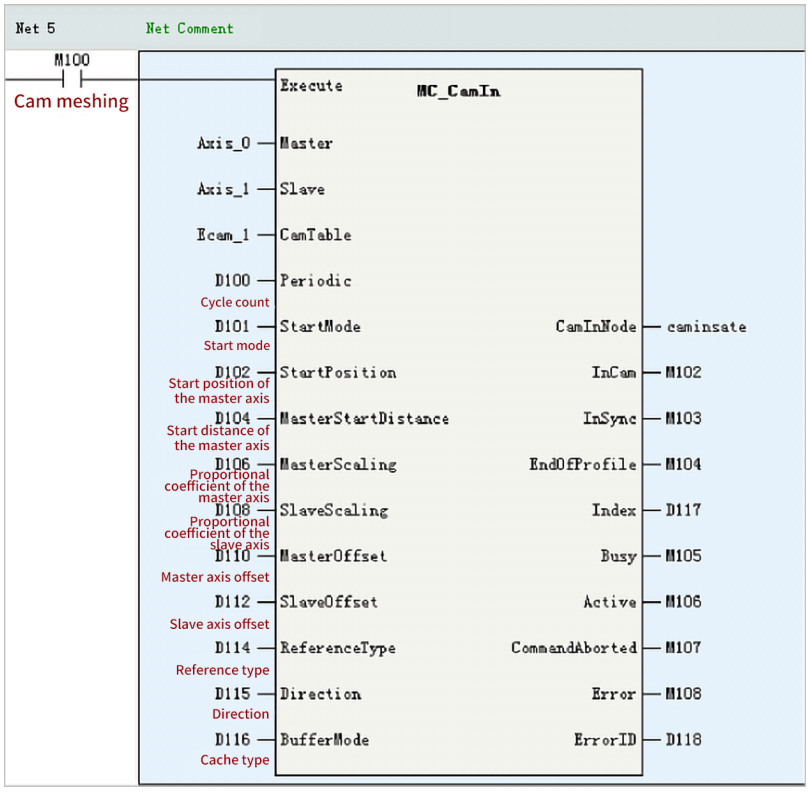
20.2.5 Uploading a Cam Table

All the cam tables saved in a board can be uploaded by using the upload feature.



20.2.6 Calling System Variables and Instructions

When a cam table is created, the software backend assigns a system variable to represent the cam table. The status of the cam table can be monitored in the PLC program and can be used as a parameter for instructions such as MC_CamIn. In addition, values of key points in the cam table can be modified and updated using the MC_GenerateCamTable instruction.



20.3 System Variables

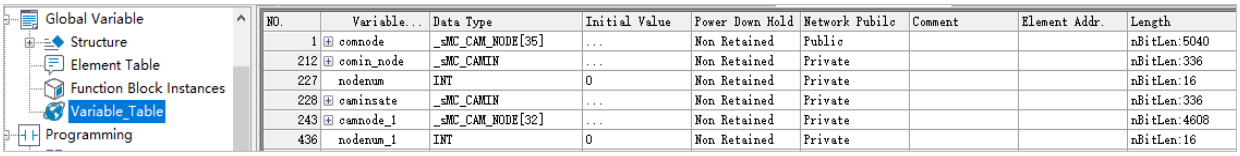
20.3.1 Cam Node

Each key point can be represented by a cam node variable, with the data type being _sMC_CAM_NODE. Member variables of this structure are shown in the following table.

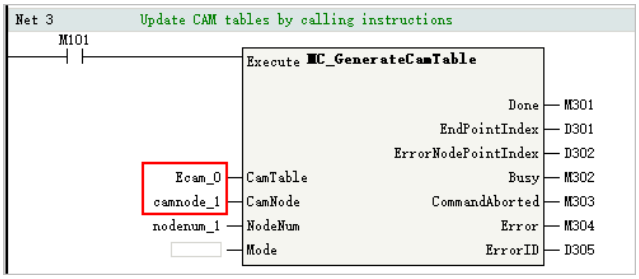
Variable	Data type	Function description
fPhase	REAL	Master axis phase
fDistance	REAL	Displacement of the slave axis
fVel	REAL	Connection speed
fAcc	REAL	Connection acceleration rate (reserved)
iCuve	INT	Curve type 0: Reserved 1: Linear 2: Quintic curve

Each cam node structure is used as a member variable of a cam table structure to store key point data of the cam table. See Cam Table Cam Tablefor details.

In the program, you can also customize cam node arrays for updating cam tables.



The MC_GenerateCamTable instruction can be used to overwrite the existing cam node array in Ecam_0 with a newly-defined camnode_1 cam node array.



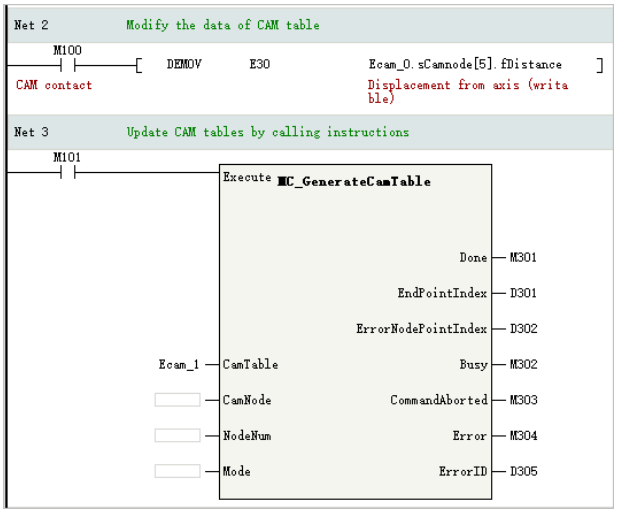
20.3.2 Cam Table

Cam tables can only be created and configured in the background and cannot be created in the program. However, coordinates of cam key points can be modified in the program to facilitate modification of process parameters.

Variable	Data type	R/W property	Description
iCamID	INT	RO	ID
bSaveState	BOOL	RO	Saving a cam table ON execution in progress

Variable	Data type	R/W property	Description
			1083=CYCLE OFF
bCheckState	BOOL	RO	Checking a cam table Set to ON when a check is in progress 1083=CYCLE OFF
bNew	BOOL	RO	Updating a cam table ON: Updating 1083=CYCLE OFF
iErrorCode	INT	RO	Error code corresponding to the cam check/save failure
iSetNodeNum	INT	RO	Total number of set key points
iActNodeNum	INT	RO	Total number of actual key points Updated after the first run and after each execution of the MC_GenerateCamTable instruction
fMaxPhase	REAL	RO	Cam cycle
sCamnode	_sMC_CAM_NODE [361]	RW	Cam key point array Needs the MC_GenerateCamTable instruction for updating after modification in the program The master axis phases must be arranged in ascending order, otherwise an error will occur.

Cam table variables are automatically generated for cam tables created in the background.

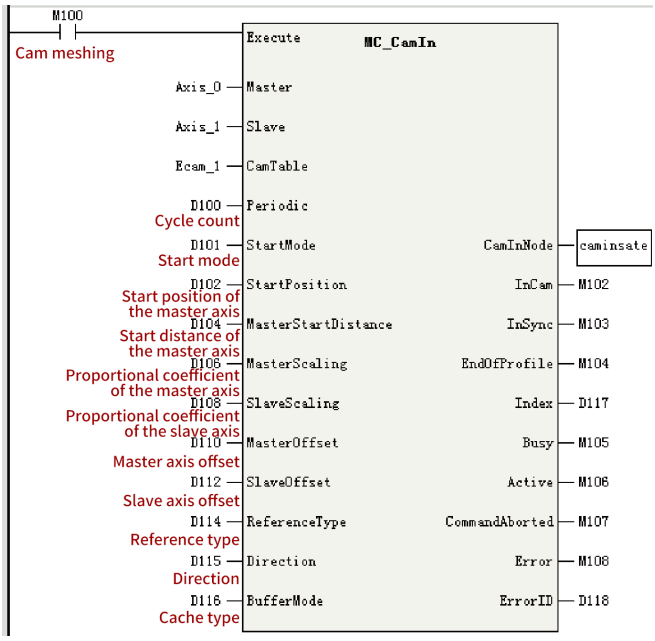


20.3.3 Cam Contact Node

The system variable for the cam contact node is _sMC_CAMIN. For details on data structure, see the following table .

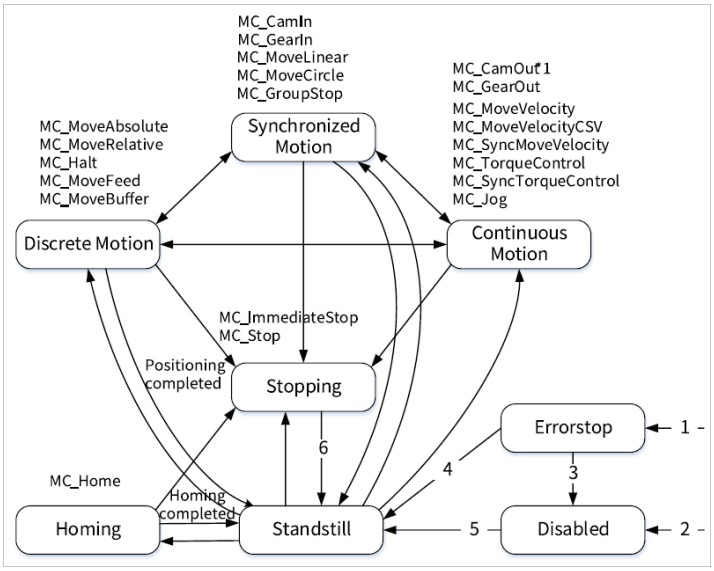
Variable	Data type	R/W property	Function description
iCamInID	INT	RO	Cam combination ID
iCAMTableID	INT	RO	The cam table being executed
iMasterID	INT	RO	Master axis ID
iSlaveID	INT	RO	Slave axis ID
iState	INT	RO	Reserved and not defined
iCamCnt	INT	RO	Number of cam cycles already executed
iNodeCnt	INT	RO	Key points waiting for execution
fMasterStartpos	REAL	RO	Start position of the master axis
fSlaveStartPos	REAL	RO	Start position of the slave axis
fphase	REAL	RO	Current phase of the master axis
fDistance	REAL	RO	Current displacement of the slave axis
fPhaseShift	REAL	RO	Super-imposed amount of phase shift
fPhaseVelocity	REAL	RO	Super-imposed velocity of phase shift
fPhaseAcc	REAL	RO	Super-imposed acceleration of phase shift

This variable can only be defined as an output variable of the MC_CamIn instruction in the program.



20.4 State Machine

The following figure shows the state machines of the electronic cam.



NOTE

Calling the MC_CamOut instruction can start the Continuous Motion state only when OutMode of the instruction is 0.

The status is defined as follows.

Status	Function description
Disabled	Disabled
ErrorStop	Fault reaction
StandStill	Enabled
Homing	Homing status
Stopping	Stopping
Discrete motion	Discrete motion
Continuous motion	Continuous motion
Synchronized motion	Synchronized motion

The following table summarizes the status transition conditions.

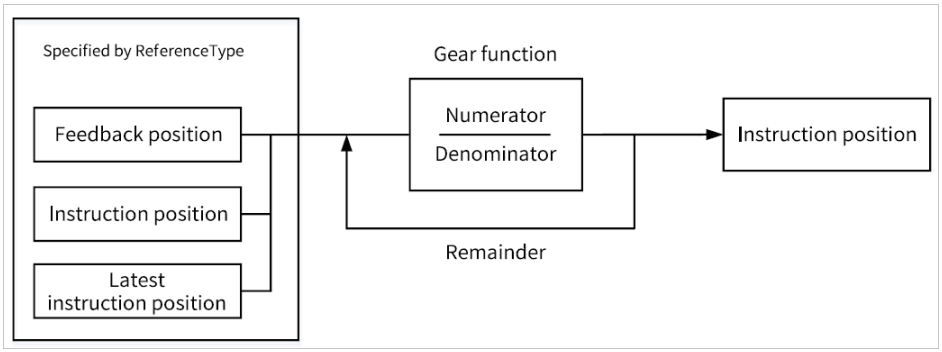
Transition	Transition condition
1	The fault detection logic of the axis has detected a fault.
2	The axis has no fault and MC_Power.Enable is OFF.
3	MC_Reset is called to reset the axis fault and MC_Power.Status=FASLE.
4	MC_Reset is called to reset the axis fault and MC_Power.Status=ON.
5	Both MC_Power.Enable and MC_Power.Status are ON.

Transition	Transition condition
6	MC_Stop(MC_ImmediateStop).Done is ON and MC_Stop(MC_ImmediateStop).Execute is OFF.

20.5 Electronic Cam Operation

20.5.1 Gear Operation

Basic block diagram



Function description

Gear operation is applicable to the following types of master and slave axes.

- Master axis: bus servo axis, local pulse axis, local encoder axis, and bus encoder axis
- Slave axis: bus servo axis and local pulse axis

Use the MC_GearIn (starting gear operation) instruction to start gear operation. Use the MC_GearOut (ending gear operation) instruction or the MC_Stop (forced stop) instruction to end synchronized gear operation.

After coming into action, the slave axis uses the velocity obtained by multiplying the master axis velocity by the gear ratio as the target velocity to perform the acceleration and deceleration actions.

Time before the slave axis reaches the target speed is called the Catching phase. Time after the slave axis reaches the target speed is called the InGear phase.

Gear operation is executed by setting the gear ratio between the master and slave axes.

When the gear ratio is positive, the slave axis moves in the same direction as the master axis. When the gear ratio is negative, the slave axis moves in the opposite direction to the master axis.

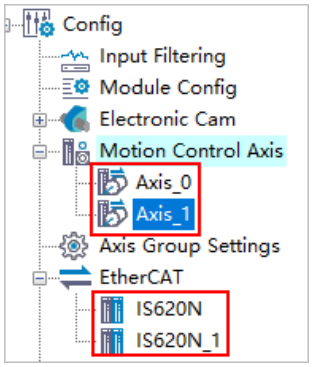
For detailed functions, see the MC_GearIn instruction in the “Electronic Cam Instructions” section of the instructions guide.

Example

Job: Create two bus servo axes and make the second axis follow the first axis at a gear ratio of 1:1 for gear operation.

Procedure:

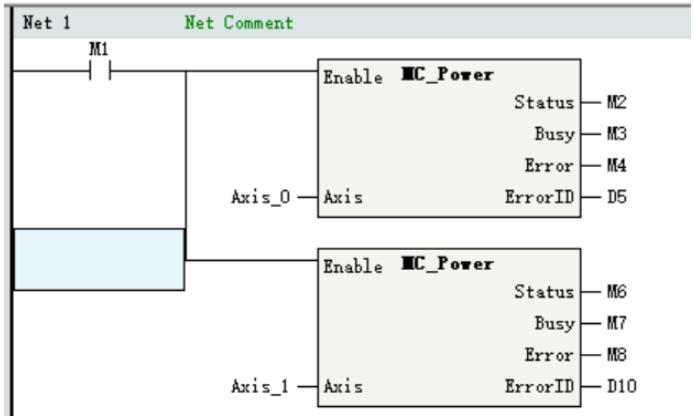
1. Create a project. Create two bus servo axes, one as a master axis and the other as a slave axis.



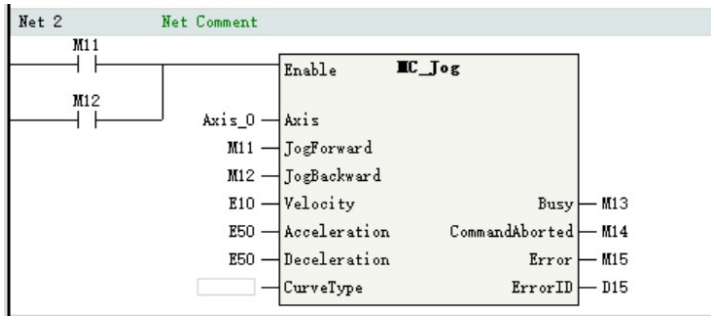
Set Axis_0 of the two bus servo axes as the master axis and Axis_1 as the slave axis.

Bind IS620N of the two servo drives to Axis_0 and IS620_1 to Axis_1.

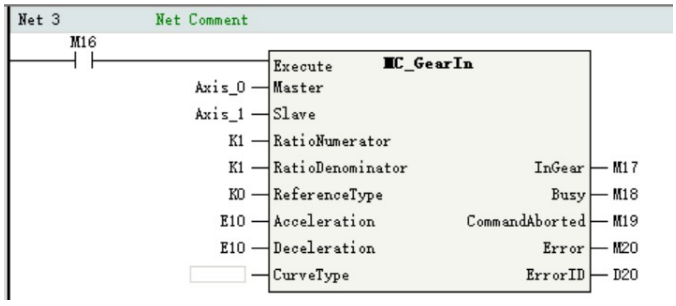
2. Call the MC_Power instruction to control the enabling of the master and slave axes.



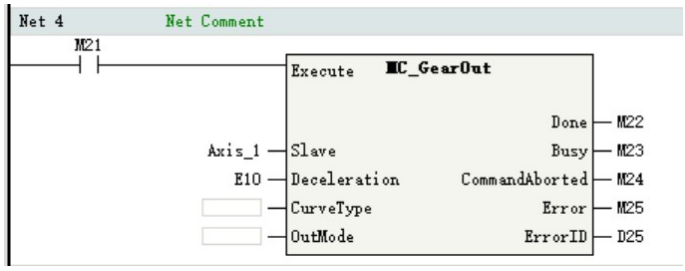
3. Call the MC_Jog instruction to control the forward and reverse motion of the master axis.



4. Call the MC_GearIn instruction to execute gear operation, with the gear ratio set to 1:1.



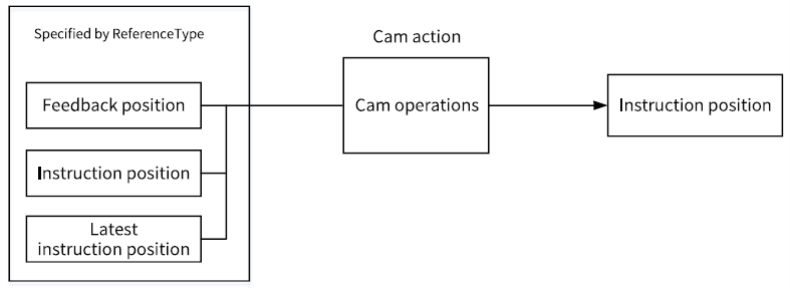
5. Call the MC_GearOut instruction to end gear operation.



20.5.2 Cam Operation

Cam operation refers to the motion of the slave axis in sync with the position of the master axis according to a cam table.

Basic block diagram



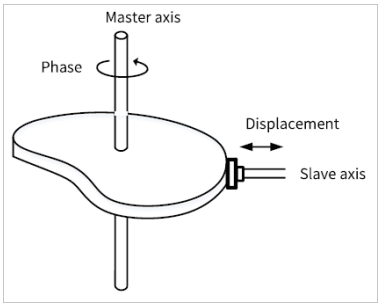
Function description

Cam operation is applicable to the following types of master and slave axes.

- Master axis: bus servo axis, local pulse axis, local encoder axis, and remote encoder axis
- Slave axis: bus servo axis or local pulse axis

Use the MC_CamIn (starting cam operation) instruction to start cam operation or change cam tables. Use the MC_CamOut (ending cam operation) instruction or the MC_Stop (forced stop) instruction to end cam operation.

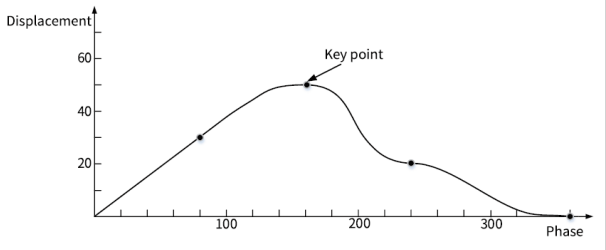
In a typical cam structure shown in the following figure, the master axis rotates periodically and the slave axis moves back and forth along a direction under the control of the master axis.



The electronic cam simulates such a structure, selecting one axis (bus servo axis, local pulse axis, local encoder axis, or remote encoder axis) as the master axis and another axis (bus servo axis or local pulse axis) as the slave axis. The master axis and the slave axis move in a synchronized way according to a set cam curve.

Cam curves

A cam curve is a 2D coordinate system, where the horizontal axis represents the phase of the master axis and the vertical axis represents the displacement of the slave axis. Set some key points in the coordinate system, and connect every two key points with a set curve (such as a straight line or a quintic curve) to form a cam curve.



Cam table

	Phase	Displacement
Start point	0	0
	80	30
	160	50
	240	20
End point	360	0

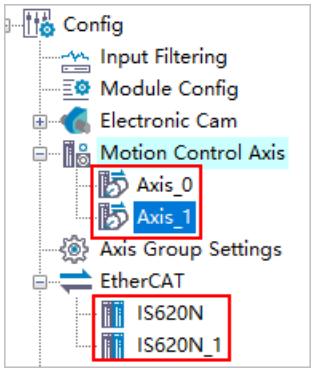
For detailed functions, see the MC_CamIn and MC_CamOut instructions in the "Electronic Cam Instructions" section of the instructions guide.

Example

> **NOTE**
>
> Create two servo axes. Set Axis_0 as the cam master axis, and Axis_1 as the cam slave axis that follows Axis_0 to execute cam operation.

Procedure

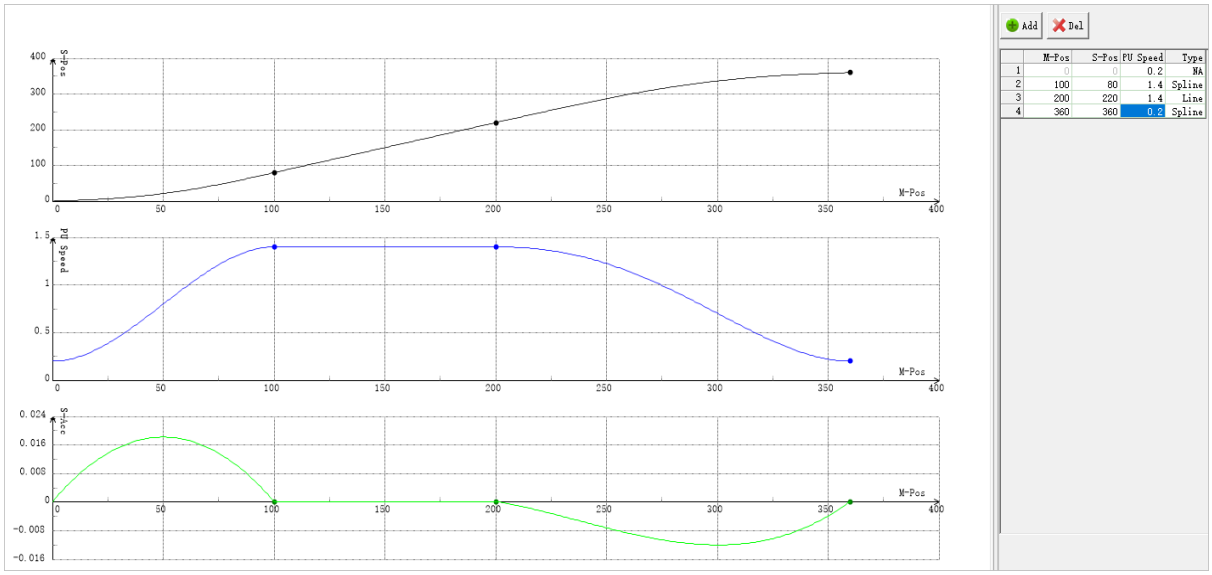
1. Create a project. Create two bus servo axes, one as a master axis and the other as a slave axis.



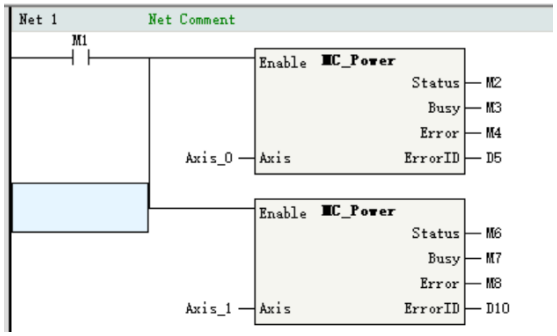
   Set Axis_0 of the two bus servo axes as the master axis and Axis_1 as the slave axis.

   Bind IS620N of the two servo drives to Axis_0 and IS620_1 to Axis_1.

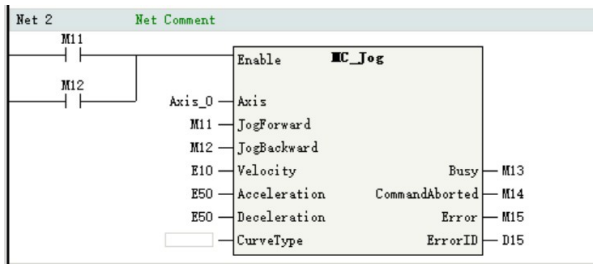
2. Create a cam table.



3. Call the MC_Power instruction to control the enabling of the master and slave axes.

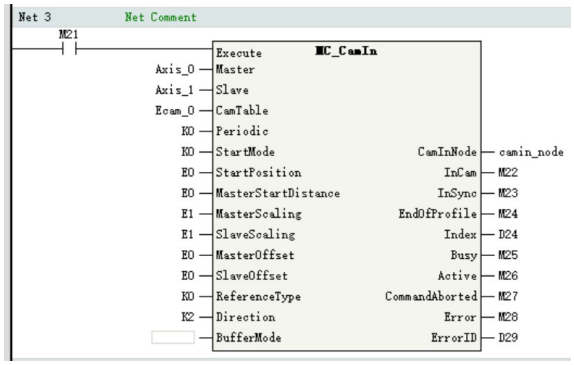


4. Call the MC_Jog instruction to control the forward and reverse motion of the master axis.

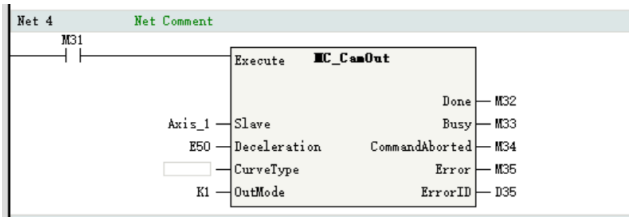


5. Call the MC_CamIn instruction to execute cam operation.

NO.	Variable...	Data Type	Initial Value	Power Down Hold	Network Pubilc	Comment	Element Addr.	Length
1	camin_node	_sMC_CAMIN	...	Non Retained	Private			nBitLen:336



6. Call the MC_CamOut instruction to end cam operation.



20.5.3 Cam Table

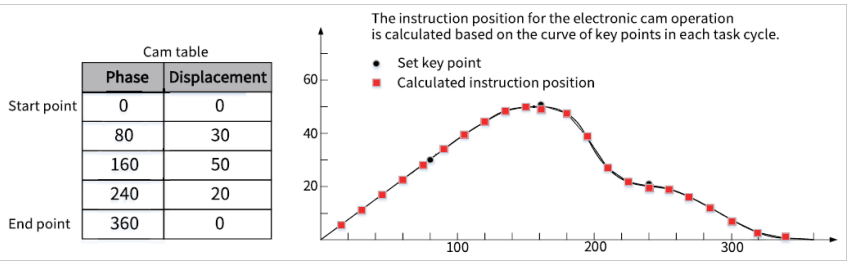
20.5.3.1 Introduction

In the cam function module, a pair of data comprised of the master axis phase and the slave axis displacement is defined as cam data, and combination of cam data is defined as a cam table.

Phase and displacement values of cam data in a cam table are relative quantities expressed in relation to the start point "0.0".

In cam operation, the slave axis displacement is calculated based on the master axis phase and the set curve type, to control the operation of the slave axis.

After a cam table is created with the cam editor in AutoShop, cam data in the cam table can be modified in the user program.



20.5.3.2 Cam Table Specifications

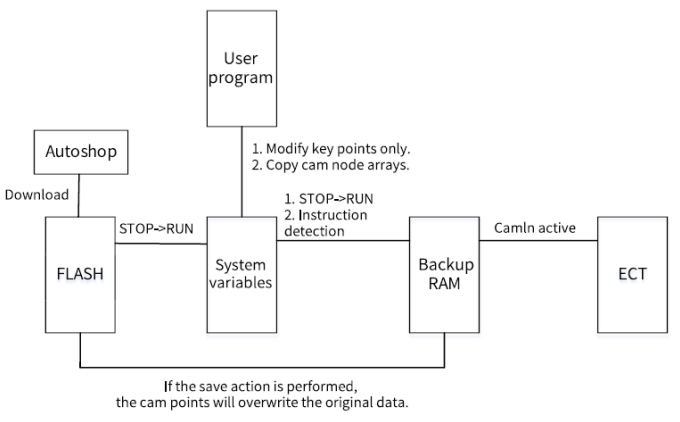
Observe the following specifications when creating cam tables.

Item	Description
Total number of cam key points supported by each cam table	361
Total number of cam tables supported	16
Number of cam tables that can be executed simultaneously in the PLC	8
Rules for switching cam tables during cam operation	Switch cam tables using the MC_CamIn instruction, and the newly selected cam table takes effect in the next cam cycle.
Reading and writing cam data	View the status and key point data of each cam table by using the global variable named after the cam table. You can modify key point data in a cam table directly and make the modification take effect by using the MC_GenerateCamTable instruction. The cam will act according to the modified cam table in the next cam cycle.

Item	Description
Saving cam tables	Modified cam tables can be saved to the non-volatile memory of the PLC using the MC_SaveCamTable instruction.

20.5.3.3 Data Flow of Cam Table

The following figure shows the data execution flow of cam tables.

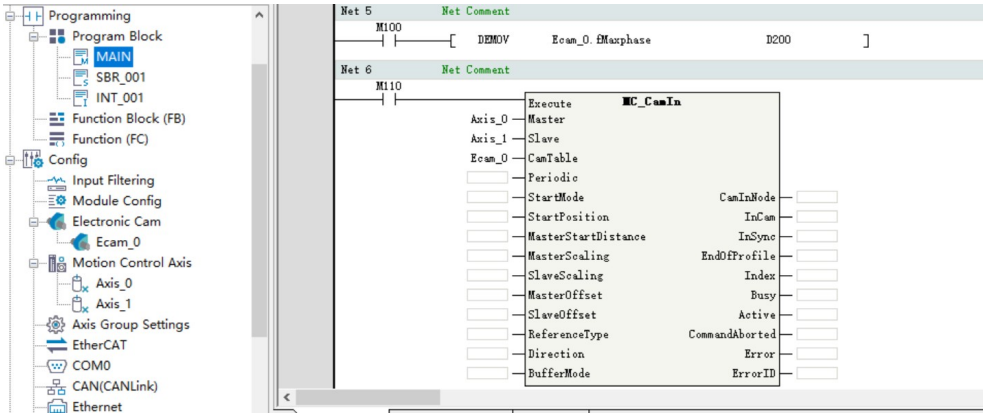


Data flow description

1. In the background, download a cam curve to the non-volatile memory.
2. In the background, upload a cam table file from the non-volatile memory.
3. When the cam table in the non-volatile memory switches from STOP to RUN after download, the cam table is loaded to the cam table system variable and initialized to the backup area.
4. The cam table in the user area is updated to the EtherCAT memory when the execution of MC_CamIn Hour after one cam motion cycle is completed. Then, EtherCAT works according to the updated cam nodes.
5. With the user program, you can modify a cam key point in the system variable or copy a new cam node array to an existing cam table, and then copy the modified cam table to the backup RAM by using the MC_GenerateCamTable instruction.
6. After modifying or creating cam key points in the user program, call the MC_GenerateCamTable instruction to check the rationality of the cam table.
7. Call the MC_SaveCamTable instruction to write the cam table in the backup area into the non-volatile memory.

20.5.3.4 Creating a Cam Table

Cam table variables can only be created in the background. Every time a cam table is added, a cam table variable is created by default. The name of the cam table variable is the name of the cam table in the configuration. You can obtain the status of the cam table by using the variable and use the status as an input parameter for cam instructions.



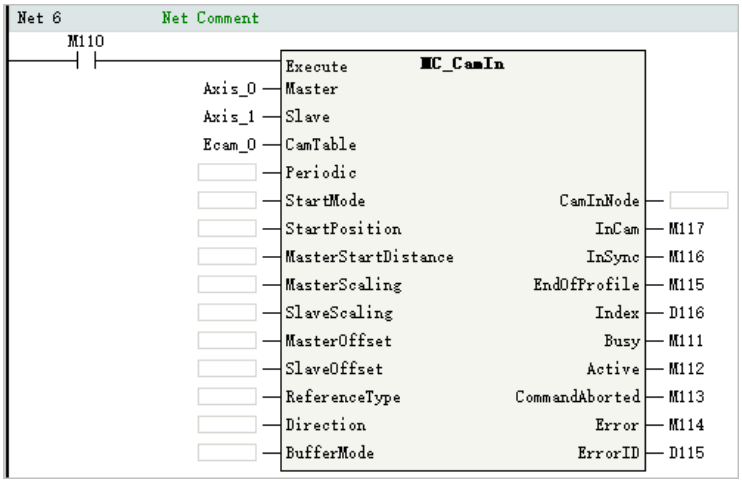
20.5.3.5 Switching the Cam Table

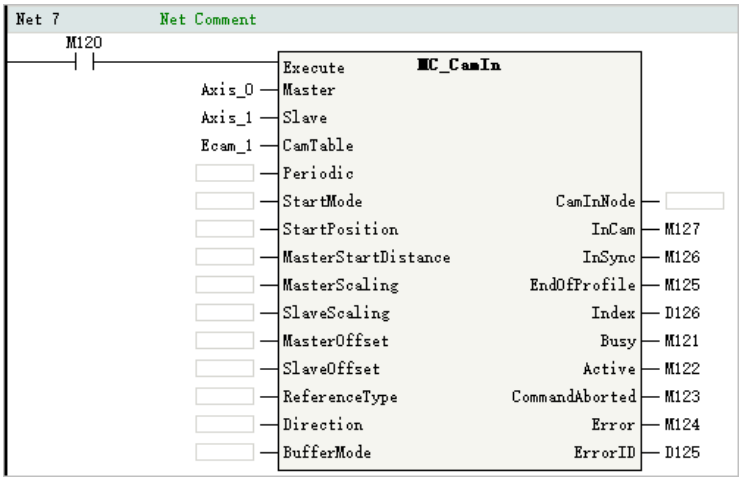
During cam execution, you can switch to a different cam table by triggering the MC_CamIn instruction. After triggering, the cam table becomes buffered, and the buffered cam table takes effect in the next cam cycle.

Only one cam table can be buffered. If multiple MC_CamIn instructions are triggered in succession, the cam table triggered earlier will be overwritten by the cam table triggered later.

Here is an example of triggering two instructions.

- Trigger M110 first. After the instruction detects that the cam parameter is set correctly, M111 (Busy) output becomes active, Axis_1 starts to move according to the cam curve set by the Ecam_0 cam table, and M112 (Active) output becomes active. If M120 is triggered before one cam cycle is completed, the Ecam_1 cam table becomes buffered and M121 (Busy) output becomes active.
- After Axis_1 completes the first cam cycle, the first cam instruction is aborted and M113 (CommandAborted) output becomes active. In this case, Axis_1 starts to move according to the cam curve set by the Ecam_1 cam table and M122 (Active) output becomes active.

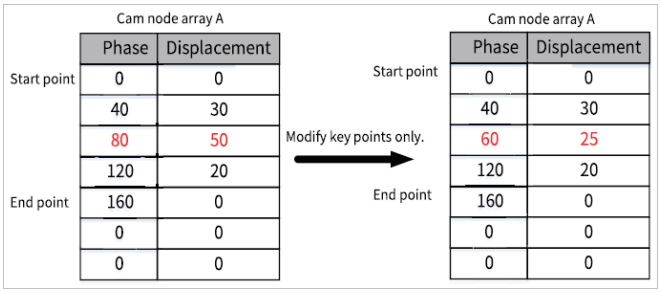




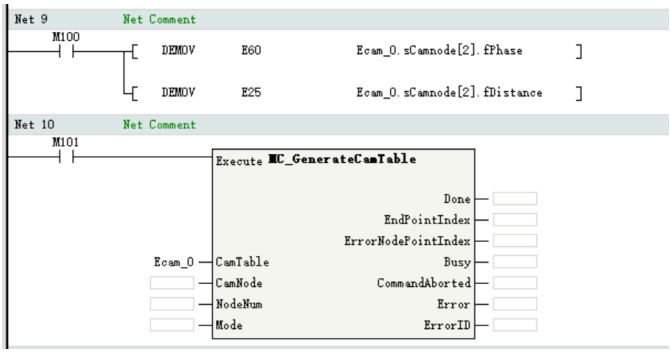
20.5.3.6 Modifying Cam Data

Cam data can be modified through the following three methods.

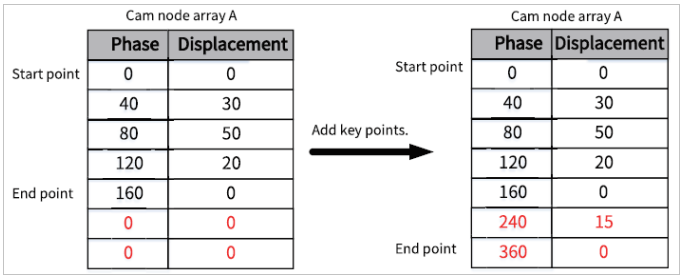
1. Modify values in the cam node array of the cam table through the PLC program, and then execute the MC_GenerateCamTable instruction. The modification takes effect in the next cam cycle.



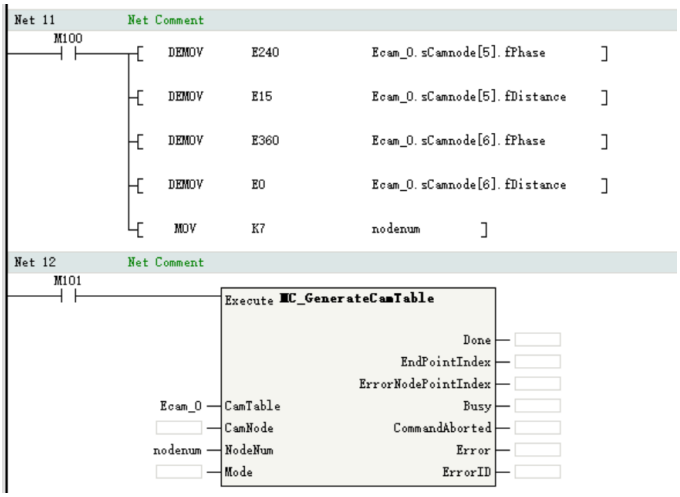
The program example is as follows:



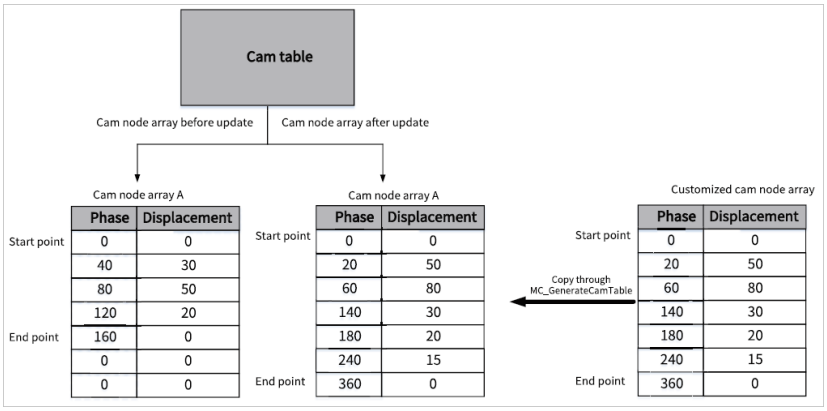
2. You can modify the number of key points in the cam table and make the modification take effect in the next cam cycle by executing the MC_GenerateCamTable instruction.



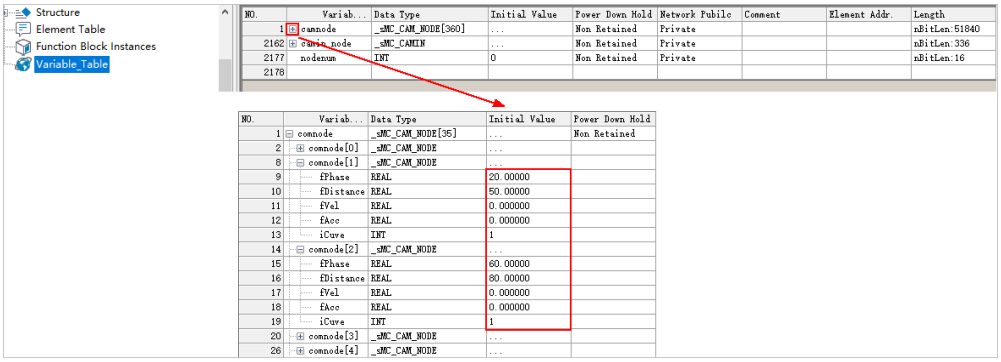
The program example is as follows:

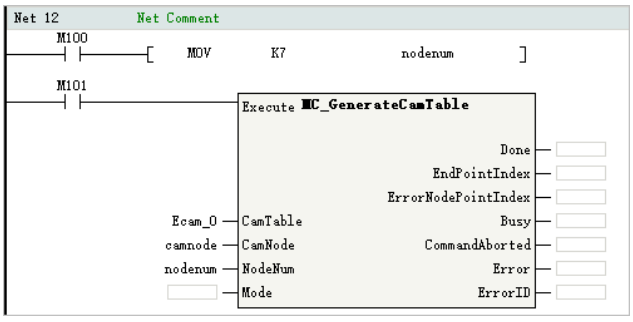


3. You can create a new cam node array by using the PLC program and copy the value in this array to the cam table by executing the MC_GenerateCamTable instruction so that the value is executed upon the next cam cycle.



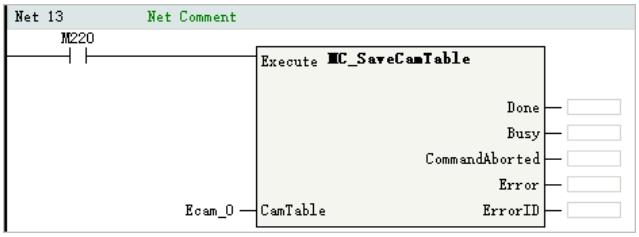
The program example is as follows:





20.5.3.7 Saving the Cam Table

Modified cam tables can be written to the non-volatile storage space by using the MC_SaveCamTable instruction.

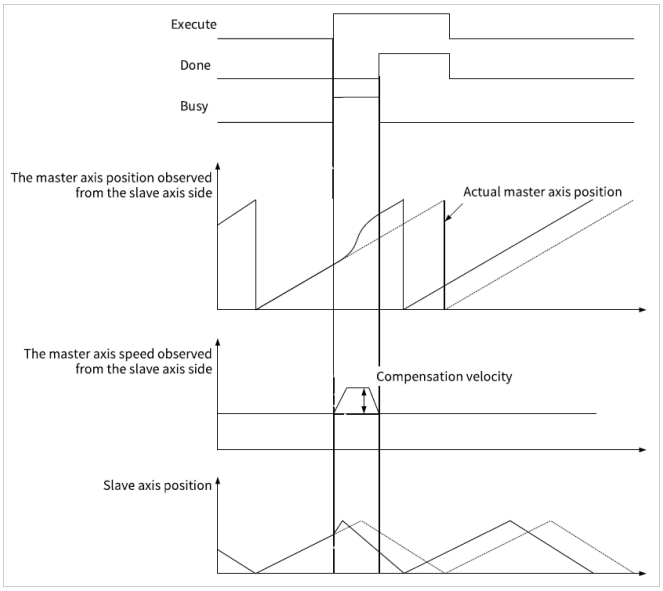


20.5.4 Master Axis Phase Shift

This function allows the master axis phase to be shifted (observed from the slave axis) for instructions in operation.

Starting the MC_Phasing (master axis phase shift) instruction can compensate the phase for the synchronized control instruction.

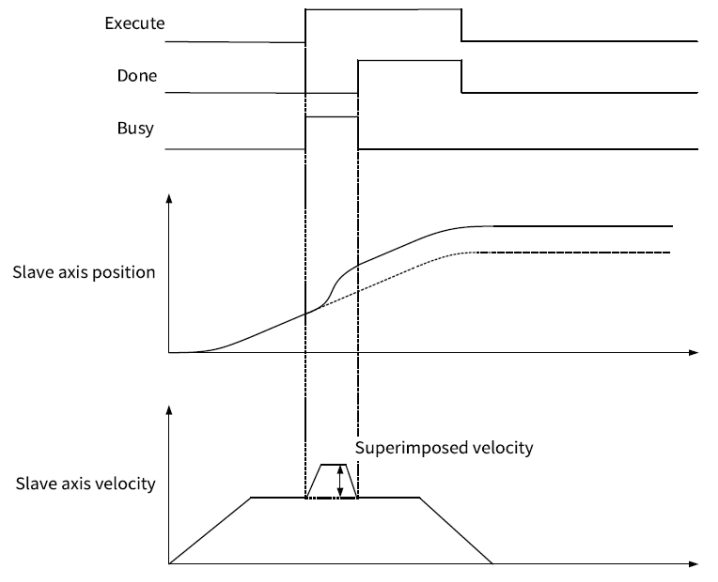
The MC_Phasing (master axis phase shift) instruction can specify parameters such as phase compensation, target speed, acceleration rate, and deceleration rate.



20.5.5 Motion Superimposition

Call the MC_MoveSuperImposed instruction to implement motion superimposition on the motion control axis.

The MC_MoveSuperImposed (motion superimposition) instruction can specify parameters such as position compensation, target speed, acceleration rate, and deceleration rate.



20.5.6 Handling Methods for Single-Axis Configuration Parameters in Cam or Gear

Methods of handling single-axis configuration parameters in cam or gear are shown in the following table.

Parameter	Solution
Gear ratio settings	The master and slave axes of the cam and gear support gear ratio modification. Set gear ratios in the “Unit Conversion Settings” interfaces of the cam and gear, respectively.
Encoder mode selection	The master and slave axes of the cam and gear support drive encoder mode setting. The encoder can be set to incremental or absolute mode. Set this parameter in the “Mode/Parameter Settings” interfaces of the cam and gear, respectively.
Circular/linear mode setting	The master and slave axes of the cam and gear support linear and circular modes. Set this parameter in the “Mode/Parameter Settings” interfaces of the cam and gear, respectively.
Limit handling	In linear mode, the slave axes of the cam and gear support software limit but do not support limit-based deceleration. The slave axes stop immediately after encountering the limit. In both linear and circular modes, the slave axes of the cam and gear support hard limit, and stop immediately after encountering the limit.
Deceleration upon axis fault	When an abnormal instruction parameter causes a fault on a slave axis, the slave axis decelerates according to the deceleration rate specified by the axis fault deceleration parameter and then enters the ErrorStop state.
Tracking error	The slave axes of the cam and gear support following error during operation.
Speed limit	Instructions related to the cam and gear are not restricted by the maximum speed parameter in the configuration.
Acceleration limit	The deceleration rate of the MC_CamOut instruction is restricted by the maximum acceleration parameter in the slave axis configuration. The instruction reports an

Parameter	Solution
	error and the axis enters the ErrorStop state when the deceleration rate exceeds the limit. The acceleration rate and deceleration rate of the MC_GearIn instruction are restricted by the maximum acceleration parameter in the slave axis configuration. The instruction reports an error and the axis enters the ErrorStop state when the acceleration rate or deceleration rate exceeds the limit. The deceleration rate of the MC_GearOut instruction is restricted by the maximum deceleration parameter in the slave axis configuration. The instruction reports an error and the axis enters the ErrorStop state when the deceleration rate exceeds the limit.
Torque limit	The torque limit value is written into the servo drive as a startup parameter and is controlled by the servo drive.

21 Offline Commissioning

21.1 Overview

> **CAUTION**
>
> Offline commissioning is available in AutoShop V4.2.0.0 and later versions.

Offline commissioning allows users to commission the logic, motion control, and communication functions of the program.

Offline commissioning covers the motion control axis, module configuration, communication, and online modification functions. It comes with a status display interface that facilitates real-time monitoring of the status of the PLC and module I/O channels. In addition, offline commissioning can be used with the online simulation function of the IT7000 series HMI to achieve simulation commissioning, making commissioning more convenient.

The following table lists the functions supported or not supported in the offline commissioning mode.

Supported or Not	Function	Description
Supported	Program	Supports main program, subprogram, and FB/FC.
		Supports the "timed interrupt" subprogram.
		Supports online modification.
	Motion control	Supports local pulse axis and EtherCAT bus servo axis.
		Supports local encoder axis; uses simulation of internal clock signal.
	Communication	Supports serial communication configuration and Modbus-RTU/Modbus-ASCII master-slave protocol. Note: • Multi-serial port models support simultaneous simulation of COM0, COM2, and COM3, corresponding to COM9, COM16 and COM17 serial port physical devices in PC device manager respectively. • Only when the PC serial port has COM9, COM16 or COM17 devices, serial communication can take effect during offline commissioning, and the COM icon of the AutoShop project tree can normally display the connection, disconnection, and stop running. • After the USB-to-RS232/485 serial port cable is connected, the COM number can be changed to COM9, COM16 or COM17 in the device manager.
		Supports Ethernet Modbus-TCP protocol, Ethernet TCP/UDP protocol, and EtherNet/IP and OPC UA protocol.

Supported or Not	Function	Description
		Supports online simulation with HMI. Communication is established through specified ports. A communication variable table is generated in the background and imported to the HMI. Monitoring objects include all components and customized variables.
	Instruction	Supports all instructions, except the HOUR, DHOUR/TWR, TRD instructions.
	Extension	Supports extension module I/O, local DIDO, and function virtualization.
	Other functions	Supports RTC clock; uses the clock built in Windows; does not support PLC time setting.
		Keeps consistent with PLC.
Not supported	Basic	Does not support time setting.
		Does not support "Set/Modify Login PLC Password" or "Delete Login PLC Password".
		Cannot enter or exit offline commissioning during compile, download, or firmware upgrade process.
	Program	Does not support hard interruption, edge interruption, or comparison interruption.
	Motion control	Does not support bus encoder axis.
	Communication	Does not support CANlink or CANopen.
		Does not support EtherCAT communication or instructions.

NOTE

- Offline commissioning of the Windows operating systems (these systems are not real-time systems) may result in some difference between the timer and the actual PLC in the offline commissioning state, involving the timed-interrupt subprogram internal clock signal, motion control and scan cycle.
- In offline commissioning, the CPU and memory usage displayed is the CPU and memory usage of the PC.
- Functions that are not supported cannot be used normally, but they do not affect offline commissioning and therefore need no special treatment.

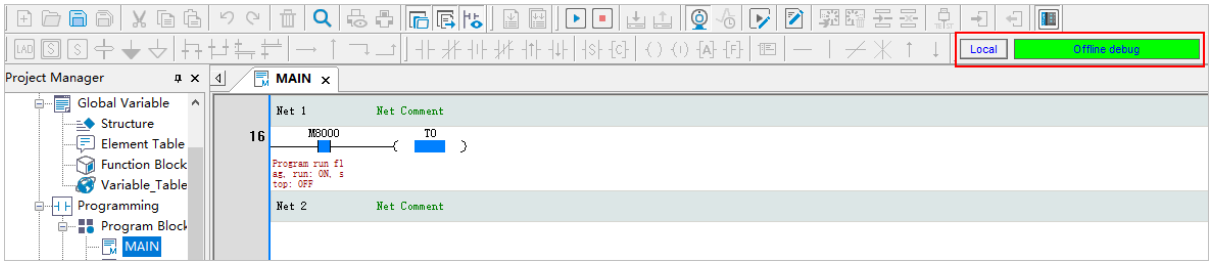
21.2 Starting Offline Commissioning

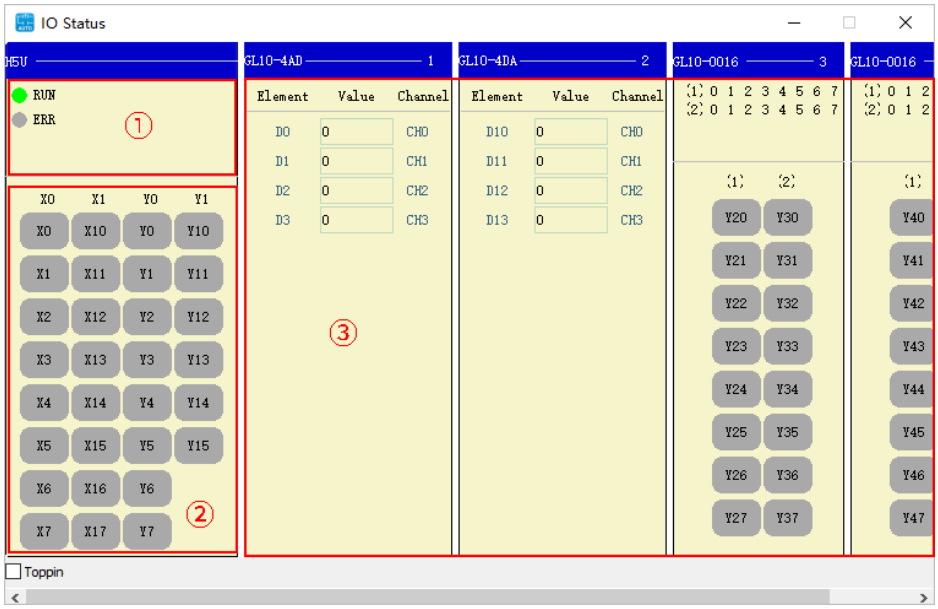
Preconditions

The project to be commissioned offline is created or opened.

Procedure

1. In the toolbar, click to start the offline debugger. Run the PLC program that has been created or opened.





The interface is comprised of the following parts.

No.	Description
①	PLC status
②	Status of I/O channels of the PLC. Green indicates active I/O, whereas gray indicates inactive I/O.
③	Status of extension modules. Status of digital extension modules are consistent with the PLC, whereas the section of analog extension modules shows the values in the current mapped registers.

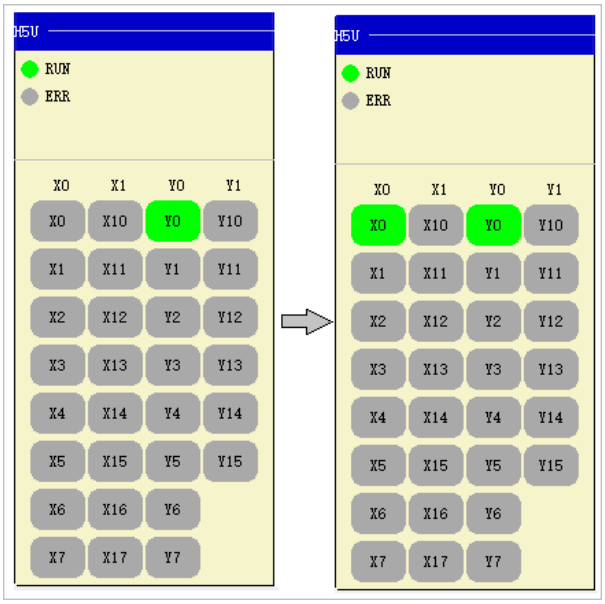
NOTE

After starting offline commissioning, you can perform the following operations:

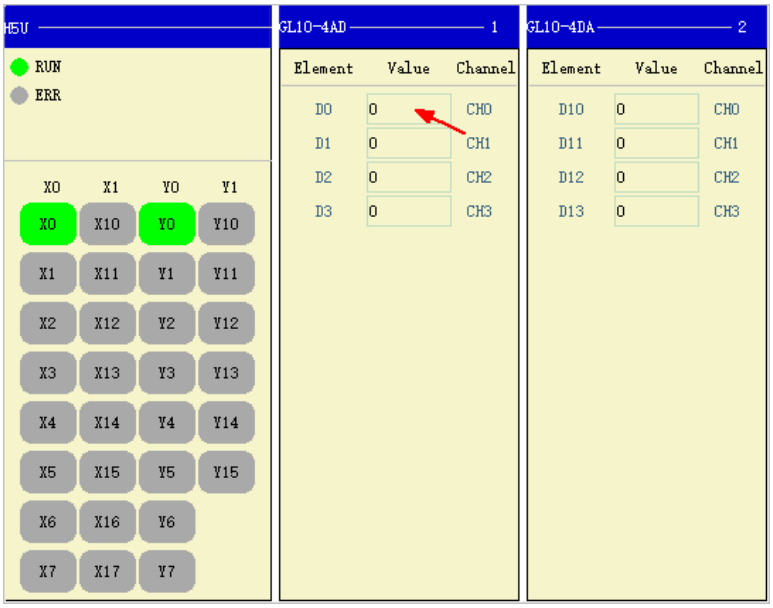
- Monitor the status of the program online, including the inputs and outputs of the PLC.
- Modify or download the PLC program.
- Modify the program online.

2. Commission digital and analog terminals.

- Commission digital terminals: For any I/O point that is not controlled by the program, directly click the I/O point to switch its ON/OFF state.
  I/O points controlled by the program run according to the logic of the program.



- Commission analog terminals: Click a value input field to enter an analog value.
  After the value is input, the parameter is written to the corresponding mapping element. Similar to digital terminals, mapping elements controlled by the program run according to the logic of the program.



NOTE

Actual sensors cannot be connected during offline commissioning. Therefore, analog input values must be typed manually for offline commissioning of analog terminals.

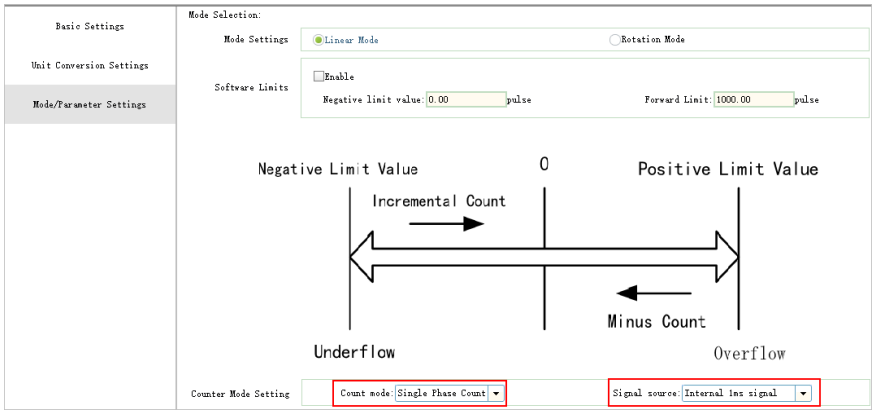
21.3 Description of Offline Commissioning for Motion Control Axis

Local pulse axis and bus servo axis

- In offline commissioning, local pulse axis and bus servo axis can be used without special modifications in the program.
- All functions, including probes and hardware limits, related to external input cannot be used in offline commissioning.
- Using the homing instruction does not need special settings. When the homing instruction is used, homing mode No. 35 is activated (that is, the current position is taken as the home).

Local encoder axis and bus encoder axis

- To use local encoder axis in offline commissioning, select the single-phase counting mode and select internal clock signal (1 ms or 1μs) as the signal source. If other modes are used, the local encoder axis cannot start counting.
- In offline commissioning, only HC_Counter and HC_Preset instructions are supported for local encoder axis.
- Bus encoder axis is not supported in offline commissioning.



21.4 Simulation and Commissioning

21.4.1 Overview

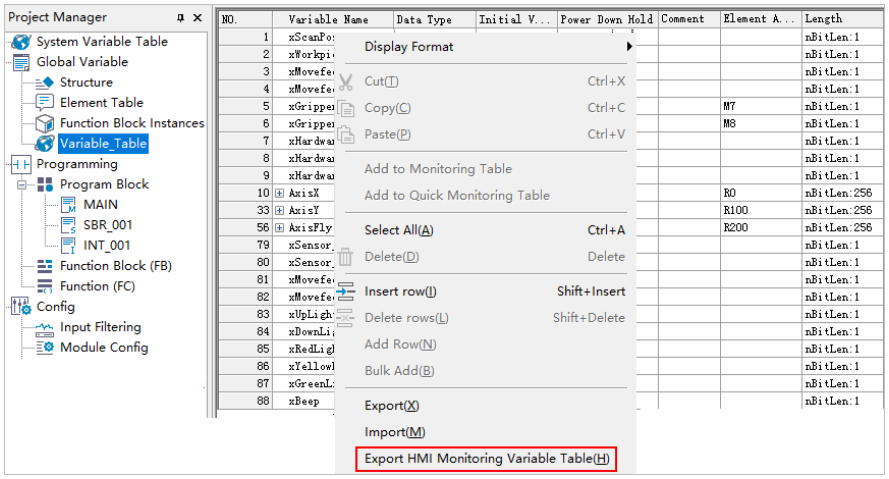
The H5U or Easy series PLC can work with the IT7000 series HMI to achieve simulation commissioning without physical objects through the commissioning feature of AutoShop and InoTouchPad software.

The PLC and HMI communicate with each other through TCP monitoring protocol, supporting customized variables and read and write operations on soft elements.

21.4.2 PLC Configuration

The PLC and HMI employ internal communication for simulation commissioning. Therefore, there is no need to configure the PLC's IP address. You only need to export the PLC's variables to the HMI monitoring variable table to complete the PLC configuration.

1. After compiling the user program, open the variable table, right-click any area of the variable table, and select "Export HMI Monitoring Variable Table" in the shortcut menu.

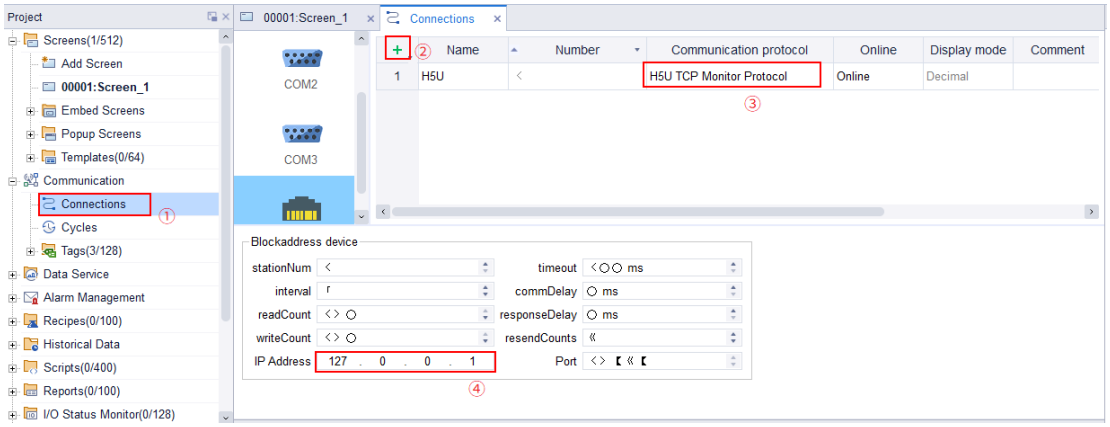


2. Select the path to save the file to be exported, enter the file name, and click "Save" to complete the export.

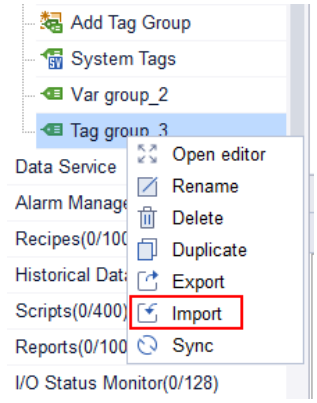
21.4.3 HMI Configuration

To use simulation commissioning, add a PLC connection on the HMI and import the variable table. For H5U, specific operation steps are as follows.

1. Create an HMI connection.
   a. Double-click "Connections" to open the connection tab.
   b. Click + to add a connection.
   c. In the "Protocol" column, select "H5U TCP monitoring protocol".
   d. In the "IP Address" field, enter "127.0.0.1".

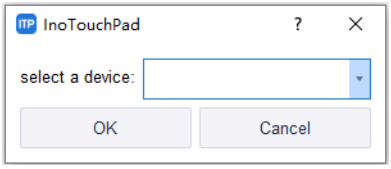


2. Import the HMI monitoring variable table.
   a. Double-click "Add Tag Group" to add a variable group.
   b. Right-click the newly created variable group and select "Import".

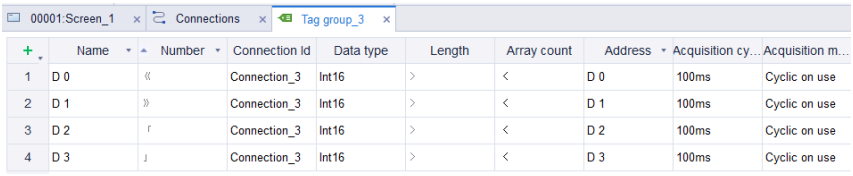


c. In the opened dialog box, select the HMI monitoring variable table exported from the PLC and click "Open".

d. In the "Select Device" field, select the HMI connection created in step 1.



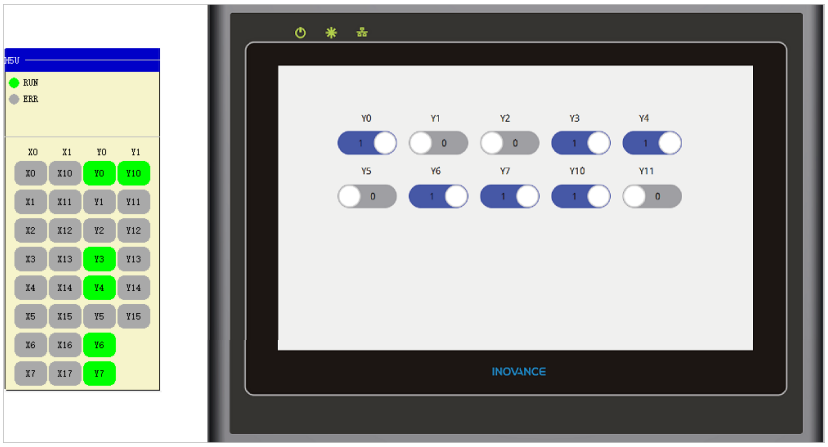
After the import is completed, variables are automatically generated in the variable group.



	Name	Number	Connection Id	Data type	Length	Array count	Address	Acquisition cy...	Acquisition m...
1	D 0		Connection_3	Int16			D 0	100ms	Cyclic on use
2	D 1		Connection_3	Int16			D 1	100ms	Cyclic on use
3	D 2		Connection_3	Int16			D 2	100ms	Cyclic on use
4	D 3		Connection_3	Int16			D 3	100ms	Cyclic on use

21.4.4 Starting Commissioning

After editing the program and importing the variable table for HMI and PLC, you can start the offline commissioning function of AutoShop and the online simulation function of InoTouchPad for commissioning.



21.5 Ethernet/IP Offline Commissioning

21.5.1 Overview

AutoShop supports EtherNet/IP communication in the Windows simulation version for temporary offline commissioning without PLC.

The EtherNet/IP simulation version has the following features same as the PLCs:

- Class 1, Class 3, and UCMM tags.
- Communication at a Requested Packet Interval (RPI) of 5 ms to 50,000 ms (this feature is not assured during offline commissioning. A recommended RPI is ≥ 100 ms).

21.5.2 Slave Offline Configuration

21.5.2.1 Overview

Example requirements: The Easy523 slave runs in the simulation version and communicates with the Easy523 master through Class 1 tags.

Device example:

Device Type	Device IP Address
Easy523 slave (simulation)	192.168.1.253
Easy523 master (physical)	192.168.1.103

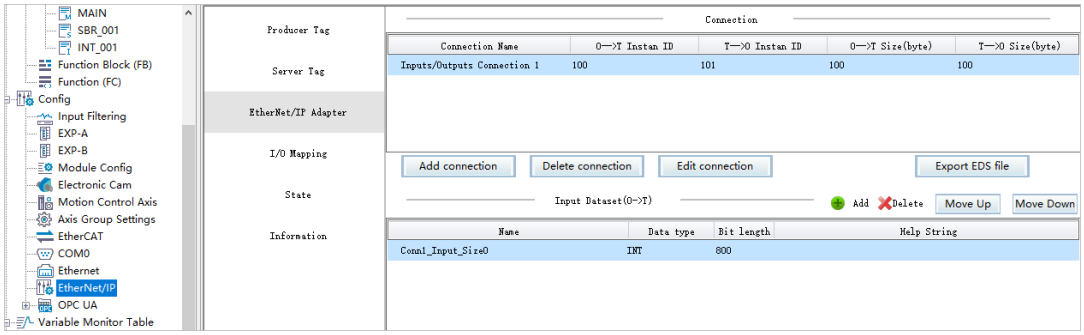
21.5.2.2 Configuring a Slave (Simulation)

Preconditions

- Only one network adapter is enabled in the PC. All other enabled network adapters must be disabled.
- Only one EtherNet/IP simulation version is running in the PC.
- The EtherNet/IP services of third parties are disabled. The EtherNet/IP port 44818/2222 is not occupied.

Procedure

1. Create a project. For details, see Creating or Opening a Project.
2. In the "Project Manager" tree on the left, double-click "EtherNet/IP". In the expanded menu, click "EtherNet/IP Adapter". The "EtherNet/IP Adapter" page is displayed.



3. Click "Add connection. The default data size is 100 bytes,

21.5.2.3 Configuring a Master (Physical)

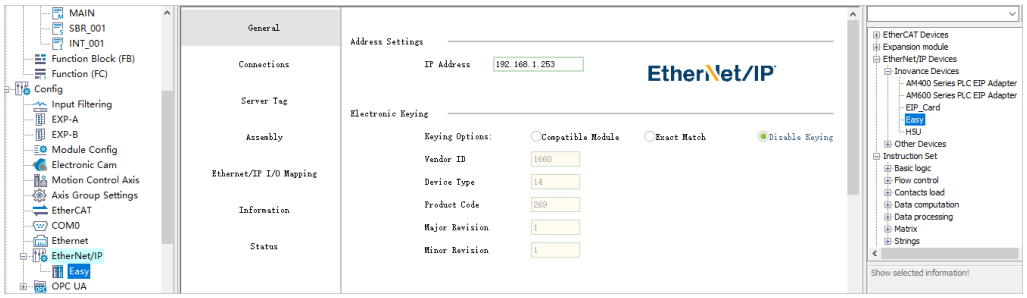
Preconditions

The PC is normally connected to the Easy523 master network.

Procedure

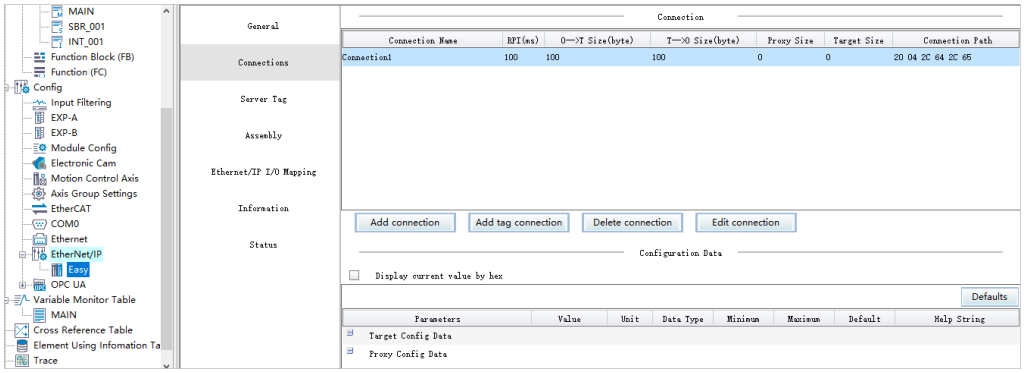
1. Create a project. For details, see Creating or Opening a Project.
2. Add the Easy523 slave as follows.
   - Method 1: In the toolbox on the right, choose "EtherNet/IP Devices" > "Inovance Devices", and then double-click "Easy" to add the Easy523 slave.
   - Method 2: In the "Project Manager" tree on the left, right-click "EtherNet/IP". In the shortcut menu displayed, select "Auto Scan". In the dialog box displayed, click "Start Scan". After the scan operation is completed, click "Update Config". Then the Easy523 slave is added.

3. Double-click "Easy". The "General" page is displayed.



4. Set "IP Address" to the IP address "192.168.1.253" of the Easy523 slave, and set "Keying Options" to "Disable Keying".

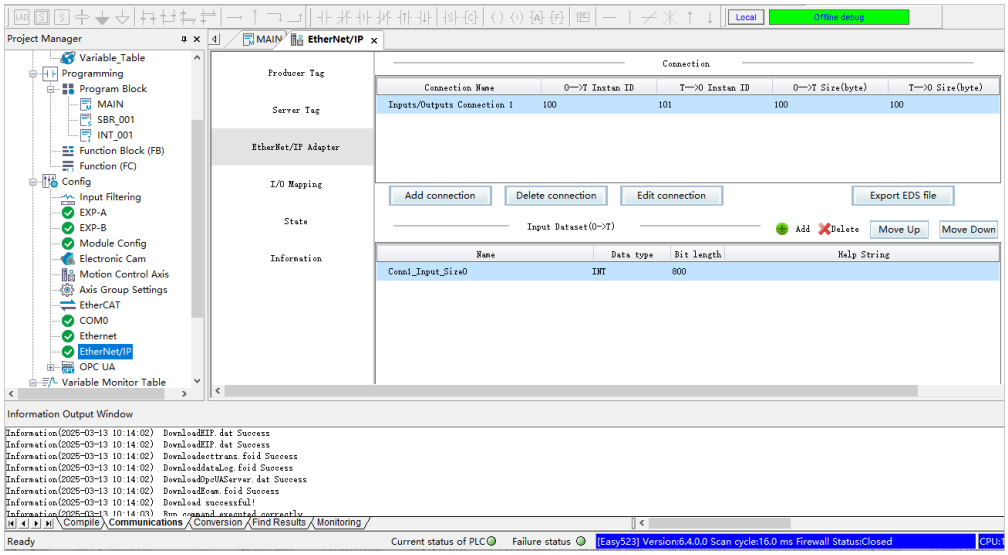
5. Click "Connect" and then click "Add connection". Set RPI to 100 ms.



21.5.2.4 Commissioning

Commissioning the slave (simulation)

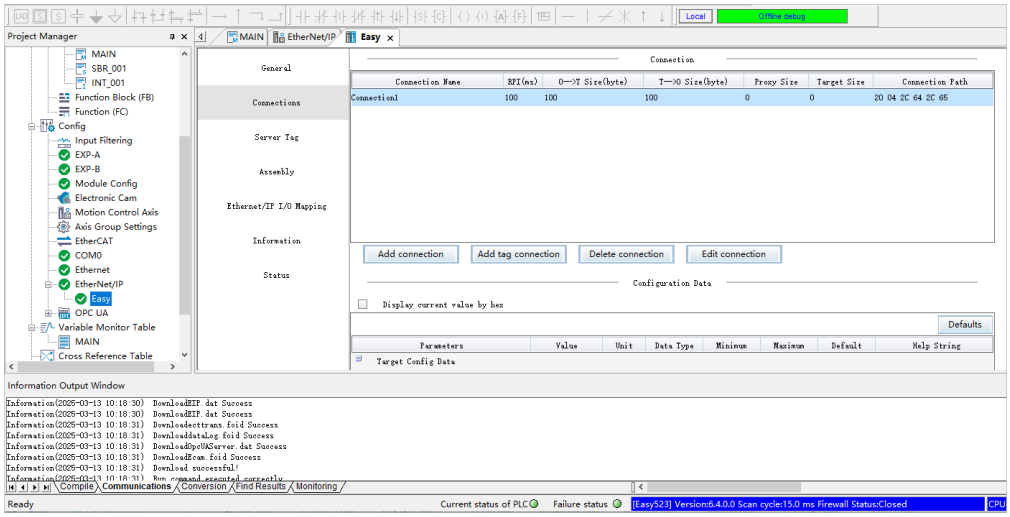
1. In the toolbar, click to access the offline commissioning page.



2. View the operation status.
   When a green icon is displayed behind "Current status of PLC" and "Failure status" in the status bar, the slave runs normally.

Commissioning the master (physical)

1. Download and run the program. For details, see Downloading a Program.



2. View the operation status.
   When a green icon is displayed behind "Current status of PLC" and "Failure status" in the status bar, the master runs normally and a Class 1 tag communication connection is established.

21.5.3 Master Offline Commissioning

21.5.3.1 Overview

Example requirements: The Easy523 master runs in the simulation version and communicates with the Easy523 slave through Class 1 tags.

Device example:

Device Type	Device IP Address
Easy523 slave (physical)	192.168.1.103
Easy523 master (simulation)	192.168.1.253

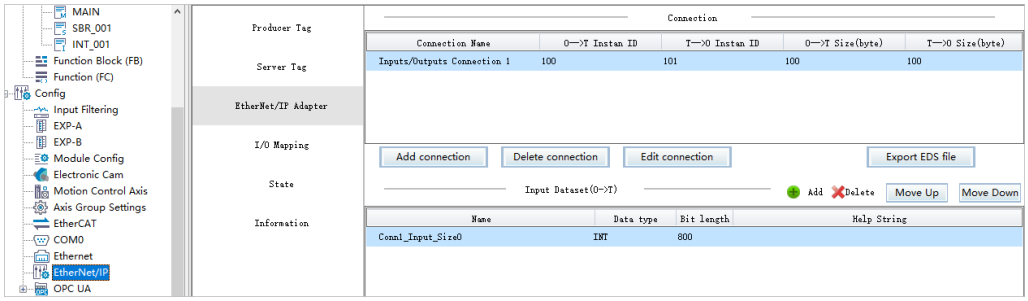
21.5.3.2 Configuring a Slave (Physical)

Preconditions

The Easy523 slave is connected to the PC network.

Procedure

1. Create a project. For details, see Creating or Opening a Project.
2. In the "Project Manager" tree on the left, double-click "EtherNet/IP". In the expanded menu, click "EtherNet/IP Adapter". The "EtherNet/IP Adapter" page is displayed.



3. Click "Add connection". The default data size is 100 bytes,

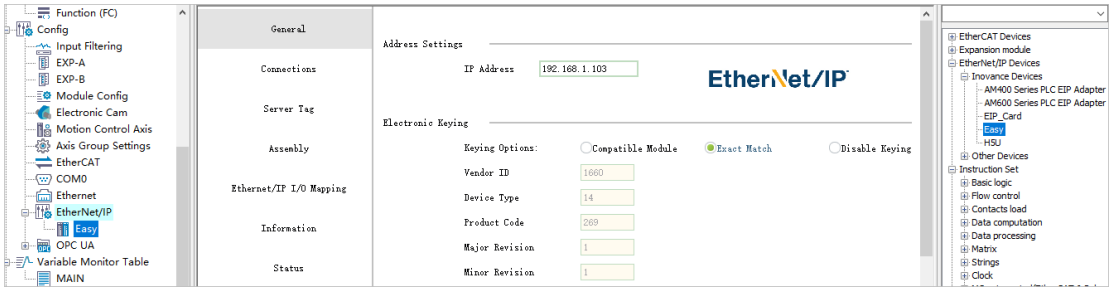
21.5.3.3 Configuring a Master (Simulation)

Preconditions

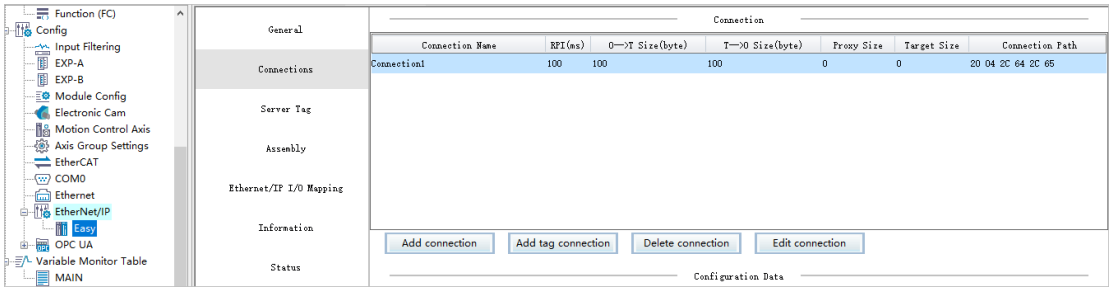
- Only one network adapter is enabled in the PC. All other enabled network adapters must be disabled.
- Only one EtherNet/IP simulation version is running in the PC.
- The EtherNet/IP services of third parties are disabled. The EtherNet/IP port 44818/2222 is not occupied.

Procedure

1. Create a project. For details, see Creating or Opening a Project.
2. Add the Easy523 slave as follows.
   - Method 1: In the toolbox on the right, choose "EtherNet/IP Devices" > "Inovance Devices", and then double-click "Easy" to add the Easy523 slave.
   - Method 2: In the "Project Manager" tree on the left, right-click "EtherNet/IP". In the shortcut menu displayed, select "Auto Scan". In the dialog box displayed, click "Start Scan". After the scan operation is completed, click "Update Config". Then the Easy523 slave is added.
3. Double-click "Easy". The "General" page is displayed.



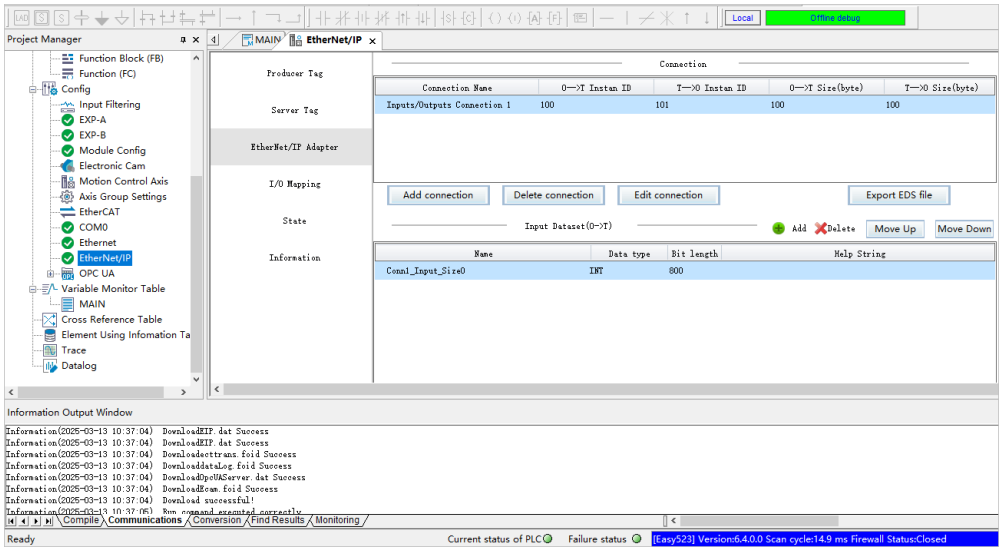
4. Set "IP Address" to the IP address "192.168.1.103" of the Easy523 slave.
5. Click "Connections" and then click "Add connection". Set RPI to 100 ms.



21.5.3.4 Commissioning

Commissioning the slave (physical)

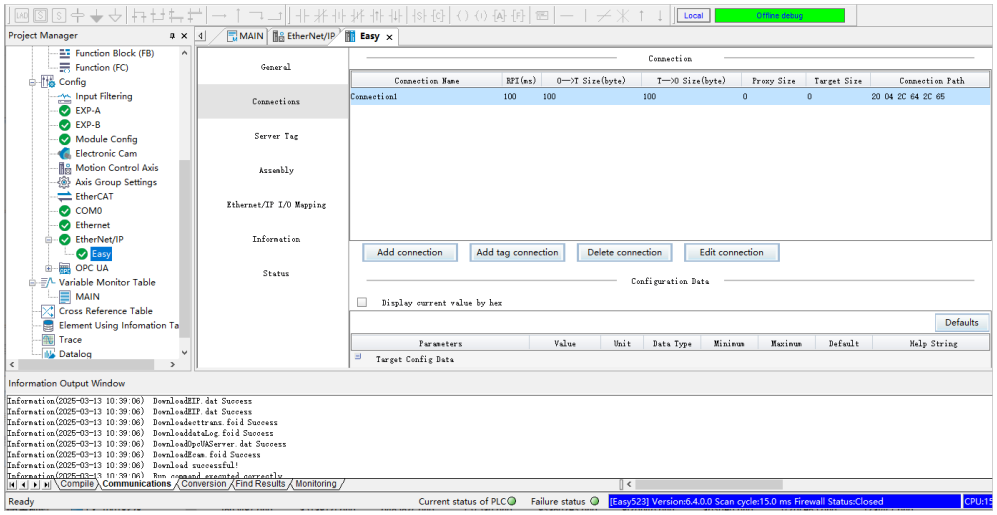
1. Download and run the program. For details, see Downloading a Program.



2. View the operation status.
   When a green icon is displayed behind "Current status of PLC" and "Failure status" in the status bar, the slave runs normally.

Commissioning the master (simulation)

1. In the toolbar, click to access the offline commissioning page.



2. View the operation status.
   When a green icon is displayed behind "Current status of PLC" and "Failure status" in the status bar, the master runs normally and a Class 1 tag communication connection is established.

21.6 OPC UA Offline Commissioning

Introduction

AutoShop allows OPC UA servers to run in the Windows simulation version for temporary offline commissioning without PLC.

The OPC UA simulation version has the same functions of the PLC.

Preconditions

- Applicable models: Easy521, Easy522, and Easy523, and Easy320
- Software version: AutoShop V4.10.2.0 and later
- Only one network adapter is enabled in the PC. All other enabled network adapters must be disabled.

Procedure

1. Choose "Control Panel" > "Network and Sharing Center" > "Change Adapter Settings", and set the network adapter state, as shown in the following figure. Right-click network adapters to be disabled, and then select "Disable".
2. When the Windows 10 system is connected to a Wi-Fi, a virtual network adapter named "microsoft wi-fi direct virtual adapter" is automatically created. This network adapter is mainly used for the Wi-Fi adapter automatically generated by wireless screen projection, such as projection through a mobile phone or a computer to the current computer. As a result, other network adapters are displayed when you run ipconfig to view the adapters on the terminal, as shown in the following figure, and thus the IP address cannot be obtained during offline commissioning. Therefore, this virtual network adapter must be disabled.
   You can disable the virtual network adapter in any of the following two ways.
   Method 1: Choose "Settings" > "System" > "Project to this computer", and disable this function.
   Method 2:
   a. Access "Control Panel", and choose "Device Manager" > "View" > "Display Hidden Devices", as shown in the following figure.
   b. Right-click "Network Adapter", select "microsoft wi-fi direct virtual adapter", and then select "Uninstall Device (U)", as shown in the following figure.
3. For the application example, see 14.3 Application Example in section 14.3.

22 Memory Management

22.1 Overview

Memory management includes customized variable and soft element variable memory management. Variable memory data at a specific moment can be obtained and used as the basis for commissioning and analysis. Variable memory data at a specific moment can also be saved as recipe data for commissioning parameters of different processes or recipe parameters of multiple steps of one process.

Specifically, variable memory data can be used in the following scenarios:

- When the program encounters an exception, obtain the current variable memory data for problem analysis.
- Obtain multiple sets of variable memory data parameters at a specific moment and save them as recipe parameter files for use by other machines.
- Monitor the values of all the data in the current variable table in real time.
- Synchronize the current variable memory data parameters to the initial values.
- Save data of different recipe parameters when commissioning different processes of one program.
- Save different sets of recipe parameters when commissioning different steps of one process.

22.2 Memory Management of Customized Variable Table

22.2.1 Expanding and Collapsing Complex Type Variables

Complex type variables contained in the customized variable table are arrays and structures. The system supports expanding and collapsing sub-members of arrays and structures. For sub-members, only the initial value, comment, and data columns are editable, while values in other columns are not editable.

Variable...	Data Type	Initial Value	Power Down Hold	Comment	Element Addr.	Length	CurValue	Value1
Axis1	BOOL	OFF	Non Retained			nBitLen:1		
⊟ Axis_30	REAL[5]	...	Non Retained			nBitLen:0		
Axis_30[0]	REAL	0.000000						
Axis_30[1]	REAL	0.000000						
Axis_30[2]	REAL	0.000000						
Axis_30[3]	REAL	0.000000						
Axis_30[4]	REAL	0.000000						
Reset_Flag	BOOL	OFF	Non Retained			nBitLen:1		

22.2.2 Monitoring Variables

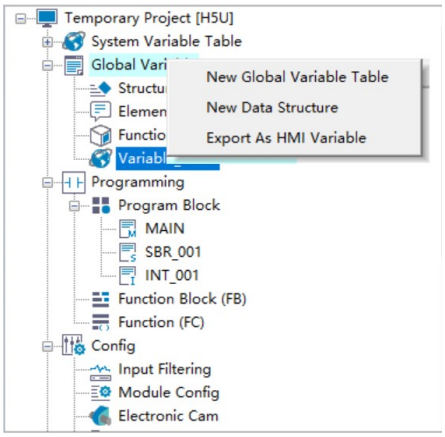
The variable table supports monitoring. Click the "Download and Monitor" button in the user project, and the system will monitor all variables displayed on the current variable table page without the need to add variables to the monitoring table separately.

Variable...	Data Type	Initial Value	Power Down Hold	Comment	Element Addr.	Length	CurValue	Val
Axis1	BOOL	OFF	Non Retained			nBitLen:1	OFF	OFF
⊟ Axis_30	REAL[5]	...	Non Retained			nBitLen:160		
Axis_30[0]	REAL	0.000000					0.000000	0.000
Axis_30[1]	REAL	0.000000					0.000000	0.000
Axis_30[2]	REAL	0.000000					0.000000	0.000
Axis_30[3]	REAL	0.000000					0.000000	0.000
Axis_30[4]	REAL	0.000000					0.000000	0.000
Reset_Flag	BOOL	OFF	Non Retained			nBitLen:1	OFF	OFF

22.2.3 Reading and Writing Memory Data

In the running state, for a single variable table, select the specified array or structure, right-click it, and select "Write Memory" to write non-null data to the PLC. In the monitoring state, right-click and select "Read Memory" to read the current variable memory data from the PLC to the selected data column of the variable table.

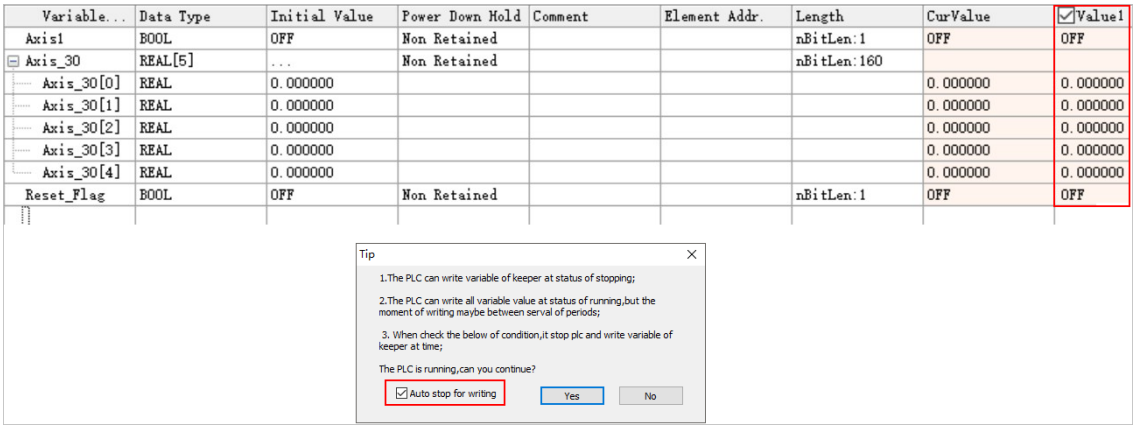
In the running state, you can upload or download newly-added variable values through one click. Specifically, right-click "Global Variable", and select "Upload engineering variable values" to batch read the current variable memory data from the PLC to the selected data columns of the variable table; or select "Download engineering variable values" to batch write non-null data in the variable table to the PLC.



NOTE

- Select the "Auto stop for writing" option, and the PLC will automatically stop before memory data is written.
- If the "Auto stop for writing" option is not selected, memory data may not be written in the same scan cycle.

In the stop state, only retentive variables can be written.

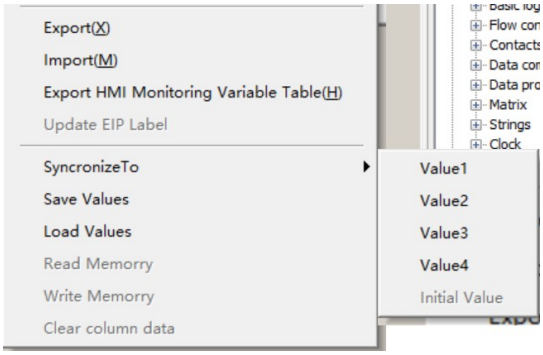


22.2.4 Synchronizing and Clearing Data

Data values, current values, and initial values can be synchronized with each other.

Synchronizing data

For example, to synchronize initial values to current values, select the target initial value item (or select multiple initial value items), right-click and select "Synchronize To" > "Value".

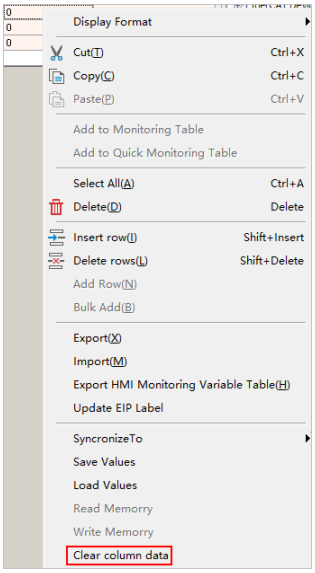


> **NOTE**
>
> Current values cannot be synchronized to initial values.

Clearing data

Select "Clear column data" to delete the currently selected data value. This operation can only be performed on data values.

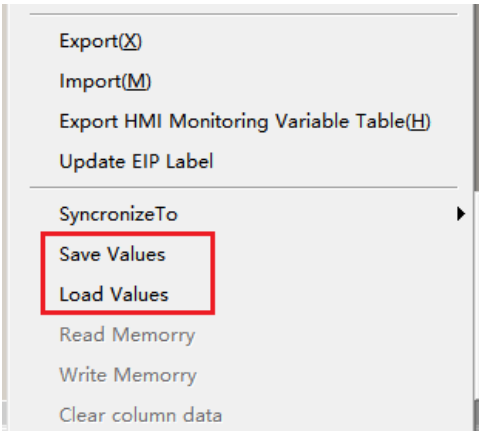
For example, to clear the Data3 column, select "Data3", right-click it, and select "Clear Column Data".



22.2.5 Saving and Loading Data

You can select “Save Values” in the shortcut menu to save specified data values in a CSV file. Then, open the CSV file using EXCEL to edit the data values.

You can also select “Load Values” in the shortcut menu to import the edited CSV recipe file to the PLC.



22.2.6 Editing Initial Values and Comments

You can directly expand the variable table to edit initial values and comments of member data, or define them in the customized variable box in the program. These two methods are equivalent to each other.

Variable Name	Data Type	Initial Value	Power Down Hold	Network Pubilc	Comment
cammode_1	_sMC_CAM_...	...	Non Retained	Private	
cammode_1[0]	_sMC_CAM_NODE	...			
fPhase	REAL	0.000000			Spindle Phas...
fDistance	REAL	0.000000			Displacement...
fVel	REAL	0.000000			Connection s...
fAcc	REAL	0.000000			Connection a...
iCuve	INT	0			Curve type (...

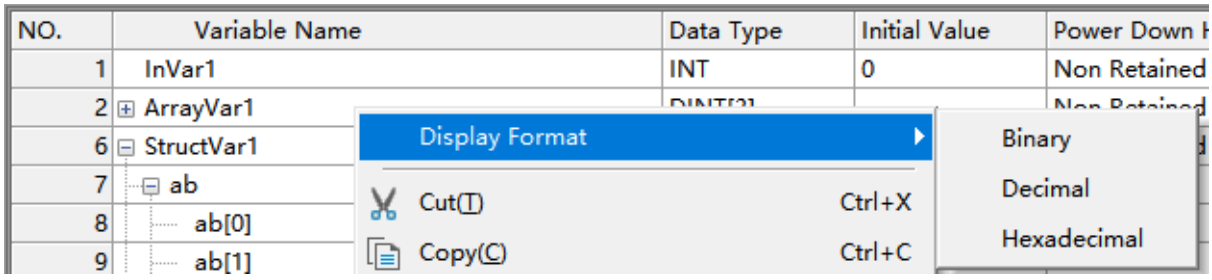
NOTE

English characters and special characters such as commas (,), brackets ([]), and parentheses (()) in comments of variable member data are automatically filtered out.

22.2.7 Changing and Displaying Base

Customized variable tables allow switchover between decimal, binary, and hexadecimal displays.

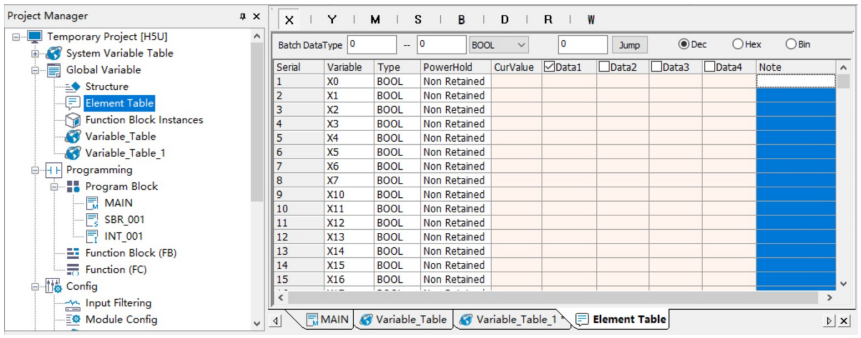
- Binary inputs can only be 0 and 1 and must be prefixed by 2#.
- Hexadecimal inputs can only be abcdef and digits and must be prefixed by 0x.
- Decimal inputs can only be digits. Other characters are all invalid.



22.3 Memory Management of Soft Elements

22.3.1 Operation Interfaces

The element table provides the comment and memory management functions, as shown in the following figure.



① Soft element switchover area. Switch to different soft elements to manage their memories and comments.

② This area is used to batch modify the data type of selected soft elements on the current page. To select soft elements, enter the start and end numbers of the elements in the two edit boxes..

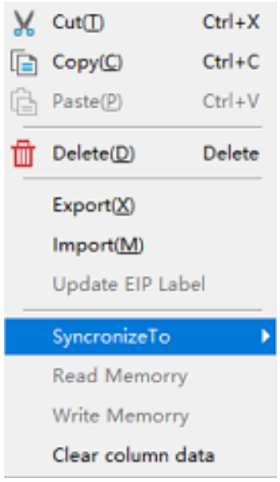
③ Click the drop-down box to select the data type to be changed to. For example, switch to the D elements, enter 50 and 100 to select the D50 to D100 elements, and then click the drop-down box to select REAL. The data type of D50, D52, D54…D100 is changed to REAL. (Note: REAL and DINT each are composed of 32 bits, occupying two soft elements.)

④ This area can redirect you to a specific element. For example, switch to the D elements, enter 1000 in this box, and then click “Jump” or press Enter. The page automatically redirects to the D1000 element. (Note: For the X and Y elements, only octal numbers of up to five digits can be entered in an input box. If the entered number exceeds the total number of elements, the page redirects to the end.)

⑤ This area is used to change the display format of the data value and current value columns. For example, clicking the “Hex” option will switch all data to hexadecimal display.

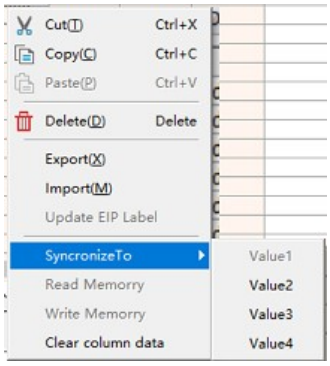
22.3.2 Data Operation

The shortcut menu of soft elements provides multiple function options, as shown in the following figure.

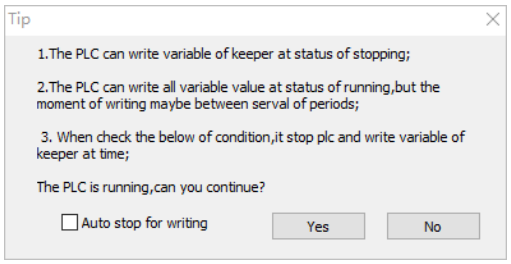


- The Cut, Copy, Paste, and Delete options are available for the value columns and the comment column.
- The Edit option is available for only six columns, including the data type column, value columns, and comment column.
  The value columns can be edited in any mode, while other columns cannot be edited in the monitoring mode. All these columns can be edited in the online modification mode.
- Import and Export: Only rows with modified data can be exported. If there is any data error in a row during import, the system reports the error in the information output window and skips the row. When the number of error rows exceeds 100, the import stops.

- Synchronize To: synchronizes all data in a selected column to a value column selected using the menu. For example, right-click a cell in the "Value1" column and choose "Synchronize To" > "Value2" to overwrite the Value2 column with the data in the Value1 column.



- Read Memory and Write Memory: Reading or writing soft element memories is allowed only in the online modification mode or monitoring mode. On the interface of a single element, tick the header of a value column, right-click and select "Read Memory" or "Write Memory" to read or write to the specific column you ticked. To read or write data values to the entire soft element table in batch, right-click "Global Variable", and select "Upload engineering variable values" to batch read the current variable memory data from the PLC to the selected data columns of the variable table; or select "Download engineering variable values" to batch write non-null data in the variable table to the PLC.

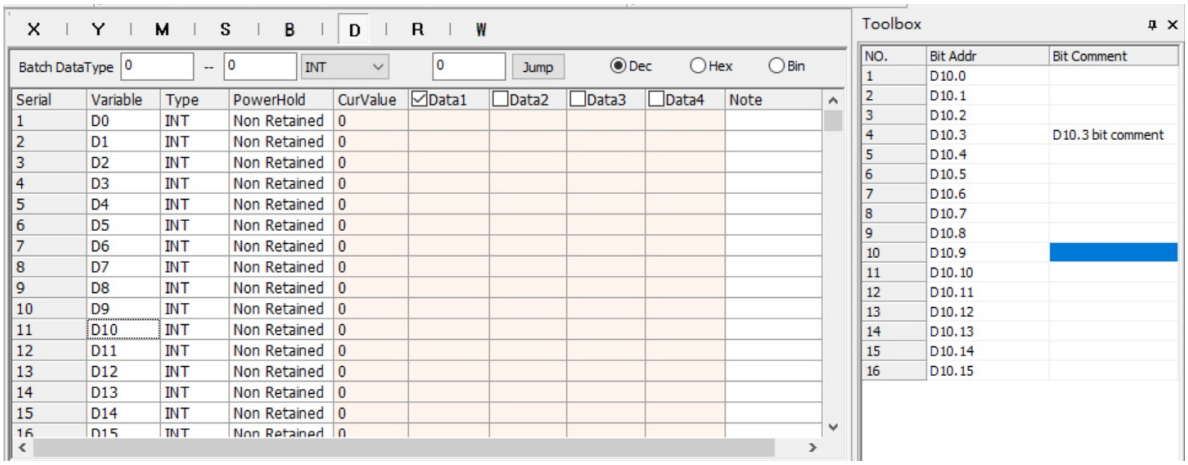


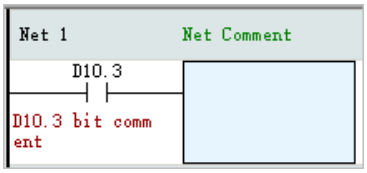
- Clear column data: Clears all data in a column.

22.3.3 Bit Comments

Select word elements (D, R, or W) in the soft element table and then select a row. Bit comments of the row are displayed in the toolbox area. You can edit the bit comments in the toolbox area.

For example, select D10, edit the comment of D10.3, and save the modification. As a result, the bit comment is displayed in the ladder diagram.





22.3.4 Rules for Editing Different Data Types

1. For bit elements (X, Y, M, S, and B), the data type can only be set to BOOL, and the values are either ON or OFF.
2. For word elements, the data type can be set to INT, DINT, or REAL. An INT element is composed of 16 bits, whereas a DINT or REAL element is composed of 32 bits.
3. A 32-bit word element occupies two element rows. Therefore, when a word element is set as the REAL or DINT type, values in the row next to the word element are automatically cleared and data type of that row cannot be edited. In addition, the data column of the row preceding the word element cannot be edited. In memory reading, the row next to a REAL or DINT element are skipped.

22.4 Function Example

Save different sets of recipe parameters when commissioning different steps of one process.

1. Create a standard project, define variables and structures and write the program for the project, and then download the project to the PLC.
2. Enter project monitoring.
3. Open the variable table and tick “Value1”. At a specific moment, right-click and select “Read Memory” to read the variable memory as the recipe parameters for the step of the specific moment. As a result, the variable memory values obtained at the specific moment are displayed in the “Value1” column. In batch reading of memory values, the memory values read through right-click at a specific moment will be read to the entire selected value column of the variable table, whether open or not, and will be used as the recipe parameters for that step.

	Variable Name	Data Type	Initial V...	Power Dow...	Comment	Element A...	Length	CurValue	☑Value1	☐Value2	☐Value3	☐Value4
46	gh_fCutLength_Min	REAL	10.00000	Retained	Wr_auto parameter_minimum cut length		nBitLen:32	0.000000	0.000000			
47	gh_fCutLength_Max	REAL	500.0000	Retained	Wr_auto parameter_maximum cut length		nBitLen:32	0.000000	0.000000			
48	gh_fSetMachineVel	REAL	50.00000	Retained	Wr_auto parameter_machine running velocity		nBitLen:32	0.000000	0.000000			
49	gh_fSetMachineJogVel	REAL	10.00000	Retained	Wr_auto parameter_machine jogging velocity		nBitLen:32	0.000000	0.000000			
50	gh_fSetMachineOnceVel	REAL	0.000000	Retained	Wr_auto parameter_machine once velocity		nBitLen:32	0.000000	0.000000			
51	gh_fSetMachineMaxVel	REAL	0.000000	Retained	Wr_auto parameter_maximum machine running velocity		nBitLen:32	0.000000	0.000000			
52	gh_fSetMachineMinVel	REAL	0.000000	Retained	Wr_auto parameter_minimum machine running velocity		nBitLen:32	0.000000	0.000000			
53	gh_diSetMachineStartDelay	DINT	0	Retained	Wr_auto parameter_machine start delay		nBitLen:32	0	0			
54	gh_fSetMachineRunAccTime	REAL	5.000000	Retained	Wr_auto parameter_machine running acceleration time		nBitLen:32	0.000000	0.000000			
55	gh_fSetMachineStartTime	REAL	3.000000	Retained	Wr_auto parameter_machine start time		nBitLen:32	0.000000	0.000000			
56	gh_fSetMachineStopTime	REAL	3.000000	Retained	Wr_auto parameter_machine stop time		nBitLen:32	0.000000	0.000000			
57	gh_fSetMachineEStopTime	REAL	2.000000	Retained	Wr_auto parameter_machine emergency stop time		nBitLen:32	0.000000	0.000000			
58	gh_diSealOpenDelayAfte...	DINT	100	Retained	Wr_auto parameter_seal open delay time after stop		nBitLen:32	0	0			
59	gh_diStartDelayAfterSe...	DINT	100	Retained	Wr_auto parameter_start delay after seal open		nBitLen:32	0	0			
60	gh_bMachineParamInitial	BOOL	OFF	Non Retained	Wr_function_restore factory defaults		nBitLen:1	OFF	OFF			
61	gh_bKnifeCamDisable	BOOL	OFF	Non Retained	Wr_function_knife Cam disable		nBitLen:1	OFF	OFF			
62	gh_bFilmCamDisable	BOOL	OFF	Non Retained	Wr_function_film Cam disable		nBitLen:1	OFF	OFF			
63	gh_bMaterialCamDisable	BOOL	OFF	Non Retained	Wr_function_material Cam disable		nBitLen:1	OFF	OFF			
64	gh_bPackageMake	BOOL	OFF	Non Retained	Wr_function_package make (no feeding)		nBitLen:1	OFF	OFF			

4. Similarly, tick “Value2”. At another moment, right-click and select “Read Memory” to obtain recipe parameters for another step and have them displayed in the “Value2” column. Batch reading of memory values is similar to the previous operation. Values obtained through batch reading are used as the recipe parameters for another step.
5. Save the recipe. Before saving the recipe, you can edit some of the obtained values (or export and edit them in an EXCEL file and then load the file). After editing, right-click and select “Save” to save the recipe data of different steps in a CSV file in the disk for backup.
6. Edit recipe values. Open the saved recipe CSV file in Excel and fine-tune the variable parameters.

	FullName	DataType	Vaule1	value2	value3	value4
3	gh_bAuto	BOOL	ON	ON		
4	gh_bStart	BOOL	OFF	OFF		
5	gh_bCycleStop	BOOL	OFF	OFF		
6	gh_bEstop	BOOL	OFF	OFF		
7	gh_bReset	BOOL	OFF	OFF		
8	gh_bMachineJog	BOOL	OFF	OFF		
9	gh_bMachineOnce	BOOL	OFF	OFF		
10	gh_bProductNum_ReReset	BOOL	OFF	OFF		
11	gh_bProductNum_NoReRes	BOOL	OFF	OFF		
12	gh_bMaterialOffsetAdd	BOOL	OFF	OFF		
13	gh_bMaterialOffsetSub	BOOL	OFF	OFF		
14	gh_bColorMarkOffsetAdd	BOOL	OFF	OFF		
15	gh_bColorMarkOffsetSub	BOOL	OFF	OFF		
16	gh_bKnifePower	BOOL	OFF	OFF		
17	gh_bFilmPower	BOOL	OFF	OFF		

7. Load the recipe. After saving the data modified in step 6, right-click in the variable table and click “Load Values”. As a result, the modified configuration parameters are displayed in the “Value1” and “Value2” columns.

8. Write the recipe values. Tick “Value2”, right-click and select “Write Memory” to write the recipe parameters for the corresponding step into the memory.

23 Configuring the PID Function

Introduction

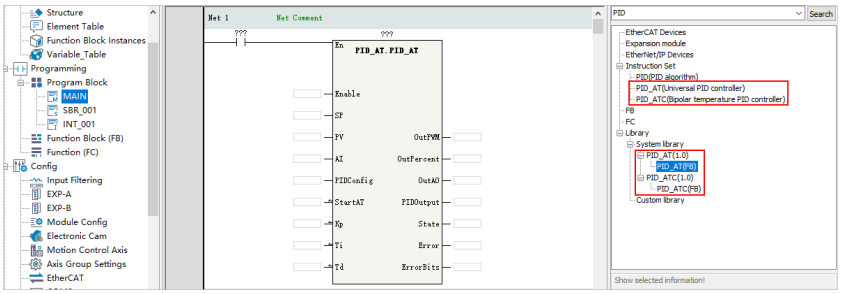
The PID wizard function guides users to configure parameters on visible pages that can be configured offline and commissioned online, to improve the ease of use of PID instructions.

Preconditions

- Applicable models
  - All Easy500 series: Easy523, Easy522, Easy521, Easy502, and Easy501
  - All Easy300 series: Easy320, Easy302, and Easy301
- Firmware version: V6.4.0.0 and later (versions earlier than V6.4.0.0 already support the PID instructions, and thus this function is not affected by the firmware version)
- Software version: AutoShop V4.10.2.0 and later

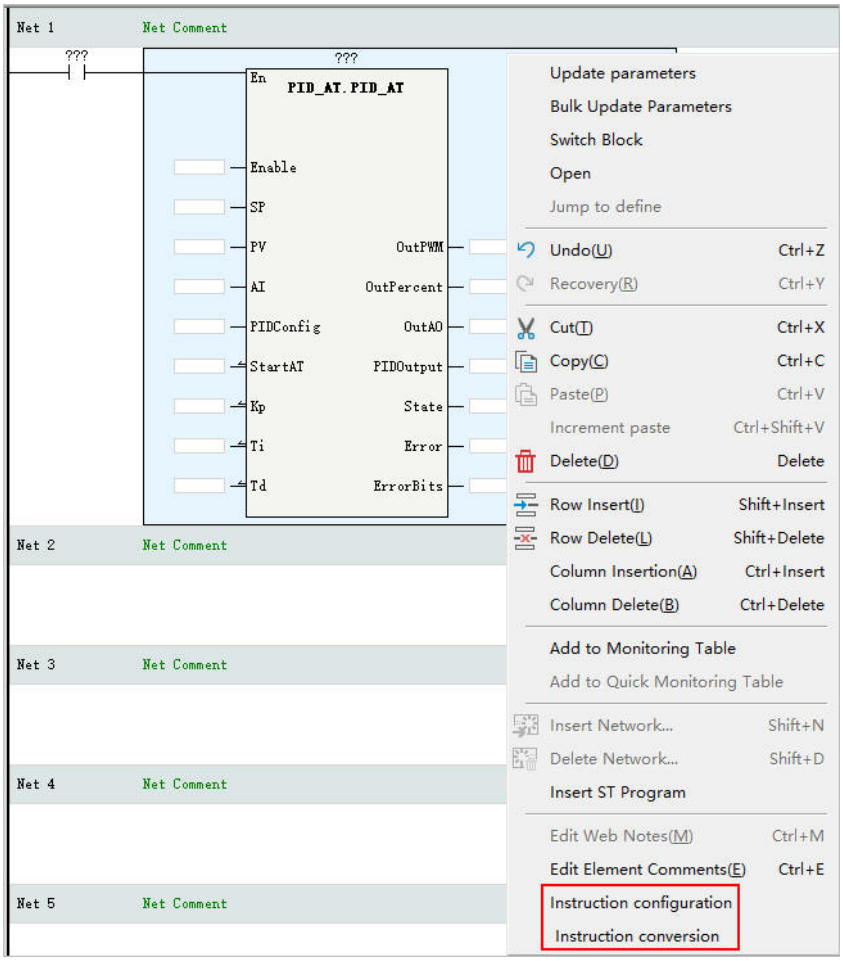
Procedure

1. Access the PID instruction configuration and PID instruction commissioning pages.
   a. PID instances are implemented by FBs of the system library. PID instructions are encapsulated into system library files and added to the system library of the toolbox by default. After the software of V4.10.2.0 and later versions are installed, library files named PID_AT and PID_ATC are generated by default. Right-click the PID instruction interface in the original instruction set and click "Add", the FB instance is added. The results of the two methods are the same.

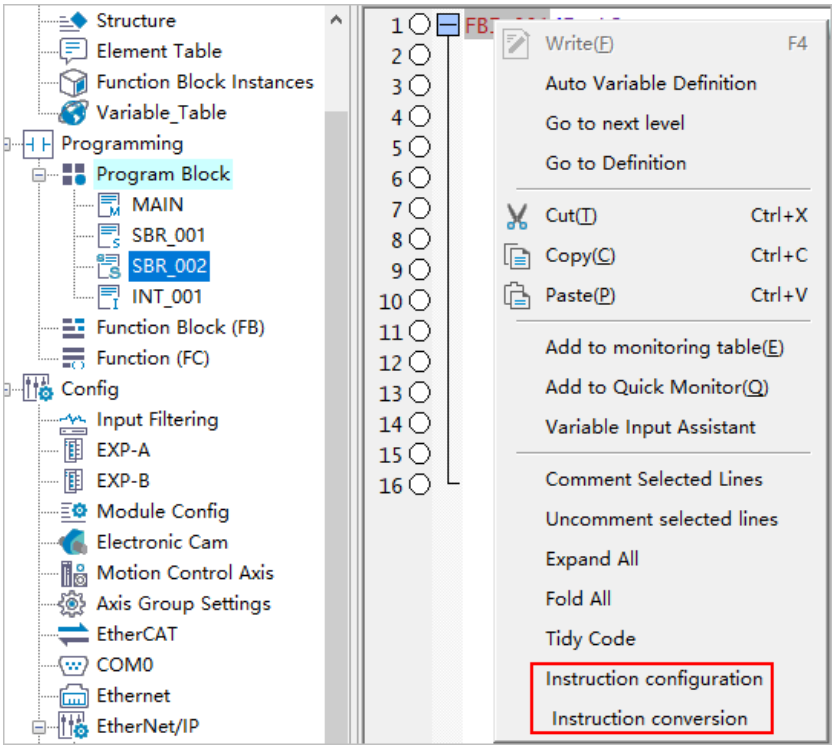


   b. In the LD program, double-click the function block in the toolbox or directly input the PID_AT and PID_ATC text to add the FB and name the FB instance. Then, right-click the instruction to view the

"Instruction configuration" and "Instruction conversion" menus, as shown in the following figure.



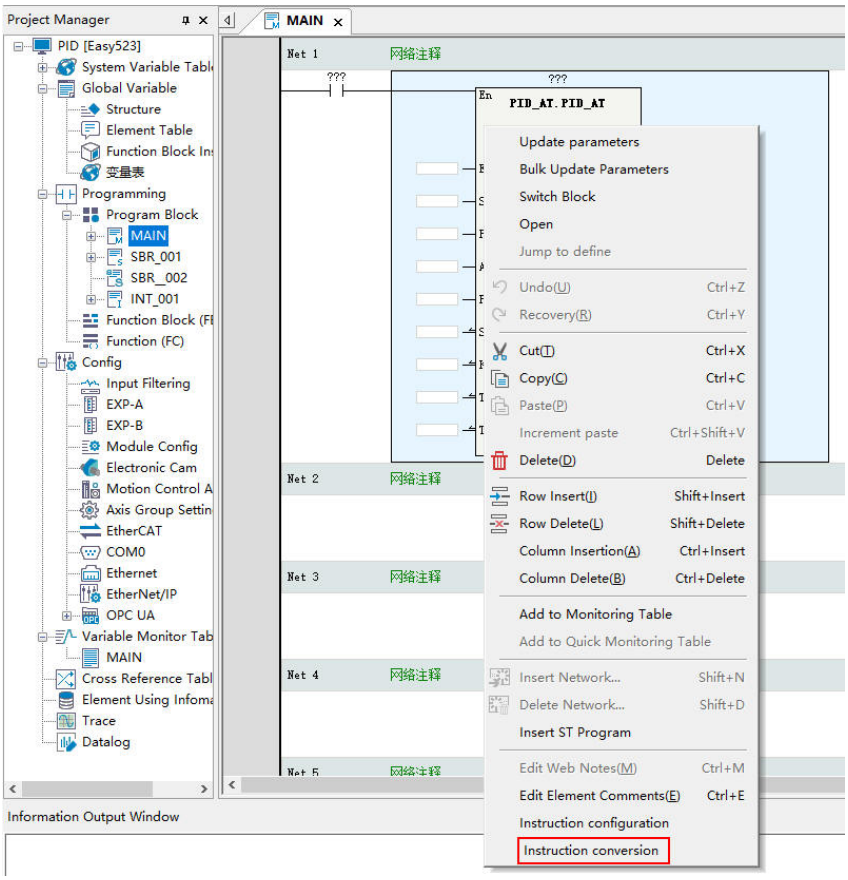
c. In the ST program, double-click the function block in the toolbox or directly input the PID_AT and PID_ATC text to add the FB and name the FB instance. Then, right-click the PID_AT or PID_ATC instruction. A context menu is displayed, in which you can select "Instruction configuration" or "Instruction conversion", as shown in the following figure.



PID_AT and PID_ATC instructions are already supported by the software earlier than V4.10.2.0. When you open an old project in the software of V4.10.2.0, the original PID instruction is still an

instruction. If you do not need the PID wizard function, use the instruction normally.

If you need the PID wizard function, right-click the PID_AT or PID_ATC instruction. The option "Instruction conversion" is used to convert the PID instruction in the old project to an FB instance of the PID. The pin parameter bound with the old project and its initial value remain unchanged after conversion, so as to guide users to the wizard function.



The following table lists PID declaration and definition specifications.

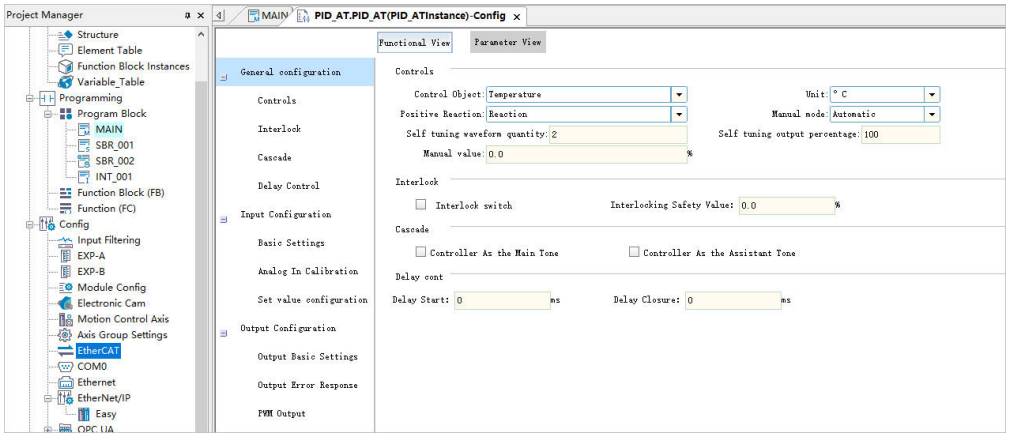
Program	PID Configuration	PID Call
LD program	Supported	Supported
ST program	Supported	Supported
FB(LD/ST)	Supported	Not supported
FB instance	Supported	Supported
FC	Declaration and definition are not supported.	Declaration and definition are not supported.

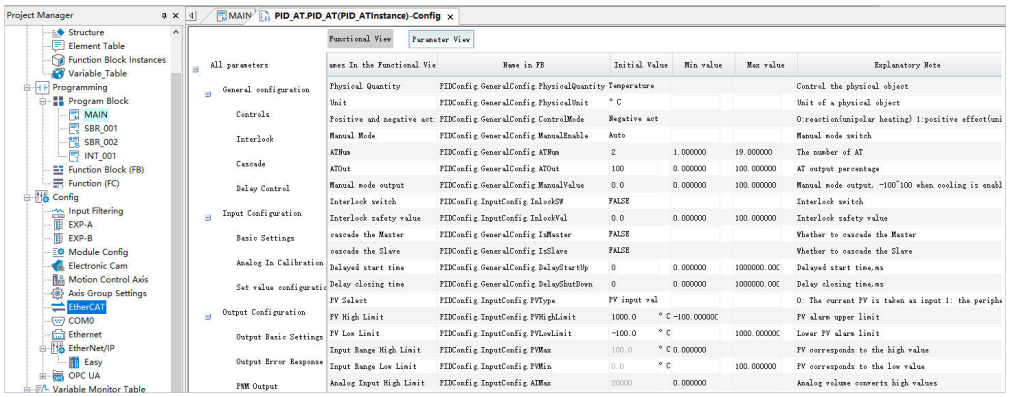
NOTE

The wizard entry is only available for the program area, not for the variable declaration area. Non-PID FB instances have no PID entry.

2. Views for configuring parameters of the PID_AT and PID_ATC instructions

a. Open the configuration page of the PID wizard. The page consists of a function view and parameter view, as shown in the following figure.

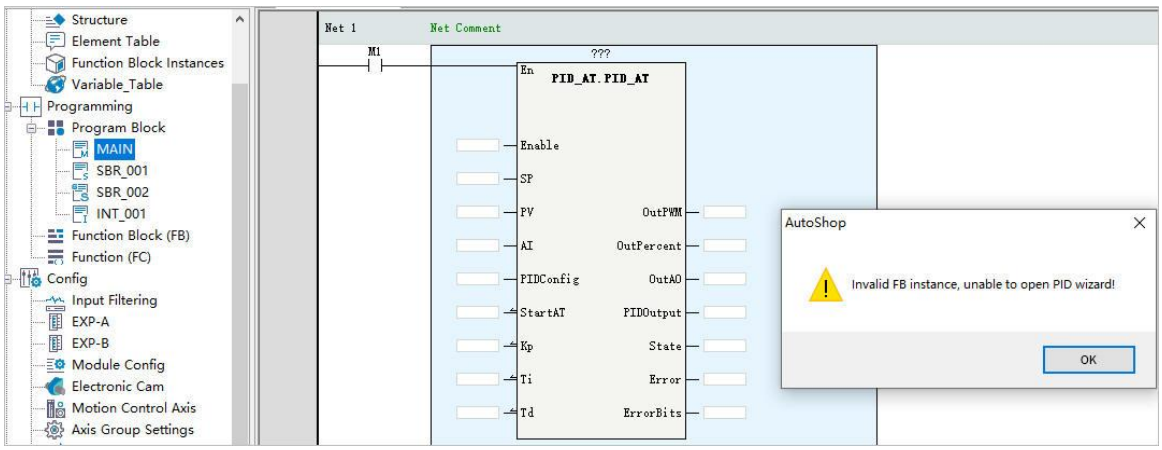




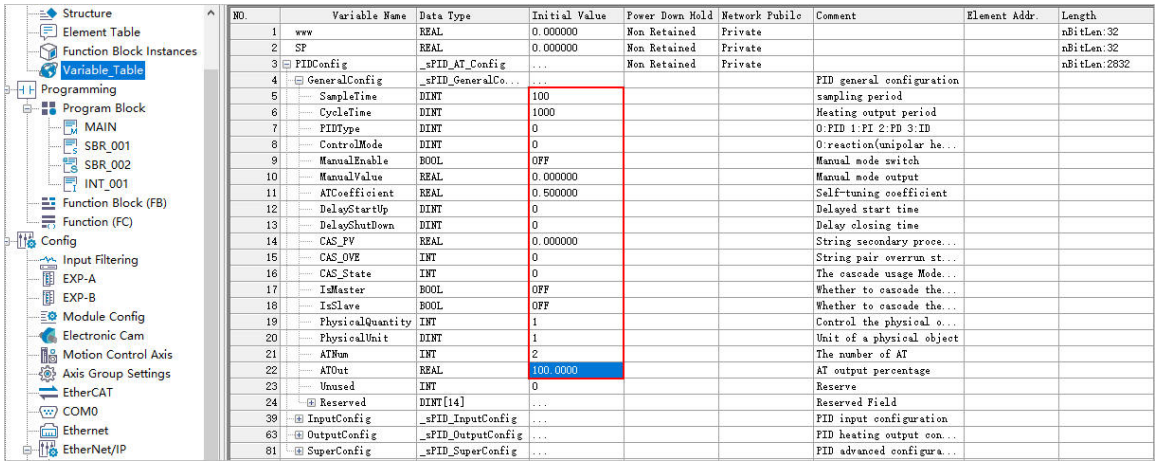
NOTE

- Default comments of the parameter view are all sourced from the function block instance. In the offline state, the initial value can be modified on the parameter view. In the online state, values in the current value column are ready-only; values in the written value column can be modified and values in the current value column are automatically updated. Other information such as name, comment, and maximum/minimum value cannot be modified.
- In the "Variable Names In the Functional View" column, hover over a variable and then its nesting is displayed.

1. Configure the PID instance. The FB instance name of the valid PID instruction must be configured. If no instance name is configured, the system prompts "Invalid FB instance, unable to open PID wizard", indicating that you cannot access the configuration and commissioning pages, as shown in the following figure.

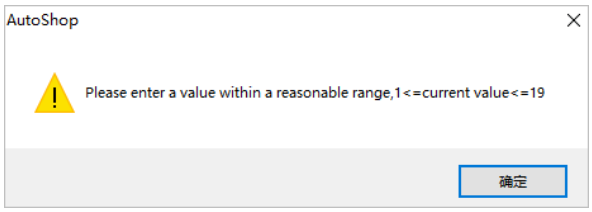


System variable structures _sPID_AT_Config and _sPID_ATC_Config provide a set of initial values. By default, the members of the corresponding structures have initial values when the variables are created, as shown in the following figure.

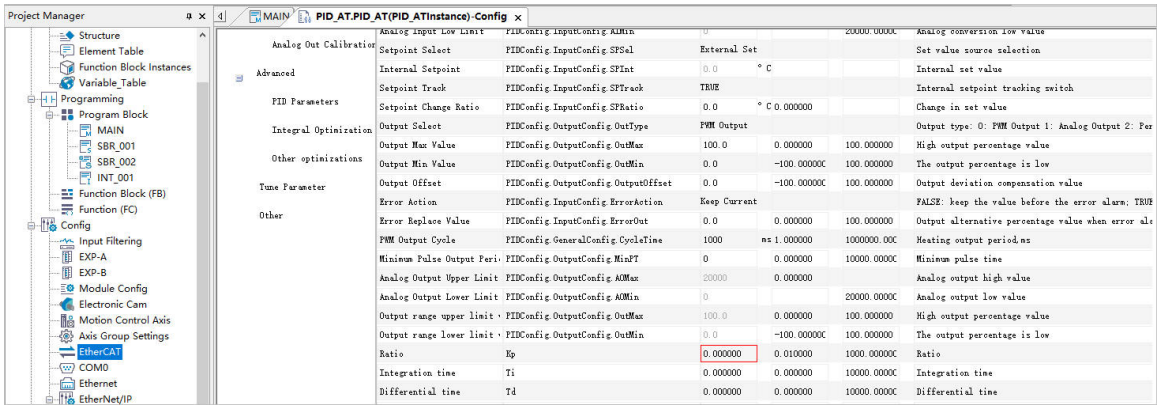


NO.	Variable Name	Data Type	Initial Value	Power Down Hold	Network Public	Comment	Element Addr.	Length
1	www	REAL	0.000000	Non Retained	Private			nBitLen:32
2	SP	REAL	0.000000	Non Retained	Private			nBitLen:32
3	PIDConfig	_sPID_AT_Config	...	Non Retained	Private			nBitLen:2832
4	GeneralConfig	_sPID_GeneralCo...	...			PID general configuration		
5	SampleTime	DINT	100			sampling period		
6	CycleTime	DINT	1000			Heating output period		
7	PIDType	DINT	0			0:PID 1:PI 2:PD 3:ID		
8	ControlMode	DINT	0			0:reaction(unipolar he...		
9	ManualEnable	BOOL	OFF			Manual mode switch		
10	ManualValue	REAL	0.000000			Manual mode output		
11	ATCoefficient	REAL	0.500000			Self-tuning coefficient		
12	DelayStartUp	DINT	0			Delayed start time		
13	DelayShutDown	DINT	0			Delay closing time		
14	CAS_PV	REAL	0.000000			String secondary proce...		
15	CAS_OVE	INT	0			String pair overrun st...		
16	CAS_State	INT	0			The cascade usage Mode...		
17	IsMaster	BOOL	OFF			Whether to cascade the...		
18	IsSlave	BOOL	OFF			Whether to cascade the...		
19	PhysicalQuantity	INT	1			Control the physical o...		
20	PhysicalUnit	DINT	1			Unit of a physical object		
21	ATNum	INT	2			The number of AT		
22	ATOut	REAL	100.0000			AT output percentage		
23	Unused	INT	0			Reserve		
24	Reserved	DINT[14]	...			Reserved Field		
39	InputConfig	_sPID_InputConfig	...			PID input configuration		
63	OutputConfig	_sPID_OutputConfig	...			PID heating output con...		
81	SuperConfig	_sPID_SuperConfig	...			PID advanced configura...		

When the FB instance of PID is initialized, it also has the default initial value, which can be modified according to the scene (offline state and online state) during application. During editing, if the input data is illegal, the system checks the input when the cursor moves outside the editing box and displays the following dialog box.



The outer border of the editing box is red, indicating that the value is below the minimum value or above the maximum value, as shown in the following figure.

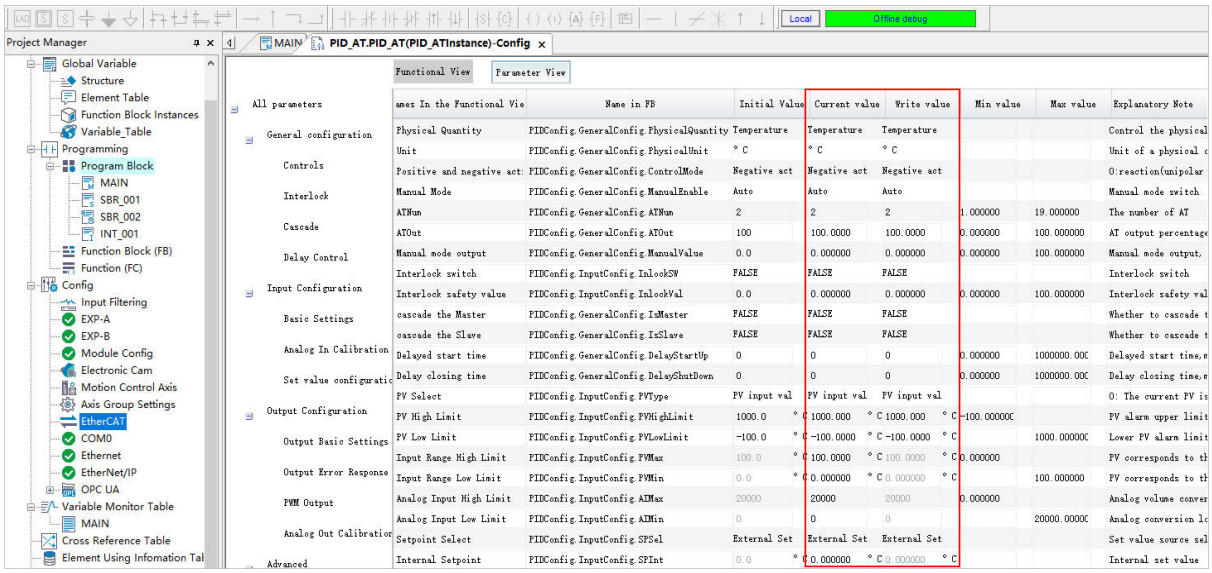


Offline state

- If the FB instance pin of PID is not bound with arguments, the initial value of the page comes from the value of the FB instance of corresponding PID. The configuration page of the wizard is refreshed synchronously with the modification of the FB instance, and the values indicated are consistent.
- If the FB instance pin of PID is bound with an argument and the initial value is modified on the configuration page, the initial value of the corresponding argument variable declaration area will be refreshed synchronously. The configuration page of the wizard is refreshed mutually synchronously with the modification of the FB instance, and the values indicated are consistent.
- The data type of FB instance pin of PID must be verified. When the data type bound by the pin does not conform to the data type defined by the pin, the system considers that no argument is bound. In this case, if you open the configuration page of PID wizard to modify the configuration, the value of FB instance of PID is modified. If the pin is bound with a legal argument, the page modification is synchronized to the value of the argument variable; otherwise, it is synchronized to the value of the member variable of the FB instance.

Online state

The current value column and the written value column are added in the parameter view. The current value column displays the current value of the parameter during monitoring, and the written value column allows you to modify the current value of the parameter, as shown in the following figure.

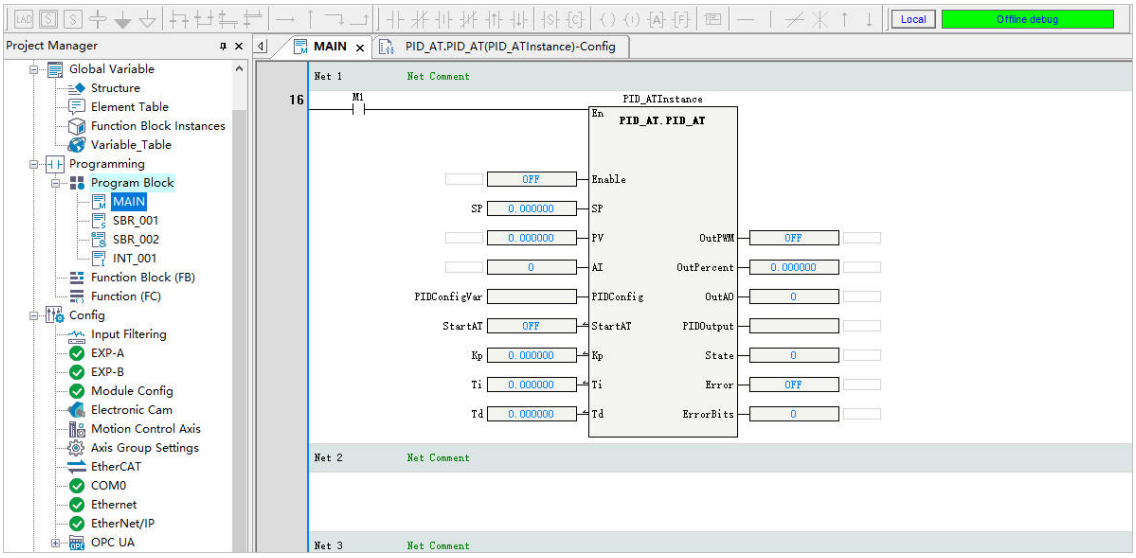


The specifications of online state are basically the same as those of offline state, but the difference is that the initial value of online state cannot be modified and in this case, the current value is modified.

- If the corresponding pin is bound with an argument, the current value of the argument is modified when writing.
- If the corresponding pin is is not bound with any argument, the current value of the FB instance is modified.

When the value is modified on the configuration page, it is equivalent to assigning a value to the variable declaration area.

- If the instance pin is bound with an argument, the initial value of the argument is directly modified.
- If the instance pin is not bound with an argument, the initial value of the parameter is directly modified, which is the same as that in the variable declaration area.



Example

When editing the parameters corresponding to the PIDConfig pin on the configuration and

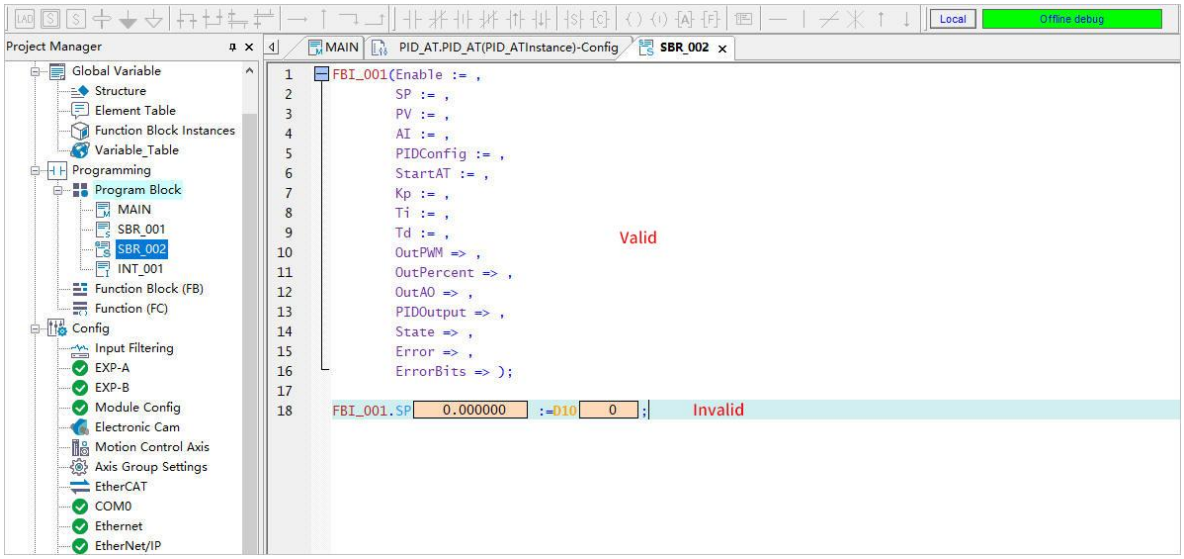
commissioning pages, the value of the structure member corresponding to the argument PIDConfigVar is actually edited.

- The function view displays the initial value in the offline state and displays the initial value and current value in the online state. When you switch between the function view and the parameter value, the initial value/current value is refreshed synchronously.
- If the modified value of a configuration is illegal, after the cursor moves outside the editing box, the pop-up box will prompt the actual valid range of data, and the editing box will automatically display the valid data value saved last time.
- Parameters that are not configured on the configuration page are obtained from the function block itself and displayed in the "Other Parameters" category of the parameter view.

> **NOTE**
>
> After the FB instance of the PID is operated, such as renamed or deleted, the opened configuration or commissioning page is automatically closed.

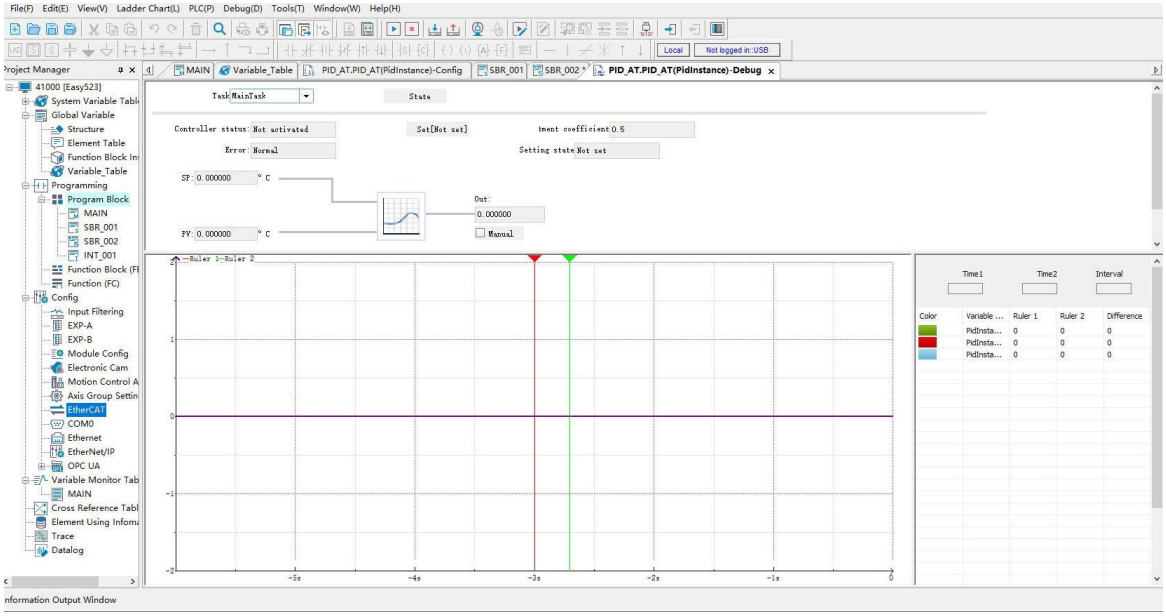
2. Bind a variable in the other parameter programming area.

ST language: It can identify the input, output, and input/output variables bound inside the function block, but the variables assigned to the pins outside the function block cannot be identified. The corresponding valid FB instance and the pin parameters in the brackets can be identified, but the expression that directly assigns values to FB instance members cannot be identified.



If the soft element or system variable is bound with a pin, it is considered as unbound in offline state, and the value of the FB instance of the PID is modified when the corresponding configuration is modified. In the online state, it is considered to be bound with an argument, and the current value can be written to it.

You can open the PID commissioning page in Step 1, as shown in the following page. The whole page is not editable in the offline state. In the online state, you can capture data, select the task type, make settings, and perform other operations on this page.



Name	Description
Task	• Its specification is consistent with the Trace task, and it supports main task and Ethercat task. Only one Trace task is allowed to capture data at the same time. If it is necessary to capture the variable waveform of another instance, you must stop the current Trace task first, and then start the Trace task that needs to be captured. • Consistent with the specification of the Trace task, the Trace function in PID and the Trace task do not affect each other. Both can be used at the same time.
Start	Click "Start" and the oscilloscope starts collecting data. Click the button again and the oscilloscope stops collecting data.
Controller status	Corresponding to the variable State, it is read-only.
Error	Corresponding to the variable ErrorBits, it is read-only. If a parameter is incorrect, an icon is displayed next to the monitoring value box, and the icon tip displays the error details.
Detailed error value logic	Obtain the value of the variable PIDOutput.ErrorID and obtain the comment from the enumeration "Controller.ERROR_ID".
Setting state	It corresponds to the variable StartAT. After this button is clicked, the set adjustment coefficient is written.
SP	• When stPIDConfigUse.InputConfig.SPSel is FALSE, the external setpoint is used. It corresponds to the variable SP. • When stPIDConfigUse.InputConfig.SPSel is TRUE, the internal setpoint is used. It corresponds to the variable stPIDConfigUse.InputConfig.SPInt.
Read	It corresponds to the variable PIDOutput.SPOut.
PV	Corresponding to the variable PV, it is read-only.
OUT	Corresponding to the variable OutPercent, it is read-only.
Manual checkbox	It corresponds to the variable PIDConfig.GeneralConfig.ManualEnable.

Name	Description
Oscilloscope page	You can right-click the page to restore the view, switch between the single channel and dual-channel, and save the configuration. The waveform cannot be uploaded. To upload the waveform, import the XML file into Trace to view the waveform.
Variable page	You do not need to add variables. Monitoring variables are added by default. Variable colors cannot be selected.

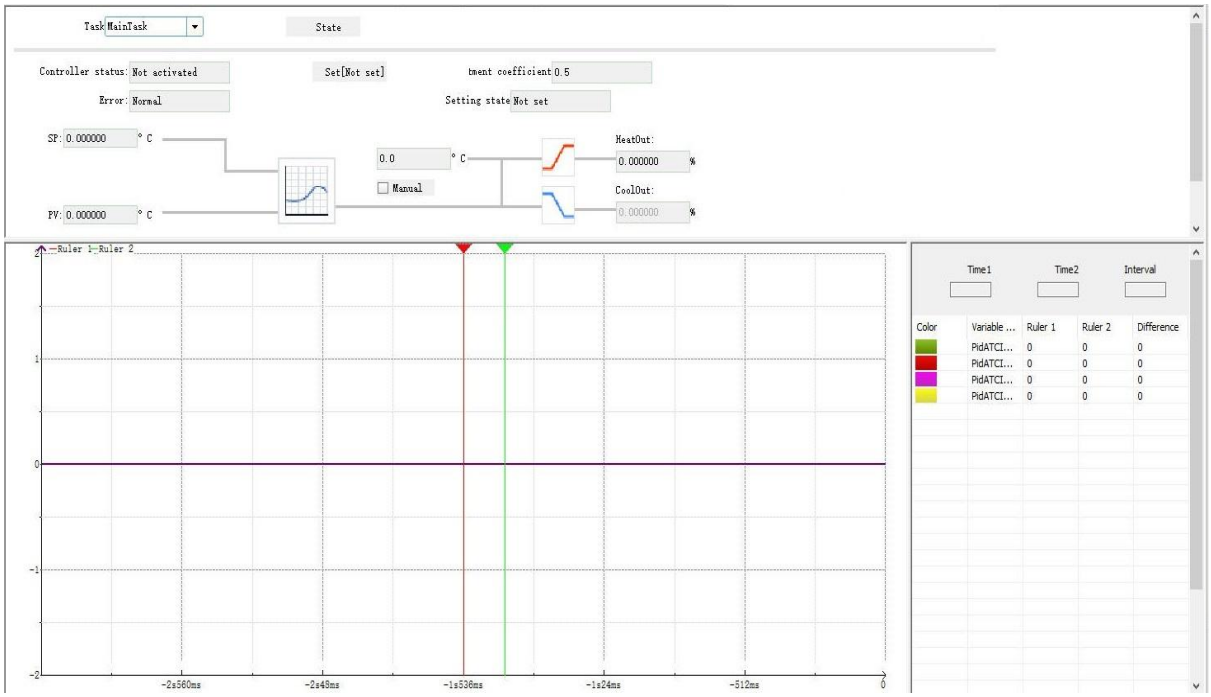
The following table lists variables corresponding to the instructions.

Instruction	Monitoring Variable
PID-AT instruction	Instance name.PIDOutput.SPOut
	Instance name.PIDOutput.PV
	Instance name.PIDOutput.OutPercent
PID-ATC instruction	Instance name.PIDOutput.SPOut
	Instance name.PIDOutput.PV
	Instance name.PIDOutput.HeatPercent
	Instance name.PIDOutput.CoolPercent

NOTE

- Data cannot be downloaded during instruction commissioning.
- Each time "Start" is clicked, the oscilloscope data of the last task is cleared.
- When the oscilloscope is started, if you switch to the offline state or close the commissioning page, the oscilloscope is stopped automatically.

PID_ATC commissioning page



The descriptions of "Task" and "Start" are the same as those of the PID-AT instruction. Specific descriptions of the PID-ATC instruction are listed in the following table.

Name	Description
HeatOUT	Corresponding to the variable HeatPercent, it is read-only. When PIDConfig.GeneralConfig. ControlMode is 0 or 1 (unipolarity control), HeatOUT is normal, while CoolOUT is grayed.
CoolOUT	Corresponding to the variable CoolPercent, it is read-only.
Manual checkbox	• In the manual mode, double-click the OUT box to write the preparation value and operate the variable PIDConfig.GeneralConfig.ManualValue. • In the auto mode, the preparation value cannot be written in the OUT box.

NOTE

- When PIDConfig.GeneralConfig. ControlMode is 0 or 1 (unipolarity control), HeatOUT is normal, while CoolOUT is grayed.
- When PIDConfig.GeneralConfig. ControlMode is 2 or 3 (dual-polarity control), HeatOUT and CoolOUT are normal.

In the online state, not all parameters can be written with values. It depends on the configuration and gray logic.

The following table lists link logic of configurations.

Input/Output Configuration	Configuration Item	Control Link Logic
Input configuration	PV selection input	• The input value cannot be edited. The AI input value can be edited. • Control: Input range upper limit

Input/Output Configuration	Configuration Item	Control Link Logic
		• Control: Input range lower limit
	Setting selection value	• The external setpoint cannot be edited. • The internal setpoint cannot be edited. • Control: Internal setpoint
Output configuration	Output selection	• PWM output (PWM output period and minimum pulse output period are enabled while other options are disabled) • Analog output (analog output upper limit, analog output lower limit, output upper limit, and output lower limit value are enabled, while other options are disabled) • Percentage output (PWM and analog output controls are not editable)

24 Project Management

24.1 Project Comparison

The project comparison function is used to compare differences between various documents of two projects. Structure, variable table, program block, function block (FB), function block (FB) instance table, function (FC), and execution block support detailed content comparison. Other project tree objects only support object difference comparison. The differences detected by detailed content comparison can be manually synchronized to the currently opened project by line synchronization or block synchronization.

Project comparison supports offline comparison and online comparison.

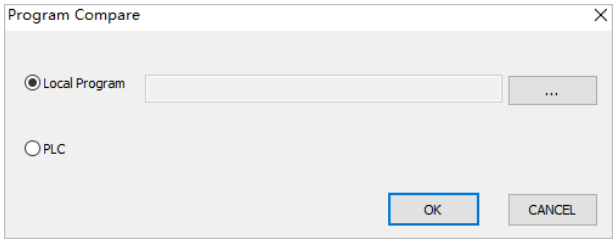
- Offline comparison: Compare the currently opened project with the project stored locally.
- Online comparison: Compare the currently opened project with the internal program of the online PLC.

NOTE

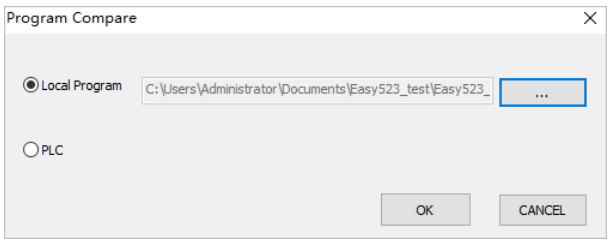
- You can open project files of .hcp, .hcpp, and .updown types. During offline comparison, project file types are not limited; for example, a .hcp file can be compared with a .hcpp type file.
- Only project files of the same model can be compared. For example, the project files of the Easy521 model can only be compared with those of the Easy521 model, but not with those of the Easy522 model.
- During project comparison, it is not allowed to edit the currently opened project. Only after exiting the project comparison mode can the project be edited.
- During project comparison, you cannot open the project files that are being compared in another AutoShop software.

Procedure

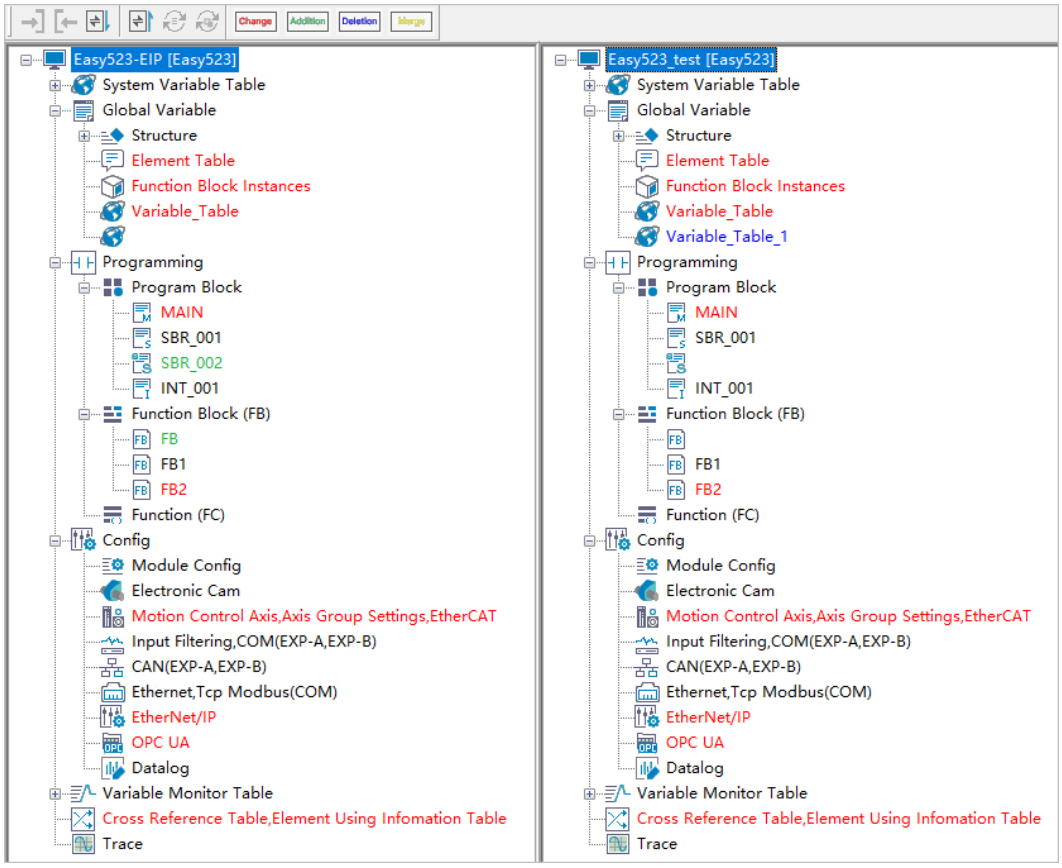
1. Run the AutoShop programming software and open a local project.
2. In the menu bar, choose "File" > "Project Compare". The "Project Compare" dialog box is displayed.



3. Select the offline comparison (local project) or online comparison (PLC) mode as needed. The following takes offline comparison as an example. Click . In the dialog box displayed, select project files (.hcp, .hcpp, or .updown) for comparison, and click "OK".



The project tree comparison page is displayed by default.



The left side represents the current project, and the right side represents the comparison project. The function or meaning of the toolbar icons on the project comparison page shown in the following table.

Icon	Meaning
	Detailed comparison. Click this icon to enter the detailed comparison page.
	Project tree comparison. Click this icon to enter the project tree comparison page.
	Next difference. Click this icon to jump to the next difference.
	Previous difference. Click this icon to jump to the previous difference.
	Line synchronization: Select a difference on the detailed comparison page, and click this icon to synchronize the line of the comparison project on the right to the current project on the left. Click the icon again to cancel the line synchronization operation and return to the state before line synchronization. Note: This icon is only available on the detailed comparison page.
	Block synchronization: Select a difference on the detailed comparison page, and click this icon to synchronize the neighboring differences before and after the current line in the comparison project on the right (that is, the block synchronization content is covered with the non-difference parts before/after the current line) to the current project on the left. Click the icon again to cancel the block synchronization operation and return to the state before block synchronization.

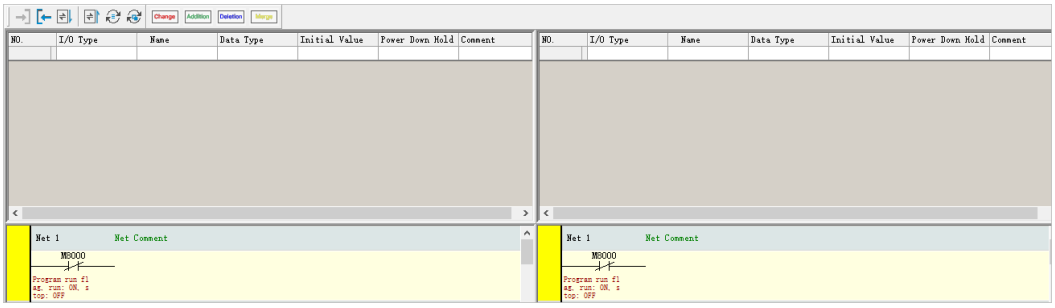
Icon	Meaning
	Note: This icon is only available on the detailed comparison page.
	Different colors, fonts, and meanings of project comparison differences are identified, and the specific meanings are shown in the table below.

Differences in project comparison are distinguished by fonts of different colors, and the meanings of fonts of different colors are shown in the following table.

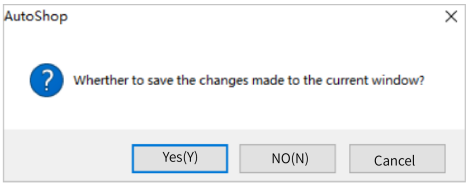
Font Color	Meaning
Black font	The contents on both sides are consistent.
Green font	This content only exists in the current project on the left, and the corresponding cell is left empty in the comparison project on the right.
Blue font	This content only exists in the comparison project on the right, and the corresponding cell is left empty in the current project on the left.
Red font	This part of the content exists in both left and right projects, but the content is different.
Yellow block	This part of the content is synchronous content. Note: The yellow block is only available on the detailed comparison page.

4. Select a project tree object in red that supports detailed content comparison. In the following example, function block (FB) is used. Double-click the function block (FB) "FB2" or click the function block (FB) "FB2" and then click to enter the detailed comparison page.

5. Select the difference and click for line synchronization or click for block synchronization. The following takes line synchronization as an example.
This difference is identified by a yellow block after synchronization.



6. Click . A dialog box is displayed, asking you whether to save the synchronization content.

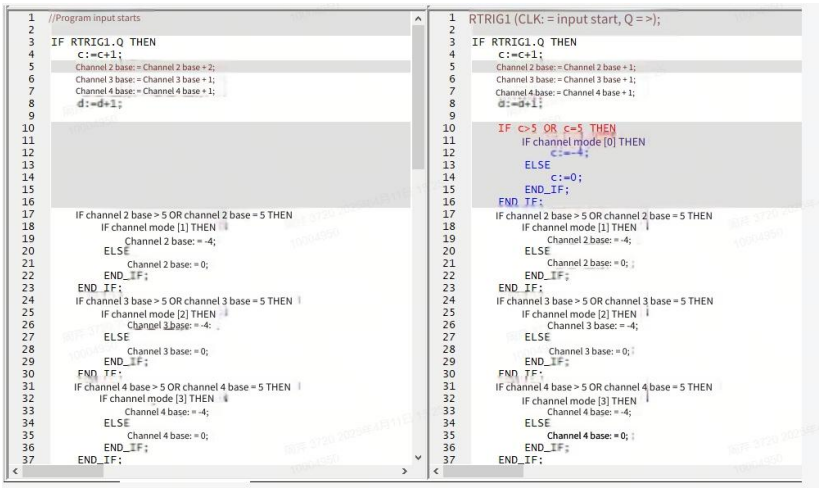


7. Click "Yes" or "No".

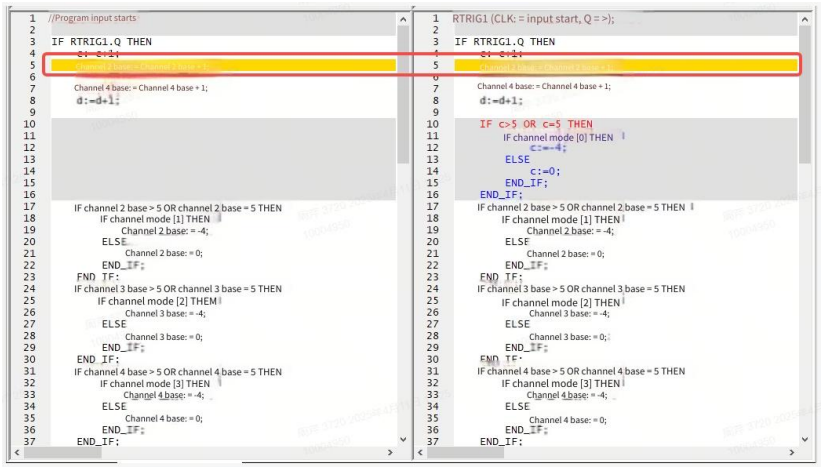
Description

- LiteST program
  Detailed comparison between LiteST programs will compare all the character contents of the programs in detail. The gray shading part in the program text is the difference between the two LiteST programs, and the different contents are distinguished by different colors.

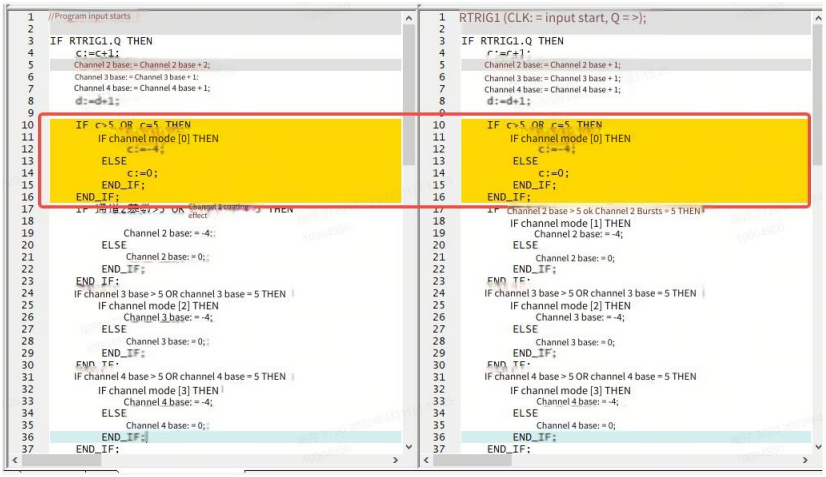
  For LiteST programs, line synchronization is to merge the contents of the line where the text cursor of the right comparison project is located into the current project on the left, while block synchronization is to synchronize the contents of the current line of the cursor and the consecutive different lines before and after. After the merger, the synchronized parts will be marked with yellow blocks (shading). A detailed comparison example of LiteST programs is shown in the following figure.



The following figure shows line synchronization result of LiteST programs.

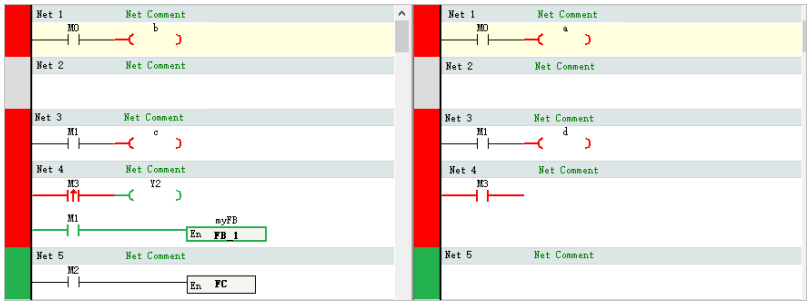


The following figure shows block synchronization result of LiteST programs.



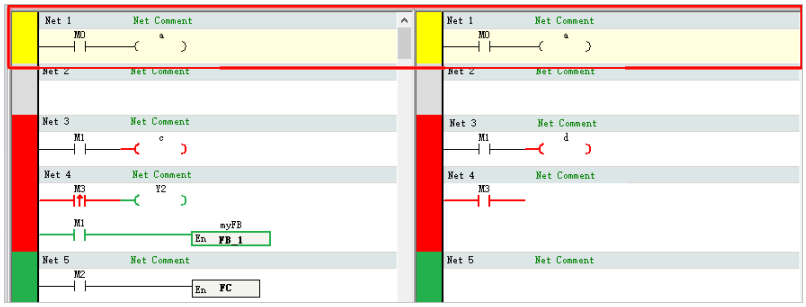
- LD program
  - Comparison between LD programs.

    Detailed comparison is performed on a network basis. If the ladder diagram network contents corresponding to the two projects are different, different colors will be marked on the left side of the corresponding network, and the differences will be identified to the elements in the line, and the contacts, function blocks (FB), and functions (FC) with specific differences will be marked with corresponding colors, as shown in the following figure.



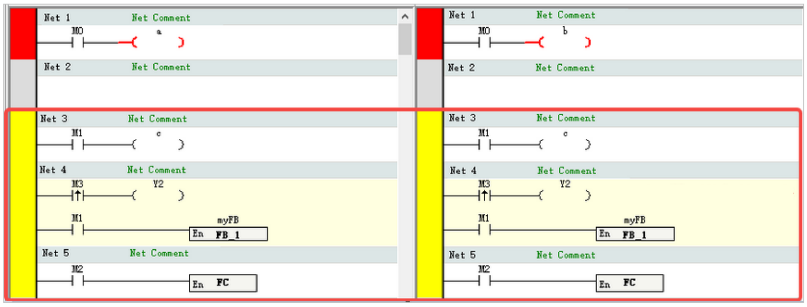
  - Line synchronization for LD programs

    Different from the synchronization function of LiteST programs, line synchronization of LD programs is to synchronize the contents of the currently selected network with differences, as shown in the following figure.



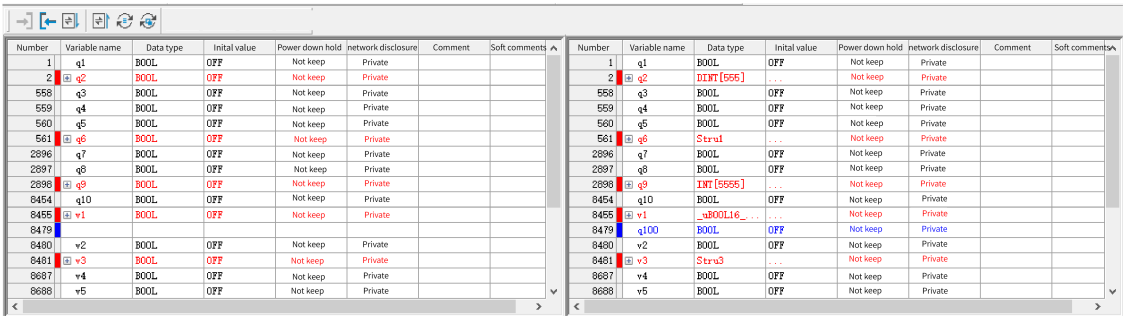
  - Block synchronization for LD programs

    Block synchronization is to synchronize the networks with continuous differences before and after the currently selected network with differences. After the synchronization is completed, the corresponding contents of the comparison project will be merged into the current project, and the left side of the network synchronization content will be marked with a yellow block, as shown in the following figure.



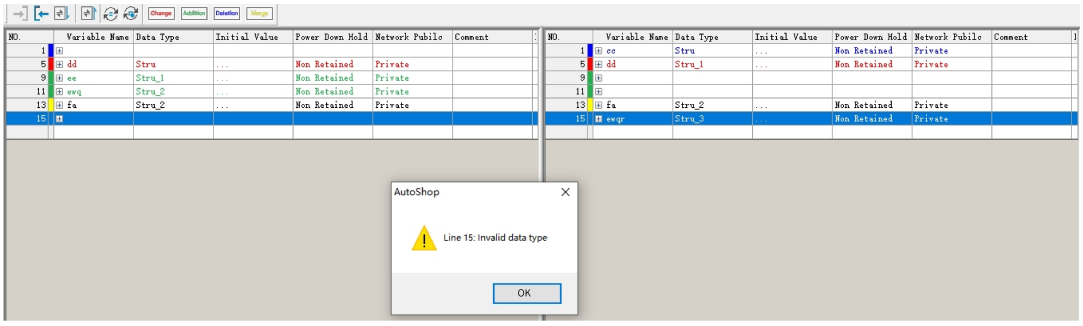
- Variable table

  Properties of detailed comparison of variables in the variable table include: data type, initial value, power failure retention, network public, comments, and soft element address. The differences of detailed comparison will be displayed in specific lines, as shown in the following figure.

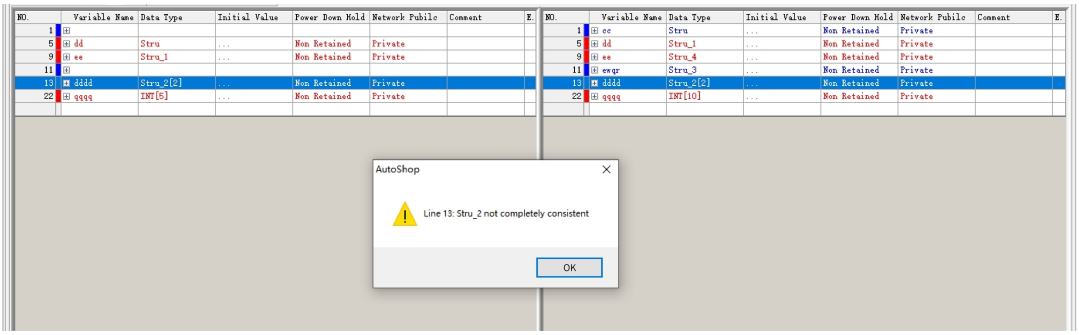


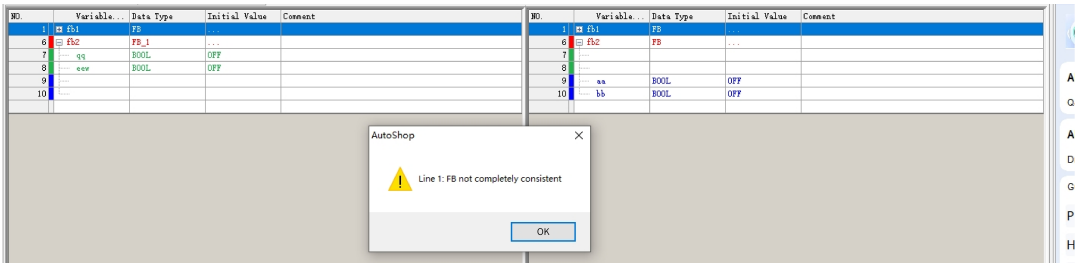
  When synchronizing variables, you can directly synchronize variables with basic data types such as BOOL, INT, and REAL. For variables with internal members, such as arrays and structures, when the parent variable is synchronized, the content of its member variables will be changed at the same time, and only the initial value and comments of the member variables are synchronized.

  - When synchronizing the structure variable or function block instance object of the comparison project into the current project, if the current project does not contain the structure or function block (FB), a prompt box with invalid data type will pop up, as shown in the following figure.

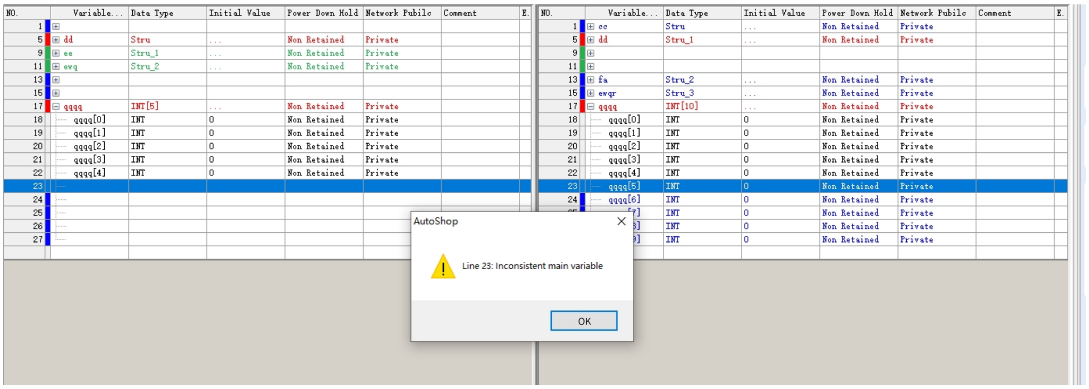


  - For synchronization of structures and function block instances, if there are differences in the contents of the structure and function block, you need to switch to the detailed comparison page of the structure and function block to synchronize the contents, and then synchronize the structure and function block instances in the variable table, as shown in the following figure.

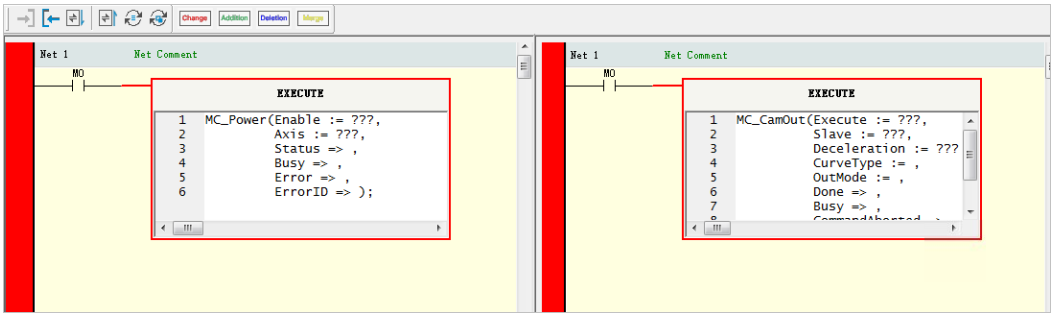




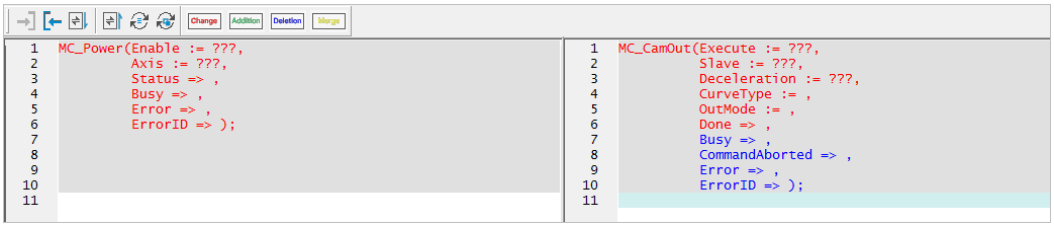
- For synchronization of member variables such as arrays and structures, it is necessary to synchronize the parent variables first. Only when the parent variables are completely consistent can member variables be synchronized; otherwise member variables cannot be synchronized, as shown in the following figure



- Execution block
  When an execution block is inserted into an LD program, the contents of the execution block can be viewed, compared, or synchronized when comparing LD programs.
  - View: When LD programs are compared, if the programs contain execution blocks, the following typos can be executed.
    - When both programs have execution blocks at the same location, you can view the execution blocks by double-clicking the execution blocks.
    - When the execution block only exists in the left program or the right program, you cannot double-click to view it.
    - When there are differences between the compared execution blocks (for example, it is shown in red in the figure), double-click to open the execution block and compare the details.



After double-clicking, compare the ListST text, as shown in the figure below.



- Exit: Click to exit the current execution block comparison view and return to the LD program comparison view. Click the icon again to exit the current LD program comparison view and return to the project tree comparison view.
- Synchronization: If synchronization is performed in the execution block, the ListST text is synchronized. If synchronization is performed in the LD comparison view, the entire program is synchronized.

24.2 Project Encryption

Introduction

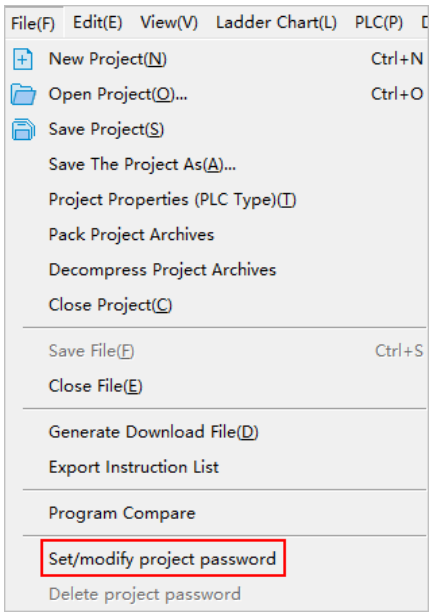
When a password is set to encrypt a project, you need to input this password to open the project.

Preconditions

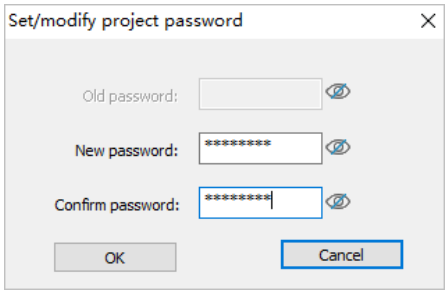
- Software version: AutoShop V4.10.2.0 and later
- The project has been opened.

Procedure

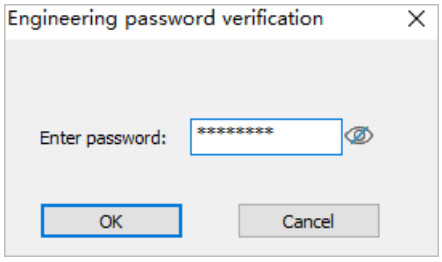
1. In the menu bar, click "Files", and then click "Set/modify project password" to set a project password, as shown in the following figure.



2. In the "Set/modify project password" dialog box displayed, input a password, click "OK" to encrypt the current project file, as shown in the following figure.



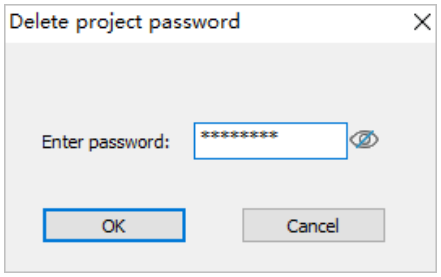
3. When you open the encrypted project file again, the "Engineering password verification" dialog box is displayed, asking you to input the project password, as shown in the following figure.



CAUTION

After a project is encrypted, you need to input the project password when opening the HCP, HCPP, Updown, and HCLIB files. If you upload an encrypted project that is downloaded to the PLC, you also need to input the project password.

To delete the project password, choose "Files" > "Delete project password" in the menu bar. In the "Delete project password" dialog box displayed, click "OK" to clear the current project password, as shown in the following figure.



NOTE

If no password is set for the current project, the "Delete project password" option is grayed.

24.3 Batch Download

Introduction

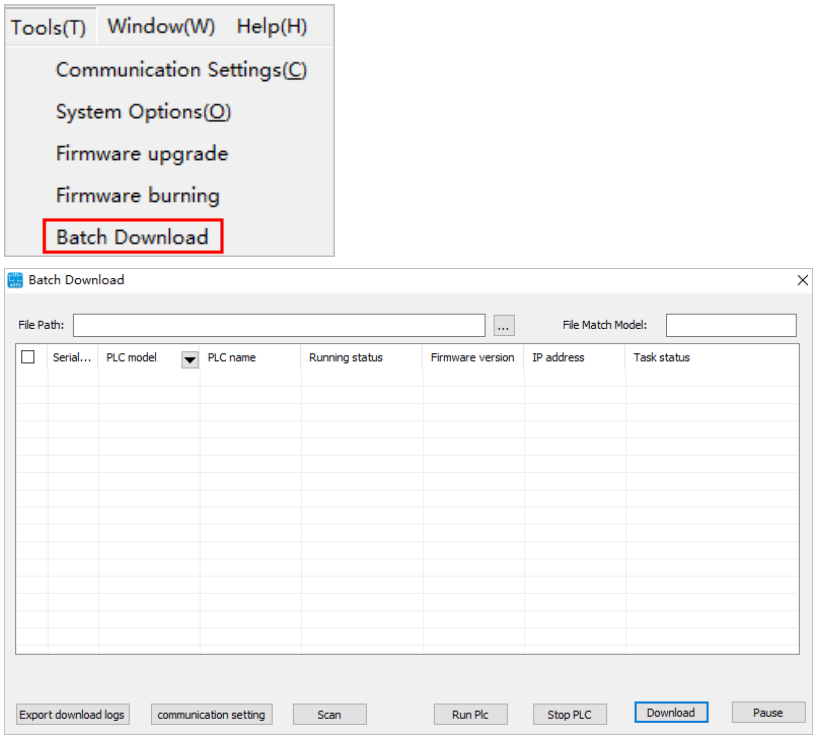
The batch download function allows you to download a program to multiple devices of the same model and update the program on all the specified devices.

Preconditions

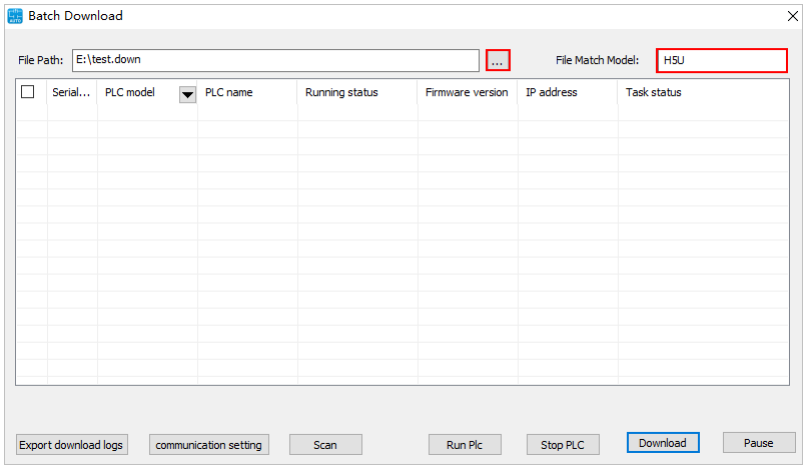
- Software version: AutoShop V4.10.2.0 and later
- "Ethernet" is selected for "Communication Settings".

Procedure

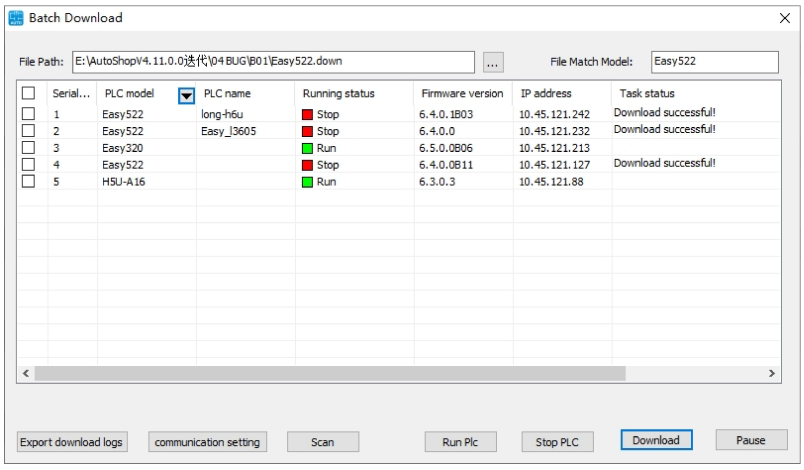
1. In the menu bar, choose "Tools (T)" > "Batch Download". The "Batch Download" dialog box is displayed, as shown in the following figure.



2. Click "..." and select the file saving path. Down and Updown files can be downloaded. After a file is selected, the model matching the file to be downloaded will be displayed on the right.



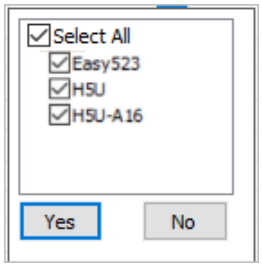
3. Click "Scan" to search for all PLCs connected to the network. The model, name, running status, firmware version, and IP address of the detected PLCs are displayed.



CAUTION

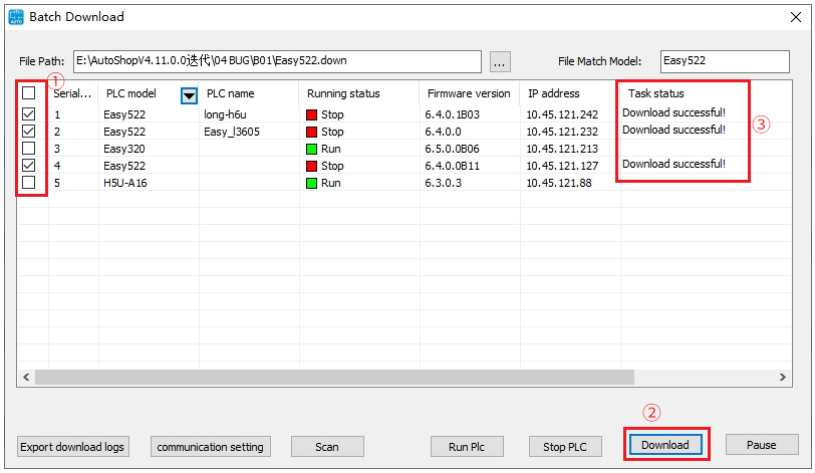
- When the running status of a detected device is "Unable to get the running status! ",check whether the AutoShop is normally connected to the PLC network.
- If the running status of a detected cross-segment device is "Unable to get the running status! ",the running status of the device can be obtained when a TCP connection is created.

4. Click the filter button in the PLC model column to display devices of the specified model and hide other devices to filter the device information.

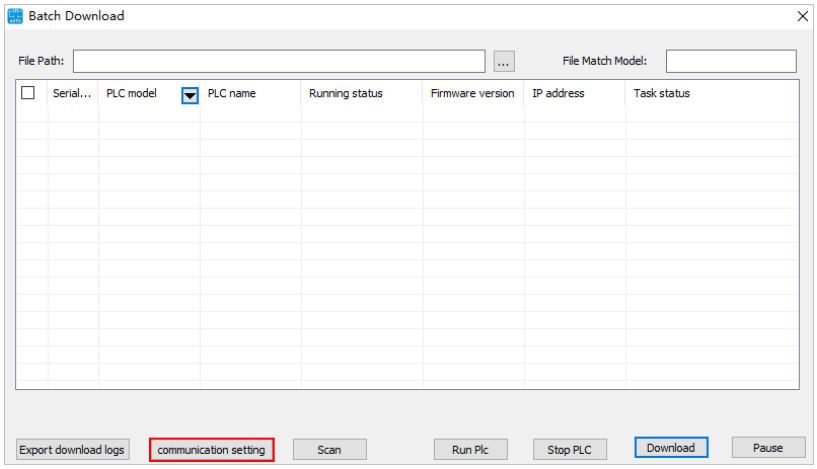


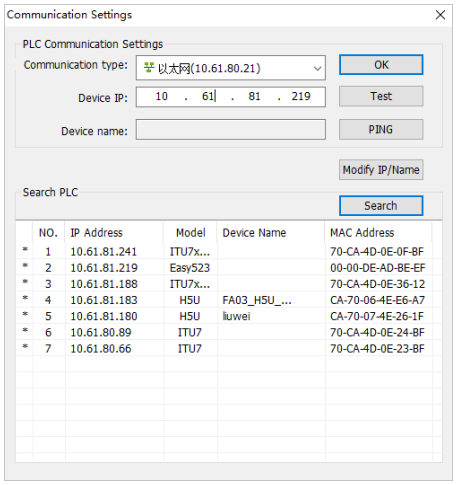
5. Check the boxes in the first columns of devices, and then these devices are selected. Click "Download" and these devices are downloaded in batch.
   During download, the download success or error details will be displayed in the task status bar and the download progress will be displayed at the window bottom. Up to 5 devices can be downloaded in batch at a time. When more than 5 devices are selected, devices after the first five ones are downloaded in sequence after the first five are downloaded.

   a. If you click "Stop PLC" during download, to ensure completeness, the system will continue to download devices that are being downloaded, but devices in the queue will no longer be downloaded.

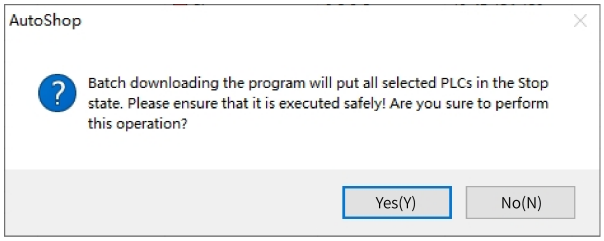


   b. ① Before scanning, downloading, or other operations, click "Communication Settings" to access the network configuration window and set the network to Ethernet in this window, as shown in the following figure.



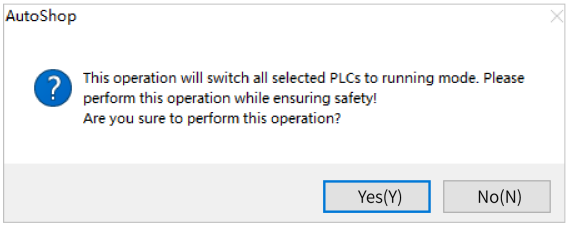


c. ② Before the PLC downloads the program, a window pops up, asking you to set all the selected devices to the Stop state. Perform this operation only when the device and personal safety are ensured. The download is started after you select "Yes (Y)", as shown in the following figure.



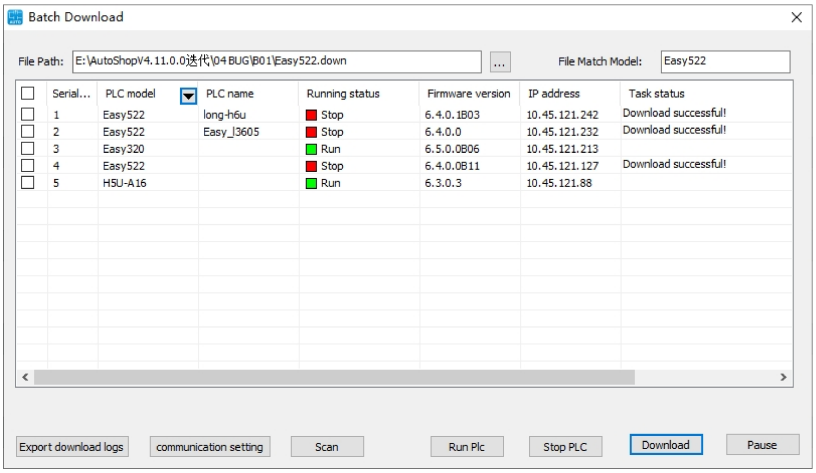
d. ③ The program can only be downloaded to devices in the same network segment of the PLC.

i. After selecting devices from the list, click "Run PLC" or "Stop PLC" to make the selected devices enter the Stop or Run state. Similarly, perform the batch download operation only when the device and personal safety are ensured.



ii. The batch download record is saved at the background. Click "Export Download Log" to export the download record and save it as a CSV file for future reference. The CSV file records the download

time, IP address, name, and download result of each device.



25 Fault Diagnosis

25.1 Fault Diagnosis Through Panel

25.1.1 Indicators

States and meanings of the panel indicators of the H5U series are shown in the following table.

Indicator	Meaning
RUN	Current system status (running or stopped) ON: Running; OFF: Stopped
ERR	System fault
BAT	Battery alarm
BF	EtherCAT bus fault
CRUN	CAN running
CERR	CAN error

States and meanings of the panel indicators of the Easy series are shown in the following table.

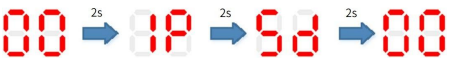
Port Type	Interface Mark	Definition	Indicator Color	Description
I/O indicator	IN/OUT	I/O status	Yellow-green	• Steady ON: The input or output is valid. • OFF: The input or output is invalid.
Running state indicator	PWR	Power supply	Yellow-green	• Steady ON: Power supply normal • OFF: Power supply abnormal
	RUN	Normal running	Yellow-green	• Steady ON: User program running • OFF: User program stopped
	ERR	Running error	Red	• OFF: No major error • Blinking: Major error
	ETH1	EtherNET1 Link	Yellow-green	• Steady ON: Connected

Port Type	Interface Mark	Definition	Indicator Color	Description
				• Blinking: Communication in progress • OFF: Disconnected
	ETH2	Ethernet link 2	Yellow-green	• Steady ON: Connected • Blinking: Communication in progress • OFF: Disconnected

25.1.2 Description of MFK Keys

25.1.2.1 Overview

The MFK key works with the LED display to support multi-function menu operations. Long-press the MFK key, and the LED display will toggle between different function menus at an interval of 2 seconds, as shown in the following figure.



When the function menu you want to enter is displayed on the LED display, release the MFK key and then short-press the MFK key to enter the function menu. Note: Short-press is a brief press for less than 2 seconds.

If you enter a menu that cannot be executed, the LED display shows an error.

Display Code	Name	Description
	E1	The PLC is in a non-secure state (running or downloading) and operation is prohibited.
	E2	No SD card or programming file is detected.
	E3	Multiple programming files are detected in the SD card.
	E4	Programming file data is abnormal or the device model is not compatible.
	E5	Password verification error

25.1.2.2 Restoring Default IP Address

The default IP address of the CPU module is 192.168.1.88. If you modified the address and forget the new address, use the "IP" menu to reset the address to the default.

Select the "IP" menu, and the LED displays a countdown from 10 to 0.

After the countdown, the IP address is reset. During the countdown, you can press the MFK to cancel the reset operation.

25.1.2.3 Downloading User Program in the SD Card

Save the program file compiled in Autoshop into the "PLCProgram" directory of the SD card, and then insert the SD card into the PLC main module. Select the "Sd" menu to download the user program in the SD card into the PLC. The LED displays the progress (00-99) during download and "PP" after download is done.

25.1.2.4 LED Display of the CPU Module

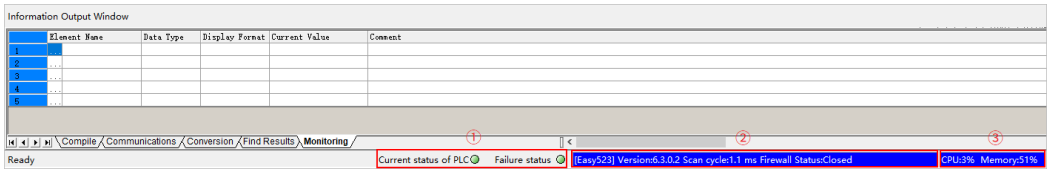
When a system fault occurs, the LED display of the CPU shows the fault code, with "Er" and the code of the fault displayed alternately. For example, when the fault code is Er1501, the LED display is as follows:

For detailed definitions on fault codes, see List of Error Codes.

25.2 Fault Diagnosis Through Software

25.2.1 Accaccessing Basic PLC Information

1. You can click in the toolbar to enter the monitoring mode. In the monitoring mode, the basic information of the PLC is displayed in the lower right corner of the interface. The basic information includes:

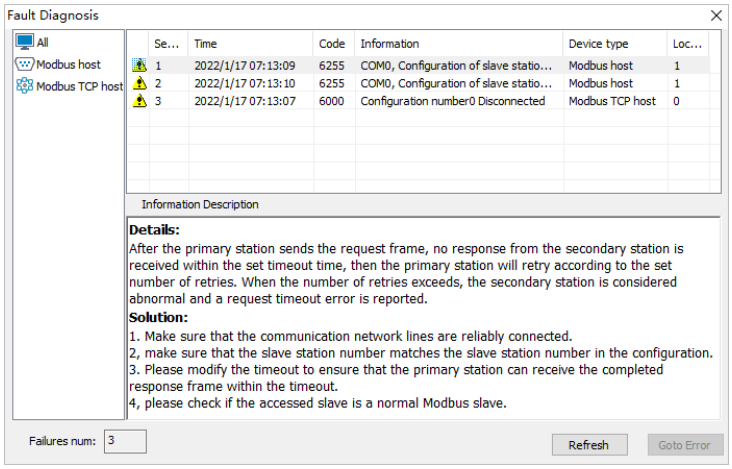


Name	Description
①	PLC status indicator and fault indicator
②	Current firmware version and program scan cycle
③	CPU and memory utilization rates

States and meanings of the panel indicators are shown in the following table.

Current PLC status		Status	
Green	Running	Green	No fault
Red	Stopped	Yellow	Minor error
-	-	Red	Major error

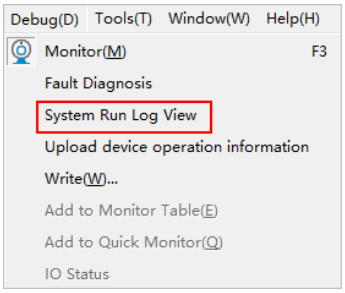
2. Double-click the fault indicator to enter the fault diagnosis page and obtain detailed fault information, as shown in the following figure.



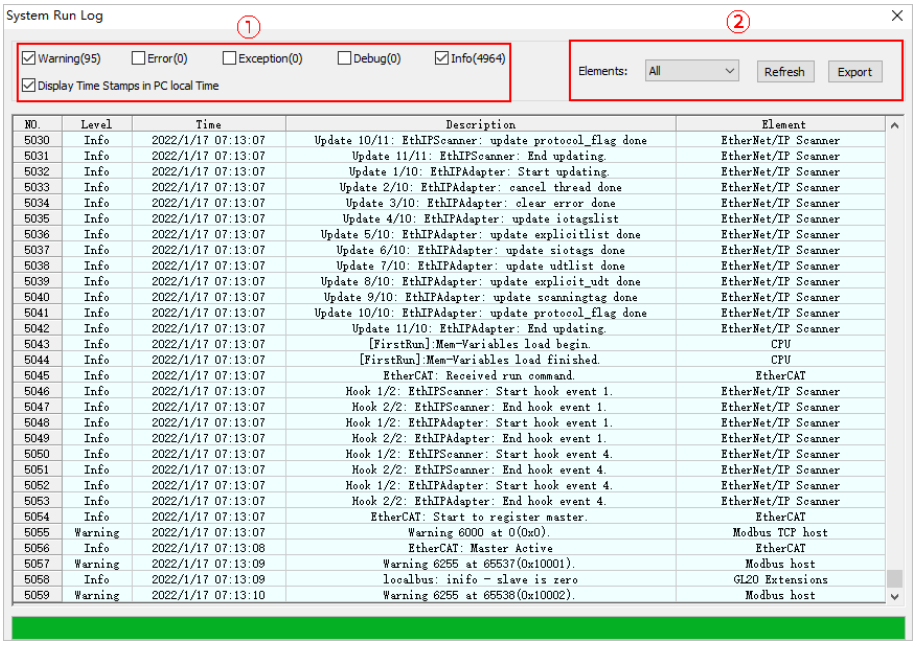
25.2.2 Viewing Operation Logs

Operation logs include fault logs and system logs. To view operation logs, follow these steps:

In the menu bar, click "Debug" and then choose "System Run Log View".



The "System Run Log" page is displayed.



Name	Description
①	Select which categories of information to display.
②	Select which elements to display, refresh the operation logs, or export the operation logs.

25.3 Error Codes

The software tool prompts various categories of error codes when faults occur in user programming. The following table lists the error codes and corresponding solutions.

Error code	Error message	Detailed description	Solution
Program			
1500	User program watchdog timed out	The user program execution time is too long and has exceeded the set program watchdog.	Increase the watchdog time as appropriate, or check whether there is a program block with unexpectedly long execution time in the user program.
1501	Undefined instruction	The instruction is not supported.	Upgrade the PLC firmware to the version that supports the instruction.
1502	The user program is incomplete, and the length is incorrect.	The user program is incomplete, and the length is incorrect.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
1503	Program authorization protection identifier error. Check whether the identifier matches.	The program authorization protection identifier is incorrect. Check whether the authorization protection identifier of the device is set correctly.	Contact the equipment provider.
1504	User program empty	The user program is empty. There is no valid program.	Re-download the user program.
1505	The block POU identifier is incorrect.	The block POU identifier is incorrect.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
1510	The subprogram identifier is incorrect.	The subprogram identifier is incorrect.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
1511	The subprogram type is incorrect.	The subprogram type is incorrect.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.

Error code	Error message	Detailed description	Solution
1512	The subprogram serial number is incorrect or out of range.	The subprogram serial number is incorrect or out of range.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
1513	The subprogram address is incorrect, duplicated, or conflicting.	The subprogram address is incorrect, duplicated, or conflicting.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
1514	The interrupt subprogram serial number is incorrect or out of range.	The interrupt subprogram serial number is incorrect or out of range.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
1515	The interrupt subprogram address is incorrect, duplicated, or conflicting.	The interrupt subprogram address is incorrect, duplicated, or conflicting.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
1516	The interrupt subprogram edge is incorrect (not rising edge or falling edge).	The interrupt subprogram edge is incorrect (not rising edge or falling edge).	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
1517	The interrupt timing duration range of the interrupt subprogram timer is incorrect.	The interrupt timing duration range of the interrupt subprogram timer is incorrect.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
1520	The OBprog program identifier is incorrect.	The OBprog program identifier is incorrect.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
1521	The OBprog program type is incorrect.	The OBprog program type is incorrect.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
1522	The OBprog program serial number is incorrect or out of range.	The OBprog program serial number is incorrect or out of range.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
1523	The OBprog program address is incorrect, duplicated, or conflicting.	The OBprog program address is incorrect, duplicated, or conflicting.	Recompile and download the user program, or recompile and

Error code	Error message	Detailed description	Solution
			download the user program after replacing the software tool.
1524	The variable quantity of the OBprog program is incorrect.	The variable quantity of the OBprog program is incorrect.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
1525	The variable length of the OBprog program is incorrect.	The variable length of the OBprog program is incorrect.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
1526	The header data of the OBprog program is incorrect.	The header data of the OBprog program is incorrect.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
1530	The LBL serial number of the CJ-LBL instruction is incorrect or out of range.	The LBL serial number of the CJ-LBL instruction is incorrect or out of range.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
1531	The LBL address of the CJ-LBL instruction is incorrect, duplicated, or conflicting.	The LBL address of the CJ-LBL instruction is incorrect, duplicated, or conflicting.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
5001	Exception in user program execution or instruction return value error, some instructions not executed	Execution of the user program is abnormal or the return value of the instruction is incorrect, and some instructions are not executed, causing program execution to end abnormally.	Check the logic of the user program for any exception in execution process or execution logic.
5010	The subprogram serial number of the CALL instruction is incorrect or out of range.	The subprogram serial number of the CALL instruction is incorrect or out of range.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
5011	The subprogram of the CALL instruction does not exist or is not initialized.	The subprogram of the CALL instruction does not exist or is not initialized.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
5012	The number of subprogram nesting levels of CALL	The number of subprogram nesting levels of the CALL instruction is out of range.	Modify the program logic to reduce the subprogram nesting levels.

Error code	Error message	Detailed description	Solution
	instruction is out of range or no more than 0.		
5013	The subprogram of the CALL instruction returns a relationship error.	The subprogram of the CALL instruction returns a relationship error.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
5014	Mismatch between subprogram call and subprogram return	Subprogram execution is abnormal. The subprogram call and subprogram return do not match.	Check whether the subprogram call and return are disordered due to the abnormal end of the user program.
5015	Interrupt subprogram undefined	The interrupt subprogram is undefined or does not exist.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
5016	The interrupt queue of the interrupt subprogram timer is full and the interrupt is lost.	The interrupt queue of the interrupt subprogram timer is full and the interrupt is lost.	Modify the interrupt subprogram attributes or logic, and reduce the number of interrupts as appropriate.
5020	The FBFC program serial number is incorrect or out of range.	The FBFC program serial number is incorrect or out of range.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
5021	The FBFC program does not exist or is not initialized.	The FBFC program does not exist or is not initialized.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
5022	The variable of the FBFC program does not exist or is not initialized.	The variable of the FBFC program does not exist or is not initialized.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
5023	The number of FBFC program nesting levels is out of range or no more than 0.	The number of FBFC program nesting levels is out of range.	Modify the program logic to reduce the FBFC program nesting levels.
5024	The FBFC program returns a relationship error.	The FBFC program returns a relationship error.	Check whether the FBFC special instruction is used in the wrong position, or recompile and download the user program.

Error code	Error message	Detailed description	Solution
5025	Mismatch between OBprog program call and program return	OBprog program execution is abnormal. The program call and program return do not match.	Check whether the program call and return are disordered due to the abnormal end of the user program.
5030	The LBL serial number of the CJ-LBL instruction is incorrect or out of range.	The LBL serial number of the CJ-LBL instruction is incorrect or out of range.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
5031	The LBL of the CJ-LBL instruction does not exist or is not initialized.	The LBL of the CJ-LBL instruction does not exist or is not initialized.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
5032	The number of nesting levels of the FOR-NEXT instruction is out of range or no more than 0.	The number of nesting levels of the FOR-NEXT instruction is out of range.	Modify the program logic to reduce the FOR-NEXT instruction nesting levels.
5033	The number of FOR-NEXT instruction loops is out of range or no more than 0.	The number of FOR-NEXT instruction loops is out of range or no more than 0.	Modify the program logic to change the number of FOR-NEXT instruction loops.
5034	The number of FOR- NEXT instruction loops is equal to 0.	The number of FOR- NEXT instruction loops is equal to 0.	Modify the program logic to change the number of FOR-NEXT instruction loops.
5035	The FOR and NEXT instructions are not paired.	The FOR and NEXT instructions are not paired.	Check whether the disorder is caused by abnormal stop of the user program.
5080	Array subscript access out of bounds	The array access subscript is greater than the maximum array subscript value, and the subscript value in use has been changed to the maximum array subscript value.	Double-click the fault code to go to the corresponding program position to modify the subscript value.
5081	The division-by-zero protection is triggered and the divisor 0 is replaced by 1 automatically.	The division-by-zero protection is triggered and the divisor 0 is replaced by 1 automatically.	Double-click the fault code to go to the corresponding program position to modify the divisor.

Error code	Error message	Detailed description	Solution
5082	The program loop has no response for a long time.	The program loop has no response for a long time.	Double-click the fault code to go to the corresponding program position to modify the loop statement.
5083	Array subscript access out of bounds	The array access subscript is less than 0, and the subscript value in use has been changed to 0.	Double-click the fault code to go to the corresponding program position to modify the subscript value.
5084	Invalid data	The floating-point data is invalid.	Check whether the input values of functions such as LN, LOG, SQRT are legal.
5101	The address of the parameter variable of the instruction is incorrect, or the variable does not exist.	The address of the parameter variable of the instruction is incorrect, or the variable does not exist.	Check whether the address of the parameter variable of the instruction is normal and whether the variable exists.
5102	The size of the parameter variable of the instruction is incorrect. The variable does not exist or is out of range.	The size of the parameter variable of the instruction is incorrect. The variable does not exist or is out of range.	Check whether the data length of the parameter variable of the instruction is out of range.
5104	The instruction parameter sequence or relationship is incorrect.	The instruction parameter sequence or relationship is incorrect.	Check whether the parameter sequence or relationship of the instruction is correct.
5105	The character string data or length of the string instruction is incorrect.	The character string data or length of the string instruction is incorrect.	Check whether the character string data of the string instruction is illegal.
5110	The serial number of the Pointer is incorrect or out of range.	The serial number of the Pointer is incorrect or out of range.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
5111	The Pointer is not initialized or does not point to a valid data variable.	The Pointer is not initialized or does not point to a valid data variable.	Check whether the Pointer is initialized and whether it points to a valid variable address.
5112	The variable pointed to by the Pointer does not exist or is out of range.	The variable pointed to by the Pointer does not exist or is out of range.	Check the variable address pointed to by the Pointer or initialize the Pointer again.
5113	The offset of the Pointer is out of range.	The offset of the Pointer is out of range.	Check whether the offset of the Pointer is too large. If yes, reduce the offset.

Error code	Error message	Detailed description	Solution
5114	The variable pointed to by the execution result of the Pointer does not exist or is out of range.	The variable pointed to by the execution result of the Pointer does not exist or is out of range.	Check whether the variable address pointed to by the execution result of the Pointer exists and whether it is out of range.
5120	Failed to instantiate the counter instruction.	Failed to instantiate the counter instruction.	Failed to instantiate the counter instruction. Recompile and download the user program.
5121	Check whether the comparand of the counter instruction is incorrect or out of range.	Check whether the comparand of the counter instruction is incorrect or out of range.	Check whether the comparand of the counter instruction is incorrect or out of range.
5130	Failed to instantiate the timer instruction.	Failed to instantiate the timer instruction.	Failed to instantiate the timer instruction. Recompile and download the user program.
5131	Check whether the comparand of the timer instruction is incorrect or out of range.	Check whether the comparand of the timer instruction is incorrect or out of range.	Check whether the comparand of the timer instruction is incorrect or out of range.
5140	Ensure that the number of SFC STL parallel branch/ parallel recombination/ selective branch/selective recombination lines is within the specified range.	Ensure that the number of SFC STL parallel branch/parallel recombination/selective branch/selective recombination lines is within the specified range.	Ensure that the number of SFC STL parallel branch/parallel recombination/selective branch /selective recombination lines is within the specified range.
5150	Failed to instantiate the function block instruction.	Failed to instantiate the function block instruction.	Failed to instantiate the function block instruction. Recompile and download the user program.
5160	The subscript variable code of the array is incorrect or does not exist.	The subscript variable code of the array is incorrect or does not exist.	Recompile and download the user program.
5161	Array subscript variable data error or out of range	The subscript variable of the array is incorrect or out of range.	Modify the value of the subscript variable so that the array falls within the allowable range.
5170	Failed to instantiate the PID_AT instruction.	Failed to instantiate the PID_AT instruction.	Failed to instantiate the PID_AT instruction. Recompile and download the user program
5171			

Error code	Error message	Detailed description	Solution
	Failed to instantiate the PID_ATC instruction.	Failed to instantiate the PID_ATC instruction.	Failed to instantiate the PID_ATC instruction. Recompile and download the user program
5172	The PID AT/PID ATC instruction input parameters PV/AI are abnormal.	The PID AT/PID ATC instruction input parameters PV/AI are abnormal.	The PID AT/PID ATC instruction input parameters PV/AI are abnormal. Check whether the input parameters PV/AI are absent or the PVType parameter is set correctly.
5600	Failed to instantiate the SerialSR instruction.	Failed to instantiate the SerialSR instruction.	Failed to instantiate the SerialSR instruction. Recompile and download the user program.
5601	The port ID of the SerialSR instruction is out of range.	The port ID of the SerialSR instruction is out of range.	Modify the port ID of the SerialSR instruction.
5602	The protocol of the SerialSR instruction is incorrect.	The protocol of the SerialSR instruction is incorrect.	Set the free protocol for the serial port by using the software tool.
5603	SerialSR instruction port conflict	Multiple instructions call the SerialSR instruction at the same time, and the instruction that fails to preempt the port reports an error.	Modify the instruction scheduling timing to implement time division multiplexing.
5604	The SerialSR instruction TX data length is out of range or smaller than 0.	The SerialSR instruction TX data length is out of range or smaller than 0.	The SerialSR instruction TX data length is out of range or smaller than 0.
5605	SerialSR instruction TX data buffer error	Failed to obtain the TX data buffer of the SerialSR instruction.	Enable this instruction again.
5606	The SerialSR instruction RX data length is out of range or smaller than 0.	The SerialSR instruction RX data length is out of range or smaller than 0.	The SerialSR instruction RX data length is out of range or smaller than 0.
5607	SerialSR instruction RX data buffer error	Failed to obtain the RX data buffer of the SerialSR instruction.	Enable this instruction again.
6580	The axis ID specified in the CANopen axis instruction is invalid.	The axis ID specified in the CANopen axis instruction is invalid.	The axis ID specified in the CANopen axis instruction is invalid. Modify the axis ID.

Error code	Error message	Detailed description	Solution
6701	Invalid memory address: element or variable non-existent	The memory address is invalid. The element or variable to access does not exist.	Modify the instruction parameter to use a valid element or variable.
6705	Invalid memory size: memory non-existent or out of range	The memory size is invalid. The number of elements or variables to access is too large or out of range.	Modify the instruction parameter to adjust the number of elements or variables.
6706	Improper data or data out of range	The instruction parameter is improper or out of the allowable range.	Refer to the instructions guide to modify the instruction parameter value.
6711	Invalid variable address: variable non-existent	The variable address is invalid. The element or variable to access does not exist.	Modify the instruction parameter to use a valid element or variable.
6712	Invalid variable size: variable out of range	The variable size is invalid. The number of elements or variables to access is too large or out of range.	Modify the instruction parameter to adjust the number of elements or variables.
6713	The variable encoding is invalid.	The variable encoding is invalid.	Recompile and download the user program, or recompile and download the user program after replacing the software tool.
CPU			
1011	FPGA initialization failed	FPGA initialization failed.	The device hardware is faulty. Replace the device and return the faulty device to the factory for repair.
1012	Interrupt initialization failed	Interrupt initialization failed.	The device hardware is faulty. Replace the device and return the faulty device to the factory for repair.
1013	Timer interrupt initialization failed	Failed to initialize the timer interrupt of the user program.	Restart the device and try again, or replace the device and return the faulty device to the factory for repair.
5200	Error in data retention upon power failure	Check whether the function of data retention upon power failure works properly.	Check whether the function of data retention upon power failure works properly.

Error code	Error message	Detailed description	Solution
5238	2038 problem imminence warning	The device will not work normally after 11:14:07 on January 19, 2038 (UTC+8).	Change the device time.
5250	Low RTC battery voltage	The battery voltage of the RTC clock is low. If the device is powered off at this time, the system time will be restored to the initial value.	Replace the battery of the RTC clock while keeping the device powered on.
5900	Network down: Ethernet IP address conflict	When the device connects to the network or starts running after stop, or when its IP address is modified, it detects whether its IP address is used by other devices in the current network. If yes, the device automatically shuts down the network to avoid conflict.	Change the device IP address.
Po	Device power-off	The device stops after two or three seconds.	No
1111	The system is running but the application program is not.	The system is running but the application program is not.	If the error message lasts for over one minute, contact the equipment provider.
Local I/O			
5300	Initialization failed	Initialization failed.	The device hardware is faulty. Replace the device and return the faulty device to the factory for repair.
5301	Invalid DI filter parameter configuration	The DI filter parameter configuration data is invalid.	Modify the DI filter parameter configuration data.
GL10 expansion module			
5400	Failed to initialize expansion module interface hardware	The hardware of the expansion module interface is faulty, which causes the initialization to fail.	The device hardware is faulty. Replace the device and return the faulty device to the factory for repair.
5401	Failed to parse expansion module configuration data	The configuration data of the expansion module cannot be parsed correctly because its	Clear the compilation information using AutoShop and recompile and download the program. If the problem persists,

Error code	Error message	Detailed description	Solution
		format does not meet requirements.	delete the module configurations and add modules and configurations again one by one.
5402	Failed to initialize expansion module interface slot	The slot of the expansion module interface is faulty, which causes the initialization to fail.	1. Check whether the expansion module interface slot is short-circuited. If yes, eliminate the short circuit. 2. Check whether the installed module hardware works properly. If not, replace the module.
5403	Expansion module not installed	The expansion module is configured but not installed.	Install the expansion module as required, or modify the configuration of the expansion module.
5404	Module installed inconsistent with module configured	The module installed in the slot must be inconsistent with the configured module; otherwise, it cannot work properly.	Install the expansion module as required and modify module configuration accordingly to ensure consistency.
5405	Expansion module interface hardware exception	The expansion module interface is abnormal.	1. Check whether the expansion module interface slot is short-circuited. If yes, eliminate the short circuit. 2. Check whether the installed module hardware works properly. If not, replace the module.
5406	Expansion module interface software exception	The expansion module interface software is abnormal.	1. Upgrade the PLC firmware. 2. If the problem persists after the firmware upgrade, replace the device and return the faulty device to the factory for repair.
5411	Module in the slot not powered	The module requires external power supply to function properly, but the external power supply is not on.	Connect the external power supply correctly according to the module specifications.
5412	Slot module hardware fault	The module has an internal fault and cannot work properly.	Replace the module and return the faulty module to the factory for repair.
5413	Slot module over-temperature	The module detected a high internal temperature that may lead to malfunction.	1. Do not install the module in an environment that does not meet the relevant temperature

Error code	Error message	Detailed description	Solution
			requirements. 2. Replace the module and return the faulty module to the factory for repair.
5419	Slot module channel input or output overflow	For the input channel, the input signal has exceeded the upper sampling threshold. Sampling cannot be performed properly, and there is a possibility that the input port may be burned. For the output channel, the output value of the corresponding channel has exceeded the set upper threshold, and signals cannot be output properly.	Input channel: Check the actual input signal value. If the signal input to this channel has exceeded the set sampling range under normal working conditions, modify the sampling range as appropriate. If the signal is abnormal, check the output device or instrument of the signal. Output channel: Check the set output value and ensure that the set output is within the set range. If the set range cannot meet requirements, modify it as appropriate.
5420	Slot module channel input or output underflow	For the input channel, the input signal has fallen below the lower sampling threshold, and sampling cannot be performed properly. For the output channel, the output value of the corresponding channel has fallen below the set lower threshold, and signals cannot be output properly.	Input channel: Check the actual input signal value. If the signal input to this channel has exceeded the set sampling range under normal working conditions, modify the sampling range as appropriate. If the signal is abnormal, check the output device or instrument of the signal. Output channel: Check the set output value and ensure that the set output is within the set range. If the set range cannot meet requirements, modify it as appropriate.
5421	Slot module channel input upper limit exceeded or current output disconnected	For the input channel, the input signal has exceeded the upper sampling threshold. At this time, the signal can be sampled normally but the accuracy cannot be guaranteed. For the current output channel, the output port is not connected to the load or	Input channel: Check the actual input signal value. If the signal input to this channel has exceeded the set sampling range under normal working conditions, modify the sampling range as appropriate. If the signal is abnormal, check the output device or instrument of

Error code	Error message	Detailed description	Solution
		the impedance of the connected load is too large, so that the current cannot be output normally.	the signal. Current output channel: Ensure that the load of the output port is connected properly and reliably, and that the load impedance is within the range specified in the module specifications.
5422	Slot module channel input lower limit exceeded or voltage output short-circuited	For the input channel, the input signal has fallen below the lower sampling threshold. At this time, the signal can be sampled normally but the accuracy cannot be guaranteed. For the voltage output channel, the output port is possibly short-circuited or the impedance of the connected load is too small, so that the voltage cannot be output normally.	Input channel: Check the actual input signal value. If the signal input to this channel has exceeded the set sampling range under normal working conditions, modify the sampling range as appropriate. If the signal is abnormal, check the output device or instrument of the signal. Voltage output channel: Ensure that the load of the output port is connected properly and reliably, and that the load impedance is within the range specified in the module specifications.
5423	Slot module channel input disconnected or output hardware faulty	For the input channel, no input signal is connected to the input port or the input signal is too weak and cannot be detected or sampled. For the output channel, the channel hardware is faulty and may have burned out.	Input channel: Ensure that the signal of the input port is normal and valid and is connected properly and reliably. Output channel: Replace the module and return the faulty module to the factory for repair.
GL20 expansion module			
5800	Failed to initialize expansion module interface hardware	The hardware of the expansion module interface is faulty, which causes the initialization to fail.	The device hardware is faulty. Replace the device and return the faulty device to the factory for repair.
5801	Failed to parse expansion module configuration data	The configuration data of the expansion module cannot be parsed correctly because its format does not meet requirements.	Clear the compilation information using AutoShop and recompile and download the program. If the problem persists, delete the module configurations

Error code	Error message	Detailed description	Solution
			and add modules and configurations again.
5802	Expansion module status setting exception	The slot of the expansion module interface is faulty, which makes it unable to switch to the running state.	Check whether the installed module hardware works properly. If not, replace the module.
5803	Expansion module not installed	The expansion module is configured but not installed.	Install the expansion module as required, or modify the configuration of the expansion module.
5804	Module installed inconsistent with module configured	The module installed in the slot must be inconsistent with the configured module; otherwise, it cannot work properly.	Install the expansion module as required and modify module configuration accordingly to ensure consistency.
5805	Expansion module disconnection	The expansion module interface is abnormal.	Check whether the installed module hardware works properly. If not, replace the module.
5806	Expansion module interface software exception	The expansion module interface software is abnormal.	1. Upgrade the PLC firmware. 2. If the problem persists after the firmware upgrade, replace the device and return the faulty device to the factory for repair.
5807	Number of expansion module interfaces faulty	The number of expansion module interfaces is faulty.	The number of mounted modules is greater than the number of expansion modules supported by the host.
5820	Power supply overvoltage of expansion module	The module requires external power supply to function properly, but the external power supply voltage exceeds the limit specified.	Connect the external power supply correctly according to the module specifications.
5821	Power supply undervoltage of expansion module	The module requires external power supply to function properly, but the external power supply voltage is excessively low.	Connect the external power supply correctly according to the module specifications.
5822	Expansion module not powered up	The module requires external power supply to function	

Error code	Error message	Detailed description	Solution
		properly, but the external power supply is not on.	Connect the external power supply correctly according to the module specifications.
5830	Expansion module overtemperature	The module detected a high internal temperature that may lead to malfunction.	1. Do not install the module in an environment that does not meet relevant temperature requirements. 2. Replace the module and return the faulty module to the factory for repair.
5840	Module hardware fault	The module has an internal fault and cannot work properly.	Replace the module and return the faulty module to the factory for repair.
5841	Module reference channel failure	The TC module has a cold end sampling fault and cannot work properly.	Replace the module and return the faulty module to the factory for repair.
5850	Module channel input disconnected or current output disconnected	For the input channel, no input signal is connected to the input port or the input signal is too weak and cannot be detected or sampled. For the current output channel, the output port is not connected to the load or the impedance of the connected load is too large, so that the current cannot be output normally.	Input channel: Ensure that the signal of the input port is normal and valid and is connected properly and reliably. Current output channel: Ensure that the load of the output port is connected properly and reliably, and that the load impedance is within the range specified in the module specifications.
5851	Module channel voltage output short circuit	For the voltage output channel, the output port is possibly short-circuited or the impedance of the connected load is too small, so that the voltage cannot be output normally.	Voltage output channel: Ensure that the load of the output port is connected properly and reliably, and that the load impedance is within the range specified in the module specifications.
5852	Module channel input over the upper limit	For the input channel, the input signal has exceeded the upper sampling threshold. At this time, the signal can be sampled normally but the accuracy cannot be guaranteed.	Input channel: Check the actual input signal value. If the signal input to this channel has exceeded the set sampling range under normal working conditions, modify the sampling range as appropriate. If the signal is abnormal, check the

Error code	Error message	Detailed description	Solution
			output device or instrument of the signal.
5853	Module channel input data exceeds lower limit	For the input channel, the input signal has fallen below the lower sampling threshold. At this time, the signal can be sampled normally but the accuracy cannot be guaranteed.	Input channel: Check the actual input signal value. If the signal input to this channel has exceeded the set sampling range under normal working conditions, modify the sampling range as appropriate. If the signal is abnormal, check the output device or instrument of the signal.
5854	Module channel input overflow	For the input channel, the input signal has exceeded the upper sampling threshold. Sampling cannot be performed properly, and there is a possibility that the input port may be burned.	Input channel: Check the actual input signal value. If the signal input to this channel has exceeded the set sampling range under normal working conditions, modify the sampling range as appropriate. If the signal is abnormal, check the output device or instrument of the signal.
5855	Module channel input underflow	For the input channel, the input signal has fallen below the lower sampling threshold, and sampling cannot be performed properly.	Input channel: Check the actual input signal value. If the signal input to this channel has exceeded the set sampling range under normal working conditions, modify the sampling range as appropriate. If the signal is abnormal, check the output device or instrument of the signal.
35010	The baud rate is incorrectly configured for channels of the 2S485/2S485-N module.	The communication parameters of the expansion module are configured incorrectly or the format does not meet the requirements.	Check the parameter configuration. For details, see GL20-2S485 Communication Module User Guide or GL20-2S485-N Communication Module User Guide.
35011	The communication mode is incorrectly configured for channels of 2S485/2S485-N module.		
35012			

Error code	Error message	Detailed description	Solution
	The bit-related parameters are incorrectly configured for channels of 2S485/2S485-N module.		
35013	The data packet size is incorrectly configured for channels of 2S485/2S485-N module.		
35014	The message interval is incorrectly configured for channels of 2S485/2S485-N module.		
35015	The TX buffer is incorrectly configured for channels of 2S485/2S485-N module.		
35016	The RX buffer is incorrectly configured for channels of 2S485/2S485-N module.		
35042	The serial parameters are incorrectly configured for the channels of the 2SCOM module.		For details on parameter configuration, see GL20-2SCOM Serial Communication Module User Guide.
35043	The data packet size is incorrectly configured for the channels of the 2SCOM module.		
35044	The timeout parameter is incorrectly configured for the channels of the 2SCOM module.		
35045	The port protocol is incorrectly configured for the channels of the 2SCOM module.		
36010	2S485/2S485-N module channel TX buffer overflows	The data in the module channel TX buffer exceeds the limit or the transmission interval is too short.	Increase the baud rate or reduce the sent data length and increase the transmission interval.

Error code	Error message	Detailed description	Solution
36011	2S485/2S485-N module channel RX buffer overflows	The data in the module channel RX buffer exceeds the limit or the reception interval is too short.	Reduce the sent data length and increase the reception interval.
36012	Frame loss occurred in frame reception by 2S485/2S485-N module channel	The module channel experiences frame count discontinuity in frame reception.	Increase the baud rate or reduce the data length and increase the transmission interval.
36013	Frame loss occurred in multiple frames reception by the 2S485/2S485-N module	The module channel experiences frame count discontinuity in multi-frame reception.	Increase the baud rate or reduce the data length and increase the transmission interval.
36014	Noise detected on 2S485/2S485-N module channel	Noise is detected on the module channel pins.	Ensure that the signal of the module channel is normal and valid and is connected properly and reliably.
36015	2S485/2S485-N module channel parity bit error	The module channel data check failed.	Ensure that the parity check and validation parameters are set correctly.
36042	SSI module channel 24 V output fault	The module channel 24 V output is disconnected or an overvoltage/undervoltage fault occurs.	Ensure that the 24 V output port of the module channel is connected properly and reliably.
36043	SSI module channel communication fault	The signal cable (D+, D-) of the module channel is disconnected or connected incorrectly.	Ensure that the signal cable (D+, D-) of the module channel is connected properly and reliably and that the signal is normal and valid.
36044	SSI module channel data error	The data is lost during module channel transmission.	Make sure that the baud rate parameters are set properly.
36058	2SCOM module channel TX buffer overflows	The data in the module channel TX buffer exceeds the limit or the transmission interval is too short.	Increase the baud rate or reduce the sent data length and increase the transmission interval.
36059	2SCOM module channel RX buffer overflows	The data in the module channel RX buffer exceeds the limit or the reception interval is too short.	Reduce the sent data length and increase the reception interval.

Error code	Error message	Detailed description	Solution
36060	Frame loss occurred in frame reception by 2SCOM module channel	The module channel experiences frame count discontinuity in frame reception.	Increase the baud rate or reduce the data length and increase the transmission interval.
36061	Frame loss occurred in multiple frames reception by the 2SCOM module	The module channel experiences frame count discontinuity in multi-frame reception.	Increase the baud rate or reduce the data length and increase the transmission interval.
36062	Noise detected on 2SCOM module channel	Noise is detected on the module channel pins.	Ensure that the signal of the module channel is normal and valid and connected properly and reliably.
36063	2SCOM module channel parity bit error	The module channel data check failed.	Ensure that the parity check and validation parameters are set correctly.
Local encoder axis			
6300	Input device not assigned or assigned input device invalid	The local encoder axis must be assigned with a high-speed counter, and each high-speed counter can only be assigned to one axis, otherwise the axis cannot work properly.	Assign a high-speed counter that has not been assigned yet in “Input Device” on the “Basic Settings” page of the axis.
6301	Axis unit conversion settings invalid	After a high-speed counter is assigned to an axis, its count value (pulse unit) is converted into the equivalent in user unit according to the unit conversion setting parameter. If the number of pulses per revolution of the encoder, the displacement of the encoder per revolution, or the gear ratio of the transmission device is set incorrectly, the axis cannot work properly.	Check the settings on the “Unit Conversion Settings” page of the axis and correct the parameter values.
6302	Axis software limit or revolution cycle configuration invalid	In linear mode, the negative limit must be less than 0, and the positive limit must be greater than 0. In rotary mode, the revolution cycle must be greater than 0. Since the high-	Linear mode: Modify the positive and negative limits to ensure that the negative limit is less than 0, the positive limit is greater than 0, and they are 32-bit integers in the range of [-

Error code	Error message	Detailed description	Solution
		speed counter is a 32-bit counter, the negative and positive limits must be within the 32-bit integer range [-2147483648, +2147483647] after being converted to pulse units.	2147483648, +2147483647] after being converted into pulse units. Rotary mode: Modify the revolution cycle to ensure that it is greater than 0 and is a 32-bit integer in the range of [-2147483648, +2147483647] after being converted into pulse units.
6303	Axis counting mode or signal source configuration invalid	The high-speed counter supports the following counting modes and signal sources: A/B phase frequency multiplication by 1: X0-A phase, X1-B phase, X2-A phase, X3-B phase A/B phase frequency multiplication by 2: X0-A phase, X1-B phase, X2-A phase, X3-B phase A/B phase frequency multiplication by 4: X0-A phase, X1-B phase, X2-A phase, X3-B phase CW/CCW: X0-CW, X1-CCW, X2-CW, X3-CCW Pulse+direction: X0-pulse, X1-direction, X2-pulse, X3-direction	Select a supported counting mode and signal source.
6304	Axis preset setting: input terminal invalid	The preset function supports the input terminals X0, X1, X2, X3, X4, X5, X6, and X7.	Select an input terminal supported by the preset function.
6305	Axis probe 1: input terminal invalid	Probe 1 supports the input terminals X0, X1, X2, X3, X4, X5, X6, and X7.	Select an input terminal supported by probe 1.
6306	Axis probe 2: input terminal invalid	Probe 2 supports the input terminals X0, X1, X2, X3, X4, X5, X6, and X7.	Select an input terminal supported by probe 2.
6307	Axis comparison output: terminal invalid	The comparative output supports the output terminals Y0, Y1, Y2, and Y3.	Select an output terminal supported by the comparison output.
6308	Axis comparison output: pulse width invalid	When the unit is ms, the time range is 0.1 ms to 6553.5 ms. When the unit is Unit, the set value must fall between 1 and	Modify the pulse width to ensure that it is within the allowable range.

Error code	Error message	Detailed description	Solution
		65535 after being converted into pulse units.	
CANlink			
6400	Station address conflict: Station address already exists in the network.	In CANlink communication, the addresses of all stations connected to the network must be unique. Address conflict detection is performed after a device node is powered on and initialized or the station address is modified. If the address is duplicated, a fault is reported and all CANlink bus activities of the node are stopped.	Change the station address to ensure that there are no duplicate addresses in the network.
6401	Slave offline	Failed to communicate with the slave because it is offline.	Check whether the CAN network connection works properly. Ensure that the connection is reliable without short circuit or open circuit, CANH and CANL are not reversely connected, and the terminal resistance is normal.
6403	CANlink configuration data inconsistent	1. The device is powered off immediately after online modification, making the configuration file not saved fully. 2. The software version does not match the firmware version.	1. Re-download the configuration file. 2. Upgrade the version.
6411	Slave configuration exception response (1) “Undefined encoding used”	During configuration of a slave, the slave returns exception response (1) “Undefined encoding used”.	Check whether the type/model of the connected device is consistent with the configuration.
6412	Slave configuration exception response (2) “Configured index exceeds the maximum value supported by the node”	During configuration of a slave, the slave returns exception response (2) “Configured index exceeds the maximum value supported by the node”.	Check whether the type/model of the connected device is consistent with the configuration.
6413			

Error code	Error message	Detailed description	Solution
	Slave configuration exception response (3) "Register address non-existent or inaccessible"	During configuration of a slave, the slave returns exception response (3) "Register address non-existent or inaccessible".	Check whether the type/model of the connected device is consistent with the configuration.
6415	Slave configuration exception response (5) "Register data length invalid"	During configuration of a slave, the slave returns exception response (5) "Register data length invalid".	Check whether the type/model of the connected device is consistent with the configuration.
6416	Waiting for slave configuration command response timed out	During configuration of a slave, waiting for slave response timed out.	Check whether the type/model of the connected device is consistent with the configuration.
6421	Slave synchronization exception response (1) "Illegal command code"	When a synchronization command is sent to a slave, the slave returns exception response (1) "Illegal command code".	Check whether the type/model of the connected device is consistent with the configuration.
6422	Slave synchronization exception response (2) "Register address non-existent or inaccessible"	When synchronization data is sent to a slave, the slave returns exception response (2) "Register address non-existent or inaccessible".	Check whether the type/model of the connected device is consistent with the configuration.
6423	Slave synchronization exception response (3) "Value beyond allowable range"	When synchronization data is sent to a slave, the slave returns exception response (3) "Value beyond allowable range".	1. Check whether the set value in the corresponding register address has exceeded the allowed range. 2. Check whether the type/model of the connected device is consistent with the configuration.
6424	Slave synchronization exception response (4) "Operation unreachable or not allowed in the current state"	When synchronization data is sent to a slave, the slave returns exception response (4) "Operation unreachable or not allowed in the current state".	Check whether the type/model of the connected device is consistent with the configuration.
6425	Slave synchronization exception response (5) "Data length invalid"	When synchronization data is sent to a slave, the slave returns exception response (5) "Data length invalid".	Check whether the type/model of the connected device is consistent with the configuration.

Error code	Error message	Detailed description	Solution
6426	Waiting for slave synchronization command response timed out	Waiting for slave response to a synchronization command timed out.	Check whether the type/model of the connected device is consistent with the configuration.
CANopen			
6401	Node offline	Failed to communicate with the node because it is offline.	Check whether the CAN network connection works properly. Ensure that the connection is reliable without short circuit or open circuit, CANH and CANL are not reversely connected, and the terminal resistance is normal.
6500	CANopen configuration data inconsistent	1. The device is powered off immediately after online modification, making the configuration file not saved fully. 2. The software version does not match the firmware version.	1. Re-download the configuration file. 2. Upgrade the version.
Modbus Master			
5500	8-bit data required for Modbus RTU serial port	Use 8-bit data for Modbus RTU serial port.	Use 8-bit data for Modbus RTU serial port.
6001	Slave returned exception response (01) “Ilegal parameter”	The parameter received in the query is not an allowable action for the server (or slave). This may be because the parameter is only applicable to new devices, and is not implementable in the unit selected. It could also indicate that the server (or slave) is in the wrong state to process a request of this type, for example, because it is unconfigured and is being asked to return register values.	Check whether the server (or slave) supports the parameter.
6002	Slave returned exception response (02) “Illegal data address”	The data address received in the query is not an allowable address for the server (or	Check whether the corresponding parameter of the server (or slave) supports all the

Error code	Error message	Detailed description	Solution
		slave). More specifically, the combination of reference number and transfer length is invalid. For a controller with 100 registers, a request with the offset 96 and the length 4 will succeed, but a request with the offset 96 and the length 5 will result in exception code 02.	addresses accessed by this configuration.
6003	Slave returned exception response (03) “Illegal data”	A value contained in the query data field is not an allowable value for the server (or slave). This indicates a fault in the structure of the remainder of a complex request, such as that the implied length is incorrect. It specifically does not mean that a data item submitted for storage in a register has a value outside the expectation of the application program, since the Modbus protocol is unaware of the significance of any particular value of any particular register.	Check whether the value is within the allowed range.
6004	Slave returned exception response (04) “Slave device fault”	An unrecoverable error occurred while the server (or slave) was attempting to perform the requested action.	Check whether slave is abnormal or faulty.
6009	Error code returned by a third-party slave	Error code returned by a third-party slave	See the third-party slave guide for solutions.
6128	Response station number and requested station number mismatch	After the master sends a request frame, the station number in the received response frame is inconsistent with that in the transmitted request frame.	Check whether the connected slave is a normal Modbus slave.
6129	Response parameter and requested parameter mismatch	After the master sends a request frame, the parameter in the received response frame is inconsistent with that in the transmitted request frame.	Check whether the connected slave is a normal Modbus slave.

Error code	Error message	Detailed description	Solution
6130	Response data address and requested data address mismatch	After the master sends a request frame, the data address in the received response frame is inconsistent with that in the transmitted request frame.	Check whether the connected slave is a normal Modbus slave.
6131	Response data value and requested data value mismatch	After the master sends a request frame, the data value in the received response frame is inconsistent with that in the transmitted request frame.	Check whether the connected slave is a normal Modbus slave.
6240	Cache address mapping in configuration invalid	The cache address mapping in the configuration is invalid and the configuration cannot be executed correctly.	Modify the cache address mapping in the configuration to a valid variable or element address.
6255	Request timed out.	After sending a request frame, if the master does not receive a response from the slave within the specified timeout period, it retries according to the set number of retries. When the retry attempts exceed the set number, the master considers the slave abnormal and reports a request timeout error.	1. Ensure that the communication network cable is connected reliably. 2. Ensure that the slave station number is consistent with the configured slave station number. 3. Modify the timeout period to ensure that the master can receive the response frame within the timeout period. 4. Check whether the connected slave is a normal Modbus slave.
Modbus TCP Master			
6000	Configuration disconnected	The Modbus TCP client fails to establish a TCP connection with the server.	1. Ensure that the communication network cable is connected reliably. 2. Check whether the slave IP address and port ID are consistent with the configuration. 3. If the client and server are connected through a network bridge, router, or gateway, make sure that the client and server gateways are set correctly.

Error code	Error message	Detailed description	Solution
6001	Slave returned exception response (01) “Illegal parameter”	The parameter received in the query is not an allowable action for the server (or slave). This may be because the parameter is only applicable to new devices, and is not implementable in the unit selected. It could also indicate that the server (or slave) is in the wrong state to process a request of this type, for example, because it is unconfigured and is being asked to return register values.	Check whether the server (or slave) supports the parameter.
6002	Slave returned exception response (02) “Illegal data address”	The data address received in the query is not an allowable address for the server (or slave). More specifically, the combination of reference number and transfer length is invalid. For a controller with 100 registers, a request with the offset 96 and the length 4 will succeed, but a request with the offset 96 and the length 5 will result in exception code 02.	Check whether the corresponding parameter of the server (or slave) supports all the addresses accessed by this configuration.
6003	Slave returned exception response (03) “Illegal data”	A value contained in the query data field is not an allowable value for the server (or slave). This indicates a fault in the structure of the remainder of a complex request, such as that the implied length is incorrect. It specifically does not mean that a data item submitted for storage in a register has a value outside the expectation of the application program, since the Modbus protocol is unaware of the significance of any particular value of any particular register.	Check whether the value is within the allowed range.

Error code	Error message	Detailed description	Solution
6004	Slave returned exception response (04) “Slave device fault”	An unrecoverable error occurred while the server (or slave) was attempting to perform the requested action.	Check whether slave is abnormal or faulty.
6101	Modbus TCP configuration data inconsistent	1. The device is powered off immediately after online modification, making the configuration file not saved fully. 2. The software version does not match the firmware version.	1. Re-download the configuration file. 2. Upgrade the version.
6128	Response station number and requested station number mismatch	After the master sends a request frame, the station number in the received response frame is inconsistent with that in the transmitted request frame.	Check whether the connected slave is a normal Modbus slave.
6129	Response parameter and requested parameter mismatch	After the master sends a request frame, the parameter in the received response frame is inconsistent with that in the transmitted request frame.	Check whether the connected slave is a normal Modbus slave.
6130	Response data address and requested data address mismatch	After the master sends a request frame, the data address in the received response frame is inconsistent with that in the transmitted request frame.	Check whether the connected slave is a normal Modbus slave.
6131	Response data value and requested data value mismatch	After the master sends a request frame, the data value in the received response frame is inconsistent with that in the transmitted request frame.	Check whether the connected slave is a normal Modbus slave.
6240	Cache address mapping in configuration invalid	The cache address mapping in the configuration is invalid and the configuration cannot be executed correctly.	Modify the cache address mapping in the configuration to a valid variable or element address.
6255	Request timed out.	After sending a request frame, if the master does not receive a	

Error code	Error message	Detailed description	Solution
		response from the slave within the specified timeout period, it retries according to the set number of retries. When the retry attempts exceed the set number, the master considers the slave abnormal and reports a request timeout error.	1. Ensure that the communication network cable is connected reliably. 2. Ensure that the slave station number is consistent with the configured slave station number. 3. Modify the timeout period to ensure that the master can receive the response frame within the timeout period. 4. Check whether the connected slave is a normal Modbus slave.
EtherNet/IP			
6600	EtherNet/IP connection instance: not connected	When the PLC is in the “Stopped” state, the connection instance is displayed as not connected.	Switch the PLC to the “Running” state.
6602	EtherNet/IP connection instance: connection failed, target device offline	1. The IP of the connection instance is inconsistent with the IP of the target device. 2. The target device is powered off or is in the “Stopped” state. 3. Network device failure.	1. Set the IP of the connection instance to the IP of the target device. 2. Switch the target device to the “Running” state. 3. Replace the switch or cable. In case of connection across network segments, configure a gateway.
6603	EtherNet/IP connection instance: connection failed, target device not responding	The target device is online, but the EtherNet/IP protocol of the target device is abnormal.	Power off and on the target device.
6604	EtherNet/IP connection instance: connection failed, target device returned an error response	The specified connection path does not exist, or the specified data size does not match.	Open the connection status page, check the connection status code and status description, and modify the connection path, data size, or other relevant parameters.
6605	EtherNet/IP connection instance: connection timeout	1. High CPU (reaching 90%) of PLC or target device can cause timeout in packet message reception.	1. Increase the PLC scan cycle, increase the RPI cycle of the EtherNet/IP connection, and reduce the number of EtherNet/

Error code	Error message	Detailed description	Solution
		2. Network device failure.	IP connections to reduce the CPU load. Use a high-performance PLC or target device. 2. Replace the switch or cable.
6607	EtherNet/IP configuration data format inconsistent	The software version does not match the board version.	Select AutoShop software and PLC firmware that are compatible with each other.
6631	EtherNet/IP display label: cannot connect to the path specified in the request	1. The IP of the parameter is inconsistent with the IP of the target device. 2. The target device is powered off or is in the “Stopped” state. 3. Network device failure.	1. Use the target device IP as the request parameter. 2. Switch the target device to the “Running” state. 3. Network troubleshooting: Replace the switch or cable. In case of connection across network segments, configure a gateway.
6632	EtherNet/IP display label: request timed out, no response received	1. High CPU (reaching 90%) of PLC or target device can cause timeout in packet message reception. 2. Network device failure.	1. Reduce the CPU load or use a high-performance PLC or target device. 2. Replace the switch or cable.
6633	EtherNet/IP display label: request succeeded, but error response received	1. The name of the specified label is inconsistent with that of the target label. 2. The element quantity of the specified label is inconsistent with that of the target label, and exceeds the limit.	1. Set the parameter according to the name of the target label. 2. Set the element quantity to a value less than or equal to the element quantity of the target label.
6634	EtherNet/IP display label: request succeeded, but data type inconsistent	The data type of the specified label is inconsistent with that of the target label.	Set the parameter according to the data type of the target label.
6635	EtherNet/IP display label: request succeeded, but data length inconsistent	The data packet returned by the target device is abnormal.	Power off and on the target device.
EtherCAT			

Error code	Error message	Detailed description	Solution
8001	Failed to read the master configuration information.	Failed to read the master configuration information.	Check whether the board software and software tool versions match.
8002	Failed to obtain slave configuration parameters.	Failed to obtain slave configuration parameters.	Check whether the board software and software tool versions match.
8003	EtherCAT startup timed out.	EtherCAT startup timed out.	1. Check whether the network is properly connected. 2. Check whether the connected slave is consistent with the configuration. 3. Check whether the slave type matches.
8004	Failed to request the master.	Failed to request the master.	Restart the PLC.
8200	Failed to write the slave startup parameters to the SDO.	Failed to write the slave startup parameters to the SDO.	1. Check whether there is an object dictionary that is not supported by the slave in the startup parameter list. 2. Check whether the value of the object dictionary is out of range.
8201	The slave is lost during operation.	The slave is lost during operation.	1. Check whether the network with the slave is disconnected. 2. Check whether the slave is powered off. 3. In the alias mode, check whether the actual alias of the slave is consistent with the alias for project configuration.
8202	The slave state machine is switched to non-OP mode.	The slave state machine is switched to non-OP mode.	1. Check whether the network with the slave is disconnected. 2. Check whether the slave is powered off.
8203	Slave state machine switching failed.	Slave state machine switching failed.	-
8204	Incorrect slave type	The slave type is incorrect.	1. Check whether the network cable is reversely connected.

Error code	Error message	Detailed description	Solution
			2. Check whether the connected device matches the configuration.
8205	PDO address error	The PDO address is incorrect.	1. Check whether the memory runs out. 2. Check whether the background and board software versions match. 3. Power off and restart the PLC.
8206	Incorrect PDO length	The PDO length is incorrect.	Check whether the background and board software versions match.
8301	Failed to switch to INIT state.	Failed to switch to INIT state.	Check whether the slave station machine supports state transition.
8302	Failed to switch to PerOP state.	Failed to switch to PerOP state.	Check whether the slave supports the CoE protocol.
8304	Failed to switch to SafeOP state.	Failed to switch to SafeOP state.	Check whether the PDO communication configuration is correct.
8308	Failed to switch to OP state	Failed to switch to OP state	1. Check the network communication quality. 2. Check whether the EtherCAT task cycle is appropriate.
8310	FMMU unit configuration error	An FMMU unit configuration error occurs.	Check whether the slave supports the FMMU unit.
8311	Email configuration error	An email configuration error occurs.	Check whether the slave supports the SM unit.
8400	ECTA configuration error	An ECTA configuration error occurs.	Check whether the configured expansion module is consistent with the actually connected expansion module.
8401	ECTA hardware error	An ECTA hardware error occurs.	1. Check whether the connection between the ECTA and the expansion module is loose. 2. Replace the ECTA.

Error code	Error message	Detailed description	Solution
8402	ECTA expansion module error	An ECTA expansion module error occurs.	1. Locate the expansion module with the ERR indicator on. 2. Read the diagnosis object dictionary of the faulty module by using ETC_ReadParameter_CoE. 3. Determine the fault type based on description of the diagnosis object dictionary of the expansion module in the ECTA application guide and eliminate the fault.
Motion Control Axis			
9001	Local axis emergency stop active	The emergency stop terminal input is active, and the pulse output is stopped.	Disable the emergency stop terminal input and then call the MC_Reset instruction to reset the fault.
9003	Overspeed	Pulse output frequency over 200 kHz	Check whether the pulse frequency obtained by multiplying the target velocity by the gear ratio exceeds 200 kHz.
9020	Homing error	The negative limit is not mapped.	Map the negative limit in the configuration interface.
9021	Homing error	The positive limit is not mapped.	Map the positive limit on the configuration interface.
9022	Homing error	The home signal is not mapped.	Map the home switch on the configuration interface.
9023	Homing error	1. The output frequency exceeds 200 kHz when the axis runs at the homing velocity. 2. The output frequency exceeds 200 kHz when the axis runs at the homing approach velocity.	1. Modify the unit conversion setting to ensure that the homing velocity and homing approach velocity do not exceed 200 kHz. 2. Change the homing velocity to ensure that the output frequency does not exceed 200 kHz. 3. Change the homing approach velocity to ensure that the output frequency does not exceed 200 kHz.

Error code	Error message	Detailed description	Solution
9024	Homing error	Homing timed out.	1. Check whether the limit signal and home signal can be connected normally. 2. Check whether the homing timeout time is too short.
9025	Homing error	The limit signal is incorrect during homing.	Check whether the limit signal that is not applicable to the current homing mode is triggered.
9030	Limit function enabled	The limit signal input is active during positioning.	Check whether the limit is reached during normal running.
9031	Synchronization error	The target number of transmitted pulses and the actual number of transmitted pulses do not match.	Check whether the limit is reached during normal positioning.
9101	Axis type error or non-existent	1. The type of the axis specified by AxisID is incorrect. 2. The axis specified by AxisID does not exist.	1. Check whether the instruction supports the axis specified by AxisID. 2. Check whether the axis specified by AxisID exists.
9102	Failed to configure the axis.	1. The axis configuration data is lost. 2. The axis configuration parameters are improper.	Check whether the parameters are correct.
9103	The MC_Reset instruction is called when the axis is not faulty.	The MC_Reset instruction is called when the axis is not faulty.	Check whether the MC_Reset instruction is called when the axis is not switched to ErrorStop state.
9104	The axis is in unknown state when the MC_ReadStatus instruction is called.	The axis is in unknown state when the MC_ReadStatus instruction is called.	Check whether the current state of the axis is uncontrollable by using the online monitoring function.
9105	Current position setting not allowed	The MC_SetPosition instruction is called during running or stop.	Set the current position when the axis is in StandStill, Poweroff, or ErrorStop state.
9106	The axis is stopping upon a fault.	The axis is stopping upon a fault.	Execute the instruction after stop upon fault is completed, the fault

Error code	Error message	Detailed description	Solution
			is resolved, and the reset instruction is executed.
9107	The parameters are improper.	The parameters are improper.	Check whether the parameters on the left of the instruction are set properly.
9108	The PLCOpen state machine is improper.	The PLCOpen state machine is improper.	Check whether the current PLCOpen state machine satisfies the execution conditions for this instruction. If not, call the relevant instruction to switch the axis to the required state.
9110	The MC_Stop instruction is called repeatedly during stop.	The MC_Stop instruction is called repeatedly during stop.	Trigger only one MC_Stop instruction at a time.
9111	The instruction linked list is lost.	The instruction linked list is lost.	1. Check whether the background version and board version match. 2. Contact the manufacturer.
9112	AxisID changed	The value of AxisID is changed while the instruction flow is active.	Do not change the axis number while the flow is active for Enable instructions such as MC_Power and MC_Jog.
9113	Reset by executing the MC_Reset instruction timed out.	Reset by executing the MC_Reset instruction timed out.	1. Check whether the drive fault can be reset. 2. Check whether the axis fault type supports reset.
9114	The axis fails to write to 0x6060.	The axis fails to write to 0x6060.	1. Check for interference in network communication. 2. Check whether the slave supports the object dictionary 0x6060.
9115	The MC_Halt instruction is called when the axis is in Stopping state.	The MC_Halt instruction is called when the axis is in Stopping state.	Do not call the MC_Halt instruction when the axis is in Stopping state.
9116	The current axis is in online commissioning mode.	The current axis is in online commissioning mode.	Check whether the current axis is in online commissioning mode. PLC motion control instructions

Error code	Error message	Detailed description	Solution
			are invalid in online commissioning mode.
9117	The instruction is not developed yet.	The instruction is not developed yet.	Reserved
9118	The acceleration (deceleration) of the instruction exceeds the maximum acceleration.	The acceleration (deceleration) of the instruction exceeds the maximum acceleration.	Check whether the acceleration (deceleration) of the instruction exceeds the maximum acceleration.
9119	The target velocity of the MC_Jog instruction exceeds the maximum jogging velocity.	The target velocity of the MC_Jog instruction exceeds the maximum jogging velocity.	Check whether the target velocity of the MC_Jog instruction exceeds the maximum jogging velocity.
9120	The target velocity exceeds the maximum velocity.	The target velocity exceeds the maximum velocity.	Check whether the target velocity of the instruction exceeds the maximum velocity.
9121	The forward and reverse motion signals of the jog instruction are both active.	The forward and reverse motion signals of the jog instruction are both active.	Ensure that the forward and reverse motion signals of the jog instruction are not active at the same time.
9122	The control word is not mapped to the EtherCAT bus axis.	The control word is not mapped to the EtherCAT bus axis.	Add the control word in the PDO and map it to the axis.
9123	The target position is not mapped to the EtherCAT bus axis.	The target position is not mapped to the EtherCAT bus axis.	Add the target position in the PDO and map it to the axis.
9124	The target torque is not mapped to the EtherCAT bus axis.	The target torque is not mapped to the EtherCAT bus axis.	Add the target torque in the PDO and map it to the axis.
9125	The status word is not mapped to the EtherCAT bus axis.	The status word is not mapped to the EtherCAT bus axis.	Add the status word in the PDO and map it to the axis.
9126	The current position is not mapped to the EtherCAT bus axis.	The current position is not mapped to the EtherCAT bus axis.	Add the feedback position in the PDO and map it to the axis.
9127	The 0x60fd is not mapped to the EtherCAT bus axis.	The 0x60fd is not mapped to the EtherCAT bus axis.	Add 0x60fd in the PDO and map it to the axis.

Error code	Error message	Detailed description	Solution
9128	The current torque is not mapped to the EtherCAT bus axis.	The current torque is not mapped to the EtherCAT bus axis.	Add the current torque in the PDO and map it to the axis.
9129	The probe control word is not mapped to the EtherCAT bus axis.	The probe control word is not mapped to the EtherCAT bus axis.	Add the probe control word in the PDO and map it to the axis.
9130	The probe status word is not mapped to the EtherCAT bus axis.	The probe status word is not mapped to the EtherCAT bus axis.	Add the probe status word in the PDO and map it to the axis.
9131	The probe position is not mapped to the EtherCAT bus axis.	The probe position is not mapped to the EtherCAT bus axis.	Add the probe position in the PDO and map it to the axis.
9132	An interrupt positioning instruction is being executed and the probe channel is occupied.	An interrupt positioning instruction is being executed and the probe channel is occupied.	The probe instruction and interrupt positioning instruction must not occupy the same probe channel at the same time. When the two instructions are called simultaneously in the program, the interrupt positioning instruction takes priority.
9133	The virtual axis mode is enabled.	The virtual axis mode is enabled.	The current instruction does not support the virtual axis mode.
9134	The imaginary axis probe is being used.	The imaginary axis probe is being used.	The H5U supports two imaginary axis probes. Check whether the current probe is out of range.
9135	The interrupt signal is not triggered in the interrupt positioning instruction.	The interrupt signal is not triggered in the interrupt positioning instruction.	During execution of the interrupt positioning instruction, no interrupt signal is detected after positioning is completed.
9136	The probe channel is occupied by another instruction during the interrupt positioning process.	The probe channel is occupied by another instruction during the interrupt positioning process.	Ensure that the probe channel is not occupied during the interrupt positioning process.
9137	The control mode 0x6060 is not mapped to the bus driver.	The control mode 0x6060 is not mapped to the bus driver.	Add 0x6060 in the PDO and map it to the axis.
9138			

Error code	Error message	Detailed description	Solution
	The control mode 0x6061 is not mapped to the bus driver.	The control mode 0x6061 is not mapped to the bus driver.	Add 0x6061 in the PDO and map it to the axis.
9139	The MC_Home instruction is called repeatedly during homing.	The MC_Home instruction is called repeatedly during homing.	Do not call the MC_Home instruction repeatedly during homing.
9140	The target torque of the instruction exceeds the maximum value.	The target torque of the instruction exceeds the maximum value.	Check whether the target torque of the instruction exceeds the positive and negative torque limits.
9141	The maximum velocity is not mapped to the bus driver.	The maximum velocity is not mapped to the bus driver.	Add 0x607f in the PDO and map it to the axis.
9142	The immediate stop instruction is active.	The immediate stop instruction is active.	Check whether the immediate stop instruction has been called.
9143	The immediate stop instruction is called repeatedly.	The immediate stop instruction is called repeatedly.	Check whether the immediate stop instruction is called repeatedly.
9144	The limit is reached during jogging.	The limit is reached during jogging.	Check whether the limit is active.
9145	Target position exceeded 9999999	The precision is reduced if a single-precision floating-point number exceeds 9999999. Therefore, the target position must not exceed this value.	1. Check whether the target position is correct. Set the target position again. 2. Change the gear ratio to ensure that the target position is not greater than 9999999.
9146	Target velocity exceeded 9999999	The precision is reduced if a single-precision floating-point number exceeds 9999999. Therefore, the target velocity must not exceed this value.	1. Check whether the target velocity is correct. Set the target velocity again. 2. Change the gear ratio to ensure that the target velocity is not greater than 9999999.
9147	Target acceleration exceeded 9999999	The precision is reduced if a single-precision floating-point number exceeds 9999999. Therefore, the target acceleration must not exceed this value.	1. Check whether the target acceleration is correct. Set the target acceleration again. 2. Change the gear ratio to ensure that the target acceleration is not greater than 9999999.

Error code	Error message	Detailed description	Solution
9148	Target deceleration exceeded 9999999	The precision is reduced if a single-precision floating-point number exceeds 9999999. Therefore, the target deceleration must not exceed this value.	1. Check whether the target deceleration is correct. Set the target deceleration again. 2. Change the gear ratio to ensure that the target deceleration is not greater than 9999999.
9149	Axis in sync control mode, abortion not allowed	A single-axis motion instruction is called when the axis is performing interpolation in sync control mode. The single-axis motion instruction reports an error.	Do not call single-axis motion instructions during interpolation.
9150	MC_Halt in execution, abortion not allowed	The MC_MoveSuperImposer instruction is called while the MC_Halt instruction is still active.	Do not call the MC_MoveSuperImposer instruction while the MC_Halt instruction is still active.
9151	The variable PulseWidth of the MC_MoveVelocityCSV instruction is out of range.	The variable PulseWidth of the MC_MoveVelocityCSV instruction is out of range.	Ensure that the parameter value is within the allowable range.
9152	The object dictionary 60FFh is not associated in I/O mapping of the bus servo axis when the MC_MoveVelocityCSV instruction is called.	The object dictionary 60FFh is not associated in I/O mapping of the bus servo axis when the MC_MoveVelocityCSV instruction is called.	Ensure that the object dictionary 60FFh is associated in I/O mapping of the bus servo axis when the MC_MoveVelocityCSV instruction is called.
9153	The probe terminal is not configured.	The probe terminal is not configured.	Check whether the software tool version supports configuration of the probe terminal ID.
9154	The value of ParameterIndex of the MC_SetAxisConfigPara instruction is out of range.	The value of ParameterIndex of the MC_SetAxisConfigPara instruction is out of range.	Ensure that the parameter value is within the allowable range.
9155	The configuration parameters of the axis are being modified, and execution of this instruction is not allowed before the modification is completed.	The configuration parameters of the axis are being modified, and execution of this instruction is not allowed before the modification is completed.	Perform the enable operation after axis initialization is completed.

Error code	Error message	Detailed description	Solution
9156	MC_SetAxisConfigPara does not support multi-execution.	MC_SetAxisConfigPara does not support multi-execution.	Note that this instruction does not support re-execution or multi-execution.
9157	The gear/cam motion instruction is not supported by the axis due to axis properties.	The gear/cam motion instruction is not supported by the axis due to axis properties.	Ensure that the axis is not in single-axis mode or that the PLC supports the motion instruction.
9200	Failed to obtain the cam table configuration file.	Failed to obtain the cam table configuration file.	1. Check whether the board software and software tool match. 2. Re-download the cam configuration table.
9201	Failed to obtain the master axis.	Failed to obtain the master axis.	1. Check whether the master axis called in the program exists. 2. Check whether the master axis has reported an error. 3. Check whether the master axis is a profile axis.
9202	Failed to obtain the slave axis.	Failed to obtain the slave axis.	1. Check whether the slave axis called in the program exists. 2. Check whether the slave axis has reported an error.
9203	Failed to obtain the cam table.	Failed to obtain the cam table.	Check whether the cam table called exists.
9204	The number of cams executed simultaneously in the PLC program exceeds the maximum allowable value.	The number of cams executed simultaneously in the PLC program exceeds the maximum allowable value.	Check whether the number of cams executed simultaneously in the program exceeds the threshold.
9205	The corresponding cam node is not found.	The corresponding cam node is not found.	1. This instruction can be called only when the slave axis is in cam engagement state. 2. The specified slave axis is not a cam node. Set the parameters again..
9206	The master axis is changed during cam engagement.	The master axis is changed during cam engagement.	Do not change the master axis during cam engagement.

Error code	Error message	Detailed description	Solution
9207	StartMode of the MC_CamIn instruction is out of range.	StartMode of the MC_CamIn instruction is out of range.	Ensure that the parameter value is within the specified range.
9208	StartPosition of the MC_CamIn instruction exceeds the maximum allowable value.	StartPosition of the MC_CamIn instruction exceeds the maximum allowable value.	Ensure that the absolute value of the floating-point number in the motion control instruction does not exceed 9999999.
9209	MasterStartDistance of the MC_CamIn instruction exceeds the maximum allowable value.	MasterStartDistance of the MC_CamIn instruction exceeds the maximum allowable value.	Ensure that the absolute value of the floating-point number in the motion control instruction does not exceed 9999999.
9210	MasterScaling of the MC_CamIn instruction exceeds the maximum allowable value.	MasterScaling of the MC_CamIn instruction exceeds the maximum allowable value.	Ensure that the absolute value of the floating-point number in the motion control instruction does not exceed 9999999.
9211	SlaveScaling of the MC_CamIn instruction exceeds the maximum allowable value.	SlaveScaling of the MC_CamIn instruction exceeds the maximum allowable value.	Ensure that the absolute value of the floating-point number in the motion control instruction does not exceed 9999999.
9212	MasterOffset of the MC_CamIn instruction exceeds the maximum allowable value.	MasterOffset of the MC_CamIn instruction exceeds the maximum allowable value.	Ensure that the absolute value of the floating-point number in the motion control instruction does not exceed 9999999.
9213	SlaveOffset of the MC_CamIn instruction exceeds the maximum allowable value.	SlaveOffset of the MC_CamIn instruction exceeds the maximum allowable value.	Ensure that the absolute value of the floating-point number in the motion control instruction does not exceed 9999999.
9214	MasterScaling of the MC_CamIn instruction is not a positive number.	MasterScaling of the MC_CamIn instruction is not a positive number.	Set this parameter to a positive number.
9215	SlaveScaling of the MC_CamIn instruction is not a positive number.	SlaveScaling of the MC_CamIn instruction is not a positive number.	Set this parameter to a positive number.
9216	ReferenceType of the MC_CamIn/MC_GearIn instruction is out of range.	ReferenceType of the MC_CamIn/MC_GearIn instruction is out of range.	Ensure that the parameter value is within the specified range.
9217	Direction of the MC_CamIn instruction is out of range.	Direction of the MC_CamIn instruction is out of range.	Ensure that the parameter value is within the specified range.

Error code	Error message	Detailed description	Solution
9218	BufferMode of the MC_CamIn instruction is out of range.	BufferMode of the MC_CamIn instruction is out of range.	Ensure that the parameter value is within the specified range.
9219	The master axis phases in the node array of the cam table are not sorted in ascending order.	The master axis phases in the node array of the cam table are not sorted in ascending order.	Sort the master axis phases in ascending order when customizing cam table nodes.
9220	The curve type setting of the node array of the cam table is out of range.	The curve type setting of the node array of the cam table is out of range.	Check whether the curve type of the cam node array is set incorrectly.
9221	The target deceleration of the MC_CamOut instruction exceeds the maximum allowable value.	The target deceleration of the MC_CamOut instruction exceeds the maximum allowable value.	Ensure that the absolute value of the floating-point number in the motion control instruction does not exceed 9999999.
9222	MC_CamOut target deceleration out of range	The target deceleration of the MC_GearOut instruction is out of range and causes stop upon a fault.	Ensure that the target deceleration is within the specified range.
9223	The target acceleration of the MC_Phasing instruction exceeds the maximum allowable value.	The target acceleration of the MC_Phasing instruction exceeds the maximum allowable value.	Ensure that the absolute value of the floating-point number in the motion control instruction does not exceed 9999999.
9224	The target acceleration of the MC_Phasing instruction is out of range.	The target acceleration of the MC_Phasing instruction is out of range.	Ensure that the target acceleration is within the specified range.
9225	The target velocity of the MC_Phasing instruction exceeds the maximum allowable value.	The target velocity of the MC_Phasing instruction exceeds the maximum allowable value.	Ensure that the absolute value of the floating-point number in the motion control instruction does not exceed 9999999.
9226	The target velocity of the MC_Phasing instruction is out of range.	The target velocity of the MC_Phasing instruction is out of range.	Ensure that the target deceleration is within the specified range.
9227	The curve type setting of the MC_GearOut instruction is out of range.	The curve type setting of the MC_GearOut instruction is out of range.	Ensure that the curve type setpoint in the instruction is within the specified range.
9228	The value of Mode of the MC_CamOut instruction is out of range.	The value of Mode of the MC_CamOut instruction is out of range.	Ensure that the value of Mode is within the specified range.

Error code	Error message	Detailed description	Solution
9229	The MC_GenerateCamTable instruction detects that the cam node array is empty.	The MC_GenerateCamTable instruction detects that the cam node array is empty.	Contact Inovance for technical support.
9230	The node quantity specified by the MC_GenerateCamTable instruction exceeds the maximum allowable value.	The node quantity specified by the MC_GenerateCamTable instruction exceeds the maximum allowable value.	Check whether the target node quantity specified in the instruction is beyond the specified range.
9231	The value of Mode of the MC_GenerateCamTable instruction is out of range.	The value of Mode of the MC_GenerateCamTable instruction is out of range.	Ensure that the parameter value is within the specified range.
9232	The node quantity specified by the MC_GenerateCamTable instruction is too small.	The node quantity specified by the MC_GenerateCamTable instruction is too small.	Ensure that the node quantity is 2 or more.
9233	The RatioNumerator parameter in the MC_GearIn instruction is set to 0.	The RatioNumerator parameter in the MC_GearIn instruction is set to 0.	Set this parameter to a non-zero floating-point number.
9234	The RatioDenominator parameter in the MC_GearIn instruction is not greater than 0.	The RatioDenominator parameter in the MC_GearIn instruction is not greater than 0.	Set this parameter to a floating-point number greater than 0.
9235	The MC_GenerateCamTable instruction is being executed when the MC_SaveCamTable instruction is called.	The MC_GenerateCamTable instruction is being executed when the MC_SaveCamTable instruction is called.	Do not call the MC_SaveCamTable instruction before the cam table data update operation is completed.
9236	The MC_SaveCamTable instruction is being executed on the cam table when the MC_GenerateCamTable instruction is called.	The MC_SaveCamTable instruction is being executed on the cam table when the MC_GenerateCamTable instruction is called.	Do not call the MC_GenerateCamTable instruction before the cam table is saved.
9237	Failed to open the cam table file during execution of the MC_SaveCamTable instruction.	Failed to open the cam table file during execution of the MC_SaveCamTable instruction.	1. Check whether the PLC memory runs out. 2. Replace the PLC.
9238			1. Check whether the PLC memory runs out.

Error code	Error message	Detailed description	Solution
	Failed to write the cam point quantity when the cam table is being saved.	Failed to write the cam point quantity when the cam table is being saved.	2. Replace the PLC.
9239	Failed to write data when the cam table is being saved.	Failed to write data when the cam table is being saved.	1. Check whether the PLC memory runs out. 2. Replace the PLC.
9240	The phase of the first point is not 0.	The phase of the first point is not 0.	Ensure that the phase of the first point is 0.
9241	The displacement of the first point is not 0.	The displacement of the first point is not 0.	Ensure that the displacement of the first point is 0.
9242	The value of Mode of the MC_GearOut instruction is out of range.	The value of Mode of the MC_GearOut instruction is out of range.	Ensure that the value of Mode is within the specified range.
9243	The target deceleration of the MC_Phasing instruction exceeds the maximum allowable value.	The target deceleration of the MC_Phasing instruction exceeds the maximum allowable value.	Ensure that the absolute value of the floating-point number in the motion control instruction does not exceed 9999999.
9244	The target deceleration of the MC_GearIn instruction exceeds the maximum allowable value.	The target deceleration of the MC_GearIn instruction exceeds the maximum allowable value.	Ensure that the absolute value of the floating-point number in the motion control instruction does not exceed 9999999.
9245	The value of Periodic of the MC_CamIn instruction is out of range.	The value of Periodic of the MC_CamIn instruction is out of range.	Ensure that the parameter value is within the specified range.
9246	The phase in the cam table exceeds the maximum allowable value.	The phase in the cam table exceeds the maximum allowable value.	Ensure that the absolute value of the floating-point number does not exceed 9999999.
9247	The absolute value of the displacement in the cam table exceeds the maximum allowable value.	The absolute value of the displacement in the cam table exceeds the maximum allowable value.	Ensure that the absolute value of the floating-point number does not exceed 9999999.
9248	The absolute value of the link velocity in the cam table exceeds the maximum allowable value.	The absolute value of the link velocity in the cam table exceeds the maximum allowable value.	Ensure that the absolute value of the floating-point number does not exceed 9999999.
9249	The gear node is empty.	The gear node is empty.	Contact the manufacturer.

Error code	Error message	Detailed description	Solution
9250	The master axis and slave axis are the same.	The master axis and slave axis are the same.	Do not use the same axis as both the master axis and slave axis of the cam gear.
9251	The configuration address of the master axis is greater than or equal to that of the slave axis.	The configuration address of the master axis is greater than or equal to that of the slave axis.	When ReferenceType is set to set position of the current cycle, ensure that the configuration address of the master axis is less than that of the slave axis.
9252	The master axis filter coefficient fFilter[0] corresponding to the slave axis is out of range.	The master axis filter coefficient fFilter[0] corresponding to the slave axis is out of range.	Ensure that the value of this variable is between 0 and 1 (0 and 1 included).
9253	The master axis filter coefficient fFilter[1] corresponding to the slave axis is out of range.	The master axis filter coefficient fFilter[1] corresponding to the slave axis is out of range.	Ensure that the value of this variable is between 0 and 1 (0 and 1 included).
9254	The master axis filter coefficient fFilter[2] corresponding to the slave axis is out of range.	The master axis filter coefficient fFilter[2] corresponding to the slave axis is out of range.	Ensure that the value of this variable is between 0 and 1 (0 and 1 included).
9255	The sum of the master axis filter coefficients corresponding to the slave axis is not 1.	The sum of the master axis filter coefficients corresponding to the slave axis is not 1.	Ensure that the sum of the master axis filter coefficients corresponding to the slave axis is 1.
9256	The start position and start distance of the master axis in the MC_CamIn instruction are improper.	The start position and start distance of the master axis in the MC_CamIn instruction are improper.	If the master axis works in linear mode and Direction in the instruction is set to positive, ensure that the cam synchronization point is not less than the cam engagement point.
9257	The start position and start distance of the master axis in the MC_CamIn instruction are improper.	The start position and start distance of the master axis in the MC_CamIn instruction are improper.	If the master axis works in linear mode and Direction in the instruction is set to negative, ensure that the cam synchronization point is not greater than the cam engagement point.
9258	The target deceleration of the MC_GearOut instruction	The target deceleration of the MC_GearOut instruction	Ensure that the absolute value of the floating-point number in the

Error code	Error message	Detailed description	Solution
	exceeds the maximum allowable value.	exceeds the maximum allowable value.	motion control instruction does not exceed 9999999.
9259	MC_Phasing target deceleration out of range	The target deceleration of the MC_Phasing instruction is out of range and causes stop upon a fault.	Ensure that the target deceleration is within the specified range.
9260	MC_GearIn target deceleration out of range	The target deceleration of the MC_GearIn instruction is out of range and causes stop upon a fault.	Ensure that the target deceleration is within the specified range.
9261	MC_GearOut target deceleration out of range	The target deceleration of the MC_GearOut instruction is out of range and causes stop upon a fault.	Ensure that the target deceleration is within the specified range.
9262	The target acceleration of the MC_GearIn instruction exceeds the maximum allowable value.	The target acceleration of the MC_GearIn instruction exceeds the maximum allowable value.	Ensure that the absolute value of the floating-point number in the motion control instruction does not exceed 9999999.
9263	The target acceleration of the MC_GearIn instruction is out of range.	The target acceleration of the MC_GearIn instruction is out of range.	Ensure that the target acceleration is within the specified range.
9264	MC_Phasing curve type setting out of range	The curve type setting of the MC_GearOut instruction is out of range.	Ensure that the curve type setpoint in the instruction is within the specified range.
9265	MC_GearIn curve type setting out of range	The curve type setting of the MC_GearOut instruction is out of range.	Ensure that the curve type setpoint in the instruction is within the specified range.
9266	MC_GearOut curve type setting out of range	The curve type setting of the MC_GearOut instruction is out of range.	Ensure that the curve type setpoint in the instruction is within the specified range.
9267	The slave axis is modified during the cam operation.	The slave axis is modified during the cam operation.	Do not modify the slave axis during the cam operation.
9268	The value of PhasingMode of the MC_Phasing instruction is out of range.	The value of PhasingMode of the MC_Phasing instruction is out of range.	Ensure that the value of the parameter is within the specified range.

Error code	Error message	Detailed description	Solution
9269	The current axis is not in cam control mode when the MC_CamOut instruction is called.	The current axis is not in cam control mode when the MC_CamOut instruction is called.	Ensure that the axis works in cam control mode when the MC_CamOut instruction is called.
9270	The current axis is not in gear control mode when the MC_GearOut instruction is called.	The current axis is not in gear control mode when the MC_GearOut instruction is called.	Ensure that the axis works in gear control mode when the MC_GearOut instruction is called.
9271	The position change of the master axis is too large within a single EtherCAT cycle during cam/gear operation.	The position change of the master axis is too large within a single EtherCAT cycle during cam/gear operation.	Ensure that the position change of the master axis is not greater than half a cam cycle within a single EtherCAT cycle.
9272	The point specified by Phase in the MC_GetCamTableDistance instruction does not fall between the start and end points.	The point specified by Phase in the MC_GetCamTableDistance instruction does not fall between the start and end points.	Ensure that the point specified by Phase is within the specified curve.
9273	The slave axis is changed during execution of the MC_GearIn instruction.	The slave axis is changed during execution of the MC_GearIn instruction.	Do not change the slave axis during execution of the MC_GearIn instruction.
9274	The value of Channel of the MC_DigitalCamSwitch instruction is out of range.	The value of Channel of the MC_DigitalCamSwitch instruction is out of range.	Ensure that the parameter value is within the specified range.
9275	The axis is not found.	The axis is not found.	Ensure that the axis specified by Axis exists.
9276	The number of tappets allowed to be executed at the same time is out of range.	The number of tappets allowed to be executed at the same time is out of range.	Ensure that the number of tappets allowed to be executed at the same time is no more than 8.
9277	The value of ReferenceType of the MC_DigitalCamSwitch instruction is out of range.	The value of ReferenceType of the MC_DigitalCamSwitch instruction is out of range.	Ensure that the parameter value is within the specified range.
9278	The value of Number of the MC_DigitalCamSwitch instruction is out of range.	The value of Number of the MC_DigitalCamSwitch instruction is out of range.	Ensure that the parameter value is within the specified range.
9279			

Error code	Error message	Detailed description	Solution
	The Switches array of the MC_DigitalCamSwitch instruction is empty.	The Switches array of the MC_DigitalCamSwitch instruction is empty.	Check whether the length of the Switches array meets requirements.
9280	The value of fPosition of the tappet array is out of range.	The value of fPosition of the tappet array is out of range.	Ensure that the parameter value is within the specified range.
9281	The value of iMode of the tappet array is out of range.	The value of iMode of the tappet array is out of range.	Ensure that the parameter value is within the specified range.
9282	The value of iDirection of the tappet array is out of range.	The value of iDirection of the tappet array is out of range.	Ensure that the parameter value is within the specified range.
9283	The value of fParameter of the tappet array is out of range.	The value of fParameter of the tappet array is out of range.	Ensure that the parameter value is within the specified range.
9284	When the tappet comparison point is set to time mode, the time setting is out of range.	When the tappet comparison point is set to time mode, the time setting is out of range.	Ensure that the parameter value is within the specified range.
9285	The selected axis is not under cam control when ReferenceType of the MC_DigitalCamSwitch instruction is set to 3.	The selected axis is not under cam control when ReferenceType of the MC_DigitalCamSwitch instruction is set to 3.	Call the MC_DigitalCamSwitch instruction after cam control takes effect.
9286	Axis communication is interrupted during tappet execution.	Axis communication is interrupted during tappet execution.	Ensure that axis communication is not interrupted during tappet execution.
9287	The comparison position start points are the same during tappet execution.	The comparison position start points are the same during tappet execution.	Ensure that the start points are not duplicate.
9288	The comparison position start and end point are the same during tappet execution.	The comparison position start and end point are the same during tappet execution.	Ensure that the start and end points are not duplicate.
9289	The selected tappet terminal is being used by another function.	The selected tappet terminal is being used by another function.	Check whether the terminal is set as the pulse output axis.

Error code	Error message	Detailed description	Solution
9290	The MC_DigitalCamSwitch instruction cannot be executed because the state of the motion control axis is improper.	The MC_DigitalCamSwitch instruction cannot be executed because the state of the motion control axis is improper.	Do not execute the MC_DigitalCamSwitch instruction in homing mode.
9291	The MasterSyncPosition setting in the MC_GearInPos instruction is out of range.	The MasterSyncPosition setting in the MC_GearInPos instruction is out of range.	Ensure that the parameter value is within the specified range.
9292	The SlaveSyncPosition setting in the MC_GearInPos instruction is out of range.	The SlaveSyncPosition setting in the MC_GearInPos instruction is out of range.	Ensure that the parameter value is within the specified range.
9293	The MasterStarDistance setting in the MC_GearInPos instruction is out of range.	The MasterStarDistance setting in the MC_GearInPos instruction is out of range.	Ensure that the parameter value is within the specified range.
9294	The Velocity setting in the MC_GearInPos instruction exceeds the system limit.	The Velocity setting in the MC_GearInPos instruction exceeds the system limit.	Ensure that the parameter value is within the specified range.
9295	The Velocity setting in the MC_GearInPos instruction exceeds the setting limit.	The Velocity setting in the MC_GearInPos instruction exceeds the setting limit.	Ensure that the parameter value is within the specified range.
9296	The Acceleration setting in the MC_GearInPos instruction exceeds the system limit.	The Acceleration setting in the MC_GearInPos instruction exceeds the system limit.	Ensure that the parameter value is within the specified range.
9297	The Acceleration setting in the MC_GearInPos instruction exceeds the setting limit.	The Acceleration setting in the MC_GearInPos instruction exceeds the setting limit.	Ensure that the parameter value is within the specified range.
9298	The Deceleration setting in the MC_GearInPos instruction exceeds the system limit.	The Deceleration setting in the MC_GearInPos instruction exceeds the system limit.	Ensure that the parameter value is within the specified range.
9299	The Deceleration setting in the MC_GearInPos instruction exceeds the setting limit.	The Deceleration setting in the MC_GearInPos instruction exceeds the setting limit.	Ensure that the parameter value is within the specified range.
9300			Ensure that the parameter value is within the specified range.

Error code	Error message	Detailed description	Solution
	The AvoidReversal setting in the MC_GearInPos instruction is out of range.	The AvoidReversal setting in the MC_GearInPos instruction is out of range.	
9301	The master axis speed is zero when the MC_GearInPos instruction is started.	The master axis speed is zero when the MC_GearInPos instruction is started.	Ensure that the master axis speed is not zero when starting this instruction.
9302	The master axis did not move during the catching phase of the MC_GearInPos instruction.	The master axis did not move during the catching phase of the MC_GearInPos instruction.	When MasterStarDistance is set to 0, ensure that the input MasterSyncPosition does not overlap with the current position of the master axis.
9303	When the MC_GearInPos instruction is started, the speed of the slave axis is not zero before entering the catching phase.	When the MC_GearInPos instruction is started, the speed of the slave axis is not zero before entering the catching phase.	Ensure that the slave axis remains stationary before entering the catching phase.
9304	Failed to enter the catching phase when the MC_GearInPos instruction is executed.	Failed to enter the catching phase when the MC_GearInPos instruction is executed.	Ensure that the master axis can enter the catching phase under the current position and motion direction conditions.
9305	The velocity of the slave axis exceeds the limit during execution of the MC_GearInPos instruction.	The velocity of the slave axis exceeds the limit during execution of the MC_GearInPos instruction.	Ensure that the parameter value is within the specified range.
9306	The cam node array specified by the MC_GenerateCamTable instruction is incorrect.	The cam node array specified by the MC_GenerateCamTable instruction is incorrect.	Ensure that the cam node array specified by the instruction is not the node array of the cam system variable.
9400	Maximum axis group quantity exceeded	The number of axis groups exceeds the maximum allowable value.	Reduce the number of axis groups in the project so that it does not exceed the maximum value.
9401	An axis in the axis group is faulty.	An axis in the axis group is faulty.	Locate the faulty axis, view the fault codes of the axis, and rectify the fault.
9402	The number of buffered interpolation instructions is greater than 8.	The number of buffered interpolation instructions is greater than 8.	Check whether the number of buffered interpolation instructions is greater than 8.

Error code	Error message	Detailed description	Solution
9403	Axis reused	An axis in the axis group is reused.	Each axis can be used in only one axis group. Check whether there is a reused axis in the axis group and replace it with an unused axis.
9404	Failed to create axis group	The x-axis or y-axis does not exist.	Check whether the x-axis and y-axis exist. An axis group consists of at least the x-axis and y-axis.
9405	Specified z-axis non-existent	The z-axis is specified in the instruction but does not exist in the configuration.	Check whether the z-axis specified in the instruction exists.
9406	Specified auxiliary axis non-existent	The auxiliary axis is specified in the instruction but does not exist in the configuration.	Check whether the auxiliary axis specified in the instruction exists.
9407	Axis group ID duplicated	The specified axis group ID has been used.	Change the axis group ID because the axis group ID must be unique.
9408	Failed to configure the axis.	Failed to configure the axis.	Check whether any axis in the axis group fails to be configured. If yes, check whether the board software and the background match.
9409	The axis ID is less than 0.	The axis ID is less than 0.	Check whether the ID of an axis in the axis group is less than 0.
9410	Axis group not released	The axis group is not released because the same MC_SetAxesGroup instruction is triggered repeatedly in a short time period.	Do not re-trigger the MC_SetAxesGroup instruction while its Busy signal output is still active.
9411	The MC_GroupStop instruction is aborted.	The MC_GroupStop instruction is aborted.	Check whether an instruction with higher priority is called while the MC_GroupStop instruction is still active.
9412	The value of CircAxes of the circular interpolation instruction is out of range.	The value of CircAxes of the circular interpolation instruction is out of range.	Check whether the value of CircAxes of the circular interpolation instruction is out of range.
9413			

Error code	Error message	Detailed description	Solution
	The value of CircMode of the circular interpolation instruction is out of range.	The value of CircMode of the circular interpolation instruction is out of range.	Check whether the value of CircMode of the circular interpolation instruction is out of range.
9414	The value of PathChoice of the circular interpolation instruction is out of range.	The value of PathChoice of the circular interpolation instruction is out of range.	Check whether the value of PathChoice of the circular interpolation instruction is out of range.
9415	The value of StopMode of the stop instruction is out of range.	The value of StopMode of the stop instruction is out of range.	Check whether the value of StopMode of the stop instruction is out of range.
9416	The x-axis is set to ring mode.	The x-axis is set to ring mode.	Do not set the motion control axis to the ring mode in an interpolation instruction.
9417	The y-axis is set to ring mode.	The y-axis is set to ring mode.	Do not set the motion control axis to the ring mode in an interpolation instruction.
9418	The z-axis is set to ring mode.	The z-axis is set to ring mode.	Do not set the motion control axis to the ring mode in an interpolation instruction.
9419	The auxiliary axis is set to ring mode.	The auxiliary axis is set to ring mode.	Do not set the motion control axis to the ring mode in an interpolation instruction.
9420	The circular interpolation instruction is triggered repeatedly.	The circular interpolation instruction is triggered repeatedly.	Do not re-trigger the same circular interpolation instruction while its Busy signal output is still active.
9421	The linear interpolation instruction is triggered repeatedly.	The linear interpolation instruction is triggered repeatedly.	Do not re-trigger the same linear interpolation instruction while its Busy signal output is still active.
9422	Failed to obtain the axis group.	Failed to obtain the axis group.	Check whether the axis group specified by GroupID has been created by calling MC_SetAxesGroup.
9423	Failed to configure the axis.	Failed to configure the axis.	Check whether an instruction is triggered when axis configuration is not completed. Check whether the

Error code	Error message	Detailed description	Solution
			communication state of all axes in the axis group is Axis ready.
9424	An axis is disabled.	An axis is disabled.	Do not call the interpolation instruction when any axis is in Disabled state.
9425	Axis in execution of single-axis motion instruction	The interpolation instruction is triggered when an axis is executing a single-axis motion instruction.	Do not call the interpolation instruction when any axis is executing single-axis motion instructions and not in StandStill state.
9426	An axis is in Stopping state.	An axis is in Stopping state.	Do not call the interpolation instruction when an axis is in Stopping state after executing the MC_Stop instruction.
9427	The axis group is in Stopping state.	The axis group is in Stopping state.	Do not call the interpolation instruction while the MC_GroupStop instruction is still active.
9428	An axis is in Homing state.	An axis is in Homing state.	Do not call the interpolation instruction when an axis is in Homing state after executing the MC_Home instruction.
9429	An axis is executing the position setting instruction.	An axis is executing the position setting instruction.	Do not call the interpolation instruction when an axis is setting the current position by executing the MC_SetPosition instruction.
9430	An axis is in commissioning state.	An axis is in commissioning state.	Do not call the interpolation instruction when any axis is in commissioning state.
9431	An axis enters the commissioning state during interpolation, which aborts instruction execution of other axes.	An axis enters the commissioning state during interpolation, which aborts instruction execution of other axes.	Check whether any axis enters the commissioning state during interpolation and aborts instruction execution of other axes.
9432	Failed to request the memory.	Failed to request the memory.	Check whether the memory runs out. Contact the manufacturer.
9433			

Error code	Error message	Detailed description	Solution
	The target velocity is 0 or less than 0.	The target velocity is 0 or less than 0.	Ensure that the target velocity of the instruction is greater than 0.
9434	The target acceleration is 0 or less than 0.	The target acceleration is 0 or less than 0.	Ensure that the target acceleration of the instruction is greater than 0.
9435	The target deceleration is 0 or less than 0.	The target deceleration is 0 or less than 0.	Ensure that the target deceleration of the instruction is greater than 0.
9436	The curve type setting is out of range.	The curve type setting is out of range.	Check whether the curve type is set to a value other than the T-shaped curve for the interpolation instruction.
9437	AbsRelMode is set incorrectly.	AbsRelMode is set incorrectly.	Check whether the parameter is set to a value other than the absolute positioning and relative positioning modes.
9438	BufferMode is set incorrectly.	BufferMode is set incorrectly.	Check whether the value of BufferMode is proper.
9439	InsertMode is set incorrectly.	InsertMode is set incorrectly.	Check whether the value of InsertMode is proper.
9440	An axis stops due to a fault.	An axis stops due to a fault.	Locate the faulty axis and rectify the fault based on the fault code.
9441	The MC_GroupStop instruction is called repeatedly.	The MC_GroupStop instruction is called repeatedly.	Do not re-trigger an MC_GroupStop instruction or call other MC_GroupStop instructions while an MC_GroupStop instruction is still active.
9442	Data buffer area not empty	The data buffer area is not empty. It is an internal fault.	Contact the manufacturer.
9443	Not a circle	No circle can be drawn due to improper parameter settings.	Update the parameter settings.
9444	Not a circle	The start, end, and border points in the circular interpolation instruction are the same point, and no circle can be drawn.	Check the input parameters of the circular interpolation instruction and ensure that the start, end, and border points can form a circle.

Error code	Error message	Detailed description	Solution
9445	The instruction buffer area is full.	The instruction buffer area is full.	Contact Inovance for technical support.
9446	The velocity of the x-axis exceeds the maximum allowable velocity.	The velocity of the x-axis exceeds the maximum allowable velocity.	Ensure that the target velocity of the x-axis is not greater than the maximum allowable velocity.
9447	The velocity of the y-axis exceeds the maximum allowable velocity.	The velocity of the y-axis exceeds the maximum allowable velocity.	Ensure that the target velocity of the y-axis is not greater than the maximum allowable velocity.
9448	Z-axis exceeded maximum velocity	Ensure that the target velocity of the z-axis is not greater than the maximum allowable velocity.	Ensure that the target velocity of the z-axis is not greater than the maximum allowable velocity.
9449	The velocity of the auxiliary axis exceeds the maximum allowable velocity.	The velocity of the auxiliary axis exceeds the maximum allowable velocity.	Ensure that the target velocity of the auxiliary axis is not greater than the maximum allowable velocity.
9450	Failed to obtain the number of axis groups.	Failed to obtain the number of axis groups.	Update the software tool to the latest version.
9451	Internal fault	Internal fault	Contact the manufacturer.
9452	The instruction is called when the axis is in StandStill state.	The instruction is called when the axis is in StandStill state.	Do not call this instruction when the axis is StandStill state.
9453	Maximum velocity exceeded	The maximum velocity specified on the axis group configuration interface is exceeded.	Check whether the target velocity of the instruction is greater than the maximum velocity specified on the axis group configuration interface.
9454	The maximum allowable acceleration (deceleration) is exceeded.	The maximum allowable acceleration (deceleration) is exceeded.	Check whether the target acceleration (deceleration) of the instruction is greater than the maximum acceleration (deceleration) specified on the axis group configuration interface.
9455	The axis group becomes faulty due to an error reported by the linear interpolation instruction.	The axis group becomes faulty due to an error reported by the linear interpolation instruction.	Identify the first linear interpolation instruction that reports the error and

Error code	Error message	Detailed description	Solution
			troubleshoot the fault based on the fault code.
9456	The axis group becomes faulty due to an error reported by the circular interpolation instruction.	The axis group becomes faulty due to an error reported by the circular interpolation instruction.	Identify the first circular interpolation instruction that reports the error and troubleshoot the fault based on the fault code.
9457	The axis group becomes faulty due to an error reported by the axis group stop instruction.	The axis group becomes faulty due to an error reported by the axis group stop instruction.	Identify the first axis group stop instruction that reports the error and troubleshoot the fault based on the fault code.
9458	The axis group becomes faulty due to an error reported by the axis group pause instruction.	The axis group becomes faulty due to an error reported by the axis group pause instruction.	Identify the first axis group pause instruction that reports the error and troubleshoot the fault based on the fault code.
9459	The x-axis in the axis group is performing the interpolation algorithm of another axis group.	The x-axis in the axis group is performing the interpolation algorithm of another axis group.	An axis can be configured in different axis groups at the same time. However, ensure that it executes the interpolation instruction of only one axis group at the same time.
9460	The y-axis in the axis group is performing the interpolation algorithm of another axis group.	The y-axis in the axis group is performing the interpolation algorithm of another axis group.	An axis can be configured in different axis groups at the same time. However, ensure that it executes the interpolation instruction of only one axis group at the same time.
9461	The z-axis in the axis group is performing the interpolation algorithm of another axis group.	The z-axis in the axis group is performing the interpolation algorithm of another axis group.	An axis can be configured in different axis groups at the same time. However, ensure that it executes the interpolation instruction of only one axis group at the same time.
9462	The auxiliary axis in the axis group is performing the interpolation algorithm of another axis group.	The auxiliary axis in the axis group is performing the interpolation algorithm of another axis group.	An axis can be configured in different axis groups at the same time. However, ensure that it executes the interpolation instruction of only one axis group at the same time.
9463			

Error code	Error message	Detailed description	Solution
	Axes in synchronous mode but not under axis group control when the MC_GroupStop instruction is called	When the MC_GroupStop instruction is called, the axes are in synchronous mode but not under axis group control, such as interpolation control or cam control.	Note that the MC_GroupStop instruction can be called only when the axes in the axis group are in synchronous mode under axis group control. Do not call the MC_GroupStop instruction when the axes enter the synchronous mode due to other instructions.
9464	Axes in synchronous mode but not under axis group control when the linear or circular interpolation instruction is called	When the linear or circular interpolation instruction is called, the axes are in synchronous mode but not under axis group control, such as interpolation control or cam control.	Note that the linear or circular interpolation instruction can be called only when the axes in the axis group are in synchronous mode under axis group control. Do not call the linear or circular interpolation instruction when the axes enter the synchronous mode due to other non-axis-group instructions.
9465	Axes in synchronous mode but not under axis group control when the MC_GroupHalt instruction is called	When the MC_GroupHalt instruction is called, the axes are in synchronous mode but not under axis group control, such as interpolation control or cam control.	Note that the MC_GroupHalt instruction can be called only when the axes in the axis group are in synchronous mode under axis group control. Do not call the MC_GroupHalt instruction when the axes enter the synchronous mode due to other instructions.
9466	NumOfTurns in the MC_MoveEllipse instruction is set incorrectly.	NumOfTurns in the MC_MoveEllipse instruction is set incorrectly.	Ensure that the parameter value is within the allowable range.
9467	AddLength in the MC_MoveEllipse instruction is set incorrectly.	AddLength in the MC_MoveEllipse instruction is set incorrectly.	Ensure that the parameter value is within the allowable range.
9468	The axis stops due to an error of the MC_MoveEllipse instruction.	The axis stops due to an error of the MC_MoveEllipse instruction.	Check the fault code of the MC_MoveEllipse instruction with an exception to locate the fault.
9469	CircAxes in the MC_MoveEllipse instruction is set incorrectly.	CircAxes in the MC_MoveEllipse instruction is set incorrectly.	Ensure that the parameter value is within the allowable range.

Error code	Error message	Detailed description	Solution
9470	CircMode in the MC_MoveEllipse instruction is set incorrectly.	CircMode in the MC_MoveEllipse instruction is set incorrectly.	Ensure that the parameter value is within the allowable range.
9471	PathChoice in the MC_MoveEllipse instruction is set incorrectly.	PathChoice in the MC_MoveEllipse instruction is set incorrectly.	Ensure that the parameter value is within the allowable range.
9472	Velocity in the MC_MoveEllipse instruction is set incorrectly.	Velocity in the MC_MoveEllipse instruction is set incorrectly.	Ensure that the parameter value is within the allowable range.
9473	Acceleration in the MC_MoveEllipse instruction is set incorrectly.	Acceleration in the MC_MoveEllipse instruction is set incorrectly.	Ensure that the parameter value is within the allowable range.
9474	Deceleration in the MC_MoveEllipse instruction is set incorrectly.	Deceleration in the MC_MoveEllipse instruction is set incorrectly.	Ensure that the parameter value is within the allowable range.
9475	BufferMode in the MC_MoveEllipse instruction is set incorrectly.	BufferMode in the MC_MoveEllipse instruction is set incorrectly.	Ensure that the parameter value is within the allowable range.
9476	Failed to draw an ellipse because the center and the lengths of the major and minor axes are set incorrectly.	Failed to draw an ellipse because the center and the lengths of the major and minor axes are set incorrectly.	Ensure that the parameter value is within the allowable range.
9477	The property of the x-axis in the axis group instruction does not support the interpolation motion.	The property of the x-axis in the axis group instruction does not support the interpolation motion.	Ensure that the x-axis is not in single-axis mode.
9478	The property of the y-axis in the axis group instruction does not support the interpolation motion.	The property of the y-axis in the axis group instruction does not support the interpolation motion.	Ensure that the y-axis is not in single-axis mode.
9479	The property of the z-axis in the axis group instruction does not support the interpolation motion.	The property of the z-axis in the axis group instruction does not support the interpolation motion.	Ensure that the z-axis is not in single-axis mode.
9480	The property of the auxiliary axis in the axis group instruction does not	The property of the auxiliary axis in the axis group	Ensure that the auxiliary axis is not in single-axis mode.

Error code	Error message	Detailed description	Solution
	support the interpolation motion.	instruction does not support the interpolation motion.	
9501	EtherCAT bus drive error	A drive error occurs. The fault code in the object dictionary 0x603F of the drive is 0xxxxx (hexadecimal).	Determine the drive fault type according to the bus drive guide and rectify the fault.
9502	The drive is disabled.	The drive is disabled.	1. Check whether the drive status word 0x6041 switches to the disabled state during motion. 2. Check whether communication is disconnected during motion.
9503	The limit is reached.	The limit is reached.	1. Check whether the software limit is configured and reached. 2. Check whether the hardware limit is reached.
9505	Failed to modify the control mode.	Failed to modify the control mode.	1. Check for interference in network communication. 2. Check whether the drive supports the object dictionary 0x6060.
9508	Homing failed	Homing failed	1. Identify the cause of the drive homing failure based on the fault code. 2. Check whether homing timed out.
9509	An axis internal calculation precision error occurs.	An axis internal calculation precision error occurs.	Check whether the floating-point data of the instruction falls beyond the single-precision floating-point number range.
9510	The following error is out of range.	The following error is out of range.	1. Check whether the acceleration is too large. 2. Check whether the set following error is too small.
9512	The servo drive is disconnected during operation.	The servo drive is disconnected during operation.	1. Check whether the drive works properly. 2. Check whether the network cable is properly connected.

Error code	Error message	Detailed description	Solution
			3. Check whether there is high-voltage interference in communication.
9513	Homing failed due to a drive fault	Homing failed due to a drive fault.	Check the fault code of the drive to eliminate the fault.
9514	Homing failed because the homing offset exceeded 32 bits.	Homing failed because the homing offset exceeded 32 bits.	Check whether the homing offset multiplied by the gear ratio exceeds 32 bits; if yes, change the gear ratio.
9515	Homing failed due to loss of the slave	Homing failed because the EtherCAT drive is lost.	Contact Inovance for technical support.
9516	Homing failed because the SDO failed to write to object dictionary 0x607C.	Homing failed because the SDO failed to write to object dictionary 0x607C.	1. Check whether the drive supports 0x607C. 2. Check the network communication quality.
9517	Homing failed because the SDO failed to write 6 into object dictionary 0x6060.	Homing failed because the SDO failed to write 6 into object dictionary 0x6060.	1. Set 0x6060 in the PDO. 2. Check the network communication quality.
9518	Homing failed because the SDO failed to read object dictionary 0x6061.	Homing failed because the SDO failed to read object dictionary 0x6061.	1. Set 0x6061 in the PDO. 2. Check the network communication quality.
9519	Homing failed because the SDO failed to write 8 into object dictionary 0x6060.	Homing failed because the SDO failed to write 8 into object dictionary 0x6060.	1. Set 0x6060 in the PDO. 2. Check the network communication quality.
9551	Failed to switch the control mode.	Failed to switch the control mode.	Check for interference in network communication.
9552	The target velocity is 0.	The target velocity is 0.	Check whether the target velocity of position instructions is appropriate.
9601	The axis stops due to parameter exception of the MC_MoveAbsolute instruction.	The axis stops due to parameter exception of the MC_MoveAbsolute instruction.	Check the instruction that reports the error, and further determine the fault based on the fault code of the instruction.
9602	The axis stops due to parameter exception of the	The axis stops due to parameter exception of the MC_MoveRelative instruction.	Check the instruction that reports the error, and further determine

Error code	Error message	Detailed description	Solution
	MC_MoveRelative instruction.		the fault based on the fault code of the instruction.
9603	The axis stops due to exception of the MC_MoveVelocity instruction.	The axis stops due to exception of the MC_MoveVelocity instruction.	Check the instruction that reports the error, and further determine the fault based on the fault code of the instruction.
9604	The axis stops due to exception of the MC_Jog instruction.	The axis stops due to exception of the MC_Jog instruction.	Check the instruction that reports the error, and further determine the fault based on the fault code of the instruction.
9605	The axis stops due to exception of the MC_MoveVelocityCSV instruction.	The axis stops due to exception of the MC_MoveVelocityCSV instruction.	Check the instruction that reports the error, and further determine the fault based on the fault code of the instruction.
9606	The axis stops due to exception of the MC_MoveBuffer instruction.	The axis stops due to exception of the MC_MoveBuffer instruction.	Check the instruction that reports the error, and further determine the fault based on the fault code of the instruction.
9607	The axis stops due to parameter exception of the MC_MoveFeed instruction.	The axis stops due to parameter exception of the MC_MoveFeed instruction.	Check the instruction that reports the error, and further determine the fault based on the fault code of the instruction.
9608	The axis stops due to parameter exception of the MC_Stop instruction.	The axis stops due to parameter exception of the MC_Stop instruction.	Check the instruction that reports the error, and further determine the fault based on the fault code of the instruction.
9609	The axis stops due to parameter exception of the MC_MoveTorque instruction.	The axis stops due to parameter exception of the MC_MoveTorque instruction.	Check the instruction that reports the error, and further determine the fault based on the fault code of the instruction.
9610	The axis stops due to parameter exception of the MC_Halt instruction.	The axis stops due to parameter exception of the MC_Halt instruction.	Check the instruction that reports the error, and further determine the fault based on the fault code of the instruction.
9611	The axis stops due to parameter exception of the MC_MoveSuperImposed instruction.	The axis stops due to parameter exception of the MC_MoveSuperImposed instruction.	Check the instruction that reports the error, and further determine the fault based on the fault code of the instruction.

Error code	Error message	Detailed description	Solution
9612	The axis stops due to an error reported by the MC_SyncMoveVelocity instruction.	The axis stops due to an error reported by the MC_SyncMoveVelocity instruction.	Check the instruction that reports the error, and further determine the fault based on the fault code of the instruction.
9613	The axis stops due to an error reported by the MC_SyncTorqueControl instruction.	The axis stops due to an error reported by the MC_SyncTorqueControl instruction.	Check the instruction that reports the error, and further determine the fault based on the fault code of the instruction.
9614	The axis stops due to an error reported by the MC_FollowVelocity instruction.	The axis stops due to an error reported by the MC_FollowVelocity instruction.	Check the instruction that reports the error, and further determine the fault based on the fault code of the instruction.
9616	The axis stops due to parameter exception of the MC_MoveAbsRelPP instruction.	The axis stops due to parameter exception of the MC_MoveAbsRelPP instruction.	Check the instruction that reports the error, and further determine the fault based on the fault code of the instruction.
9617	The axis stops because the SDO failed to write to object dictionary 0x6060.	The axis stops because the SDO failed to write to object dictionary 0x6060.	1. Set 0x6060 in the PDO. 2. Check the network communication quality.
9618	The axis stops because the SDO failed to read object dictionary 0x6061.	The axis stops because the SDO failed to read object dictionary 0x6061.	1. Set 0x6061 in the PDO. 2. Check the network communication quality.
9619	Number of pulses of target movement amount in profile position mode exceeding the limit	The number of pulses of target movement amount in profile position mode exceeds the limit.	Reduce the number of pulses per revolution of the motor/encoder or modify the gear ratio.
9620	The axis stops due to parameter exception of the MC_MoveVelocityPV instruction.	The axis stops due to parameter exception of the MC_MoveVelocityPV instruction.	Check the instruction that reports the error, and further determine the fault based on the fault code of the instruction.
9621	The axis stops due to an error reported by the MC_JogP instruction.	The axis stops due to an error reported by the MC_JogP instruction.	Check the instruction that reports the error, and further determine the fault based on the fault code of the instruction.
9622	The axis stops due to parameter exception of the MC_MoveBufferPP instruction.	The axis stops due to parameter exception of the MC_MoveBufferPP instruction.	Check the instruction that reports the error, and further determine the fault based on the fault code of the instruction.

Error code	Error message	Detailed description	Solution
9623	The axis stops due to parameter exception of the MC_TorqueControlPT instruction.	The axis stops due to parameter exception of the MC_TorqueControlPT instruction.	Check the instruction that reports the error, and further determine the fault based on the fault code of the instruction.
9159	The profile velocity 0x6081 is not mapped to the bus driver.	The profile velocity 0x6081 is not mapped to the bus driver.	Add 0x6081 in the PDO and map it to the axis.
9160	The profile acceleration 0x6083 is not mapped to the bus driver.	The profile acceleration 0x6083 is not mapped to the bus driver.	Add 0x6083 in the PDO and map it to the axis.
9161	The profile deceleration 0x6084 is not mapped to the bus driver.	The profile deceleration 0x6084 is not mapped to the bus driver.	Add 0x6084 in the PDO and map it to the axis.
9162	The target velocity of the MC_Jog instruction in the jog mode exceeds the maximum jogging velocity.	The target velocity of the MC_Jog instruction in the jog mode exceeds the maximum jogging velocity.	Check whether the target velocity of the MC_Jog instruction in the jog mode exceeds the maximum jogging velocity.
9163	The forward and reverse motion flags of the MC_JogP instruction are both active.	The forward and reverse motion flags of the MC_JogP instruction are both active.	Check whether the forward and reverse motion flags of the MC_JogP instruction are both active.
9164	The actual feedback speed 0x606C is not mapped to the bus driver.	The actual feedback speed 0x606C is not mapped to the bus driver.	Add 0x606C in the PDO and map it to the axis.
9165	The target action is cached in the current buffer.	The target action is cached in the current buffer.	Re-activate the new target action after the target action is executed in the buffer.
9166	The torque ramp 0x6087 is not mapped to the bus driver.	The torque ramp 0x6087 is not mapped to the bus driver.	Add 0x6087 in the PDO and map it to the axis.
9167	The velocity limit of the profile torque instruction exceeds the maximum velocity specified on the axis configuration interface.	The velocity limit of the profile torque instruction exceeds the maximum velocity specified on the axis configuration interface.	Check whether the velocity limit of the profile torque instruction exceeds the maximum velocity specified on the axis configuration interface.

Error code	Error message	Detailed description	Solution
9701	The encoder axis instruction failed to request the memory.	The encoder axis instruction failed to request the memory.	1. Check whether the PLC memory runs out. 2. Contact the manufacturer.
9702	1. The encoder axis type is incorrect. 2. The requested encoder axis does not exist. 3. The instruction is not supported in offline commissioning.	1. The encoder axis type is incorrect. 2. The requested encoder axis does not exist. 3. The instruction is not supported in offline commissioning.	1. The instruction does not support the set axis type. Check whether the axis type setting is incorrect. 2. The instruction is not supported in offline commissioning.
9703	Failed to configure the axis.	Failed to configure the axis.	Check whether the board software and the software tool match.
9704	Counter operation command is not configured in I/O mapping of the encoder axis.	Counter operation command is not configured in I/O mapping of the encoder axis.	Configure Counter operation command in I/O mapping of the encoder axis.
9705	Counter status is not configured in I/O mapping of the encoder axis.	Counter status is not configured in I/O mapping of the encoder axis.	Configure Counter status in I/O mapping of the encoder axis.
9706	Encoder present position is not configured in I/O mapping of the encoder axis.	Encoder present position is not configured in I/O mapping of the encoder axis.	Configure Encoder present position in I/O mapping of the encoder axis.
9707	Pulse rate is not configured in I/O mapping of the encoder axis.	Pulse rate is not configured in I/O mapping of the encoder axis.	Configure Pulse rate in I/O mapping of the encoder axis.
9708	The positive limit of the encoder axis is not greater than the negative limit.	The positive limit of the encoder axis is not greater than the negative limit.	Ensure that the positive limit of the encoder axis is greater than the negative limit.
9709	The positive limit of the encoder axis is greater than 2147483647 after being converted into the pulse unit.	The positive limit of the encoder axis is greater than 2147483647 after being converted into the pulse unit.	Ensure that the positive limit of the encoder axis is less than or equal to 2147483647 after being converted into the pulse unit.
9710	The negative limit of the encoder axis is less than -2147483648 after being	The negative limit of the encoder axis is less than -	Ensure that the negative limit of the encoder axis is greater than or equal to -2147483648 after

Error code	Error message	Detailed description	Solution
	converted into the pulse unit.	2147483648 after being converted into the pulse unit.	being converted into the pulse unit.
9711	The revolution cycle of the encoder axis in ring mode is greater than 2147483647 after being converted into the pulse unit.	The revolution cycle of the encoder axis in ring mode is greater than 2147483647 after being converted into the pulse unit.	Ensure that the revolution cycle of the encoder axis in ring mode is less than or equal to 2147483647 after being converted into the pulse unit.
9712	The encoder axis is changed while the ENC_Counter instruction is still active.	The encoder axis is changed while the ENC_Counter instruction is still active.	Do not change the encoder axis while the ENC_Counter instruction is still active.
9713	GR10-2HCE module fault	GR10-2HCE module fault	Check the fault code object dictionary of the GR10-2HCE module and troubleshoot the fault according to the fault code.
9714	Failed to reset the encoder axis fault.	Failed to reset the encoder axis fault.	1. The current fault of the encoder axis does not support reset. 2. The encoder shaft enters the faulty state immediately after the fault is reset. Check the axis fault codes and slave fault codes to further determine the fault type.
9715	The ENC_Reset instruction is called when the encoder axis is not faulty.	The ENC_Reset instruction is called when the encoder axis is not faulty.	Do not call the ENC_Reset instruction when the encoder axis is not faulty.
9716	The value of TriggerMode of the ENC_Preset instruction is out of range.	The value of TriggerMode of the ENC_Preset instruction is out of range.	Ensure that the parameter value is within the allowable range.
9717	Position of the ENC_Preset instruction is set to a value greater than 9999999.	Position of the ENC_Preset instruction is set to a value greater than 9999999.	Set Position of the ENC_Preset instruction to a value less than or equal to 9999999.
9718	Physical output command is not configured in I/O mapping of the encoder axis.	Physical output command is not configured in I/O mapping of the encoder axis.	Configure Physical output command in I/O mapping of the encoder axis.
9719	The preset position or comparison output position of the encoder axis	The preset position or comparison output position of	Ensure that the preset position or comparison output position of the encoder axis instruction is

Error code	Error message	Detailed description	Solution
	instruction is greater than the positive limit.	the encoder axis instruction is greater than the positive limit.	less than or equal to the positive limit.
9720	The preset position or comparison output position of the encoder axis instruction is less than the negative limit.	The preset position or comparison output position of the encoder axis instruction is less than the negative limit.	Ensure that the preset position or comparison output position of the encoder axis instruction is greater than or equal to the negative limit.
9721	The preset position or comparison output position of the encoder axis instruction is greater than 2147483647 or less than -2147483648 after being converted into the pulse unit.	The preset position or comparison output position of the encoder axis instruction is greater than 2147483647 or less than -2147483648 after being converted into the pulse unit.	Ensure that the preset position or comparison output position of the encoder axis instruction is between -2147483648 and +2147483647 after being converted into the pulse unit.
9722	The preset position or comparison output position of the encoder axis instruction is greater than or equal to the revolution cycle in ring mode.	The preset position or comparison output position of the encoder axis instruction is greater than or equal to the revolution cycle in ring mode.	Ensure that the preset position or comparison output position of the encoder axis instruction is less than the revolution cycle in ring mode.
9723	The value of ProbeID of the ENC_TouchProbe instruction is out of range.	The value of ProbeID of the ENC_TouchProbe instruction is out of range.	Ensure that the parameter value is within the allowable range.
9724	The value of TriggerEdge of the ENC_TouchProbe instruction is out of range.	The value of TriggerEdge of the ENC_TouchProbe instruction is out of range.	Ensure that the parameter value is within the allowable range.
9725	The value of TerminalSource of the ENC_TouchProbe instruction is out of range.	The value of TerminalSource of the ENC_TouchProbe instruction is out of range.	Ensure that the parameter value is within the allowable range.
9726	The value of TriggerMode of the ENC_TouchProbe instruction is out of range.	The value of TriggerMode of the ENC_TouchProbe instruction is out of range.	Ensure that the parameter value is within the allowable range.
9727	The probe status word is not associated in I/O mapping of the encoder axis.	The probe status word is not associated in I/O mapping of the encoder axis.	Ensure that the probe status word is associated in I/O mapping of the encoder axis.
9728			

Error code	Error message	Detailed description	Solution
	The probe feedback position is not associated in I/O mapping of the encoder axis.	The probe feedback position is not associated in I/O mapping of the encoder axis.	Ensure that the probe feedback position is associated in I/O mapping of the encoder axis.
9729	The control word is not associated in I/O mapping of the encoder axis.	The control word is not associated in I/O mapping of the encoder axis.	Ensure that the control word is associated in I/O mapping of the encoder axis.
9730	The probe window function of the encoder axis is enabled, but the start position of the window is not less than the end position.	The probe window function of the encoder axis is enabled, but the start position of the window is not less than the end position.	Ensure that the start position of the probe window is less than the end position.
9731	The Xn0 terminal is not assigned with the probe function.	The Xn0 terminal is not assigned with the probe function.	Assign the probe function to the Xn0 terminal.
9732	The Xn1 terminal is not assigned with the probe function.	The Xn1 terminal is not assigned with the probe function.	Assign the probe function to the Xn1 terminal.
9742	Compare mode is not configured in I/O mapping of the encoder axis.	Compare mode is not configured in I/O mapping of the encoder axis.	Configure Compare mode in I/O mapping of the encoder axis.
9743	Compare pulse/time is not configured in I/O mapping of the encoder axis.	Compare pulse/time is not configured in I/O mapping of the encoder axis.	Configure Compare pulse/time in I/O mapping of the encoder axis.
9744	Compare size/step is not configured in I/O mapping of the encoder axis.	Compare size/step is not configured in I/O mapping of the encoder axis.	Configure Compare size/step in I/O mapping of the encoder axis.
9745	Compare point value 1 is not configured in I/O mapping of the encoder axis.	Compare point value 1 is not configured in I/O mapping of the encoder axis.	Configure Compare point value 1 in I/O mapping of the encoder axis.
9746	Compare point value 2 is not configured in I/O mapping of the encoder axis.	Compare point value 2 is not configured in I/O mapping of the encoder axis.	Configure Compare point value 2 in I/O mapping of the encoder axis.
9747			

Error code	Error message	Detailed description	Solution
	Physical output status is not configured in I/O mapping of the encoder axis.	Physical output status is not configured in I/O mapping of the encoder axis.	Configure Physical output status in I/O mapping of the encoder axis.
9748	Compare error code is not configured in I/O mapping of the encoder axis.	Compare error code is not configured in I/O mapping of the encoder axis.	Configure Compare error code in I/O mapping of the encoder axis.
9749	Current compare number/ position is not configured in I/O mapping of the encoder axis.	Current compare number/ position is not configured in I/O mapping of the encoder axis.	Configure Current compare number/position in I/O mapping of the encoder axis.
9750	Failed to obtain the start address of the array of the single-axis array comparison output instruction.	Failed to obtain the start address of the array of the single-axis array comparison output instruction.	1. Check whether the PLC memory is sufficient. 2. Check whether the background and board software match. 3. Check whether the array of the instruction is out of bounds.
9751	Failed to obtain the start address of the axis group of the axis group array comparison output instruction.	Failed to obtain the start address of the axis group of the axis group array comparison output instruction.	1. Check whether the PLC memory is sufficient. 2. Check whether the background and board software match. 3. Check whether the array of the instruction is out of bounds.
9752	The bus encoder axis is not associated with any slave.	The bus encoder axis is not associated with any slave.	Associate the bus encoder axis with a slave.
9753	The x-axis and y-axis of the axis group array comparison instruction are not associated with the same slave.	The x-axis and y-axis of the axis group array comparison instruction are not associated with the same slave.	Associate the x-axis and y-axis of the axis group comparison output instruction with the same slave.
9754	The x-axis of the axis group array comparison instruction is not associated with the first channel of the slave.	The x-axis of the axis group array comparison instruction is not associated with the first channel of the slave.	Associate the x-axis of the axis group comparison output instruction with the first channel of the slave.
9755	The y-axis of the axis group array comparison	The y-axis of the axis group array comparison instruction is	Associate the y-axis of the axis group comparison output

Error code	Error message	Detailed description	Solution
	instruction is not associated with the second channel of the slave.	not associated with the second channel of the slave.	instruction with the second channel of the slave.
9756	The Yn0 terminal is not assigned with the one-dimensional comparison output function.	The Yn0 terminal is not assigned with the one-dimensional comparison output function.	Assign the one-dimensional comparison output function to the Yn0 output terminal corresponding to the channel.
9757	The absolute value of the start value of the encoder axis step comparison output instruction is greater than 9999999.	The absolute value of the start value of the encoder axis step comparison output instruction is greater than 9999999.	Ensure that the absolute value of the floating-point number in the motion instruction does not exceed 9999999.
9758	The absolute value of the end value of the encoder axis step comparison output instruction is greater than 9999999.	The absolute value of the end value of the encoder axis step comparison output instruction is greater than 9999999.	Ensure that the absolute value of the floating-point number in the motion instruction does not exceed 9999999.
9759	The absolute value of the step of the encoder axis step comparison output instruction is greater than 9999999.	The absolute value of the step of the encoder axis step comparison output instruction is greater than 9999999.	Ensure that the absolute value of the floating-point number in the motion instruction does not exceed 9999999.
9760	The absolute value of Parameter of the encoder axis step comparison output instruction is greater than 9999999.	The absolute value of Parameter of the encoder axis step comparison output instruction is greater than 9999999.	Ensure that the absolute value of the floating-point number in the motion instruction does not exceed 9999999.
9761	The value of Mode of the encoder axis comparison output instruction is out of range.	The value of Mode of the encoder axis comparison output instruction is out of range.	Ensure that the parameter value is within the allowable range.
9762	The time for time control of the encoder axis comparison output is out of range.	The time for time control of the encoder axis comparison output is out of range.	Ensure that the parameter value is within the allowable range.
9763	The step of the encoder axis step comparison output instruction is 0.	The step of the encoder axis step comparison output instruction is 0.	Set the step of the step comparison output instruction to a value other than 0.
9764			

Error code	Error message	Detailed description	Solution
	The start position of the step comparison output instruction of the encoder axis is equal to the end position.	The start position of the step comparison output instruction of the encoder axis is equal to the end position.	Ensure that the start position of the step comparison output instruction is not equal to the end position.
9765	The start position of the step comparison output instruction of the encoder axis is less than the end position, but the step is negative.	The start position of the step comparison output instruction of the encoder axis is less than the end position, but the step is negative.	Set the step to a positive value.
9766	The start position of the step comparison output instruction of the encoder axis is greater than the end position, but the step is positive.	The start position of the step comparison output instruction of the encoder axis is greater than the end position, but the step is positive.	Set the step to a negative value.
9767	The value of Size of the encoder axis array comparison output instruction is out of range.	The value of Size of the encoder axis array comparison output instruction is out of range.	Ensure that the parameter value is within the allowable range.
9768	The absolute value of the target position of the encoder axis array comparison output instruction is greater than 9999999.	The absolute value of the target position of the encoder axis array comparison output instruction is greater than 9999999.	Ensure that the absolute value of the floating-point number in the motion instruction does not exceed 9999999.
9769	The axis is performing one-dimensional comparison output and must not be aborted by a two-dimensional comparison output instruction.	The axis is performing one-dimensional comparison output and must not be aborted by a two-dimensional comparison output instruction.	Wait for the one-dimensional comparison output to complete or stop the one-dimensional comparison output before executing the two-dimensional comparison output instruction.
9770	The EtherCAT slave is disconnected during operation.	The EtherCAT slave is disconnected during operation.	Check whether the EtherCAT slave is disconnected during operation.
9771	The bus encoder axis is in offline commissioning mode.	The bus encoder axis is in offline commissioning mode.	The bus encoder axis does not support the offline commissioning mode.
9772			

Error code	Error message	Detailed description	Solution
	The DI terminal is not assigned with the preset position function.	The DI terminal is not assigned with the preset position function.	Assign the preset position function to the DI terminal before calling the preset position instruction.
9773	The value of Parameter in the comparison instruction is out of range when the pulse output mode is selected.	The value of Parameter in the comparison instruction is out of range when the pulse output mode is selected.	Do not set Parameter to 0 or a negative value when the pulse output mode is selected in the comparison instruction.
9774	The 2HCE module fails when the comparison output instruction is called.	The 2HCE module fails when the comparison output instruction is called.	1. Ensure that the input parameters are within the allowable range. 2. Check whether I/O mapping of the encoder axis is manually modified and whether it meets the I/O mapping configuration requirements of the comparison output instruction.
9775	The set position in ring mode is less than 0.	The set position in ring mode is less than 0.	Set the position in ring mode to a value greater than or equal to 0.
9776	The Y00 terminal is not assigned with the two-dimensional comparison output function.	The Y00 terminal is not assigned with the two-dimensional comparison output function.	Assign the two-dimensional comparison output function to the Y00 output terminal corresponding to the channel.
9777	The axis is performing two-dimensional comparison output and cannot be aborted by a one-dimensional comparison output instruction.	The axis is performing two-dimensional comparison output and cannot be aborted by a one-dimensional comparison output instruction.	Wait for the two-dimensional comparison output to complete or stop the two-dimensional comparison output before calling the one-dimensional comparison output instruction.
9800	Failed to read the number of motion control axes.	Failed to read the number of motion control axes.	Change the background version.
9801	The number of motion control axes is out of range.	The number of motion control axes is out of range.	Reduce the number of axes since the H5U supports at most 32 axes.
9802	The axis failed to request internal storage space.	The axis failed to request internal storage space.	1. Check whether the memory runs out. 2. Contact the manufacturer.

Error code	Error message	Detailed description	Solution
9803	Failed to obtain axis parameters.	Failed to obtain axis parameters.	Change the background version.
9804	Failed to obtain the slave.	Failed to obtain the slave.	Change the background version.
9805	Failed to obtain the system variable.	Failed to obtain the system variable.	1. Check whether the memory runs out. 2. Return the machine to the manufacturer for analysis.
9806	Parameters related to the gear ratio are set improperly.	Parameters related to the gear ratio are set improperly.	1. Ensure that the numerator and denominator of the gear ratio are greater than 0. 2. Ensure that the number of pulses per revolution of the motor/encoder is greater than 0. 3. Ensure that the displacement per revolution of the rotary table is between 0.000001 and 9999999.
9807	The software limiting parameters are set improperly.	The software limiting parameters are set improperly.	1. Ensure that the positive limit is not greater than 9999999. 2. Ensure that the negative limit is not greater than 9999999. 3. Ensure that the negative limit is not greater than the positive limit.
9808	The linear/rotary mode parameter is set improperly.	The linear/rotary mode parameter is set improperly.	Note that only the linear mode and rotary mode are supported.
9809	The revolution cycle is set improperly.	The revolution cycle is set improperly.	Ensure that the revolution cycle is between 0.01 and 9999999.
9810	The encoder mode is set improperly.	The encoder mode is set improperly.	Ensure that the encoder mode is set properly. Note that only the incremental mode and absolute value mode are supported.
9811	The homing parameter is set improperly.	The homing parameter is set improperly.	1. Do not modify the homing mode of the bus servo axis. If you want to modify the homing mode of the bus servo axis, write to the SDO. 2. Check whether the homing mode is set properly. Note that

Error code	Error message	Detailed description	Solution
			only the values 17 to 30 and 35 are supported.
9812	The Modbus address setting of the limit, home, or probe terminal is out of range.	The Modbus address setting of the limit, home, or probe terminal is out of range.	1. Check whether the set address is out of the range of Modbus addresses. 2. Select an address among X0 to X7 for the home signal. 3. Select an address among X0 to X7 for the probe signal.
9813	The pulse output mode of the local pulse axis is set improperly.	The pulse output mode of the local pulse axis is set improperly.	Check whether the pulse output mode of the local pulse axis is set improperly.
9814	The limiting deceleration is set improperly.	The limiting deceleration is set improperly.	Ensure that the limiting deceleration is between 0.0001 and 9999999.
9815	The deceleration upon axis fault is set improperly.	The deceleration upon axis fault is set improperly.	Ensure that the maximum velocity is between 0.0001 and 999999.
9816	The maximum velocity is set improperly.	The maximum velocity is set improperly.	Ensure that the maximum velocity is between 0.0001 and 999999.
9817	The maximum positive torque is set improperly.	The maximum positive torque is set improperly.	Ensure that the maximum positive torque is between 1 and 65534.
9818	The maximum negative torque is set improperly.	The maximum negative torque is set improperly.	Ensure that the maximum negative torque is between 1 and 65534.
9819	The maximum jogging velocity is set improperly.	The maximum jogging velocity is set improperly.	Ensure that the maximum jogging velocity is between 0.0001 and the maximum velocity.
9820	The maximum acceleration is set improperly.	The maximum acceleration is set improperly.	Ensure that the maximum acceleration is between 0.0001 and 9999999.
9821	The following error threshold is set improperly.	The following error threshold is set improperly.	Ensure that the following error threshold is between 0.0001 and 9999999.

Error code	Error message	Detailed description	Solution
9822	The velocity reach threshold is set improperly.	The velocity reach threshold is set improperly.	Ensure that the velocity reach threshold is between 0.0001 and 9999999.
9823	The homing velocity is set improperly.	The homing velocity is set improperly.	1. Ensure that the homing velocity is between 0.0001 and 9999999. 2. Ensure that the homing velocity is not greater than the maximum velocity. 3. Ensure that the value obtained by multiplying the homing velocity by the gear ratio is between 1 and 2148483647.
9824	The homing approach velocity is set improperly.	The homing approach velocity is set improperly.	1. Ensure that the homing approach velocity is between 0.0001 and 9999999. 2. Ensure that the homing approach velocity is not greater than the maximum velocity. 3. Ensure that the value obtained by multiplying the homing approach velocity by the gear ratio is between 1 and 2148483647. 4. Ensure that the homing approach velocity is lower than the homing velocity.
9825	The homing position mode setting is out of range.	The homing position mode setting is out of range.	Ensure that the parameter value is within the allowable range.
9826	The homing acceleration setting is improper.	The homing acceleration setting is improper.	1. Ensure that the homing acceleration is between 0.0001 and 9999999. 2. Ensure that the homing acceleration is not greater than the maximum acceleration.
9827	The homing timeout time is out of range.	The homing timeout time is out of range.	Ensure that the homing timeout time is greater than or equal to 1.

26 FAQ

26.1 FAQ for AutoShop Installation

- **When you install AutoShop, the system prompts that the software cannot be installed because SWbemObjectSet is invalid.**

  1. Symptoms
     When you install AutoShop, the system prompts that the software cannot be installed because SWbemObjectSet is invalid.
  2. Cause Analysis
     The .Net framework is incomplete or the WMI control is damaged.
  3. Solution
     a. On the PC, choose "Control Panel" > "Programs and Functions", and check whether the VC library and .NetFramework library are complete. If any library is missing, download the library from the Microsoft official website and install the library.Reinstall the AutoShop software.
     b. If the problem still persists, check whether the WMI is correct.
     c. On the PC, open the "RUN" command line, input "wmimgmt.msc", and click "OK".
     d. In the WMI manager, right-click "WMI Control" and select "Properties". If the following information is displayed, the WMI is damaged.
     e. Use the admin permission to run the MOF.bat file in the Tools folder under the AutoShop installation directory and view the control properties again. If the following information is displayed, the WMI is fixed.
     f. Reinstall the AutoShop software.

- **AutoShop installation failed because the ordinal number cannot be found.**
  - Symptoms
    AutoShop installation failed because the ordinal number cannot be found.
  - Cause Analysis
    The system DLL is damaged.
  - Solution
    - Fix the system DLL use tools such as 360 or Kingsoft Guards.
    - Restart the PC and reinstall the AutoShop software.
    - Restart the PC and reinstall the AutoShop software.

- **Errors such as MoveFile failure are prompted during AutoShop installation.**
  - Symptoms
    Errors such as MoveFile failure are prompted during AutoShop installation, as shown in the following figure.
  - Cause Analysis
    The current user does not have the permission to operate the AutoShop installation directory.
  - Solution
    - Get the admin permission.
    - Disable 360 and other safety software.
    - Reinstall the AutoShop software.

26.2 FAQ for AutoShop Usage

- **The AutoShop page has garbled characters.**
  - **Symptoms**
    The page has garbled characters after AutoShop is opened.
  - **Cause Analysis**
    A Chinese language packet is installed, but the system language is not set to Chinese.
  - **Solution**
    - In a Windows 7 system, in the control panel, choose "Time, Language, and Area" > "Area and Language" > "Management" to change the "Current language used in non-Unicode programs" to Chinese.
    - In a Windows 10 system, on the Windows Settings page, choose "Time and Languages" > "Areas" > "Other Dates, Time and Areas Settings" > "Areas" > "Management", and change the "Current language used in non-Unicode programs" to Chinese.
- **The system promotes "The operation you want to perform is not supported" when you open the applicable table control, or you cannot create a project.**
  - **Symptoms**
    The system promotes "The operation you want to perform is not supported" when you open the applicable table control, or you cannot create a project.
  - **Cause Analysis**
    AutoShop uses the vsFlex8l.ocx table control, which is applied to function modules such as variable table, monitoring table, quick monitor table, FB, and FC structure. When this problem occurs, a common cause is that the table control is not registered.
  - **Solution**
    When installing AutoShop, the Reg.bat batch instruction will be called, and the function of this batch instruction is to call the command line regsvr32+control path to register the control. Restricted by computer permissions, the table registration may be failed, so you need to register the control manually.

    Manually register the control in any of the following ways.
    - Method 1: In the installation path of AutoShop, locate Reg.bat, and run it as an admin. If this failed, edit Reg.bat, delete "/s" in the script, and run it again to view the failure causes.
    - Method 2: Place the vsFlex8l.ocx file in the installation path to C:\Windows\System32. Run cmd as an admin, input the command regsvr32 C:\Windows\System32\vsFlex8l.ocx, and then press Enter.
      Open the "Search" menu.

      Run cmd as an admin.

      Input the command regsvr32 C:\Windows\System32\vsFlex8l.ocx.
    - Method 3: Place the vsFlex8l.ocx file in the installation path to C:\Windows\SysWOW64. Run cmd as an admin, input the command regsvr32 C:\Windows\ SysWOW64\vsFlex8l.ocx, and then press Enter.
- **The system prompts that the user program verification results are inconsistent in offline commissioning state.**
  - **Symptoms**

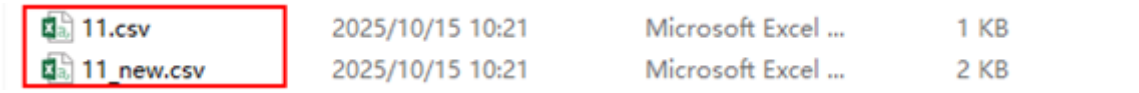
The system prompts that the user program verification results are inconsistent in offline commissioning state.

- **Cause Analysis**
  The option "Check the program before monitoring" is checked but "Download the program after starting offline commissioning" is not selected. As a result, the project in the offline debugger is different from the current project, and the verification fails.
- **Solution**
  In the menu bar of AutoShop, choose "Tools" > "System Options", and select "Download the program after starting offline commissioning". If you do not want to download the program after starting offline commissioning, deselect "Check the program before monitoring".

◦ **Two CSV files are exported when you click "Export".**

> NOTE
>
> The software of AutoShop V4.11.0.0 and later supports Unicode encoding.

- **Symptoms**
  After you right-click "Export HMI Monitoring Variables Table" in the variable table area, two CSV files are exported.

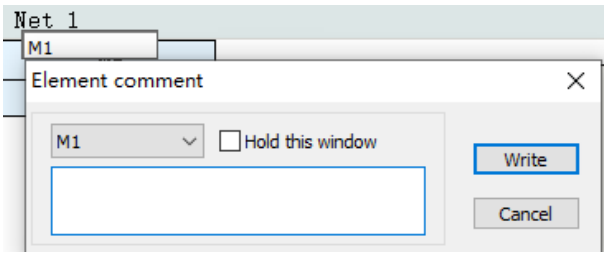


> CAUTION
>
> Several export functions involve this symptom, such as global variable table export, function block (FB) export, and function (FC) export.

- **Possible Cause**
  Version V4.11.0.0 uses Unicode encoding, unifying the character encoding format for both Chinese and English texts. To ensure the export function works correctly across both old and new versions, executing the export function will generate two files:
  - _new.csv: Encoded in Unicode, compatible with versions 4.11.0.0 and later.
  - .csv: Encoded in GB2312, compatible with HMI import and all versions.
- **Solution**
  Select the .csv file from the two exported CSV files for use.

◦ **During programming, after you edit any soft element/variable, the "Element comment" window pops up.**

- **Symptom**
  During programming, after you double-click any variable name, the "Element comment" window pops up.



- **Possible Cause**

  Clicking the soft element/variable comment switch activates the comment function, requiring comments to be edited when double-clicking any variable name.

- **Solution**

  Click the soft element/variable comment switch again to turn it off.

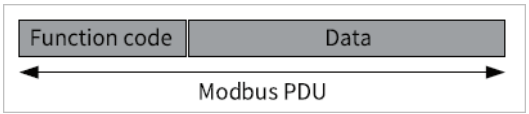
27 Appendix

27.1 Modbus Protocol

27.1.1 Description of Modbus Message

Introduction

The Modbus application protocol defines a simple protocol data unit (PDU) independent of the underlying communication layers.



The mapping of Modbus protocol on different buses or networks can introduce some additional fields on the protocol data unit. The client initiating a Modbus transaction builds a Modbus PDU and then adds an additional field to build an appropriate communication PDU.

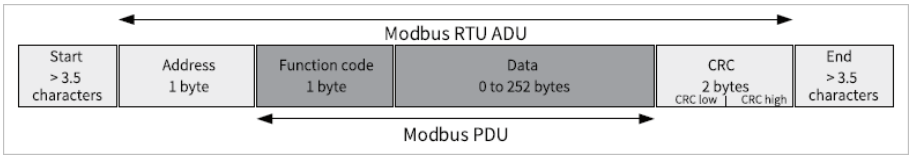
Data encoding

Modbus uses a "big-Endian" representation for addresses and data items. This means that when multiple bytes are transmitted, the most significant bit is sent first. Example:

Register Size	Value	Definition
16-bit	0x1234	The first byte sent is 0x12, then 0x34.

27.1.2 Modbus RTU Message Frame

When a device communicates over a Modbus serial link in the Remote Terminal Unit (RTU) mode, each 8-bit byte in the message consists of two 4-bit hexadecimal characters. The RTU mode features high data density, leading to higher throughput than the ASCII mode at the same baud rate. However, each message must be transmitted as a continuous character stream.



27.1.3 Modbus ASCII Message Frame

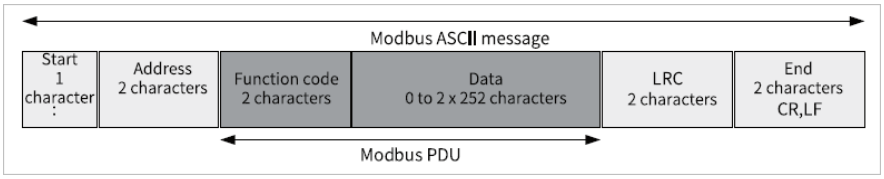
When a device on a Modbus serial link is configured to communicate in the American Standard Code for Information Interchange (ASCII) mode, each 8-bit byte in the message is sent as two ASCII characters. This mode is used when a communication link or device cannot comply with the timing management of the RTU mode.

NOTE

This mode is less efficient than the RTU mode because one byte requires two characters.

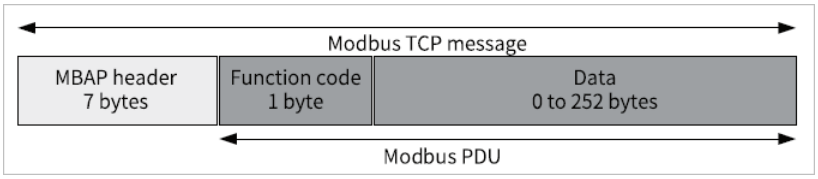
For example, the byte 0x5B is encoded as two characters: 0x35 and 0x42 (0x35 = "5" and 0x42 = "B" in ASCII encoding).

In the ASCII mode, a message is delimited by special characters at the beginning and end of the frame. A message must start with a colon (:) (ASCII hexadecimal 3A) and end with a carriage return-line feed (CR LF) pair (ASCII hexadecimal 0D and 0A).



27.1.4 Modbus TCP Message Frame

The following figure illustrates the encapsulation of Modbus requests or responses in a Modbus-TCP/IP network.



TCP/IP uses a special message header to identify the Modbus application data unit. Such a message header is called the Modbus Protocol Application Header (MBAP).

In the MBAP header, a single-byte unit identifier is used instead of the Modbus slave address field that is commonly used on a Modbus serial link. The unit identifier is used for communication between devices (such as bridges, routers, and gateways) that support multiple independent Modbus terminal units through a single IP address.

The MBAP header includes the following fields.

Field	Length	Description	Client	Server
Transaction meta identifier	2 bytes	Identifier of the Modbus request/response transaction being processed	Started by the client	Copied by the server from the received request
Protocol identifier	2 bytes	0: Modbus protocol	Started by the client	Copied by the server from the received request
Length	2 bytes	Number of bytes of the next field	Started by the client (request)	Started by the server (response)
Unit identifier	1 byte	Identifier of the remote slave connected on a serial link or other bus	Started by the client	Copied by the server from the received request

- The header contains 7 bytes.
- Transaction processing identifier: used to pair transaction processing. In the response, the Modbus server copies the transaction identifier from the request.

- Protocol identifier: used for multiplexing within the system. The value 0 indicates the Modbus protocol.
- Length: indicates the number of bytes of the next field, including the unit identifier and data field.
- Unit identifier: used for routing within the system. This field is specifically used for communication with Modbus or Modbus+ serial link slaves through gateways between Ethernet TCP-IP networks and Modbus serial links. The Modbus client sets this field in the request, and the server must return the same value in the response.

27.1.5 Parameter Definition

27.1.5.1 Modbus Data Model

Modbus bases its data model on a series of tables that have distinguishing characteristics. The four primary tables are:

Item	Object type	Access type	Description
Discrete inputs	Single bit	Read-only	This type of data can be provided by an I/O system.
Coil	Single bit	Read-write	This type of data can be altered by an application program.
Input registers	16-bit	Read-only	This type of data can be provided by an I/O system.
Holding registers	16-bit	Read-write	This type of data can be altered by an application program.

27.1.5.2 Parameter List

Function Code	Definition
01 (0x01)	Read coils
02 (0x02)	Read discrete inputs
03 (0x03)	Read multiple registers
04 (0x04)	Read input registers
05 (0x05)	Write single coil
06 (0x06)	Write single register
15 (0x0F)	Write multiple coils
16 (0x10)	Write multiple registers

27.1.5.3 Parameter Description

01 (0x01): Read coils/02 (0x02): Read discrete inputs

This function code is used to read 1 to 2000 contiguous status of coils (or discrete inputs) in a remote device.

The following table lists the request PDUs.

No.	Meaning of Data (Byte)	Number of Bytes	Description
1	Function code	1	0x01: Read coils; 0x02: Read discrete inputs
2	Coil head address	2	Upper bits are followed by lower bits. See coil addressing. For details, see descriptions about "Function codes and addresses" in Configuring Modbus-RTU or Modbus-ASCII Slave or Configuring Modbus-TCP Slave
3	Number of coils	2	Upper bits are followed by lower bits (N). The maximum value of N is 2000.

The following table lists the response PDUs.

No.	Meaning of Data (Byte)	Number of Bytes	Description
1	Function code	1	0x01: Read coils/0x02: Read discrete inputs
2	Number of bytes	1	Value: (N + 7)/8
3	Coil status	(N + 7)/8	Every 8 coils are combined into one byte. If the number of coils is not a multiple of 8, undefined bits are filled with 0. The first 8 coils are in the first byte, and the coil with the smallest address is in the least significant bit. This pattern continues for the rest of the coils.

The following table lists the error response PDUs.

No.	Meaning of Data (Byte)	Number of Bytes	Description
1	Function code	1	Function code + 0x80; 0x81: Read coils/0x82: Read discrete inputs
2	Exception code	1	0x01, 0x02, 0x03, or 0x04. See the exception code list.

03 (0x03): Read multiple registers/04 (0x04): Read input registers

This function code is used to read the content of a contiguous block of holding registers (or input registers) in a remote device.

The following table lists the request PDUs.

No.	Meaning of Data (Byte)	Number of Bytes	Description
1	Function code	1	0x03: Read multiple registers/0x04: Read input registers
2	Register head address	2	Upper bits are followed by lower bits. See register addressing. For details, see descriptions about "Function codes and addresses" in Configuring Modbus-RTU or Modbus-ASCII Slave or Configuring Modbus-TCP Slave
3	Number of registers	2	Upper bits are followed by lower bits (N). The maximum value of N is 125.

The following table lists the response PDUs.

No.	Meaning of Data (Byte)	Number of Bytes	Description
1	Function code	1	0x03: Read multiple registers/0x04: Read input registers
2	Number of bytes	1	Value: N x 2
3	Register value	N x 2	Every two bytes represent one register value, with upper bits followed by lower bits. The register with the minimum address is in the foremost.

The following table lists the error response PDUs.

No.	Meaning of Data (Byte)	Number of Bytes	Description
1	Function code	1	Function code + 0x80; 0x83: Read multiple registers/ 0x84: Read input registers
2	Exception code	1	0x01, 0x02, 0x03, or 0x04. See the exception code list.

05 (0x05): Write single coil

This function code is used to write a single output to either ON or OFF in a remote device.

The following table lists the request PDUs.

No.	Meaning of Data (Byte)	Number of Bytes	Description
1	Function code	1	0x05: Write single coil
2	Coil address	2	Upper bits are followed by lower bits. See coil addresses. For details, see descriptions about

No.	Meaning of Data (Byte)	Number of Bytes	Description
			"Function codes and addresses" in Configuring Modbus-RTU or Modbus-ASCII Slave or Configuring Modbus-TCP Slave
3	Coil status	2	Upper bits are followed by lower bits. ON is 0xFF00, while OFF is 0x0000.

The following table lists the response PDUs.

No.	Meaning of Data (Byte)	Number of Bytes	Description
1	Function code	1	0x05: Write single coil
2	Coil address	2	Upper bits are followed by lower bits. See coil addressing. For details, see descriptions about "Function codes and addresses" in Configuring Modbus-RTU or Modbus-ASCII Slave or Configuring Modbus-TCP Slave
3	Coil status	2	Upper bits are followed by lower bits. A non-zero value is valid.

The following table lists the error response PDUs.

No.	Meaning of Data (Byte)	Number of Bytes	Description
1	Function code	1	Function code + 0x80; 0x85: Write single coil
2	Exception code	1	0x01, 0x02, 0x03, or 0x04. See the exception code list.

06 (0x06): Write single register

This function code is used to write a single holding register in a remote device.

The following table lists the request PDUs.

No.	Meaning of Data (Byte)	Number of Bytes	Description
1	Function code	1	0x06: Write single register
2	Register address	2	Upper bits are followed by lower bits. See register value addressing. For details, see descriptions about "Function codes and addresses" in Configuring Modbus-RTU or Modbus-ASCII Slave or Configuring Modbus-TCP Slave

No.	Meaning of Data (Byte)	Number of Bytes	Description
3	Register value	2	Upper bits are followed by lower bits.

The following table lists the response PDUs.

No.	Meaning of Data (Byte)	Number of Bytes	Description
1	Function code	1	0x06: Write single register
2	Register address	2	Upper bits are followed by lower bits. See register value addressing. For details, see descriptions about "Function codes and addresses" in Configuring Modbus-RTU or Modbus-ASCII Slave or Configuring Modbus-TCP Slave
3	Register value	2	Upper bits are followed by lower bits.

The following table lists the error response PDUs.

No.	Meaning of Data (Byte)	Number of Bytes	Description
1	Function code	1	Function code + 0x80; 0x86: Write single register
2	Exception code	1	0x01, 0x02, 0x03, or 0x04. See the exception code list.

15 (0x0F): Write multiple coils

This function code is used to force each coil in a sequence of coils to either ON or OFF in a remote device.

The following table lists the request PDUs.

No.	Meaning of Data (Byte)	Number of Bytes	Description
1	Function code	1	0x0F: Write multiple coils
2	Coil head address	2	Upper bits are followed by lower bits. See coil addressing. For details, see descriptions about "Function codes and addresses" in Configuring Modbus-RTU or Modbus-ASCII Slave or Configuring Modbus-TCP Slave
3	Number of coils	2	Upper bits are followed by lower bits (N). The maximum value of N is 1968.
4	Number of bytes	1	Value: (N + 7)/8
5	Coil status	(N + 7)/8	

No.	Meaning of Data (Byte)	Number of Bytes	Description
			Every 8 coils are combined into one byte. If the number of coils is not a multiple of 8, undefined bits are filled with 0. The first 8 coils are in the first byte, and the coil with the smallest address is in the least significant bit. This pattern continues for the rest of the coils.

The following table lists the response PDUs.

No.	Meaning of Data (Byte)	Number of Bytes	Description
1	Function code	1	0x0F: Write multiple coils
2	Coil head address	2	Upper bits are followed by lower bits. See coil addressing. For details, see descriptions about "Function codes and addresses" in Configuring Modbus-RTU or Modbus-ASCII Slave or Configuring Modbus-TCP Slave
3	Number of coils	2	Upper bits are followed by lower bits.

The following table lists the error response PDUs.

No.	Meaning of Data (Byte)	Number of Bytes	Description
1	Function code	1	Function code + 0x80; 0x8F: Write single register
2	Exception code	1	0x01, 0x02, 0x03, or 0x04. See the exception code list.

16 (0x10): Write multiple registers

This function code is used to write a block of contiguous registers (1 to 123 registers) in a remote device.

The following table lists the request PDUs.

No.	Meaning of Data (Byte)	Number of Bytes	Description
1	Function code	1	0x10: Write multiple registers
2	Register head address	2	Upper bits are followed by lower bits. See register addressing. For details, see descriptions about "Function codes and addresses" in Configuring Modbus-RTU or Modbus-ASCII Slave or Configuring Modbus-TCP Slave
3		2	

No.	Meaning of Data (Byte)	Number of Bytes	Description
	Number of registers		Upper bits are followed by lower bits (N). The maximum value of N is 123.
4	Number of bytes	1	Value: N x 2
5	Register value	N x 2	Register value

The following table lists the response PDUs.

No.	Meaning of Data (Byte)	Number of Bytes	Description
1	Function code	1	0x10: Write multiple registers
2	Register head address	2	Upper bits are followed by lower bits. See register addressing. For details, see descriptions about "Function codes and addresses" in Configuring Modbus-RTU or Modbus-ASCII Slave or Configuring Modbus-TCP Slave
3	Number of registers	2	Upper bits are followed by lower bits.

The following table lists the error response PDUs.

No.	Meaning of Data (Byte)	Number of Bytes	Description
1	Function code	1	Function code + 0x80; 0x90: Write single register
2	Exception code	1	0x01, 0x02, 0x03, or 0x04. See the exception code list.

27.1.6 List of Fault Codes

Code	Name	Description
0x01	Illegal function code	The function code received in the query is not an allowable action for the server (or slave). This may be because the function code is only applicable to new devices, and is not implementable in the unit selected. It could also indicate that the server (or slave) is in the wrong state to process a request of this type, for example, because it is unconfigured and is being asked to return register values.
0x02	Illegal data address	The data address received in the query is not an allowable address for the server (or slave). More specifically, the combination of reference

Code	Name	Description
		number and transfer length is invalid. For a controller with 100 registers, a request with the offset 96 and the length 4 will succeed, but a request with the offset 96 and the length 5 will result in exception code 0x02.
0x03	Illegal data value	A value contained in the query data field is not an allowable value for server (or slave). This indicates a fault in the structure of the remainder of a complex request, such as that the implied length is incorrect. It specifically does not mean that a data item submitted for storage in a register has a value outside the expectation of the application program, since the Modbus protocol is unaware of the significance of any particular value of any particular register.
0x04	Slave device fault	An unrecoverable error occurred while the server (or slave) was attempting to perform the requested action.

27.2 Firmware Programming and Upgrade

27.2.1 USB Firmware Programming

CAUTION

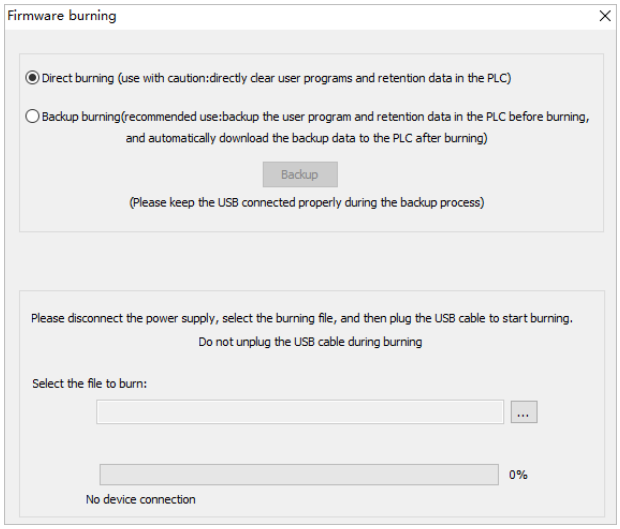
- The PLC firmware programming function of AutoShop is only available for the Easy series.
- When multiple AutoShop software applications are opened, only one of them is allowed to perform programming, while others cannot access the "Firmware burning" function.
- The Easy series models involve two programming files: one for the Easy30X, Easy32X, and Easy50X models, and the other for the Easy52X models.
- All the original applications will be deleted after the firmware programming is completed.

Preconditions

USB cable and firmware programming file are ready. You can contact Inovance technical support to obtain the firmware programming file.

Procedure

1. Turn off the PLC power.
2. In the menu bar, choose "Tools" > "Firmware burning". The "Firmware burning" page is displayed, as shown in the following figure.



3. Click "..." and select the firmware programming file.
4. Plug the USB cable (if the USB cable is already connected, remove and re-plug it). A dialog box pops up to prompt whether to program. Click "OK" to start programming.
   After the programming succeeds, the "Burned successfully" prompt box is displayed.

27.2.2 SD Card Firmware Programming

CAUTION

- During programming using an SD card, power-off is strictly prohibited. Otherwise, the PLC may become unusable or other serious abnormalities may occur.
- All the original applications will be deleted after the firmware programming is completed.

Preconditions

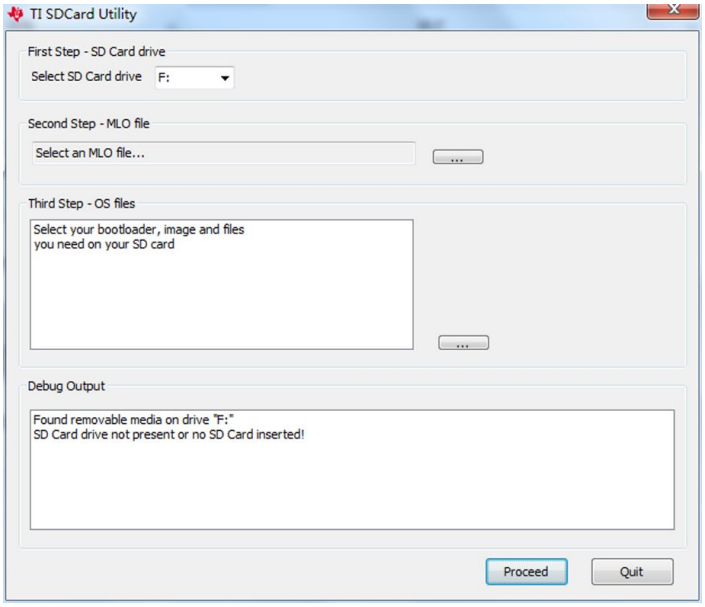
You have obtained the SD card preparation software from Inovance technical support.

Procedure

To program the firmware using an SD card, program the SD card first according to the following steps.

1. Prepare hardware.
   Prepare an SD (TF) card as shown in the figure, with the card storage capacity no more than 32 G.

2. Insert the SD card into a card reader and plug the card reader into the USB port of the computer.
3. The SD card programming method varies with the PLC type.
   - The SD card programming procedure for the H5U series PLC is as follows.
     a. Double-click the "PhoenixCard" to open the SD card programming tool.
        The following page is displayed and shows which disk the card reader is located, as shown in the following figure.



b. Select the MLO file in the software package and operate in the order shown in the following figure.

c. Click the button specified in the following figure to open the folder where the programming content is located. Press Ctrl+A to select all the files and then click "Open".

d. Click "Proceed". After the following page is displayed, click "Boot" and then "Start" to format the SD card.

e. After the formatting, click "Close" to start programming. After the programming is completed, the following page is displayed.

f. Insert the SD card into the SD card slot of the controller.

g. Power off the PLC and then power it on. When the LED display shows UU, it indicates that the programming has started. The programming process may take about one minute. When the programming succeeds, the LED display shows 00 or CC. Remove the SD card to complete the programming.

- The SD card programming procedure for the Easy series PLC is as follows.

a. Double-click the "PhoenixCard" to open the SD card programming tool.

b. On the following page, click "Refresh drive letter" to identify the SD card.

c. Click "Firmware".

d. Select the files to be programmed and click "Open".

e. Select "Mass production card" and then click "Burn card".
As shown in the following figure, the programming is completed.

f. Insert the programmed SD card into the GE20-TF extension card, and insert the extension card to slot B of the PLC. Program starts after power-on.
If the RUN and ERR indicators on the PLC blink quickly for 3 seconds, firmware programming starts. If the RUN and ERR indicators are steady ON, firmware programming is in progress. If the RUN and ERR indicators blink slowly, firmware programming is completed.

g. After firmware programming, power off the PLC, remove the SD card, and then power the PLC.

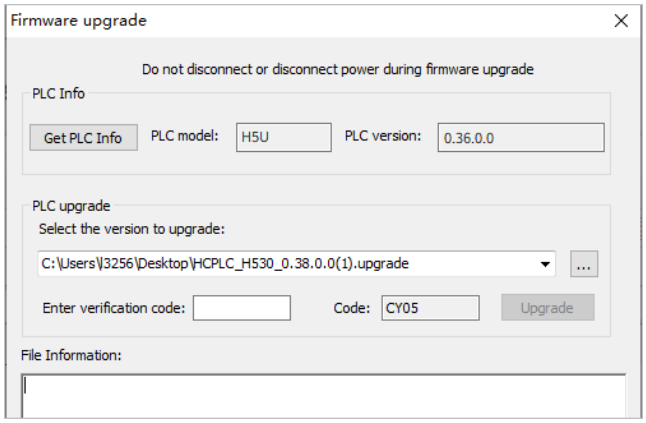
NOTE

- During firmware programming, ensure that the PLC is powered normally. Powering off the PLC during programming may cause the PLC to fail to start or function normally. In such a case, try again to program the firmware using the SD card. If the programming fails, send the PLC to the factory for repair.
- During firmware programming, if the LED display shows flashing ERR, it means that the programming is successful but the PLC detects a program error or communication error during running.

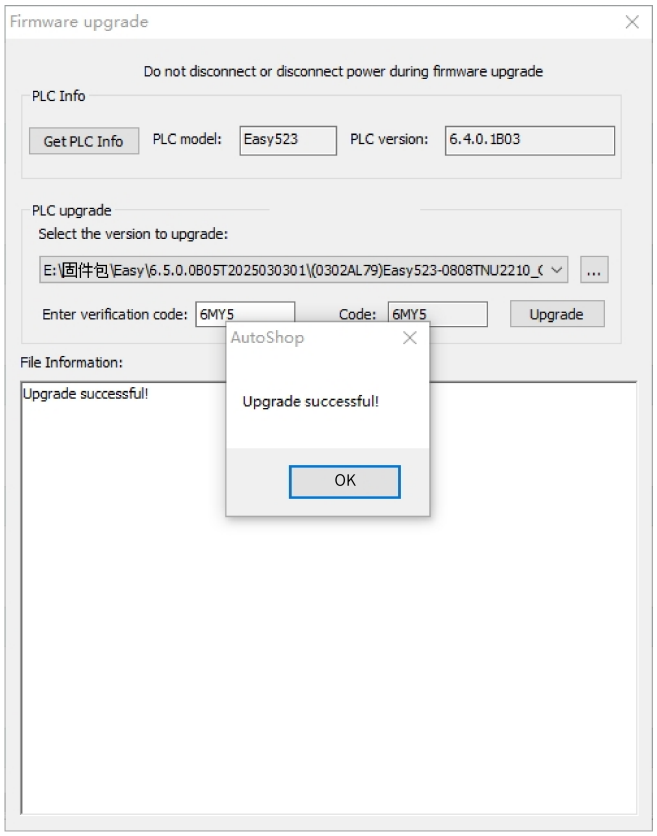
27.3 Firmware Upgrade

The H5U series PLCs support firmware upgrade only through Ethernet, and the Easy series PLCs support firmware upgrade through Ethernet and USB drive. The original application will be retained after the firmware upgrade is completed.

1. In the menu bar, choose "Tools" > "Firmware upgrade". The "Firmware upgrade" dialog box is displayed.



2. Select the firmware version to be upgraded, enter the verification code, and click "Upgrade".
3. Wait for the "Upgrade successful" prompt box to pop up, and the firmware upgrade is completed.



> **NOTE**
>
> During firmware upgrade, ensure that the PLC is powered normally. Powering off the PLC during upgrade may cause the PLC to fail to start or function normally. In most of such cases, firmware upgrade can still be done by using an SD card. Otherwise, the PLC needs to be returned to the factory for repair.-

27.4 Function Application of Download File Generation

27.4.1 Generating Down Files

27.4.1.1 Overview

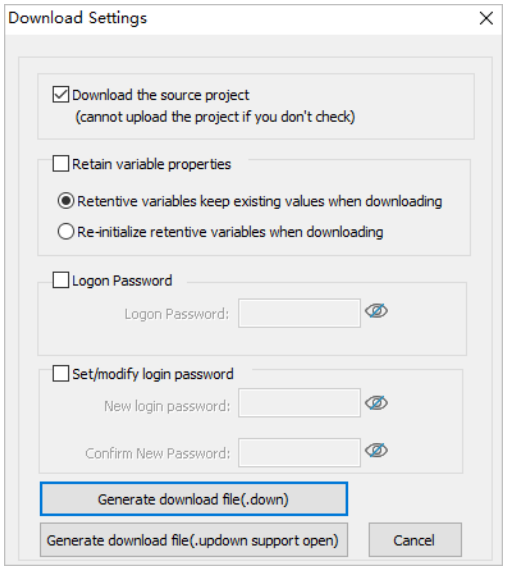
The download file feature refers to the ability of the PLC project to compile and generate a Down file, which can be downloaded without opening the original project.

- Batch update or upgrade PLC projects using an SD card.
- Update a PLC project using AutoShop software tool.

27.4.1.2 Generating a Down File

Before downloading a Down file, generate the Down file in the AutoShop background. The specific steps are as follows.

1. Open the PLC project and choose "File" > "Generate download file (D)".
2. In the "Download Settings" dialog box displayed, set the download file properties, and then click "Generate download file (.down)".



Name	Description
"Download the source project (cannot upload the project if you don't check)"	Select this option to enable project uploading, or deselect this option to disable project uploading.
"Retain variable properties"	• Retain existing values of retentive variables when downloading; or • Re-initialize retentive variables when downloading.

Name	Description
"Logon password"	• If the PLC does not have a login password, leave this option deselected. • If the PLC has a login password, this option must be selected and the PLC login password must be entered. If this option is not selected or the password is incorrect, the upgrade will fail.
"Set/modify login password"	• To modify the current login password of the PLC, fill out the "Logon Password", "New login password", and "Confirm New Password" fields. • To set a login password when the PLC does not have a login password, just fill out the "New login password" and "Confirm New Password" fields.
	Shows or hides passwords.

NOTE

- The password takes effect immediately after the Down file is upgraded.
- After the password takes effect, the PLC will log out the current user.

3. Select the archive path for the Down file and click "OK".

27.4.1.3 Upgrading the Down File through an SD Card

Put the Down file compiled using AutoShop in the "PLCProgram" directory of the SD card, and then insert the SD card into the PLC main module. Press and hold the MFK key on the PLC panel for three seconds to enter the "Sd" menu. Press the MFK key again to start programming the user program in the SD card into the PLC host. The LED display shows the programming progress (00 to 99), and shows "PP" after the programming is completed.

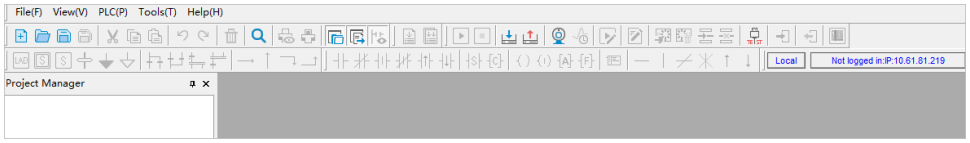
If the password is incorrect, the indicator prompts E5, indicating a password verification error.

NOTE

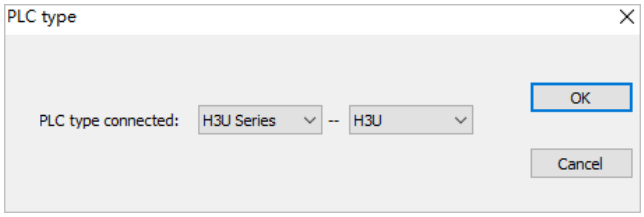
Down files can be upgraded through SD cards only when PLCs are stopped.

27.4.1.4 Downloading the Down File Through AutoShop

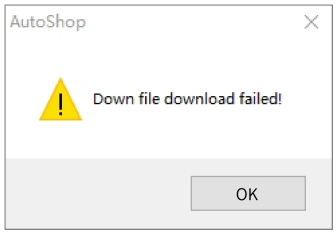
To download the generated Down file, close the project first, as shown in the following figure.



1. Click "Download" and select the PLC type connected.



2. Click "Download". If the password verification fails, the system pops up a prompt box, and you need to enter the correct password.



> **NOTE**
>
> If the PLC is in the logged-in state when the Down file is downloaded in the background, the PLC will be logged out first.

27.4.1.5 Downloading and Uploading the Down File Through HMI

Preconditions

Uploading or downloading Down files through HMI requires a firmware version of V0.8.8.27 or later.

Downloading Down files

1. Enter the control panel of the IT7000 series HMI.
   - In InoTouchPad, create an IT7000 project, configure a button, and configure the system function "OpenControlPanel" for the button to enter the control panel.
   - Power on the IT7000 HMI and press and hold the screen to enter the control panel.
2. Click the "Download" menu in the control panel.
3. From the drop-down list of "Mounted Device", select "USB", and select a project file suffixed with .down.

> **NOTE**
>
> When the operation object is Easy but you select project files of H3U or other series, the system prompts "Failed to open the down file".

4. From the drop-down list of "Device model", select the target device series.

> **NOTE**
>
> The Easy series PLCs support uploading and downloading projects through the network port. Therefore, you do not need to set "PLC Upgrade Port" and "Baud rate".

5. Click "Download", and enter the IP address of the PLC.
6. Click "OK".

Uploading Down files

1. Enter the control panel of the IT7000 series HMI.
2. Click the "Download" menu in the control panel.
3. From the drop-down list of "Device model", select the target device series.

> **NOTE**
>
> The Easy series PLCs support uploading and downloading projects through the network port. Therefore, you do not need to set "PLC Upgrade Port" and "Baud rate".

4. Click "Upload". In the dialog box displayed, enter the IP address of the PLC and the name of the file to be uploaded in the "File name" field.

> **NOTE**
>
> To upload a file, you need to enter the device IP address and rename the file to make the file name end with ".down", for example, "test.down".
>
> The password for upload is the password of the uploaded Down file.

5. Click "OK".

27.4.1.6 Compatibility Issues

When a Down file generated by AutoShop V4.0.0.0 is downloaded to PCB software of V3.0.0.0 or earlier version, errors such as Down upgrade failure, file format error, and parsing failure may occur.

27.4.2 Generating Updown Files

27.4.2.1 Overview

Updown files are Down files that can be uploaded. Updown files enable upload and download of user programs.

AutoShop allows you to compile a project into an Updown file, and open the Updown file in AutoShop to further edit the project.

- Updown files can be downloaded and uploaded through HMI, enabling quick copy and transfer of projects between different PLCs.
- Updown files can be downloaded and uploaded using SD cards.
- Updown files can also be downloaded and uploaded through AutoShop, making project management flexible.

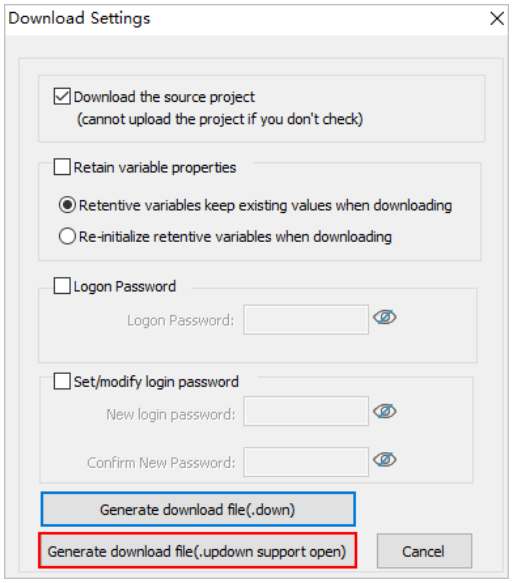
Operations involved are:

Function	Tool
Compile and generate Updown files	AutoShop
Open and edit Updown files	AutoShop
Upload and download Updown files through AutoShop	AutoShop
Upload and download Updown files through HMI	HMI (IT7000)
Upload and download Updown files through SD cards	SD card

27.4.2.2 Generating an Updown File

Follow these steps to generate an Updown file:

1. In the programmed and compiled project, choose "File" > "Generate download file". The "Download Settings" dialog box is displayed.
2. Click "Generate download file(.updown support open)" to generate an Updown file.



Name	Description
"Download the source project"	Select this option to enable project uploading, or deselect this option to disable project uploading.
"Retain variable properties"	Retain existing values of retentive variables when downloading; or Re-initialize retentive variables when downloading.
"Logon password"	The PLC login password, which must be consistent with the login password of the target PLC to complete the download of the Updown file. If the target PLC does not have a login password, do not select the "Logon Password" option.

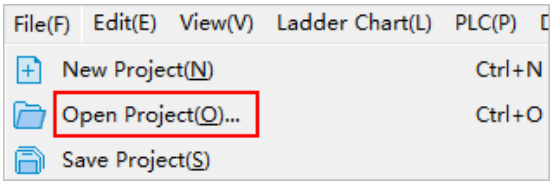
Name	Description
"Set/modify login password"	After a successful Updown file download, the target PLC's login password is updated to the "New login password".
	Shows or hides passwords.

3. Select the archive path for the Updown file and click "OK".

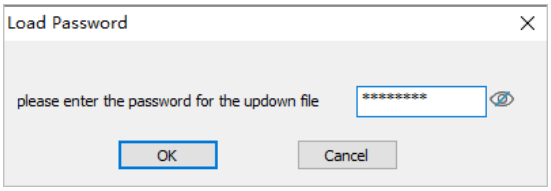
27.4.2.3 Opening the Updown File

Follow these steps to open and edit an Updown file:

1. Choose "File" > "Open Project", as shown in the following figure.



2. In the "Open" window displayed, select the "*.updown" file you want to open.
3. After selecting the file, click "Open" to open the Updown file. If the Updown file is password-protected, enter the password for verification before the file can be opened.



NOTE

- If the Updown file has "Set/modify login password" enabled, use the "Set/modify login password" of the Updown file for verification.
- If the Updown file only has "Logon Password" enabled, use the "Logon Password" of the Updown file for verification.

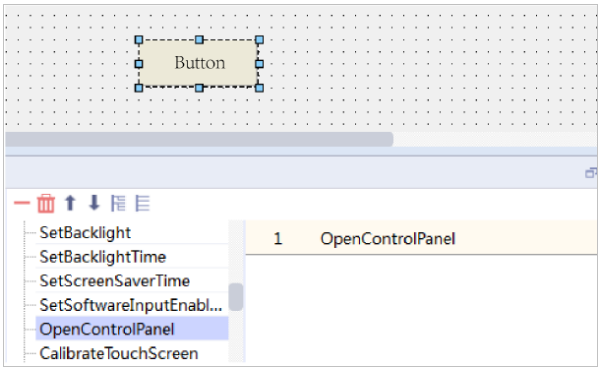
27.4.2.4 Downloading and Uploading the Updown File through HMI

Preconditions

Uploading or downloading Updown files through HMI requires a firmware version of 0.8.8.27 or later.

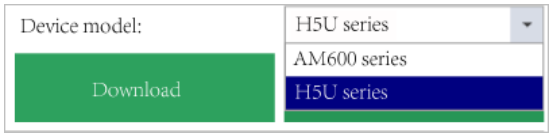
Downloading Updown files

1. Enter the control panel of the IT7000 series HMI.
   - In InoTouchPad, create an IT7000 project, configure a button, and configure the system function "OpenControlPanel" for the button to enter the control panel.



- Power on the IT7000 HMI and press and hold the screen to enter the control panel.

2. Click the "Download" menu in the control panel.
3. In the window that is displayed, select the corresponding mounted device and the Updown file to be downloaded.
4. Select the target device series and click "Download".



5. In the window that is displayed, enter the IP address of the PLC device and click "OK".

Uploading Updown files

The steps for upload are similar to those for download. The control panel interface is as shown in the following figure.

> **NOTE**
>
> - To upload a file, you need to enter the device IP address and rename the file to make the file name end with ".updown", for example, "test.updown".
> - The password for upload is the password of the uploaded Updown file.
> - Both Down and Updown files can be downloaded, but only Updown files can be uploaded.

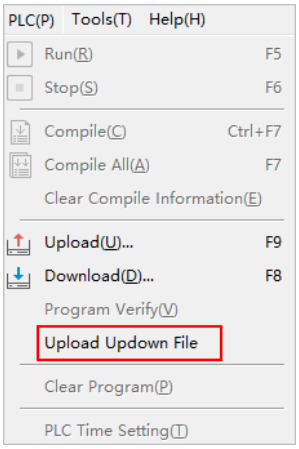
27.4.2.5 Downloading and Uploading the Updown File Through AutoShop

Downloading Updown files

Close the project. Click "Download" in the toolbar. After connecting to the PLC, select the "*.updown" file to be downloaded in the window that is displayed, as shown in the following figure.

Uploading Updown files

Close the project. Choose "PLC" > "Upload Updown File", as shown in the following figure.



NOTE

If an Updown file that has been downloaded to the PLC is downloaded again in the Down format or through the background, inconsistency may occur between the Updown file uploaded and the actual project running in the PLC.

27.4.2.6 Downloading the Updown File through an SD Card

Put the Updown file compiled using AutoShop in the "PLCProgram" directory of the SD card, and then insert the SD card into the PLC main module. Press and hold the MFK key on the PLC panel for three seconds to enter the "Sd" menu. Press the MFK key again to start programming the user program in the SD card into the PLC host. The LED display shows the programming progress (00 to 99), and shows "PP" after the programming is completed.

If the password is incorrect, the indicator prompts E5, indicating a password verification error.

27.5 Communication Application of Customized Variables

27.5.1 Overview

In the feature of customized variables, addresses of variables are automatically allocated by the software. Therefore, the variables cannot be accessed directly using fixed addresses. Customized variables can be accessed in the following two ways:

- HMI tag communication: only available for Inovance IT7000 series touch screens
- Mapping address: applicable to all devices that support the Modbus protocol

27.5.2 Example Project Requirements

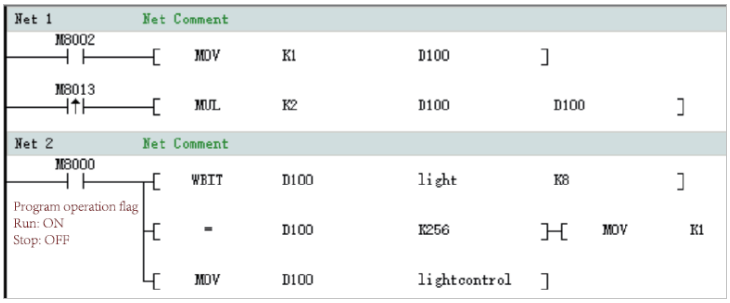
Write an H5U marquee program and have the corresponding bit elements and control word status displayed through the IT7000 HMI.

27.5.3 PLC Programming

27.5.3.1 Accessing Customized Variables Through HMI Tag Communication Function

1. Create a PLC project. For details, see Creating or Opening a Project.

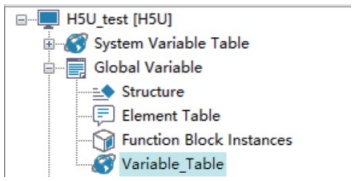
2. Write a PLC program. Write a program as shown in the following figure, where "light" is an array containing eight Bool-type variables and "light_control" is an INT-type variable used for HMI display.



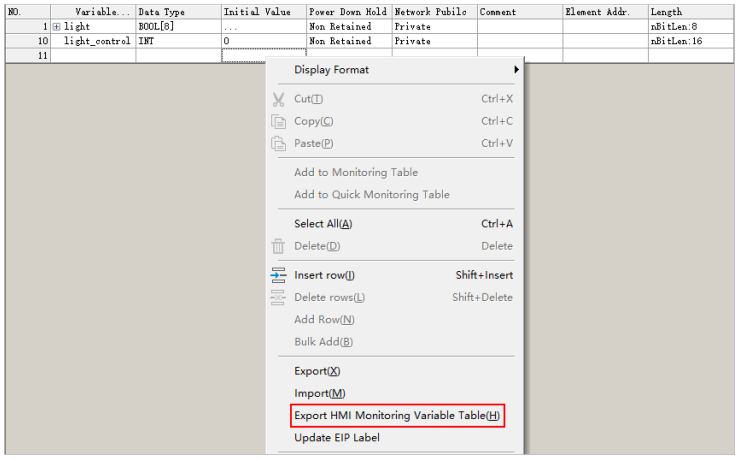
3. Compile the project and export the variable table.

a. After compiling the project, click in the toolbar to edit the program.

b. After the compilation is completed, double-click "Variable_Table" in the "Project Manager" tree on the left to access the variable table page.

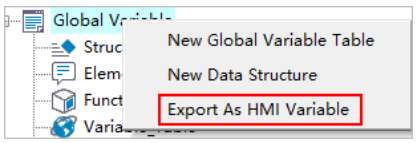


c. Right-click the variable table and click "Export HMI Monitoring Variable Table".



d. In the window displayed, set an archive path and file name, and then click "Save" to complete the export of the variable table.

e. (Optional) To export all customized variables, right-click "Global Variables" and select "Export As HMI Variable".



4. Download the PLC program. After exporting the variable table, click to download the program to the PLC.

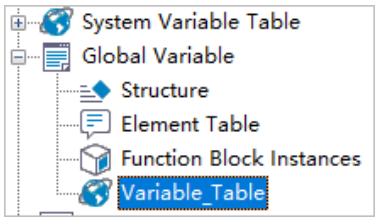
5. Set the PLC IP address. For details, see "Communication Connection".

27.5.3.2 Accessing Customized Variables Through Address Mapping

1. Create a project and write a PLC program.

2. Allocate variable addresses.

   a. In the project management pane, double-click "Variable_Table" to access the variable table page.



   b. Allocate soft element addresses for customized variables.

   The light variable is a BOOL-type array and occupies M0 to M7, a total of eight bits, after being mapped to M0.

NO.	Variable...	Data Type	Initial Value	Power Down Hold	Network Public	Comment	Element Addr.	Length
1	⊞ light	BOOL[8]	...	Non Retained	Private		M0	nBitLen:8
10	light_control	INT	0	Retained	Private		D1000	nBitLen:16

   c. Compile the project to automatically generate the allocated addresses.

   Click the compile button to compile the program. After the compilation is completed, the software automatically generates the allocated addresses.

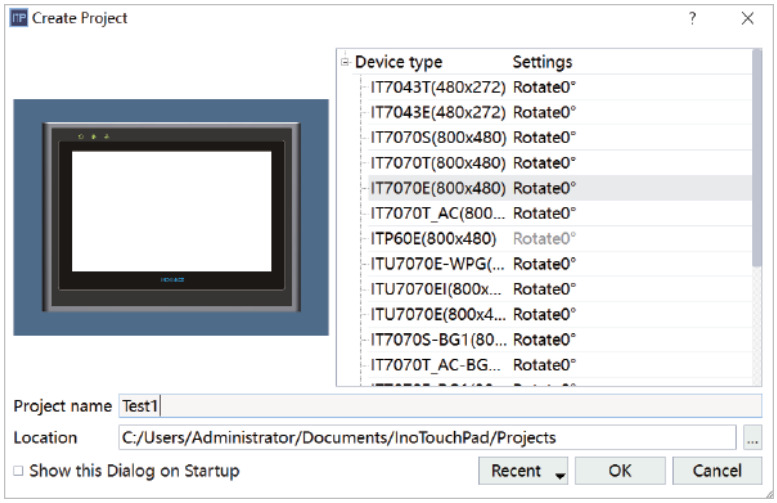
NO.	Variable...	Data Type	Initial Value	Power Down Hold	Network Public	Comment	Element Addr.	Assign Addr.	Length
1	⊞ light	BOOL[8]	...	Non Retained	Private		M0	0x0100000	nBitLen:8
10	light_control	INT	0	Retained	Private		D1000	0x0403E80	nBitLen:16

3. Download the PLC program and set the PLC IP address.

27.5.4 HMI Configuration

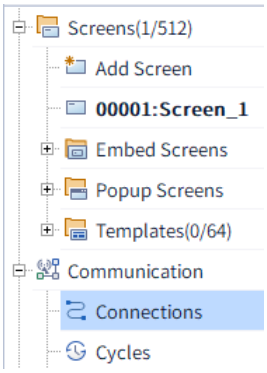
27.5.4.1 Accessing Customized Variables Through HMI Tag Communication Function

1. Create an HMI project. Open InoTouchPad. Create a project. Set an archive path, name, and device type for the project, and then click "OK".

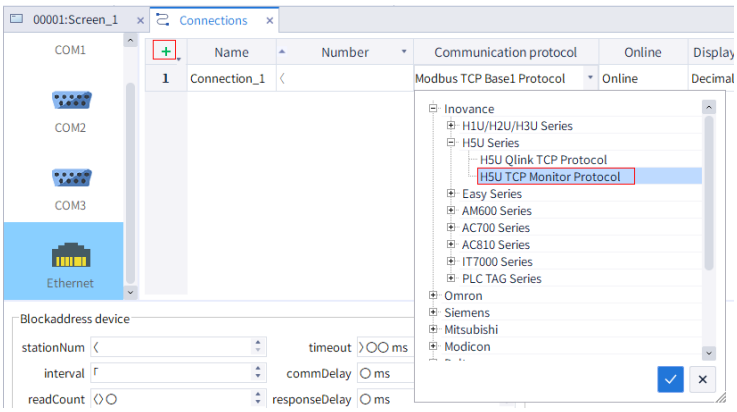


2. Create a communication connection.

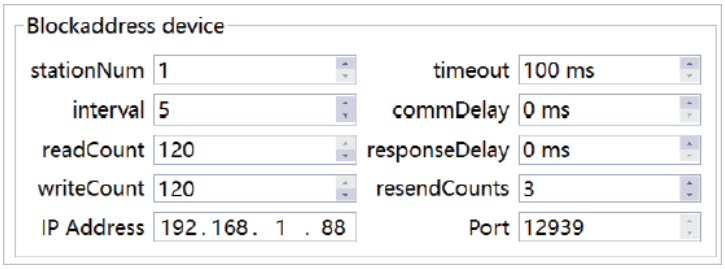
   a. In the project management pane on the left, double-click the connection tab.



b. On the connection management page, click the "+" icon to add a connection, and select the H5U TCP monitoring protocol.

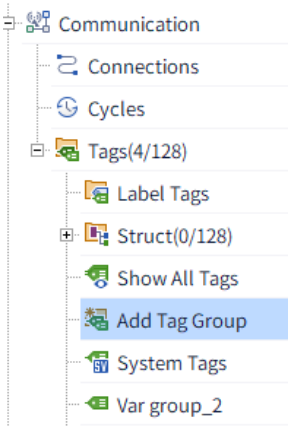


c. Enter the IP address of the connected PLC to complete the setup.

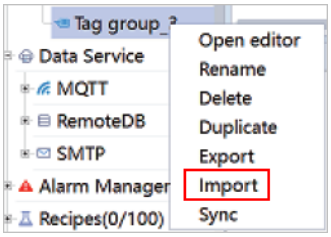


3. Add variables.

a. Click "Tags" to expand the variable menu. Click "Add Tag Group" to add a variable group.

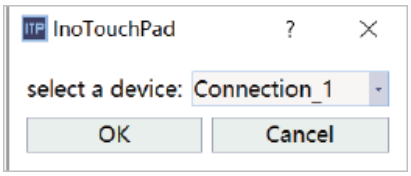


b. Right-click the newly added variable group and select "Import".



c. In the pop-up window, select the variable table exported from the PLC, and click "Open".

d. Select the created H5U connection and click "OK".

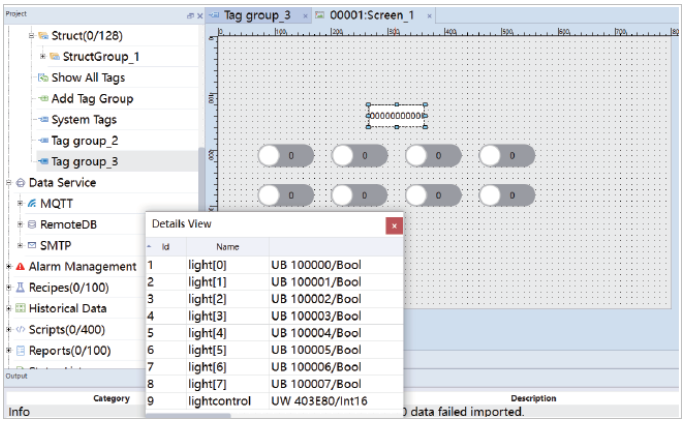


e. The variables are successfully added to the HMI variable table.

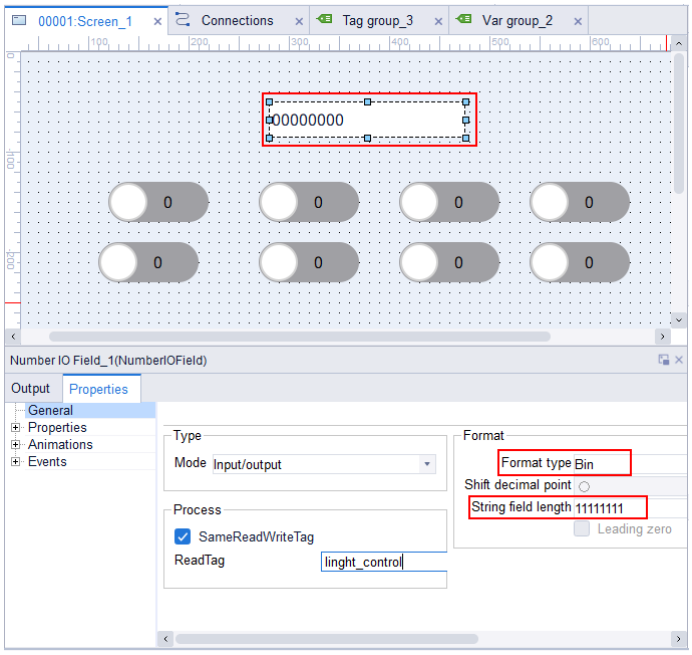
	Name	Number	Connect...	Data type	Length	Array c...	Address	Acquisit...	Acquisit...	Data log Id
1	light[0]	1	Connec...	Bool	1	1	UB 100...	100ms	Cyclic o...	<Undefin...
2	light[1]	2	Connec...	Bool	1	1	UB 100...	100ms	Cyclic o...	<Undefin...
3	light[2]	3	Connec...	Bool	1	1	UB 100...	100ms	Cyclic o...	<Undefin...
4	light[3]	4	Connec...	Bool	1	1	UB 100...	100ms	Cyclic o...	<Undefin...
5	light[4]	5	Connec...	Bool	1	1	UB 100...	100ms	Cyclic o...	<Undefin...
6	light[5]	6	Connec...	Bool	1	1	UB 100...	100ms	Cyclic o...	<Undefin...
7	light[6]	7	Connec...	Bool	1	1	UB 100...	100ms	Cyclic o...	<Undefin...
8	light[7]	8	Connec...	Bool	1	1	UB 100...	100ms	Cyclic o...	<Undefin...
9	lightco...	9	Connec...	Int16	2	1	UW 40...	100ms	Cyclic o...	<Undefin...

4. Configure the HMI.

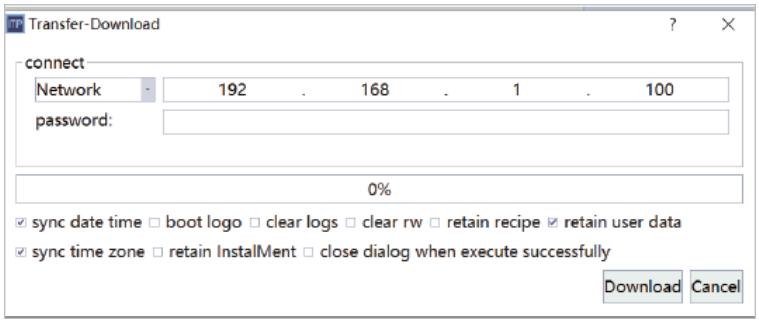
a. The HMI allows you to directly drag variables for programming. Drag variables from the detailed view to the programming interface one by one to complete programming.



b. Double-click the value I/O field, set the display format to binary, and set the character field length to eight bits for easy observation.



5. Download the HMI program. Click . In the window that is displayed, set the target HMI IP and click "Download". Wait for the progress bar to reach 100% to complete the HMI program download.

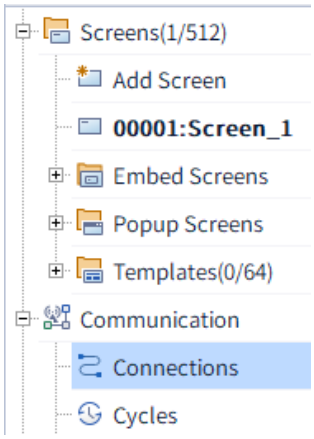


27.5.4.2 Accessing Customized Variables Through Address Mapping

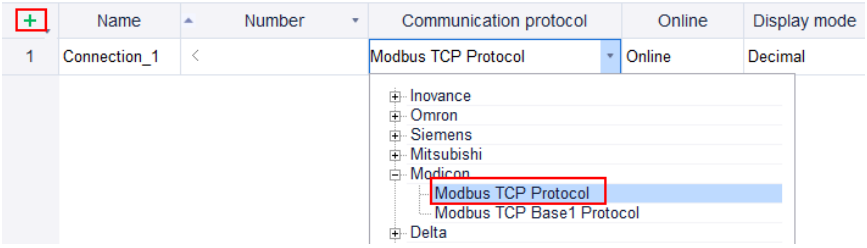
1. Create an HMI project.

2. Create a communication connection.

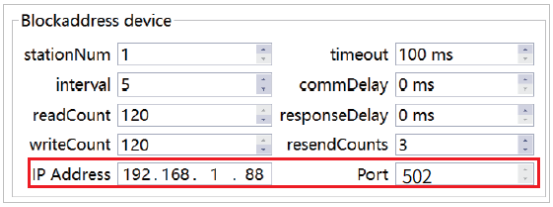
a. In the "Project Manager" tree on the left, double-click "Connections".



b. On the connection management page, click the "+" icon to add a connection, and select the Modbus-TCP monitoring protocol.

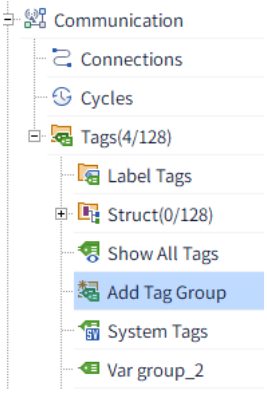


c. Enter the IP address of the connected PLC and set the port number to 502 to complete the setup.

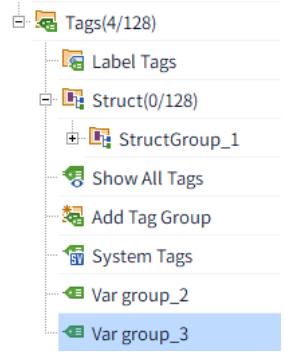


3. Add a variable.

a. Click "Tags" to expand the variable menu. Click "Add Tag Group" to add a variable group.



b. Double-click the newly created variable group to open it.



c. Add variables to be monitored.

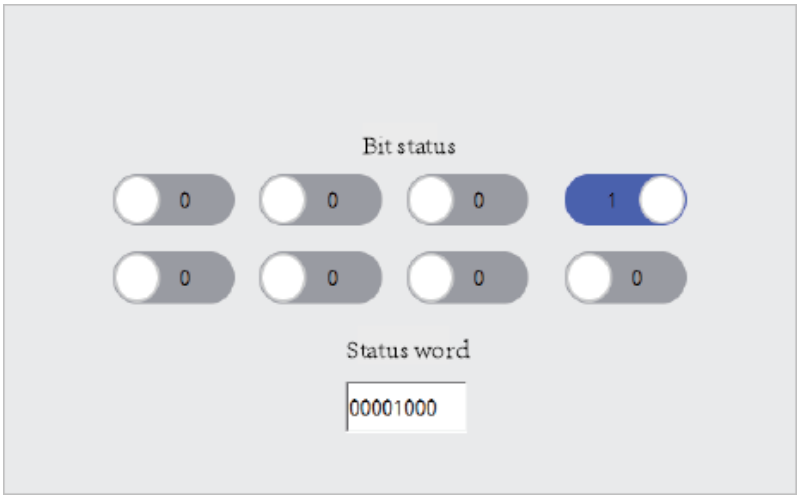
Addresses of the added variables depend on the addresses allocated in the PLC.

	Name	Number	Connection Id	Data type	Length	Array count	Address	Acquisition c…	Acquisition …	Data log Id	Logging cycle…	Logging acq…
1	M0	1	Connection_1	Bool	1	1	1x 0	1s	Cyclic on use	<Undefined>	1s	Cyclic conti…
2	M1	2	Connection_1	Bool	1	1	1x 1	1s	Cyclic on use	<Undefined>	1s	Cyclic conti…
3	M2	3	Connection_1	Bool	1	1	1x 2	1s	Cyclic on use	<Undefined>	1s	Cyclic conti…
4	M3	4	Connection_1	Bool	1	1	1x 3	1s	Cyclic on use	<Undefined>	1s	Cyclic conti…
5	M4	5	Connection_1	Bool	1	1	1x 4	1s	Cyclic on use	<Undefined>	1s	Cyclic conti…
6	M5	6	Connection_1	Bool	1	1	1x 5	1s	Cyclic on use	<Undefined>	1s	Cyclic conti…
7	M6	7	Connection_1	Bool	1	1	1x 6	1s	Cyclic on use	<Undefined>	1s	Cyclic conti…
8	M7	8	Connection_1	Bool	1	1	1x 7	1s	Cyclic on use	<Undefined>	1s	Cyclic conti…
9	D1000	9	Connection_1	Int16	2	1	4x 1000	1s	Cyclic on use	<Undefined>	1s	Cyclic conti…

4. Configure HMI and download the HMI project.

27.5.5 Example Running Result

After communication is established through the tag communication feature, the bit states of customized variables and the corresponding status words can be monitored through the HMI.



Service and Support

Should you encounter a safety accident during the use or operation of the product, or face challenges in operating and maintaining the equipment, which remain unresolved after the relevant documentation is consulted, we provide multiple channels to ensure prompt resolution:

- Channel #1: Contact service@inovance.com.
- Channel #2: Visit https://www.inovance.com/global to access document downloads, after-sales support, spare parts ordering, repair applications, and authenticity verification services.
- Channel #3: Download My Inovance app (https://zshc-eu.inovance.com/download-pc/) where you can access products info and documentation, and query product parameters.

We are committed to providing you with quick and professional technical support, and we look forward to your satisfaction and trust.

19012249A19

Copyright © Shenzhen Inovance Technology Co., Ltd.

Shenzhen Inovance Technology Co., Ltd.

www.inovance.com

Add.: Inovance Headquarters Tower, High-tech Industrial Park,
Guanlan Street, Longhua New District,
Shenzhen 518000, P.R. China

Tel: (0755) 2979 9595 Fax: (0755) 2961 9897

Suzhou Inovance Technology Co., Ltd.

www.inovance.com

Add.: No. 52, Tian E Dang Road, Wuzhong District, 215104,
Suzhou City, Jiangsu Province, P.R. China

Tel: (0512) 6637 6666 Fax: (0512) 6285 6720