CTH200 Series PLC

User Manual

Edition: V1.10

Release date: 06/2019

Shenzhen Co-trust Technology Co.,Ltd

Copyright

Trademarks

, TrustPLC, CoPanel, COTRUST are registered trademarks of COTRUST.

Some of other designations used in this document are also registered trademarks; the owner's rights may be violated if they are used by third parties for their own purposes.

Disclaimer of Liability

We have checked the contents of this manual for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in this manual are reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are welcomed.

Safety Guidelines

Only qualified person are allowed to install, operate and maintenance on CTH200 Series PLC. Qualified persons are defined as persons who are authorized to commission, ground, tag circuits, equipment, and systems in accordance with established safety practices and standards.

COTRUST has no responsibility for any consequence caused by using this product.

This manual contains notices that you should observe to ensure your personal safety, as well as to protect the product and connected equipment. These notices are highlighted in the manual by a warning triangle and are marked as follows according to the level of danger:

Warning
"Warning" sign indicates operation not as required may result in severe personal injuries or even death.

Caution
"Caution" sign indicates operation not as required may result in personal injuries or device damage

Notice
"Notice" sign indicates necessary supplements or explanations

Precautions for Use

There must be safety circuit to ensure the operation security of PLC system when the external power off or PLC failed. Notices in design for use including:

Emergency brake circuit, safety circuit, positive/reverse operation interlock circuit and the upper/lower limit interlock switch of position for preventing device damage must be implemented in the PLC external circuit.

To ensure the device operation security, the external guard circuit and safety mechanism for the output signal of major accident is a must.

The PLC can detect system exceptions which resulting all output turned off, to ensure the device operation normal, external control circuits need to be added.

When the Relay or Resistor Unit in PLC was damaged, the ON/OFF for PLC Outputs cannot be controlled.

The Power system should fitted with lightning protection device, to prevent the PLC interfaces, like power/signal inputs and controller outputs from being damaged by lightning over-voltage.

Precautions for Installation

Do not install PLC at places with: dust, smut, conductive dust, corrosive gas, combustible gas, vibration and shock. Do not expose the PLC in environment like high temperature, moisture condensation and raining. Thunder strike, fire or disoperation can also cause product damage.

Prevent the metal filing and cable outlet falling into the PLC ventilation hole when wiring and tightening the screws, for it may cause fire, failure and misoperation.

After installing PLC, there must be no sundries in case of fire, fault and misoperation caused by bad heat dissipation.

Hot plugging is not allowed, shield cable must be used to increase immunity of inference.

In situations with serious interference, shielded twisted pair is recommended to improve the anti-interference ability of the system.

Precautions for Wiring

Before Installation and wiring, all external powers must be cut-off, otherwise there maybe electric shock and device damage.

Cover the terminal plate before powering on, follow the instructions in this manual to connect power.

PLC in/out signals wires cannot be parallel with other high voltage or interference lines, they should be layout in separate slots.

Separate the PG terminal on CPU with that of high voltage.

Operation and Maintenance

Do not touch the terminals with power on, in case of electric shock and device damage.

Clean and tighten screws, connect and dismantle cables or expansion modules after power off, or it may cause electric shock and device damage

Do not dismantle the controller in case of damaging the internal parts.

Please reading this manual before modifying, testing, startup and stop programming.

Product Discard

The combustion of electrolytic capacitor on PCB can cause explosion, the main material on PLC is plastic which would generate toxic gas when burnt.

Please follow the local environment regulations to dispose the abandoned products.

Preface

Thank you for choosing CTH200 PLC. Please reading this document carefully before using, so that you could use it more securely and utilizing the abundant functions.

This manual concludes hardware specifications, features and installation of CTH200 PLC and its expansion modules, besides, you can also refer to the instructions set and order number.

Contents abstract:

- ❑ Chapter 1 describes the characteristics of CTH200 series PLC;
- ❑ Chapter 2 show you how to use CTH200 PLC;
- ❑ Chapter 3 introduces the installation methods and dimensions of CTH200 series PLC;
- ❑ Chapter 4 introduces the specifications of CTH200 series PLC;
- ❑ Chapter 5 describes the networks and communication protocols of CTH200 series PLC;
- ❑ Chapter 6 illustrates the main features of CTH200 series PLC by some examples;
- ❑ Chapter 7 describes the Power budget calculation of CTH200 system;
- ❑ Chapter 8 provides the fault diagnosis for system Runtime;
- ❑ Appendix provides application examples of multiple expansion modules and specified libraries, and all the products order number.

Applicable Objects

All information about TrustPLC CTH200 PLC installing and debugging are oriented for Engineers, Installation technician, maintainer and automation qualified electrician.

Online Help

For related product materials and technical support, please visit: http://www.co-trust.com

Reversion History

Date	Version	Reversion description
June 2019	V1.10	• Delete original chapter 5.1 and 5.2. • Add chapter 5.5, 5.6 and 5.7. • Add chapter 4.7.4. attach the order info of the PN module and components in appendix Q • Add appendix N, add appendix O to replace original chapter 3.4 • Delete SDO instruction

Catalogue

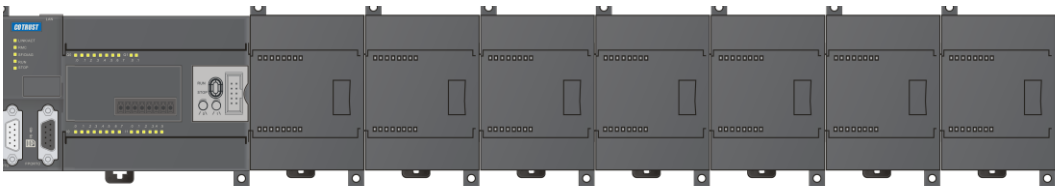
1 Product Overview

CTH200 series PLC is a kind of automatic controller product introduced by COTRUST, which aiming at the rapidly developing OEM market in China. Fuled by the abundant experience of COTRUST engineers and the quick response to marketing demand of industry 4.0, it owns traits from rich-model to Ethernet to excellent performance. CTH200 series PLC will not just offering OEM clients with cost-effective solution of micro automation, but speed the development process of industry 4.0 in China.

CTH200 is a general terms for all CPU products(nine types in all), including CPU H224, CPU H226L, CPU H224X, CPU H226XL and CPU H228XL(the CPU H228XL is just relay-output while others support both relay and transistor). Supporting for variety of modules, such as digital I/O modules, analog I/O modules, temperature sampling modules and so on gives you flexible choice in front of industrial automation problem.

CTH200 PLC communicates with computer via Ethernet port or RS485 port, and programs on MagicWorks PLC (version V2.08 or above) or MicroWin(not suit for Ethernet port).

CTH200 system can be made up by one single CPU as well as one CPU with up to 7 various optional expansion modules.



1.1 CPU Introduction

CTH200 PLC is small robust programmable controller which encapsulating not only micro-processor but also multiple digital I/O points, which can expand up to 7 modules via communication bus.

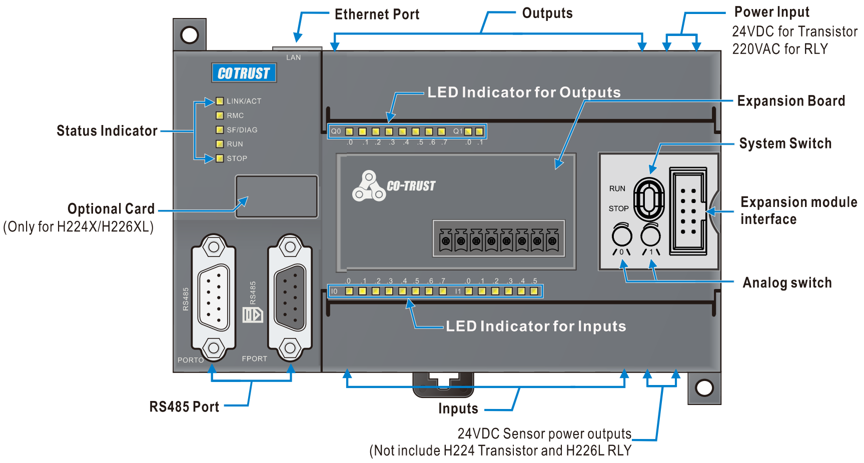


Figure 1-1 CTH200 PLC structure diagram

Table 1-1 CTH200 CPU

Specifications	Order number
CPU	
CPU H224 12KB program/8KB data, 24VDC power supply, 14DI/10DO transistor-source outputs, 0.5A, 1 PPI port, 1 freeport, 1 Ethernet port,3x50KHz motion outputs	CTH2 214-1AD33-0X24
CPU H224 12KB program/8KB data, 220VAC power supply, 14DI/10DO relay outputs, 2A, 1 PPI port, 1 freeport, 1 Ethernet port	CTH2 214-1BD33-0X24
CPU H226L 12KB program/8KB data,24VDC power supply, 24DI/16DO transistor-source outputs, 0.5A, 2 PPI/freeports,1 Ethernet port, 3x50KHz motion outputs	CTH2 216-2AD33-0X40
CPU H226L 12KB program/8KB data, 220VAC supply, 24DI/16DO relay outputs, 2A, 2 PPI/freeports,1 Ethernet port	CTH2 216-2BD33-0X40
CPU H224X 16KB program/108KB data, 24VDC supply, 14DI/10DO transistor-source outputs, 0.5A, 1 PPI port, 1 freeport, 1 Ethernet port, 2x50KHz outputs (Pulse/Dir or PTO/PWM)	CTH2 214-1AX33-0X24
CPU H224X 16KB program/108KB data, 220VAC supply, 14DI/10DO relay outputs, 2A, 1 PPI port, 1 freeport, 1 Ethernet port	CTH2 214-1BX33-0X24
CPU H226XL 72KB program/110KB data, 24VDC supply, 24DI/16DO transistor-source outputs, 0.5A, 2 PPI/freeports,1 Ethernet port, 2x50KHz outputs (Pulse/Dir or PTO/PWM)	CTH2 216-2AX33-0X40
CPU H226XL 72KB program/110KB data, 220VAC supply, 24DI/16DO relay outputs, 2A, 2 PPI/freeports, 1 Ethernet port	CTH2 216-2BX33-0X40
CPU H228XL 96KB program/110KB data, 220VAC supply, 36DI/24DO relay outputs, 2A, 2 PPI/freeports, 1 Ethernet port	CTH2 218-3BX33-0X60

CPU features:

Reliability and Stability

- Resist ESD/EFT, tested by strict industrial environment, CE certified
- Three proofings processed, suited for various harsh industrial environment
- User program and data can be kept permanently
- Multiple password protection, one-way downloading for kernel program, permanent confidential
- Anti-thunder for COM ports, with high reliability
- With build-in real-time clock

Robust communication capability

- Integrate 2 or 3 COM ports
- Support MPI, PPI, DP, Freeport, ModBus and Ethernet communication protocols
- Exchanging 200Bytes per read-write operation, enhanced networking communication

High intelligence

- Integrated parameter self-tuning fuzzy logic algorithm and temperature control PID library, with high accuracy and dynamic performance
- Build-in plenty of integrated functions with simplified programming and flexible control
- High-speed closed loop can be delivered to support some high-speed system application
- Abundant motion control functions, suitable for some synchronization and positioning applications

High speed and high-capacity

- Logical operation speed: 0.22μs for H224 and H226L ; 0.15μs for H224X/H226XL/H228XL
- Floating-point calculation speed: 12μs for H224/H226L; 8μs for H224X/H226XL/H228XL
- Large program capacity: 12K-72Kbyte
- Large data space: 8K-110Kbyte
- High-speed Input counter: 4 for H224 and H226L; 6 for H224X/H226XL/H228XL
- High speed pulse output: 3 for H224 and H226L; 2 for H224X/H226XL/H228XL

Ultra system expandation

- Analog I/O: 32AI/32AQ for H224/H226L; 194AI/194AQ for H224X/H226XL/H228XL
- Digital I/O: 128DI/128DQ for H224/H226L; 640DI/640DQ for H224X/H226XL/H228XL
- Expansion I/O modules up to 7

1.2 Expansion Modules

CTH200 CPU offers local I/Os, you can also use expansion modules for more additional I/Os and communication functions. The various expansion modules include digital I/O modules, analog I/O modules, temperature collection modules and DP communication modules etc. These modules consist of different I/O number, used for configuring various scale of I/Os with high cost performance. All modules are installed with standard DIN35 rail.

Table 1-2 Specifications of expansion modules

Specifications	Order number
SM221 digital input module, 8 inputs, 24VDC	CTH2 221-1BF32
SM221 digital input module, 16 inputs, 24VDC	CTH2 221-1BH32
SM221 digital input module, 32 inputs, 24VDC	CTH2 221-1BL32
SM222 digital output module, 8 transistor outputs, 24VDC, 0.5A(with Output Protection)	CTH2 222-1BF32
SM222 digital output module, 16 transistor outputs, 24VDC, 0.5A(with Output Protection)	CTH2 222-1BH32
SM222 digital output module, 32 transistor outputs, 24VDC, 0.5A(with Output Protection)	CTH2 222-1BL32
SM222 digital output module, 8 relay outputs, 2A	CTH2 222-1HF32
SM222 digital output module, 16 relay outputs, 2A	CTH2 222-1HH32
SM223 digital input/output module, 4*24VDC inputs, 4 transistor outputs,	CTH2 223-1BF32

24VDC, 0.5A(with Output Protection)	
SM223 digital Input/Output module, 8*24VDC inputs, 8 transistor outputs, 24VDC, 0.5A (with Output Protection)	CTH2 223-1BH32
SM223 digital Input/Output module, 16*24VDC inputs, 16 transistor outputs, 24VDC, 0.5A (with Output Protection)	CTH2 223-1BL32
SM223 digital input/output module, 4*24VDC inputs, 4 relay outputs,2A	CTH2 223-1HF32
SM223 digital input/output module, 8*24VDC inputs, 8 relay outputs, 2A	CTH2 223-1PH32
SM223 digital input/output module, 16*24VDC inputs, 16 relay outputs, 2A	CTH2 223-1PL32
SM231 Analog Input module with 4 inputs, 0~20 mA current input or ±5V, ±2.5V, 0~10V, 0~5V voltage input, isolated 12 bit resolution	CTH2 231-0HC32
SM231 high precision analog input modules, 8 inputs, voltage input, and opto-isolator 16 bits precision.	CTH2 231-0HF32
SM231 high precision analog input modules, 8 inputs, current input, opto-isolator 16 bits precision.	CTH2 231-1HF32
SM231 analog voltage input module with 8 inputs, ±2.5V, 0~10V, 0~5V voltage input or optional 0~20 mA current input, isolated 12 bits resolution	CTH2 231-5HF32
SM231 thermal resistance temperature Input module, 2 RTDs, isolated 16 bits resolution	CTH2 231-7PB32
SM231 thermal resistance temperature Input module, 4 RTDs, isolated 16 bits resolution	CTH2 231-7PC32
SM231 thermal resistance temperature Input module, 4xTC, J/K/R/S/T/E/N, isolated 16 bits precision	CTH2 231-7PD32
SM231 thermal resistance temperature Input module, 8xTC, J/K/R/S/T/E/N, isolated 16 bits precision	CTH2 231-7PF32
SM231 thermocouple PID module, 4 J/K, with intelligent PID, isolated 16 bits precision	CTH2 231-7TD32
SM231 thermocouple PID module, 8 J/K, with intelligent PID, isolated 16 bits precision	CTH2 231-7TF32
SM231 8 current inputs, 0~20mA/4~20mA, with intelligent PID, isolated 16 bits precision	CTH2 231-7HF32
SM231 hybrid temperature Input module, 2 NTC or PT100, 2 0~20mA current or ±5V, ±10V, 0~10V, 0~5V voltage input, isolated 16 bit precision	CTH2 231-7ND32
SM231 thermal resistance temperature Input module, 8NTC/PT100, isolated 16 bits precision	CTH2 231-7NF32
SM231 weighing module, 1 sensor input, 50Hz sample frequency, 0.01% accuracy, 6VDCx150MA excitation power output, isolated 16 bits precision	CTH2 231-7WA32
SM232 Analog Output Module, dual ±10V supply or 0~20mA current outputs, isolated 12 bits voltage or 11 bits current precision	CTH2 232-0HB32
SM232 Analog Output Module, quad ±10V supply or 0~20mA current outputs, isolated 12 bits voltage or 11 bits current precision	CTH2 232-0HD32
SM235 analog Input/Output Module, 4 voltage/current inputs, 1 voltage/current output, isolated 12 bits voltage or 11 bits current precision	CTH2 235-0KD32
SM253 motion control module, 2 uniphase or AB phase HSC input,	CTH2 253-1BH32

200KHz, 2xPTO/PWM outputs, 200KHz, COTRUST motion control libraries	
SM277A Profibus DP Slave Interface Module, 12M traffic rate, photoelectric isolated	CTH2 277-0AA32
SM277B Profibus DP Slave module, 1.5Mbps traffic rate, photoelectric isolated	CTH2 277-0AB32
SM277C CAN Slave module, 8DI/6DO, photoelectric isolated, up to 7 extendable modules	CTH2 277-0AC32

Table 1-3 BD Expansion Boards

Module Name	Specifications	Order number
EBH AMS-03	Analog I/O Expansion Board, 2*12 bits precision inputs, 1*12 bits precision voltage/current output	CTH2 AMS-03S1-EB
EBH-AMS-06	Analog I/O Expansion Board, 4*12 bits precision voltage inputs, 2*12 bits precision voltage output	CTH2 AMS-06S1-EB
EBH-AMS-06	Analog I/O expansion board, 4*12 bits precision voltage inputs, 2*12 bits precision current output	CTH2 AMS-06S2-EB
EBH CAN-01	CAN Master communication expansion board, 1Mbps, photoelectric isolated	CTH2 CAN-01S1-EB
EBH PWM04	Charging guidance expansion board, two-channel voltage input, single-channel PWM output.	CTH2 PWM-04S1-EB

Notice

H224/H226L do not support CAN-01 master expansion board.

Features of expansion modules:

Filter technology

All the analog modules integrate CPU itself, with high stability due to advanced filter technology.

Digital module

All the digital I/O modules are equipped with optoelectronic isolation and disturbance rejection.

Temperature acquisition module

Integrated bus isolation, power isolation and interchannel isolation, high immunity from interference and high sampling accuracy, with intelligent fault diagnosis.

Intelligent module

The PID modules like Temperature Control can promote the respond speed of program executing.

Communication module

The expansion board for CAN master and the DP Slave module can significantly increase the interconnectivity and communication performance.

Analog I/O expansion board

The analog I/O expansion board has 2x12 bits voltage inputs and 1x12 bit voltage/current input.

1.3 Maximum System Configuration

- Up to 7 expansion modules per PLC connection
- Digital image register: 128DI/128DQ for H224/H226L; 640DI/640DQ for H224X/H226XL/H228XL, in which including private image for CAN communication.
- Analog image register: 32AI/32AQ for H224/H226L; 194AI/194AQ for H224X/H226XL/H228XL, in which including private image for CAN communication.

1.4 Network Architecture

The typical network architecture for CTH200 system is shown as following:

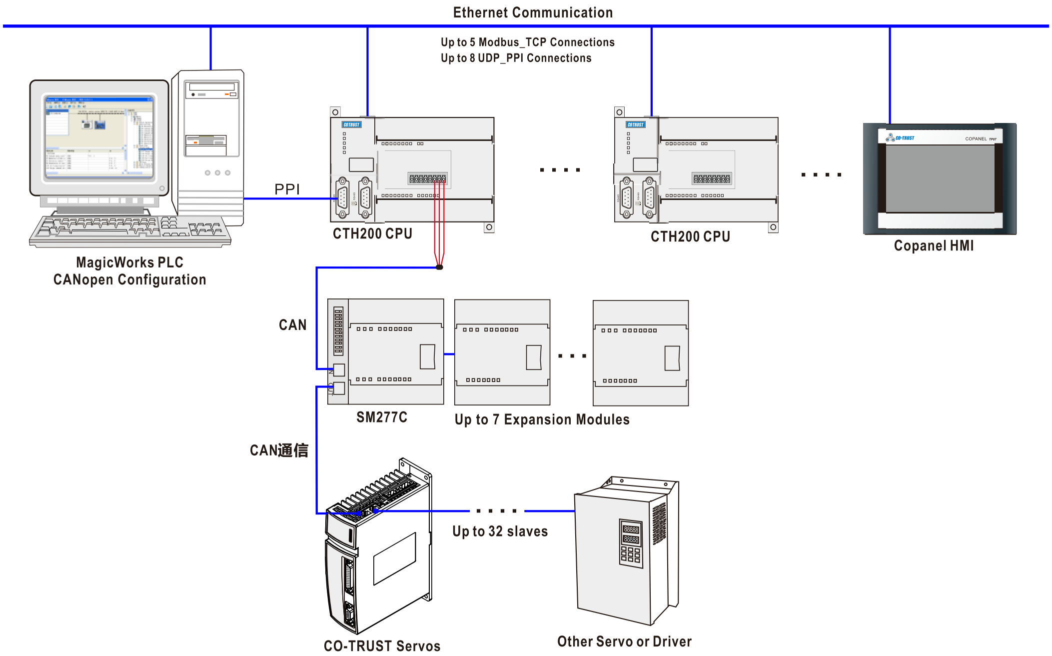


Figure 1-2 CTH200 Network Architecture

Note:

➢ Inter-PLC communication

H224/H226L: 8 UDP_PPI connections, max. 200 bytes per connection. 5 MODBUS_TCP connections, maximum 240 bytes per connection.

H224X/H226XL: 8 UDP_PPI connections, 4 masters and 4 slaves, max. 200 bytes per connection. 4 MODBUS_TCP connections, including 2 masters and 2 slaves, max. 240 bytes per connection.

H228XL: 8 UDP_PPI connections, including 4 master and 4 slave connections, max. 200 bytes per connection. 6 MODBUS_TCP connections, including 3 masters and 3 slaves, max. 240 bytes per connection

➢ PLC connect with HMI (or third-party software), with HMI or the third-party software as master

UDP_PPI: H224/H226L connect up to 8 HMI while H224X/H226XL/H228XL up to 4.

MODBUS_TCP: H224/H226L can connect up to 5 HMI; H224X/H226XL up to 2 and H228XL up to 3.

➢ CPU can access other devices as master, equivalently a master connection for the CPU itself.

➢ CPU can be accessed as slave by other devices, equivalently a slave connection for the CPU itself.

1.6 Standards and Specifications

CTH200 series PLC have been approved for various international and industrial standards, the specifications for operating environment are shown as table 1-4:

Table 1-4 Standards and specifications of CTH200 series PLC

Environmental condition			
Items		**H224/H226L**	**H224X/H226XL/H228XL**
Transport and storage			
Temperature		-25℃ ~ 70℃	-40℃~+85℃
Atmosphere press		1080hPa~660hPa(corresponding hight-1000m ~ +3500m)	
Relative humidity		5%~95%, non-condensation	10%~95%, non-condensation
Falling		1m, 5 times, transport package	
Working environment			
Temperature	Horizontal	0℃~55℃	
	Vertical	0℃~45℃	
Atmosphere press		1080hPa~795hPa(corresponding hight-1000m ~ +2000m)	
Relative humidity		10%~95%, non-condensation	
Pollutant concentration		Lower salt mist, moist and dust fog SO_2<0.5ppm, relative humidity <60%, non-condensation H_2S<0.1ppm, relative humidity <60%, non-condensation	
EMC- interference rejection			
Electrostatic discharge IEC61000-4-2		Contact-discharge: ±4KV (grade A) Air-discharge: ±8KV (grade A)	
Electric fast-transient pulse group IEC61000-4-4		Power line: 2KV, 5KHz(grade A) Signal line: 2KV, 5KHz (I/O coupling clamp, grade A) 1KV, 5KHz (COM coupling clamp, grade A)	
Surge IEC61000-4-5		Power line: 2KV(asymmetric), 1KV(symmetric, grade B)	
Radiofrequency electromagnetic radiation IEC61000-4-3		80MHz~1GHz, 10V/m, 80%AM(1KHz, grade A) 1.4GHz~2GHz, 3V/m, 80%AM(1KHz) 2GHz~2.7GHz, 1V/m, 80%AM(1KHz)	
Radiofrequency conduction interference IEC61000-4-6		0.15MHz~80MHz, 10V/m, 80%AM(1KHz, grade A) 15KHz~150KHz, 10V/m, 80%AM(1KHz)	

DC input short-interrupt and voltage change IEC61000-4-29	Short-interrupt: 10ms Voltage change: 80%~120%, 100ms
Antidumping vibration wave IEC61000-4-12	Power line: 1KV Digital I/O(0(24V or higher): 1KV
EMC – conduction and radiated emission	
Radiation(noise) EMI EN55011, type A 1 group	Measuring distance: 10m 30MHz~230MHz, < 40dB(uV/m) peak value 230MHz~1000MHz, < 47dB(uV/m) peak value
Conduction(noise) AC power interference EN55011, type A 1 group	Measuring distance: 10m 0.15~0.5MHz, < 79dB(uV/m)peak value;< 66dB(uV/m) average 0.5~5MHz, < 73dB(uV/m)peak value;< 60dB(uV/m) average 5~30MHz, < 73dB(uV/m)peak value;< 60dB(uV/m) average
Environment test index	
Hot operation IEC60068-2 Cold operation IEC60068-2	60℃ 16h -10℃ 16h
Hot startup IEC60068-2 Cold startup IEC60068-2	60℃ 2h -10℃ 2h
Hot/cold cycle operation IEC60068-2	3h of dwell time under -10℃~60℃, temperature rising speed 1℃/min, 2 cycles
Hot storage IEC60068-2 Cold storage IEC60068-2	70℃ 72h -40℃ 72h
Hot/cold shock IEC60068-2	3h of dwell time under -40℃~70℃, temperature changing time<1min, 5 cycles
Hot and humid IEC60068-2	40℃ 48h
Alter hot and humid IEC60068-2	25℃~55℃ 95%, 2 cycles
Sine vibration(bare machine) IEC60068-2	5~150Hz, 0.05G^2/Hz 150Hz~500Hz -3dB/oct, 1 hour/axis; X, Y and Z 3 axes in all
Strike (bare machine)IEC60068-2	15G, 11ms pulse, 3 times/direction
Hybrid gas erosion test IEC60068-2-60	H2S: 0.1ppm, NO2: 0.2ppm, CL2: 0.02ppm, temperature: 30℃, humidity: 75%, period: 4 days
High-press insulation test index	
24V/5V standard circuit 110V/220V circuit to earth 110V/220V circuit 110V/220V to 24V/5V circuit	500 VAC 1500 VAC 1500 VAC 1500 VAC

2 Getting Started

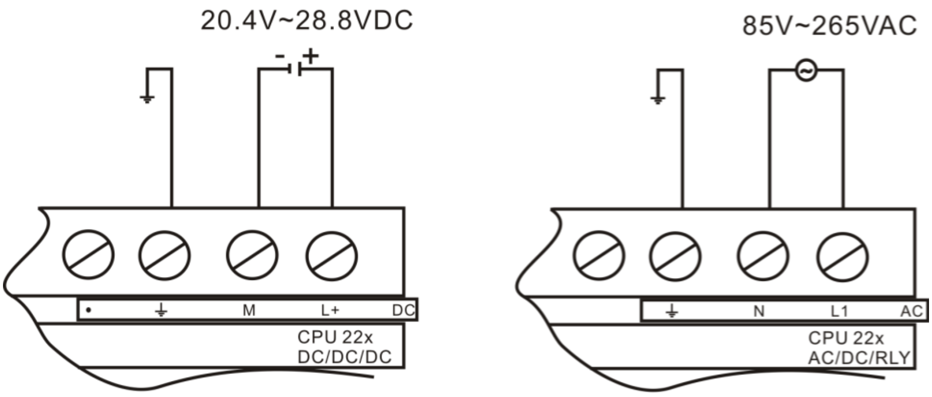
Example in this chapter shows you how to program and compile as well as its connection with CTH200 series PLC in MagicWorks PLC.

2.1 Connecting to CTH200 PLC

- After connecting the CTH200 CPU to program device with the RS485 cable, power it on.
- CTH200 series PLC supports Ethernet communication, please connect the CTH200 CPU to program device with Standard Ethernet cable.

Power supply for CTH200 CPU

The following Figures are diagram for 2 models of CPU:



DC Power AC Power

Figure 2-1 Power up the CTH200

Warning

Don't install and wire the CTH200 series PLC with power on, faulty operation can cause serious damage for machines and personal death. Ensure disconnecting power while install or disassemble any electric device.

Connecting with RS485 cable

Figure 2-2 shows how to connect CTH200 CPU with programming device by RS485 cable.

Notice

Please install the drive for PLC program cable referring to Appendix *K Instructions of ETHERNET_SET.*

1. Connect USB port of the Cable to the communication port of program computer (PC), thus can generate a virtual serial port.
2. Connect RS485 interface of the cable to the Port0 or Port1 of CTH200 CPU.

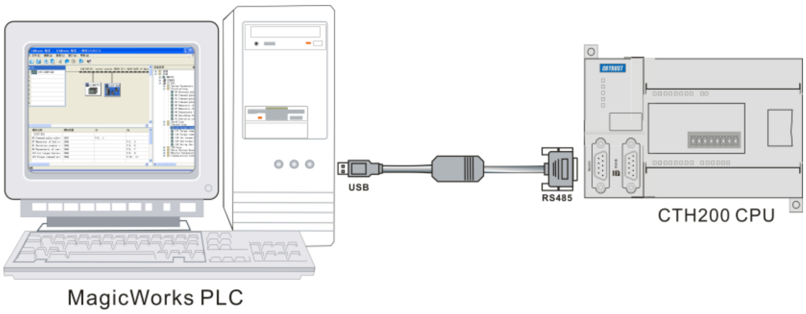


Figure 2-2 Cable connection between CTH200 PLC and PC

PLC connection with net cable

Figure 2-3 illustrates how to use standard net cable to connect CTH200 PLC with PC.

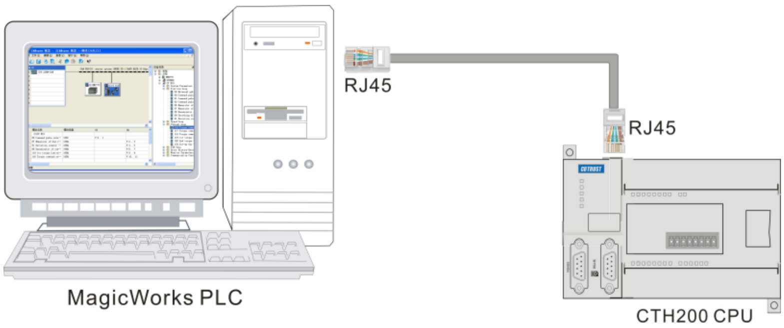


Figure 2-3 connection between CTH200 PLC&PC using net cable

MagicWorks PLC

Double-click the icon to start MagicWorks PLC software, select “File ->New” to create a new project. Figure 2-4 shows a project tree, in which you can click icons to open MagicWorks PLC components.

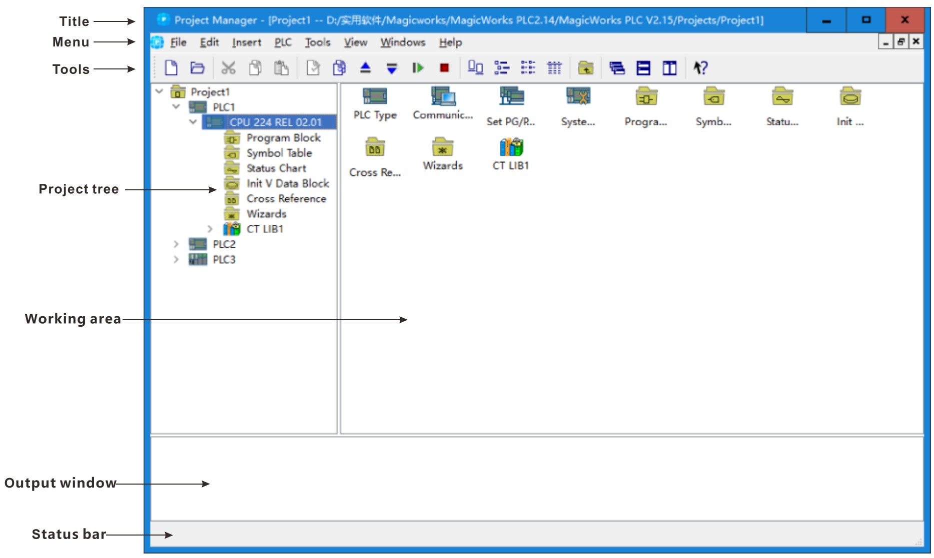


Figure 2-4 MagicWorks PLC Operation interface

2.2 Communicate with CTH200

2.2.1 Serial Port Communication Mode

The steps for PPI communication with MagicWorks PLC:

1. Click the “Set PG/PC Interface” in the “Communications” screen. Select the interface, the default is “PC/PPI Cable (PPI)”.

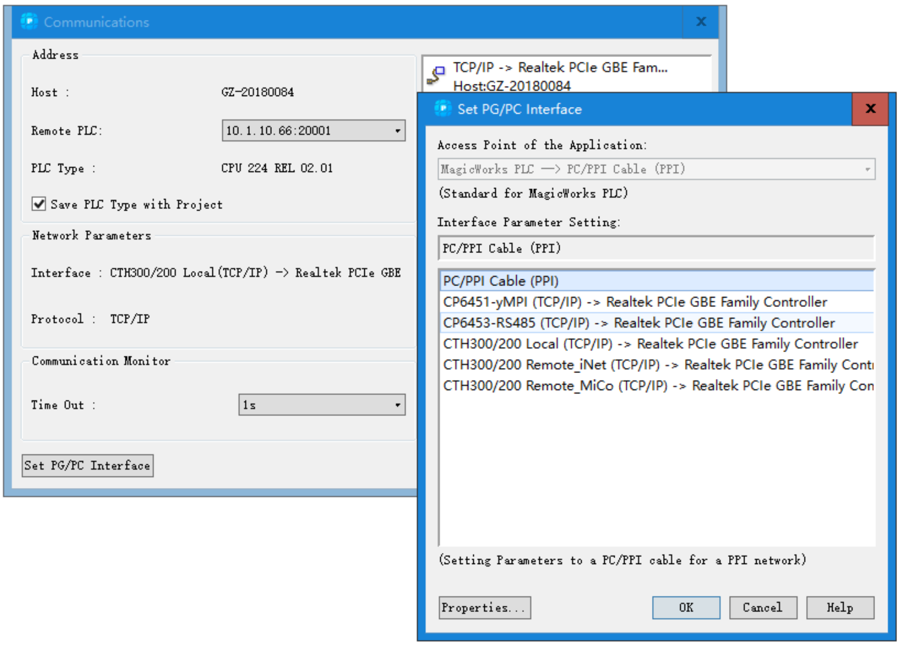
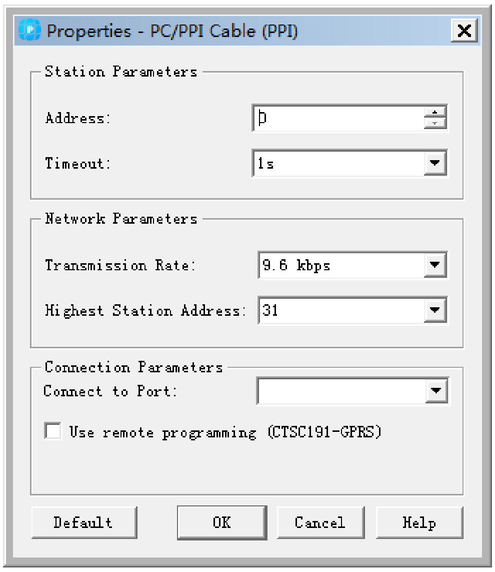


Figure 2-5 MagicWorks PLC serial port communication interface

2. Select the “PC/PPI Cable (PPI)” in “Set PG/PC Interface”, then click the “Properties” button in this dialog to set the communication parameters

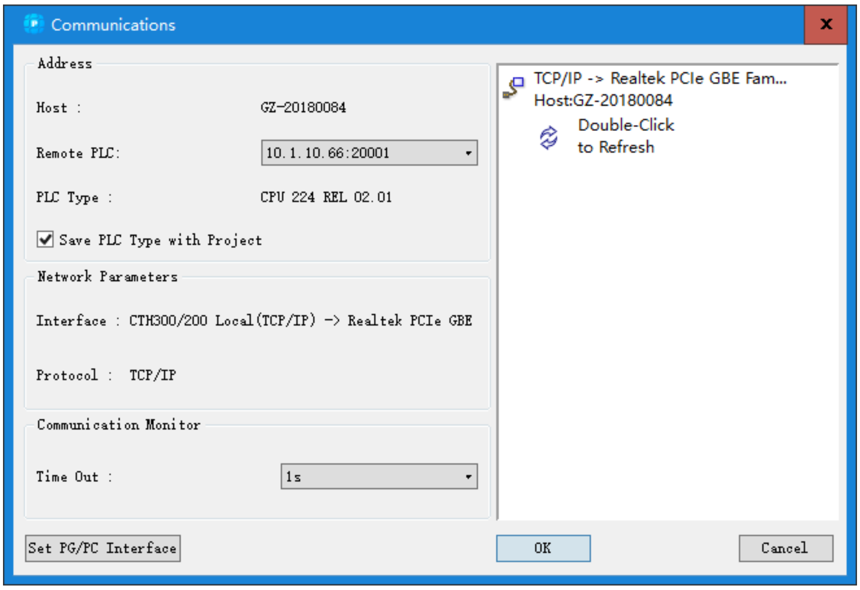


1) PC/PPI Address: 0

2) Interface: COM1

3) Transmission rate: 9.6Kbps

3. Click OK to return to the “Communications” page. Then double-click Refresh to find the PLC.



If communication failed, Please do examine as the following steps:

1) Check wiring

Please use the cable provided by COTRUST (Order No. CTS7-191-USB) and ensure the connector remained intact. Switch PLC to STOP when the communication protocol had changed previously.

2) Check the drive

The cable provided by COTRUST must have matched drive, please obtain the drive from the following site: http://www.co-trust.com

3) Check the communication settings

- Select the PPI protocol (PC/PPI Cable).
- Ensure the selected COM port not hold by other programs.
- Select the suitable baud rate.

2.2.2 Ethernet Communication Mode

The steps for TCP/IP communication with MagicWorks PLC:

1. Click the “Set PG/PC Interface” in the “Communications” screen
2. Choose “CPU300/200 Local (TCP/IP) -> Realtek PCIe GBE Family Controller”

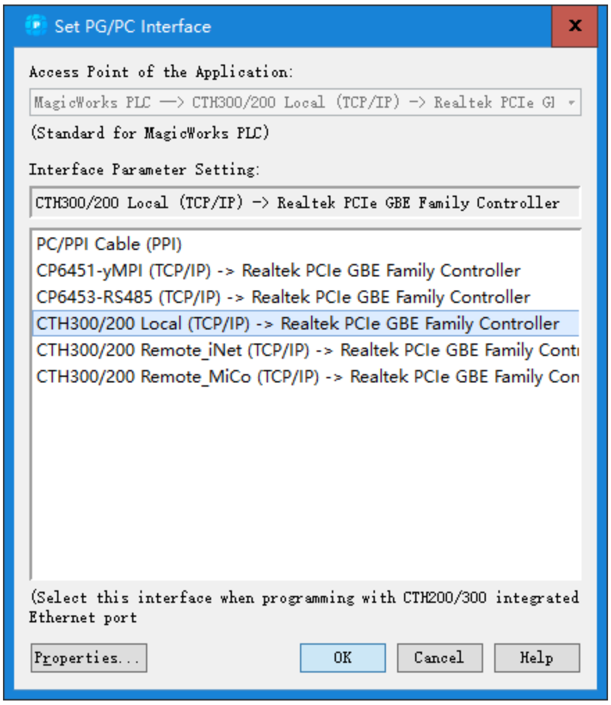
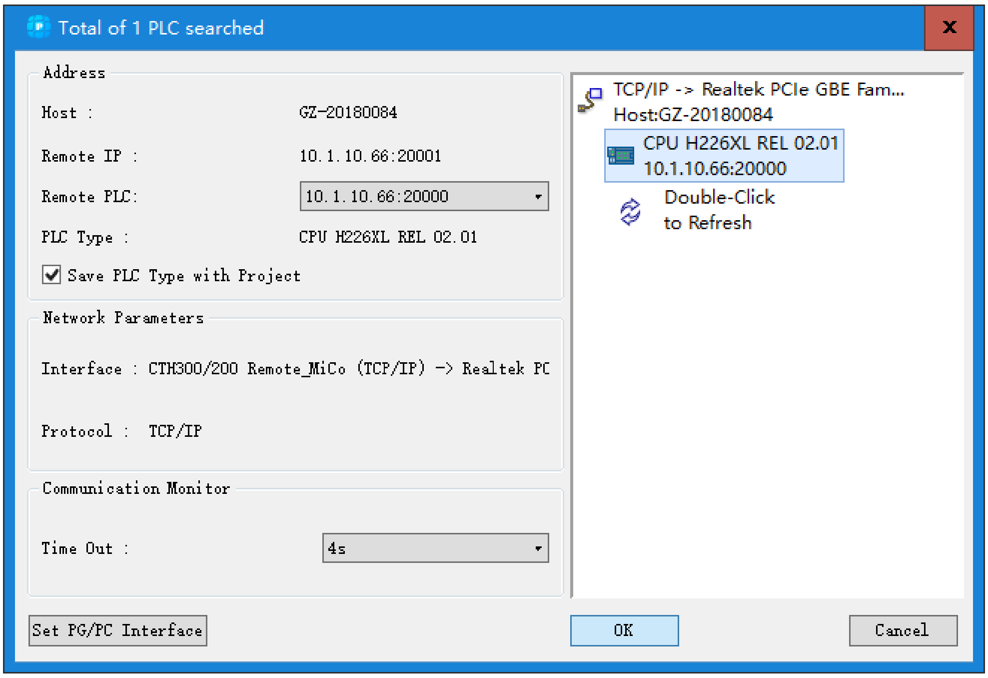
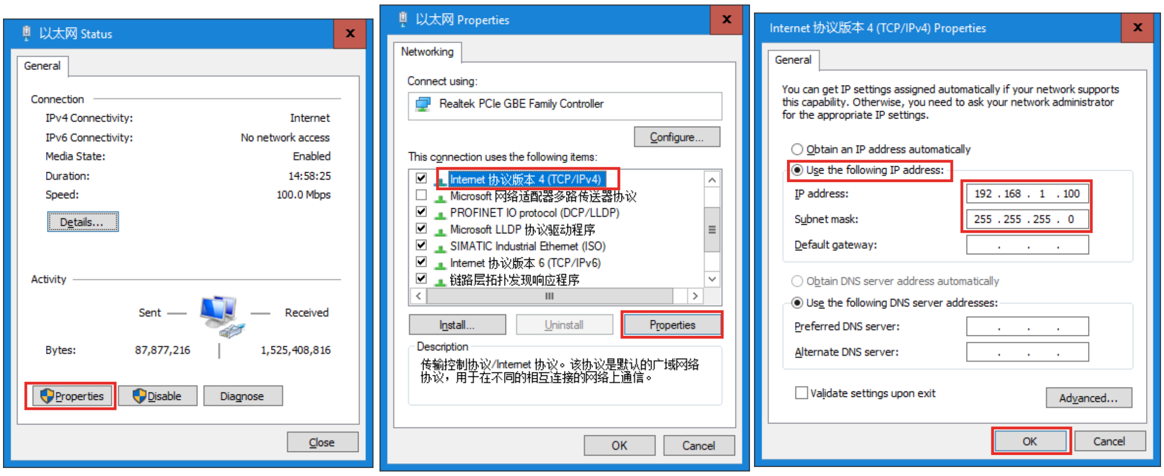


Figure 2-6 MagicWorks PLC Ethernet communication interface

3. Establish communication with CTH200 PLC: Double-Click the Refresh icon in the following dialog, MagicWorks PLC will search for and show the connected CTH200 PLC.



Users can choose the CTH200 station and click OK after searching. If the MagicWorks PLC hasn not found the CTH200 CPU, please check wring first and then validate the Communication settings, after which repeat the above steps. When the PLC communication has established, The IPs for PC and PLC should be in same segment but can not identical, as shown in the following figure, IP for PLC searched by PC is 192.168.1.202, then the PC IP should be 192.168.1.XXX (in which XXX ranges from 1 to-255). Please set PC IP as following procedures:

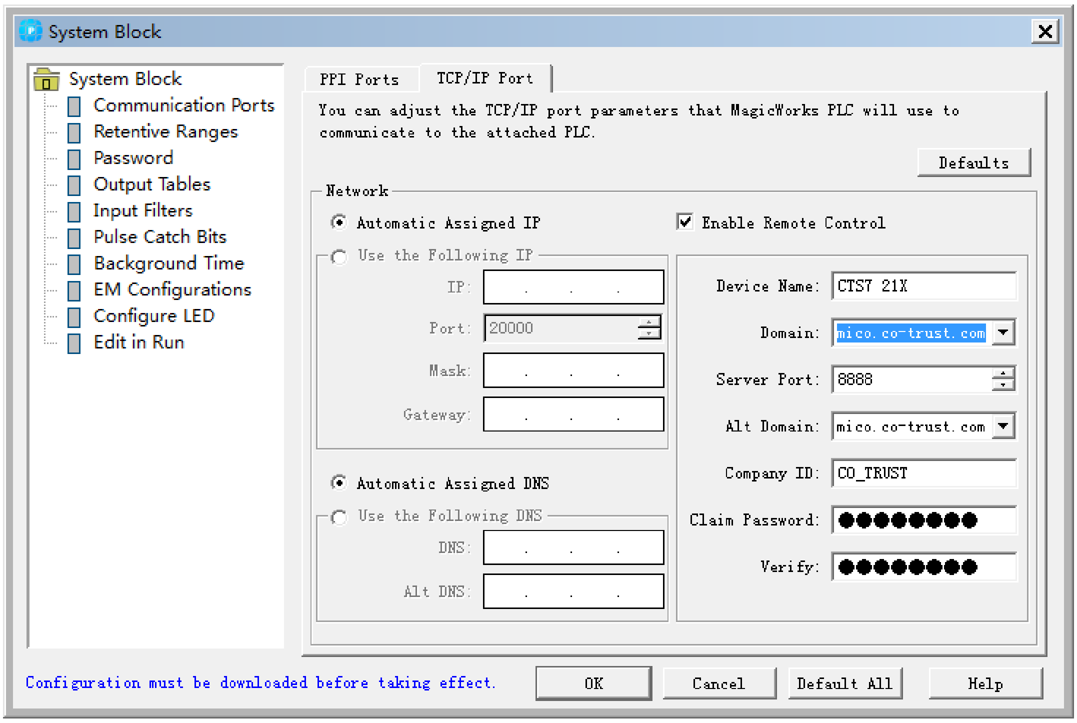


Step 1 **Step 2** **Step 3**

2.2.3 Modify IP Address in System Block

After establishing communication connection, you can modify the IP address of CPU through the System Block in MagicWorks PLC.

Note: downloading the System Block into PLC to make the modification effective. Then you need to search PLC again to establish connection (refer to chapter *2.2.2 Ethernet Communication Mode*)



2.3 Create Program Network

This section describes how to create, download and run PLC program for CTH200 PLC. The following program examples use 3 instructions to shift left 1 bit for value transferred in QB0 at 1s interval. This example uses the ladder diagram editor to enter the program, and describes the program compilation, download and operation process.

Pictures below shows the ladder diagram and statement list to explain logic relationship in the program.

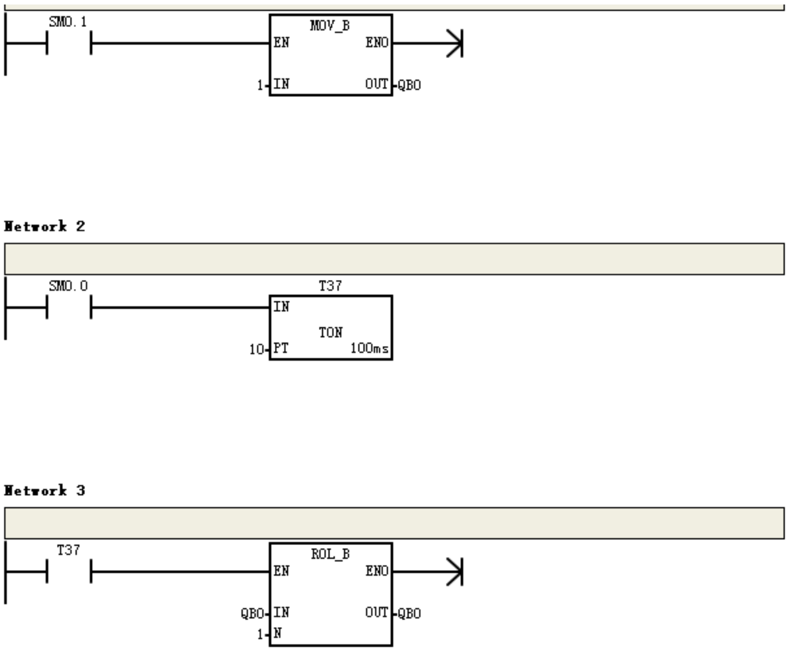


Figure 2-7 Example of CTH200 program

Segment comment:

Network 1:
LD SM0.1 // Enable by first scan of SM0.1
MOVB 1, QB0 // Transfer 1 into QB0
Network 2:
LD SM0.0 // Enable signal
AN T37 // Circulate timer signal
TON T37, 10 // Set timer T37, Time 100ms x 10 = 1s
Network 3:
LD T37 // Set enable pulse via T37
RLB QB0, 1 // QB0 shift 1 bit left, with the PLC LEDs light on at 1s interval.

2.3.1 Editing

Click the Program Block to open the program editor, as shown in figure 2-8. User can drag&drop the ladder instructions into program editor, or use the shortcut for available instructions directly.

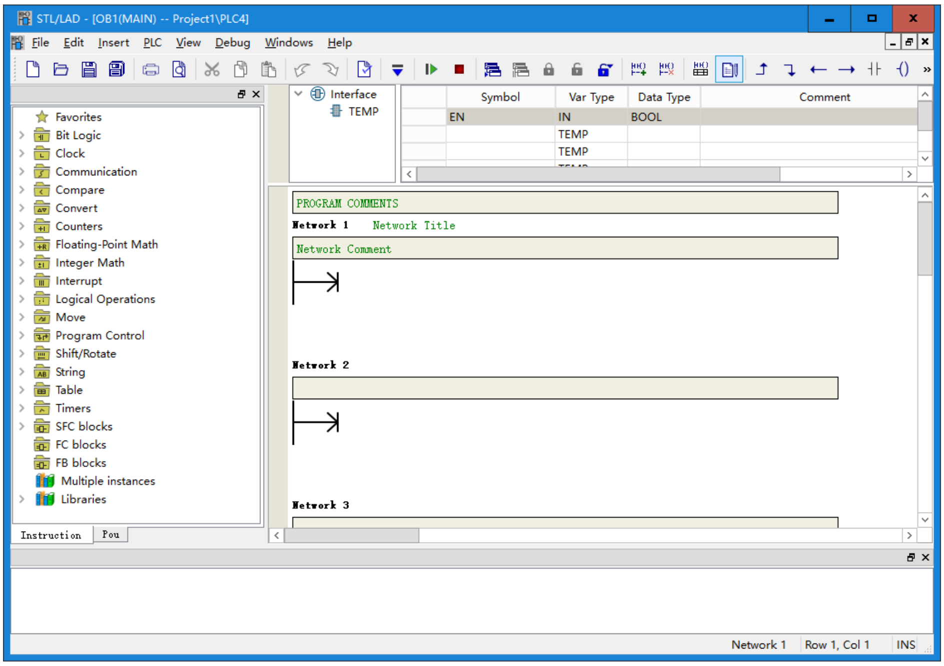


Figure 2-8 Program editor

2.3.2 Compiling

The program needs to be compiled when editing completely:

1) Select menu command “PLC -> Compile” or click Compile button 

2) The output window at bottom will show status for compiling, if there occurs error, users can click the prompt to view details

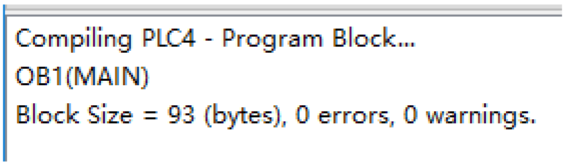


Figure 2-9 Program compile result

Save project:

- ➢ Select menu command “File -> Save as”.
- ➢ Enter the required project name in “Name” dialog.
- ➢ Select the required project Storage.

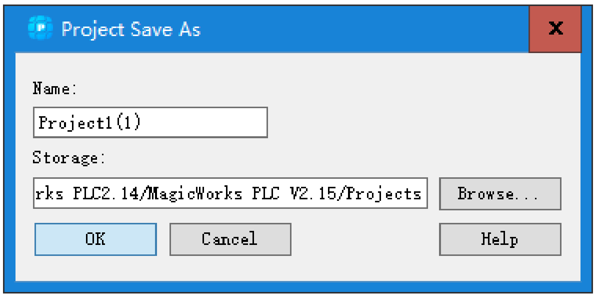


Figure 2-10 Project Storage

2.3.3 Downloading

Click the download icon in toolbar or select the menu command “PLC -> Download” to download the program into CTH200, as shown in Figure 2-11 and 2-12. CTH200 PLC support read/write online, which means user can write instructions or parameters into PLC during Running, without switching to STOP.

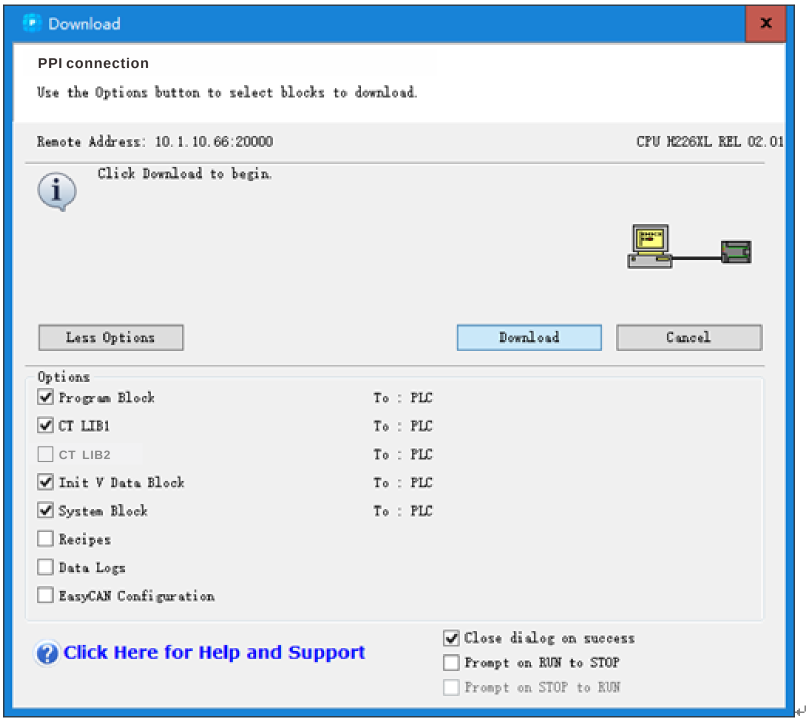


Figure 2-11 PPI connection to download program

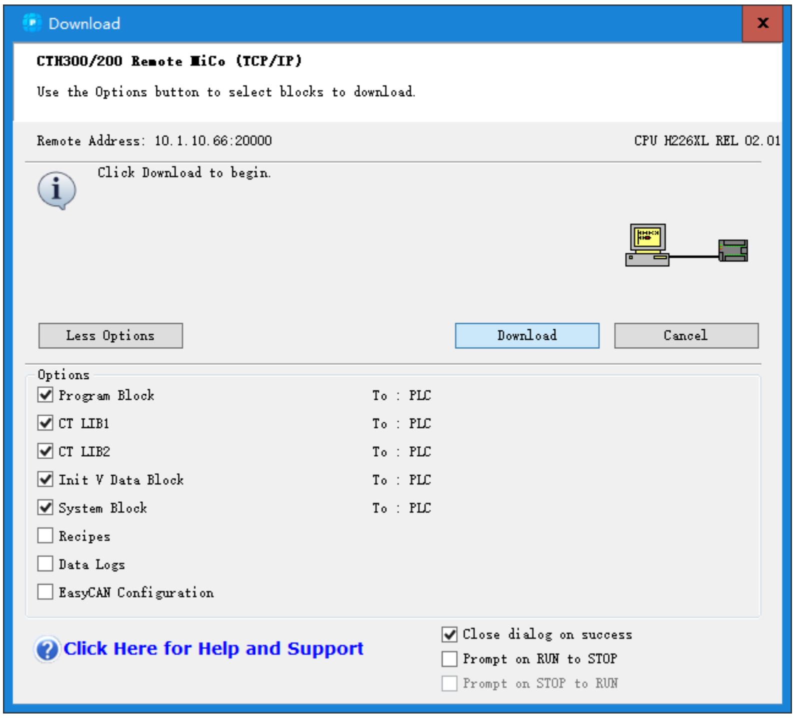


Figure 2-12 Ethernet connection to download program

2.3.4 Running

MagicWorks PLC can switch the CTH200 PLC into RUN mode and execute program when the Mode Switch of PLC set to RUN:

1) Click the RUN icon in toolbar or select the menu command "PLC -> RUN".

2) Click Yes to enter RUN mode and the CPU would execute program, Q0.0-Q0.7 will light on circularly at 1s interval, which means left shaft the value circularly in Q memory.

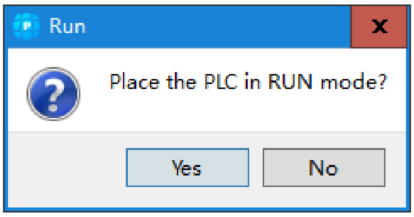


Figure 2-13 Run program

Users can monitor the program via project tree -> state table. MagicWorks PLC would show the appropriate results. Click the icon or select menu command "PLC -> STOP" would stop the PLC.

3 Installation

The design of CTH200 PLC is convenient for installation, they can be fixed into backplane of the cabinet via mounting hole, or use the DIN clamp to mount on a standard DIN rail. It's compact structure allow users utilizing space efficiently.

This chapter shows you how to install and wiring the CTH200 PLC.

3.1 Important Notices

CTH200 PLC can be installed on the backplane of cabinet or on the standard DIN rail, vertically or horizontally. Users must observe the following notices:

Isolate the PLC with Heat, HV and Electronic noise

According to the general conventions, PLC with low voltage must be isolated with HV and electronic noise sources.

When mounted on the backplane of cabinet, the PLC should be arranged into lower temperature area of the cabinet to extend its lifetime.

The AC power line, high energy and high switching frequency DC line, low voltage signal line and communication cable should be avoided putting into one slot.

Make room for heat dissipation and wiring

CTH200 series PLC are designed with natural ventilation and heat dissipation, with at least 30mm space above and below the module. Distance between the front plane and back plane must be at least 80mm.

Notice

Comparing with horizontal installation, the max. Ambient temperature allowed with vertical installation should lack for 10℃, and CPU should be installed below all expansion modules.

Enough spaces should be left for cable wiring and connecting when installing CTH200 PLC.

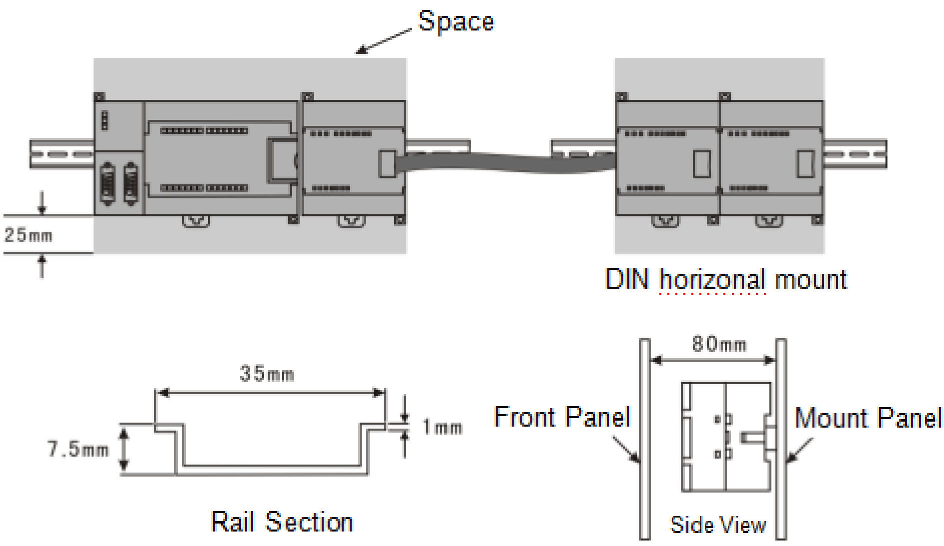


Figure 3-1 Installation diagram

Power budget

CTH200 series PLC has an internal power supply offer 24VDC for itself, the expansion modules and other device.

It also provide 5VDC logic power supply, which can be used for power supply of any expansion in the system. Pay close attention to the system configuration to ensure that the CPU can provide the 5V power for the selected expansion module. If the configuration requires more power than the CPU provides, some expansion modules must be removed or select CPU that can provide more power.

CTH200 series CPU also provide 24VDC sensor power supply (except PSC266), which can provide 24VDC for input, relay coil on expansion module or other power supply. If the required power exceeds budget, an external 24VDC power supply must be added. For specific CTH200 CPU 24VDC sensor power budget, refer to Chapter *6.3 UDP_PPI Communication*

If you need an external 24VDC power supply, make sure that the power supply is not connected in parallel to the sensor power supply of the CTH200 series CPU. To improve electrical noise protection, please connect to the common terminal (M) of different power sources.

Warning

An external 24VDC power supply connected in parallel to CTH200DC sensor power supply can cause a conflict because each power supply seeks to establish its own potential output. This conflict will shorten the service life of one or two power supplies or cause an immediate failure, then uncertain operation of the PLC system. The unpredictable operation may cause personal injury and equipment damage.

3.2 Installation Dimension

Both PLC and expansion modules have mounting holes, which can be installed at back plane conveniently. The following figure shows the Installation Dimension for all PLC and expansion modules.

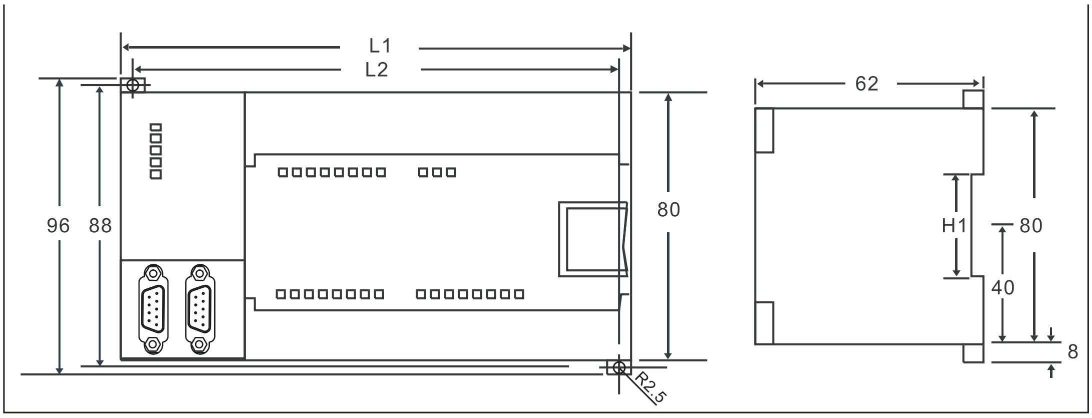


Figure 3-2 CTH200 PLC Installation Dimension (unit: mm)

Table 3-1 Dimensions of CTH200 series PLC and expansion modules

PLC	L1(mm)	L2(mm)
CPU H224/H224X	137	129
CPU H226X/H226XL	196.5	188.5
CPU H228XL	200	195
CTH2 221-1BL32, CTH2 222-1BL32 CTH2 223-1BL32, CTH2 223-1PL32	137	129
CTH2 221-1BF32, CTH2 222-1BF32, CTH2 222-1HF32 CTH2 223-1BF32, CTH2 223-1HF32, CTH2 232-0HB32	46	38
CTH2 221-1BH32, CTH2 222-1BH32, CTH2 223-1BH32 CTH2 223-1PH32, CTH2 231-0HC32, CTH2 235-0KD32 CTH2 231-0HF32, CTH2 231-1HF32, CTH2 231-5HF32 CTH2 231-7HB32, CTH2 231-7HC32, CTH2 231-7PB32 CTH2 231-7PC32, CTH2 231-7PD32, CTH2 231-7PF32 CTH2 231-7TF32, CTH2 231-7TD32, CTH2 231-7HF32 CTH2 231-7ND32, CTH2 231-7NF32, CTH2 277-0AA32 CTH2 277-0AB32	71.3	62.3
CTH2 277-0AC32	90	82

3.3 Installation Method

CTH200 PLC can be installed on standard DIN35 rail or panel.

Prerequisites

Make sure the equipment has been powered off before assembling and disassembling, meanwhile, all related devices must also be powered off.

Warning

Don’t assemble and disassemble the CTH200 PLC and related devices with power, otherwise can cause electric shock or malfunction, even serious damage, injury or death.

First get the CTH200 PLC powered off, then replacing and installing correctly. Meanwhile the direction and location must be correct when replacing modules, or it can result in damage, injury or death.

Caution

Incorrect modules would cause CTH200 PLC program failure.

Install/Disassemble CPU and expansion modules

Following the installation method below:

- Mounting on panel

1) Locate the open holes according to the dimensions on Figure 3-1;
2) Fix the modules on backplane with appropriate screws;
3) If expansion modules are used, connect the flat cable for expansion modules to the extended port below the front cover.

- Mounting on DIN rail

1) Fix the rail on backplane with 80mm distance.
2) Open the DIN clamp below the module, lock the module back on DIN rail.
3) If expansion modules are used, connect the flat cable for expansion modules to the extended port below the front cover.
4) Spin the module to approach the DIN rail, then close the DIN clamp.
5) Check if the DIN clamp fit the rail closely.
6) Don’t push the front of module, you can push the open holes instead to prevent damage.

- Mounting terminal strip

1) Open the front cover of the terminal position.
2) Make sure the module pins are alignment with the holes on terminal strip.
3) Push the terminal strip down into module and lock it up.

Caution

When mounting the CTH200 PLC vertically in high vibration environment, the DIN rail blocks should be used, thus backplane mounting is highly recommended to achieve high vibration protection.

- Disassemble CPU or expansion modules

1) Remove the power of CTH200 CPU.
2) Dismantle all cables and wires on module.
3) Open the front cover to pull up the extended flat cable from neighbor expansion module if there has one.
4) Remove the mounting screw or open the DIN clamp.
5) Remove the CPU and modules.

- Disassemble terminal strip

1) Open the cover on terminal strip.
2) Plug the screwdriver into the slot of terminal block as shown in the following picture.
3) Pull down and pray the strip out.

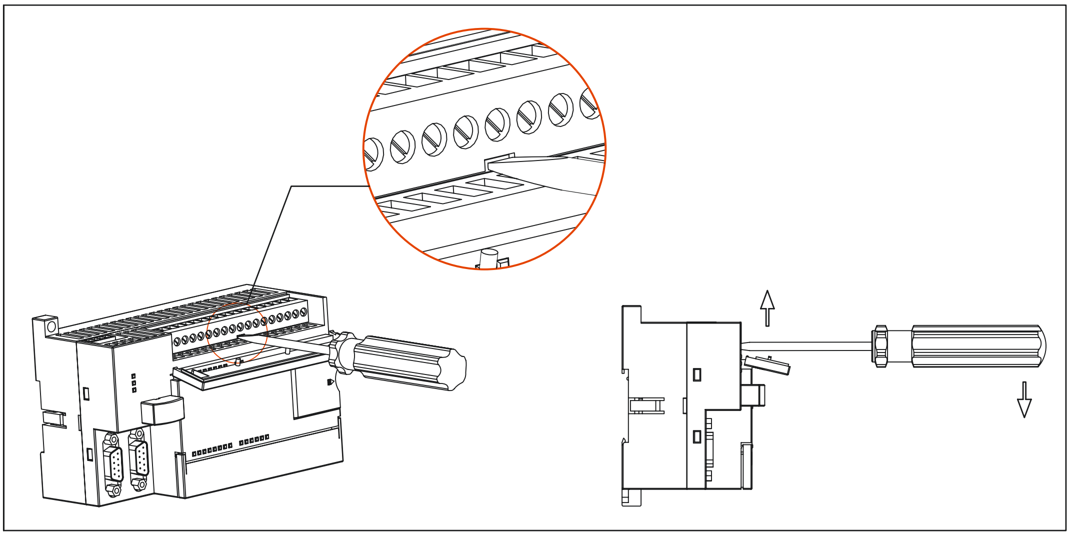


Figure3-3 Dismantle the terminal strip

3.4 Grounding and Wiring

It's important to implement wire and ground correctly for all electrical equipment, which can ensure the best system operation features and provide better noise protection.

CTH200 PLC and related devices must all be powered off before grounding and wiring. The available Electrical coding rules must be obeyed and related safety standards should be followed while installing and operating all devices. Keep contact with regional standard request.

Please always following appropriate safety precautions and ensure that power of CTH200 PLC is cut-off before attempting to install or remove the CTH200 PLC or related equipment.

Warning
Grounding or wiring with power could result in death or serious injury to personnel, and/or damage to equipment.

Safety must be considered when designing ground and wire for CTH200 system in case triggering the device malfunction. Therefore, all safety rules should be implemented to protect operator and device.

Warning

The control equipment may cause the misoperation of the equipment it controls, which may lead to death or serious personal injury and equipment damage. Therefore, the TrustPLC CTH200 system must have the emergency stop function, electromechanical interlock or other redundant safety facilities independent of the CTH200 system.

3.5 Suppression Circuit

Inductive loads should be implemented with suppression circuits to limit voltage rise when the control output turns off. Suppression circuits protect your outputs from premature failure due to high inductive switching currents. In addition, suppression circuits limit the electrical noise generated during switching inductive loads.

Caution

The effectiveness of a given suppression circuit depends on the application, and you must verify it for your particular use. Ensure that all components used in your suppression circuit are rated for use in the application.

DC Outputs and Relay Outputs which Control DC Loads

The DC outputs have internal protection that is adequate for most applications. Since the relays can be used for either a DC or an AC load, internal protection is not provided.

Figure 3-4 shows a sample suppression circuit for a DC load. In most applications, the addition of a diode (A) across the inductive load is suitable, but if your application requires faster turn-off times, then the addition of a Zener diode (B) is recommended. Make sure the Zener diode can provide enough current for output circuit.

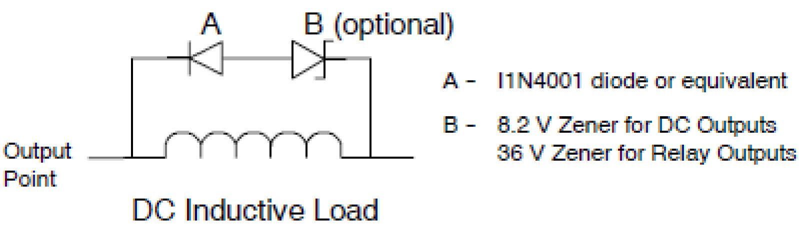


Figure 3-4 Suppression Circuit for DC Load

AC Outputs and Relay Outputs for AC load

The AC outputs have internal protection that is adequate for most applications. Since the relays can be used for either DC or AC load, internal protection is not provided.

Figure 3-5 shows an instance for the suppression circuit of AC load. In most applications, the additional MOV resistor can be used for limit the peak voltage and protect the internal circuit in CTH200 PLC. Make sure the operation voltage for MOV is at least 20% greater than regular line voltage.

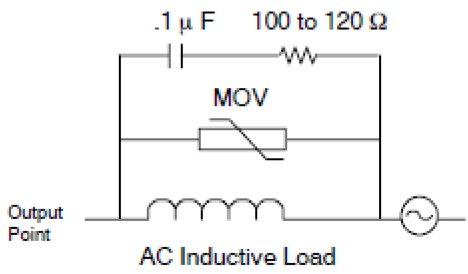


Figure 3-5 Suppression Circuit for an AC Load

4 Technical Specifications

CTH200 series PLC and the expansion modules enjoys different specifications.

4.1 General Specifications

Table 4-1-1 Electrical and environmental specifications

Items	Description
Rated input voltage	AC: 220V, DC: 24V
Voltage range	AC: 85 V~265 V, DC: 20.4 ~ 28.8V
Operation temperature	0 ~ 55℃
Storage temperature	-25 ~ 70 ℃
Ventilation	Inlet Air 50.8 mm below unit
Humidity	5 ~ 95% (non-condensing)
Electrical Interference	Pulse width 50ns, repetition frequency 5kHz, 2,000V voltage peak
Vibration	Frequency 10~57Hz, amplitude 0.1mm, acceleration 1.0g, 10 times each dimension
Impact resistance	15g for 11ms, 3 times at 3 dimensions
High voltage insulation	Terminal to ground 2200VDC, I/O port to other terminal 1500VAC, dwell time 1 minute
Ground connection	Type 3 (single point grounding with multi-branch)
Operating environment	Dust proof, Noncorrosive environment
Toppling	100 mm, 4 drops, unpacked
Free Fall	1 m, 5 times, packed for shipment

4.2 CPU

4.2.1 CPU Specifications

This section introduces the whole general specifications for CTH200 PLC, including Inputs, Outputs and Power Supply.

Table 4-2-1 Performance specifications of CTH200 PLC

Items	Description		
	H224/H226L	H224X/H226XL	H228XL
Dimension(W×H×D)	137×96×71.3mm	196.5×96×71.3mm	200×100×69.5mm

Boolean execution speed		0.22µs	0.15µs
Float execution speed		12µs	8µs
User Program		8+4KB	H224X: 12+4KB H226XL: 24+48KB H228XL: 48+24+24KB
Data space		8KB	H224X: 8+100KB H226XL/H228XL: 10+100KB
Data hold with power-off		Max.8KB, permanent	Max. 10KB, 200 hours
		FlashROM	Super-capacitor and lithium battery
Digital I/O Image		128DI/128DQ (not including image for CAN communication)	640DI/640DQ (including image for CAN communication)
Analog I/O Image		32AI/32AQ (not including image for CAN communication)	194AI/194AQ (including image for CAN communication)
Bit Memory (M)		256 bits	
Local Memory (L)		64 bytes	
SCR (S)		256 bits	
Pulse inputs		14	36
HSC	Single-phase	4 x 50KHz (HSC0~HSC3), only support HSC mode 0 and 9	6 x 50KHz (HSC0~HSC5)
	Two-phase	--	4 x 30KHz (not including HSC3 and HSC5)
HSC input voltage		18~26V	
HSP output (Transistor output)		Motion control outputs: 3 at 50KHz, Q0.0~Q0.2	Motion control outputs: 2 at 50KHz PTO/PWM: 2 at 50KHz Q0.0~Q0.1
Timed interrupts		2 with 1 ms resolution	
Edge interrupts		4 up and/or 4 down	
Max. No. of expansion modules		7	
BD expansion board		1	
CAN expansion board		No	Support
LED indicator		SF (RED): ON - system error, OFF - normal	
		RUN (Green): ON - Run, OFF - Stop	
		STOP (Orange): ON - Stop, OFF - Run	
		RMC (Green): ON - Enable RMC, OFF - Disable RMC	
		LINK/ACT (Green): ON - connected, flash - transfer, OFF-disconnected	
Run switch		RUN/STOP, reset IP settings with 6 times dial in 2s (Note: RUN→STOP counts 1 and so as with STOP→RUN)	

Analog potentionmeter	None	2 with 8 bits resolution	None
Real-time clock	Built-in with resolution of ±120s/m.		
Program Card	Supported. Users can choose to “overwrite/not overwrite” the program/data blocks.		
Battery Card	No	Supported	
Memory Card	No	Supported	
Timer	256 in all 1ms: 4 10ms: 16 100ms: 236	512 in all 1ms: 4 10ms: 272 100ms: 236	
Counter	256 (data-hold while power-off)		

Table 4-2-2 Power specifications for CTH200 PLC

Item	Description
Input Voltage	85~264VAC (47-63HZ), 20.4~28.8VDC, with anti-reverse protection
Inrush current	8A @ 264VAC, 6A @ 28.8VDC
Isolation (field to logic)	AC: 1500VAC DC: Not isolated
Hold up time (power off)	120/240VAC: 10ms/20ms 24VDC: 10ms
Expansion bus +5V	660mA
BD expansion board +5V	200mA
Communication port +5V	10mA, 100R resistor
Sensor supply 24VDC	<=300mA, ripple noise (<10MHZ) <1V PP

Table 4-2-3 Communication specifications for CTH200 PLC

CPU	**H224/H226L**	**H224X/H226XL**	**H228XL**
RS485 communication			
COM ports numbers	H224/H224X: 2, 1 PPI and 1 freeport H226L/H226XL: 2, PPI/Freeport H228XL: 2, PPI/Freeport		
PPI Baud rate	9.6Kbps, 19.2Kbps, 187.5Kbps		
Freeport Baud rate	1.2Kbps~115.2Kbps		
Cable length (max.)	With isolated repeater: 1000m at 187.5Kbps, 1200m at 38.4Kbps Without isolated repeater: 50m		
Stations (max.)	32 for each segment, 126 for each network		
Masters (max.)	32		
Isolation	No		
Ethernet communication			
Interface type	1 standard Ethernet port		
Communication standard	IEEE802.3		

Transmission speed	10Mbps/100Mbps self-adaption	
Self-adaption cross-connect	Yes	No
Industrial Ethernet interface (10/100Mbps)	RJ45	
Protocol type	UDP_PPI, MODBUS_TCP, Ethernet	
Configuration method	PC Searching PLC and downloading configuration via Ethernet port	
Connections (Max.)	Up to 8 UDP and 6 TCP connections for each PLC	
IT connection (Max.)	8 UDP_PPI connections 5 MODBUS_TCP connections, irrespective of master/slave	8 UDP_PPI connections, 4 masters/4 slaves. 4 MODBUS_TCP connections, 2 masters/2 slaves
User data volume	Up to 200 bytes used for TCP/IP Up to 240 bytes used for MODBUS_TCP	
COM	Local: 1~65535 Remote MiCO server: mico.co-trust.com Port: 20000	
Time for start or restart after reset	Approx. 5s	
Indicators	SF (RED): ON - system error, OFF - normal	
	RUN (Green): ON - Run, OFF - Stop	
	STOP (Orange): ON - Stop, OFF - Run	
	RMC (Green): ON - Enable RMC, OFF - Disable RMC	
	LINK/ACT (Green): ON - connected, flash - transfer, OFF - disconnected	
Cable length (Max.)	100m	
Isolation	COM port is isolated	
Cable	Ethernet: CAT5e shield cable	

Table 4-2-4 Digital input specifications of CTH200 PLC

Item		**H224/H224X**	**H226L/H226XL**	**H228XL**
Input	Counts	14	24	36
	Type	Drain/Source		
Rated Voltage		24VDC		
Voltage Range		20.4~28.8VDC		
Surge		35VDC for 0.5s		
Logic 1 (min.)		15 VDC, 2.5mA		
Logic 0 (max.)		5 VDC, 1mA		
Permissible leakage current (max.)		1mA		

Input filter		Configurable, H224/H226L support 3.4ms and 6.4ms; H224X/H226XL/H228XL support 0.2ms, 0.4ms, 0.8ms, 1.6ms, 3.2ms, 6.4ms, 12.8ms, default as 6.4ms
Isolation (field to logic) Isolation groups		500VAC for 1 minute Refer to wiring diagrams
Inputs on simultaneously		All
Max Cable length	Unshielded	300 m for normal inputs
	Shielded	500 m for normal inputs, 50 m for HSC inputs

Table 4-2-5 Digital output specifications of CTH200 CPUs

<table>
<tr><th colspan="2" rowspan="2">Items</th><th>Transistor</th><th colspan="2">Relay</th></tr>
<tr><th>H224/H226L
H224X/H226XL</th><th>H224/H226L
H224X/H226XL</th><th>H228XL</th></tr>
<tr><td colspan="2">Type</td><td>Solid State-MOSFET (Sourcing)</td><td colspan="2">Dry contact</td></tr>
<tr><td colspan="2">Rated voltage</td><td>24VDC</td><td colspan="2">24VDC or 110V/220VAC</td></tr>
<tr><td colspan="2">Voltage Range</td><td>20.4~28.8VDC</td><td colspan="2">5~30VDC: 5~250VAC</td></tr>
<tr><td colspan="2">Max Surge current</td><td>8A for 100ms</td><td colspan="2">5A for 4s@10% duty cycle</td></tr>
<tr><td colspan="2">Max Rated current per point</td><td>0.5A</td><td colspan="2">2.0A</td></tr>
<tr><td colspan="2">Max Rated current per common</td><td>6A</td><td colspan="2">8A</td></tr>
<tr><td colspan="2">Max Lamp load</td><td>5W</td><td colspan="2">DC 30W, AC 200W</td></tr>
<tr><td colspan="2">On State resistance (contact)</td><td>0.3 Ω typical (0.6 Ω max.)</td><td colspan="2">0.2 Ω (maximum when new)</td></tr>
<tr><td colspan="2">Isolation Optical (galvanic, field to logic)</td><td>500VAC for 1 minute</td><td colspan="2">1500VAC for 1 minute</td></tr>
<tr><td rowspan="2">Delay (max.)</td><td>Off to on</td><td colspan="3">H224/H226L(transistor): max 15μs
H224/H226L(relay): max 50μs
H224X/H226XL(transistor): 2μs(Q0.0, Q0.1), max 15μs (all other)
H224X/H226XL/H228XL(relay): max 10ms</td></tr>
<tr><td>On to off</td><td colspan="3">H224/H226L(transistor): max 130μs
H224/H226L(relay): max 200μs
H224X/H226XL(transistor): 10μs(Q0.0, Q0.1), max 130μs (others)
H224X/H226XL/H228XL(relay): max 10ms</td></tr>
<tr><td colspan="2">Lifetime mechanical cycles</td><td>--</td><td colspan="2">10,000,000 (no-load)</td></tr>
<tr><td colspan="2">Lifetime contacts</td><td colspan="3">100,000 (rated load 2A)</td></tr>
<tr><td colspan="2">Outputs on simultaneously</td><td colspan="3">All</td></tr>
<tr><td colspan="2">Connecting two outputs in parallel</td><td>No</td><td colspan="2">Yes, only outputs in same group</td></tr>
<tr><td rowspan="2">Max Cable length</td><td>Shielded</td><td colspan="3">500m</td></tr>
<tr><td>Unshielded</td><td colspan="3">150m</td></tr>
</table>

Specifications of program and data memory

Table4-2-6 Program Specifications of CTH200 PLC

Items	Description	
Instructions	Basic instruction set for CTH200	
Software	MagicWorks PLC/Step7 MicroWIN (Program for using Ethernet port is only supported with MagicWork PLC)	
Interface	RS485/Ethernet port	
Program online	H224X/H226XL/H228XL	
Program language	STL/LAD	
POU	Types/No.	Main program: 1(OB1)
		Subprogram: 128(0-127)
		Interrupt routine: 126(2-127), OB0 reserved, OB1 as main program
	Nesting depth	Main Program: 8 levels
		Interrupt routine: 1 level
Accumulator	4	
Built-in library	MODBUS RTU Communication library MODBUS TCP Communication library PID_T Communication library (H224/H226L supports 16 channels; H224X/H226XL/H228XL supports 64 channels)	

Table 4-2-7 Data memory specifications

Items	H224/H226L	H224X/H226XL /H228XL	H224/H226L	H224X/H226XL/ H228XL
	Digital input image area (I)		Digital output image area (Q)	
Bit address range	I0.0~I15.7	I0.0~I79.7	Q0.0~Q15.7	Q0.0~Q79.7
Byte address range	IB0~IB15	IB0~IB79	QB0~QB15	QB0~QB79
Word address range	IW0~IW14	IW0~IW78	QW0~QW14	QW0~QW78
DWord address range	ID0~ID12	ID0~ID76	QD0~QD12	QD0~QD76
Access	Immediate /direct/indirect access			
Data hold	Power-down data retention is not supported			
	Analog input image area (AI)		Analog output image area (AQ)	
Bits for each channel	16 bits			
Word address range	AIW0~AIW62	AIW0~AIW386	AQW0~AQW62	AQW0~AQW386
Access	Immediate /direct/indirect access			
Data hold	Power-down data retention is not supported			

	Variable memory(V)		
	H224/H226L	H224X	H226XL/H228XL
Storage (bytes)	8K	8K (extend up to 108K)	10K (extend up to 110K)

Bit address range	V0.0~V8191.7	V0.0~V8191.7	V0.0~V10239.7
Byte address range	VB0~VB8191	VB0~VB8191	VB0~VB10239
Word address range	VW0~VW8190	VW0~VW8190	VW0~VW10238
DWord address range	VD0~VD8188	VD0~VD8188	VD0~VD10236
Access	Immediate /direct/indirect access		
Data hold	Power-down data retention is not supported		
	Special memory (SM)		
	H224/H226L	**H224X/H226XL/H228XL**	
Storage	550bytes	650bytes	
Bit address range	SM0.0~SM549.7	SM0.0~SM649.7	
Byte address range	SMB0~SMB549	SMB0~SMB649	
Word address range	SMW0~SMW548	SMW0~SMW648	
DWord address range	SMD0~SMD546	SMD0~SMD646	
Access	The first 30 bytes can be accessed with read-only, support direct/indirect access		
Note: For detailed information about SM, refer to appendix *L Instruction Set*.			
Internal memory (M)			
Storage (bytes)	32 bytes		
Bit address range	M0.0~M31.7		
Byte address range	MB0~MB31		
Word address range	MW0~MW30		
DWord address range	MD0~MD28		
Access	Direct /indirect access		
Data hold	Configurable as all or partial retention at power-down		
Local variable (L)			
Storage (bytes)	64 bytes		
Bit address range	L0.0~L63.7		
Byte address range	LB0~LB63		
Word address range	LW0~LW62		
DWord address range	LD0~LD60		
Access	Direct access		
Data hold	Retention for subprogram (only for H224X and H226XL), no data retention.		
Accumulator register (AC)			
Counts	4		
Bit address range	Not support		
Byte address range	AC0~AC3		
Word address range	AC0~AC3		

DWord address range	AC0~AC3			
Access	Direct access			
Data hold	Not support			
Sequence Control Relay (S)				
Storage (bytes)	32			
Bit address range	S0.0~S31.7			
Byte address range	SB0~SB31			
Word address range	SW0~SW30			
DWord address range	SD0~SD28			
Access	Direct /indirect access			
Data hold	Not support			
Timer (T) - H224/H226L				
Type	**Resolution**	**Quantity**	**Number**	**Max. time**
TONR	1ms	2	T0, T64	32.767s
	10ms	8	T1~T4, T65~T68	327.67s
	100ms	54	T5~T31, T69~T95	3276.7s
TON/TOF	1ms	2	T32, T96	32.767s
	10ms	8	T33~T36, T97~T100	327.67s
	100ms	182	T37~T63, T101~T255	3276.7s
Timer (T) - H224X/H226XL/H228XL				
Type	**Resolution**	**Quantity**	**Number**	**Max. time**
TONR	1ms	2	T0, T64	32.767s
	10ms	8	T1~T4, T65~T68	327.67s
	100ms	54	T5~T31, T69~T95	3276.7s
TON/TOF	1ms	2	T32, T96	32.767s
	10ms	8+256	T33~T36, T97~T100, T256~T511	327.67s
	100ms	182	T37~T63, T101~T255	3276.7s
Access	Counter register support both direct and indirect access, only direct access for status bit			
Retention	Configurable for the current count value, not for status bit			
Counter (C)				
No.	256			
Counting mode	Upward/downward/both upward and downward			
Max count value	32767			
Access	Support direct/indirect accessing for counter register, only direct accessing for status bit			
Retention	Configurable for the current count value, not for status bit			

Table 4-2-8 Data types of CTH200 PLC

Type	Size	Description	Value range
BOOL	1bit	BOOL value	0~1
Byte	8bits	Unsigned byte	0~255

Word	16bits	Unsigned integer	0~65535
Integer	16bits	Signed integer	-32768~+32767
Dword	32bits	Unsigned double integer	0~4294967295
Double Integer	32bits	Signed double integer	-2147483648~+2147483647
Real	32bits	IEEE 32 bits float	+1.175495E-38~+3.402823E+38 -1.175495E-38~-3.402823E+38
String	1~255 bytes	ASCII string: 1 bytes characters + ASCII characters	None

Password access control

Table 4-2-9 Password access control of CTH200

Items	Level 1	Level 2	Level 3	Level 4
Read-write user data	Y	Y	Y	Y
RUN/Stop/power on reset	Y	Y	Y	Y
Read-write real time clock	Y	Y	Y	Y
Write Q at STOP	Y	Validate password	Validate password	Validate password
Mandatory data	Y	Validate password	Validate password	Validate password
Upload program block/DB/hardware configuration	Y	Y	Validate password	N
Download program block/DB/hardware configuration	Y	Validate password	Validate password	Validate password (hardware configuration is not permitted to download)
Clear program block/DB/hardware configuration	Y	Validate password	Validate password	Validate password (hardware configuration is not permitted to delete. Allow to delete all three)
Edit at Runtime	Y	Validate password	Validate password	N
First or multiple scan	Y	Validate password	Validate password	Validate password
Refresh scan	Y	Validate password	Validate password	Validate password
Project Comparison	Y	Y	Validate password	N
Program condition monitoring (timestamp compare is allowed)	Y	Y	Y	Y
Program condition monitoring (timestamp compare is not allowed)	Y	Validate password	Validate password	N

The real time clock and interrupt

Table 4-2-10 Real time clock of CTH200

Factory setting	Not set, fixed at 00:00:00 of 1/1/1990, Sunday

Retention at power down	About 100h (typical at 25℃)	
Resolution	Bias <120s each month	
Read clock	Read via TODR/TODRX instruction or software	
Set clock	Set via TODW/TODWX instruction or software	
General clock format (8 bytes)		
T BYTE	**Description**	**Byte data**
0	Year(0-99)	Current year (BCD value)
1	Month(1-12)	Current Month (BCD value)
2	Date(1-31)	Current date (BCD value)
3	Hour(0-23)	Current hour (BCD value)
4	Minute(0-59)	Current minute (BCD value)
5	Second(0-59)	Current second (BCD value)
6	0	Reserved, always set as 00
7	Day of the week(1-7)	The current day of the week, 1=Sunday (BCD value)
Extended clock format (19 bytes)		
0	Year(0-99)	Current year (BCD value)
1	Month(1-12)	Current Month (BCD value)
2	Date(1-31)	Current date (BCD value)
3	Hour(0-23)	Current hour (BCD value)
4	Minute(0-59)	Current minute (BCD value)
5	Second(0-59)	Current second (BCD value)
6	0	Reserved, always set as 00
7	Day of the week (1-7)	The current day of the week, 1=Sunday (BCD value)
8	Time zone	00H-03H, 08H,10H-13H, FFH
9	Modified hours (0-23)	Modified values, hour (BCD value)
10	Modified minutes (0-59)	Modified values, minute (BCD value)
11	Starting month (1-12)	Starting month in DST (BCD value)
12	Starting date (1-31)	Starting date in DST (BCD value)
13	Starting hour (0-23)	Starting hour in DST(BCD value)
14	Starting minute (0-59)	Starting minute in DST (BCD value)
15	Ending month (1-12)	Ending month in DST (BCD value)
16	Ending date (1-31)	Ending date in DST (BCD value)
17	Ending hour (0-23)	Ending hour in DST (BCD value)
18	Ending minute (0-59)	Ending minute in DST (BCD value)

The following table lists interrupts supported by CTH200.

Table 4-2-11 Interrupts supported by CTH200

Groups and priority	Number	Group	Description
Communication and diagnostic events (Top Priority)	8	0	Port 0: receive character
	9	0	Port 0: transfer complete
	23	0	Port 0: complete receiving message
	24	0	Port 1: complete receiving message

	25	0	Port 1: receive character
	26	0	Port 1: transfer complete
	36	0	Module diagnostics interrupt
Disperse (Medium Priority)	0	1	Rising edge, I0.0
	2	1	Rising edge, I0.1
	4	1	Rising edge, I0.2
	6	1	Rising edge, I0.3
	1	1	Falling edge, 10.0
	3	1	Falling edge, I0.1
	5	1	Falling edge, I0.2
	7	1	Falling edge, I0.3
	12	1	HSC0 CV=PV
	27	1	HSC0 direction change
	28	1	HSC0 External recovery/Zphase
	13	1	HSC1 CV=PV
	14	1	HSC1 direction change
	15	1	HSC1 External recovery
	16	1	HSC2 CV=PV
	17	1	HSC2 direction change
	18	1	HSC2 External recovery
	19	1	PTO 0 complete interrupt
	20	1	PTO 1 complete interrupt
	32	1	HSC3 CV=PV
	29	1	HSC4 CV=P
	30	1	HSC4 direction change
	31	1	HSC4 External recovery/Z phase
	33	1	HSC5 CV=PV
Timer (Lowest priority)	10	2	Timer interrupt 0
	11	2	Tmer interrupt 1
	21	2	Timer T32 CT=PT interrupt
	22	2	Timer T96 CT=PT interrupt

4.2.2 Diagrams of CPU Structure and Wiring

PLC structure

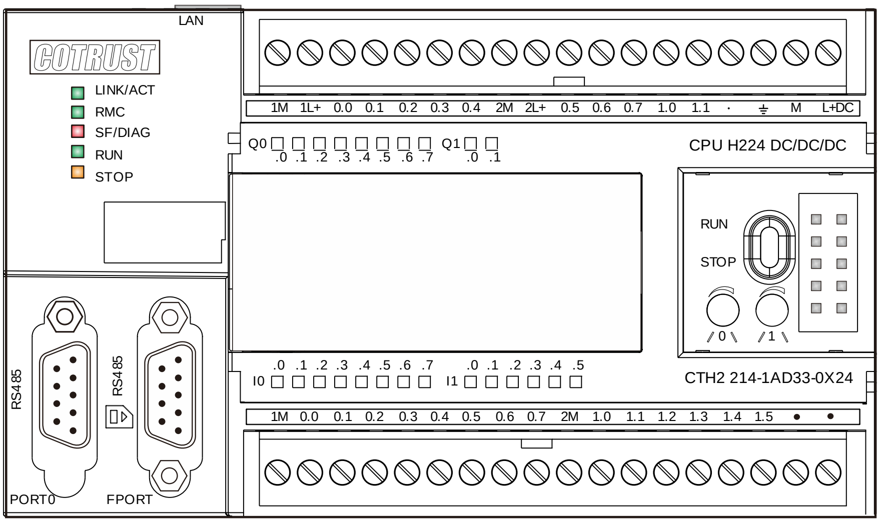


Figure 4-1 CTH2 214-1AD33-0X24

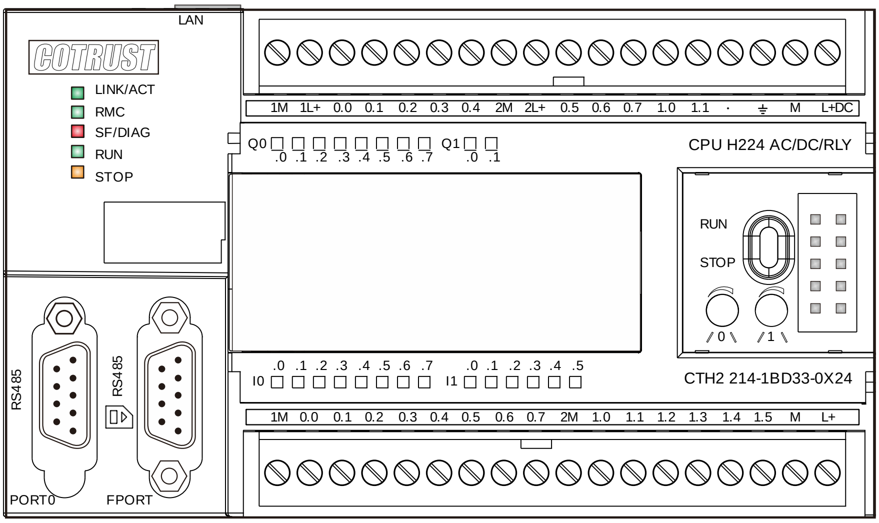


Figure 4-2 CTH2 214-1BD33-0X24

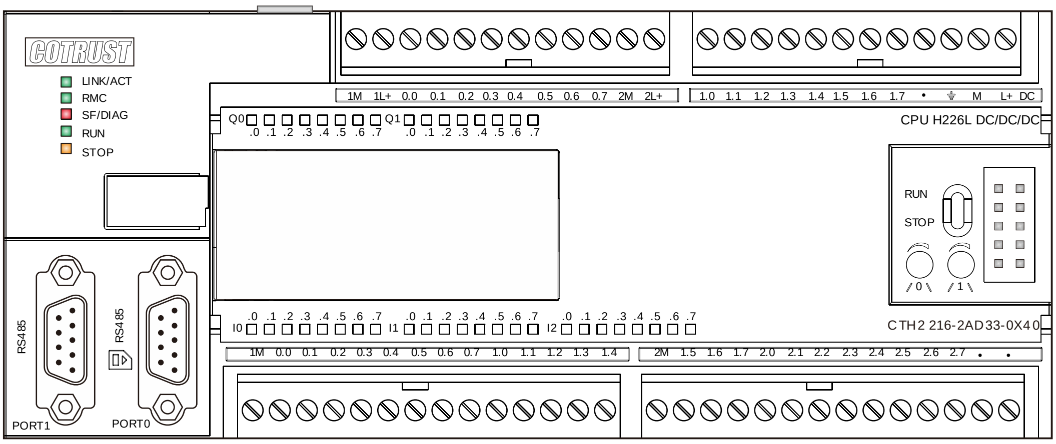


Figure 4-3 CTH2 216-2AD33-0X40

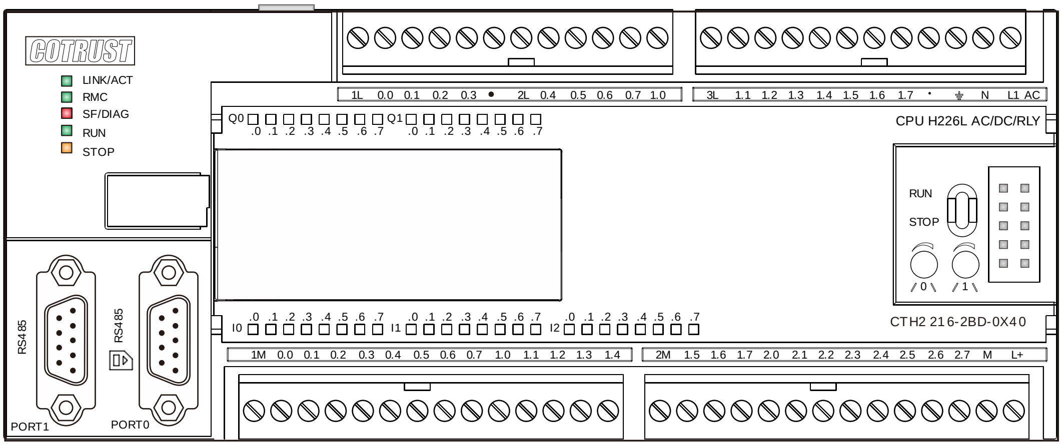


Figure 4-4 CTH2 216-2BD33-0X40

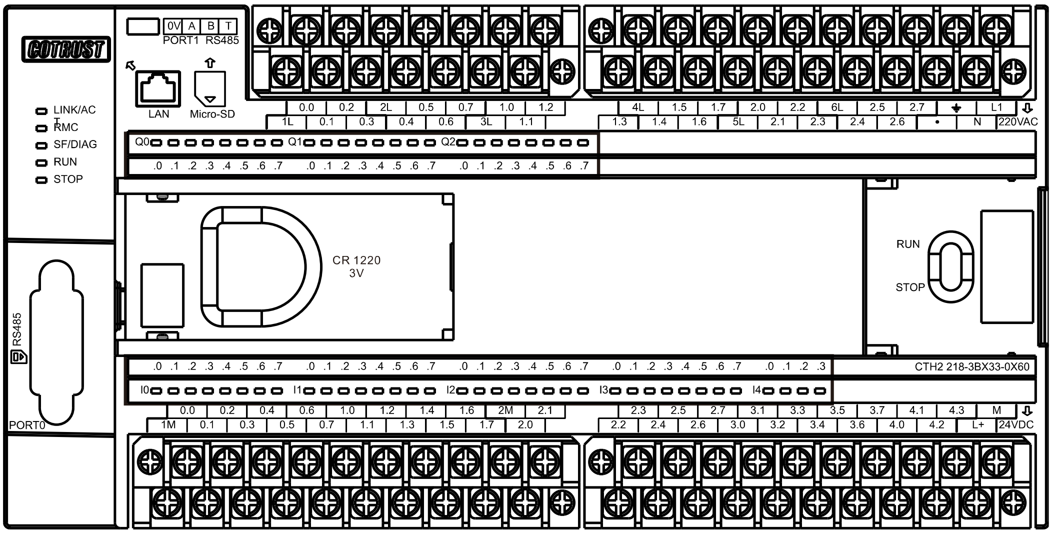


Figure 4-5 CTH2-218-3BX33-0X60

Wiring Diagrams

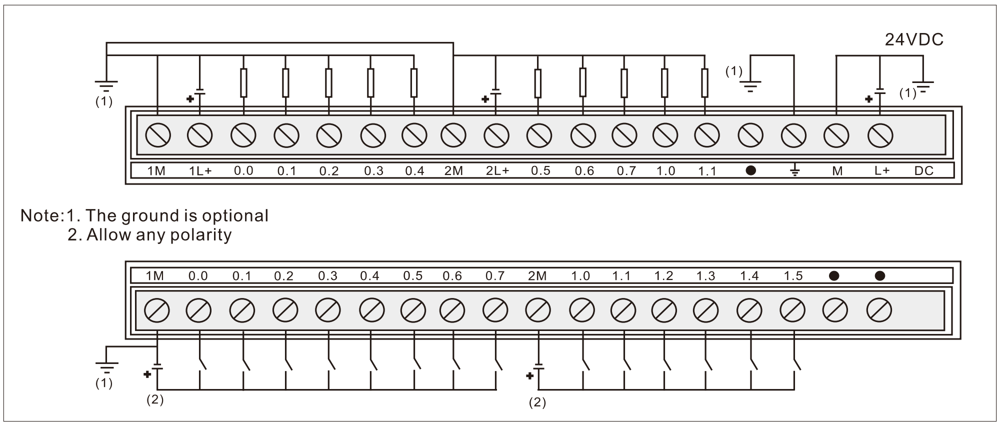


Figure 4-6 CTH2 214-1AD33-0X24 transistor output

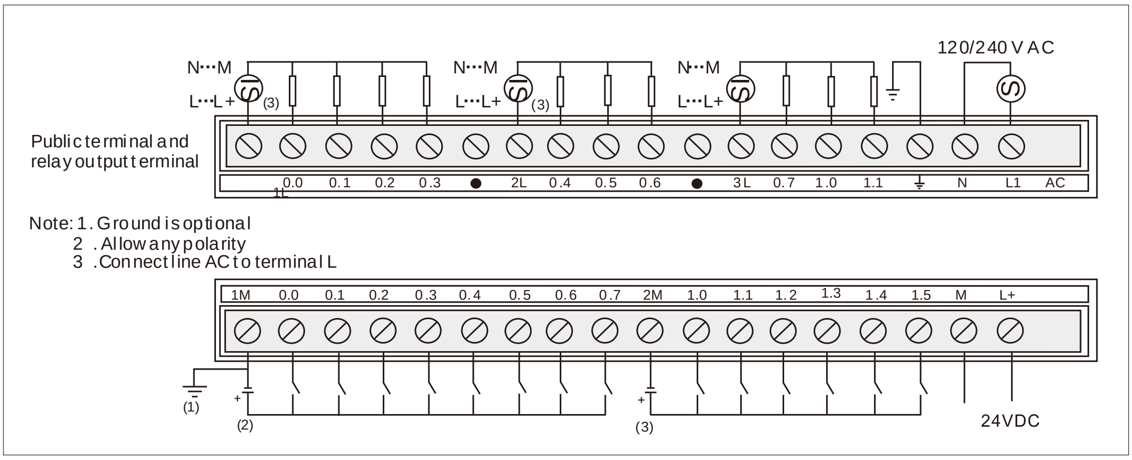


Figure 4-7 CTH2 214-1BD33-0X24 relay output

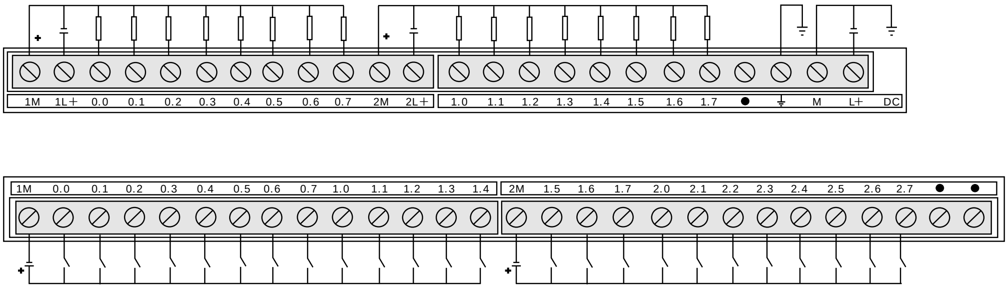


Figure 4-8 CTH2 216-2AD33-0X40 transistor output

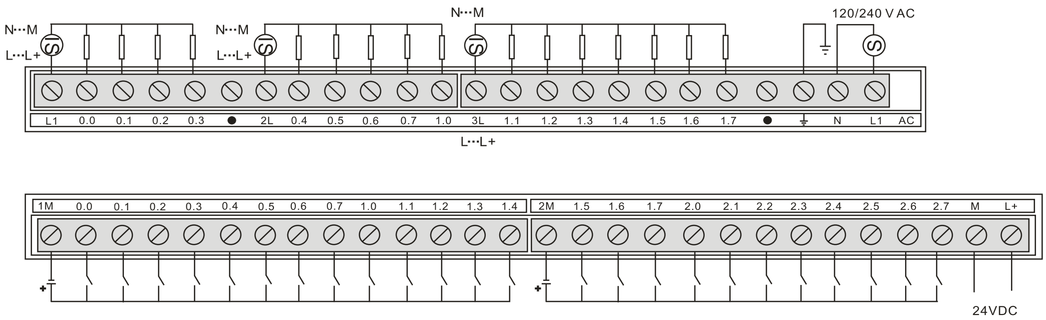


Figure 4-9 CTH2 216-2BD33-0X40 relay output

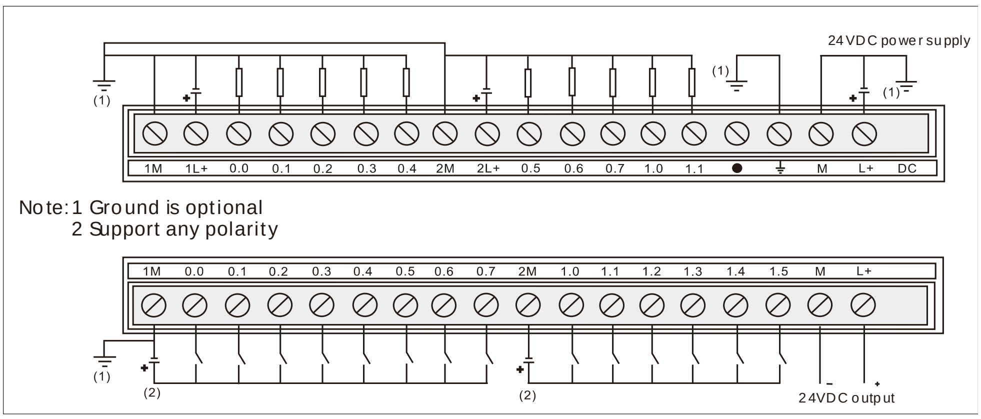


Figure 4-10 CTH2 214-1AX33-0X24 transistor output

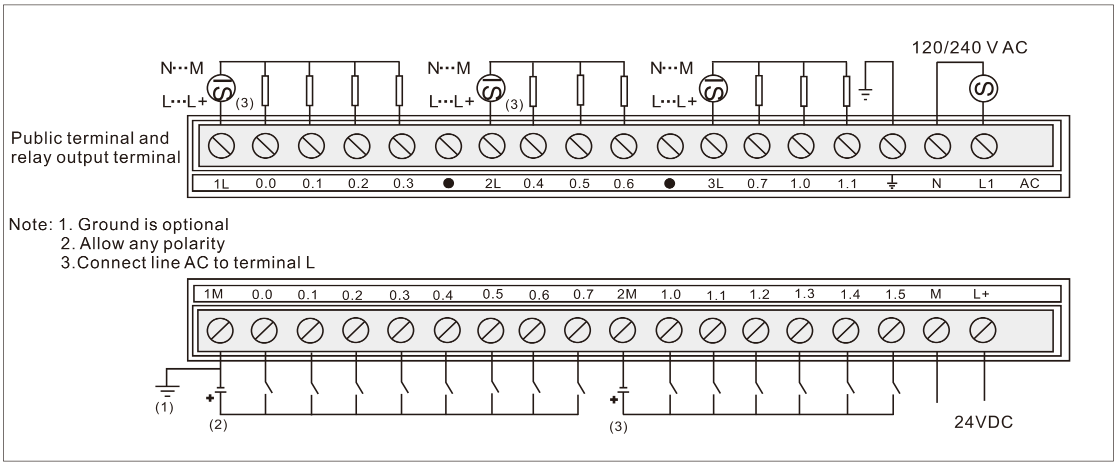


Figure 4-11 CTH2 214-1BX33-0X24 relay output

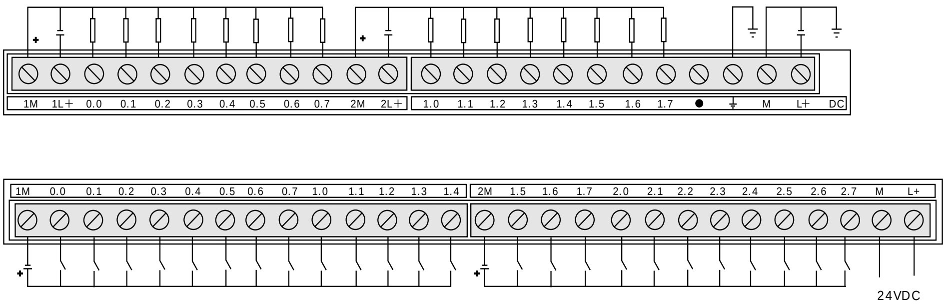


Figure 4-12 CTH2 216-2AX33-0X40 transistor output

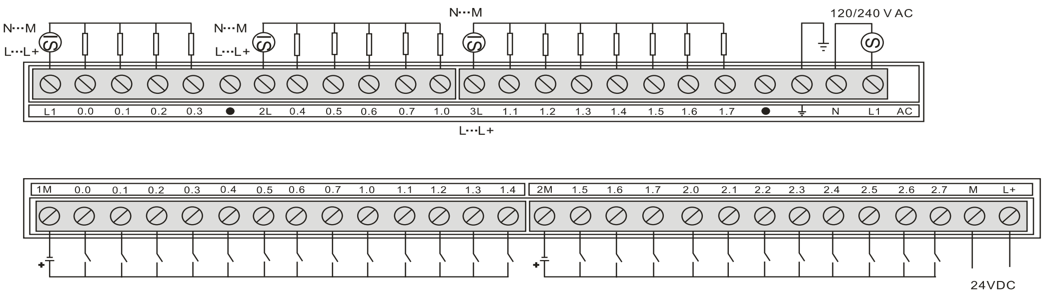


Figure 4-13 CTH2 216-2BX33-0X40 relay output

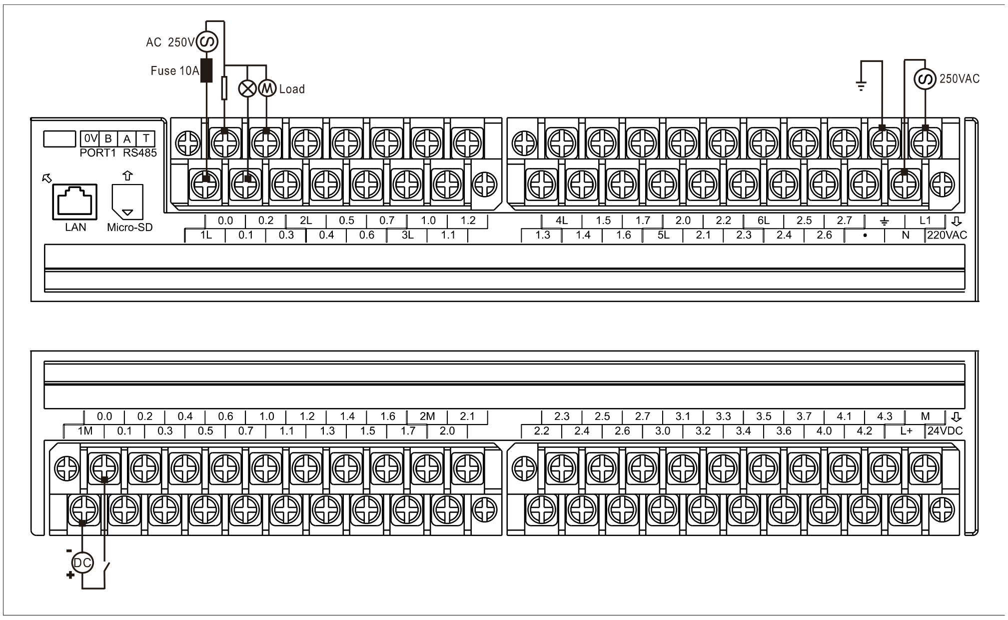


Figure 4-14 CTH2 218-3BX33-0X60 relay output

4.2.3 Definition of Communication Port

CTH200 series PLC, with RS485 interface and standard Ethernet port, supports both serial and Ethernet communication, definition of these two port shows as below:

Table 4-2-12 Definition of standard RS485 communication port

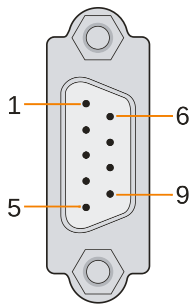
Connector	Pin Number	Port 0	Port 1
	1	Shell ground	Shell ground
	2	+24V Return	+24V Return
	3	RS485 signal B	RS485 signal B
	4	RTS (TTL)	RTS (TTL)
	5	+5V Return	+5V Return
	6	+5V, 100Ω	+5V, 100Ω
	7	+24V	+24V
	8	RS485 signal A	RS485 signal A
	9	NC	NC
	Connector shell	Shell ground	Shell ground

Table 4-2-13 4-pin RS485 Communication port (only for CPU H228XL)

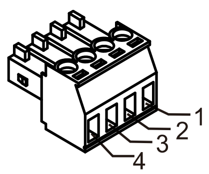
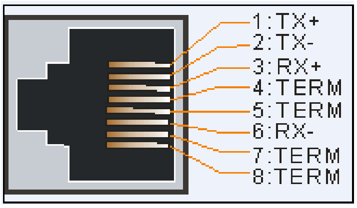
Connector	Pin Number	Signal	Definition
	1	0V	Logic common
	2	A	RS485 signal A
	3	B	RS485 signal B
	4	T	Terminal resistor, connect with pin2

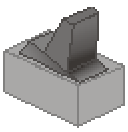
Table 4-2-14 RJ45 Ethernet port

RJ45 Connector	Pin Number	Signal	Definition
	1	TX+	Data send positive
	2	TX-	Data send negative
	3	RX+	Data receive positive
	4	TERM	--
	5	TERM	--
	6	RX-	Data receive negative
	7	TERM	--
	8	TERM	--

> **Note**
> Define RS485 signal of some meters as A-positive, B-negative. Positive to positive and negative to negative when wiring (universal).

4.2.4 DIP switch

Table 4-2-15 Definition of DIP switch

Two-state switch	State	Operation	Signal definition
	ON	UP	RUN
	OFF	Down	STOP
		Dial 6 times in 2 seconds	Reset IP (RUN->STOP counts, vice versa)

4.2.5 Making Standard Net Cable

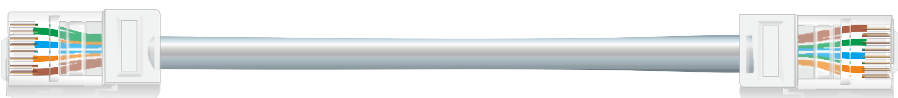
It is recommended to use standard network cable for CTH200 communication. There's two forms of standard network cable: parallel line and cross line. The ends of parallel lines are standard 568A or 568B; the cross line is standard 568A at one end while 568B at the other end. The standard line sequence color of 568A/568B shows as below:

568B standard line sequence: white and orange, orange, white and green, blue, white and blue, green, white and brown, brown.

568A standard line sequence: white and green, green, white and orange, blue, white and blue, orange, white and brown, brown.

How to make standard cable:

Parallel line(both sides are 568A standard or 568B standard)



Cross line(One end is 568A standard and the other end is 568B standard)

4.3 Digital Expansion Modules Specifications

CTH200 series PLC are equipped with variety of digital expansion modules, including input, output and I/O modules.

Table 4-3-1 CTH200 series PLC digital expansion modules order info

Expansion modules	Order No.
SM221 digital input module, 8 inputs, 24VDC	CTH2 221-1BF32
SM221 digital input module, 16 inputs, 24VDC	CTH2 221-1BH32
SM221 digital input module, 32 inputs, 24VDC	CTH2 221-1BL32
SM222 digital output module, 8 transistor outputs, 24VDC, 0.5A(with output protection)	CTH2 222-1BF32
SM222 digital output module, 16 transistor outputs, 24VDC, 0.5A(with output protection)	CTH2 222-1BH32
SM222 digital output module, 32 transistor outputs, 24VDC, 0.5A(with output protection)	CTH2 222-1BL32
SM222 digital output module, 8 relay outputs, 2A	CTH2 222-1HF32
SM222 digital output module, 16 relay outputs, 2A	CTH2 222-1HH32
SM223 digital I/O module, 4 24VDC inputs, 4 transistor outputs, 24VDC, 0.5A(with output protection)	CTH2 223-1BF32
SM223 digital I/O module, 8 24VDC inputs, 8 transistor outputs, 24VDC, 0.5A(with output protection)	CTH2 223-1BH32
SM223 digital I/O module, 16 24VDC inputs, 16 transistor outputs, 24VDC, 0.5A(with output protection)	CTH2 223-1BL32
SM223 digital I/O module, 4 24VDC inputs, 4 transistor outputs, 2A	CTH2 223-1HF32
SM223 digital I/O module, 8 24VDC inputs, 8 transistor outputs, 2A	CTH2 223-1PH32
SM223 digital I/O module, 16 24VDC inputs, 16 transistor outputs, 2A	CTH2 223-1PL32

4.3.1 Digital Input Modules Specifications

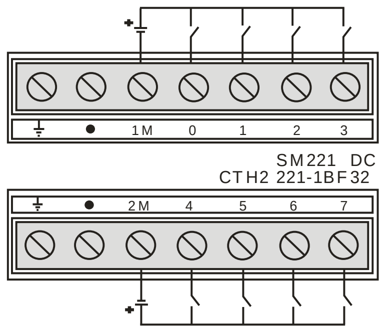
Table 4-3-2 Digital Input Modules Specifications

Features	8 inputs, 24VDC	16 inputs, 24VDC	32 inputs, 24VDC
Order No.	CTH2 221-1BF32	CTH2 221-1BH32	CTH2 221-1BL32
Dimension(W×H×D)	46 × 96 × 62mm	71.3 × 96 × 62mm	138 × 96 × 62mm
Power supply consumption			
Power	2W	3W	
+5VDC current	57mA	79mA	179mA
Input features			
Counts	8	16	32
Type	Drain/Source		
Input voltage			
Rated voltage	24VDC		
Max. Permit Voltage	30VDC		
Surge	35VDC, 0.5s		

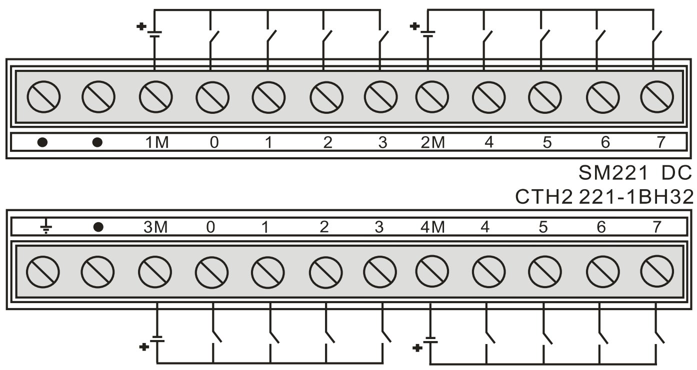
Signal "1"	15~30V		
Signal "0"	0~5V		
Isolation			
Optical (galvanic, field to logic)	500VAC, last within 1 minute		
Isolation points for each Group	4		8
Input delay (max.)	4.5ms		
Leakage current (max.)	1mA AC		
Maximum Cable length			
Unshielded	300m		
Shielded	500m		
Output on simultaneously			
40℃	8	16	32
50℃	8	16	32

Wiring diagram

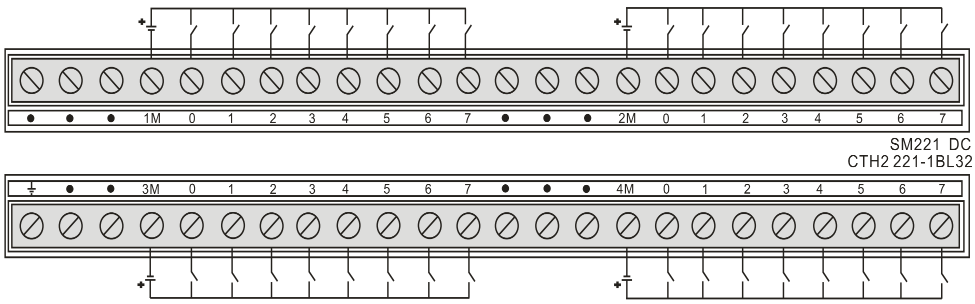
SM221 digital input module (CTH2 221-1BF32)



SM221 digital input module (CTH2 221-1BH32)



SM221 digital input module (CTH2 221-1BL32)



4.3.2 Digital Output Modules Specifications

Transistor Output

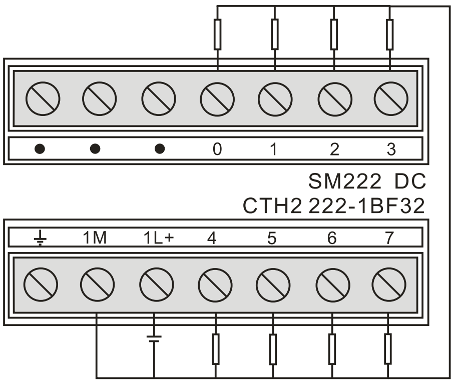
Table 4-3-3 Digital Output Modules Specifications

Features	8 output, 24VDC	16 output, 24VDC	32 output, 24VDC
Order No.	CTH2 222-1BF32	CTH2 222-1BH32	CTH2 222-1BL32
Dimension(W×H×D)	46 × 96 × 62mm	71.3 × 96 × 62mm	138 × 96 × 62mm
Power supply consumption			
Power	2W	3W	
+5VDC current	57mA	79mA	174mA
Output features			
Counts	8	16	32
Type	Solid MOSFET		
Output voltage			
Rated voltage	24VDC		
Max. Permit Voltage	20.4~28.8VDC		
Surge	Minimum 20VDC		
Signal "1"	Maximum 0.1VDC(10KΩ load)		
Output current			
Signal "1"	0.5A		
Output groups	1	2	4
Output counts for each group	8		
Output on simultaneously	8	16	32
Maximum current each group	4A		
Lamp load (max.)	5W		
On state resistance (contact)	0.3Ω		
Leakage current (max.)	10μA		
Surge current (max.)	8A, 100ms		
Isolation			

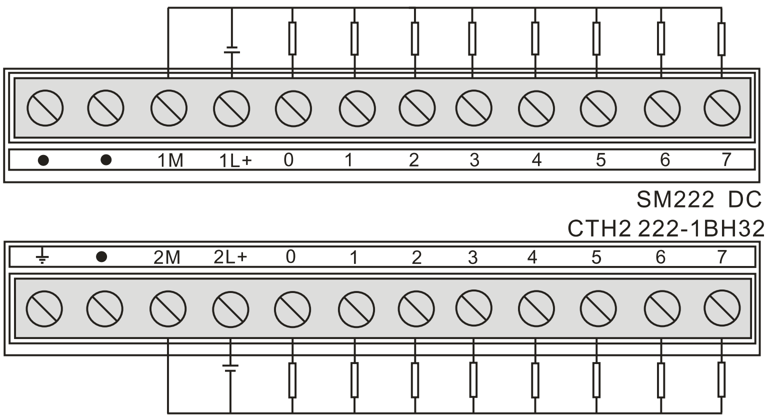
Optical (galvanic, field to logic)	500VAC, last within 1 minute
Isolation groups	8 points
Output delay(RL = 50Ω)	
Off－on	Maximum 50μs
On－off	Maximum 200μs
Maximum Cable length	
Unshielded	150m
Shielded	500m

Wiring diagram

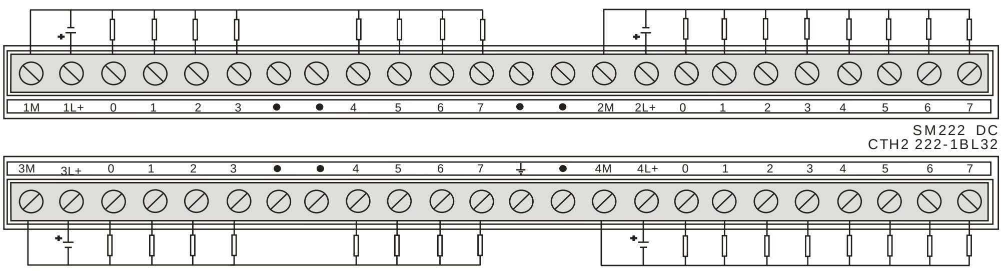
SM222 transistor output modules (CTH2 222-1BF32)



SM222 transistor output modules (CTH2 222-1BH32)



SM222 transistor output modules (CTH2 222-1BL32)



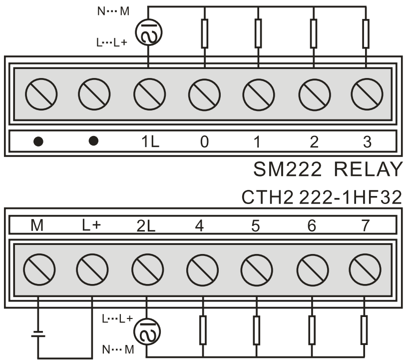
Relay output

Table 4-3-4 Digital output modules specifications

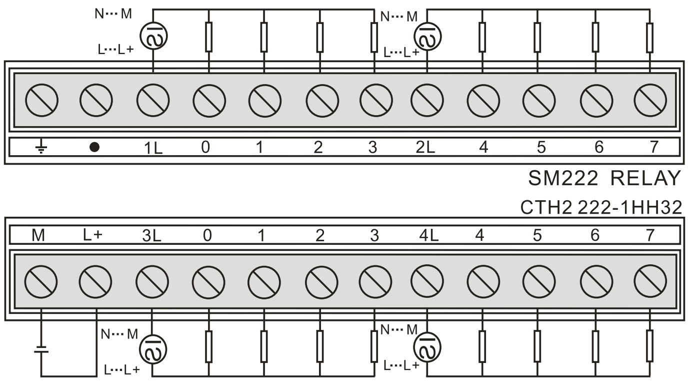
Features	8 outputs, relay	16 outputs, relay
Order No.	CTH2 222-1HF32	CTH2 222-1HH32
Dimension(W×H×D)	46 × 96 × 62mm	71.3 × 96 × 62mm
Power supply consumption		
Power	2W	3W
+5VDC current	68mA	115mA
Output features		
Counts	8	16
Type	Relay—dry contact	
Rated output voltage	DC: 5~30V, AC: 5~250V	
Output current		
Signal "1"	2A	
Output groups	2	4
Output counts for each group	4	
Output on simultaneously	8	16
Maximum current each group	8A	
Lamp load (max.)	5W	
On state resistance (contact)	0.2Ω	
Leakage current (max.)	7A, with contact closing	
Surge current (max.)	Additional provide	
Isolation		
Counts	4	
Coil and logic current	No	
Coil and contact	1500VAC, last 1 minute	
Resistance (Coil and contact)	Minimum 100MΩ	
Relay features		
On/off delay	15ms	
On/off frequency(max)	1Hz	
On/off time(no-load)	30,000,000	
Contact life(rated load)	300,000	
Maximum Cable length		
Unshielded	150m	
Shielded	500m	

Wiring diagram

SM222 relay output module(CTH2 222-1HF32)



SM222 relay output module (CTH2 222-1HH32)



4.3.3 Digital Input/Output Module Specifications

Transistor Output

Table 4-3-5 Digital I/O Module Specifications

Features	4 DI/DO, 24VDC	8 DI/DO, 24VDC	16 DI/DO, 24VDC
Order No.	CTH2 223-1BF32	CTH2 223-1BH32	CTH2 223-1BL32
Dimension			
Size (W×H×D)	46 × 96 × 62mm	71.3 × 96 × 62mm	138 × 96 × 62mm
Power supply			
Power loss	2W	3W	
+5VDC Consumption	57mA	73mA	115mA
Input			
Counts	4	8	16
Type	drain/source		
Input Voltage			
Rated Voltage	24VDC		
Max. rated voltage	30VDC		
Surge	35VDC, 0.5s		
Logic “1”	15~30V		
Logic “0”	0~5V		
Isolation			
Optical (galvanic,	500VAC last within 1 minute		

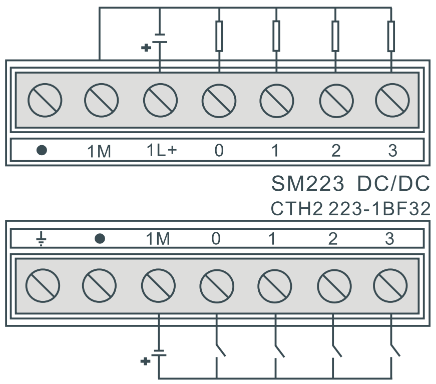
field to logic)			
Isolation Groups	4 points	8 points	
Input delay (max.)	4.5ms		
Leakage current (max.)	1mA AC		
Output on simultaneously			
40℃	4	8	16
50℃	4	8	16
Output			
Counts	4	8	16
Type	Solid MOSFET		
Output Voltage			
Rated load Voltage	24VDC		
Range	20.4~28.8VDC		
Logic "1"	Min. 20VDC		
Logic "0"	Max. 0.1VDC (at 10KΩ load)		
Output Current			
Rated current per point (max.)	0.5A		
Outputs per common	4	8	
Output on simultaneously	4	8	16
Rated current per common (max.)	2A	4A	
Lamp load (max.)	5W		
On state resistance (contact)	0.3Ω		
Leakage current (max.)	10mA		
Surge current (max.)	8A, 100ms		
Isolation	Optical (galvanic, field to logic), 500VAC for 1 minute		
Isolation groups	4		4/4/8
Output Delay (RL = 50Ω)			
Off to On	Max 50ms		
On to Off	Max 200ms		
Cable length (max.)			
Unshielded	150m		
Shielded	500m		

Caution

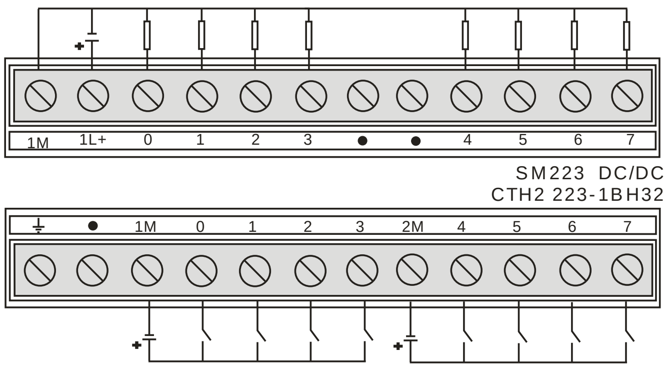
24VDC digital expansion modules support short, overcurrent and overvoltage protection.

Wiring diagram

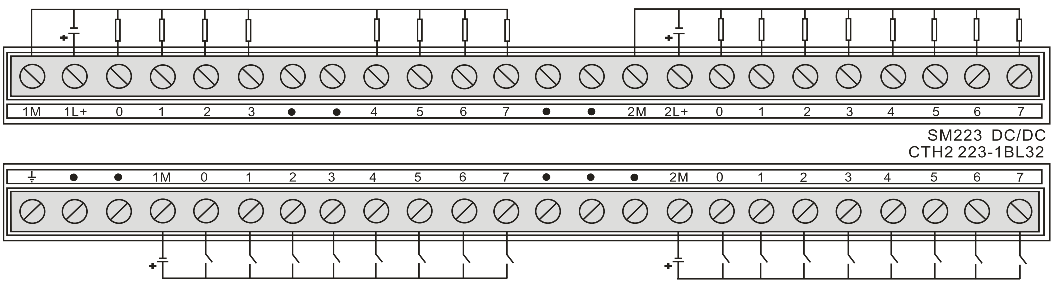
SM223 digital I/O module(CTH2 223-1BF32)



SM223 digital I/O module (CTH2 223-1BH32)



SM223 digital I/O module (CTH2 223-1BL32)



Relay Output

Table 4-3-6 Digital I/O modules Specifications

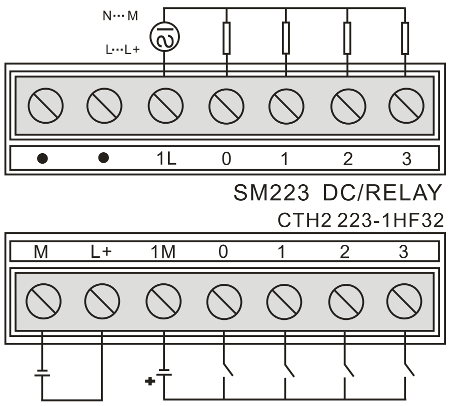
Features	4DI×24VDC 4DO×Relay	8DI×24VDC 8DO×Relay	16DI×24VDC 16DO×Relay
Order No.	CTH2 223-1HF32	CTH2 223-1PH32	CTH2 223-1PL32
Dimension			
Size(W×H×D)	46 × 96 × 62mm	71.3 × 96 × 62mm	138 × 96 × 62mm
Power Supply			
Power Loss	2W	3W	6W

+5VDC current	58mA	89mA	150mA
L+ current	output 9mA per point when switch on		
L+ Voltage	20.4－28.8VDC		
Input			
Counts	4	8	16
Type	Drain/Source		
Input voltage			
Range	24VDC		
Max. Permitted voltage	30VDC		
Surge voltage	35VDC, 0.5s		
Logic "1"	15~30V		
Logic "0"	0~5V		
Isolation			
Optical (galvanic, field to logic)	500VAC for 1 minute		
Isolation Groups	4 point		8 point
Input delay (max.)	4.5ms		
Leakage current (max.)	1mA AC		
Output on simultaneously			
40℃	4	8	16
50℃	4	8	16
Output			
No.	4	8	16
Type	Relay－dry contact		
Output voltage	DC: 5~30V, AC: 5~250V		
Output Current			
Signal "1"	2A		
Rated current per common (max.)	8A		
Light load	DC: 30W, AC: 200W		
Contact resistance	0.2Ω		
Surge current (max)	7A, with contact closing		
Short-circuit protection	Additional provide		
Relay features			
Switching delay (max.)	15ms		
Switching frequency (max.)	1Hz		
Lifetime mechanical cycles	30,000,000		

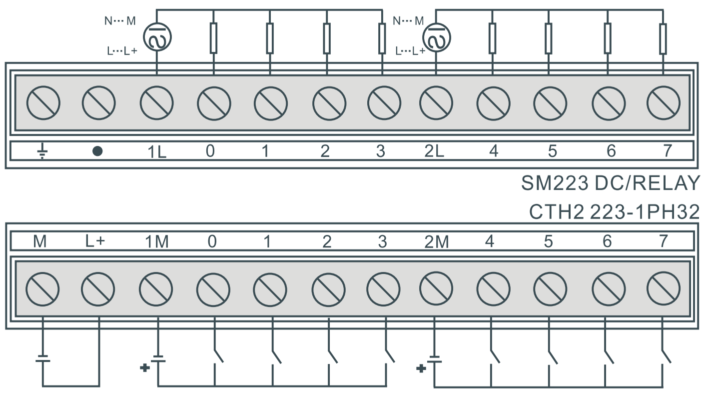
Lifetime contacts	300,000
Cable length (max)	
Unshielded	300m
Shielded	500m

Wiring diagram

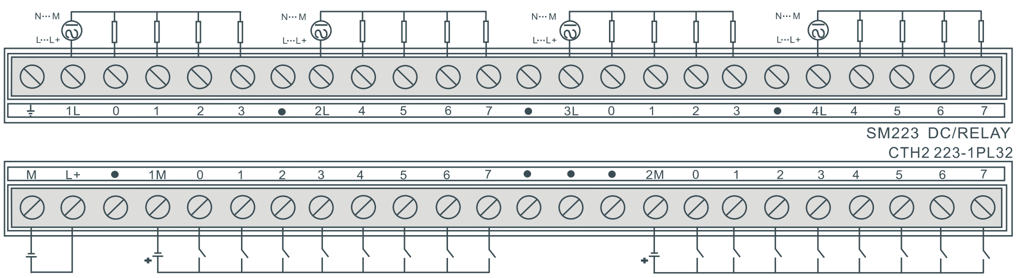
SM223 digital I/O modules (CTH2 223-1HF32)



SM223 digital I/O modules (CTH2 223-1PH32)



SM223 digital I/O modules (CTH2 223-1PL32)



4.4 Analog Expansion Modules Specification

Table 4-4-1 Analog expansion modules order info.

Expansion modules	Order No.
SM231 analog input module, 4 inputs, 0~20mA current or ±5V, ±2.5V, 0~0V, 0~5V voltage input, isolated 12 bits precision	CTH2 231-0HC32
SM231 high precision analog input module, 8 inputs, voltage input, opto-isolator, 16 bits precision	CTH2 231-0HF32
SM231 high precision analog input module, 8 inputs, current input, opto-isolator, 16 bits precision	CTH2 231-1HF32
SM231 analog voltage input module, 8 inputs, ±2.5V, 0~10V, 0~5V voltage input, two optional channels with 0~20mA current input, isolated 12 bits precision	CTH2 231-5HF32
SM232 analog output module, 2 outputs, ±10V voltage or 0~20mA current output, isolated voltage 12 or current 11 bits precision	CTH2 232-0HB32
SM232 analog output module, 4 outputs, ±10V voltage or 0~20mA current output, isolated voltage 12 or current 11 bits precision	CTH2 232-0HD32
SM235 analog I/O module, 4 voltage or current inputs/1 voltage or current output, isolated voltage 12 or current 11 bits precision	CTH2 235-0KD32

4.4.1 Analog Input Module Specification

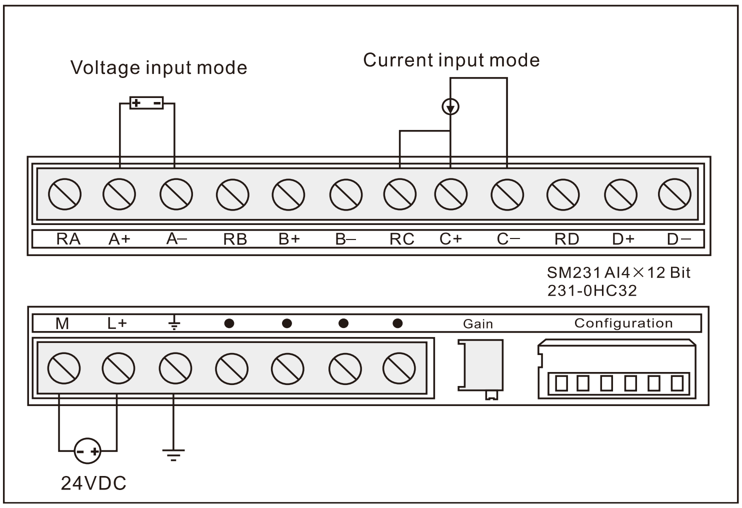
Table 4-4-2 SM231-0HC specifications

Features	
Dimension (W×H×D)	71.3×96×62mm
LED	24V power , ON: Normal, OFF: Faulted
Power Supply	
+5VDC Consumption	87mA
L+	17mA
L+ Coil Voltage	20.4~28.8VDC
Power consumption	2W
Analog Input	
Counts	4
Isolation (field to logic)	500VAC for 1 minute Opto-couplers isolation
Input type	Differential
Input ranges	
Voltage (unipolar)	0~10V, 0~5V
Voltage (bipolar)	±5V, ±2.5V
Current	0~20mA
Data word format	
Unipolar	0~32000
Bipolar	-32000~32000
Resolution, full-scale	
Voltage (unipolar)	2.5mV(0~10V range); 1.25mV(0~5V range)

Voltage (bipolar)	2.5mV(±5V range); 1.25mV(±2.5V range)
Current	5μA(0~20mA range)
Analog to digital conversion time	< 300μs
Analog input step response	1.5ms
Common mode rejection	40dB, DC 60Hz
Common mode voltage	Signal voltage plus common mode voltage must be ≤ 12V
DC Input impedance	≥10MΩ
Maximum input voltage	30V
Maximum input current	30mA

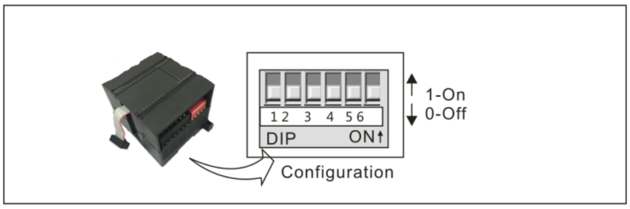
Wiring Diagrams

SM231 AI Module (CTH2 231-0HC32)



Calibration and range

【Calibration and the position of range switch】



【Input calibration】

Because the calibration adjustment affects the OPAMP behind the analog multiplexer, it affects all user input channels. In addition, because the component parameters of the input channels in front of the multiplexer may differ, there is a slight difference of the same input signal on different channels even after calibration.

The input has been filtered in the module, so the measurements is stable. Start the analog input filter for all input of the module for better performance parameters. When calculating the average value, select the sampling times above 64.

Calibration steps:

1) Cut off external power of modules, configure DIP switch to select needed range.

2) Connect CPU with module power, wait for 15min above.

3) Use a transmitter, a voltage input source or a current input source to add the zero signal to module input.

4) Read measurement of the input in CPU.

5) Adjust OFFSET potentiometer till the reading is zero, finish zero set calibration.

6) Input a full scale signal to read the measurement in CPU.

7) Adjust GAIN potentiometer till the reading is 32000.

8) Repeat 3~7 when necessary.

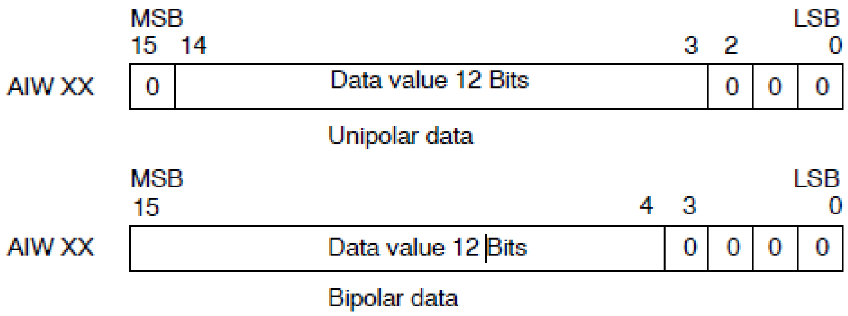
【Select range】

The following table shows how to set the range of SM231 4AI with DIP switch. Use 1, 2 and 3 to select the analogy input range which can be set to the same.
Attention: the unused DIP switch SW4 ~ SW6 must be set to OFF.

Table 4-4-3 DIP switch configuration of SM231-0HC

Unipolar			**Full-range**	**Resolution**
SW1	**SW2**	**SW3**		
ON	OFF	ON	0 - 10V	2.5mV
ON	ON	OFF	0 - 5V	1.25mV
ON	ON	OFF	0 - 20mA	5μA
Bipolar			**Full-range**	**Resolution**
SW1	**SW2**	**SW3**		
OFF	OFF	ON	±5V	2.5mV
OFF	ON	OFF	±2.5V	1.25mV

Input Data Format



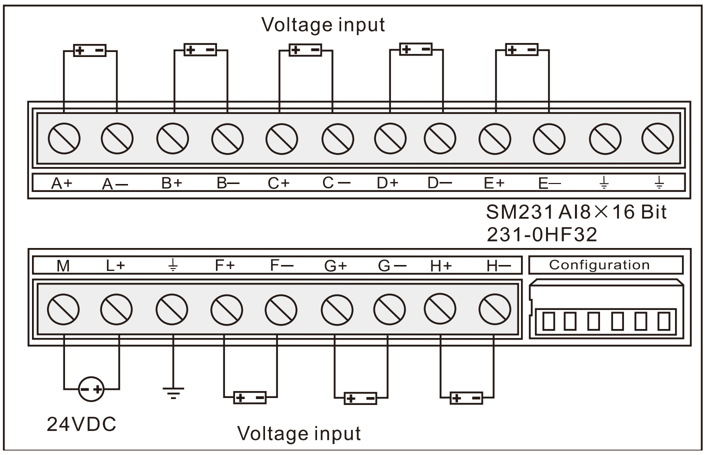
Notice

The 12 bits of the analog-to-digital converter (ADC) readings are left-justified in the data word format. The MSB is the sign bit: zero indicates a positive data word value.
In the unipolar format, the three trailing zeroes cause the data word to change by a count of eight for each one-count change in the ADC value.
In the bipolar format, the four trailing zeroes cause the data word to change by a count of sixteen for each one count change in the ADC value.

Table 4-4-4 SM231-0HF analog input module specifications

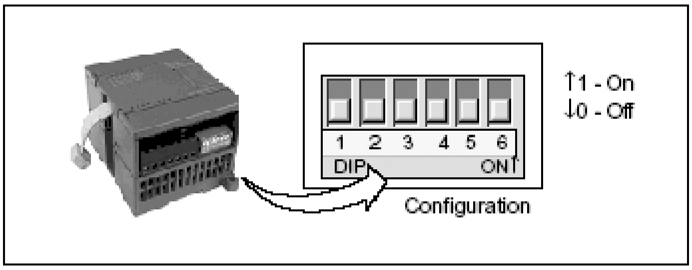
Features	
Dimension (W×H×D)	71.3 × 96 × 62mm
LED	24V power , ON: Normal, OFF: Faulted
Power	
+5VDC Consumption	87mA
L+	50mA
L+ Coil Voltage	20.4~28.8VDC
Power consumption	2.5W
Analog Input	
Counts	8
Isolation (field to logic)	Opto-couplers isolation, 500V AC for 1 minute
Type	Differential
Signal range	
Voltage (Unipolar)	0~10V, 0~5V
Voltage (Bipolar)	±2.5V (channel 0~7)
Current	0~20mA (channel 6~7)
Data word format	
Unipolar	0~32000
Bipolar	-32000~32000
Resolution, full-scale	
Voltage (Unipolar)	2.5mV (0~10V range); 1.25mV(0~5V range)
Voltage (Bipolar)	2.5mV(±5V range); 1.25mV(±2.5V range)
Analog to digital conversion time	Approx. 10ms
Conversion time for 8 channels	100ms
Common mode rejection	40dB, DC - 60Hz
Common mode voltage	-12V≤ Signal voltage plus common mode voltage ≤+12V
DC Input impedance	Voltage ≥2MΩ Current 250Ω
Maximum input voltage	30VDC
Maximum input current	32mA
A/D Converter resolution	Unipolar 12bits, Bipolar 11bits+sign bit
Measure principle	successive approximation
Measuring error	0.5% (max)
Simulate input addressing	AIW

Wiring diagram



Range selection and software configuration

【Range selection switch position】



【Range selection】

The following table shows how to set the range of SM231 8AI with DIP switch. Use 1, 2 and 3 to select the analogy input range which can be set to the same.
Attention: The unused DIP switch SW4 ~ SW6 must be set to OFF.

Table 4-4-5 DIP switch configuration of SM231-0HF

Unipolar			**Full-range**	**Resolution**
SW1	**SW2**	**SW3**		
ON	OFF	ON	0-10V	300μV
ON	ON	OFF	0 - 5V	150μV
Bipolar			**Full-range**	**Resolution**
SW1	**SW2**	**SW3**		
OFF	OFF	ON	± 5V	150μV
OFF	ON	OFF	± 2.5 V	75μV

【Software Configuration】
For SM231 8AI×16 bits, reading in VW instead of AIW. The relative location of the module is different, so is the corresponding address. Calculation formula:

<u>x(VWx) = Slot no. × 64 + Input channel × 2</u>

The slot number of the address quick reference table corresponds to the installation location of the module. The first expansion module close to the CPU has the slot number of 0, the second expansion module has the slot number of 1, and so on. There are 8 input channels, from A to G, and the corresponding number is 0 to 7.

Table 4-4-6 Quick reference of address based on formula above:

VWxx	channel0	channel 1	channel 2	channel 3	channel 4	channel 5	channel 6	channel 7
Slot 0	VW0	VW2	VW4	VW6	VW8	VW10	VW12	VW14
Slot 1	VW64	VW66	VW68	VW70	VW72	VW74	VW76	VW78
Slot 2	VW128	VW130	VW132	VW134	VW136	VW138	VW140	VW142
Slot 3	VW192	VW194	VW196	VW198	VW200	VW202	VW204	VW206
Slot 4	VW256	VW258	VW260	VW262	VW264	VW266	VW268	VW270
Slot 5	VW320	VW322	VW324	VW326	VW328	VW330	VW332	VW334
Slot 6	VW384	VW386	VW388	VW390	VW392	VW394	VW396	VW398

Notice

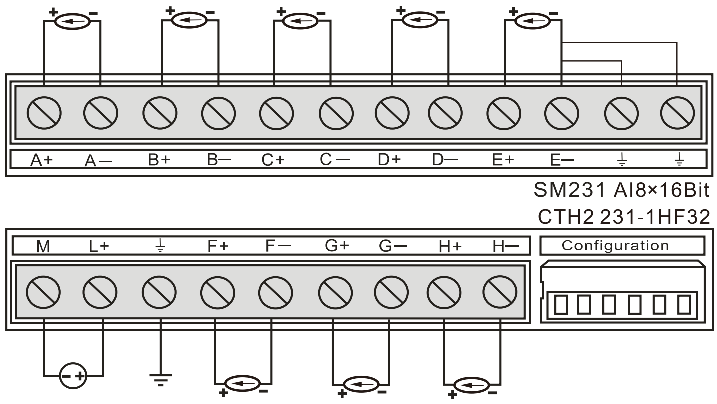
Since the address assigned by COTRUST's TD2X text display panel and SIEMENS 'TD200 text display panel is fixed at VW0, if you need to use both TD2X or TD200 and SM231 8AI modules in your system, the SM231 8AI module cannot be installed in the first position (Slot 0), otherwise it will not work properly.

Table 4-4-7 SM231-1HF analog input module specifications

Features	
Dimension(W×H×D)	120.5×80×62 mm
Weight	210 g
Insert I/O terminal	No
Power consumption	
Bus(5V DC)	87mA
L+	30MA
L+ voltage range	20.4~28.8V DC
Power	1W
Analog input features	
Counts	8
Type	Current
Range	0~20mA; 4~20mA
Data Word format	0~32000
Resolution	0.000625 mA (0~20mA); 0.0005 mA(4~20mA)
Measurement precision	0.1%
Input resistance	250Ω
Update time	100ms(8 channels)
CMV	120V AC
CMR	90dB, 60Hz @ DC
Input filter attenuation	-3db @ 325 HZ
Diagnosis	
Diagnosis program	+24VDC power indicator light, on: power supply normal off: no power, all the channel data turn to 32766 DIP switch wrong setting, all the channel data turn to 32767

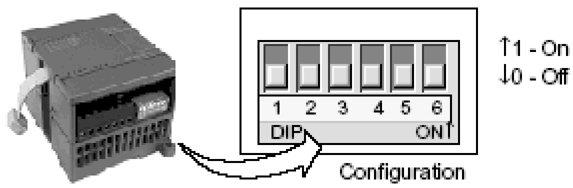
Isolation	
Field to logic	500V AC
Field to 24VDC	500V AC
24VDC to logic	500V AC

Wiring diagram



Range selection and software configuration

【Range selection switch position】



【Range Selection】

The following table shows how to set the range of SM231 8AI with DIP switch. Use 1, 2 and 3 to select the analogy input range which can be set to the same.

Attention: The unused DIP switch SW4 ~ SW6 must be set to OFF.

Table 4-4-8 DIP switch configuration of SM231-1HF

Unipolar			Full-range input	Resolution
SW1	SW2	SW3		
OFF	OFF	OFF	0-20mA	0.000625mA
	OFF	ON	4-20mA	0.0005mA

【Software Configuration】

For SM231 8AI×16 bits, reading in VW instead of AIW. The relative location of the module is different, so is the corresponding address. Calculation formula:

x(VWx) = Slot no. × 64 + Input channel × 2

The slot number of the address quick reference table corresponds to the installation location of the module. The first expansion module close to the CPU has the slot number of 0, the second expansion module has the slot number of 1, and so on. There are 8 input channels, from A to G, and the corresponding number is 0 to 7.

Table 4-4-9 Quick reference of address based on formula above:

VWxx	channel0	channel 1	channel 2	channel 3	channel 4	channel 5	channel 6	channel 7
Slot 0	VW0	VW2	VW4	VW6	VW8	VW10	VW12	VW14
Slot 1	VW64	VW66	VW68	VW70	VW72	VW74	VW76	VW78
Slot 2	VW128	VW130	VW132	VW134	VW136	VW138	VW140	VW142
Slot 3	VW192	VW194	VW196	VW198	VW200	VW202	VW204	VW206
Slot 4	VW256	VW258	VW260	VW262	VW264	VW266	VW268	VW270
Slot 5	VW320	VW322	VW324	VW326	VW328	VW330	VW332	VW334
Slot 6	VW384	VW386	VW388	VW390	VW392	VW394	VW396	VW398

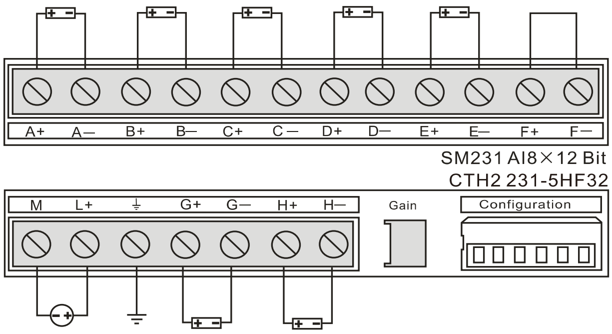
Table 4-4-10 SM231-5HF specifications

Features	
Dimension (W×H×D)	71.3 × 96 × 62mm
LED	24V power , ON: normal, OFF: faulted
Power	
+5VDC Consumption	87mA
L+	50mA
L+ Coil Voltage	20.4~28.8VDC
Power consumption	2.5W
Analog Input	
Counts	8
Isolation (field to logic)	Opto-couplers isolation, 500V AC for 1 minute
Type	Differential
Signal range	
Voltage (Unipolar)	0~10V, 0~5V
Voltage (Bipolar)	±2.5V (channel 0~7)
Current	0~20mA (channel 6~7)
Data word format	
Unipolar	0~32000
Bipolar	-32000~32000
Resolution, full-scale	
Voltage (Unipolar)	2.5mV (0~10V range); 1.25mV(0~5V range)
Voltage (Bipolar)	2.5mV(±5V range); 1.25mV(±2.5V range)
Analog to digital conversion time	Approx. 10ms
Conversion time for 8 channels	100ms
Common mode rejection	40dB, DC - 60Hz
Common mode voltage	-12V≤ Signal voltage plus common mode voltage ≤+12V
DC Input impedance	Voltage ≥2MΩ Current 250Ω
Maximum input voltage	30VDC
Maximum input current	32mA

A/D Converter resolution	Unipolar 12bits, Bipolar 11bits+sign bit
Measure principle	successive approximation
Measuring error	0.5% (max)
Simulate input addressing	AIW

Wiring diagram

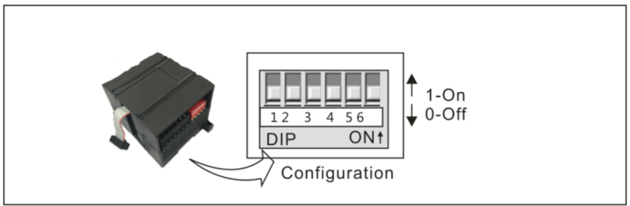
SM231 high precision AI module (CTH2 231-5HF32)



<Note> Short connect the unused terminal in the figure above, for F+ and F-.

Range selection and software configuration

【Range selection switch position】



【Range selection】

The following table shows how to set the range of SM231 8AI with DIP switch. Use 3, 4 and 5 to select the analogy input range, use 1 and 2 to select current input mode(ON means current input mode for channel 6, OFF means voltage mode), for switch 2, ON means current input mode for channel 7, OFF means voltage mode.

Attention: The unused DIP switch SW4 ~ SW6 must be set to OFF.

Table 4-4-11 DIP switch configuration of SM231-5HF

Unipolar			**Full-range**	**Resolution**
SW3	**SW4**	**SW5**		
ON	OFF	ON	0-10V	2.5mV
ON	ON	OFF	0-5V	1.25mV
			0-20mA	5μA
Bipolar			**Full-range**	**Resolution**
SW3	**SW4**	**SW5**		
OFF	OFF	ON	±5V	2.5mV
OFF	ON	OFF	±2.5V	1.25mV

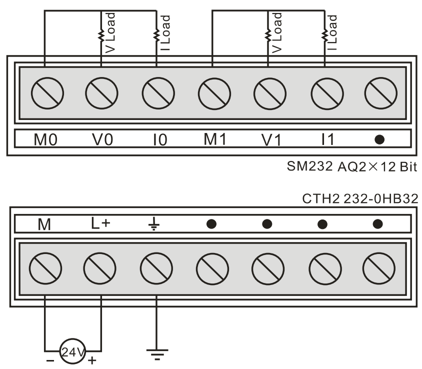
4.4.2 Analog Output Module Specification

Table 4-4-12 Analog output modules specifications

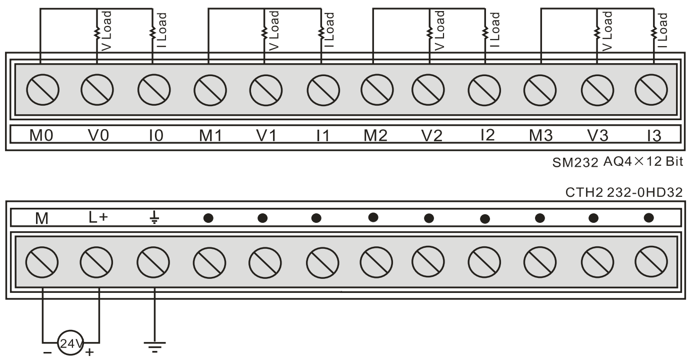
Modules	SM232: 2 AQ × 12 bits	SM232: 4 AQ × 12 bits
Order No.	CTH2 232-0HB32	CTH2 232-0HD32
Features		
Dimension (W×H×D)	46 × 96 × 62mm	71.3 × 96 × 62mm
LED	24V indicator, ON: Normal, OFF: Faulted	
Power consumption		
+5VDC current	87mA	
L+	61mA	112mA
L+ Coil Voltage	20.4~28.8VDC	
Power consumption	2W	
Analog Output		
Counts	2	4
Isolation (field to logic)	Opto-couplers isolation, 500V AC for 1 minute	
Output range		
Voltage	±10V	
Current	0~20mA	
Resolution		
Voltage	12 bits	
Current	11 bits	
Data Word Format		
Voltage	-32000~+32000	
Current	0~32000	
Measurement error	Typical: Full scale ±0.5%; Worst case: Full scale ±2%	
Stable time		
Voltage	100μs	
Current	2ms	
Maximum drive@24V user power		
Voltage	Minimum 5000Ω	
Current	Maximum 500Ω	

Wiring diagram

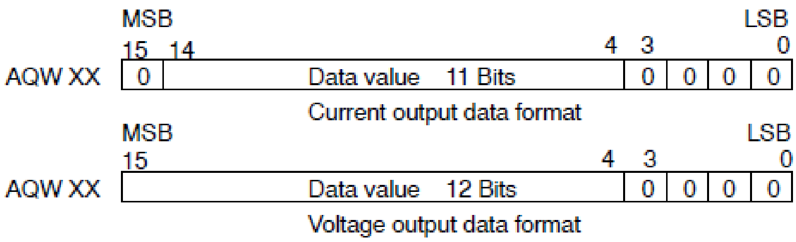
SM232 analog output modules (CTH2 232-0HB32)



SM232 analog output modules (CTH2 232-0HD32)



Output Data Format



Notice

The 12 bits of the digital-to-analog converter (DAC) readings are left-justified in the output data word format. The MSB is the sign bit: zero indicates a positive data word value. The four trailing zeroes are truncated before being loaded into the DAC registers. These bits have no effect on the output signal value.

4.4.3 Analog I/O Modules Specifications

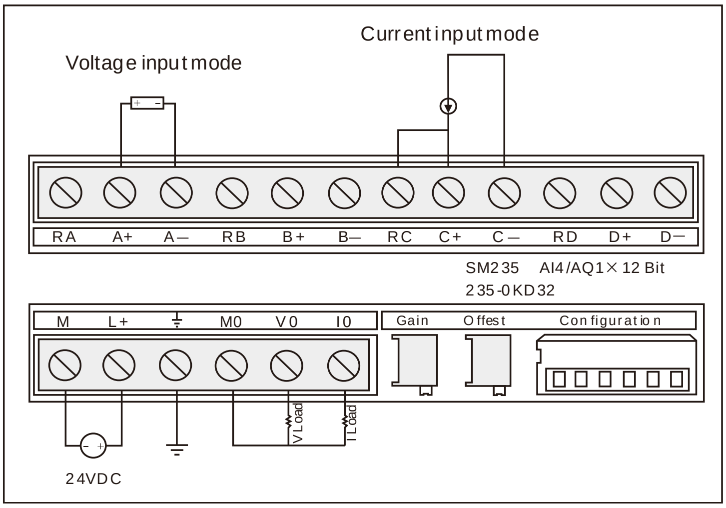
Table 4-4-13 Analog I/O modules specifications

Features	
Dimension (W×H×D)	71.3 × 96 × 62mm
LED	24V power , ON: normal, OFF: faulted
Power Supply	
+5VDC current	87mA
L+	48mA
L+ Coil Voltage	20.4~28.8VDC
Power consumption	2W
Analog Input	
Counts	4
Isolation (field to logic)	Opto-couplers isolation, 500V AC for 1 minute
Type	Differential input
Scale Range	
Voltage input (Unipolar)	0-10V, 0-5V, 0-1V, 0-500mV, 0-100mV, 0-50mV
Voltage input (bipolar)	±10V,±5V,±2.5V,±1V,±500mV, ±250mV, ±100mV,±50mV,±25mV

Current input	0~20mA
Data word Format (Full-Scale)	
Unipolar	0~32000
Bipolar	-32000~32000
Input Resolution	
Voltage input (Unipolar)	2.5mV(0~10V range); 1.25mV(0~5V range)
Voltage input (bipolar)	2.5mV(±5V range); 1.25mV(±2.5V range)
Current input	5μA(0~20mA range)
Analog to digital conversion time	< 300μs
Analog input step response	1.5ms
Common mode rejection	40dB, DC~60Hz
Common mode voltage	Signal voltage plus common mode voltage must less than 12V
DC Input impedance	> 10MΩ
Maximum input voltage	30V
Maximum input current	30mA
A/D converter resolution	12 bits
Analog Output	
Counts	1
Voltage output	±10V
Current output	0~20mA
Output resolution	
Voltage output	12 bits
Current output	11 bits
Data Word format	
Voltage output	-32000~+32000
Current output	0~32000
Accuracy	
Voltage output	Typical: Full scale ±0.5%, worst case: Full scale ±2%
Current output	
Settling time	
Voltage output	100μs
Current output	2ms
Max drive@24V DC	
Voltage output	Min 5000Ω
Current output	Max 500Ω

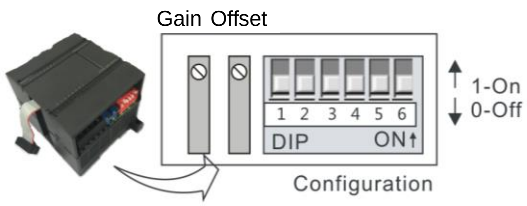
Wiring diagram

SM235 analog hybrid I/O module (CTH2 235-0KD32)



Calibration, range and gain selection

【Calibration and DIP configuration】



【Input Calibration】

Because the calibration adjustment affects the OPAMP behind the analog multiplexer, it affects all user input channels.In addition, because the component parameters of the input channels in front of the multiplexer may differ, there is a slight difference of the same input signal on different channels even after calibration.

The input has been filtered in the module, so the measurements is stable. Start the analog input filter for all input of the module for better performance parameters. When calculating the average value, select the sampling times above 64.

Calibration steps:

1) Cut off external power of modules, configure DIP switch to select needed range.

2) Connect CPU with module power, wait for 15min above.

3) Use a transmitter, a voltage input source or a current input source to add the zero signal to module input.

4) Read measurement of the input in CPU.

5) Adjust OFFSET potentiometer till the reading is zero, finish zero set calibration.

6) Input a full scale signal to read the measurement in CPU.

7) Adjust GAIN potentiometer till the reading is 32000.

8) Repeat 3~7 when necessary.

【Range and gain selection】

Table 4-4-14 shows how to configure the SM235 module using DIP switches. Use SW1 and SW6 to select the analog input range and resolution. Set all inputs the same analog input range and format. Table 4-4-16 shows how to select unipolar/bipolar (SW6), GAIN (SW4 and SW5) and attenuation (SW1, SW2 and SW3). ON is closed, OFF is open.

Table 4-4-14 SM235-0KD DIP switch configuration(unipolar)

Unipolar						Full-range	Resolution
SW1	**SW2**	**SW3**	**SW4**	**SW5**	**SW6**		
ON	OFF	OFF	ON	OFF	ON	0 - 50mV	12.5μV
OFF	ON	OFF	ON	OFF	ON	0 - 100mV	25μV
ON	OFF	OFF	OFF	ON	ON	0 - 500mV	125μV
OFF	ON	OFF	OFF	ON	ON	0 - 1V	250μV
ON	OFF	OFF	OFF	OFF	ON	0 - 5V	1.25μV
ON	OFF	OFF	OFF	OFF	ON	0 - 20mA	5μA
OFF	ON	OFF	OFF	OFF	ON	0 - 10V	2.5mV

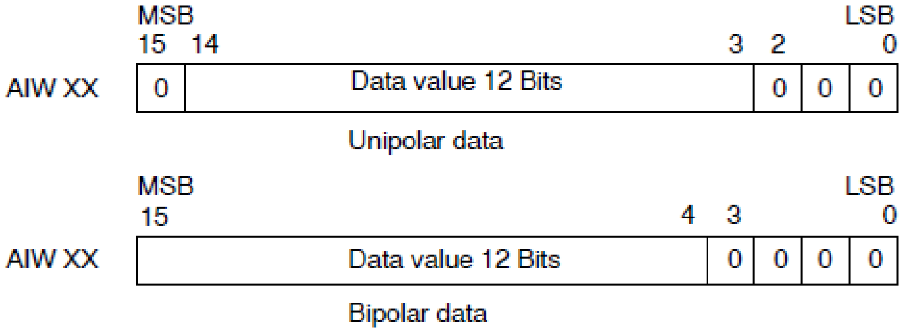
Table 4-4-15 SM235-0KD DIP switch configuration(bipolar)

Bipolar						Full-range	Resolution
SW1	**SW2**	**SW3**	**SW4**	**SW5**	**SW6**		
ON	OFF	OFF	ON	OFF	OFF	±25mV	12.5μV
OFF	ON	OFF	ON	OFF	OFF	±50mV	25μV
OFF	OFF	ON	ON	OFF	OFF	±100mV	50μV
ON	OFF	OFF	OFF	ON	OFF	±250mV	125μV
OFF	ON	OFF	OFF	ON	OFF	±500mV	250μV
OFF	OFF	ON	OFF	ON	OFF	±1V	500μV
ON	OFF	OFF	OFF	OFF	OFF	±2.5V	1.25mV
OFF	ON	OFF	OFF	OFF	OFF	±5V	2.5mV
OFF	OFF	ON	OFF	OFF	OFF	±10V	5mV

Table 4-4-16 SM235-0KD unipolar/bipolar, gain and attenuation configuration

SM235 switch						Unipolar/bipolar	Gain	Attenuation
SW1	**SW2**	**SW3**	**SW4**	**SW5**	**SW6**			
---	---	---	---	---	ON	Unipolar	---	---
---	---	---	---	---	OFF	Bipolar	---	---
---	---	---	OFF	OFF	---	---	x1	---
---	---	---	OFF	ON	---	---	x10	---
---	---	---	ON	OFF	---	---	x100	---
---	---	---	ON	ON	---	---	--	---
ON	OFF	OFF	---	---	---	---	---	0.8
OFF	ON	OFF	---	---	---	---	---	0.4
OFF	OFF	ON	---	---	---	---	---	0.2

Input Data Format



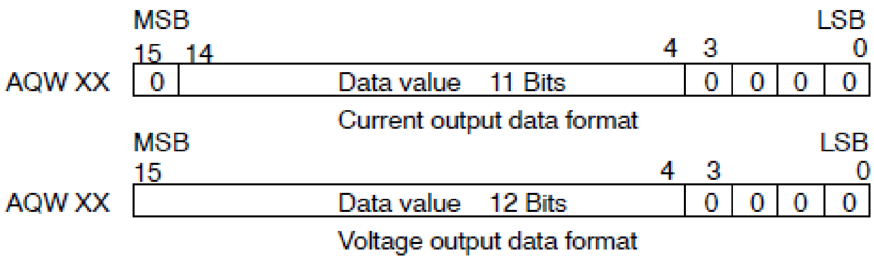
Notice

The 12 bits of the analog-to-digital converter (ADC) readings are left-justified in the data word format. The MSB is the sign bit: zero indicates a positive data word value.

In the unipolar format, the three trailing zeros cause the data word to change by a count of eight for each one-count change in the ADC value.

In the bipolar format, the four trailing zeros cause the data word to change by a count of sixteen for each one count change in the ADC value.

Output Data Format



Notice

The 12 bits reading of the digital-to-analog converter (DAC) are left-justified in the output data word format. The MSB is the sign bit: zero indicates a positive data word value. The four trailing zeroes are truncated before being loaded into the DAC registers. These bits have no effect on the output signal value.

4.5 Thermocouple and RTD Expansion Module Specifications

Table 4-5-1 Hybrid temperature input modules order No.

Modules	Order No.
SM231 thermal resistance temperature input module, 2 RTD, isolated 16 bits precision	CTH2 231-7PB32
SM231 thermal resistance temperature input module, 4 RTD, isolated 16 bits precision	CTH2 231-7PC32
SM231 thermocouple temperature input module, 4 TC, J/K/R/S/T/E/N, isolated 16 bits precision	CTH2 231-7PD32
SM231 thermocouple temperature input module, 8 TC, J/K/R/S/T/E/N, isolated 16 bits precision	CTH2 231-7PF32

SM231 hybrid temperature input module, 2NTC/PT100, 2 points 0~20mA current or ±5V, ±10V, 0~10V, 0~5V voltage input, isolated 16 bits precision	CTH2 231-7ND32
SM231 thermal resistance temperature input module, 8NTC/PT100, isolated 16 bits precision	CTH2 231-7NF32

4.5.1 RTD Module Specifications

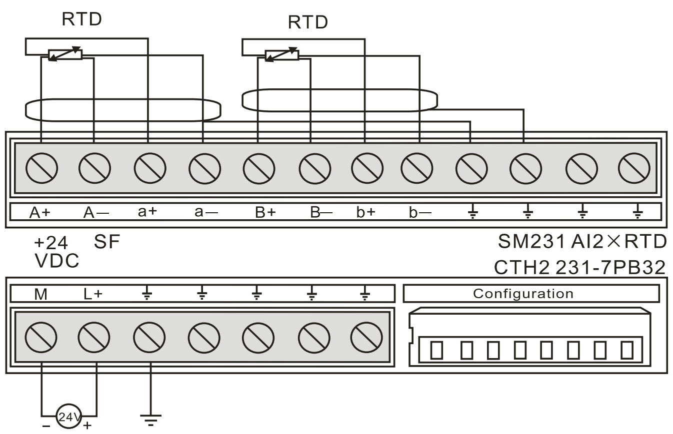
Table 4-5-2 RTD modules specifications

Modules	SM231: 2AI×RTD	SM231: 4AI×RTD
Order No.	CTH2 231-7PB32	CTH2 231-7PC32
Features		
Dimension (W×H×D)	71.3 × 96 × 62mm	
LED	24VDC Power indicator, ON=No Fault, OFF=No 24VDC Power; SF indicator, ON= Module fault; Flash = Override or open wire , OFF=No Fault	
Power supply		
+5VDC current	87mA	
L+	34mA	37mA
L+ Coil Voltage	20.4~28.8VDC	
Power	1.7W	
Input		
Type	Module ground referenced RTD	
Counts	2	4
Input range	RTD types (select one per module): Pt-100Ω, 200Ω, 500Ω, 1000Ω(α=3850ppm, 3920ppm, 3850.55ppm, 3916ppm, 3902ppm) Pt-10000Ω(α=3850ppm) Cu-9.035Ω(α=4720ppm) Ni-100Ω, 120Ω, 1000Ω(α=6720ppm, 6178ppm) R-150Ω, 300Ω, 600Ω	
RTD measuring range	Pt-100Ω, 200Ω, 500Ω, 1000Ω: -200℃~850℃ Pt-10000Ω: -200℃~600℃ NI-0.00672: -80℃~260℃ NI-0.006178: -60℃~300℃ Cu-0.004270: -200℃~260℃ <Note> Error when exceeding this range, refer Table 4-30 for Diagnostic details.	
Isolation		
Field to logic	500VAC	
Field to 24 VDC	500VAC	
24 VDC to logic	500VAC	
Common mode input range (input channel to input channel)	0	

Common mode rejection	>120dB@120VAC	
Input resolution		
Temperature	0.1℃/0.1℉	
Voltage	15bits plus sign	
Measuring principle	Σ-Δ	
Module update time: All channels	425ms	825ms
Wire length to sensor	Max 100m	
Wire loop resistance	20Ω, 2.7Ω for Cu	
Noise suppression	85dB@ 50Hz/60Hz/400Hz	
Data word format		
Input impedance	>10MΩ	
Maximum input voltage	30VDC(sense), 5VDC(source)	
Resolution	15bits plus sign	
Input Filter attenuation	-3dB@21kHz	
Basic error	0.1% FS (resistance)	
Repeatability	0.05% FS	
AI Addressing method	AIW Addressing	

Wiring diagram

SM231 thermal resistance input module 2AI x RTD (CTH2 231-7PB32)



SM231 thermal resistance input module 4AI x RTD (CTH2 231-7PC32)

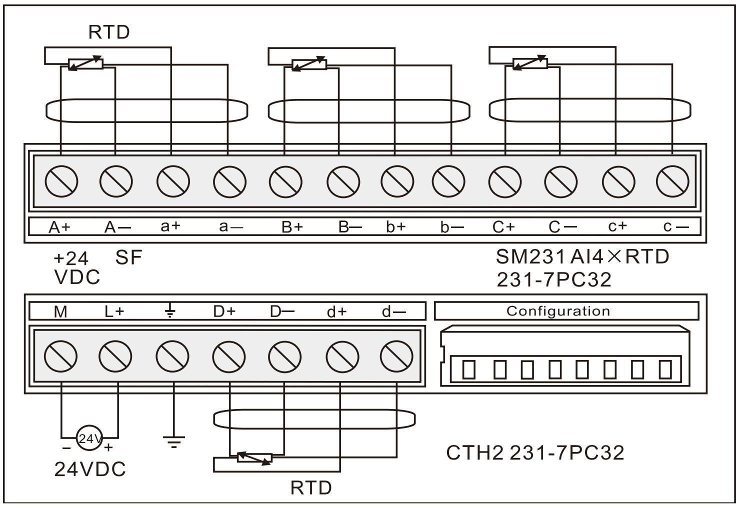


Table 4-5-3 RTD module diagnostic messages

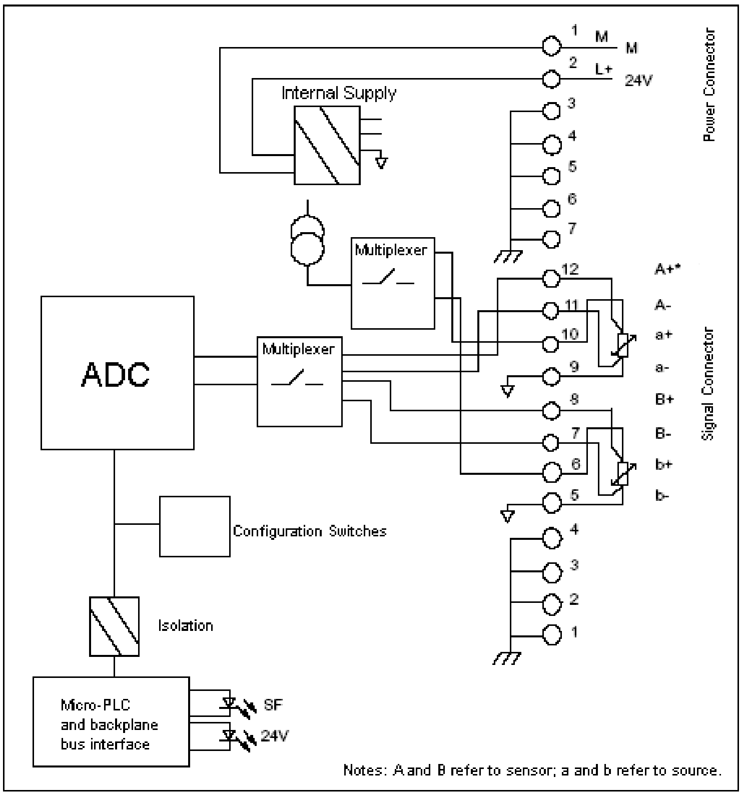
Type	Channel Data	SF LED	24V LED	Range status bit	24V power failure
No power module	32766	OFF	OFF	0	1
Disconnection	32767(Upscale) -32768(Downscale)	Flash	ON	1	0
Out of temperature range	32767(Upscale) -32768(Downscale)	Flash	ON	1	0

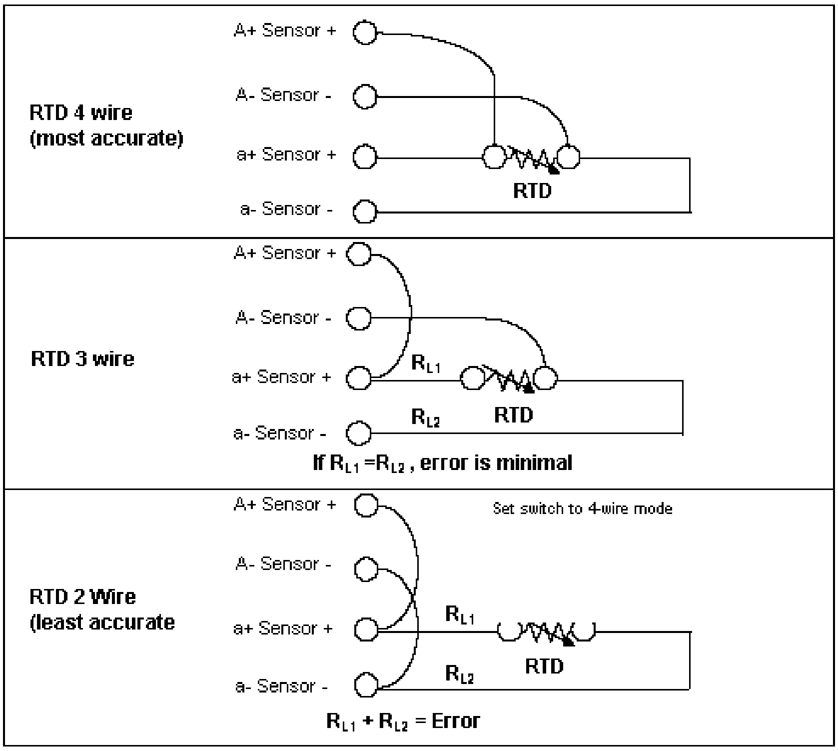
<Note> For relevant error code of each Module, please check the SMB8~21 according to modules sequence.

Table 4-5-4 SMB8~21 Diagnose Information

SMB8	Module 1 flag register
SMB9	Module 1 error register
SMB10	Module 2 flag register
SMB11	Module 2 error register
SMB12	Module 3 flag register
SMB13	Module 3 error register
SMB14	Module 4 flag register
SMB15	Module 4 error register
SMB16	Module 5 flag register
SMB17	Module 5 error register
SMB18	Module 6 flag register
SMB19	Module 6 error register
SMB20	Module 7 flag register
SMB21	Module 7 error register

Wiring between RTD and sensor





Notes: A refers to sensor; a refers to source

You can either directly connect the thermal resistance sensor to the SM231 module of CTH200 or use expansion wiring. The shielded wire, connect it to the ground point of PIN1-4 on signal connector if you use, can reach the best noise resistance.

For unused channel, wire it with a resistor in place of the RTD to prevent open wire detection from causing the SF LED to blink. The resistor must be the nominal value of the RTD.

Connect power supply to the PIN 1 and 2 of connector, and PIN 3 to its nearby shell. There are 3 methods for connecting RTD module to the sensor. Among that, the RTD 4-Wire has the best accuracy while the RTD 2-Wire has the lowest, therefore it's recommended to only using RTD 2-Wire in application which not emphasis accuracy.

【RTD module Configuration】

SM231 module allows CTH200 series PLC to connect with multiple thermal resistance sensor. Use DIP switches to select RTD type, wiring method, temperature scale and burnout direction. The DIP switches are located on the bottom of the module as shown in Figure bellow. For the DIP switch settings to take effect, you need to power cycle the PLC and/or the user 24V power supply.

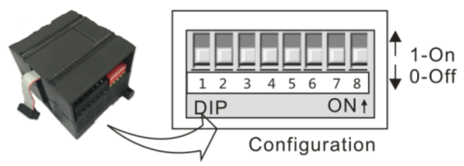


Table 4-5-5 SM231 RTD DIP switch configuration

RTD Type	SW1	SW2	SW3	SW4	SW5
100 Pt 0.003850(Default)	0	0	0	0	0
200Ω Pt 0.003850	0	0	0	0	1
500Ω Pt 0.003850	0	0	0	1	0
1000Ω Pt 0.003850	0	0	0	1	1
100Ω Pt 0.003920	0	0	1	0	0
200Ω Pt 0.003920	0	0	1	0	1
500Ω Pt 0.003920	0	0	1	1	0
1000Ω Pt 0.003920	0	0	1	1	1
100Ω Pt 0.00385055	0	1	0	0	0
200Ω Pt 0.00385055	0	1	0	0	1
500Ω Pt 0.00385055	0	1	0	1	0
1000Ω Pt 0.00385055	0	1	0	1	1
100Ω Pt 0.003916	0	1	1	0	0
200Ω Pt 0.003916	0	1	1	0	1
500Ω Pt 0.003916	0	1	1	1	0
1000Ω Pt 0.003916	0	1	1	1	1
100Ω Pt 0.00302	1	0	0	0	0
200Ω Pt 0.003902	1	0	0	0	1
500Ω Pt 0.003902	1	0	0	1	0
1000Ω Pt 0.003902	1	0	0	1	1
Reserved	1	0	1	0	0
100Ω Ni 0.00672	1	0	1	0	1
120Ω Ni 0.00672	1	0	1	1	0
1000Ω Ni 0.00672	1	0	1	1	1
100Ω Ni 0.006178	1	1	0	0	0
120Ω Ni 0.006178	1	1	0	0	1
1000Ω Ni 0.006178	1	1	0	1	0

10000Ω Pt 0.003850	1	1	0	1	1
10Ω Cu 0.004270	1	1	1	0	0
150Ω FS Resistance	1	1	1	0	1
300Ω FS Resistance	1	1	1	1	0
600Ω FS Resistance	1	1	1	1	

Table 4-5-6 DIP switch settings

SW6	Calibration	SW7	Scale Unit	SW8	Wiring method
0	Upscale (+3276.7 degrees)	0	Celsius (℃)	0	3-wire
1	Downscale (-3276.8 degrees)	1	Fahrenheit (℉)	1	2-wire or 4-wire

4.5.2 Thermocouple Module Specifications

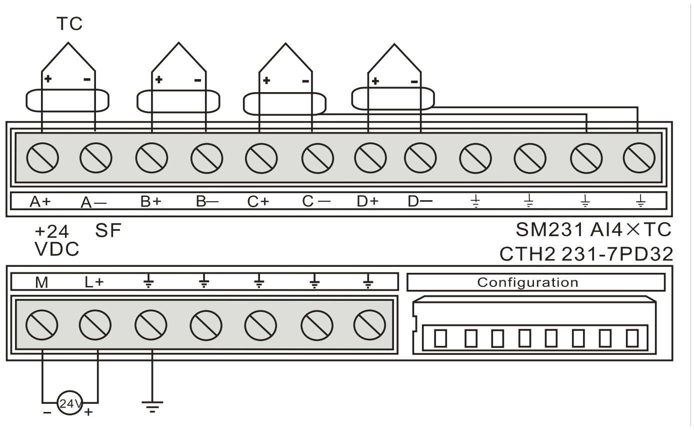
Table 4-5-7 SM231 Thermocouple Module Specifications

Modules	SM231: 4AI×TC	SM231: 8AI×TC
Features		
Dimension (W×H×D)	71.3 × 96 × 62mm	
LED	24VDC indicator: ON=No fault, OFF=no 24VDC power SF Indicator: ON=Module fault, Flash=Outrange or , OFF=No fault	
Power		
+5VDC current	87mA	
L+	30mA	
L+ coil voltage range	20.4~28.8VDC	
Power	1.7W	
Input		
Input type	Float TC	
Counts	4	8
Input range	TC type: S, T, R, E, N, K, J Voltage range: +/-80mV **<Note>** For temperature measuring range, please refer to the TC measuring range in the end of this section. For error details, please refer to the Table 4-5-8 and 4-5-9.	
Isolation		
Field to Logic	500VAC	
Field to 24VDC	500VAC	
24V to Logic	500VAC	
Common mode input range (input channel to input channel)	120VAC	
Common mode rejection	>120dB@120VAC	
Sampling features		
Temperature resolution	0.1℃/0.1℉	
Voltage resolution	15 bits + sign bit	

Transfer principle	Σ-Δ	
Module update time: All channels	425ms	825ms
Wire length to sensor	Max. 100m	
Conductor loop resistance	Max. 100Ω	
Suppression of interference	85dB@ 50Hz/60Hz/400Hz	
Data word format	Voltage: -27648~+27648	
Input impedance	>1MΩ	
Max. Input voltage	30VDC	
Input Filter attenuation	-3dB@ 21kHz	
Basic error	0.1% FS(voltage)	
Repeatability	0.05% FS	
cold junction error	±1.5℃	
Addressing method	AIW Addressing	VW Addressing

Wiring diagram

SM231 thermocouple input module (CTH2 231-7PD32) x 4TC



SM231 thermocouple input module (CTH2 231-7PF32) x 8TC

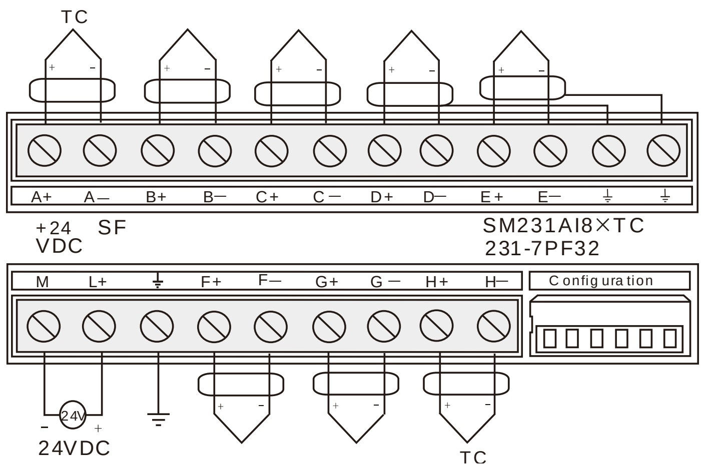


Table 4-5-8 TC module Diagnose

Error type	Channel data	SF LED	24V LED	Range status bit	24V Power fault

No power	32766	OFF	OFF	0	1
Open Wire	32767 (Upscale) -32768 (Downscale)	Flash	ON	1	0
Temperature Outrange	32767 (Upscale) -32768 (Downscale)	Flash	ON	1	0

<Note> For relevant error code of each Module, please check the SMB8~21 according the modules sequence.

Table 4-5-9 SMB8~21 Diagnose information

SMB8	Module 1 flag register
SMB9	Module 1 error register
SMB10	Module 2 flag register
SMB11	Module 2 error register
SMB12	Module 3 flag register
SMB13	Module 3 error register
SMB14	Module 4 flag register
SMB15	Module 4 error register
SMB16	Module 5 flag register
SMB17	Module 5 error register
SMB18	Module 6 flag register
SMB19	Module 6 error register
SMB20	Module 7 flag register
SMB21	Module 7 error register

TC measuring range

Data byte (1 digit=0.1°C) Dec	Hex	Type J	Type K	Type T	Type E	Type R,S	类型N	±80mV	
32767	7FFF	>1200.0°C	>1372.0°C	>400.0°C	>1000.0°C	>1768.0°C	>1300.0°C	>94.071mV	OF
↑	↑							↑	↑
32511	7EFF							97.071mV	OR
:	:								
27649	6C01							80.0029mV	
27648	6C00					↑		80mV	
:	:								
17680	4510		↑			1768.0°C			
:	:								
13720	3598		1372.0°C Overrange				↑		
:	:								
13000	32C8	↑	1300.0°C				1300.0°C		NR
:	:								
12000	2EE0	1200.0°C							
:	:				↑				
10000	2710			↑	1000.0°C				
:	:								
4000	0FA0			-400.0°C		400.0°C			
:	:								
1	0001	0.1°C	0.1°C	0.1°C	0.1°C	0.1°C	0.1°C	0.0029mV	
0	0000	0.0°C	0.0°C	0.0°C	0.0°C	0.0°C	0.0°C	0.0mV	
-1	FFFF	-0.1°C	-0.1°C	-0.1°C	-0.1°C	-0.1°C	-0.1°C	-0.0029mV	
:	:					Underrange			
-500	FE0C					-50.0°C			
-1500	FA24	-150.0°C				↓			
:	:								
-2000	F830	Underrange	-200.0°C						
:	:								
-2100	F7CC	-210.0°C	Underrange						
:	:								
-2550	F60A			-255.0°C	-255.0°C				
:	:			Underrange	Underrange				
-2700	F574	↓	-270.0°C	-270.0°C	-270.0°C		-270.0°C		
:	:								
-27648	9400		↓	↓	↓		↓	-80mV	
-27649	93FF							-80.0029mV	UR
:	:								
-32512	8100							-94.071mV	
#	#							↓	↓
-32768	8000	<-210.0°C	<-270.0°C	<-270.0°C	<-270.0°C	<-50.0°C	<-270.0°C	<-94.07mV	UF
Accuracy over full span		S0.1%	S0.3%	S0.6%	S0.1%	S0.6%	S0.1%	S0.1%	
Accuracy (normal range without cold junction)		S1.5°C	S1.7°C	S1.4°C	S1.3°C	S3.7°C	S1.6°C	S0.10°C	
Cold junction error		S1.5°C	S1.5°C	S1.5°C	S1.5°C	S1.5°C	S1.5°C	NA	

*OF = Overflow; OR = Overrange; NR = Normal range; UR = Underrange; UF = Underflow

↑ indicates that all analog values greater than this and below the open wire threshold report the overflow data value, 32767 (0x7FFF).
↓ indicates that all analog values less than this and greater than the open wire threshold report the underflow data value, --32768 (0x8000).

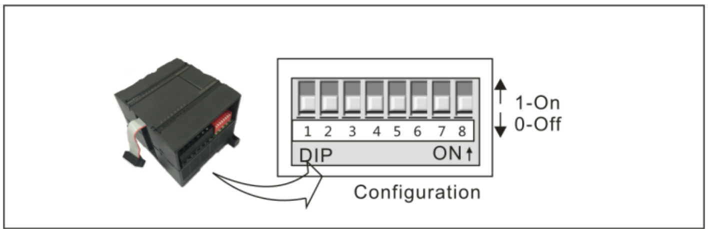
DIP switch and software configuration

● DIP switch configuration

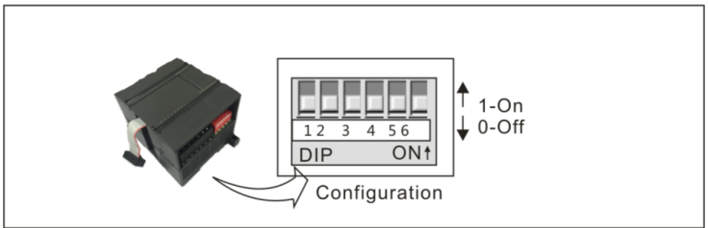
SM231 TC module enjoys 7 types thermocouple: J, K, E, N, S, T and R. It has a convenient isolated interface and allow to connect micro analog signal (range ±80mV), All TC on module must be the same type and it's better to use isolated TC sensor.

Users can use DIP switch to configure TC type, Open Wire detection, Scale Unit, Cold junction compensation and open-circuit fault direction as shown in the following figures.

For SM231 4TC module, SW1~SW3 are used for selecting TC type, SW4 remained OFF (unused), SW5 used for Open Wire Detect Direction, SW6 for Open Wire Detect Enable, SW7 for Scale Unit, SW8 for Cold junction.



For SM231 8TC module, SW1~SW3 are used for selecting TC type, SW4 for Open Wire Detect Direction, SW5 for Scale Unit, SW6 for Cold junction compensation. There is no need for user to set Open Wire Detect Enable for it has been fixed yes.



The DIP configuration can be effective by cut-off and recycle the PLC/user's power supply.

Table 4-5-10 Selecting TC type:

TC type	SW1	SW2	SW3
J(default)	0	0	0
K	0	0	1
T	0	1	0
E	0	1	1
R	1	0	0
S	1	0	1
N	1	1	0
+/- 80mV	1	1	1

Table 4-5-11 SM231 Module DIP Configuration

Items	SM231 4TC		SM231 8TC	
	Switch	Setting	Switch	Setting
Open Wire Detect Direction	SW5	0: Upscale (+3276.7 Degrees)	SW4	0: Upscale (+3276.7 Degrees)
		1: Downscale (-3276.8 Degrees)		1: Downscale (-3276.8 Degrees)
Open Wire Detect Enable	SW6	0: Yes, 1: No	Fixed at Enable	
Scale Unit	SW7	0: Celsius, 1:Fahrenheit	SW5	0: Celsius, 1:Fahrenheit
Cold Junction	SW8	0: Yes, 1: No	SW6	0: Yes, 1: No

● Software Configuration

For SM231 8AI×TC Input module, readings were saved in VW more than AIW, the specific address depends on module location, calculated as following:

x(VWx) = Slot no. × 64 + Input channel no.× 2

Slot number corresponding with installation location, and the slot 0 is the first expansion module nearest to CPU, then the second is slot 1, and so on. There are all 8 Input channels from A to H and numbered as 0 to 7.

Based on the above equation, there is the address look-up table.

Table 4-5-12 SM231 TC module address look-up table

VWxx	Channel 0	Channel 1	Channel 2	Channel3	Channel 4	Channel5	Channel6	Channel 7
Slot 0	VW0	VW2	VW4	VW6	VW8	VW10	VW12	VW14
Slot 1	VW64	VW66	VW68	VW70	VW72	VW74	VW76	VW78
Slot 2	VW128	VW130	VW132	VW134	VW136	VW138	VW140	VW142
Slot 3	VW192	VW194	VW196	VW198	VW200	VW202	VW204	VW206
Slot 4	VW256	VW258	VW260	VW262	VW264	VW266	VW268	VW270
Slot 5	VW320	VW322	VW324	VW326	VW328	VW330	VW332	VW334
Slot 6	VW384	VW386	VW388	VW390	VW392	VW394	VW396	VW398

Tip

As COTRUST TD2X and SIEMENS TD200 have address fixed at VW0, if you need to use TD2X/TD200 and SM231 8TC Module simultaneously, the SM231 8TC module cannot be installed at Slot 0, or it cannot operate normally.

4.5.3 Hybrid Temperature Module Specifications

CTH2 231-7ND32 module specifications

SM231 NTC, the hybrid analog input extension module of CTH200 series PLC, provide 4-channel analog acquisition, two for connecting the thermistor NTC temperature sensor or thermal resistance PT100 temperature sensor, the other two for collecting voltage/current signal input, input accuracy of all channels (involve the sign bit) are 16 bits. Mainly used for sterilization equipment or central air conditioning equipment, both temperature measurement and pressure signal measurement needed occasions.

Table 4-5-13 SM231-7ND module specifications

Order No.	CTH2 231-7ND32
Features	
Dimension (W×H×D)	71.3 × 96 × 62mm
LED indicator	24VDC indicator: ON=No fault, OFF=No 24VDC power SF Indicator: ON=Module fault, Flash=Out range or disconnect, OFF=No fault
Power supply	
+5VDC current	87mA
L+	60mA
L+ coil voltage range	20.4~28.8VDC
Power	1.7W
Input	
Thermistor input range	Thermistor Type: Pt-100(3850ppm, 3920ppm, 3850.55ppm, 3916ppm, 3902ppm) NTC(R25=10kΩ/B=3950, R25=10kΩ/B=3435)
Temperature Range	Pt-100: -50℃~850℃ NTC(R25=10K, B=3950): -40℃~120℃ NTC(R25=10K, B=3435): -40℃~150℃
Voltage input	0V~5V, 0V~10V, ±5V, ±10V
Current input	0~20mA
Input counts	2PT100/2NTC and 2AI
Isolation	
Field to Logic	500VAC
Field to 24VDC	500VAC
24V to Logic	500VAC
Common mode rejection	>120dB@120VAC
Sampling	
Temperature resolution	0.1℃/0.1℉
Voltage resolution	15bits ＋ sign bit
Transfer principle	Σ-Δ
Module update time: All channels	425ms
Wire length to sensor	Max. 100m
Conductor loop resistance	Max. 20Ω
Suppression of interference	85dB@50Hz/60Hz/400Hz
Data word format	Temperature (NTC: R25=10kΩ, B=3950K): -400~1200(only for channel 1/2) Temperature (NTC: R25=10kΩ, B=3435K): -400~1500(only for channel 1/2) Temperature (PT100): -500~2000(only for channel 1/2) Voltage/Current: Unipolar 0~32000, Bipolar -32000~+32000(only for channel 3/4)

Input impedance	Voltage input>10MΩ; Current input=250Ω; NTC input>10MΩ
Max. Input voltage	30VDC(detect), 5VDC(source)
Input Filter attenuation	-3dB@21kHz
Basic error	0.1％FS(resistance)
Repeatability	0.05％FS
Addressing method	AIW

Application Environment

■ Temperature: horizontal installation 0-55℃, vertical installation 0-45℃

■ Humidity: 95% Non - Condensing

Usage

SM231 NTC can be used for expanding CTH200 CPU analog measurement, connecting with CPU by Bus interface.

Table 4-5-14 SM231-7ND module configuration

Input type	SW1	SW2	SW3	SW4	SW5
	Valid for channel 1 and 2			Valid for channel 3 and 4	
100ΩPT0.003850(default)	0	0	0	—	—
100ΩPT0.003920	0	0	1	—	—
100ΩPT0.00385055	0	1	0	—	—
100ΩPT0.003916	0	1	1	—	—
100ΩPT0.003902	1	0	0	—	—
NTC: R25=10kΩ, B=3950K	1	0	1	—	—
NTC :R25=10kΩ, B=3435K	1	1	0	—	—
Forbidden	1	1	1	—	—
0—5V	—	—	—	0	0
0—20mA	—	—	—	0	0
0—10V	—	—	—	0	1
-10V—10V	—	—	—	1	0
-5V—5V	—	—	—	1	1

Table 4-5-14 SM231-7ND module configuration

SW6	Calibration direction	SW7	Scale Unit	SW8	Wiring scheme
0	Positive (+3276.7°)	0	Celsius (℃)	0	3-wire
1	Negative (-3276.8°)	1	Fahrenheit (℉)	1	2-wire or 4-wire

【Startup Steps】

1) Wiring the L+ and M terminals with 24VDC power supply.

2) Connect the module to CPU via the I/O expansion bus.

3) Connect the CPU power supply and communication port.

4) Connect the input signals according to wiring diagram, leave terminals of unused channel dangling, ground the EARTH terminal according to Grounding specification (single-point ground).

5) Configure the sensor type and calibration direction as requested.

6) Switch on the power supply of CPU and the module.

【Obtain input values】

1) You can read 2 bytes of input data per channel from the user program or Configuration software (such as MagicWorks PLC of Cotrust or MicroWIN of Siemens).

2) NTC full scale reading: -400~1200(R25=10K B=3950), Temperature range: -40.0~120.0℃; -400~1500(R25=10K B=3435), Temperature range: -40.0~150.0℃

PT100 full scale reading: -500~2000, Temperature range: -50.0~200.0℃

Readings at outrange or open wire depend on the scale direction (-32768 or 32767).

3) Voltage/Current full-range reading: -32000~32000, overrange readings up to -32767/32765 are still valid.

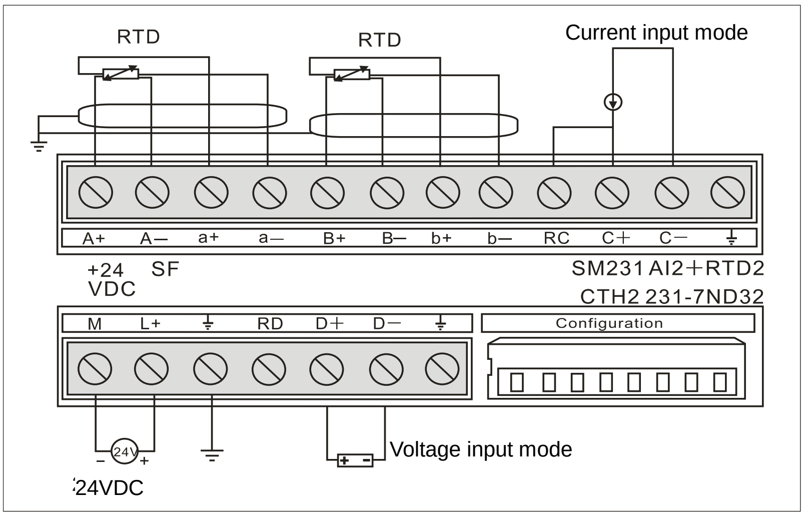
4) Readings for all channels are 32766 while no power.

Wiring Diagrams

SM231 temperature input module (CTH2 231-7ND32)

Wiring diagram

SM231 hybrid input module (CTH2 231-7ND32)



CTH2 231-7NF32 module specifications

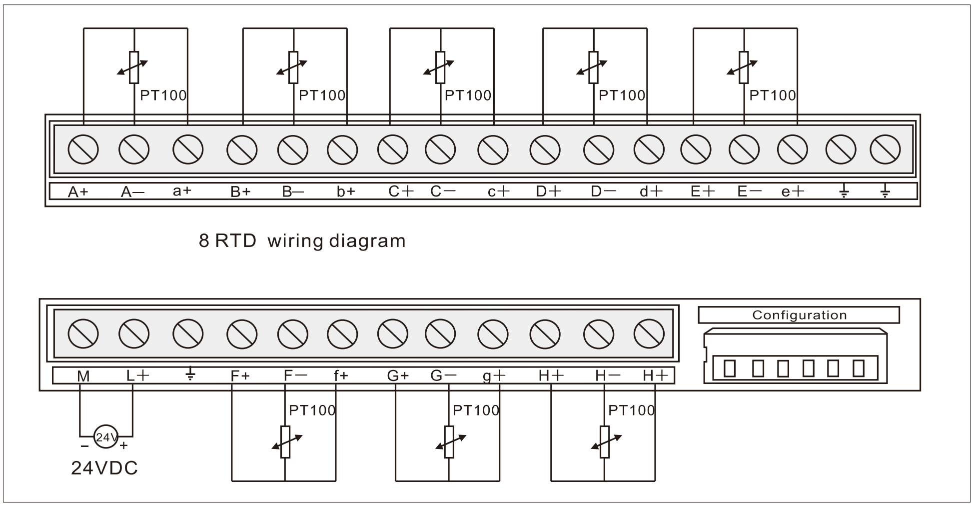
Table 4-5-15 SM231-7NF specifications

Order No.	CTH2 231-7NF32
Dimension (W x H x D)	71.3 x 96 x 62mm
LED indicator	24VDC indicator: ON=No fault, OFF=no 24VDC power SF Indicator: ON=Module fault, Flash=Outrange or line broken, OFF=No fault
Power supply	
+5VDC current	87mA
L+	32.5mA

L+ voltage range	20.4~28.8VDC
Power dissipation	1.8W
Input	
Input type	Pt100: α=3850PPm/3920PPM/3850.55PPM/3916PPM/3902PM NTC: R25=10k/B=3950 or R25=10k/B=3435
Temperature scale range	Pt100: -50℃~800℃ NTC(R25=10k/B=3950): -40℃~120℃ NTC(R25=10k/B=3435): -40℃~150℃ **<Note>** Error when exceeding this range, refer Table 4-30 for Diagnostic details.
Counts	8PT100/8NTC
Plug-in I/O terminal	Yes
Power isolation	Yes
Field-Logic	500VAC
Field-DC 24 V	500VAC
DC 24 V-Logic	500VAC
Common-mode rejection	120dB@120VAC
Sampling features	
Temperature resolution	0.1℃/0.1℉
Measurement principle	Σ-Δ
Module update time: All channels	825ms
Max Wire length	100m
Max Wire loop resistance	20Ω
Suppression of interference	85dB@50/400 Hz
Input impedance	>1KΩ
Measurement accuracy	±0.3℃
Open-wire detection	Pt100 detects 3 wires (NTC detects 2 wires), up to 3 minutes

Wiring diagram

SM231 temperature input module Pt100 (CTH2 231-7NF32)



SM231 temperature input module NTC (CTH2 231-7NF32)

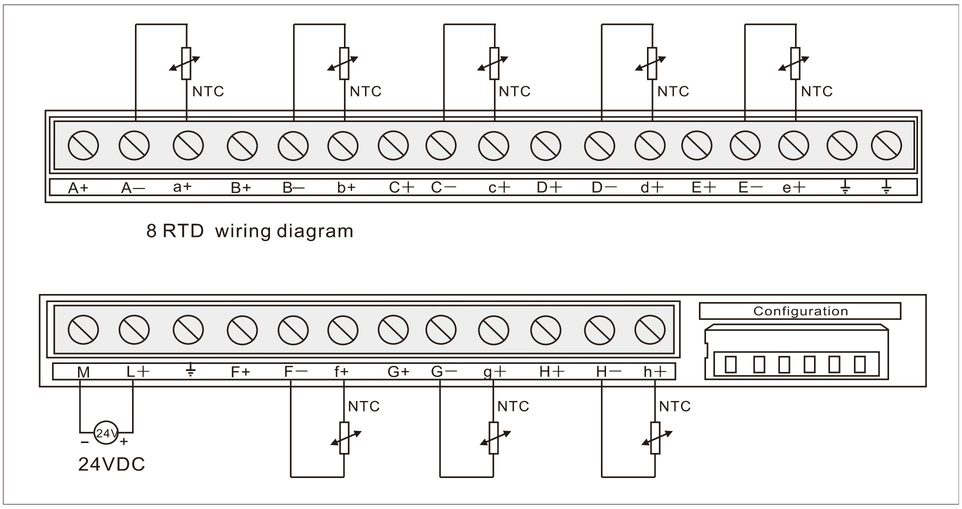


Table 4-5-16 SM231-7NF DIP switch configuration

Input type	SW1	SW2	SW3
100Ω Pt 0.003850(default)	**0**	**0**	**0**
100Ω Pt 0.003920	**0**	**0**	**1**
100Ω Pt 0.00385055	**0**	**1**	**0**
100Ω Pt 0.003916	**0**	**1**	**1**
100Ω Pt 0.00302	**1**	**0**	**0**
NTC R25=10K/B=3950	**1**	**0**	**1**
NTC R25=10K/B=3435	**1**	**1**	**0**
Forbidden	**1**	**1**	**1**

Table 4-5-16 SM231-7NF DIP switch configuration

SW4	Calibration direction	SW5	Scale Unit	SW6	Reserved
0	Positive (+3276.7 °)	0	Celsius (℃)	0	No influence
1	Negative (-3276.8 °)	1	Fahrenheit (℉)	1	No influence

Software configuration

For SM231 8AI×16bits Analog input module, readings are saved in VW instead of AIW, the specific address depends on module location, calculated as following:

x(VWx) = Slot no. × 64 + Input channel no.× 2

Slot number corresponding with installation location, and the slot 0 is the first expansion module nearest to CPU, then the second is slot 1, and so on. There are all 8 Input channels from A to H and numbered as 0 to 7.

The address look-up table based on equation above shows as below:

Table 4-5-17 SM231-7NF quick reference

VWxx	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
Slot 0	VW0	VW2	VW4	VW6	VW8	VW10	VW12	VW14
Slot 1	VW64	VW66	VW68	VW70	VW72	VW74	VW76	VW78
Slot 2	VW128	VW130	VW132	VW134	VW136	VW138	VW140	VW142
Slot 3	VW192	VW194	VW196	VW198	VW200	VW202	VW204	VW206
Slot 4	VW256	VW258	VW260	VW262	VW264	VW266	VW268	VW270
Slot 5	VW320	VW322	VW324	VW326	VW328	VW330	VW332	VW334
Slot 6	VW384	VW386	VW388	VW390	VW392	VW394	VW396	VW398

Notice

As COTRUST TD2X and SIEMENS TD200 have address fixed at VW0, if you need to use TD2X/TD200 and SM231 8AI Module simultaneously, the SM231 8AI module cannot be installed at Slot 0, or else it cannot operate.

4.6 PID Module Specifications

Table 4-6-1 PID modules order No.

Modules	Order No.
SM231 thermocouple PID module, 4 J/K types, with intelligent PID, isolated 16 bits precision	CTH2 231-7TD32
SM231 thermocouple PID module, 8 J/K types, with intelligent PID, isolated 16 bits precision	CTH2 231-7TF32
SM231, 8 current inputs, 0-20mA/4-20mA, with intelligent PID, isolated 16 bits precision	CTH2 231-7HF32

【Main features】

- Isolation between Bus, Power and all channels, with high reliability and interference immunity.
- 16 bits sampling accuracy, with hardware filter technology to obtain more accurate and stable measuring values.
- Power supply equip with anti-protection and Surge absorption, suited for harsh industrial environment.
- Integrated advanced fuzzy logic control algorithm, accurately control temperature without program and well motion performance.
- PID output, which can be PWM or Analog-Bipolar output, support both heating and cooling control.

【Usage specifications】

- Insulation thermocouple should be used to achieve good immunity from interference
- Use Shielded wire, grounded, as signal line.
- GND terminal must be connect to the ground.
- Short unused channels to eliminate the Break line fault alarm

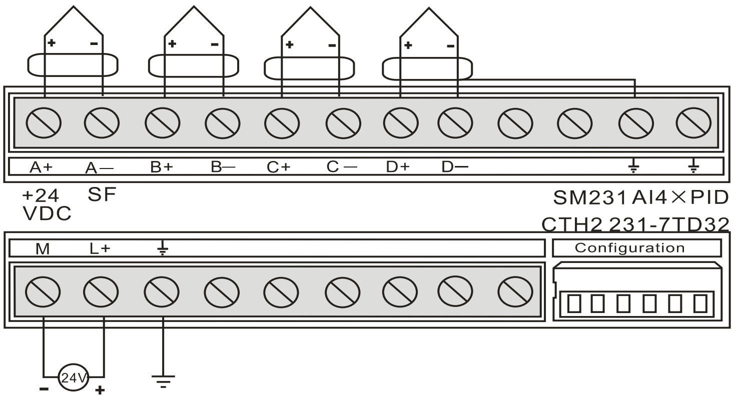
Table 4-6-2 SM231-7TD/7TF thermocouple PID module specifications

Modules	SM231 4AI×TC PID	SM231 8AI×TC PID
Order No.	CTH2 231-7TD32	CTH2 231-7TF32
Features		
Dimension (W×H×D)	71.3 × 96 × 62mm	
LED indicator	24VDC indicator: ON=No fault, OFF=no 24VDC power SF Indicator: ON=Module fault, Flash=Out range or disconnect, OFF=No fault	
Power supply		
+5VDC current	87mA	
L+	34mA	39mA
L+ coil voltage range	20.4~28.8VDC	
Power	1.8W	
Input		
Type	Float TC	
Range	K-type TC	
Counts	4	8
Isolation		
Field to Logic	500VAC	
Field to 24VDC	500VAC	
24VDC to Logic	500VAC	
Common mode input range (input channel to input channel)	120VAC	
Common mode rejection	>120dB@120VAC	
Sampling features		
Temperature resolution	0.1℃/0.1℉	
Voltage resolution	15bits + sign bit	
Transfer principle	Σ-Δ	
Module update time: All channels	405ms	825ms
Wire length to sensor	Max 100m	
Conductor loop resistance	Max 100Ω	
Suppression of interference	85dB@ 50Hz/60Hz/400Hz	
Data word format	Voltage: -27648~+27648	
Input impedance	>1MΩ	
Max. Input voltage	30VDC	

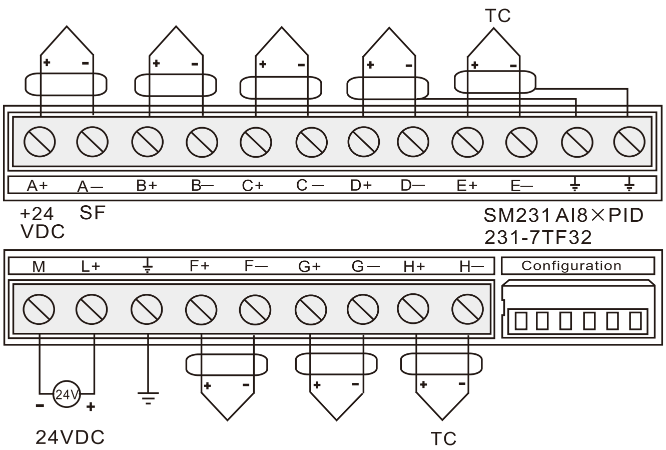
Temperature resolution	15bits + sign bit
Input Filter attenuation	-3dB@ 21kHz
Basic error	0.1% FS(voltage)
Repeatability	0.05% FS
Cold junction error	±1.5℃
Diagnostic program	LED: EXTF, SF
PID features	
PID algorithm	PID+FUZZY parameter adaptive tuning
Sampling time	1s
Min output pulse width	10ms
PID Type	P, PI, PD, PID
PID output type	Analog or PWM control
PID output polarity	Bipolar or Unipolar

Wiring diagram

SM231 thermocouple PID module(CTH2 231-7TD32)



SM231 thermocouple PID module (CTH2 231-7TF32)



PID address and parameter configuration

Table 4-6-3 PID address computation

Address name	Formula	Note
PID parameter address	A=(2048+S*256)+16*C	S = slot No. (0~6) C = channel No. (SM231-7TF: 0~7, SM231-7TD: 0~3)
PID positive impulse output address	X=(2048+S*256)+12	
PID negative impulse	Y=(2048+S*256)+13	

output address		

Table 4-6-4 PID parameter output (Module to CPU)

Items	Address	Value range	Actual value
Actual temperature	VW A	-2000~13000	-200~1300 degrees
Status word	VW A+2		
PID Analog output	VW A+4	-32000~32000	

Table 4-6-5 PID parameter input (CPU to Module)

Items	Address	Value range	Actual value
Setting temperature	VW A+128	-2000~13000	-200~1300 Degree
Control byte		VB A+130 bit =0	VB A+130bit = 1
	V(A+130).0	PID not run, no output	PID run
	V(A+130).1	Integral works all the time, but Kp not self-adaptive	Integral separation and Kp self-adaptive
	V(A+130).2	PID unipolar output, 0~32000	PID bipolar output, -32000~32000, support heating and cooling
	V(A+130).3	Unused	
	V(A+130).4	Integral works	Integral not work
	V(A+130).5	Differential works	Differential not work
	V(A+130).6	Filter the actual temperature value, enhance Interference immunity	Not filter the actual temperature value
PID pulse output period	VW A+132	1~255	1~255s
Kp (proportional coefficient)	VW A+134	0~9999	0~999.9
Ti (integral time)	VW A+136	0~3600	0~3600s
Td (differential time)	VW A+138	0~3600	0~3600s

Table 4-6-6 Output pulse address

Channel number	Output pulse address	
	Positive	Negative
Channel 0	V X.0	V Y.0
Channel 1	V X.1	V Y.1
Channel 2	V X.2	V Y.2
Channel 3	V X.3	V Y.3
Channel 4	V X.4	V Y.4
Channel 5	V X.5	V Y.5
Channel 6	V X.6	V Y.6
Channel 7	V X.7	V Y.7

Application example:

Calculating the address of last PID loop on the second expansion module SM231-7TF.
According to the above statement: S=1, C=7
A Address, A=2048 + 1 * 256 + 16 * 7 = 2416
X Address, X=2048 + 1 * 256 + 12 = 2316
Y Address, Y=2048 + 1 * 256 + 13 = 2317
Set or read out the parameter values according to the following parameter address descriptions:
VW2544 //Set temperature
VB2546 //Control word (parameter self-adaptive, bipolar output)
VW2548 //Pulse output period
VW2550 //Kp
VW2552 //Ti (s)
VW2554 //Td (s)
VW2416 //Actual temperature
VW2418 //Status word
VW2420 //PID analog output
V2316.7 //Positive pulse output
V2317.7 //Negative pulse output

Do not use the V memory occupied by the PID module you use when program other block for sake of PID module normal usage. You can call the specific parameter configuration library of SM231 PID.

Address the module occupied in slot 0: VW2048 to VW2298

Address the module occupied in slot 1: VW2304 to VW2554

Address the module occupied in slot 2: VW2560 to VW2810

Address the module occupied in slot 3: VW2816 to VW3066

Address the module occupied in slot 4: VW3072 to VW3322

Address the module occupied in slot 5: VW3328 to VW3578

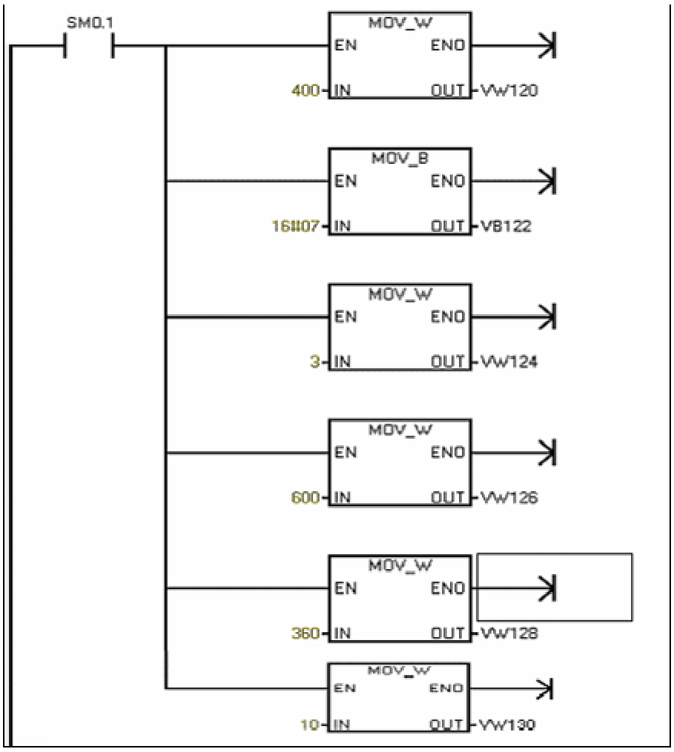
Address the module occupied in slot 6: VW3584 to VW3834

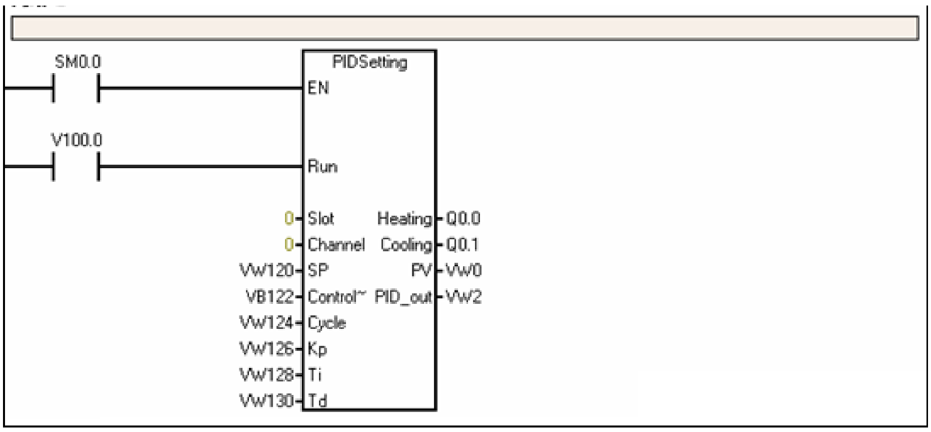
Ladder:

```
NOTE
The parameter address of PID module occupy CPU V storage area.
To ensure PID module normally use, please do not use CPU V storage area the PID
Module occupied when you write other program.
The address module occupied in slot 0: VW2048 to VW2098
The address module occupied in slot 1: VW2304 to VW2554
The address module occupied in slot 2: VW2560 to VW2810
The address module occupied in slot 3: VW2816 to VW3066
The address module occupied in slot 4: VW3072 to VW3322
The address module occupied in slot 5: VW3328 to VW3578
The address module occupied in slot 6: VW3584 to VW3834
```

Network 1

```
NOTE
set parameter of first EM231 PID module loop(channel 0)  of first expansion
module(slot 0):
Assuming that PID parameters need to be changed after debugging of this loop,
PIDSetting is called to set parameters of this loop
No need to calculate PID parameter address, just input the slot and channel number
where input loop is, and enable Sun to run the loop.
This case including network 1 and 2, network 1 for PID parameter initialization, and
network 2 for PID running.
Q0.0 is positive pulse output, Q0.1 is negative pulse output, VW0 is actual
temperature, VW2 is PID analog output.
you can also modify PID setting parameter to use other address, corresponding
address in this case is:
set temperature--VW120, control Word--VB122, pulse output cycle--VW124, proportion
coefficient--VW126
integral time--VW128, differential time--VW130
```





【DIP Configuration】

SM231 TC module, supports J/K-type thermocouple, provide a 6-bit DIP switch to set measurement Unit, Cold-junction compensation and open-circuit fault direction. As with SM231 8TC, it detect open-wire forcely.

6-bit DIP switch is located below the module as shown in the following figure. SW1~SW2 are unused, SW3 for TC type, SW4 for open-wire detect direction, SW5 for measurement Unit, SW6 for Cold-junction enable. Cut-off and recycle the PLC power to make the DIP configuration effective.

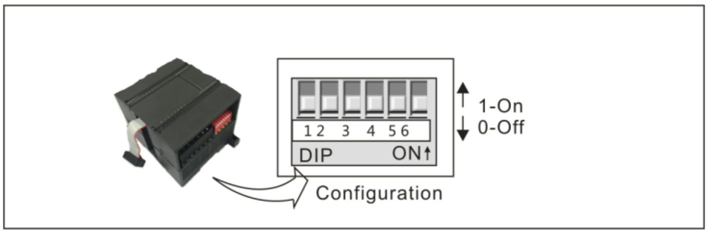


Table 4-6-7 SM231 DIP switch configuration

Items	Switch	Setting
TC type	SW3	0 - J; 1 - K
Open-wire detect direction	SW4	0 - Upscale (+3276.7 degrees); 1 - Downscale (-3276.8 degrees)
Measurement Unit	SW5	0 - Celsius; 1 - Fahrenheit
Cold-junction enable	SW6	0 - Yes; 1 - No

Thermocouple measurement range

Data byte (1 digit=0.1°C) Dec	Hex	Type J	Type K	Type T	Type E	Type R,S	类型N	±80mV	
32767	7FFF	>1200.0°C	>1372.0°C	>400.0°C	>1000.0°C	>1768.0°C	>1300.0°C	>94.071mV	OF
↑	↑							↑	↑
32511	7EFF							97.071mV	OR
:	:								
27649	6C01							80.0029mV	
27648	6C00					↑		80mV	
:	:								
17680	4510		↑			1768.0°C			
:	:								
13720	3598		1372.0°C Overrange				↑		
:	:								
13000	32C8	↑	1300.0°C				1300.0°C		NR
:	:								
12000	2EE0	1200.0°C							
:	:				↑				
10000	2710			↑	1000.0°C				
:	:								
4000	0FA0			-400.0°C		400.0°C			
:	:								
1	0001	0.1°C	0.1°C	0.1°C	0.1°C	0.1°C	0.1°C	0.0029mV	
0	0000	0.0°C	0.0°C	0.0°C	0.0°C	0.0°C	0.0°C	0.0mV	
-1	FFFF	-0.1°C	-0.1°C	-0.1°C	-0.1°C	-0.1°C	-0.1°C	-0.0029mV	
:	:					Underrange			
-500	FE0C					-50.0°C			
-1500	FA24	-150.0°C				↓			
:	:								
-2000	F830	Underrange	-200.0°C						
:	:								
-2100	F7CC	-210.0°C	Underrange						
:	:								
-2550	F60A			-255.0°C	-255.0°C				
:	:			Underrange	Underrange				
-2700	F574	↓	-270.0°C	-270.0°C	-270.0°C		-270.0°C		
:	:								
-27648	9400		↓	↓	↓		↓	-80mV	
-27649	93FF							-80.0029mV	UR
:	:								
-32512	8100							-94.071mV	
#	#							↓	↓
-32768	8000	<-210.0°C	<-270.0°C	<-270.0°C	<-270.0°C	<-50.0°C	<-270.0°C	<-94.07mV	UF
Accuracy over full span		S0.1%	S0.3%	S0.6%	S0.1%	S0.6%	S0.1%	S0.1%	
Accuracy (normal range without cold junction)		S1.5°C	S1.7°C	S1.4°C	S1.3°C	S3.7°C	S1.6°C	S0.10°C	
Cold junction error		S1.5°C	S1.5°C	S1.5°C	S1.5°C	S1.5°C	S1.5°C	NA	

*OF = Overflow; OR = Overrange; NR = Normal range; UR = Underrange; UF = Underflow

↑ indicates that all analog values greater than this and below the open wire threshold report the overflow data value, 32767 (0x7FFF).

↓ indicates that all analog values less than this and greater than the open wire threshold report the underflow data value, --32768 (0x8000).

Table 4-6-8 231-7HF32 current PID module specifications

Features		
Dimension(W x H x D)		120.5 x 80 x 62 mm
Weight		210 g
Insert I / O terminal		No
Power consumption		
Bus(5 V DC)		87 mA
L+		37 mA
Power		1.8 W
Input features		
Type		8, analog
Input range/resistance		Current input
Maximum Permitted input voltage		30 V DC
Isolation	Field-logic	500 V AC
	Logic- 24 VDC	500 V AC
	24 VDC-logic	500 V AC
Update time		825 ms(all channels)
Measurement principle		SIGMA-DELTA
Resolution		15 bit + signal
Temperature		0.1 ℃ / 0.1 ℉
Noise suppression		85 dB
Noise frequency		50 / 60 / 400 Hz
CMV		120 V AC
CMR		120 dB @120 V AC
Basic error		0.1% FS(current)
Repeatability		0.05% FS
Diagnosis program		LED: EXTF, SF
Maximum cable length		100 m —sensor
Max cable resistance		10 Ω
PID features		
PID algorithm		PID+FUZZY parameter self-adjusted
Sampling time		1s
Minimum output pulse		10ms
PID type		P, PI, PD, PID
PID output type		Analog or PWM
PID output polarity		Unipolar or bipolar

SM231 current PID module(CTH2 231-7HF32)

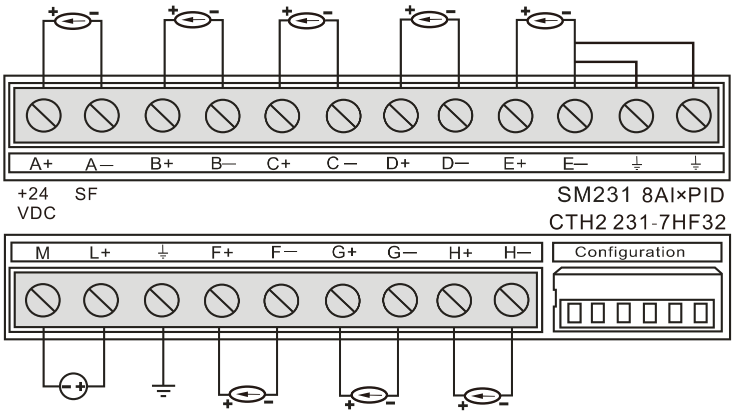


Table 4-6-9 DIP switch configuration

SW1	SW2	SW3	SW4	SW5	SW6	Range	Resolution
ON	ON	OFF	OFF	OFF	OFF	0~20mA	0.1℃/F
ON	ON	ON	OFF	OFF	OFF	4~20mA (default)	0.1℃/F

PID address and parameter configuration

Table 4-6-10 PID address computation

Address name	Formula	Note
PID parameter address	A=(2048+S*256)+16*C	S = Slot No. (0~6) C = Channel No. (SM231-7HF: 0~7)
PID positive pulse output address	X=(2048+S*256)+12	
PID negative pulse output address	Y=(2048+S*256)+13	

Table 4-6-11 PID parameter output (Module to CPU)

Items	Address	Value range	Actual value
Actual temperature	VW A	-2000~13000	-200~1300 degrees (default 1300)
Status word	VW A+2		
PID Analog output	VW A+4	Bipolar: -32000~32000 Unipolar: 0~+32000	

Table 4-6-12 PID parameter input (CPU to Module)

Items	Address	Value range	Actual value
Setting temperature	VW A+128	-2000~13000	-200~1300 Degree
Control byte	VB A+130	VB A+130 bit =0	VB A+130bit = 1
	V(A+130).0	PID not run, no output	PID run
	V(A+130).1	Integral works all the time, but Kp not self-adaptive	integral separation and Kp self-adaptive
	V(A+130).2	PID unipolar output, 0~32000	PID bipolar output, -32000~32000, support heating and cooling

	V(A+130).3	Unused	
	V(A+130).4	Integral works	Integral not work
	V(A+130).5	Differential works	Differential not work
	V(A+130).6	Filter the actual temperature value, enhance Interference immunity	Not filter the actual temperature value
PID pulse output period	VW A+132	1~255	1~255s
Kp (proportional coefficient)	VW A+134	0~9999	0~999.9
Ti (integral time)	VW A+136	0~3600	0~3600s
Td (differential time)	VW A+138	0~3600	0~3600s
Range	VW A+140	0~13000	0~1300°

Table 4-6-13 Output pulse address, formula: X=(2048+S*256)+13; Y=(2048+S*256)+13:

Channel number	**Output pulse address**	
	Positive	**Negative**
Channel 0	V X.0	V Y.0
Channel 1	V X.1	V Y.1
Channel 2	V X.2	V Y.2
Channel 3	V X.3	V Y.3
Channel 4	V X.4	V Y.4
Channel 5	V X.5	V Y.5
Channel 6	V X.6	V Y.6
Channel 7	V X.7	V Y.7

Example:

Calculate last PID loop address of the second expansion module after 231-7HF , Kp=1200, Ti=360, Td=50.set the pulse output cycle =2s, last channel of second module: S=1, C=7	
A address	A=2048+1*256+16*7=2416
X address	X=2048+1*256+12=2316
Y address	y=2048+1*256+13=2317

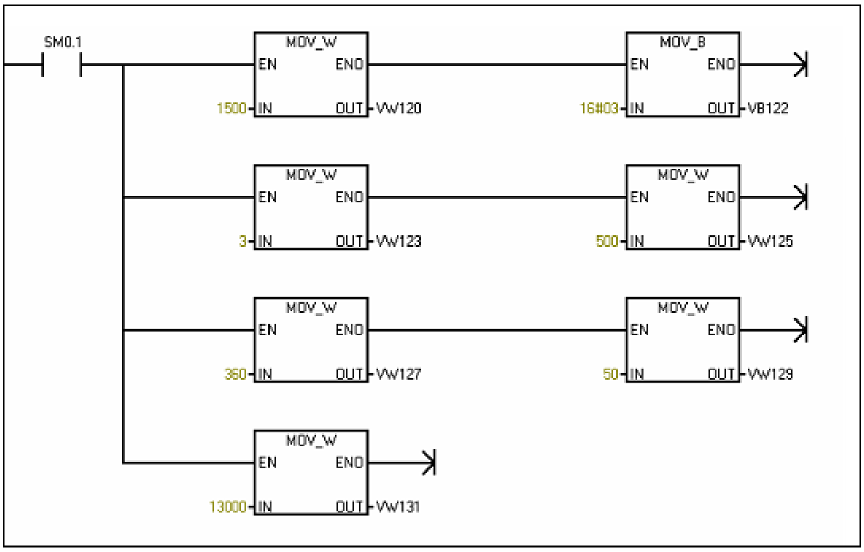
```
DB BLOCK parameters: // S=1, C=7
VW2544 500          // Setting temperature
VB2546 16#07        // Control Word (PID enable, self-adjusted and bipolar output
VW2548 2            // Pulse output cycle
VW2550 1200         // Kp proportional coefficient
VW2552 360          // Ti integral time(s)
VW2554 50           // Td differential time(s)
VW2556 13000        // Range 1300 ℃(default)
```

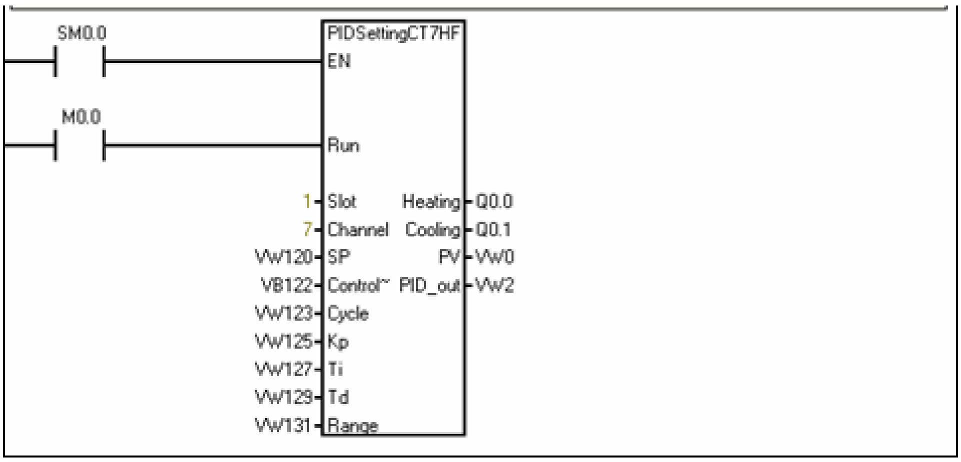
```
VW2416 0            // Actual temperature
VW2418 0            // Status Word
VW2420 0            //PID analog output
```

```
NOTE
The parameter address of PID module occupy CPU V storage area.
To ensure PID module normally use, please do not use CPU V storage area the PID
Module occupied when you write other program.
The address module occupied in slot 0: VW2048 to VW2300
The address module occupied in slot 1: VW2304 to VW2556
The address module occupied in slot 2: VW2560 to VW2812
The address module occupied in slot 3: VW2816 to VW3068
The address module occupied in slot 4: VW3072 to VW3324
The address module occupied in slot 5: VW3328 to VW3580
The address module occupied in slot 6: VW3584 to VW3836
```

Network 1

```
NOTE
set parameter of first EM231-7HF PID module loop(channel 0)  of first expansion
module(slot 0):
Assuming that PID parameters need to be changed after debugging of this loop,
PIDSettingCT7HF is called to set parameters of this loop(you need to add
em231-7hf_pid library first). S
no need to calculate PID parameter address, just input the slot and channel number
where input loop is, and enable Run to run the loop.
this case including network 1 and 2, network 1 for PID parameter initialization, and
network 2 for PID running.
Q0.0 is positive pulse output, Q0.1 is negative pulse output, VW0 is actual
temperature, VW2 is PID analog output.
you can also modify PID setting parameter to use other address, corresponding
address in this case is:
set temperature--VW120, control Word--VB122, pulse output cycle--VW123, proportion
coefficient--VW125
integral time--VW127, differential time--VW129, temperature sensor range--VW131
```





4.7 Communication Module Specifications

Table 4-7-1 Communication modules Order info

Modules	Order No.
SM277A Profibus DP slave interface module, 12M, optoelectronic isolation	CTH2 277-0AA32
SM277B Profibus DP slave module, 1.5M, optoelectronic isolation	CTH2 277-0AB32
SM277C CAN slave module, with 8DI/6DO, optoelectronic isolation, expand to 7 modules	CTH2 277-0AC32

Both SM277A and SM277B adopt photoelectric isolation technology, with high reliability and strong anti-interference ability. These two modules are equipped with terminal resistance, bus connection, no need for special network connector. Its power supply has the function of reverse connection protection and surge absorption, and can be used in harsh working environment.

However, SM277A needs to cooperate with the CPU, while SM277B can be used as a standalone slave station with six extension modules, the scale of slave station I/O up to 128DI/128DO, 48AI/24AQ.

4.7.1 SM277A DP Slave Interface Module Specifications

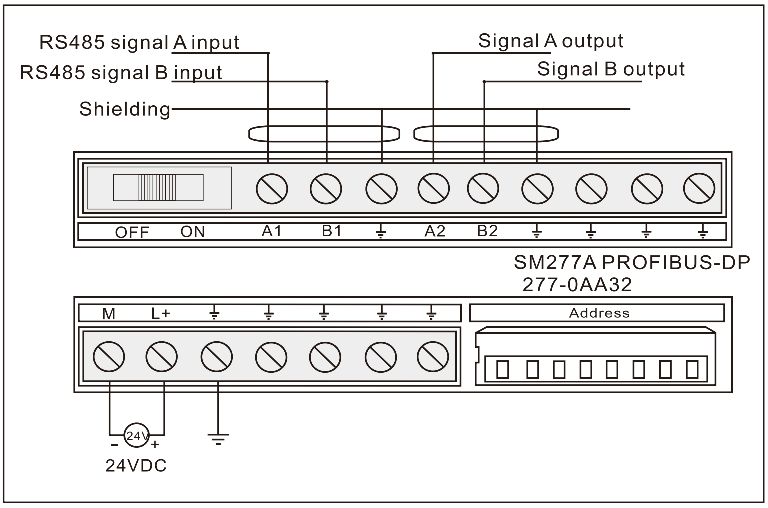
Table 4-7-2 SM277A modules specifications

General				
Order No.	CTH2 277-0AA32			
Dimension (W×H×D)	71.3 × 96 × 62mm			
Port number	1			
Electric interface	Isolated RS485			
Native I/O	0			
Max expansion modules	6			
PROFIBUS-DP/MPI baud rate (auto setting)	9.6/19.2/45.45/93.75/187.5/500Kbps; 1/1.5/3.6/12Mbps			
Protocol	PROFIBUS-DP and MPI			
LED	**OFF**	**Red**	**Flash Red**	**Green**
DX MODE	Not in data exchange mode	--	--	In data exchange mode
DP ERROR	No error	Leave the data exchange mode	parameterize/ configuration error	--
POWER	No 24V power	--	--	24VDC power normal
CPU FAULT	Good health	Internal fault	--	--
Cable length (Max)				
< 93.75Kbps	1200m			
187.5Kbps	1000m			
500Kbps	400m			
1 ~ 1.5Mbps	200m			

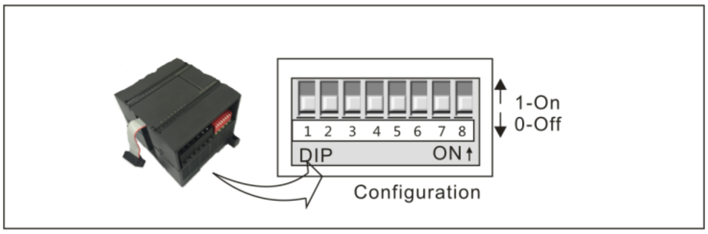
3 ~ 12Mbps	100m
Network performance	
Station address	0 ~ 126 (setting by DIP switch)
Max stations per segment	32
Max stations per network	126, up to 125 SM277 stations
MPI connections	Total 6, 2 reserved (1 used for PG, the other used for OP)
24VDC Power requirement	
Voltage range	20.4 ~ 28.8VDC (class 2 or PLC sensor power)
Power consumption	2.3W
Max current (Module only with port active)	70mA
Ripple noise (＜10Mhz)	＜1V peak to peak (Max)

Wiring diagram

SM277A DP slave interface module (CTH2 277-0AA32)



DIP Address configuration



DIP SW1-8 (in binary), SW1 for the LSB, SW8 for the MSB (must be “OFF”), SW1-SW7 set as “ON” - “1”, “OFF” - “0”, calculated as following:

Address=$SW1\times2^0$+ $SW2\times2^1$ +$SW3\times2^2$ +$SW4\times2^3$+ $SW5\times2^4$ +$SW6\times2^5$ +$SW7\times2^6$

Table 4-7-3 Address switch configuration

Address	SW1~SW8	Address	SW1~SW8	Address	SW1~SW8
0	0000 0000	**43**	1101 0100	**86**	0110 1010
1	1000 0000	**44**	0011 0100	**87**	1110 1010

2	0100 0000	**45**	1011 0100	**88**	0001 1010
3	1100 0000	**46**	0111 0100	**89**	1001 1010
4	0010 0000	**47**	1111 0100	**90**	0101 1010
5	1010 0000	**48**	0000 1100	**91**	1101 1010
6	0110 0000	**49**	1000 1100	**92**	0011 1010
7	1110 0000	**50**	0100 1100	**93**	1011 1010
8	0001 0000	**51**	1100 1100	**94**	0111 1010
9	1001 0000	**52**	0010 1100	**95**	1111 1010
10	0101 0000	**53**	1010 1100	**96**	0000 0110
11	1101 0000	**54**	0110 1100	**97**	1000 0110
12	0011 0000	**55**	1110 1100	**98**	0100 0110
13	1011 0000	**56**	0001 1100	**99**	1100 0110
14	0111 0000	**57**	1001 1100	**100**	0010 0110
15	1111 0000	**58**	0101 1100	**101**	1010 0110
16	0000 1000	**59**	1101 1100	**102**	0110 0110
17	1000 1000	**60**	0011 1100	**103**	1110 0110
18	0100 1000	**61**	1011 1100	**104**	0001 0110
19	1100 1000	**62**	0111 1100	**105**	1001 0110
20	0010 1000	**63**	1111 1100	**106**	0101 0110
21	1010 1000	**64**	0000 0010	**107**	1101 0110
22	0110 1000	**65**	1000 0010	**108**	0011 0110
23	1110 1000	**66**	0100 0010	**109**	1011 0110
24	0001 1000	**67**	1100 0010	**110**	0111 0110
25	1001 1000	**68**	0010 0010	**111**	1111 0110
26	0101 1000	**69**	1010 0010	**112**	0000 1110
27	1101 1000	**70**	0110 0010	**113**	1000 1110
28	0011 1000	**71**	1110 0010	**114**	0100 1110
29	1011 1000	**72**	0001 0010	**115**	1100 1110
30	0111 1000	**73**	1001 0010	**116**	0010 1110
31	1111 1000	**74**	0101 0010	**117**	1010 1110
32	0000 0100	**75**	1101 0010	**118**	0110 1110
33	1000 0100	**76**	0011 0010	**119**	1110 1110
34	0100 0100	**77**	1011 0010	**120**	0001 1110
35	1100 0100	**78**	0111 0010	**121**	1001 1110
36	0010 0100	**79**	1111 0010	**122**	0101 1110
37	1010 0100	**80**	0000 1010	**123**	1101 1110
38	0110 0100	**81**	1000 1010	**124**	0011 1110
39	1110 0100	**82**	0100 1010	**125**	1011 1110
40	0001 0100	**83**	1100 1010	**126**	0111 1110
41	1001 0100	**84**	0010 1010		
42	0101 0100	**85**	1010 1010		

Table 4-7-4 Terminal resistance setting

Terminal resistance	Meaning
ON	Located on the last node of network
OFF	Not Located on the last node of network

4.7.2 SM277B DP Slave Module Specifications

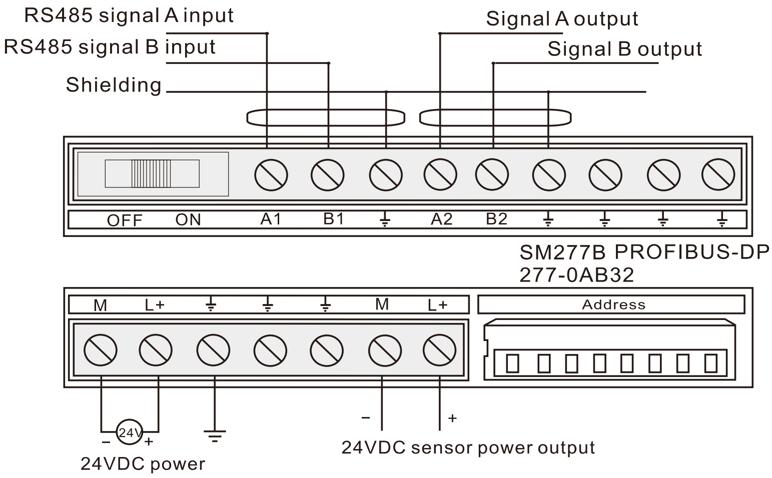
Table 4-7-5 SM277B modules specifications

General	
Order No.	CTH2 277-0AB32
Dimension (W×H×D)	71.3 × 96 × 62mm
Port number	1
Electrical interface	Isolated RS485 (500VAC)
PROFIBUS-DP baud rate (auto-set)	9.6/19.2/45.45/93.75/187.5/500Kbps; 1/1.5Mbps
Protocol	PROFIBUS-DP V0 slave device
Native I/O	None
Extended I/O	
Max extended modules permitted	6
Max I/Os (Image)	256(128 In/128 Out)
LED indicator	
ON(Green)	Light on after power up
SF(Red)	Light on when SM277B found error during serf-check (including address outrange) or expansion I/O module fault.
BF(Red)	Flash when no exchange with Master.
Cable length (max)	
< 93.75Kbps	1200m
187.5Kbps	1000m
500Kbps	400m
1~1.5Mbps	200m
Network performance	
Station address	0~126(set by DIP switch)
Max stations per segment	32
Max stations per network	126, up to 125 SM277B stations
24VDC power requirement	
Voltage range	20.4~28.8VDC (class 2 or PLC sensor power)
Isolation	None
Power consumption	8W
Output current for sensor (24VDC)	
Voltage range	20.4~28.8VDC (class 2 or PLC sensor power)
Max output current at 24V	400mA
Output current for expansion module (5VDC)	

Max output current at 5V	660mA

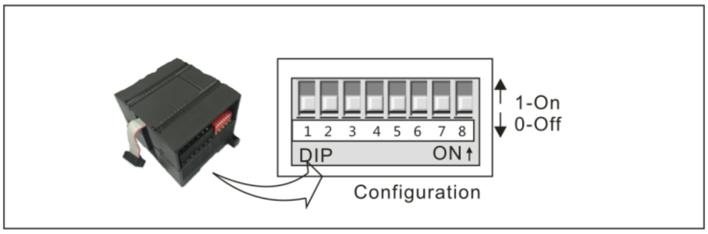
Wiring diagram

SM277B PROFIBUS DP slave module (CTH2 277-0AB32)



Set PROFIBUS address

You can use DIP switch to set PROFIBUS address. Cut-off and recycle the power of SM277B module to make them effective after you change the address.



DIP switch SW1-8 (in binary), SW1 for the LSB, SW8 for the MSB (must be “OFF”), SW1-SW7 set as “ON” - “1”, “OFF” - “0”, calculated as following:

Address $=SW1\times2^0+SW2\times2^1+SW3\times2^2+SW4\times2^3+SW5\times2^4+SW6\times2^5+SW7\times2^6$

Table 4-7-6 DIP Address configuration

Address	SW1~SW8	Address	SW1~SW8	Address	SW1~SW8
0	0000 0000	**43**	1101 0100	**86**	0110 1010
1	1000 0000	**44**	0011 0100	**87**	1110 1010
2	0100 0000	**45**	1011 0100	**88**	0001 1010
3	1100 0000	**46**	0111 0100	**89**	1001 1010
4	0010 0000	**47**	1111 0100	**90**	0101 1010
5	1010 0000	**48**	0000 1100	**91**	1101 1010
6	0110 0000	**49**	1000 1100	**92**	0011 1010
7	1110 0000	**50**	0100 1100	**93**	1011 1010
8	0001 0000	**51**	1100 1100	**94**	0111 1010
9	1001 0000	**52**	0010 1100	**95**	1111 1010
10	0101 0000	**53**	1010 1100	**96**	0000 0110
11	1101 0000	**54**	0110 1100	**97**	1000 0110
12	0011 0000	**55**	1110 1100	**98**	0100 0110

13	1011 0000	**56**	0001 1100	**99**	1100 0110
14	0111 0000	**57**	1001 1100	**100**	0010 0110
15	1111 0000	**58**	0101 1100	**101**	1010 0110
16	0000 1000	**59**	1101 1100	**102**	0110 0110
17	1000 1000	**60**	0011 1100	**103**	1110 0110
18	0100 1000	**61**	1011 1100	**104**	0001 0110
19	1100 1000	**62**	0111 1100	**105**	1001 0110
20	0010 1000	**63**	1111 1100	**106**	0101 0110
21	1010 1000	**64**	0000 0010	**107**	1101 0110
22	0110 1000	**65**	1000 0010	**108**	0011 0110
23	1110 1000	**66**	0100 0010	**109**	1011 0110
24	0001 1000	**67**	1100 0010	**110**	0111 0110
25	1001 1000	**68**	0010 0010	**111**	1111 0110
26	0101 1000	**69**	1010 0010	**112**	0000 1110
27	1101 1000	**70**	0110 0010	**113**	1000 1110
28	0011 1000	**71**	1110 0010	**114**	0100 1110
29	1011 1000	**72**	0001 0010	**115**	1100 1110
30	0111 1000	**73**	1001 0010	**116**	0010 1110
31	1111 1000	**74**	0101 0010	**117**	1010 1110
32	0000 0100	**75**	1101 0010	**118**	0110 1110
33	1000 0100	**76**	0011 0010	**119**	1110 1110
34	0100 0100	**77**	1011 0010	**120**	0001 1110
35	1100 0100	**78**	0111 0010	**121**	1001 1110
36	0010 0100	**79**	1111 0010	**122**	0101 1110
37	1010 0100	**80**	0000 1010	**123**	1101 1110
38	0110 0100	**81**	1000 1010	**124**	0011 1110
39	1110 0100	**82**	0100 1010	**125**	1011 1110
40	0001 0100	**83**	1100 1010	**126**	0111 1110
41	1001 0100	**84**	0010 1010		
42	0101 0100	**85**	1010 1010		

Table 4-7-7 Terminal resistance

Terminal resistance	Meaning
ON	Located on the last node of network
OFF	Not Located on the last node of network

4.7.3 SM277C CANopen Slave Module Specifications

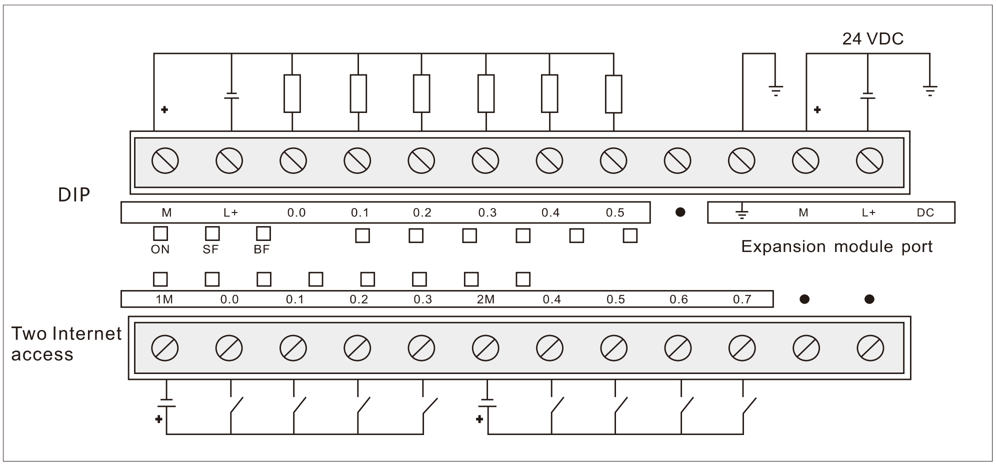
CANopen slave interface module implements optoelectronic isolation with high reliability and high interference immunity. Two RJ45 CANopen communication ports can be used with CAN master system to compromise a distributed control system. Up to 7 CTH200 modules can be expanded by using this module (PID module is not supported), build in 8DI/6DO of transistor type and a 12-pin address switch for selecting station address, baud rate and terminal resistor.

Table 4-7-8 SM277C module specifications

General		
Order No.		CTH2 277-0AC32
Dimension (W×H×D)		71.3 × 96 × 62 mm
Digital input		
Ports		2 RJ45 port for CANopen
Rated input voltage		24VDC
Type		Source/Drain (IEC class 1 source point)
Max continuous voltage		30VDC
Voltage surge		35VDC for 0.5s
Input delay		6.4ms (min. 6.3ms)
Isolation (Field to Logic)		Yes
Output on simultaneously		8 (all at 55℃)
Cable length	Shield	Max 500m
	Unshielded	Max 300m
Digital Output		
Type		Solid-MOSFET (Source)
Rated output voltage		24VDC
output voltage range		20.4~28.8VDC
Output current (max at logic 1)		0.5A
Output groups		1
Outputs per common		6
Output on simultaneously		6
Maximum current for single group		4.5A
Surge current		8A for 100ms
Isolation		Optical couple isolation, 500VAC for 1 minute
Output delay	OFF-ON	Max 15μs
	ON-OFF	Max 130μs
Cable length	Shield	Max 500m
	Unshielded	Max 300m

Wiring diagram

SM277C CANopen slave module (CTH2 277-0AC32)



DIP switch configuration

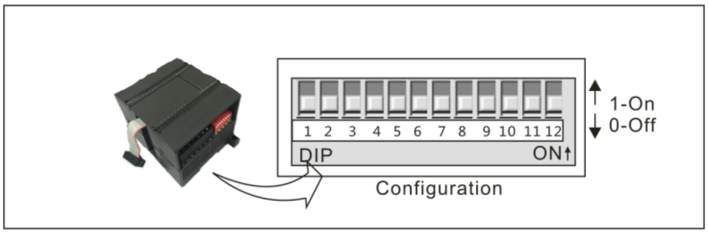
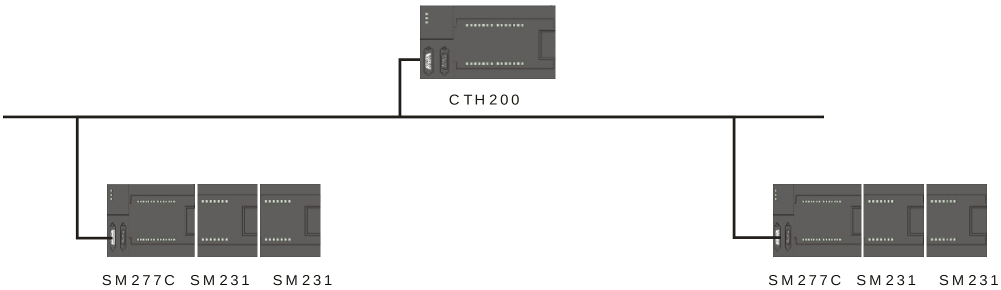


Table 4-7-9 DIP switch configuration of SM277-0AC32

<table>
<tr><th colspan="9">SM277-0AC32 DIP switch</th></tr>
<tr><th colspan="2">Switch</th><th colspan="8">Usage</th></tr>
<tr><td>SW1</td><td rowspan="7">Node address</td><td colspan="8" rowspan="7">In binary: 1 - lowest bit, 7 - highest bit
Note: 0 - global address, 0 is not allowed during operation.</td></tr>
<tr><td>SW2</td></tr>
<tr><td>SW3</td></tr>
<tr><td>SW4</td></tr>
<tr><td>SW5</td></tr>
<tr><td>SW6</td></tr>
<tr><td>SW7</td></tr>
<tr><td>SW8</td><td rowspan="4">Baud rate</td><td colspan="8">In binary: 8 - lowest bit, 10 - highest bit</td></tr>
<tr><td>SW9</td><td>DIP setting</td><td>111</td><td>110</td><td>101</td><td>100</td><td>011</td><td>010</td><td>001</td></tr>
<tr><td rowspan="2">SW10</td><td>Baud rate (Kbps)</td><td>1000</td><td>800</td><td>500</td><td>250</td><td>125</td><td>50</td><td>20</td></tr>
<tr><td>Max length (m)</td><td>25</td><td>50</td><td>100</td><td>250</td><td>500</td><td>1000</td><td>2500</td></tr>
<tr><td>SW11</td><td>Terminal resistor</td><td colspan="8">“ON” for device at network interface, “OFF” for the other.</td></tr>
<tr><td>SW12</td><td>reserved</td><td colspan="8">—</td></tr>
</table>

Network Architecture



Hardware Diagnostic

SM 277C status LED can be used to diagnose the CANopen master-slave network.
Switch on the power of SM 277C, “ON” LED (Green) for the power would light on. If the BF and SF LED remain OFF, indicate the SM 277C works normal, on or flash mean fault in hardware configuration or wiring. If the main system running well, see table below for the reason for error:

Table 4-7-10 Function of SM 277C LED lights

LED	ON	OFF	FLASH	Note
ON	Normal power	No power	---	Power indicator
SF	Expansion I/O module fault	Expansion I/O module no fault	---	System Fault
BF	CAN network not detected	CAN network detected	Inconsistent configuration	Bus Fault

4.7.4 SM277-PN Slave Module Specifications

Table 4-7-11 Specifications of SM277-PN module

Features	
Dimension(W×H×D)	71×80×62 mm
Power consumption	
Rated input voltage	24V DC
Input voltage range	20.4V~28.8V DC
Input current	0.8A
Power voltage of bus	+5V DC
Power current of bus	1.1A
Isolation	External power is isolated from system power
Power protection	Reverse-connection protection and surge absorption
LED lights features	
24V power light(green)	ON = 24VDC power supply normal
	OFF = No 24VDC power supply
SF indicator light(red)	ON = Expansion I/O bus or PROFINET module malfunction
	OFF = No error
BF indicator light(red)	ON = PROFINET bus communication error (not connect with switch, no Internet detected)
	Blinking = different configuration
	OFF = No error
MT indicator light(yellow) (maintenance)	ON = Synchronization lost
	OFF = No error
P1R/P2R port indicator light (green)	ON = With connection to switch /PN master station
	OFF = No connection to switch /PN master bus
P1R/P2R port indicator light(yellow)	ON = With data transceiver to /PN master bus
	OFF = No data transceiver to switch /PN master bus

Table 4-7-12 Specifications of SMS277-PN module

Expansion I/O specifications	
Max modules numbers each slave bus support	8(digital modules, analog modules, temperature modules and PID control modules, except other type modules)

Protocol type	CTH200 PLC self-defined 4MHZ bus protocol
Max IO configuration each slave bus	Analog IO max configuration is 64AI/32AQ Digital IO max configuration is 256DI/256DQ
Topological structure	Support stellate, tree form, linetype and annular structure
PROFINET communication port	
Communication port	One double RJ45 port
Data transmission speed	Transmission speed of Ethernet is 10Mbps
	PROFINET transmission speed is 100Mbps, full duplex
Ethernet	Ping, arp, network diagnosis (SNMP)/MIB-2, LLDP
Cycle of transmission	250us~4ms
Third-party PROFINET master station	S7-300/400, SMART CPU and S7-1200/1500
Max communication distance between slave station	100m(100BASE-TX)
Topological structure	Support stellate, tree form, linetype and annular structure
Isolation	Communication port isolated
Hardware configuration	
File to be imported	PROFINET GSD file in XML format
PN slave station	Support expansion of 8 slot after adding CTH2 277-0PN32 modules
	Digital modules, analog modules, temperature modules and PID control modules, cannot be other types modules

4.8 Weighing Module Specifications

Table 4-8-1 Specifications of SM231 weighing module

Order No.	CTH2 231-7WA32
LED	Power indicator, ON＝24VDC power supply normal, OFF=no 24VDC power supply SF indicator, flash = overrange or line broken, OFF＝ no error
Power supply	
+5VDC consumption	＜140mA
L+	＜100mA
L+ coil voltage range	20.4~28.8VDC
Power dissipation	5W
Input features	
Type	Strain gage, with 4-wire or 6-wire
Range	0~1mV/V 0~2mV/V 0~4mV/V
Number	One-channel weighing sensor
Weighing sensor resistor	Without Explosion-proof interface: 40 Ω< R < 4010 Ω With Explosion-proof interface: 87 Ω< R < 4010 Ω

Isolation	
Field to Logic	500VAC
Field to 24VDC	500VAC
24V to Logic	500VAC
Common mode rejection	>120dB@120VAC
Resolution performance	
Temperature effect on zero	≤±0.1μV/K
Measurement principle	Σ-Δ
Cable length to senor	Max. 500 m
Noise rejection	85db@50Hz/60Hz
data word format	Voltage: 0~65535
Input resolution	16 bits
Basic error	0.05％Fs
Linear error	0.01％Fs

Terminal connection

Weighing module is a measurement module with high accuracy, which can be used for low level signal down to 1.5μV reliably. To acquire operation with no fault, it’s essential to assemble and wire cable correctly.

Following these rules to wire the weighing sensor:

1) If more than 1 sensor would be connected (in parallel), use a terminal box. If distance from the senor to module is larger than to terminal box, a specified expansion box.

2) Cable Shield should always connect to the Cable gland of terminal box or expansion box.

3) Twisted Pair wire should be used and shielded:

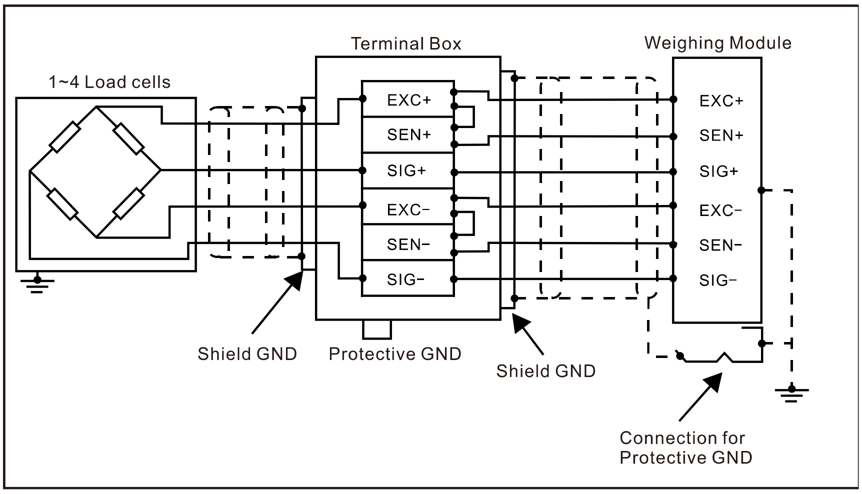
Sensor wire SEN+/SEN-

Voltage measurement wire SIG+/SIG-

Power voltage wire EXC+/EXC-.

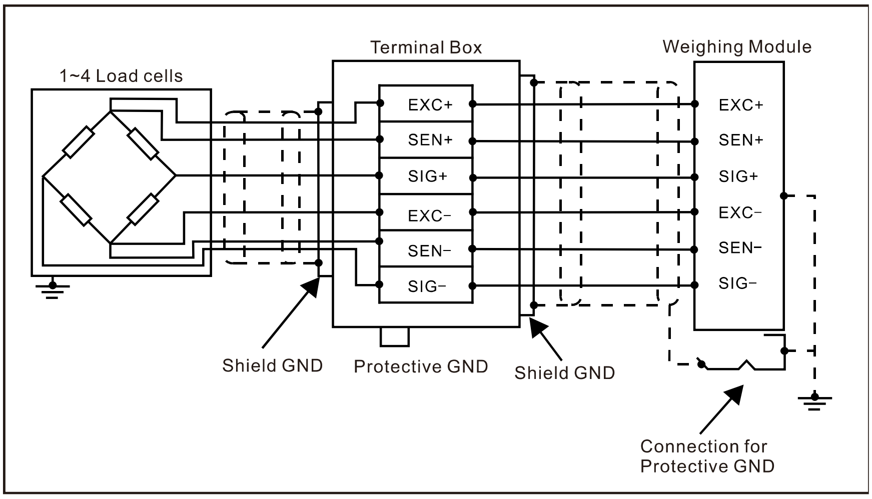
4) The shield must be connected to the shielded joint clamp.

1. 4-wire connection for the weighing sensor



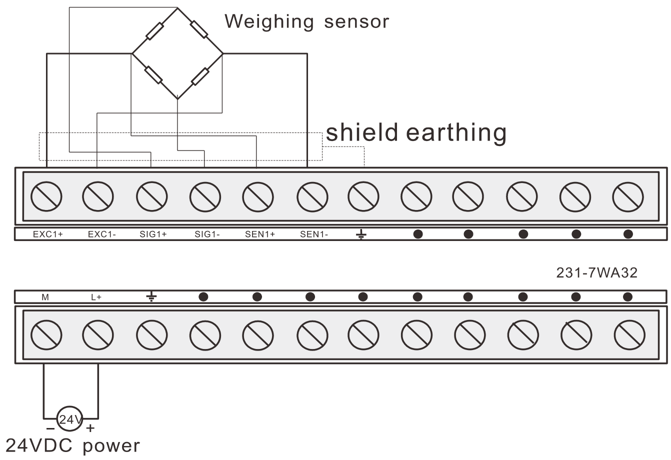
When connecting 4-wire sensor, you must terminal the EXC+ with SENS+, EXC- with SENS-, or else the module cannot work normally.

2. 6-wire connection for the weighing sensor



Wiring Diagram

SM231 weighing module (CTH2 231-7WA32)



4.9 Motion Control Module Specifications

Table 4-9-1 Specifications of motion control module

Order number	**CTH2 253-1BH32**
General	**24VDC input**
Input counts	8
Type	Drain/Source (IEC class1/Drain)
Rated voltage	24 VDC at typical 5mA
Maximum Continuous permissible voltage	30 VDC
Surge voltage	35 VDC for 0.5 s
Logic 1(Min)	15.6 VDC(I0.0, I0.1, I0.2, I0.4, I0.5, I0.6) at 2.72mA 12.8VDC(I0.3, I0.7) at 2.55mA
Logic 0(Max)	15.4VDC(I0.0, I0.1, I0.2, I0.4, I0.5, I0.6) at 2.69mA 12.6 VDC(I0.3, I0.7) at 2.51mA
Input delay	< 1.1us(I0.0, I0.1, I0.2, I0.4, I0.5, I0.6)

		< 1ms(I0.3, I0.7)
Connection of 2 Wire Proximity Sensor (Bero) Permissible leakage current		1mA
Isolation (Field to Logic) Optical Isolation (Galvanic)		Yes 500 VAC for 1 minute
HS input rate HSC Logic 1=16~26 VDC		200KHz(single/dual phase) (I0.0, I0.4) 200KHz (A/B phase) (I0.0 & I0.1, I0.4 & I0.5) (Input Wave shape ratio 40%~60%)
Number of inputs on simultaneously		All
Cable length	Shield	500m for standard input, 50m for HSC (using shielded twisted pair for HSC and ground the shield)
	Unshielded	300m for standard input
HSC max commutating frequency		50KHz
General		**Transistor output**
Output counts		8
Type		Solid – MOSFET(drain)
Rated voltage		24 VDC
Output voltage range		5~28.8 VDC
Surge current (max)		8A for 100ms
Logic 1(max)		0.5V
Logic 0(min)		VCC-0.5V
Rated current per point (max)		0.5A
Rated current per common (max)		2.0A
Leakage current (max)		10μA
lighting load (max)		3.5W
Sensing clamp voltage		L+ - 48 VDC, 1W power
On resistance		Typical 0.3 Ω (max 0.6Ω)
Isolation Optical Isolation (Galvanic) Isolation groups		500 VAC, 1 minute Refer to the wiring diagrams below
Delay (max) Off to On On to Off		0.2μs(Q0.0, Q0.1, Q0.2, Q0.3), 50μs(Q0.4, Q0.5, Q0.6, Q0.7) 0.2μs(Q0.0, Q0.1, Q0.2, Q0.3), 130μs(Q0.4, Q0.5, Q0.6, Q0.7)
Pulse frequency (max)		200KHz(Q0.0, Q0.2)
Outputs on simultaneously		All at 55°C
Two outputs parallel		Yes, only for outputs in the same group
Cable length	Shield	500m

	Unshielded	150m

Caution
DO commands executing would be delayed a while during communication transfer; if there is only one SM253 motion control module attach after the CPU, the delay time for DO from enable to execute is about 780μs; if there are 5 motion control modules, the delay time would be 930μs.

SM253 modules are used for motion control, with 2 independent HSC MC253_HSC0 and MC253_HSC1 (200KHZ for single/dual phases); 2 channels of independent 200KHz HSP output which support instructions like MC253_PTP/SPEED_CTL/PWM in the motion_ctrl_module_lib. Refer to the Appendix *J SM253 Motion Control Library* for more details about the motion control library.

Wiring Diagrams

SM253 position control module (CTH2 253-1BH32)

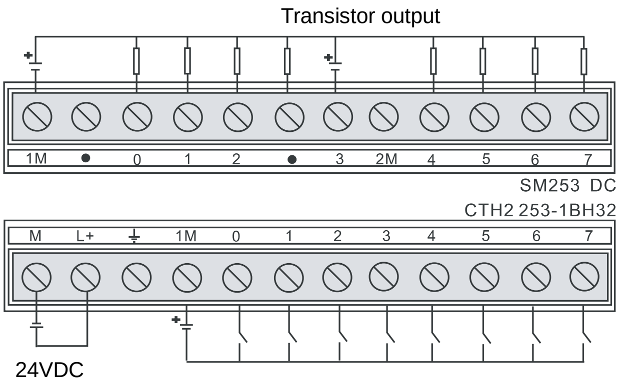


Table 4-9-2 I/O definition

Input	Definition	Output	Definition
I0.0	Pulse input for MC253_HSC0	Q0.0	Pulse output for axis 0
I0.1	External direction signal for MC253_HSC0	Q0.1	Direction signal for axis 0
I0.2	Reset signal for MC253_HSC0	Q0.2	Pulse output for axis 1
I0.3	Emergency stop for motion axis 0	Q0.3	Direction signal for axis 1
I0.4	Pulse input for MC253_HSC1	Q0.4	Normal outputs
I0.5	External direction signal for MC253_HSC1	Q0.5	
I0.6	Reset signal for MC253_HSC1	Q0.6	
I0.7	Emergency stop for motion axis 1	Q0.7	

4.10 BD Expansion Board Specifications and Installation

Table 4-10-1 Order No. of BD expansion board

Description	Order No.
EBH AMS-03 Analog I/O Expansion Board, 2*12bits voltage	CTH2 AMS-03S1-EB

resolution, 1*12bits voltage/current output resolution	
EBH-AMS-06 Analog I/O Expansion Board,4*12bits voltage resolution, 2*12bits voltage output resolution	CTH2 AMS-06S1-EB
EBH CAN master communication Expansion Board, 1Mbps, optoelectronic isolation	CTH2 CAN-01S1-EB

4.10.1 Analog Expansion Board Specifications

Analog Expansion Board EBH-AMS-03/EBH-AMS-06 can be used with CTH200 modules by invoking in the CTH200_CPU expansion slot.

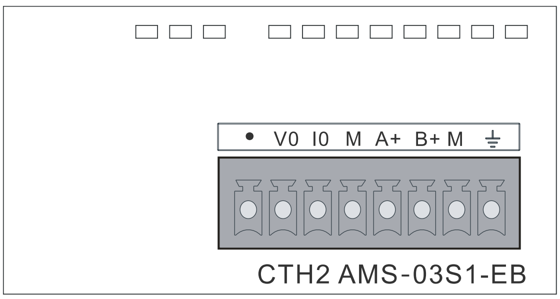
Table 4-10-2 Analog Expansion Board I/O specifications

Items	Specifications		
	EBH-AMS-03	**EBH-AMS-06S1**	**EBH-AMS-06S2**
Features			
Dimension (W×H×D)	24×14.4×68.2mm		
Voltage supply	+5VDC		
Current	60mA		200mA
LED			
Power LED	Health status of power: ON - 5VDC normal, OFF - no power		
SF LED	Calibration status, FLASH - calibration failure		
Analog input			
Power supply	5VDC, output current < 200mA		
Counts	2	4	
Type	Single-end voltage input		
Range	±10V		
Data word format for full-scale	-32000 to +32000		
DC input impedance	>100KΩ		
Max input voltage	30VDC		
Resolution	11 bits + 1 sigh bit		
Isolation	None		
Accuracy			
Worst case (0-55 ˚)	±2.5% full scale		
Typical (25 ˚)	±1.0% full scale		
Repeatability	±0.05% full scale		
Conversion time from analog to digital	125ms		
Transform principle	Σ-Δ		
Step response	Max 250ms		
Noise suppression	Typical -20dB@50Hz		
Analog output			
Counts	1	2	
Signal range	Voltage 0-10V		--
	Current 0-20mA	--	Current 0-20mA

Data word format for full-scale	0 to +32000		
Resolution	12bits		
LSB value	Voltage 5.00μA	--	Voltage 5.00μA
	Current 2.50mV		--
Isolation	None		
Accuracy			
Worst case (0-55 ˚)	Voltage ±3% full-scale		--
	Current ±2% full scale	--	Current ±2% full scale
Typical (25 ˚)	Voltage ±1% full scale		--
	Current ±1% full scale	--	Current ±1% full scale
Settling time ● Resistive load ● Capacitive load ● Inductive load	0.1ms		 0.2 ms 3.3 ms 0.5 ms(1mH)
Max output drive	Voltage ≥5000Ω		Current ≤500Ω

Terminal definition

1. Schematic and terminal diagrams for AMS-03 are shown in the following figures. Table following give the definition of each terminal.



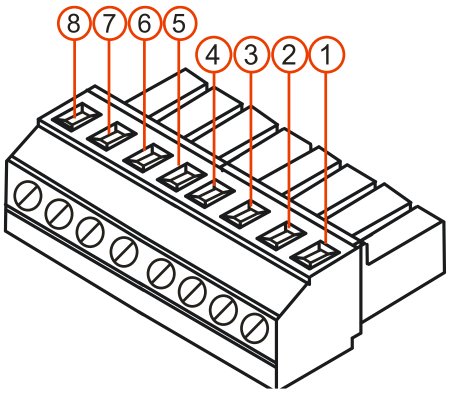
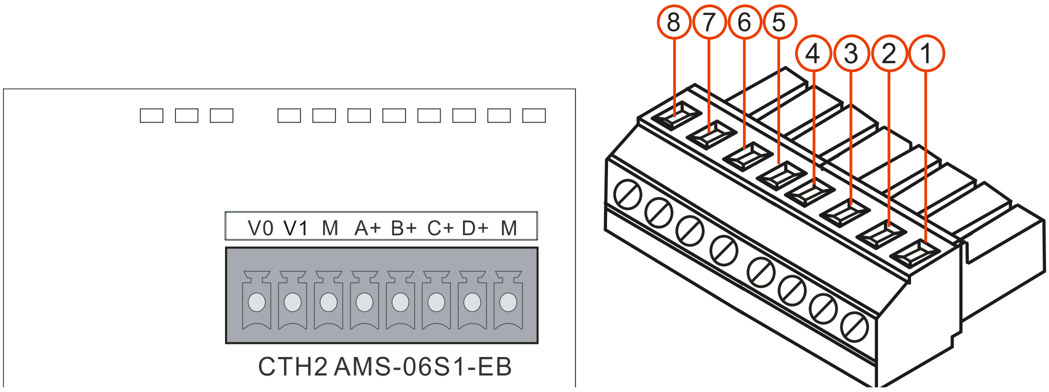


Table 4-10-3 AMS-03 Terminal definition

Terminal	Signal definition	Terminal	Signal definition
1	Float	5	Analog Input A+
2	Voltage output VO	6	Analog Input B+
3	Current output IO	7	Ground GND-M
4	Ground GND-M	8	Earth

2. Schematic and terminal diagrams for AMS-06 are shown in the following figures



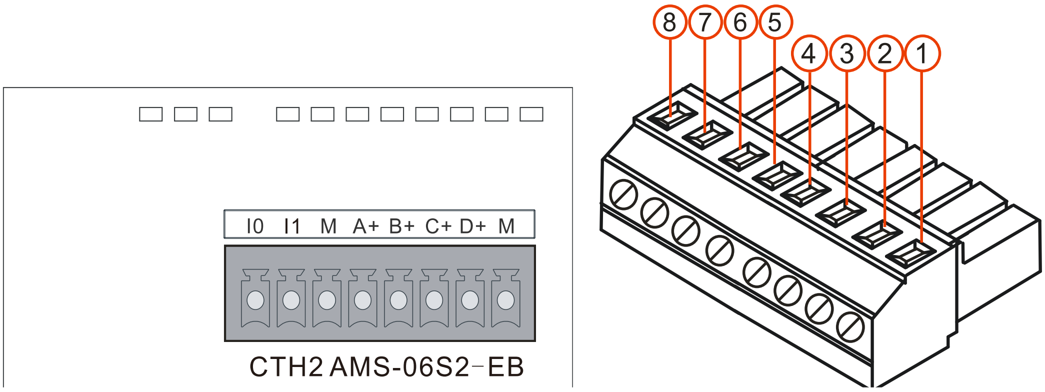


Table 4-10-4 AMS-06 Terminal definition

Terminal	Signal definition	Terminal	Signal definition
1	Output V0/I0	5	Analog Input B+
2	Output V1/I1	6	Analog Input C+
3	Ground GND-M	7	Analog Input D+
4	Analog Input A+	8	Ground GND-M

<**Note**> A+/B+/C+/D+ are voltage inputs for positive or negative voltage, V1 must be connected with Terminal 3, D+ connected with Terminal 8. M is the common Ground, any output connecting directly with M is not allowed, or it will cause output short.

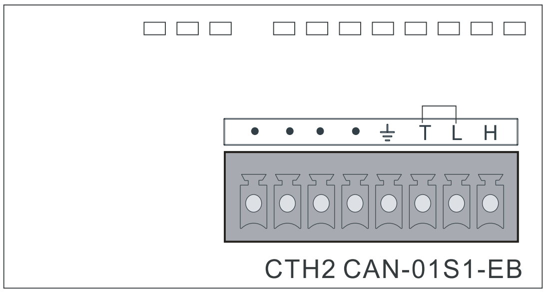
4.10.2 CAN Expansion Board Specifications

Table 4-10-5 Specifications of CAN-01 expansion board

Basic features							
Dimension (W×H×D)	137 x 96 x 62mm						
Suitable product	CPU H224X/H226XL/H228XL						
Power dissipation	9W						
CANopen communication							
Communication interface	1 (8 Pin)						
Transmission rate (kbps)	1000	800	500	250	125	50	20
Max length (m)	25	50	100	250	500	1000	2500
Max station address	127						
Station address range	1-127						
Max number of slaves	32						

Max Digital access		No. of Bytes	start address in memory
	Input	64	IB16
	Output	64	QB16
Max Analog access		No. of channels	start address in memory
	Input	8	AIW64
	Output	8	AQW64

Schematic and terminal diagrams of CAN-01 are shown in the following figures.



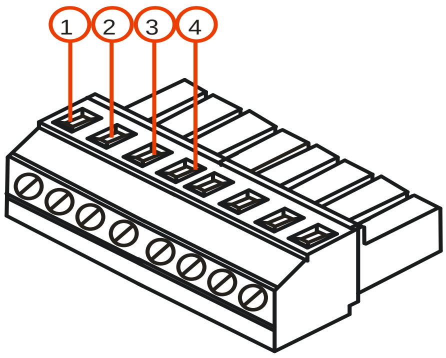


Table 4-10-6 Terminal definition

Terminal	Signal definition
1	CAN_H
2	CAN_L
3	Terminal 2 with 3 when matching Termination resistor
4	Ground

4.10.3 Circuit Guidance Expansion Board Specifications

Circuit guidance expansion board, one of the CTH200 series PLC expansion boards, is used to detect the connection and charging status between charging piles and charging gun or car, which can be used to control the charging speed. Suitable both for AC and DC charging.

Table 4-10-7 Specifications of circuit guidance expansion board

Order No.	CTH2 PWM-04S1-EB
Features	
Dimension(W×H×D)	28×14.4×68.2 mm
Power	
Bus power voltage	+5V DC
Bus power current	< 200mA
LED	
Power indicator	ON = 5VDC power supply normal, OFF = no power supply
SF indicator	Flashing =calibration fail, flash one time= successfully calibration
CC indicator	ON = connected, OFF = disconnected
Function	
Expansion	With bus expansion function
Output protection	With TVS protection

Filter		Combined hardware filter with software one
Power supply		With 5V DC power supply for modules
Input features		
Type		Single-ended input(Unipolar/bipolar)
Counts		2
Full-scale voltage		0 ~ +12V
Max input voltage		+30V DC
Input resistance		≥100K
Data format		0 ~ +32000
Input step response		Max. 200ms
DAC time		50ms
Isolation		No
Resolution		12 bits
Noise suppression		Typical: -20dB@50HZ
Accuracy	Worst	±2% full-range(0℃-55℃)
	Typical	±1% full-range(25℃)
Repeatability		±0.05% full-range
Output features		
Type		Unipolar/bipolar
Counts		2
Voltage output		+ 12V ± 5%
PWM output		±12V ± 5%
PWM parameter		
PWM range data format		0 ~ 1000
Full-range		0(-12V); 1000(+12V)
Output duty cycle tolerance		±0.5% full-range
Frequency range		1000Hz ± 3%
Voltage output time		< 3us
Rise time Tr		< 2us
Fall time Tf		< 2us
Max output drive		≥1000**Ω**
+12V enable output current		12mA(RL=0**Ω**)
		6mA(RL=1K**Ω**)
		4mA(RL=2K**Ω**)

Terminal definition

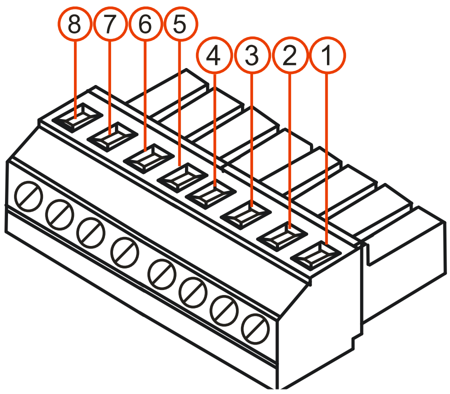


Table 4-10-8 terminal definition

Bit	Definition	Bit	Definition
1	NC	5	CP
2	NC	6	PWM
3	NC	7	+12V
4	CC	8	PE

Diagram

CTH2 PWM-04S1-EB expansion board suits for AC and DC charging guidance

1) Diagram below shows the function of AC charging guidance:

+12V as the output port (1K resistance connected);

PWM output ±12V, pulse changeable PWM;

CC detect the status of cable connection (0X12--disconnected, 0X00--connected);

CP detect PWM output and connect +12V in spare time via switch S1;

PE refer to the earth

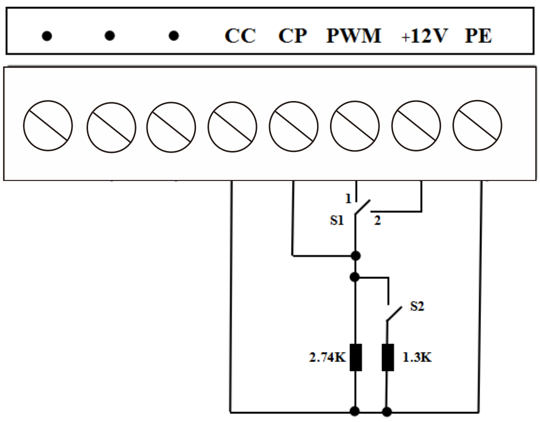
Wiring explanation:

◆ When non-loaded, S1 switch to position 2, where CP will detect 12V, CP will detect the output status (status value--0X01);

◆ When the car connected, load 2.74K resistance, CP detect 9V in result in charging status 2 (status value--0X02);

◆ Then switch S1 to position 1, output ± 9V PWM pulse changeable;

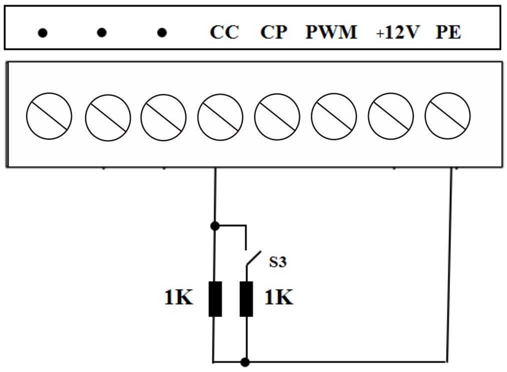
◆ Finally close S2 to make 1.3K resistance and 2.74K in parallel, then the total resistance external is about 1K, CP detect 6V PWM, in result in charging status 3(status value--0X03);



AC charging guidance diagram

2) Diagram below shows the function of DC charging guidance:

CC detect the connection status of DC charging cable, 12V means disconnected (0X12), CC will detect 6V(status 0X06) when connect 1K resistance; CC detect 4V(status 0X04) when S3 closed.



DC charging guidance diagram

Visiting mode

The visiting mode of expansion board is SM area address visit, range from SMB114--SMW124, table 4-10-9 shows the definition of SM visiting address:

Table 4-10-9 map access address

Function	Map access address	Function	Map access address
CC input	SMW116	PWM output	SMW124
CP input	SMW118	Reserved output	SMW126
Guidance	SMW120	Module type	SMB114
Reserved input	SMW122	Module status	SMB115

The type and status of modules will display in corresponding address when PLC access expansion board, details show as table 4-10-10

Table 4-10-10 module status definition

<table>
<tr><th colspan="2">Name</th><th>Function</th><th>Value</th><th>Explanation</th></tr>
<tr><td colspan="2" rowspan="2">Module type</td><td>With module</td><td>0x1E</td><td rowspan="2">Access one time when power on</td></tr>
<tr><td>Without module</td><td>0x00</td></tr>
<tr><td rowspan="9">Guidance status</td><td rowspan="3">DC charging</td><td>CC disconnected</td><td>0x12</td><td>CC detect voltage 12V</td></tr>
<tr><td>CC connect status1</td><td>0x06</td><td>CC detect voltage 6V</td></tr>
<tr><td>CC connect status 2</td><td>0x04</td><td>CC detect voltage 4V</td></tr>
<tr><td rowspan="5">AC charging</td><td>CC disconnected</td><td>0x12</td><td>Same as DC</td></tr>
<tr><td>CC connected</td><td>0x00</td><td>CC detect voltage 0V</td></tr>
<tr><td>Charging status 1</td><td>0x01</td><td>CP detect voltage 12V</td></tr>
<tr><td>Charging status 2</td><td>0x02</td><td>CP detect voltage 9V</td></tr>
<tr><td>Charging status 3</td><td>0x03</td><td>CP detect voltage 6V</td></tr>
<tr><td>Abnormal</td><td>Guidance detection abnormal</td><td>0xFF</td><td>CC/CP detect nonstandard voltage</td></tr>
<tr><td rowspan="2">Module status</td><td colspan="2">Run normal</td><td>0x00</td><td>Expansion board normal</td></tr>
<tr><td colspan="2">Communication fail</td><td>0x01</td><td></td></tr>
</table>

	Calibration fail	0x02	SF indicator flashing
	Access timeout	0xFF	Cannot access

4.10.4 Installation of BD Board

Following the below steps to install Analog I/O and CAN expansion boards:

1. Make sure the power of CPU and related devices are all cut-off, lay down the CPU module.
2. Remove the cover plate along the left-side of Expansion Board.
3. Plug the Expansion Board into board slot, make sure the pins are fit closely with the socket.
4. Make the right-side of cover plate inclined to insert into slot on board slot of the CPU.
5. Press the left-side of the cover plate inclining, fit it onto CPU again.

Notice

- CTH200 series CPU all support Analog I/O expansion board, but only H224X, H226XL and H228XL support CAN-01.
- CAN-01 can be used for CPU directly, but Analog I/O expansion board can be used for CPU only by invoking ExBoard_H200 library or special memory. For more details, please refer to chapter *6.4.2 Expansion Board Access Address*.
- Hot-plug is not supported for the board.

5 Communication Protocol and Networking Mode

CTH200 series PLC support various ways of network communication, from normal serial port communication to Ethernet. Supporting communication protocols including PPI, MPI, Freeport, CANopen, Ethernet, profibus-DP and PROFINET. Choose protocol in your need, see chapter *2.2 Communicate with CTH200* for specific communication mode and connection.

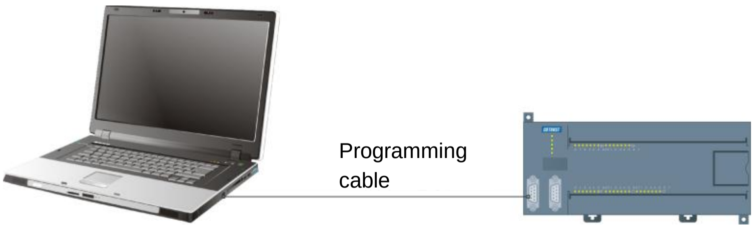
5.1 PPI Communication

PPI is a master-slave protocol by which the master sends request to the slave for responding. Slaves would be standby without master request. CTH200 PLC support PPI protocol to use NETR/NETW instructions transferring data with baud rate 9.6kbps, 19.2kbps and 187.5kbps. CTH200 PLC uses the build-in programming port as normal Communication networking interface, no need for additional configuration for module or software.

Up to 31 PLCs, Copanel HMIs, Siemens OP/TP panel or MPI cards as station on Host involve in the PPI communication network.

Single master in PPI

The following example is one-master with one-slave. Programming PG/PC as Master, CTH200 PLC as Slave.



Copanel HMI as Master, CTH200 PLC as Slave.

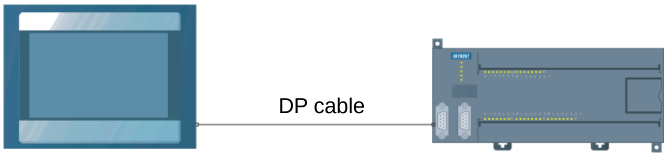
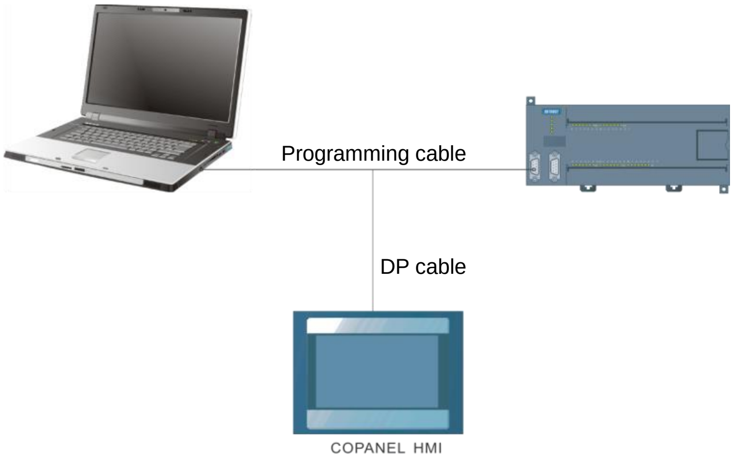


Figure 5-1 single master PPI network

Note: It's recommended to use PLC Programming cable from Cotrust. Communication cable selection is depend on specific HMI, to avoid devices damaging.

Multiple masters in PPI

The following example is two masters with one slave. Programming PG/PC and HMI as Masters, CTH200 PLC as Slave.



The following example is multiple masters with multiple slaves. Programming PG/PC and HMI as Masters, CTH200 PLC as Slaves.

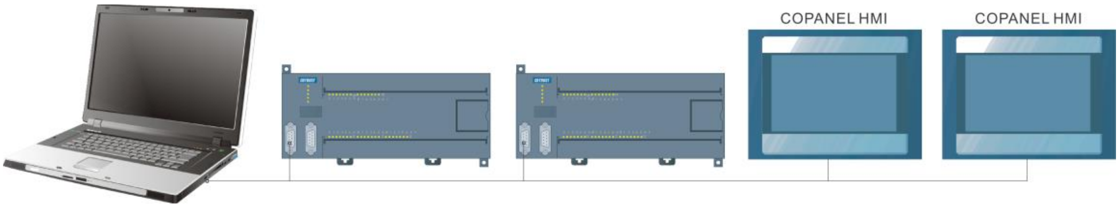


Figure 5-2 Multi masters PPI network

The following example is multiple CTH200 CPUs interconnection, which using NETR/NETW instructions to achieve network communication. Each CTH200 CPUs can be either Master or Slave, they communicate with each other via the PPI port.

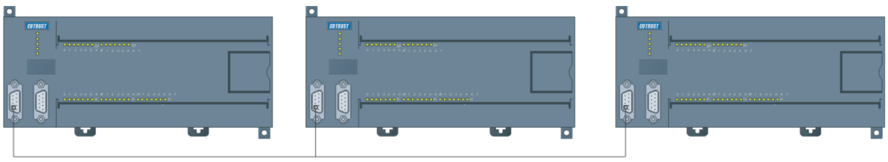


Figure 5-3 Multi CPU interconnection network

5.2 MPI Communication

MPI Support master-slave and master-master communication.

CTH200 PLC can be connected to MPI network by using built-in interface with baud rate 19.2Kbps or 187.5Kbps, and also communicate with SIMATIC S7-300/S7-400 CPU as slaves in MPI network.

Multiple masters in MPI network

As shown in the following figure, S7-300 PLC and Copanel HMI as masters in MPI network, CTH200 PLC as master. S7-300 master PLC read/write the data of CTH200 by using XGET and XPUT instructions.

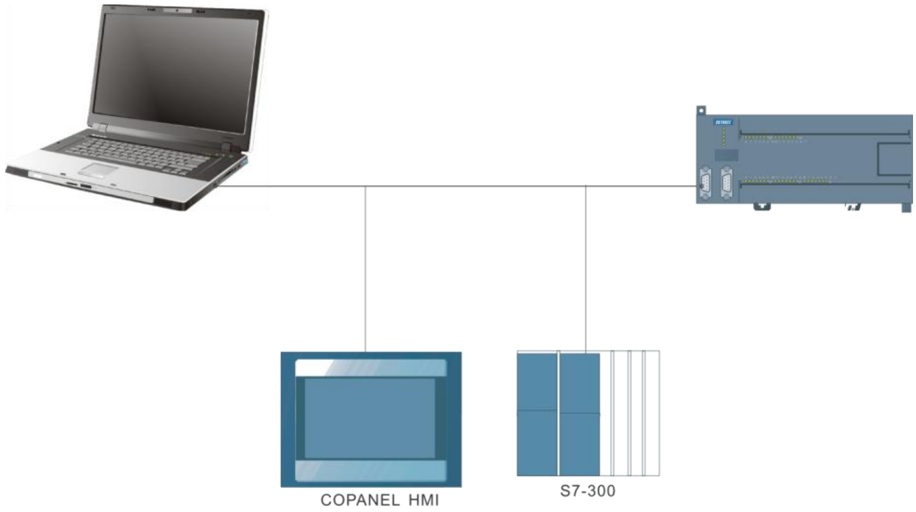


Figure 5-4 Multi masters MPI network

5.3 Freeport Communication

By using Freeport communication, CTH200 PLC can communicate with any third-party devices which disclosed communication protocols. These devices include Data acquisition module, controller, printer or bar code reader, drives, modems, host PC and so on.

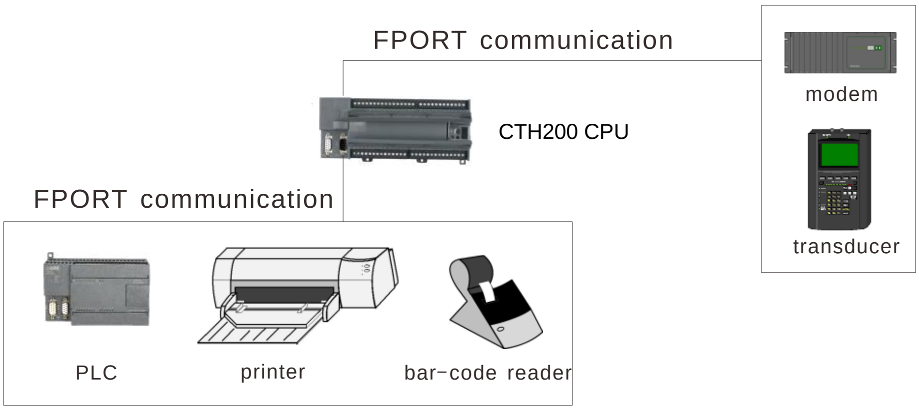


Figure 5-5 Freeport communication

5.4 CANopen Communication

CANopen bus communication is a fast master-slave network with high-reliability. Expand a CAN communication board(CTH CAN-01S1-EB) on CPU(H224X/H226XL/H228XL) to be CAN master in the CTH200 CANopen communication system, which could exchange data with remote CAN slave(CTH2 277-0AC32) and its expansion boards as well as communicate with servo(COTRUST E10) or frequency converter supporting CANopen communication.

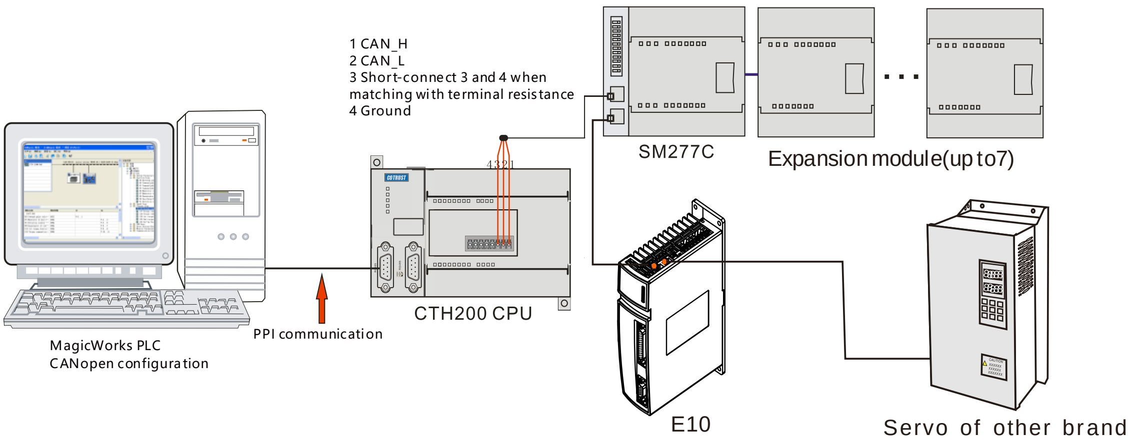


Figure 5-6 CANopen communication framework

As shown in the above figure, CTH200 PLC connects with SM277 CANopen module and its expansion modules via CAN-01 expansion board, thus to control the Servo Drives.

Note: For more details about related terminals and wiring on the expansion board, please refer to chapter *4.10.2 CAN Expansion Board Specifications*.

5.5 Ethernet Communication

CTH200 series PLC support various ways of communication, from Ethernet to Modbus TCP and UDP/PPI to telecommunications, monitor and data exchange using Mico.

CTH200 series PLC, which could communicate with other CPU modules, HMI or PC, support Modbus/TCP, UDP/PPI protocols for local Ethernet communication as well as remote program and monitor by Mico, a cloud platform where you can real-time view the status of remote devices.

Refer to chapter *6.2 Modbus TCP Communication* and 6.3 UDP_PPI Communication for the examples. Refer to *Mico Remote Monitor Manual* and *How to Use Mico for Program and Data Monitor on Remote PLC* for details on remote program and data monitor.
Download address of manuals: http://www.co-trust.com/Download/index.html

The typical network architecture of CTH200 shows as below:

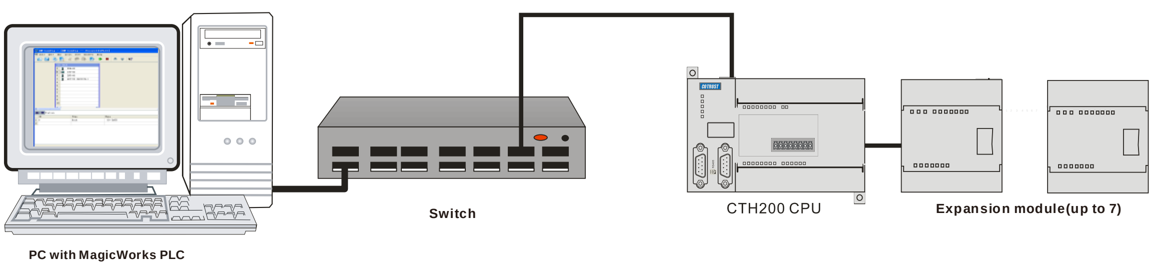


Table 5-7 Ethernet communication framework

5.6 Profibus-DP Communication

With high reliability, CPU, as a slave of some master, enable to connect to PROFIBUS field bus to exchange data fast and in real time as well as operate alone by DP slave module. Figure 5-9 and 5-10 show the profibus-DP network architecture of SM277A and SM277B DP slave respectively.

Refer to appendix G SM277A Module and appendix H SM277B Module for more information on using SM277A and SM277B DP slave

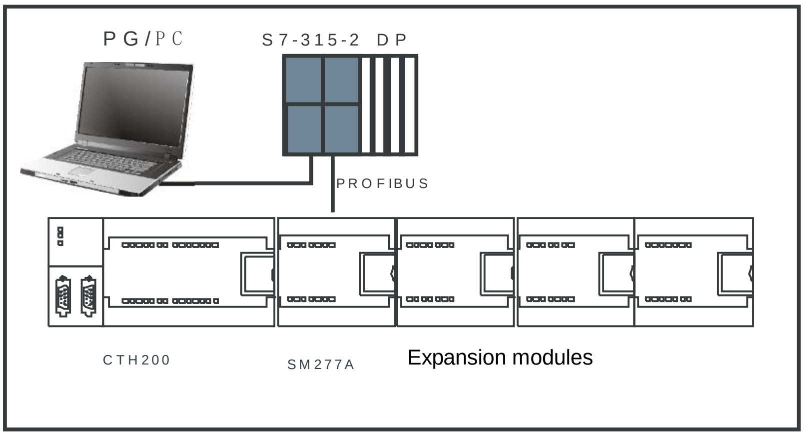


Figure 5-8 Construction of SM277A PROFIBUS-DP network

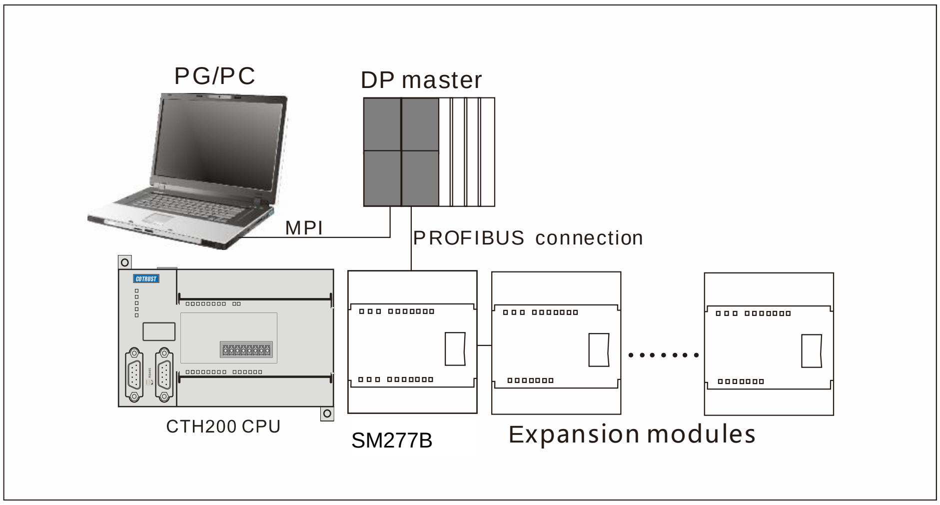


Figure 5-9 Construction of SM277B PROFIBUS-DP network

5.7 PROFINET Communication

Connect expansion modules(up to 8), as the remote I/O expansion, to PN master by SM277PN slave module. Typical PROFINET network architecture shows as bellow:

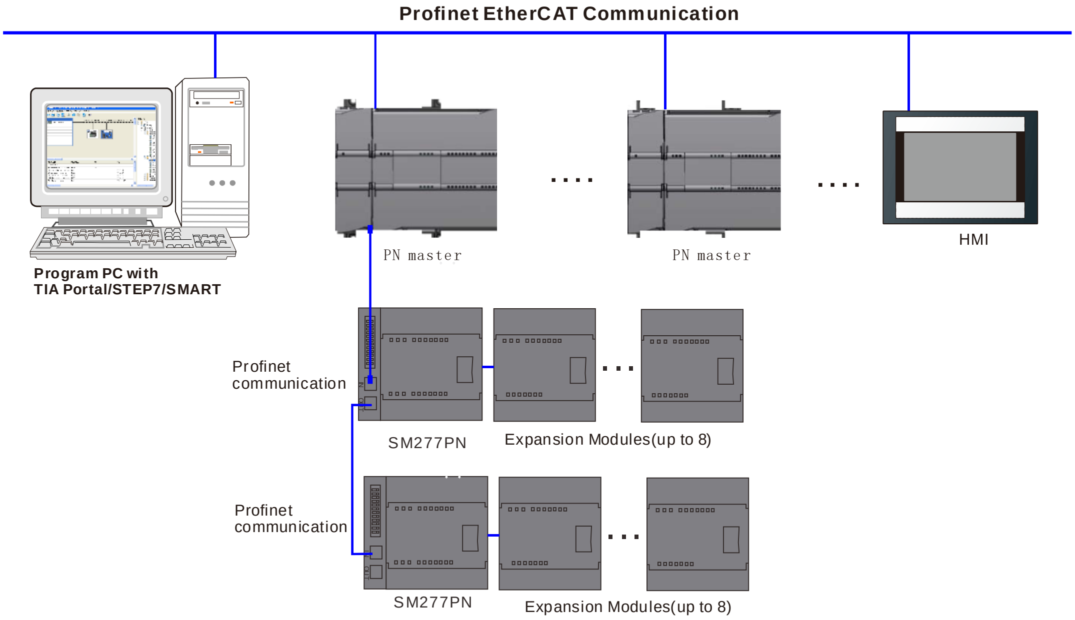


Figure 5-10 PROFINET network architecture

Refer to *CTH200 series profinet slave module user manual* for More details and example of PROFINET communication module.

Download address: http://www.co-trust.com/Download/index.html

6 Applications

Examples in this chapter show you the various ways of communication the CTH200 series PLC supported.

6.1 CANopen Communication

Example in this section shows you the data transmission via CANopen communication.

6.1.1 Components

This example use CAN master expansion Board to build CANopen network consisting of slave modules. The example using CAN master expansion board build a CANopen network consisting of slave modules.

CANopen communication components

Table 6-1 Example components

Components	Explanation
A PG/PC with MagicWorks PLC	MagicWorks PLC V2.08 version and above
A PLC programming cable	Connect program station with master station
CTH200 CPU-control module	CTH200 series PLC as CANopen master station
CAN master station expansion board	CAN-01 master station expansion board
A slave device	SM277C
CANopen bus cable	Shielded network cable or twisted-pair
Three expansion board which can be connected to SM277C	Two digital modules and one analog module of CTH200
Two servo drives	One E10 and other brand servo

6.1.2 Network Connection

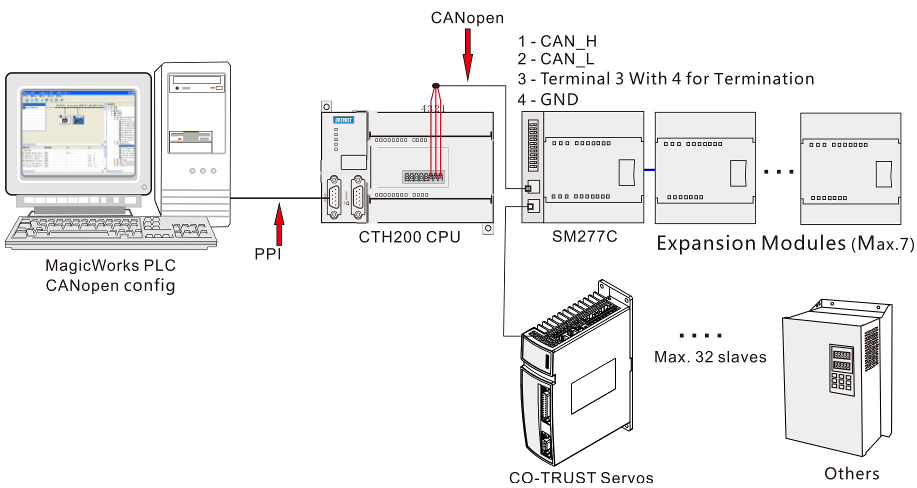


Figure 6-1 CANopen communication framework

Figure below shows how to connect CANopen bus with CPU terminal:

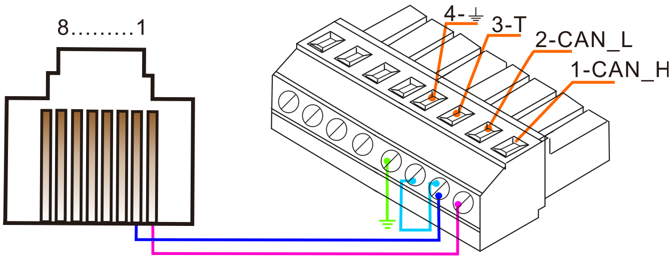


Figure 6-2 CANopen bus connection

Note: Please short-connect 2 with 3 when matching with terminal resistance

6.1.3 Operating Steps

CANopen network configuration

1. Open the EasyCAN configuration interface

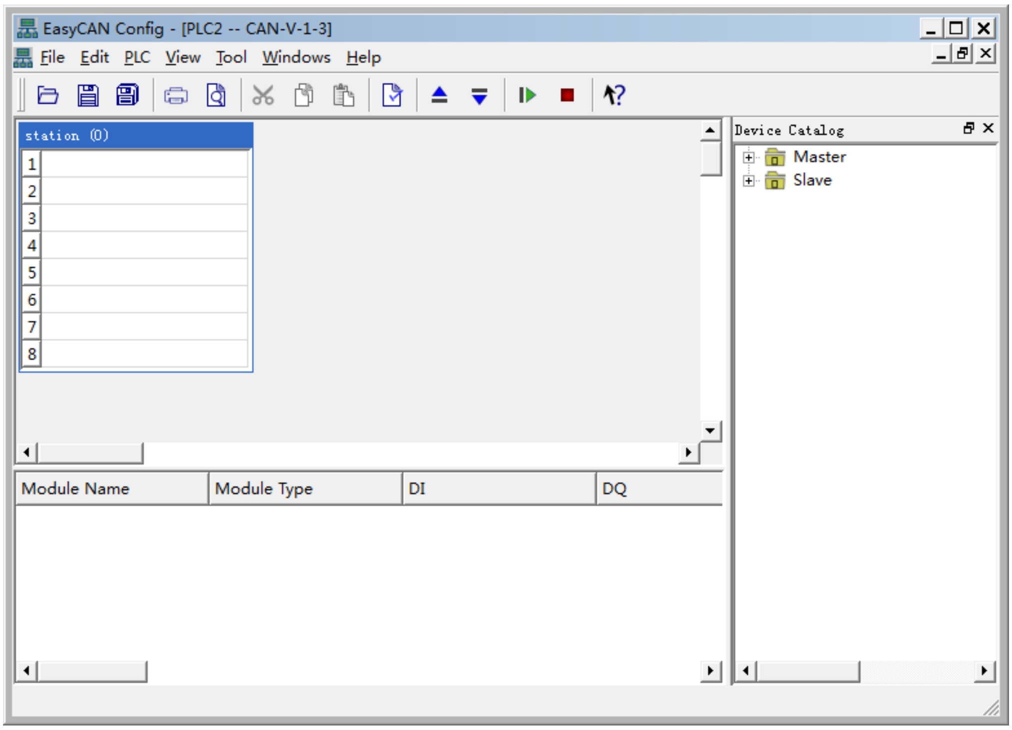


Figure 6-3 EasyCAN configuration page

2. Add Master station

The CANopen Bus editor shows in the following figure, select the CTH200 PLC corresponding with device currently in use in the “Device Catalog” of Master, drag and drop it into slot 1 of Station 0.

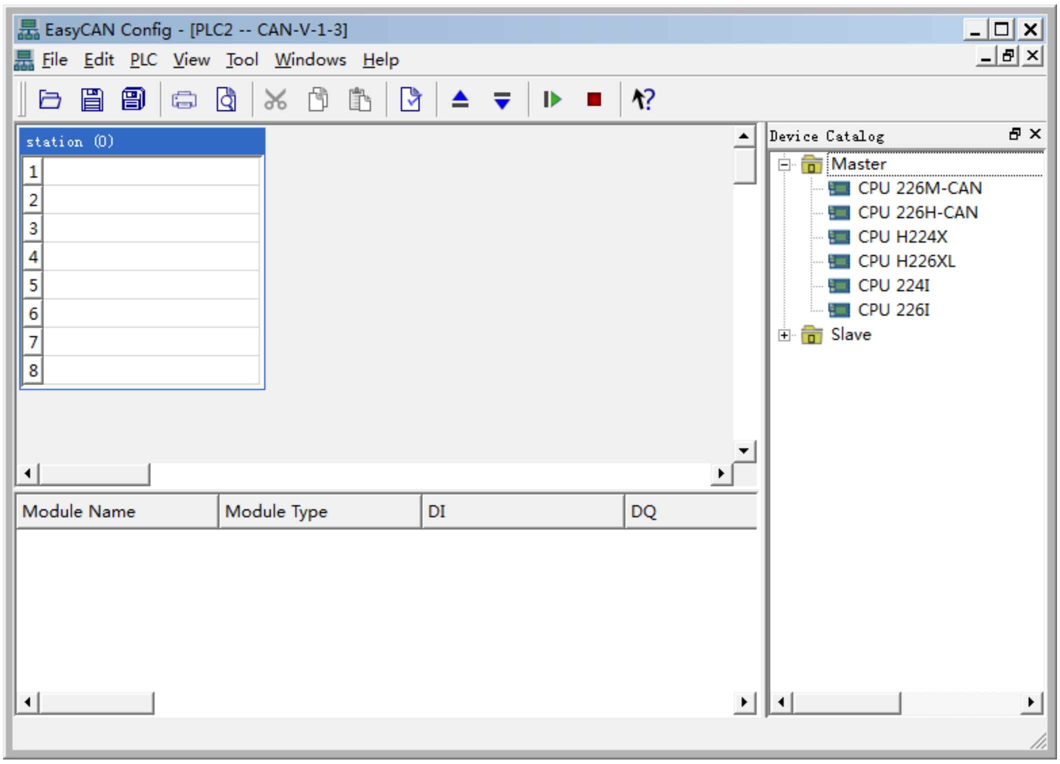


Figure 6-4 Add master station

Then, double-click the master CPU H226XL in slot 1, an information dialog would pop-up, shown in the following figure.

Set Baud Rate 1000kbps, select CPU for Master Type, 1 for Node ID and check the “Use CAN Bus” box.

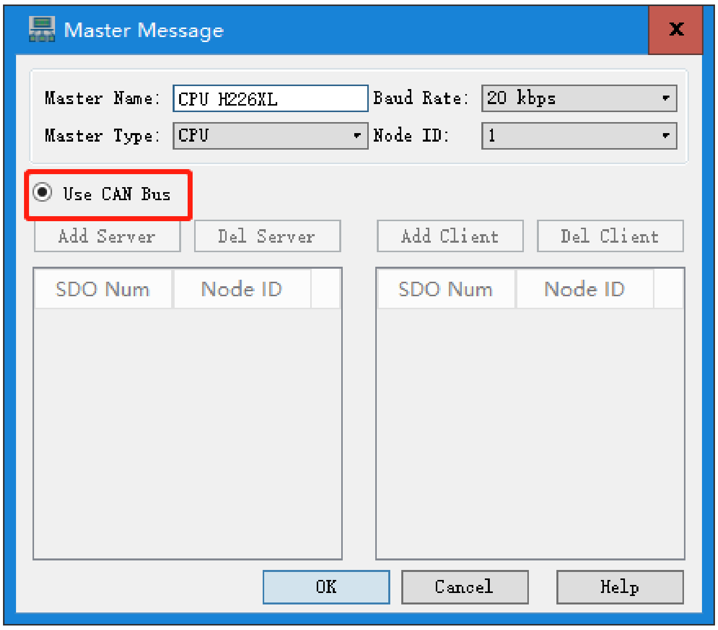


Figure 6-5 Configure master

3. Configure slave station

1) Add Slave station

Click to expand the Device Catalog, open the slave node to select the slave device corresponding with current device, drag and drop it into the configuration area, while it would be connected with CAN BUS automatically.

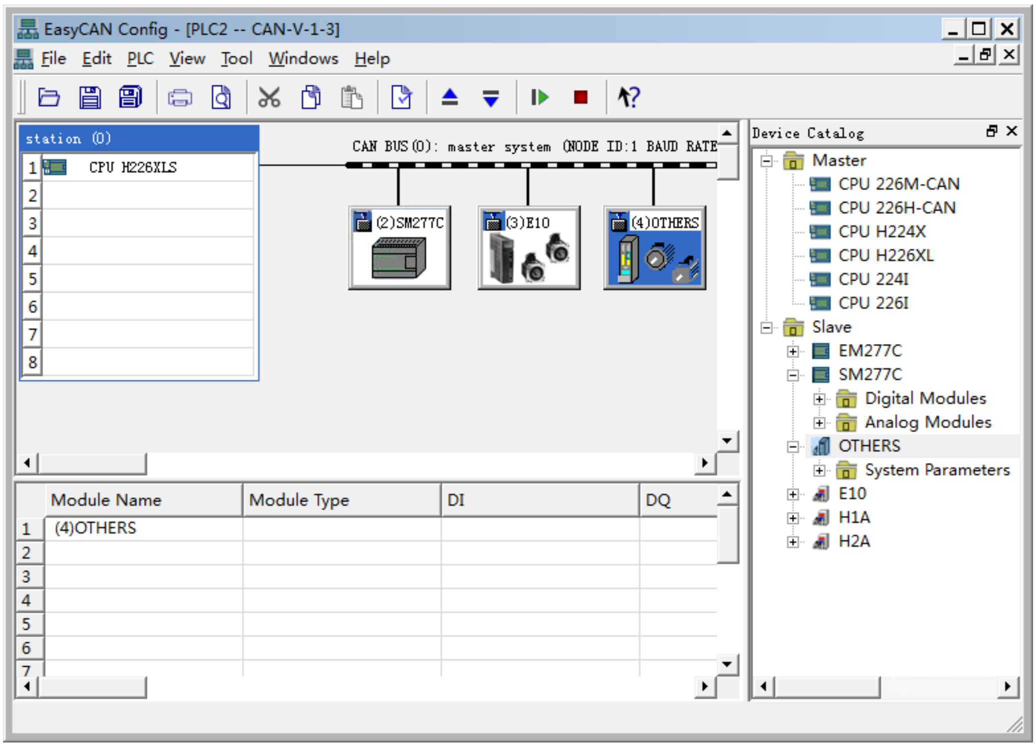


Figure 6-6 Add slave

2) Setting slave parameters

Double-click the icon below CAN BUS to pop-up the Slave Info Dialog.

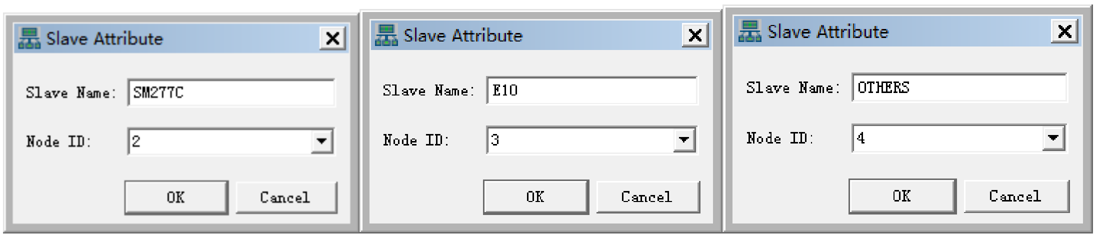


Figure 6-7 Configure slave

Here you can set the following slave parameters:

■ Slave Name: Name of the Slave (Default as SM277C, CT E10 or OTHERS).

■ Node ID: Range from 1 to 127.

3) Configure expansion modules for Slave SM277C

Click the right-side of Device Tree to open the Analog module or Digital module node under SM277C, check the module corresponding with current device, drag and drop it into specified slot, then the module can be added into current slave.

You can also double-click the required module nodes on Device Tree to add them into current slave in order. After which the I/Q address would be assigned automatically.

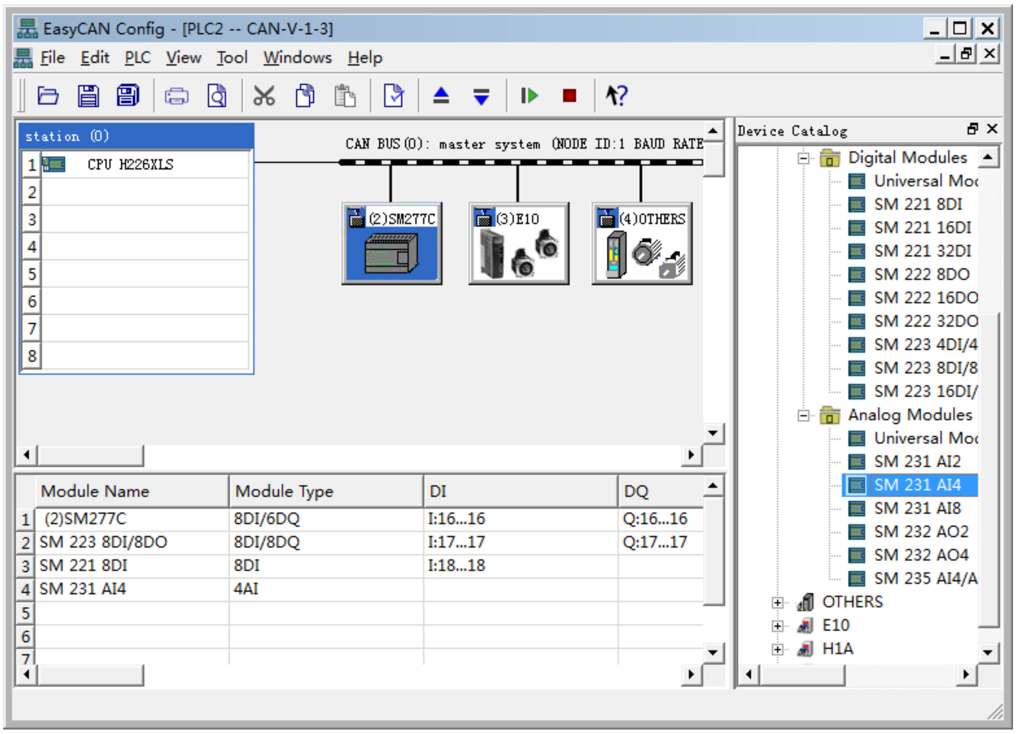


Figure 6-8 Configure slave

Modify the I/O Configuration

MagicWorks PLC V2.08 and above supports user-defined Start Address for I/O and adds V Memory image to allow users select Memory image. Double-click the non-blank line to pop-up this dialog:

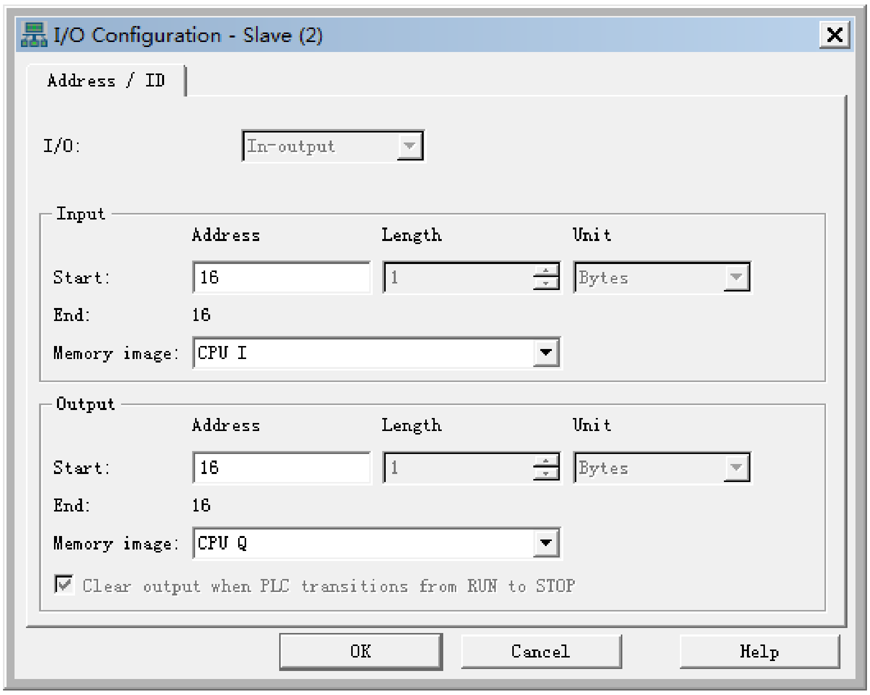


Figure 6-9 I/O configuration

You can modify the Start and End Addresses for Module I/Os and Memory image, the latter is related with R/W attributes for currently selected parameters:

Digital Input: Image to CPU I Memory and V Memory;

Digital Output: Image to CPU Q Memory and V Memory;

Analog Input: Image to CPU AI Memory and V Memory;
Analog Output: Image to CPU AQ Memory and V Memory;

Table 6-2 Valid address range for Memories

Memory	Address range
I	IB16~IB79
Q	QB16~QB79
AI	AIW64~AIW386
AQ	AQW64~AQW386
V	VB0~Vmax

Important Note: For specified modules, the I/O Type and Length cannot be modified unless the module type is Universal Module. The system would check the validity automatically for selected address. If corresponding module type cannot be found under SM277C, you can add Universal Module and make custom settings.

4) Configure Parameters for Slave CT E10

Click the Device Tree to open the various parameter types under CT E10, select required parameters and add them into module list. After that the minimal available I/Q addresses would be assigned automatically, you can modify that by double-clicking.

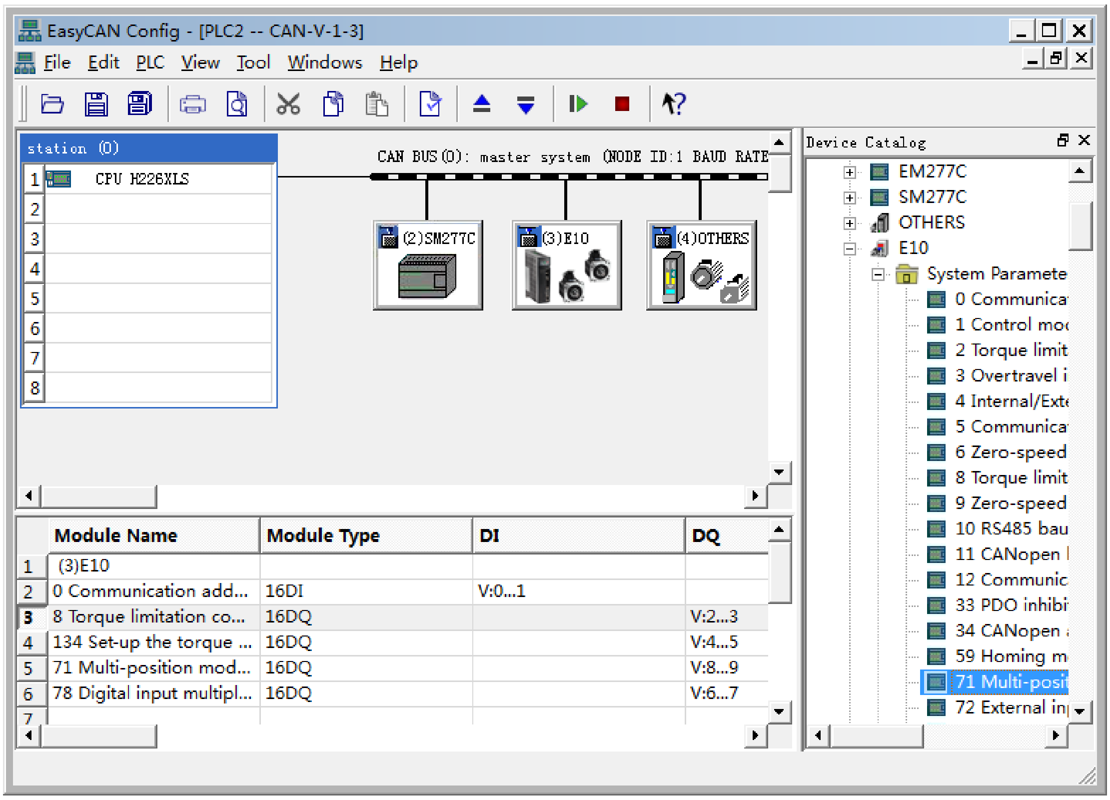


Figure 6-10 Configure slave E10

Modify IO Configuration

Double-click any parameter in the module list, the following dialog appears:

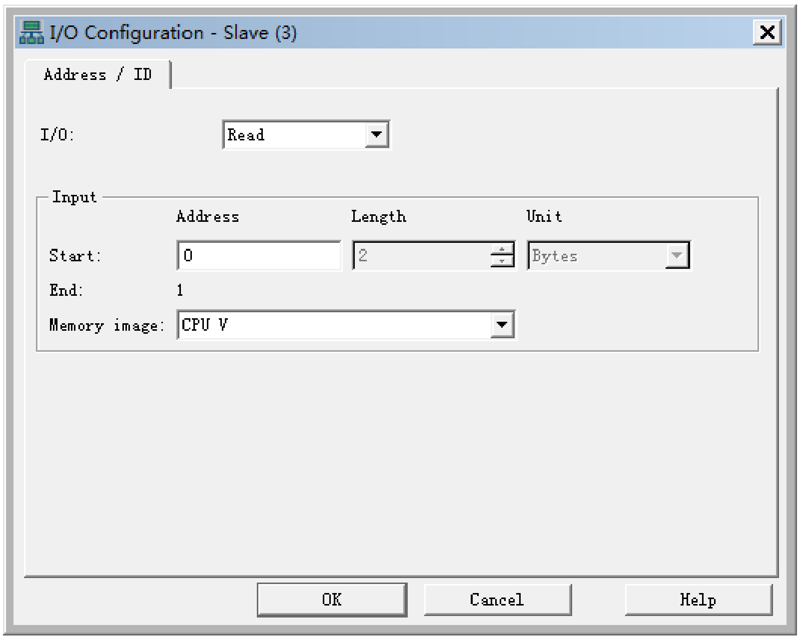


Figure 6-11 I/O configuration

I/O: Set the Read/Write attribute for parameters.

Start

• Address: set the Start address for parameters.

• Length: use Default value, not editable.

• Unit: use Default value, not editable.

End: Use the value calculated by system.

Memory image: Select the Memory image type for CPU, the available type is related with I/O Read/Write type.

5) Configure Parameters for Third-party Slave

MagicWorks PLC V1.73 and above supports the third-party CANopen slave configuration, click to expand the Device Tree on the right, open OTHERS slave node to select required parameters and add them into module list. After that the minimal available I/Q addresses would be assigned automatically, you can modify that by double-clicking.

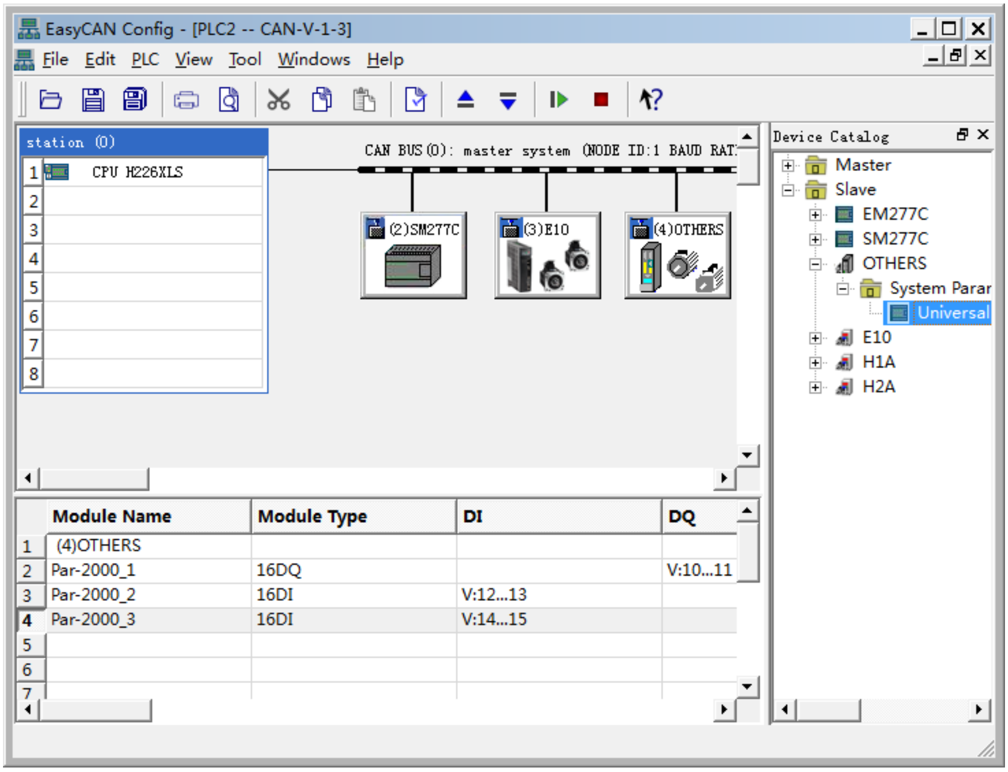


Figure 6-12 Configure third-party slave

Modify IO Configuration

Double click any parameter in the module list, the following dialog appears:

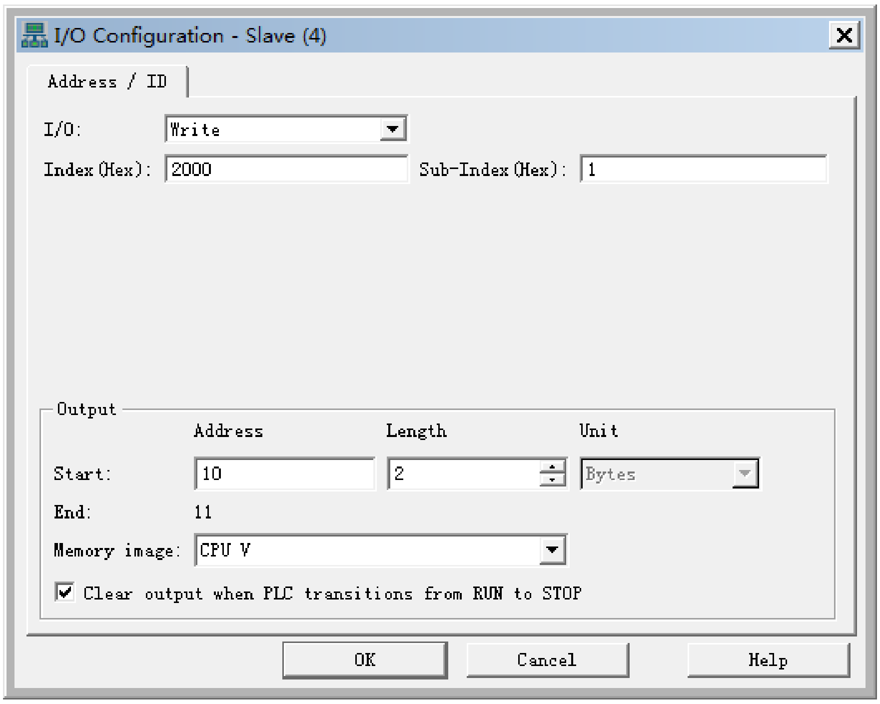


Figure 6-13 I/O configuration

I/O: Set the Read/Write attribute for parameters.

Index(Hex): Main index for selected slave parameter.

Sub-Index(Hex): Sub index for selected slave parameter.

Start

• Address: Set the Start address for parameters.

• Length: It's required for Total length of configured R/W data is multiple of 4.

• Unit: Use Default value, not editable.

End: Use the value calculated by system.

Memory image: Select the Memory image type for CPU, the available type is V and Q.

Clear output when PLC STOP: Check this option to clear the contents in image address; or else the contents remain unchanged.

Notice

- For valid address range, refer to Table 6-2;
- The EDS Importing for third-party slave is not supported currently, please refer to the related product manuals for inputting main index and sub index;
- In actual connection. If the third-party slave has detected error, the corresponding SMB information area would display : 0x7 configuration parameter error;
- For details of Cotrust Servo Drives, please visit at http://www.co-trust.com

4. Connection and Configuration of Hardware

➢ Connect CPU H226XL with PC by using communication cable (the USB end connects to the USB socket of PC, the RS485 end connects to the RS485 port of CPU H226XL).

➢ Connect the CAN port on CAN-01 board of CPU to the CAN port of SM277C by using communication cable.

➢ Set the DIP switch referring to the Table 6-3: Communication rate is 1000kbps, Node ID is 4.

Figure 6-3 DIP switch selection

DIP7-DIP1 (Node address): in binary, MSB - DIP7, LSB - DIP1.

DIP10-DIP8 (Baud rate): in binary, MSB - DIP10, LSB - DIP8.

Notice

0 is the global address, which is forbidden while using.

Table 6-3 DIP switch configuration

DIP10	DIP9	DIP8	Communication rate (kbit/s)	Max distance(m)
OFF	OFF	ON	20	2500
OFF	ON	OFF	50	1000
OFF	ON	ON	125	500
ON	OFF	OFF	250	250
ON	OFF	ON	500	100

ON	ON	OFF	800	50
ON	ON	ON	1000	25

DIP11 (Terminal resistance): Set as ON only if the device is the last one in the device ring.

DIP12: Reserved.

- ➢ Attach the SM223 8DI/8DO, SM221 8DI and SM231 AI2*16BIT modules after the extended port of SM277C.
- ➢ Then check and confirm the wiring correctly.
- ➢ Switch on the power supplies for various devices in the system.

5. Save the current project after CANopen configuration completed, compile it in the EasyCAN configuration interface, download the CANopen configuration into PLC.

6. CPU H226XL has extended 64Byte I /64Byte Q / 162Words AI / 162 Words AQ Memory for CANopen as the Address image of each module. The I/Q Address of modules in EasyCAN configuration shows in the following figure.

	Module Name	Module Type	DI	DQ
1	(2)SM277C	8DI/6DQ	I:16...16	Q:16...16
2	SM 223 8DI/8DO	8DI/8DQ	I:17...17	Q:17...17
3	SM 221 8DI	8DI	I:18...18	
4	SM 231 AI4*12Bit/16Bit	4AI		
5				
6				

Module

After configuration, you can monitor the I/Os for SM277C, SM221 8DI, SM223 8DI/8DO and SM231 AI4*12Bits/16Bits, as shown in the following:

	Address	Format	Current Value	
1	Q16.0	Bit	2#1	6 Outputs for SM277C(Q16.0~Q16.5)
2	Q16.1	Bit	2#1	
3	Q16.2	Bit	2#1	
4	Q16.3	Bit	2#1	
5	Q16.4	Bit	2#1	
6	Q16.5	Bit	2#1	
7	Q17.0	Bit	2#0	8 Outputs for SM223 8DI/8DO(Q17.0~Q17.7)
8	Q17.1	Bit	2#1	
9	Q17.2	Bit	2#1	
10	Q17.3	Bit	2#0	
11	Q17.4	Bit	2#0	
12	Q17.5	Bit	2#1	
13	Q17.6	Bit	2#1	
14	Q17.7	Bit	2#1	
15	Q18.0	Bit	2#0	8 Inputs for SM231 8DI(I18.0~I18.7)
16	Q18.1	Bit	2#0	
17	Q18.2	Bit	2#0	
18	Q18.3	Bit	2#0	
19	Q18.4	Bit	2#0	
20	Q18.5	Bit	2#0	
21	Q18.6	Bit	2#0	
22	Q18.7	Bit	2#0	
23	AIW64	Signed	-32768	4 Inputs for SM231 4AI(AIW64~70)
24	AIW66	Signed	-32768	
25	AIW68	Signed	-32768	
26	AIW70	Signed	-32768	

6.1.4 Diagnosis

Use indicator of SM 277C/E10 and MagicWorks PLC SMB status word to diagnose CANopen master-slave network.

Diagnose by hardware

SM277C LED status

After powered, the POWER LED of SM 277C indicating "ON" (green) will light up.

If the BF and SF LED lights remain off, then SM 277C is functioning normally.

If the BF or SF LED lights up or flickers, there is an error in hardware configuration or wiring. If the operation of the main system is trouble-free, the cause of the error can be referred to the following table:

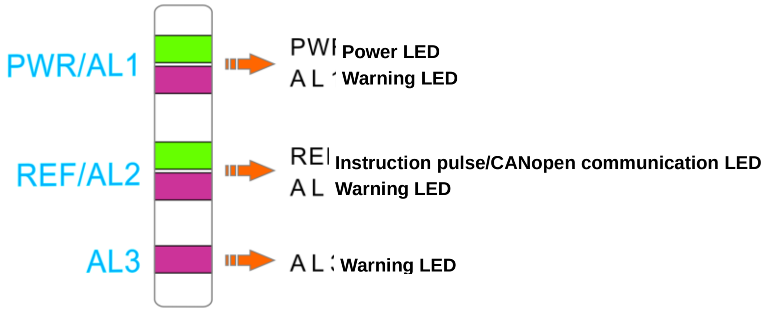
Table 6-4 SM 277C LED function

LED	On	Off	Flicker	Note
ON	Power on	No power	---	Power indicator
SF	IO expansion module no error	IO expansion module no error	---	System error
BF	No CAN network	No CAN network	Inconsistent configuration	Bus error

E10 CANopen communication indicator light

If the slave station connect to E10 servo, the E10 CANopen communication indicator light keep off before the communication connection. The indicator light on if the communication succeed, indicating that the CANopen communication is normal., the CANopen communication indicator light turn off if the CANopen communication is abnormal.

E10 CANopen communication indicator light:



Diagnose by MagicWorks PLC

CTH200 series PLC is allocated with 100 bytes SM. Users can get error information by viewing status bytes. The meaning of CAN site status word is shown in the following table:

Table5-22 CAN station status table

	Byte	Address distribution	Status value explanation
SMB (status)	100	SMB550: CAN communication status of master station(CPU)	0x00: Initialize 0x01: Disconnect 0x04: Stop 0x05: Run 0x7f: Pre-run 0xff: Configuration data error
		SMB551-SMB582: CAN communication status of 1~32 slave station(Node ID from small to large)	0x00: Initialize 0x01: Disconnect 0x04: Stop 0x05: Run 0x7f: Pre-run 0x7: Configuration parameter error
		SMB583-SMB614: Run warning of 1~32 slave station (Node ID from small to large)	bit7: IO information configuration compare with 277C reality: 0: No error 1: Error Bit6-Bit0: 277C IO expansion module diagnostic information bit0 indicate the first module, bit6 indicate seventh module: 0: No warning 1: Warning (Above is the definition of 277C slave station different from others)
		SMB615-SMB649	Reserved

Note

SMB550-SMB649 is read-only memory, user cannot write

6.2 Modbus TCP Communication

This Guide leads users to create an application by using a specific instance, thus can help users understand the Modbus TCP communication function of CTH200.

In the Modbus TCP communication network, with CPU as Slave, the communication is independent with the whole cycle period; with CPU as Master, the Receiving / Sending is controlled by user program.

When using the EtherNET port for Modbus TCP communication, CTH200 PLC can serve as Modbus TCP slave directly without any configuration, the default port number is 502 and if you need to modify IP address, please refer to Chapter *2 Getting Started*.

When CTH200 PLC serve as Modbus TCP master to communicate with other slaves, you need to use Modbus TCP Wizard or Ct_Mbus_master_tcp_single library to configure the master communication.

Notice

Please refer to "*MaigcWorks PLC user Manual*" for how to use the Modbus TCP Wizard and Ct_Mbus_master_tcp_single library. The related Manuals and Library files can be downloaded from http://www.co-trust.com/.

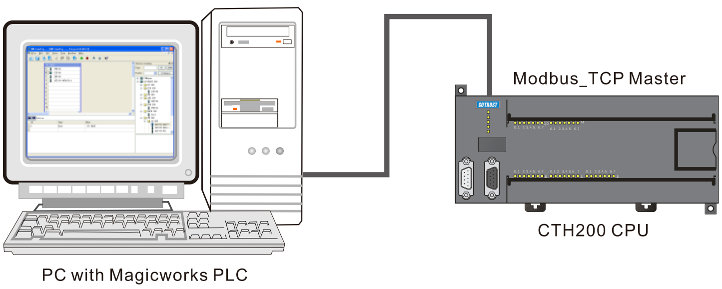
6.2.1 Components

Table 6-6 Example components

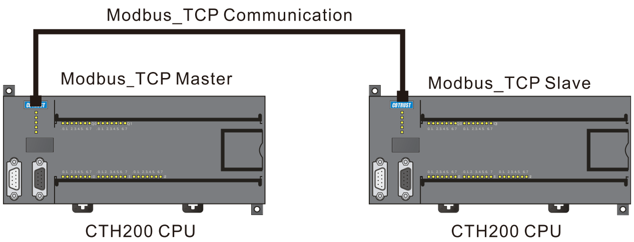
Components	Description
Program Device PG\PC	Installed with MagicWorks PLC (V2.08 and above), used for CTH200 PLC configuration, programming and debugging.
CPU	Two CTH200 PLC, one for Modbus_TCP master, the other for Modbus_TCP slave, they communicate with each other via EtherNET port.
Standard Network Cable	• connect CTH200 PLC and Program device • connect PLC (Modbus_TCP master) and PLC (Modbus_TCP slave)

6.2.2 Network Connection

Connect Program device with CTH200 PLC by standard network cable, then use the CT_Modbus_TCP library from COTRUST to program the Modbus master (CTH200):



Connect Modbus_TCP master with slave by standard network cable, the master will write and then read data from specified address to the slave to realize the Modbus TCP communication:



Notice

CPU H224X/H226XL/H228XL not support adaptive crossover, thus the PLC should use Crossover Cable to implement ModBus_TCP.

6.2.3 Operating Steps

Step 1: Connection

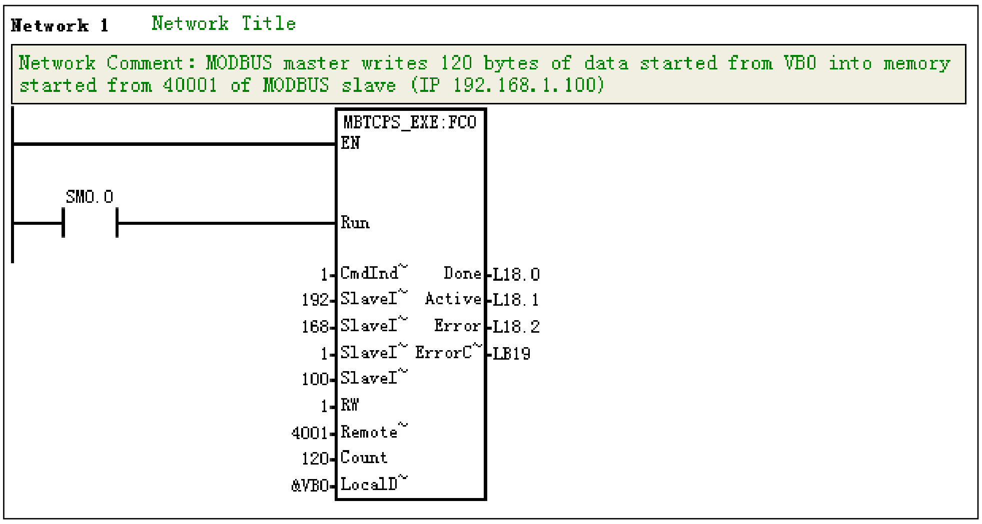
Connect PC with Modbus_TCP master with a standard network cable.

Step 2: Set Communications

Create a new project in MagicWorks PLC and add a CTH200 station, refer to chapter *2.2 Communicate with CTH200* PLC Communication Settings to establish a communication connection between CTH200 PLC and PC.

Step 3: Program for the Modbus master (CTH200 PLC)

1) Open the Program Block in MagicWorks PLC to program for the Modbus_TCP master. You can use either Ct_Mbus_master_tcp_single library or Modbus TCP Wizard, here are MBTCPS_EXE instruction from Ct_Mbus_master_tcp_single library as example:



<Note> If multiple R/W operations need to be configured, it's recommended to use the Modbus TCP Wizard in the MagicWorks PLC. For details, refer to *Magicworks PLC User Manual.*

2) When finishing program, compile and download it into Modbus_TCP master device.

The following table is the definition of MBTCPS_EXE instruction

Symbol	Variable type	Data type	Description
EN	IN	BOOL	Enable instruction
RUN	IN	BOOL	Start communication, edge triggered
CmdIndex	IN	BYTE	number for calling MBTCPS_EXE, which cannot be repeated, valid range 1~255
SlaveIP0	IN	BYTE	1st byte of the slave address
SlaveIP1	IN	BYTE	2nd byte of the slave address
SlaveIP2	IN	BYTE	3rd byte of the slave address
SlaveIP3	IN	BYTE	4th byte of the slave address
SlavePort	IN	WORD	Monitored port of slave, if the slave is a CTH200 PLC, then the port is default as 502.
RW	IN	BYTE	Read = 0; Write = 1
RemoteAddr	IN	DWORD	ModBus Address (default as 40001)

Count	IN	WORD	No. of elements (1-120 words or 1-1920bits)
LocalDataptr	IN	DWORD	Local data pointer (such as &VB1000)
Done	OUT	BOOL	Done flag (0 = Undone; 1 = Done)
Active	OUT	BOOL	Instruction activation (0 = Active; 1 = Inactive)
Error	OUT	BYTE	0: No error 1: Error
ErrorCode			Error code, valid when Done = 1 Details refer to the following List.

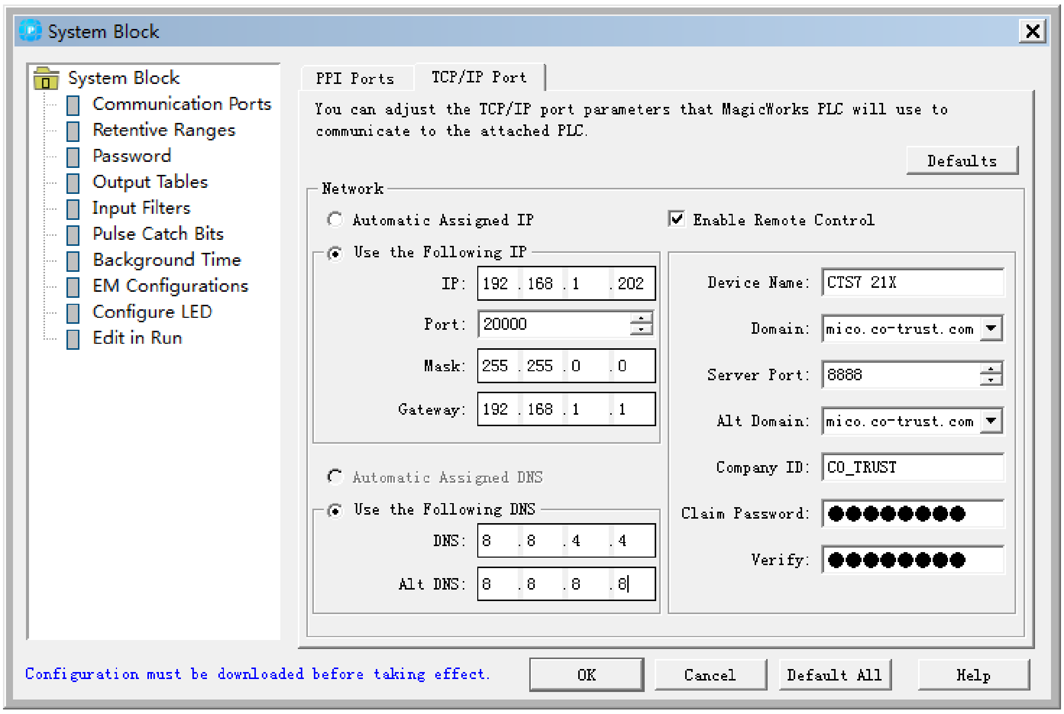
Definition of output Error code:

Error Code	Description
0	No error
1	Reached the max number of connections
2	Establishing connection
3	Timeout error
4	Error with requested parameter
5	Instruction not enabled
6	Connection is busy handling other request

Step 4: Modbus TCP master interconnect with its slave

When configuring Modbus TCP slave, the slave IP must be corresponding with IP set by master. If you need to modify IP address of the slave, open the System Block in Magicworks PLC as following figure to operate here.

Additionally, for ModBus TCP communication between PLC, it's required that the port number of slave should not less than 1024.



Step 5: Debug

Following the above orders to connect the master and slave, then read out 120 data started from

Modbus_TCP address from 40001 in the state table of MagicWorks PLC. Same content with data in the memory for Modbus_TCP master means successful Modbus TCP communication.

6.2.4 ModBus TCP Slave Address Mapping

ModBus address is consist of Data Type and 5~6 character values of the offset. The first 1 or 2 is Data Type, others are values conforming to the data type. ModBus-TCP slave support the following addresses:

Address Image of slave to TCP protocol

ModBus slave address	CTH200 address
000001	Q0.0
000002	Q0.1
000003	Q0.2
...	...
000127	Q15.6
000128	Q15.7
010001	I0.0
010002	I0.1
010003	I0.2
...	...
010127	I15.6
010128	I15.7
030001	AIW0
030002	AIW2
030003	AIW4
...	...
030032	AIW62
040001	VW0
040002	VW0+2
...	...
04xxxx	VW0+2 x (xxxx-1)

6.3 UDP_PPI Communication

This section show you an application of CTH200 UDP_PPI communication (EtherNET Port).

In the UDP_PPI network, CPU works as master which is independent with the whole cycle period; and the Receiving / Sending of message is controlled by user program.

When using the EtherNET port for Modbus TCP communication, CTH200 PLC can serve as Modbus TCP slave directly without any configuration, the default port number is 502 and if you need to modify IP address, please refer to the Chapter *2 Getting Started*.

The CTH200 PLC can serve as UDP_PPI master or slave to communicate with other devices in the same LAN by using NETW/NETR or UDP_NETR/UDP_NETW instructions. The CTH200 PLC can also be used as slave of HMI in the UDP_PPI network.

Notice

For details about how to use NETW/NETR Wizard, please refer to *"MaigcWorks PLC user Manual"*. The related Manuals and Library files downloaded address: http://www.co-trust.com.

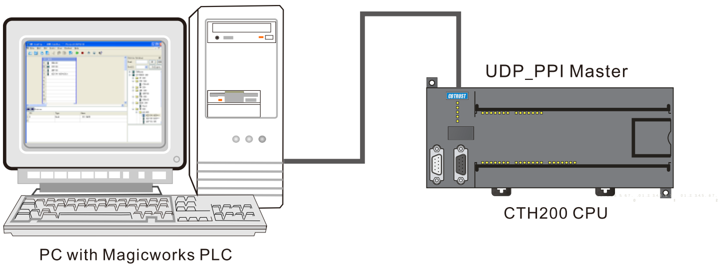
6.3.1 Components

Table 6-5 UDP_PPI example components

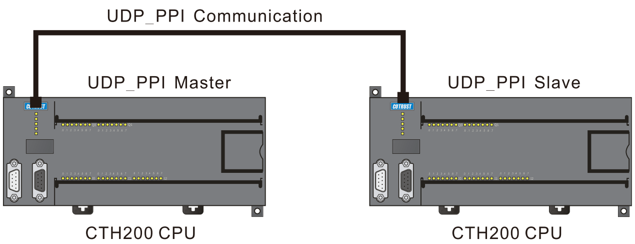
Components	Description
Program Device PG\PC	Installed with MagicWorks PLC (V2.08 and above), used for CTH200 PLC configuration, programming and debugging.
CPU	Two CTH200 PLC, one for UDP_PPI master, the other for UDP_PPI slave, they communicate with each other via EtherNET port.
Standard Network Cable	• Connect the CTH200 PLC with PG/PC • Connect CTH200 PLC (UDP_PPI master) with CTH200 PLC (UDP_PPI slave)

6.3.2 Network Connection

Connect the PG\PC with CTH200 with standard cable to be used for UDP_PPI master (CTH200 PLC) programming:



Connect UDP_PPI master and its slave with standard network cable, UDP_PPI master writes data from specified address to the UDP_PPI slave, then read data from UDP_PPI slave, thus realize the UDP_PPI communication:



Notice

CPU H224X, H226XL and H228XL do not support adaptive crossover, thus a crosswire should be used for UDP_PPI communication.

6.3.3 Operating Steps

Step 1: Connect cable

Using a standard cable to connect the PC with UDP_PPI master (CTH200).

Step 2: Set communication

Create a new project in MagicWorks PLC and add a CTH200 station, refer to chapter *2.2 Communicate with CTH200* PLC Communication Settings to establish a communication connection between CTH200 PLC and PC.

Step 3: Program for UDP_PPI master (CTH200)

There are two methods to Read/Write the UDP_PPI Master.

Using UDP_NETR/UDP_NETW Instructions to program for the UDP_PPI Master

Parameter table for UDP_NETR/UDP_NETW Instruction:

D	A	E	0	Error Code	0
The 1st byte for IP Address					1
The 2nd byte for IP Address					2
The 3rd byte for IP Address					3
The 4th byte for IP Address					4
The upper byte for port number					5
The lower byte for port number					6
The 1st byte for remote station pointer <I, Q, M, V, DB> (4 bytes)					7
The 2nd byte for remote station pointer					8
The 3rd byte for remote station pointer					9
The 4th byte for remote station pointer					10
Data length					11
Data byte 0					12
Data byte 1					13
...					...
Data byte 199					211

D: Done (Function completely), 0= No, 1=Yes

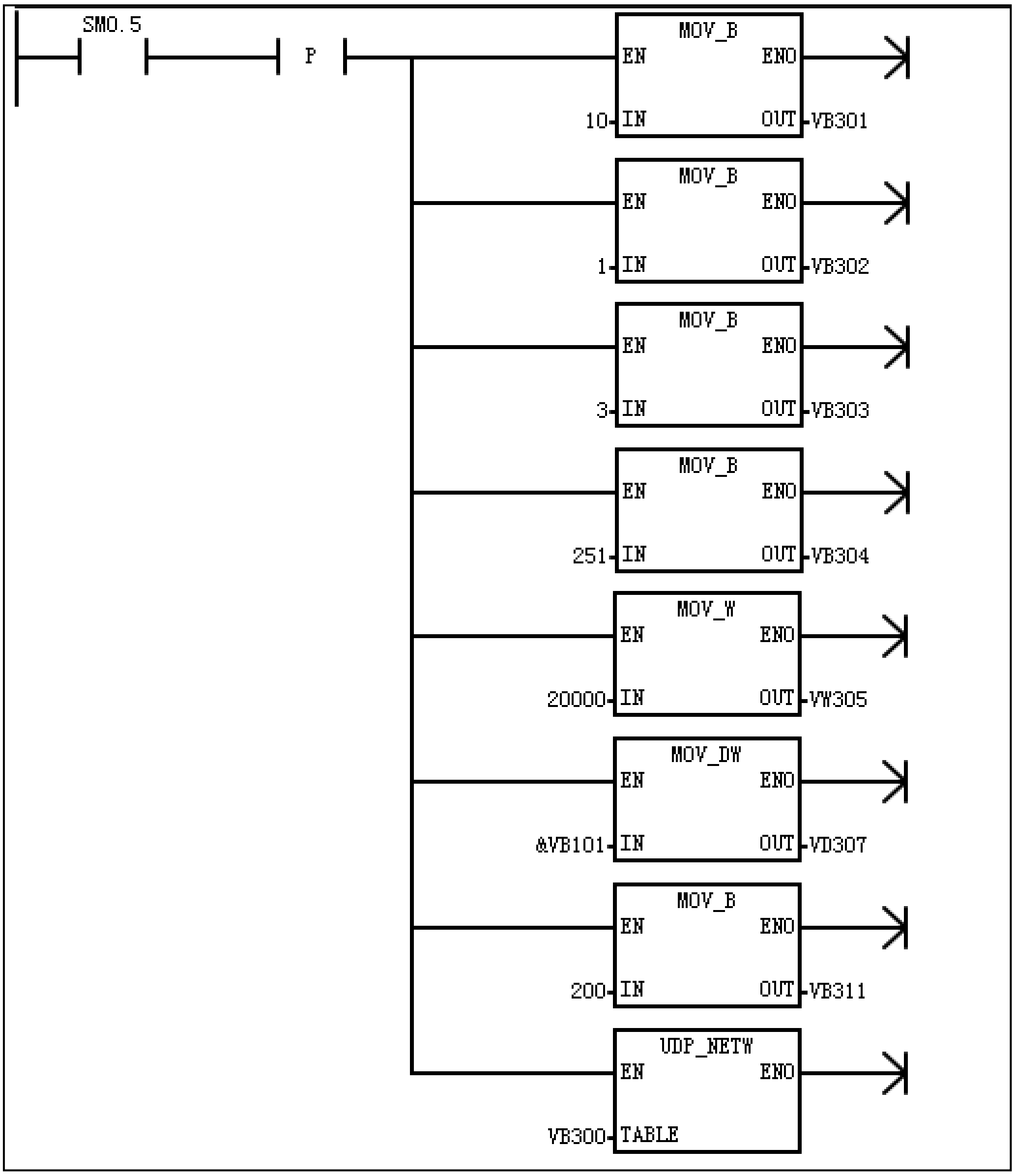
A: Active (function sequence), 0=No, 1=Yes

E: Error, 0=No, 1=Yes

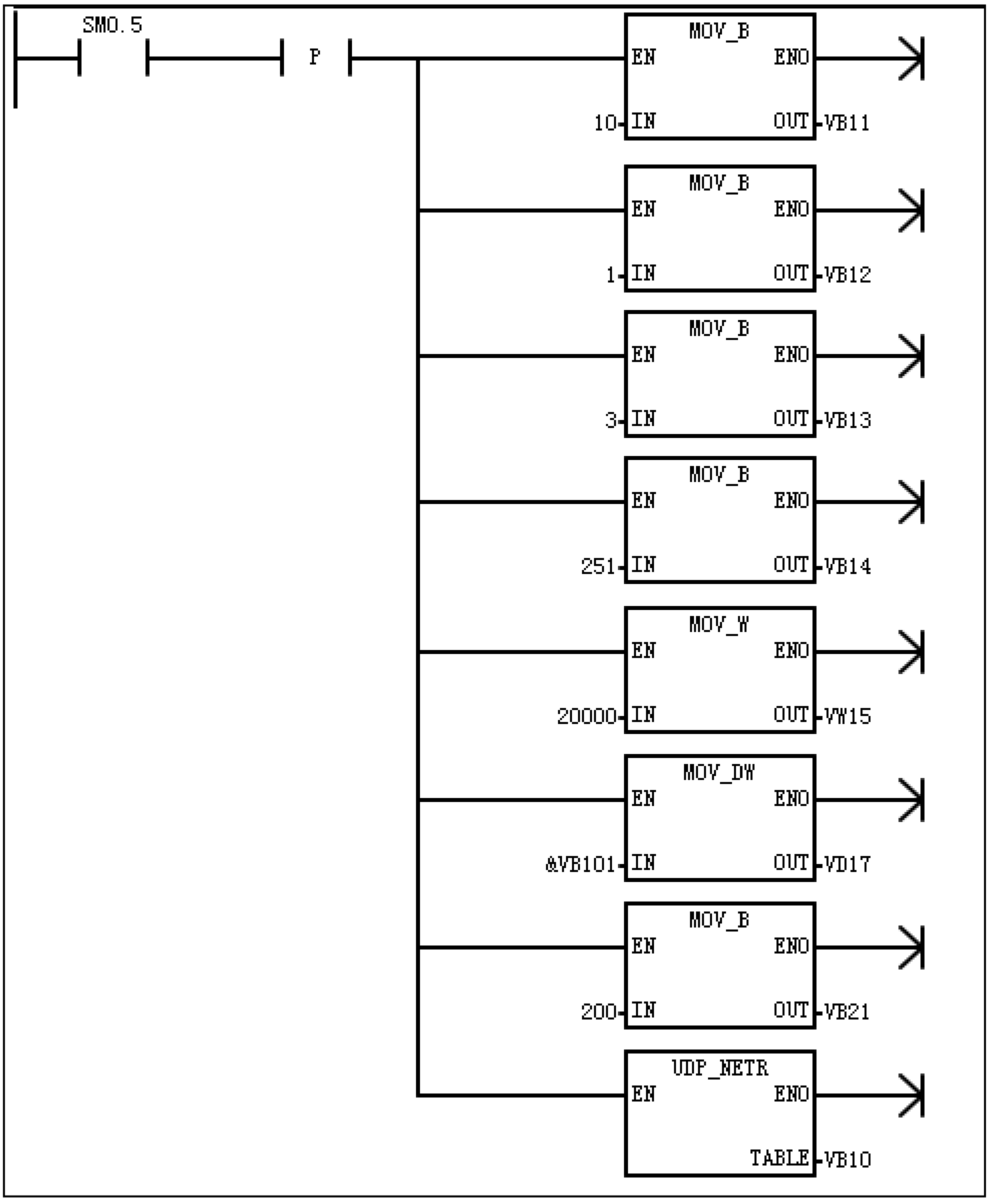
The lower 4 bits of the 1st byte is error code, defined as following:

Error Code	Description
0	No error
1	Timeout Error; no response from remote station
2	Receive error; validation, frame or checksum error in the response
3	Offline error; duplicate address or Conflicts resulted by hardware fault
4	Queue overflow error; activated more than 8 UDP_NETR/UDP_NETW block
6	illegal parameter; UDP_NETR/UDP_NETW table contains an illegal or invalid value
7	No resources; the remote station is busy (uploading or downloading sequences)
8	Error in layer 7; violate the application protocol
9	Information error; Data address error or Data length incorrect

Network1: Write the 200 bytes data of the UDP_PPI master (started from: VB312) into UDP_PPI slave(IP: 10.1.3.251, started from: VB101).

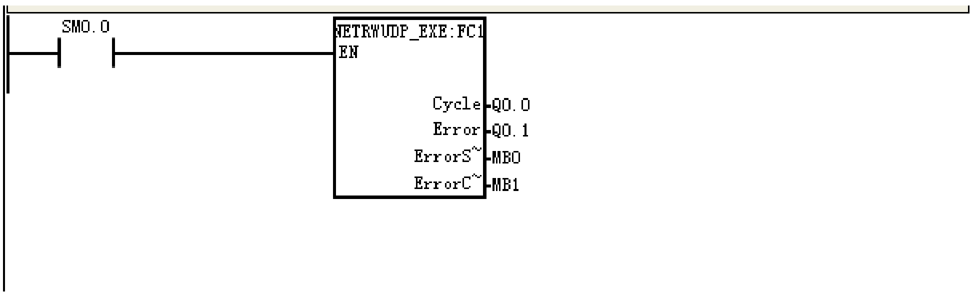


Network2: Read the 200 bytes data from the UDP_PPI slave (IP: 10.1.3.251, started from: VB101) into UDP_PPI master (started from: VB22) .



2) Program the UDP_PPI master by NETR/NETW-UDP instruction Wizard

Select “wizard” in the subordinate options of CTH200, then double click “NETR/W-UDP wizard” to open a dialog box for configuration. After that, call FC in network of OB1:



3）After editing the program, compile and download it into UDP_PPI master (must be called by SM0.0).

Step 4: UDP_PPI master communicates with slave

1) Power up the UDP_PPI master and slave.

2) Connect the EtherNET ports of UDP_PPI master and slave for communication.

6.3.4 UDP_PPI Slave Address Mapping

The normal NETR/NETW instruction is used for UDP_PPI communication, kinds of registers can be configured and the Address Image for slave is direct mapping.

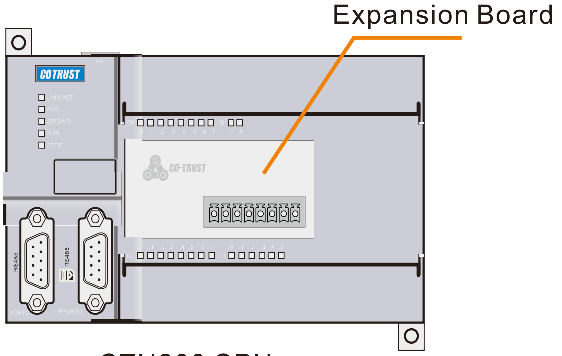
6.4 Analog I/O Expansion Board

CTH200 series CPU can be equipped with Analog I/O Expansion Board. The board embedded in to master CPU via bus connection. It provides collected analog signals, processing results and diagnostic information for CPU by bus interface, then the master CPU will process these data based on specific user program and send Digital data to related expansion modules, which can control the size of analog signals.

There are two methods to access Analog Board: calling dedicated expansion instruction library or Access SM memory directly.

6.4.1 Installation Notes

Dismantling the up cover plate on CPU as shown in the gray box of following figure, align the pins of expansion board to fix it, then cover the up plate, cautions for these operations.



For terminal connections between Analog Expansion Board and CPU, please refer to chapter *4.10.1 Analog Expansion Board Specifications*.

6.4.2 Expansion Board Access Address

The usage for Analog expansion board is different with CAN-01 board which can be used directly by inserting, but it must use dedicated instructions or special SM memory.

- **Access SM area directly**

SMW116~SMW126 of CPU is used for Analog image of expansion board. First with AI, 4 words started from SMW116; then with AQ, 2 words from SMW124.

2AI/1AQ: SMW116 for AIW0, SMW118 for AIW2, SMW124 for AQW0

Function	Image address	Function	Image address

A+ input	SMW116	Module type	SMB114
B+ input	SMW118	Module status	SMB115
VO output	SMW124		

4AI/2AQ: SMW116 for AIW0, SMW118 for AIW2, SMW120 for AIW4, SMW122 for AIW6, SMW124 for AQW0, SMW126 for AQW2

Function	Image address	Function	Image address
A+ input	SMW116	V0/I0 output	SMW124
B+ input	SMW118	V1/I1 output	SMW126
C+ input	SMW120	Module type	SMB114
D+ input	SMW122	Module status	SMB115

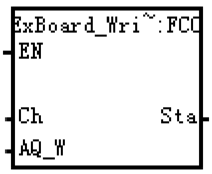
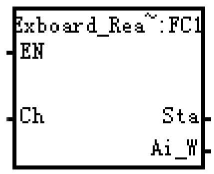
SMB115 Module status defined as in the following table:

Name	Function	Value
Module type	With module	0x1E: 4AI/2AQ 0x19: 2AI/1AQ 0x20: CAN expansion board
	No module	0x00
Module status	Normal	0x00
	Communication error	0x01
	Calibration failed	0x02
	Access error	0xFF

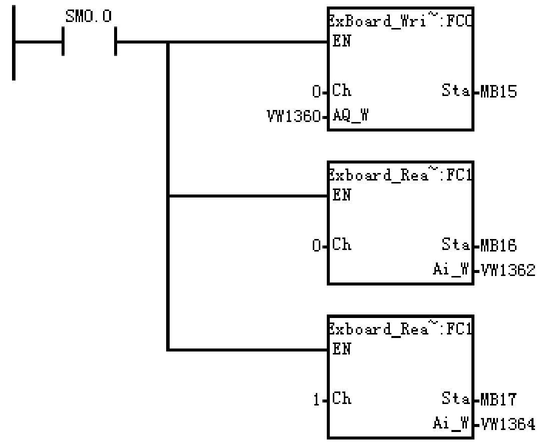
- **Calling expansion instruction library**

Users can use the Magicworks PLC dedicated library Exboard_H200 to access expansion board, you can down load it from our website. For information about how to import this library, refer to appendix *A Using CT-MODBUS Master and Slave Libraries*.

Table 6-6 expansion instructions

Write Access			**Read Access**		
Name: ExBoard_WriteAq			Name: ExBoard_ReadAi		
					
Symbol	variable type	Data type	Symbol	variable type	Data type
Ch	IN	BYTE	Ch	IN	BYTE
AQ_W	IN	WORD	Ai_W	OUT	WORD
Sta	OUT	BYTE	Sta	OUT	BYTE
Comments					
Ch: channel number, value: 0 AQ_W: AQ value Sta: 0 - normal; -1 - error			Ch: channel number, value: 0~1 Ai_W: Ai value Sta: 0 - normal; -1 - error		

Example

<table>
<tr><th>Ladder diagram</th><th>STL and program comments</th></tr>
<tr><td>
</td><td>LD SM0.0
CALL ExBoard_WriteAq, 0, VW1360, MB15
CALL Exboard_ReadAi, 0, MB16, VW1362
CALL Exboard_ReadAi, 1, MB17, VW1364
// enable the board to program
// call the ExBoard_WriteAq, write the AQ of Ch 0, store the status value into MB15
// call the Exboard_ReadAi, read the Ai of Ch 0, store the status value into MB16
// call the Exboard_ReadAi, read the Ai of Ch 1, store the status value into MB17</td></tr>
</table>

6.5 Recipe and Data Log

6.5.1 Recipe

MagicWorks PLC software as well as CPU H224X, H226XL and H228XL support recipe. MagicWorks PLC provides the Recipe Wizard to help you organize define recipes.

All recipes are stored in the memory card instead of the PLC. Therefore, to use the recipe feature, it is a must to insert an optional 64kB or 256kB memory card to the PLC.

A recipe, however, will be read into PLC memory when the user program is dealing with it. There may be, for example, chocolate chip, sugar, and oatmeal for cookies recipes. But only one type of cookies can be made at a time, thus you must select proper recipe to read into PLC memory.

Figure 6-16 illustrates a process for making multiple types of cookies using recipes. Each recipe for cookie is stored in the memory card. The operator selects which type of cookie to make on TP10 text display, and the user program reads that recipe into memory.

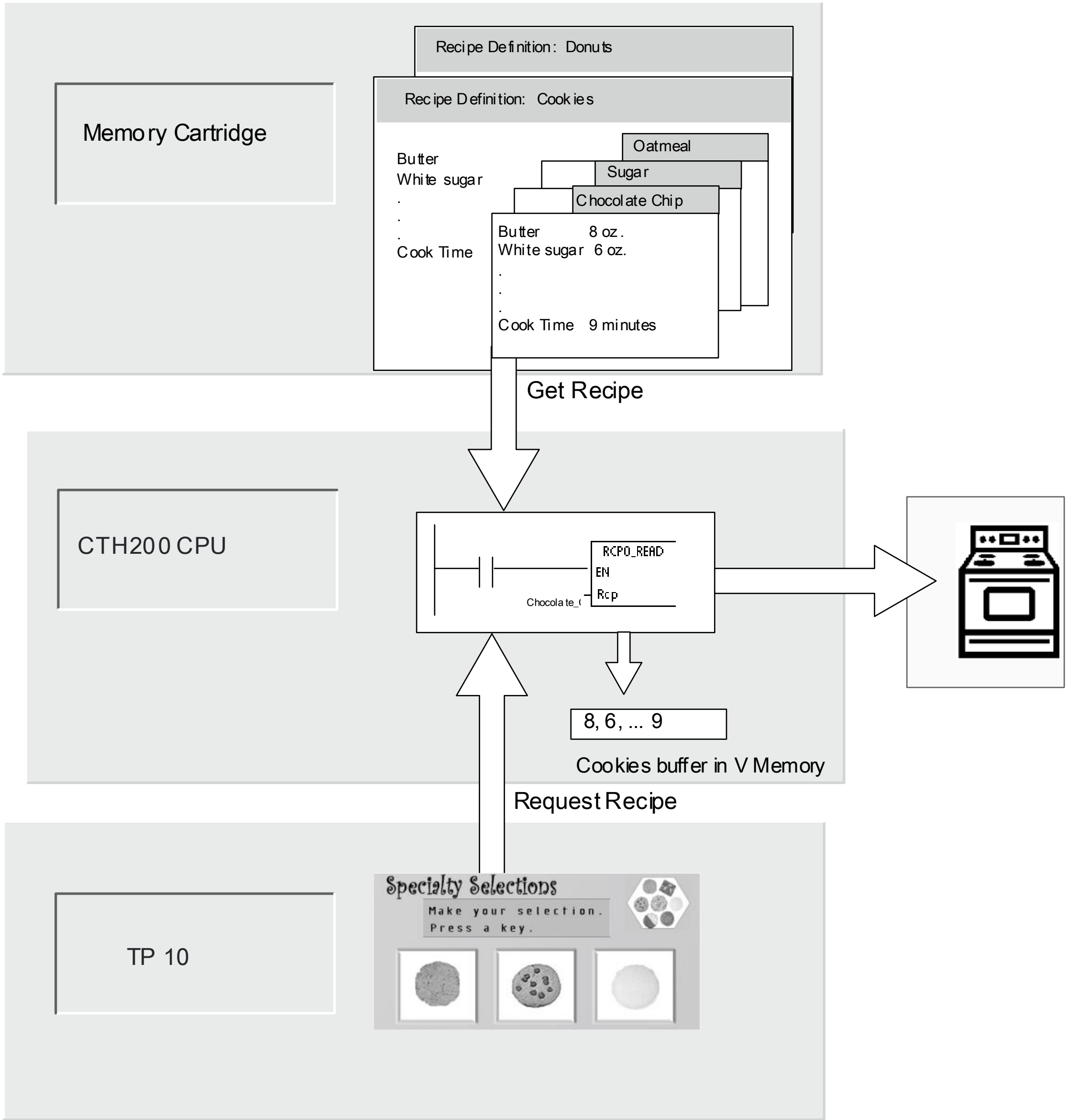


Figure 6-16 Example of Recipe Application

Recipe Definition and Terminology

The following definitions and terms help you understand the Recipe Wizard.

1. Recipe construction is the set of project components generated by the Recipe Wizard, concluding instruction subroutines, data block tabs, and symbol tables.

2. Recipe setting is a collection of recipes that have the same parameters set. However, the parameters value can vary from the recipe.

3. Recipe is the set of parameter values that provides the information of producing a product or controlling a process.

For example, recipe definitions can be different, such as donuts and cookies. And the cookie recipe definition may contain many different recipes, such as chocolate chip and sugar cookies. Example fields and values are shown in the following table.

Example of Recipe Definition -- Cookies

Filed Name	Data Type	Chocolate_Chip (recipe 0)	Sugar (recipe 1)	Comment
Butter	Byte	8	8	Ounces
White_Sugar	Byte	6	12	Ounces
Brown_Sugar	Byte	6	0	Ounces
Eggs	Byte	2	1	each
Vanilla	Byte	1	1	Teaspoon
Flour	Byte	18	32	Ounces
Baking_Soda	Real	1.0	0.5	Teaspoon
Baking_Powder	Real	0	1.0	Teaspoon
Salt	Real	1.0	0.5	Teaspoon
Chocolate_Chips	Real	16	0.0	Ounces
Lemon_Peel	Real	0.0	1.0	Tablespoon
Cook_Time	Real	9.0	10.0	Minutes

Using the Recipe Wizard

Use the Recipe Wizard to create recipes, stored in the memory Card, and its set. Recipes and its definitions can be entered directly in the Recipe Wizard. Later changes to individual recipes can be made by running the Recipe Wizard again or by programming with the RCPx_WRITE instruction subroutine:

1. A symbol table for each recipe definition. Each table includes symbol that has the same name as the recipe field names. These symbols define the V memory addresses to access the recipe values currently loaded in memory. Each table also includes a symbolic constant to reference each recipe.

2. A data block tab for each recipe definition. This tab defines the initial values for each V memory address represented within the symbol table.

3. A RCPx_READ instruction subroutine. This instruction is used to read the specified recipe from the memory card to V memory.

4. A RCPx_WRITE instruction subroutine. This instruction is used to write recipe values from V memory to the memory card.

Defining Recipes

Select the Tools > Recipe Wizard menu command to use it to create a recipe. The first screen is an introductory screen defining the basic operations of the recipe wizard. Click on the Next button to begin configuring your recipes. To create a recipe definition, follow the steps below. See Figure 6-17.

1. Specify the field names for the recipe definition. Each name will become a symbol in your project as previously defined.

2. Select a data type from the drop down list.

3. Input a default value, for all new recipes specified within this definition to begin with, and comment for each name.

4. Click “Next” to create and edit recipes for this recipe definition.

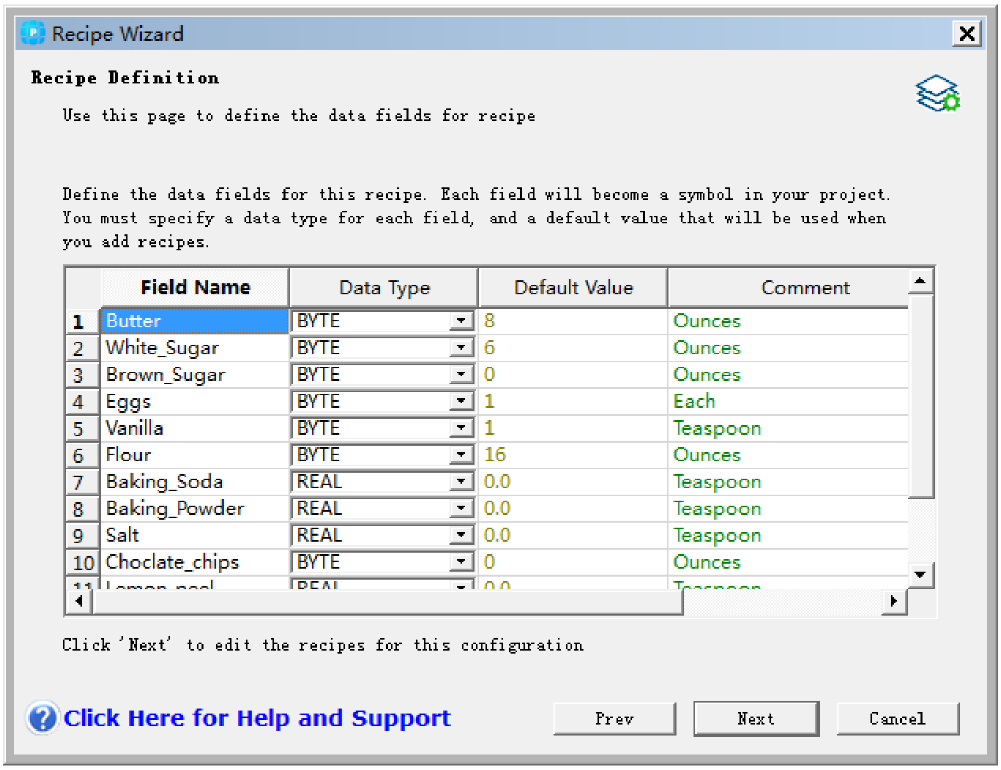


Figure 6-17 Defining Recipes

Use as many rows as necessary to define all data fields in the recipe. You can have up to four different recipe definitions. The number of recipes for each definition is limited only by the available space within the memory card.

Creating and Editing Recipes

Window of creating and editing recipes allows you to create single recipe and allocate value for it. Each editable column represents a unique recipe.

Click “Add Recipe(s)” to add a new one. Each recipe takes the default, which assigned when creating recipe set, as initial value. Another way to create a new recipe is to right click the existing recipe column to choose to paste or insert one.

Each new recipe will be given a default name, in form of DEFx_RCPy, refer to the recipe definition and number.

To create and edit recipes, follow the steps below. See Figure 6-18.

1. Click “Next” to the “Create and Edit Recipe” interface.

2. Select “New” to insert a new recipe.

3. Rename an appropriate name in replace of the default one.

4. Change the values in each recipe data set as needed.

5. Click “Next”.

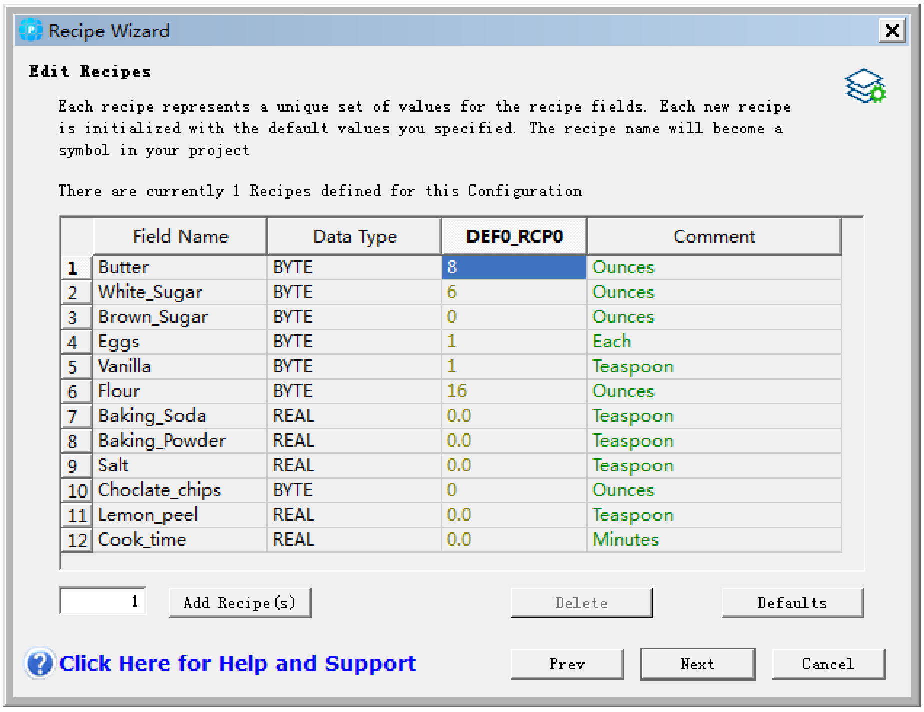


Figure 6-18 Creating and Editing Recipes

Allocating Memory

In the interface of “Allocate Memory” you can assign the V memory’s beginning address where you start to store the recipes read from memory card. You can either select the V memory address yourself or use the one, with correct length as well as unused, recommended by recipe wizard. To allocate memory, follow the steps below.

1. To select the V memory address where you want the recipe to be stored, click the window and enter the address.
2. To let the Recipe Wizard select an unused V memory block of the correct size, click the “Suggest Address”.
3. Click the “Next”.

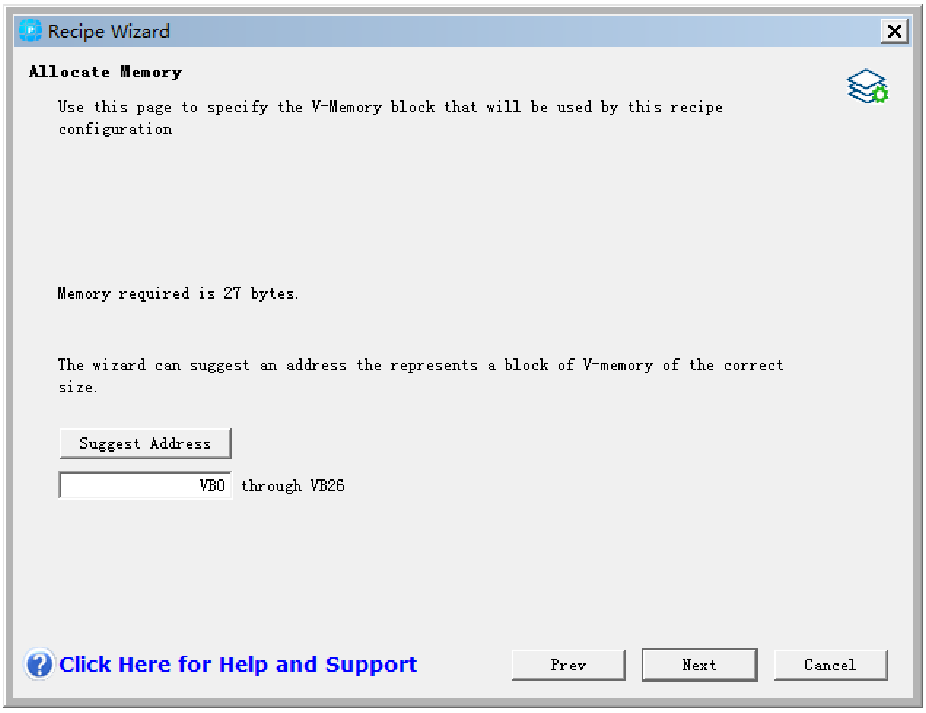


Figure 6-19 Allocating Memory

Project Components

The different components to be added to project are listed in the “Project Components” page as figure 6-20, click “Finish” to add. Each recipe configuration owns a unique name with recipe set name(RCPx) at the end in project tree.

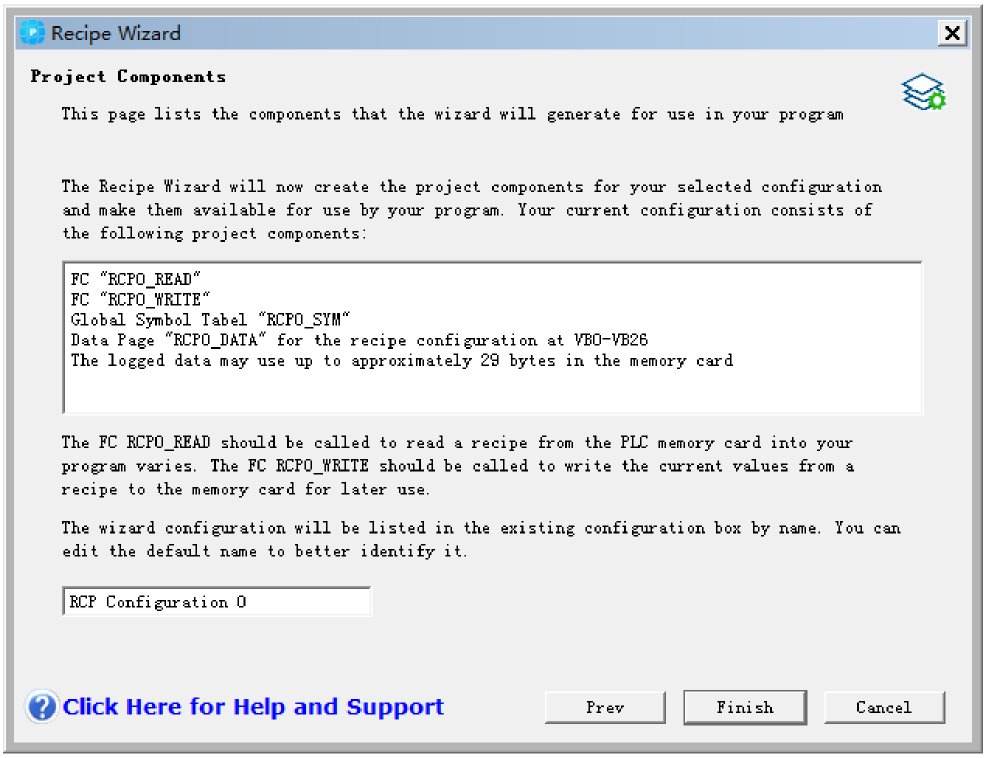
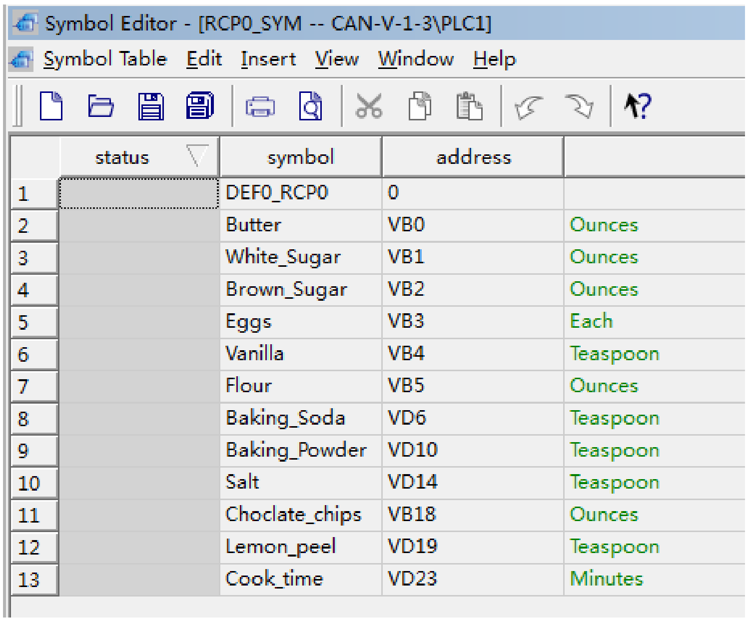


Figure 6-20 Project Components

Using the Symbol Table

Create a symbol table using general value to indicate recipe for each recipe set. Use these symbols to indicate the recipes you want in instruction RCPx_READ and RCPx_WRITE as figure 6-21. As each domain owns a symbol name given by table, you could also visit the recipe value in V memory with these symbols.



	status	symbol	address	
1		DEF0_RCP0	0	
2		Butter	VB0	Ounces
3		White_Sugar	VB1	Ounces
4		Brown_Sugar	VB2	Ounces
5		Eggs	VB3	Each
6		Vanilla	VB4	Teaspoon
7		Flour	VB5	Ounces
8		Baking_Soda	VD6	Teaspoon
9		Baking_Powder	VD10	Teaspoon
10		Salt	VD14	Teaspoon
11		Choclate_chips	VB18	Ounces
12		Lemon_peel	VD19	Teaspoon
13		Cook_time	VD23	Minutes

Figure 6-21 Symbol Table

Downloading the Project with a Recipe Configuration

To download a project which contains a recipe configuration, following the steps below. See Figure 6-22.

1. Select File > Download.
2. Ensure to check the Program Block, Data Block, and Recipes boxes in the dialog.
3. Click the “Download”.

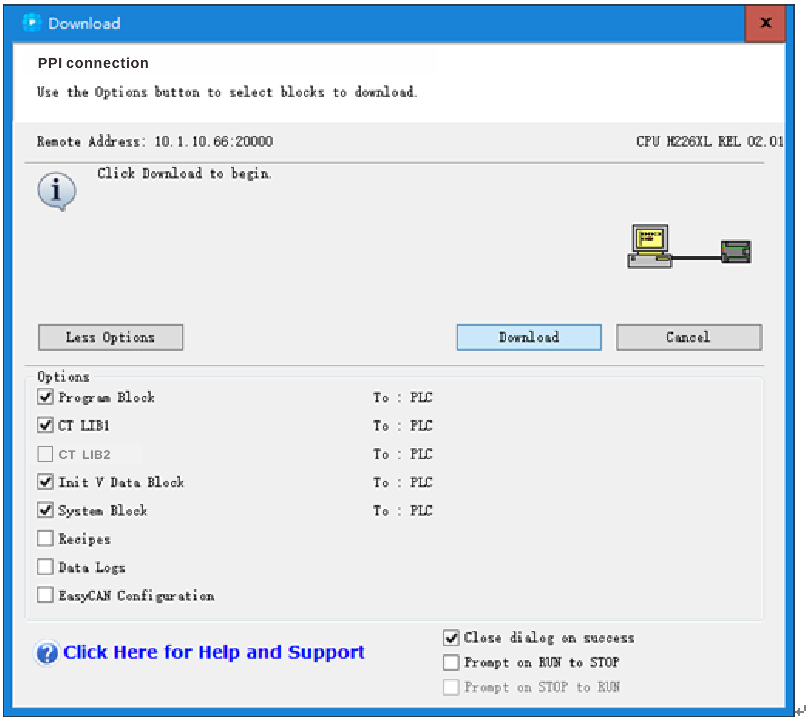


Figure 6-22 download recipe

Delete Existing Recipe Configurations

Edit existing recipe configurations as steps below. See Figure 6-23.

1. Click on the configuration drop down list and select an existing recipe configuration.
2. Click on the Delete Configuration button to delete an existing recipe configuration.

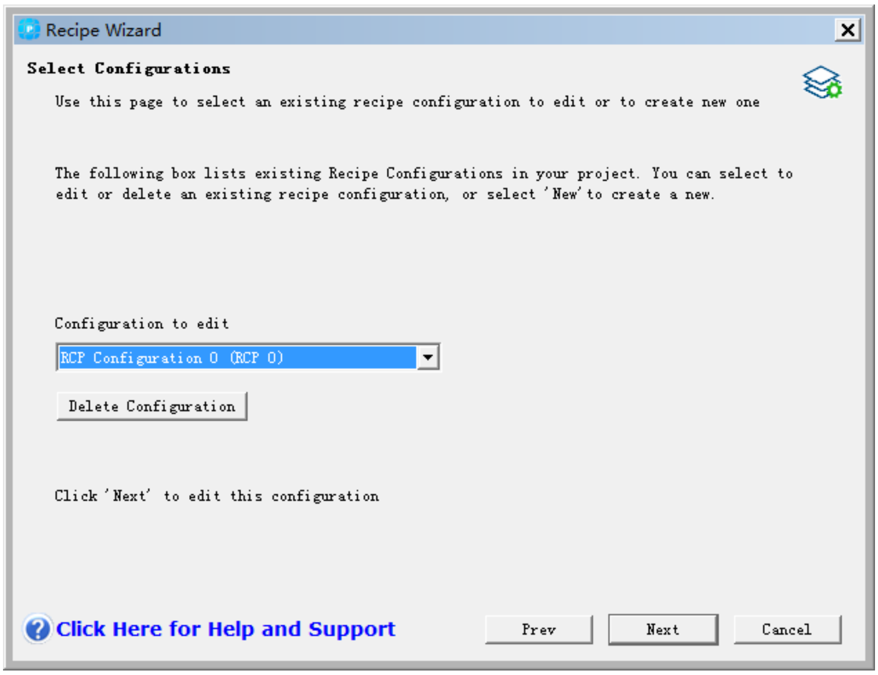


Figure 6-23 Editing recipe

Instructions Created by the Recipe Wizard

RCPx_Read Subroutine

The Subroutine RCPx_READ created by the Recipe Wizard is used to read an individual recipe from the memory card to the specified area in V memory.

The x in the RCPx_READ instruction means the recipe set number of the recipe that you want to read. High level of EN input enables the execution of the instruction. And the Rcp input identifies which recipe to be loaded from the memory card. The Error output returns the result of the instruction execution. See Table 6-8 for definitions of the error codes.

RCPx_Write Subroutine

The Subroutine RCPx_WRITE created by the Recipe Wizard is used to replace a recipe in the memory card with the contents of the recipe contained in V memory.

The x in the RCPx_WRITE instruction means the recipe set number of the recipe that you want to replace. High level of the EN input enables the execution of the instruction. And the Rcp input identifies the recipe that will be replaced in the memory card.
The Error output returns the result of the execution of this instruction. See Table 6-8 for definitions of the error codes.

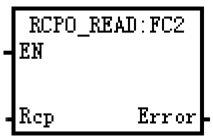
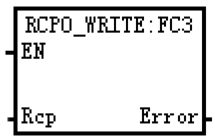
RCPx_Read Subroutine	RCPx_Write Subroutine
	

Table 6-7 Valid Operands for the Recipe Subroutine

Input/Output	Data Type	Operands
Rcp	word	VW, IW,QW, MW, SW, SMW, LW, AC, *VD, *AC, *LD, constant
Error	byte	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

Table 6-8 Error Codes of the Recipe Instructions

Error code	Description
0	No Error
132	Access to the memory card failed

Notice

The EEPROM used in the memory card will support a limited number of write operations. Typically one million write cycles. Once reach the limit, the EEPROM will not operate properly.

Thus make sure that you do not enable the RCPx_WRITE instruction each scan. Or the memory card will wear out in a relatively short period of time.

6.5.2 Data Log

MagicWorks PLC provides the Data Log Wizard to store process measurement data into memory card. Moving process data to the memory card frees V memory addresses that would otherwise be required to store this data

With this feature, you can permanently reserve the process data log, contains a time date stamp, in program control. Up to 4 independent data log, whose format can be defined by new data record wizard, can be configured.

All data logs are stored in the memory card, to use it, insert an optional 64K or 256K memory card in your PLC. See Appendix O Battery Card and Memory Card for information about the memory cards.

Use the S7-200 Explorer to upload the contents of your data logs to your computer. An example of a Data Log application is shown as Figure 6-24.

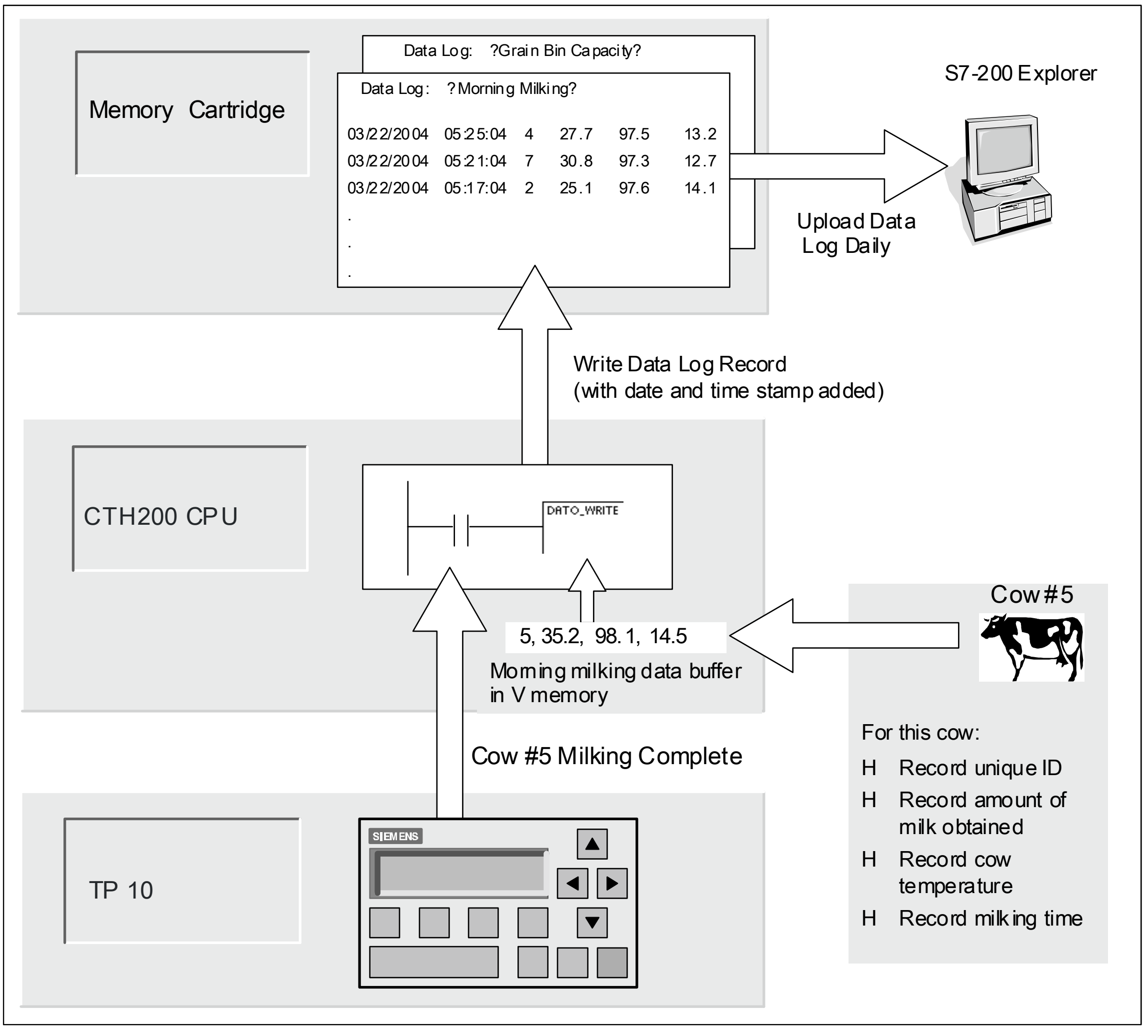


Figure 6-24 Example of Data Log Application

Definition and Terminology of Data Log

The following definitions and terms explain you the Data Log Wizard:

1. A data log is a group of records usually ordered by date and time. Each record represents

some process events that record a set of process data. The organization of this data is defined by the data log wizard.

2. A data log record is a single row of data written into data log.

Using the Data Log Wizard

The Data Log Wizard can configure four data logs at most, which is used to:

1. Define the format of the data log record
2. Select data log'options such as time stamp, date stamp, and clear data log on upload
3. Specify the maximum number of records that can be stored in the data log
4. Create project code that is used to store records in the data log.

Steps using Data Log Wizard to create a data log:

1. Create a symbol table for each data log which includes symbol names same as that of data log field. Each symbol defines the V memory addresses that store the current data log. Each table also includes a symbolic constant to recognize each data log.

2. A data block tab for each data log record that assigns V memory addresses for each data log field. Your program uses these V memory addresses to accumulate the current log data set.

3. Create a DATx_WRITE subroutine to copy the specified data log from V memory to the memory card. Each execution adds a new data record to the log stored in the memory card.

Data Log Options

You can configure the following options for the data log. See Figure 6-25.

Time Stamp

You can have a Time Stamp in each data log record. Select this option, the CPU will automatically add a time stamp each record when the user program write a data log.

Data Stamp

You can add a Date Stamp to each data log record. When selected, the CPU automatically includes a date stamp with each record when the user program commands a data log write.

Clear Data Log

You can clear all the record in data log after uploading the data. the data log will be cleared each time it is uploaded if you selected this.

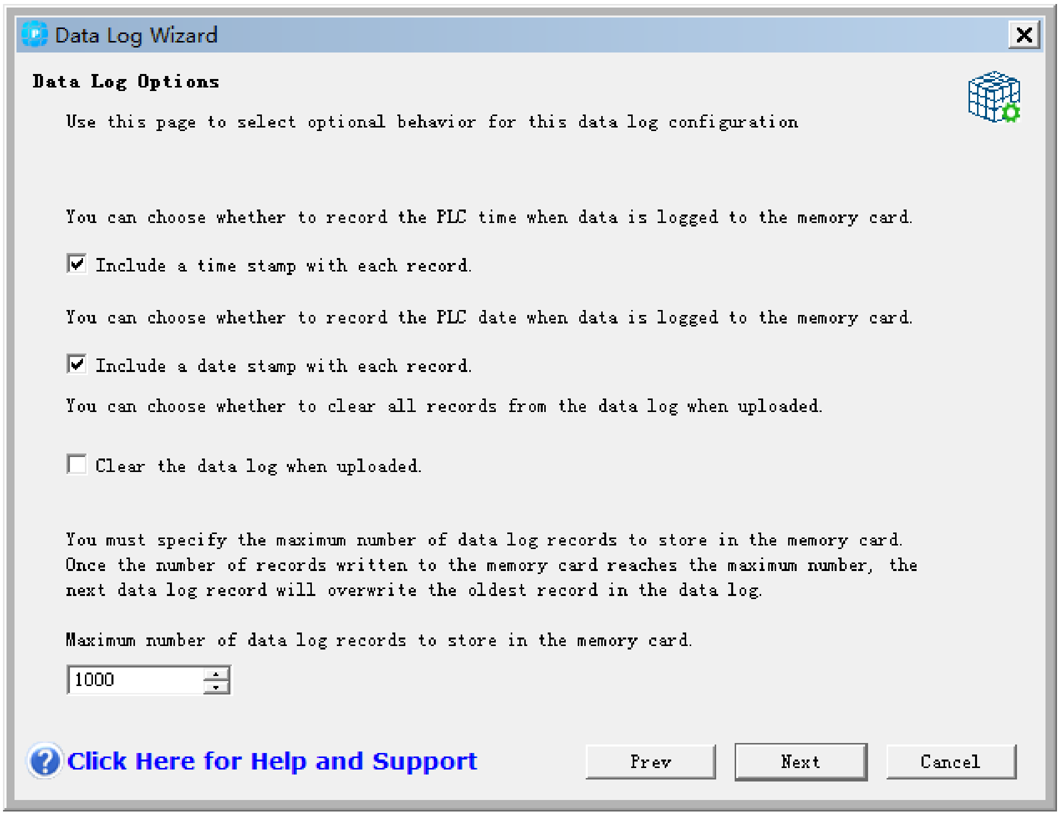


Figure 6-25 Data Log Options

Data log is a circular queue (when full, a new record will replace the oldest one). You must specify the maximum number of records, which is 65534 and default as 1000, stored in the data log.

Defining the Data Log

Assign domain, as a symbol in project, as well as its data format for data log. Each data log record contains 4 ~203 bytes of data. Follow the steps below to define the data field. See Figure 6-26.

1. Click “Field Name” to enter the name, which is the symbol referenced by the user program.

2. Click “Data Type” cell and select a type from the drop down list.

3. Click “Comment” to enter a comment,

4. Use as many rows as necessary to define a record.

5. Click “Next”.

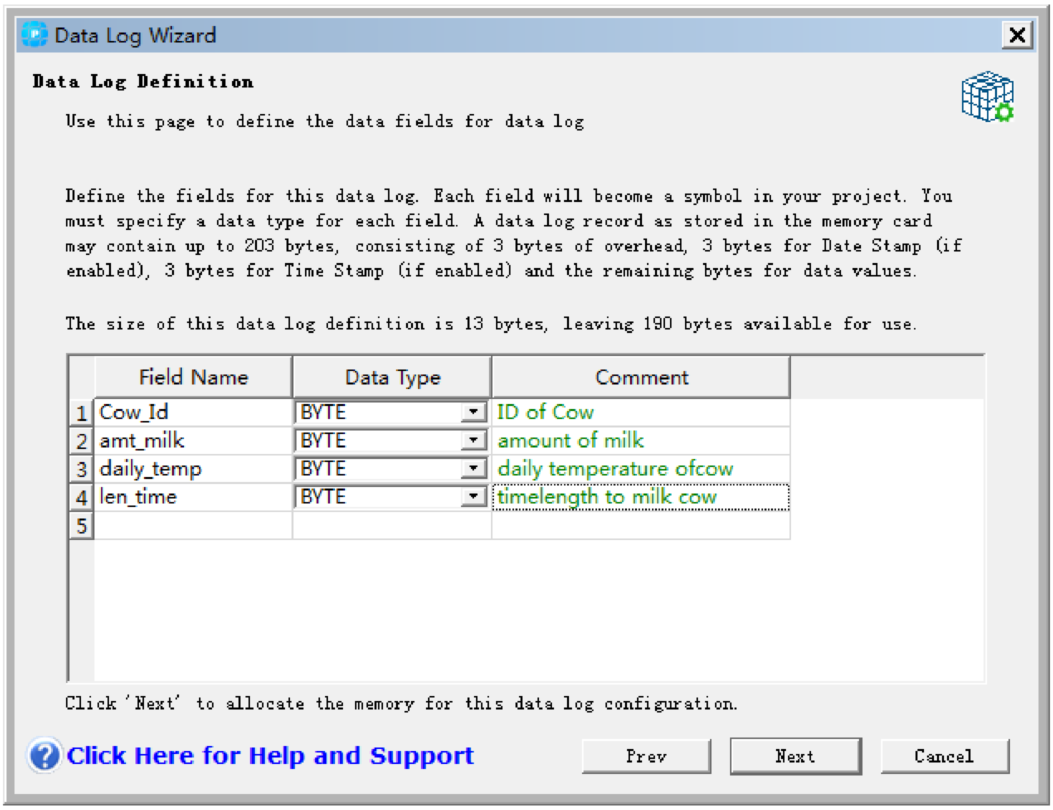


Figure 6-26 Define the Data Log

Allocating Memory area

The data log wizard creates a block, a storage address where data log record stored before write into memory card, in the V memory area of PLC. Assign beginning address for the V memory area where you put configuration. You can either select the V memory area address yourself or use the one, with correct length and used, recommended by wizard. The size of the block varies from your choices in the Data Log wizard. See Figure 6-27.

Follow the steps to allocate memory area:

1. To select the V memory address where the data log record will be constructed, click “Suggested Address” to enter the address.

2. For the Data Log Wizard selecting an unused V memory block of the correct size, click the “Suggested Address”.

3. Click “Next”.

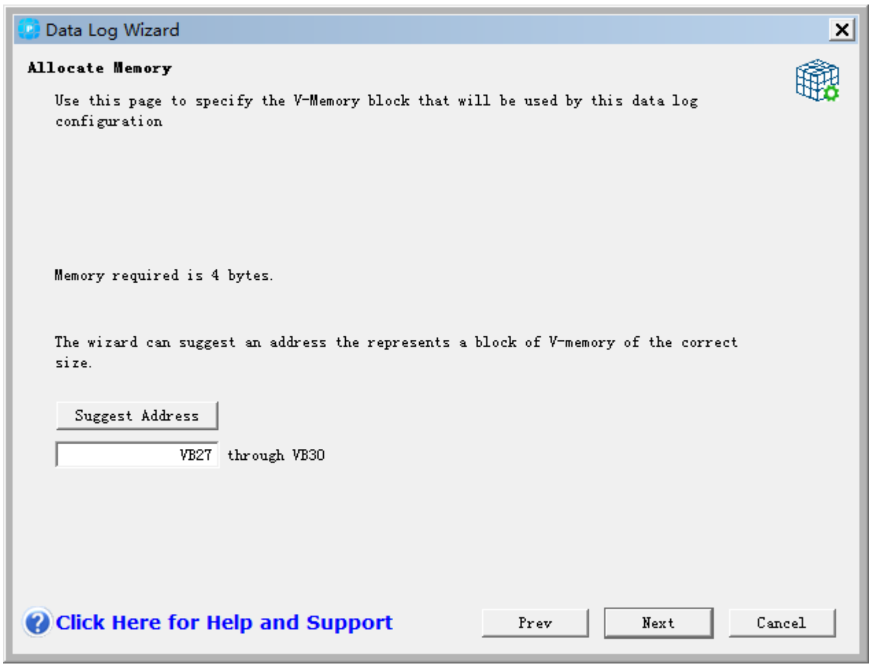


Figure 6-27 Allocate Memory

Project Components

Different components to be added into your project are listed in "Project Components", as figure 6-28. Click "Finish" to complete the data log wizard and add these components to project. Each data log construction owns a unique name with data log set(DATx) attached at the end.

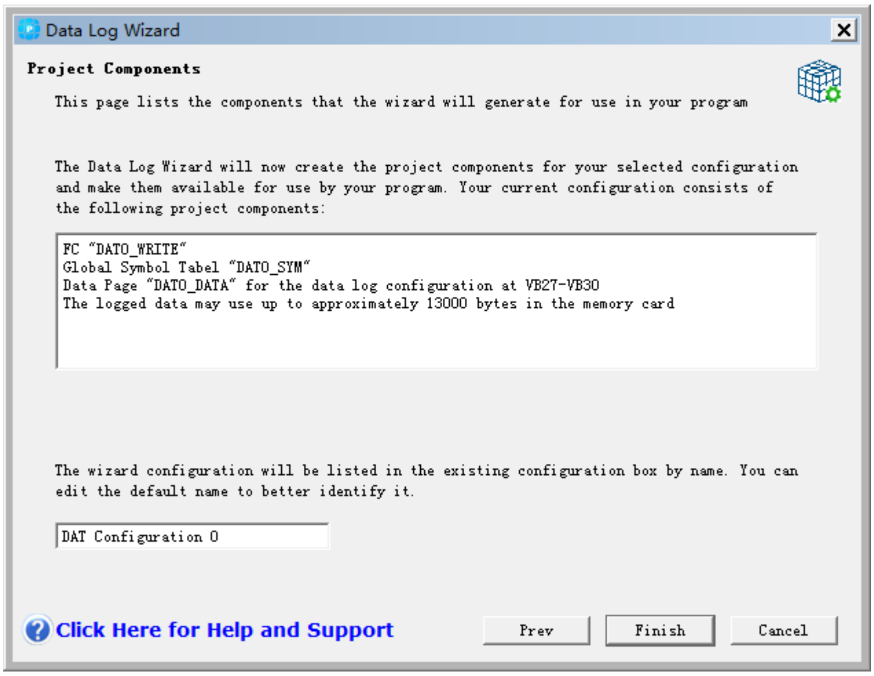


Figure 6-28 Project components

Symbol Table

Create a symbol table, which can be used in DATx_WRITE as well as to visit the value of data log in V memory, for each data log and define some constant to indicate it. Each table create symbol name for the domain in data log.

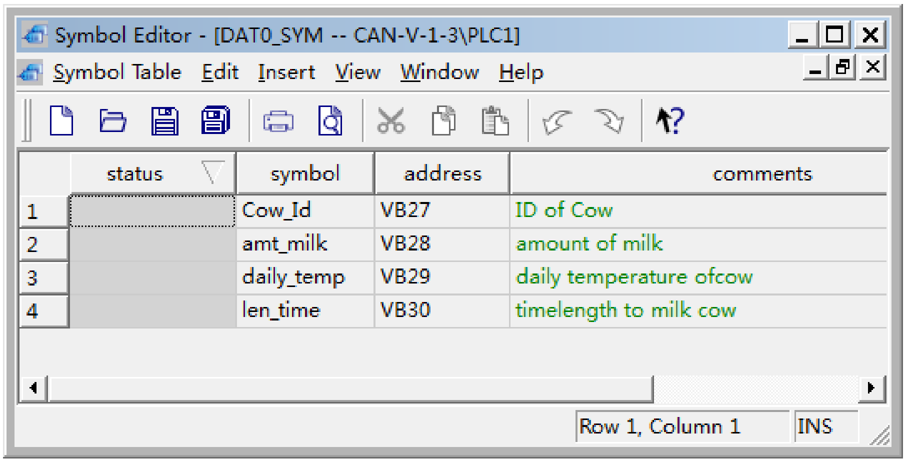


Figure 6-29 Symbol Table

Downloading a Project with Data Log Configuration

Download a project with data log to CPU before using the record. If a project has a data log configuration, the download window has the option of “Data Log” checked by default.

Notice

When you download a project with data log, the existing log in the memory card will lost.

To download a project that contains data log configurations, follow steps below. See Figure 6-29.

1. Select File > Download.
2. Ensure the “Data Logs” checked in dialog as figure below.
3. Click “Download”.

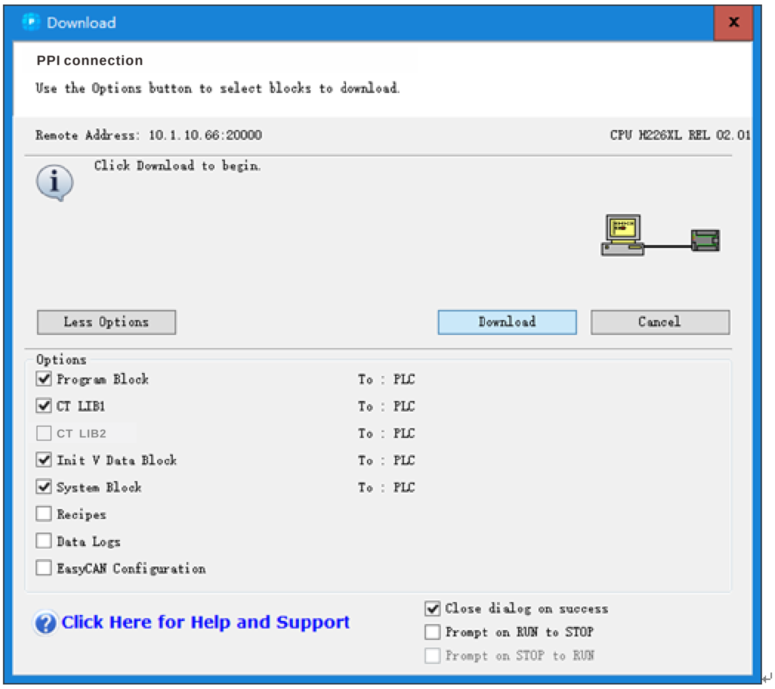


Figure 6-29 download data logs

Using the S7-200 Explorer

The S7-200 Explorer is used to read data log from the memory card, and then store it in a Comma separated Values (CSV) file. Each read create a new file, format as PLC address, data log name, date and time, to store into data log catalogue. You can decide, by right click to choose, whether to automatically start the application relevant to CSV extension name when data log has been read successfully. The catalogue of data log would be assigned in the process of installation. The default catalogue would be c:\programfiles\siemens\Microsystems with STEP 7 installed or c:\siemens\Microsystems if not.

To read a data log, follow the steps below:

1. Open Windows Explorer to find My S7-200 Network.

2. Select the My S7-200 Network folder.

3. Select the correct S7-200 PLC folder.

4. Select the memory card folder.

5. Find the correct data log configuration files to name as DAT Configuration x (DATx).

6. Right click the context menu, choose “Upload”.

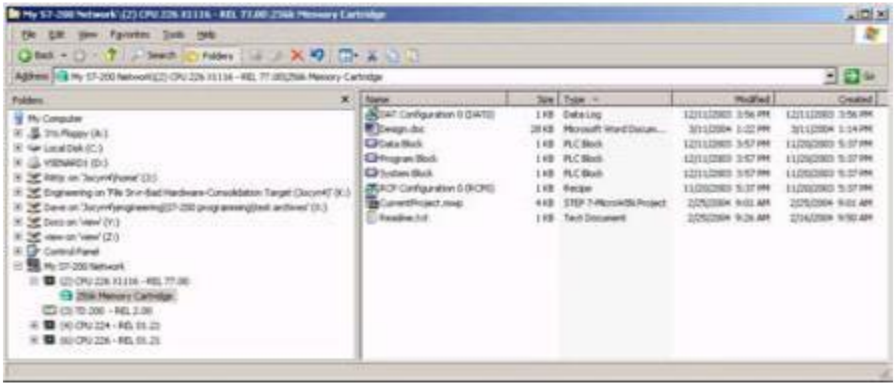


Figure 6-30 using S7--200 Explorer

Edit Existing Data Log Configuration

Steps to edit existing data log shows as below:

1. Click the dropdown list to select an existing data log configuration as shown in Figure 6-31.

2. Click “Delete Configuration” to delete an existing data log configuration. You can have up to four different data logs.

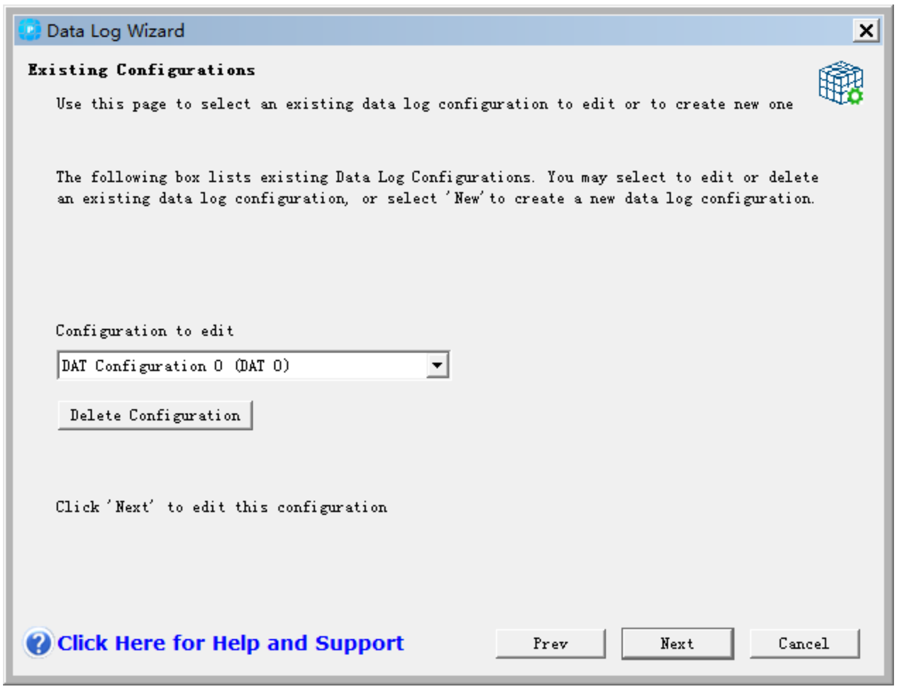


Figure 6-31 Edit Existing Data Log Configurations

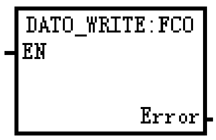
Instruction Created by the Data Log Wizard

The Data Log Wizard will add a subroutine Instruction in your project.

DATx_WRITE Subroutine

The subroutine DATx_WRITE is used to write current value of data log domain into memory card. Picture below shows this subroutine.

DATx_WRITE subroutine



Return Error 132 when the instruction fails to access the memory card.

Table 6-9 Parameters of the DATAx_WRITE Subroutine

Inputs/Outputs	Data Type	Description
Error	Byte	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

Notice

The times for write operation, typically 1m times, have limitation, whose overrange would result in ineffective EEPROM. So please ensure not to execute DATx_WRITE instruction every program cycle or the card would be damaged immediately.

7 Power Budget

CTH200 series CPU has an internal power supply for the unit and the expansion modules. It also provides 5VDC logic power supply, which can be used for power supply of any expansion in the system. Pay close attention to the system configuration to ensure that the CPU can provide the 5V power for the selected expansion modules. If the configuration requires more power than the CPU provides, some expansion modules must be removed or select CPU that can provide more power. See Chapter *7.2 5VDC Power* for detailed information on CTH200 CPU's 5VDC logic budget and expansion module 5VDC power requirements.

CTH200 series CPU also provide 24VDC sensor power supply (except PSC266), which can provide 24VDC for input, relay coil on expansion module or other power supply. If the required power exceeds budget, an external 24VDC power supply must be added. For specific CTH200 CPU 24VDC sensor power budget, refer to Chapter *7.3 24VDC Power*. If you need an external 24VDC power supply, make sure that the power supply is not connected in parallel to the sensor power supply of the CTH200 series CPU. To improve electrical noise protection, please connect to the common terminal (M) of different power sources.

7.1 Power Requirements

CPU provides 5VDC and 24VDC power supply itself:

Each CPU has an external 24VDC power supply, which can provide 24VDC for input and relay coil of the expansion module.

When expansion unit connect, CPU provides 5VDC power. If the 5VDC requirement of the extension unit exceeds the CPU rating power output, the extension unit must be unloaded until the demand is within the predetermined power supply. Chapter *4 Technical Specifications* provides information on CPU power budget and expansion module power requirements.

Note
If 5VDC power exceeds CPU power budget, it may unable to connect maximum modules the CPU allowed.

Notice
An external 24VDC power supply connected in parallel to CTH200 DC sensor power supply can cause a conflict because each power supply seeks to establish its own potential output. This conflict will shorten the service life of one or two power supplies or cause an immediate failure, then uncertain operation of the PLC system. The unpredictable operation may cause personal injury and equipment damage.

CTH200 CPU's 24VDC sensor and any external power supply should be supplied at different points. A separate connection for utility power is allowed.

7.2 5VDC Power

Table 7-1 5VDC power consumption table

CPU and DP module	5VDC current offered for expansion IO (mA)	Expansion module	5VDC current (mA)
CPU H224/H224X H226L/H226XL/H228XL SM277B/SM277C SM277PN	660 660 660 1100	SM 221-1BF	57
		SM 221-1BH	79
		SM 221-1BL	179
		SM 222-1BF	57
		SM 222-1BH	79
		SM 222-1BL	174
		SM 222-1HF	68
		SM 222-1HH	115
		SM 223-1BF	57
		SM 223-1BH	73
		SM 223-1BL	115
		SM 223-1HF	58
		SM 223-1PH	89
		SM 223-1PL	150
		SM 231-0HC	87
		SM 231-0HF	87
		SM 231-1HF	87
		SM 231-5HF	87
		SM 232-0HB	87
		SM 232-0HD	87
		SM 235-0KD	87
		SM 231-7PB	87
		SM 231-7PC	87
		SM 231-7PD	87
		SM 231-7PF	87
		SM 231-7TD	87
		SM 231-7TF	87
		SM 231-7HF	87
		SM 231-7ND	87
		SM 231-7NF	87
		SM 231-7WA	140
		SM 277A	70
		SM 253 motion control module	90

Calculate max I/O configuration of CTH200 system according to the above table, refer to Chapter *7.4 Power Calculation Example*

7.3 24VDC Power

Table 7-2 24VDC power consumption table

24VDC power supply (mA)		Expansion module 24VDC current consumption (mA)	
CPU H224/H224X CPU H226L/H226XL CPU H228XL SM277B	300 300 300 400	SM222-1BL	54
		SM222-1HF	80
		SM222-1BF	14
		SM222-1BH	22
		SM222-1HH	159
		SM223-1BF	10
		SM223-1BH	11
		SM223-1BL	22
		SM223-1HF	40
		SM223-1PH	80
		SM223-1PL	159
		SM231-0HC	17
		SM231-0HF	31
		SM231-1HF	30
		SM231-5HF	31
		SM232-0HD	112
		SM232-0HB	61
		SM235-0KD	48
		SM231-7PB	34
		SM231-7PC	37
		SM231-7PD	130
		SM231-7PF	30
		SM231-7ND	60
		SM231-7NF	33
		SM231-7HF	37
		SM231-7TD	34
		SM231-7TF	39
		SM 231-7WA	100
		SM277A	70
		SM253 motion control module	80

7.4 Power Calculation Example

Use following design methods to determine how many power configurations can be provided by the ontology unit power supply.

Table 7-3 shows power calculation of CTH200 PLC, including:

- CPU H224X
- SM223 module, and SM223-1PH32 has 8DC inputs/8 relay outputs
- SM221 module, and SM221-1BF32 has 8DC inputs
- The device has 62 inputs and 42 outputs

In this example, the CPU provides enough 5VDC current for the expansion module, but not enough 24VDC current for all input and expansion relay coils from the sensor. I/O requires 536mA while CTH200 CPU only provides 280mA. The device requires at least 236mA of 24VDC additional power to operate all contained 24VDC inputs and outputs.

Table7-3 Power budget table

CPU power budget	5VDC	24VDC
Current CPU H224X offered	660 mA	300 mA
Minus		
System requirement	5VDC	24VDC
CPU H224X, 14 inputs		14* 4 mA= 56 mA
4 SM 223-1PH32, 5V current	4*89 mA= 356mA	
2 SM 221-1BF32, 5V current	2*57 mA= 114 mA	
4 SM 223-1PH32, 8 inputs each		4*8*4 mA= 128 mA
4 SM 223-1PH32, 8 relay coil each		4*8*9 mA= 288 mA
2 SM 221-1BF32, 8 inputs each		2*8*4 mA= 64 mA
Total demand power consumption	470 mA	536 mA
Equal		
Current balance budget	5VDC	24 VDC
Total current balance	190 mA	-236 mA

8 Fault Diagnosis

CPU execute the following operations when faulted:

1）Get into STOP Mode

2）Light the SF/DIAG (Red) LED and STOP indicator up

3）Disconnect the outputs

Check the following conditions first:

- CTH200 CPU and expansion modules are powered normally.
- I/O terminals for CTH200 CPU and expansion modules are fastened by using screws and connector.
- Check the communication cable connect normally.
- Adjust the Baud rate, Port or IP address if you still cannot find the PLC.

Except methods above, you can also read the diagnostic information from MagicWorks PLC, or inspect the LED indicators for PLC internal and external exception.

8.1 Diagnose by MagicWorks PLC

The faulted status would last until the error cleared, then open MagicWorks PLC software → double-click the project interface → select menu item“PLC”→“Information” to check the error information.

Table 8-1 Diagnosis function

Supported event type	**Code and Description**	
	Code	**Event Description**
CPU nonfatal error	0x00	No fatal errors present; no error
	0x01	HSC box enabled before executing HDEF box
	0x02	Conflicting assignment of input interrupt to a point already assigned to a HSC
	0x03	Conflicting assignment of inputs to an HSC already assigned to input interrupt or other HSC
	0x04	Attempted execution of an instruction that is not allowed in an interrupt routine
	0x05	Attempted execution of a second HSC/PLS with the same number before completing the first (HSC/PLS in an interrupt routine conflicts with HSC/PLS in main program)
	0x06	Indirect addressing error
	0x07	TODW (Time-of-Day Write) or TODR (Time-of-Day Read) data error

	0x08	Maximum user subroutine nesting level exceeded
	0x09	Simultaneous execution of XMT/RCV instructions on Port 0
	0x0A	Attempt to redefine a HSC by executing another HDEF instruction for the same HSC
	0x0B	Simultaneous execution of XMT/RCV instructions on Port 1
	0x0C	Reserved
	0x0D	Reserved
	0x0E	Reserved
	0x0F	Illegal numeric value in compare contact instruction
	0x10	Reserved
	0x11	Reserved
	0x12	Reserved
	0x13	Illegal PID loop table
	0x80	Program too large to compile; reduce program size
	0x81	Reserved
	0x82	Illegal instruction; check instruction mnemonics.
	0x83	Missing MEND or instruction not allowed in main program: add MEND instruction, or remove
	0x85	Missing FOR; add FOR instruction or delete NEXT instruction.
	0x86	Missing NEXT; add NEXT instruction or delete FOR instruction.
	0x87	Missing label (LBL, INT, SBR); add the appropriate label.
	0x88	Missing RET or instruction not allowed in a subroutine: add RET to the end of the subroutine or remove incorrect instruction
	0x89	Missing RETI or instruction not allowed in an interrupt routine: add RETI to the end of the interrupt routine or remove incorrect instruction.
	0x8B	Illegal JMP to or from an SCR segment
	0x8C	Duplicate label (LBL, INT, SBR); rename one of the labels.
	0x8D	Illegal label (LBL, INT, SBR); ensure the number of labels allowed was not exceeded.
	0x90	Illegal parameter; verify the allowed parameters for the instruction.
	0x91	Range error (with address information); check the operand ranges.
	0x92	Error in the count field of an instruction (with count information); verify the maximum count
	0x93	Size
	0x94	FOR/NEXT nesting level exceeded.
	0x95	Missing LSCR instruction (Load SCR)
	0x96	Missing SCRE instruction (SCR End) or disallowed instruction before the SCRE instruction
	0x97	User program contains both unnumbered and numbered EV/ED instructions
	0x98	Illegal edit in RUN mode (edit attempted on program with unnumbered EV/ED instructions)

	0x99	Too many hidden program segments (HIDE instructions)
	0x9A	Attempt to switch to Freeport mode while in a user interrupt
	0x9B	Illegal index (string operation in which a starting position value of 0 is specified)
CPU fatal error	0x00	No fatal errors present
	0x01	Reserved
	0x02	Reserved
	0x03	Scan watchdog time-out error
	0x04	Reserved
	0x05	Reserved
	0x06	Reserved
	0x07	Reserved
	0x08	Reserved
	0x09	Reserved
	0x0A	Reserved
	0x0B	Reserved
	0x0C	Reserved
	0x0D	Reserved
	0x0E	Reserved
	0x0F	Reserved
	0x10	Internal software error
	0x11	Compare contact indirect addressing error
	0x12	Compare contact illegal floating point value
	0x13	Reserved
	0x14	Program is not understood by this S7-200
	0x15	Compare contact range error
Log once for each scanning cycle		
Diagnose events Refer to table 8-2	0x00	No fault
	0x01	Module is busy
	0x02	Module time-out with no response
	0x03	Module type unmatched
	0x04	Module version unmatched
	0x05	Software error
	0x06	Module waiting flag is time-out
	0x07	Bus ACK error
	0x08	Bus CRC validation error
	0x10	Memory shift outrange
	0x11	Module not ready
	0x12	Module configuration error
	0x13	Module not support this instruction
	0x15	Module internal diagnose
	0x16	Module has no power

Table 8-2 Special memory diagnosis

SMB8	Module 1 flag register
SMB9	Module 1 error register
SMB10	Module 2 flag register
SMB11	Module 2 error register
SMB12	Module 3 flag register
SMB13	Module 3 error register
SMB14	Module 4 flag register
SMB15	Module 4 error register
SMB16	Module 5 flag register
SMB17	Module 5 error register
SMB18	Module 6 flag register
SMB19	Module 6 error register
SMB20	Module 7 flag register
SMB21	Module 7 error register
SMB200~SMB549	Status for intelligent module

8.2 Diagnose by CTH200 CPU Modules

Table 8-4 Description for CTH200 LED indicators

Indicator	Color	Description
SF/DIAG	Red	ON: system fault, OFF: no fault
RMC	Green	ON: CPU communicates with remote server successfully (Ethernet port configured correctly) OFF: communication for remote server is failed or forbidden (communication access controlled by DIP switch)
RUN	Green	ON: system operation, OFF: system stop
STOP	Orange	ON: system stop, OFF: system operation
LINK/ACT	Green	ON: connected, FLASH: Transmission, OFF: Disconnected

<Note>: STOP and SF LED would light on synchronously started from the beginning of power lose to power down, and the system would log an event.

Appendix

A Using CT-MODBUS Master and Slave Libraries

A.1 CT_MODBUS library

4 libraries in all, which is master and slave libraries for PORT0 and PORT1.

CT_MODBUS function mainly used for Siemens CPU program and data. CT_MODBUS function block is built-into the CPU, not require data space, provided for user as a set of functions.

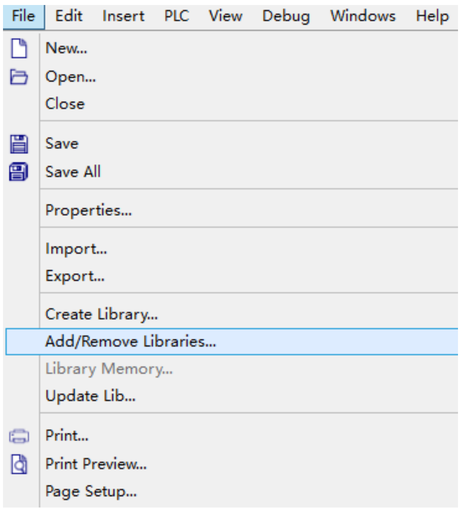
Notice

To get free libraries, visit at http://www.co-trust.com

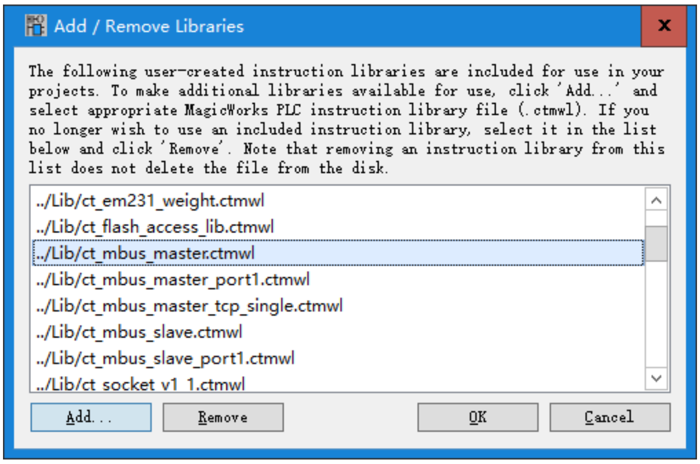
A.2 Installation Instruction

【Add library file】

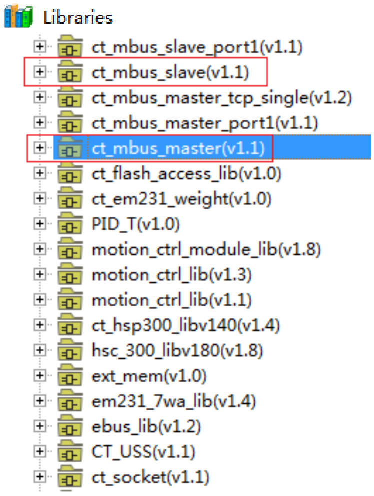
In the "File"--"Add/Delete Library", locate the file "ct_mbus_master.mwl"&"ct_mbus_slave.mwl", as shown in the below figure:



Click "Add" after you find "ct_mbus_master.mwl" and "ct_mbus_slave.mwl"

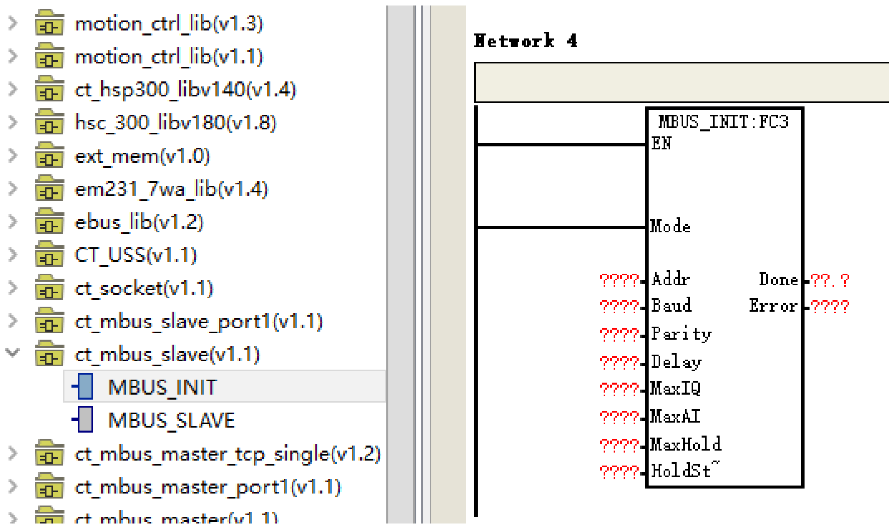


You can find CT_MBUS_MASTER and CT_MBUS_SLAVE library under “libraries” of directory tree after installation:



【Call CT_MODBUS library】

Click “Network” you need to add function block, double click “MBUS_INIT”, “MBUS_SLAVE”, “MBUS_CTRL”, “MBUS_MASTER” in “libraries”, the corresponding function block will shows in “Network”:



A.3 CT_MODBUS Library Function Explanation

【Modbus address】

Modbus addresses are typically 5 - or 6-character values containing data type and offset. The first one or two characters determine the data type, and the last four characters match the data type. The Modbus master station maps this address to correct function.

Modbus slave station instruction supports the following addresses: 00001 to 00128 are the actual output, corresponding to Q0.0--Q15.7; 10001 to 10128 are actual inputs corresponding to I0.0 -- I15.7; 30001 to 30032 are analog input registers corresponding to AIW0 to AIW62; 40001 to 4XXXX is the holding register, corresponding to V memory area.

All Modbus addresses are numbered from 1. The following table shows the correspondence between Modbus addresses and CTH200 addresses. The Modbus slave protocol allows you to limit the number of inputs, outputs, analog inputs, and holding registers (V zones) that Modbus master can access.

Modbus address	CTH200 address
000001	Q0.0
000002	Q0.1
000003	Q0.2
……	……
000127	Q15.6
000128	Q15.7
010001	I0.0
010002	I0.1
010003	I0.2
……	……
010127	I15.6
010128	I15.7
030001	AIW0
030002	AIW2
030003	AIW4
……	……
030032	AIW62
040001	HoldStart
040002	HoldStart+2
040003	HoldStart+4
……	……
04xxxx	HoldStart+2 x (xxxx-1)

【Using Modbus slave station protocol instructions】

※ CT_MODBUS slave station protocol instructions occupy CTH200 CPU resource

1) According to different Modbus protocol libraries, the FPORT 0 or 1 is used as the slave station protocol the condition that cannot be used for any other purpose, including communication with

MagicWorks PLC or SETP7-Micro/WIN, FPORT communication. The MBUS_INIT controls Port is set to Modbus is still PPI.

2) All SM corresponding to selected FPORT

3) Occupy 92 byte program space

※ Steps use Modbus slave station protocol in CTH200 program

1) MBUS_INIT can be used to initialize or modify Modbus communication parameters when there are only one loop cycle. When you insert the MBUS_INIT, several hidden subroutines and interrupt service routines are automatically added to your program.

2) Use only one MBUS_SLAVE instruction in your program. It execute in each cycle and serves all requests received.

3) Connect CTH200CPU with Modbus master station by communication cable.

※ All functions the Modbus slave station protocol instruction support

Modbus slave protocol instruction supports Modbus RTU protocol. It uses the FPORT function of S7--200, and supports most common Modbus functions. Support following Modbus functions:

Function	Description
1	Read single/multiple coil (actual output) status. 1 return any digital outputs ON/OFF status(Q)
2	Read single/multiple contactor (actual input) states. 2 returns the on/off status for any number of input points(I)
3	Read single/multiple hold registers. 3 returns the contents of V memory, keep the register in Modbus as the word-type, and up to 120 words in a request.
4	Read single/multiple input registers. 4 returns simulated input values.
5	Write single coil (actual output). 5 sets the actual output point which is not mandatory to the specified value. User program can override the values written by the Modbus request.
6	Write a single hold register. 6 writes a single hold register value to the S7-200 V memory.
15	Write multiple coils (actual output). 15 writes multiple actual output values to the Q image region of S7-200. The starting output point must be the beginning of a byte (such as Q0.0 or Q2.0), and the number of outputs to write is a multiple of 8. This is the limit of Modbus slave station protocol instruction. These points are not mandatory, user program can override the values written by the Modbus request.
16	Write multiple hold registers. 16 write multiple save registers to S7-200 V zone. Up to 120 words can be written in a request.

※ MBUS_INIT instruction

The MBUS_INIT directive is used to enable and initialize or disable Modbus communication. The MBUS_INIT directive must execute error-free before using the MBUS_SLAVE directive. Before proceeding to the next instruction, the MBUS_INIT instruction must finish executing and set Done bit immediately. The MBUS_INIT directive should be executed only once per communication state change. Therefore, EN input should use edge detection elements to trigger with pulses, or only execute during the first cycle.

Parameter address	Explanation	Type	Value range	Note
Mode	Select communication protocol: Input 1 defines Port as the Modbus protocol and enables it, and 0 defines Port as the PPI and disables Modbus protocol.	Bit		
Addr	Set site address	Byte	1~247	
Baud	Set baud rate	DWord	1.2K, 2.4K, 4.8K, 9600, 19.2K, 38.4K, 57.6K, 115.2	
Parity	Set parity	Byte	0—No parity 1—Odd parity 2—Even parity	All settings use one stop bit.
Delay	Extends standard Modbus information end timeout condition by increasing the specified number of ms for the standard Modbus information timeout.	Int	0~32767	Unit: ms
MaxIQ	Set available I and Q.	Int	0~128 0 means forbid I/O read and write	Recommend MaxIQ 128, allow to ask all IO of CTH200.
MaxAI	Sets the number of word input registers (AI).	Int	0~32 0 means forbid read analog input.	Recommend MaxAI value: H224/H224X-32 H226L/H226XL-32 /H228XL-32
MaxHold	Sets the number of available V storage word hold registers.	Int	0~32767	Unit: Word
HoldStart	Sets the starting address of the available holding register in the V storage area.	DWord	Pointer point to V memory	
Done	When the MBUS INIT instruction completes, the Done output is on.	Bit		
Error	Error output byte contains the result of the execution of the instruction.	Byte		

※ MBUS_SLAVE instruction

MBUS_SLAVE instruction is used to service requests from the Modbus master station and must be executed every cycle in order to check and respond to Modbus requests. When EN input switch on, it execute each cycle without input parameters.

Parameter explanation:

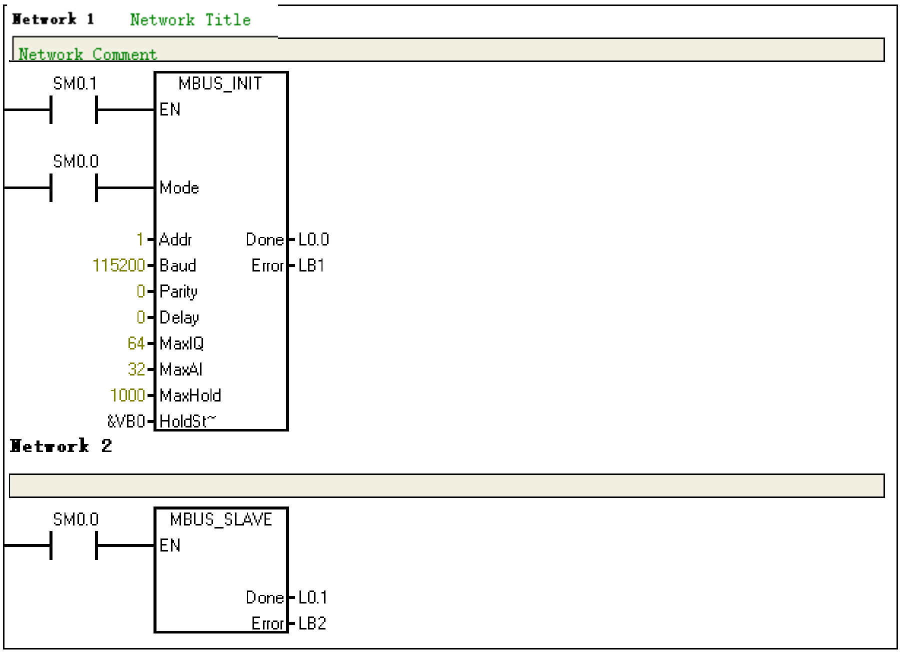
Parameter address	**Explanation**	**Type**	**Value range**	**Note**
Done	When MBUS_SLAVE respond to Modbus request Done output on. If no service request, Done output disconnect.	Bit		
Error	Error output byte contains the result of the execution of the instruction.	Byte	Error code as following	Valid only when Done is on. Error code will not change if Done is off.

Error code table:

0	No error
1	Memory area range error
2	Illegal baud rate or parity
3	Illegal slave station address
4	Illegal value of Modbus parameter
5	Holding register and Modbus slave station symbol address overlap
6	Receive parity error
7	Receive CRC error
8	Illegal function request/no supported functions
9	Illegal memory area address in request
10	Slave station function disable

※ Modbus slave station protocol instruction example

LAD program below established a no-parity slave station, address 1 and baud rate 115200:



※ MBUS_INIT parameter configuration explanation

Addr	Set slave station address 1
Baud	Set baud rate 115200
Parity	Set parity no-parity
Delay	Set delay time 0ms
MaxIQ	Set maxIQ 64 I/O(000001-0000064 and 010001-010064)
MaxAI	Set maxAI 32 analog inputs(030001-030032)
MaxHold	Set maxhold number of V area holding register(word as unit)
StartHold	Starting address of S7-200 V area holding register that Modbus master station can visit (such as &VB0).

【Use Modbus master station protocol instructions】

※ Modbus master station protocol instructions occupy CTH200CPU resource

1) According to different Modbus protocol libraries, the FPORT 0 or 1 is used as the slave station protocol. When Port 0 or Port 1 is used as Modbus protocol communication, it cannot be used for any other purpose, including communication with MagicWorks PLC or SETP7-Micro/WIN, FPORT communication. The MBUS_INIT controls Port is set to Modbus is still PPI.

2) All SM corresponding to selected FPORT

3) Occupy 119 byte program space

※ MBUS_CTRL instruction

Use SM0.0 to call MBUS_CTRL instruction to initialize master station and enable it.

Parameter explanation:

Parameter address	Explanation	Type	Value range	Note
Mode	Set communication mode: 1 enable Modbus protocol, and 0 recovery system to PPI protocol.	Bit		
Baud	Set baud rate	DWord	11200, 2400, 4800, 9600, 19200, 38400, 57600, 115200	
Parity	Set parity	Byte	0—No parity 1—Odd parity 2—Even parity	All settings use one stop bit.
Timeout	Time master station wait for slave station, ms as unit.	Int	1~32767	Typical value 1000ms.
Done	Completion bit, automatically set 1 after initialization.	Bit		
Error	Initialization error code	Byte	0—no error 1—illegal parity selection 2—illegal baud-rate selection 3—illegal mode selection	Valid only when Done bit is 1.

※ MBUS_MSG instruction

Use SM0.0 to call Modbus RTU master station R/W subroutine MBUS_MSG, send a Modbus request after First connected. One R/W function (MBUS_MSG) enable at same time.

Parameter explanation:

Parameter address	Explanation	Type	Value range	Note
First	R/W request bit	Bit		Use pulse trigger for each new R/W request.
Slave	Set slave station address	Byte	1~247	
RW	Operation order	Byte	0~read 1~write	
Addr	Select data type of R/W	DWord	00000 to xxxx—on-off output 10000 to 1xxxx--on-off input 30000 to 3xxxx—analog input 40000 to 4xxxx—holding register	
Count	Communication data number (number of word or bit)	Int		Maximum R/W of Modbus master station MBUS_MSG instruction is 120 words.
DataPtr	Data pointer, for read instruction, put data back into this data area; for write instruction, the data to be written is placed in this data area	DWord		
Done	Completion bit	Bit		
Error	Error code		Error code as following	Valid only when Done bit is 1.

Error code:

0	No error
1	Response parity error
2	unused
3	Receive timeout(slave station nonresponse)
4	Request parameter error
5	Modbus/FPORT disable
6	Modbus is busy for other request
7	Response error(response is not request operation)
8	Response CRC parity error
101	Slave station do not support request
102	Slave station do not support data address
103	Slave station do not support this data type
104	Slave station device fault
105	Slave station received information, but response delayed
106	Slave station is busy and refuse the information
107	Slave station refuse the information
108	Slave station register parity error

B Using CTH200 CPU Expansion Program Space

B.1 Function Introduction

The dynamic library function block is a special function provided by CTH200 CPU to expand the user programming program space and increase program privacy. It is an independent program block which is downloaded to PLC in advance and compiled to the program when the application is downloaded.

B.2 Instruction

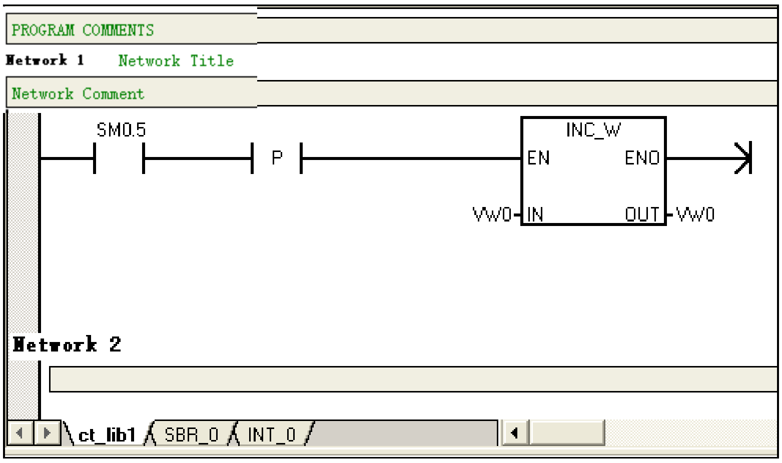
【Usable range of Dynamic Library】

CTH200CPU can load up to two dynamic libraries (" ct_lib1 "and" ct_lib2 "). The dynamic libraries different CPUs support and their sizes are shown in table below:

CPU type	ct_lib1	ct_lib2
H224/H224X	Allow 4K	Not allow
H226L/H226XL/H228XL	Allow 24K	Allow 24K

【Create Dynamic Library】

In the project, create all the program blocks used as dynamic library, name it as ct_lib1 or ct_lib2, download to the PLC, in the PLC will generate library function including all the subroutine blocks in the project of the dynamic connection library.



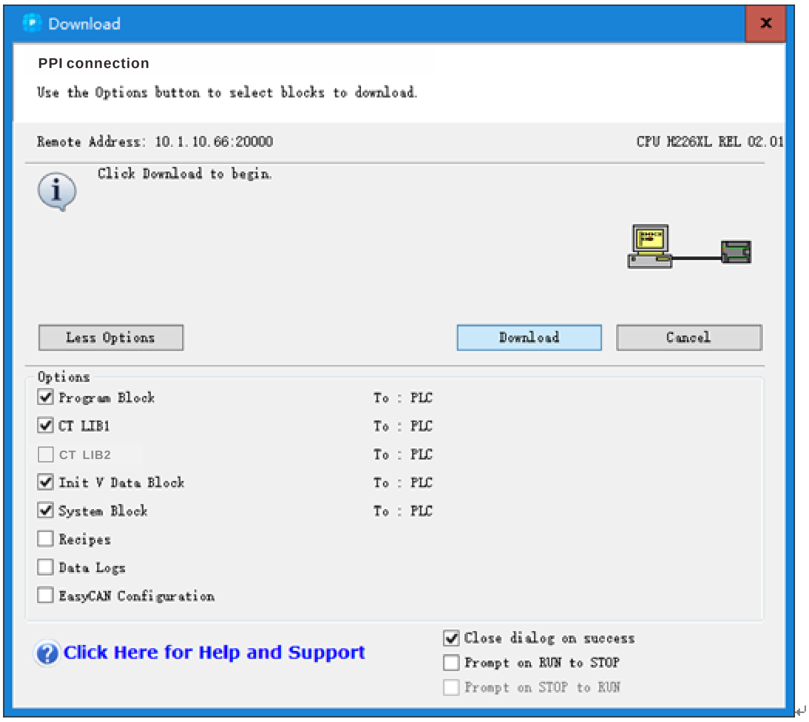
【Download Dynamic Library】

Download dynamic library program that edited well to the PLC (as shown in the figure below), in the PLC will generate a library function including all the subroutines in the project block dynamic connection library.

Every time download a new dynamic library, the PLC original library and program block is completely cleared. A dynamic library named ct_lib1 is generated in the PLC.

Note

Make sure only to download program block when download dynamic library.



Download another dynamic library named ct_lib2 to PLC the same way.

【Use Dynamic Library】

First create the empty blocks the same name with PLC subroutine of dynamic libraries, such as ct_lib1 or ct_lib2, then call it in project, after the program downloaded to the PLC, the empty block in the CPU will be replaced with the original download same name library functions, the corresponding dynamic database program execute when CPU run.

Note

Load the dynamic library before you load the program

【Clear Dynamic Library】

When downloading the new dynamic library, the original dynamic library in the PLC is cleared. Download a main program name ct_lib1 or ct_lib2 empty block project into PLC, the corresponding dynamic library is completely cleared.

C Using CTH200 PLC Expand 100K Data Block Library

C.1 Function Introduction

Extension data space is addressable 100K data storage out of CTH200 CPU existed storage area, its keep features same with V memory data space, keep the data by super capacitor when CPU power off, maximum data keep time up to 100 hours, the data storage capacity is 100 k bytes. COTRUST has provided library instructions for this data access. Users can add the library provided by COTRUST to MicroWin, access this data space through the read-write instructions provided in the library, and realize data exchange between extended data space and other data space.

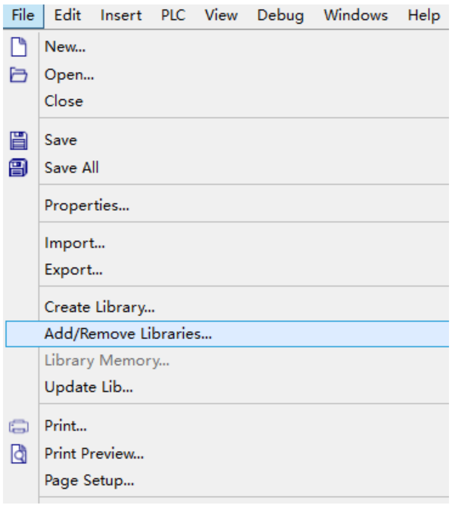
Note

Library download address: www.co-trust.com

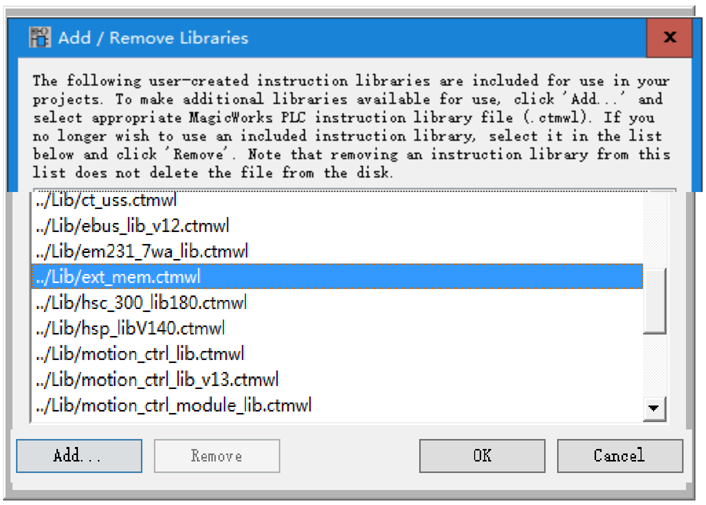
C.2 Instruction

【Add library file】

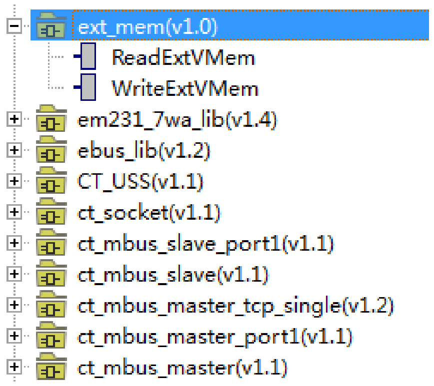
“File” -> “Add/delete library”, find “ext_mem.ctmwl” as shows below:



Find file where you save ext_mem.ctmwl, click “Add”

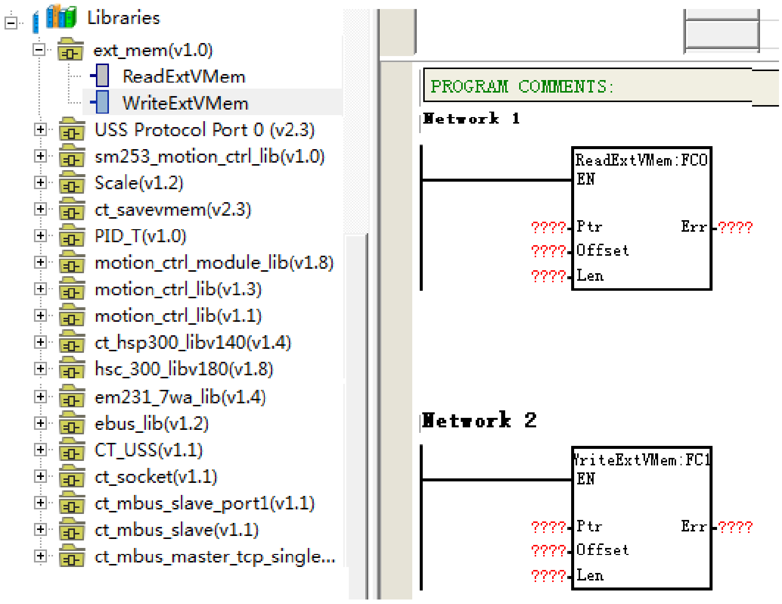


You can see new ext_mem library under directory tree “Libraries” after installation.



【Call ext_mem library】

Click “Network” to add a function block, then double-click “ReadExtVMem” and “WriteExtVMem” to get the function block in “Network”. As the following figure:



C.3 Ext_mem library Function Explanation

【Use ReadExtVMem to read data from expansion data space】

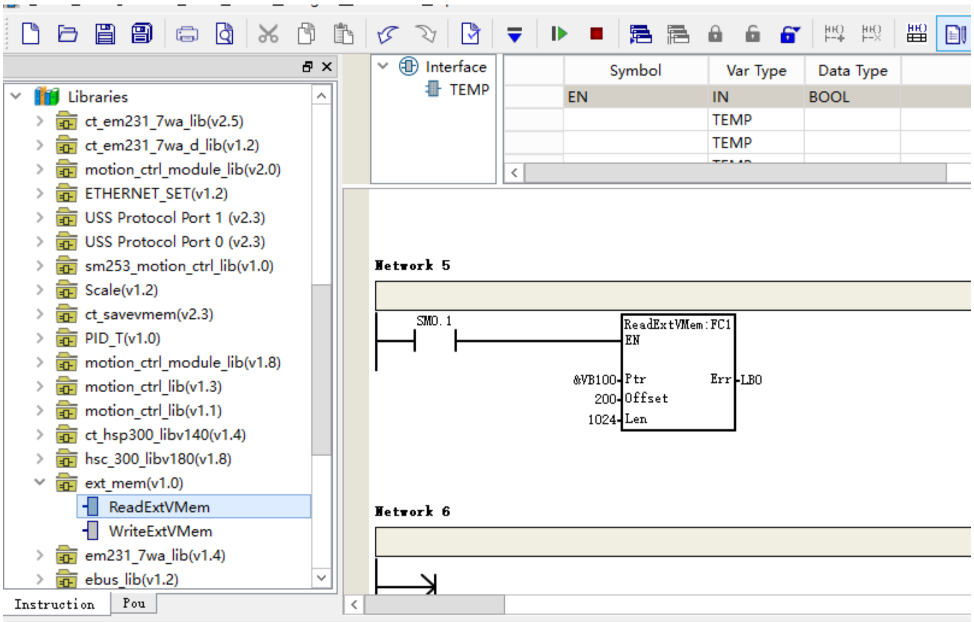
Parameter address	Explanation	Type	Value range	Note
Ptr	Memory address pointer to read	DWord		Such as: &VB0,&IB0
Offset	Read starting offset address in expansion memory	DWord	0-102399	
Len	Read memory length(byte)	DWord		
Err	Return value indicate if it read successfully	BYTE		0 means read successfully others fail.

【Use WriteExtVMem to write data to expansion data space】

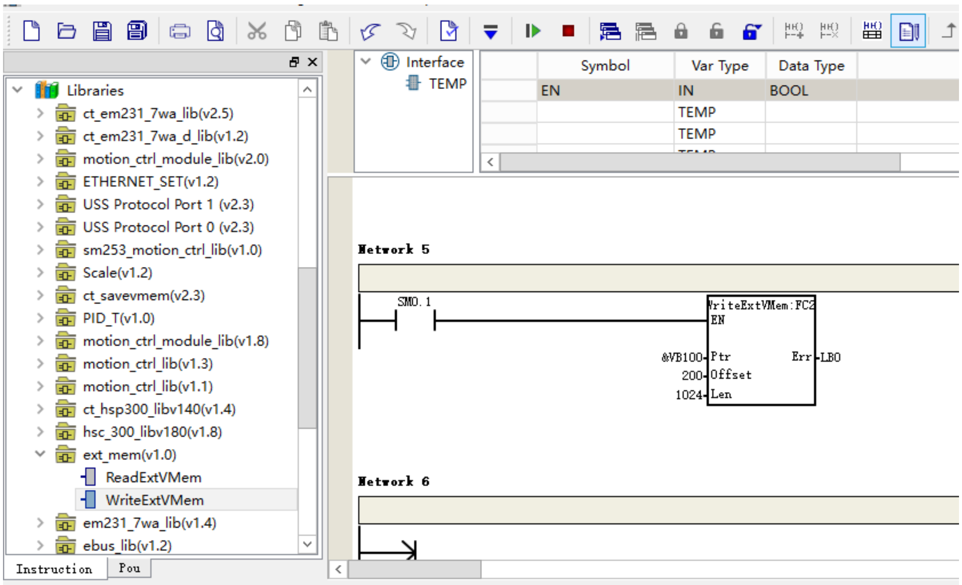
Parameter address	Explanation	Type	Value range	Note
Ptr	Write memory length(byte)	DWORD		Such as: &VB0,&IB0
Offset	Return value indicate if it write successfully	DWORD	0-102399	
Len	Write memory length(byte)	DWORD		
Err	Return value indicate if it write successfully	BYTE		00 means write successfully others fail.

C.4 Application Example

1. Read 1024 bytes from offset 200 to VB100 in extended memory



2. Write the first 1024 bytes of VB100 to the extended memory starting at offset 200



D Using PID Control Library of CTH200 PLC

D.1 Using CPU Embedded PID_T Library

【Introduction】

PID_T function block integrate inside the CPU, does not occupy user data space, offered to user as a library function. It mainly aims at intelligent temperature control PID function, with self-tuning and self-adaptive function. No complex programming, but only need to call and set some simple parameters to use, and accurate temperature control.

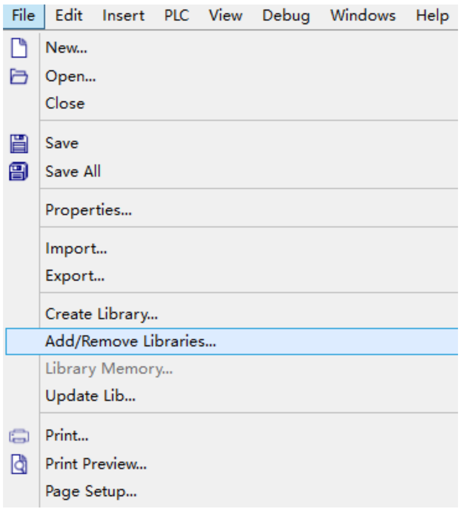
Note

Library download address: www.co-trust.com

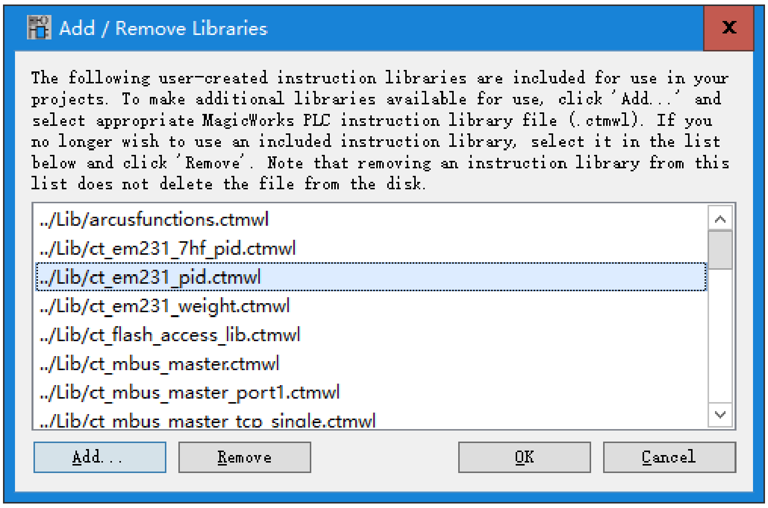
【Software Installation】

※ Add library file

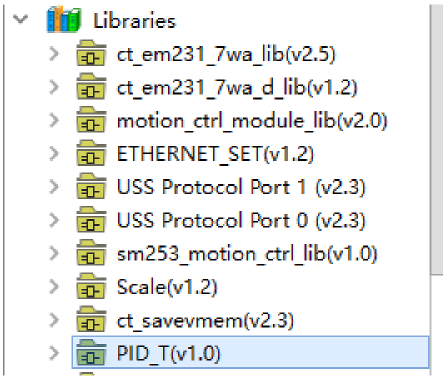
“File” -> “Add/delete library”, find “pid_t.ctmwl”, as the following figure



Find file where you save pid_t.ctmw, click “Add”

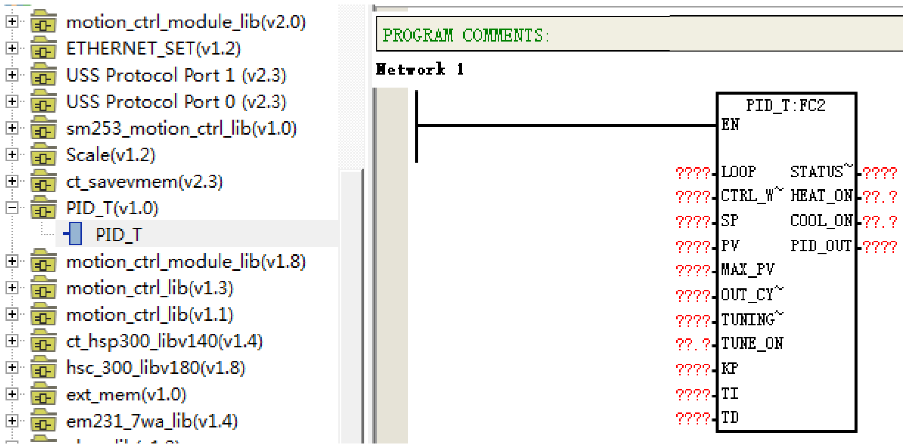


You can see new PID_T library under directory tree “Libraries” after installation.



※ Call PID_T library

Click “network” to add a function block, then double-click “PID_T” to get the function block in “network”. As the following figure:



【PID_T library function explanation】

Address parameter explanation

Parameter address	Explanation	Type	Value range	Note
LOOP	Belong to which PID, no repeat. Start from 0.	Word, constant or variable	0-63	Control the loop ID
CTRL_WORD	Control Word, control PID run.	Word, constant or variable		Common control Word: 1) 16#03(Only heating output with adaptive function) 2) 16#07(Only cold output with adaptive function)
SP	Setting value	Word, constant or variable	-32768-32767	Unit: 0.1℃

PV	Measurement(feedback value)	Word, variable	-32768-32767	Unit: 0.1℃
MAX_PV	Maximum measurement	Word, constant or variable	-32768-32767	Unit: 0.1℃
OUT_CYCLE	Pulse output cycle	Word, constant or variable	1-255	Unit: s
TUNING_K	Self-tuning coefficient	DWord, float	0.5-2.0	0.5: The system control is required small overshoot 1.0: Normal response 2.0: The system control is required to response fast and large overshoot
TUNING_ON	Start self-tuning	Bit, variable		Automatic reset after self-tuning
Kp	Kp	Word, variable		If the word variable is assigned constant, self-tuning function cannot be performed
Ｔi	Ｔi	Word, variable	1-3600	Unit: s If the word variable is assigned constant, self-tuning function cannot be performed
Td	Td	Word, variable	0-3600	Unit: s If the word variable is assigned constant, self-tuning function cannot be performed
STATUS_WORD	Status Word	Word, variable		Run and warning status
HEAT_ON	Heat-output	Bit		
COOL_ON	Cool-output	Bit		
PID_OUT	PID analog output	Word, constant or variable	Only define heat output: 0-32000. Cool output: -32000-32000	

【Control Word and status Word bit address】

Control word bit address definition:

Bit	Setting	Note
0	0	PID stop
	1	PID run
1	0	Integral always work, Kp do not self-tuning
	1	Integral separate Kp self-tuning
2	0	PID unipolar output
	1	PID bipolar output
3	0	Reserved
	1	Reserved
4	0	Integral works
	1	Integral does not work
5	0	Differential works
	1	Differential does not work
6		Reserved
7		Reserved

Status word bit address definition:

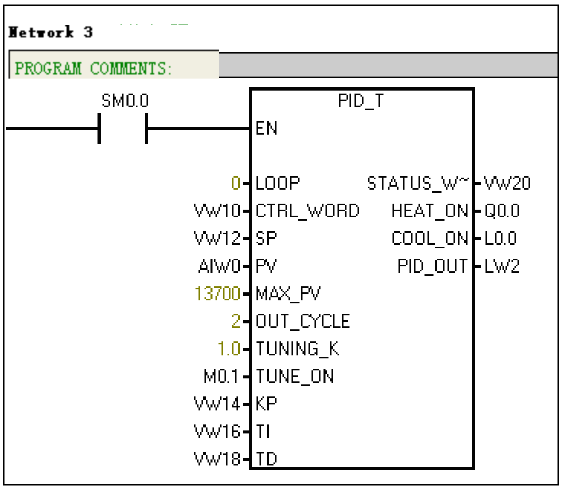
Bit	Setting	Note
0	0	No disconnection error
	1	Disconnection error
1	0	Not implemented self-tuning
	1	Self-tuning
2	0	No self-tuning error
	1	Self-tuning error
3	0	No heating
	1	Heating
4	0	No cooling
	1	Cooling
5	0	PID stop
	1	PID run
6		Reserved
7		Reserved

【Example】

Take a temperature zone as example, and the system and I/O configuration table is as follows:

System configuration	CPU226L+231-7PD32	Temperature acquisition using four - way thermocouple module	
Control request	1 Only heating-output, no cooling-output 2 Request self-tuning 3 K thermocouple		
I/O distribution			
Q0.0	Heating-output		
AIW0	Temperature input	K thermocouple	
M0.0	PID run/stop bit		
M1.0	Self-tuning start bit		

Application program



Explanation

PID_T parameter explanation

Parameter	Address or value	Explanation	Note
LOOP	0	0 for first loop	
CTRL_WORD	VW10		
SP	VW12		
PV	AIW0		
MAX_PV	13700	Maximum input is 13700 for K thermocouple	
OUT_CYCLE	2	2s, pulse output cycle	
TUNING_K	1.0		
TUNING_ON	M0.1	Set from 1, reset after setting.	
Kp	VW14	Kp, setting value will automatically write to this variable after setting, users can adjust themselves.	
Ｔi	VW16	Ｔi, setting value will automatically write to this variable after setting, users can adjust themselves.	
Td	VW18	Td, setting value will automatically write to this variable after setting, users can adjust themselves.	
STATUS_WORD	VW20	Status Word	
HEAT_ON	Q0.0	Heating output	
COOL_ON	L0.0	Use a local variable for unused	
PID_OUT	LW2	Use a local variable for unused	

D.2 Using CTH200 PLC Thermocouple PID Control Module Library

【Function introduction】

The "PID_setting" function library is specially designed to provide parameter setting for the thermocouple PID expansion module (e.g., SM231-7TD, SM231-7TF). Integrate PID algorithm inside the thermocouple PID module. No complex programming, but only needs to call "PID_setting" library to set some simple parameters to use module, and the temperature control is accurate.

Library download address: www.co-trust.com

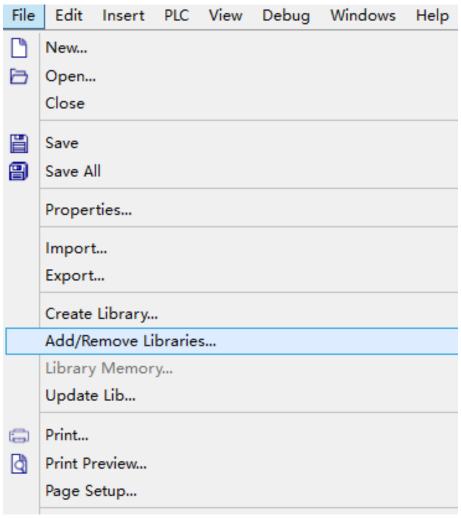
Note

- Apply to CTS7 231-7TD32, CTS7 231-7TF32 modules;
- Use SM231-7TD32 and SM231-7TF32 will occupy some V memory area, do not use V memory area when programming
- Siemens CPU222 cannot use the library for data space restriction

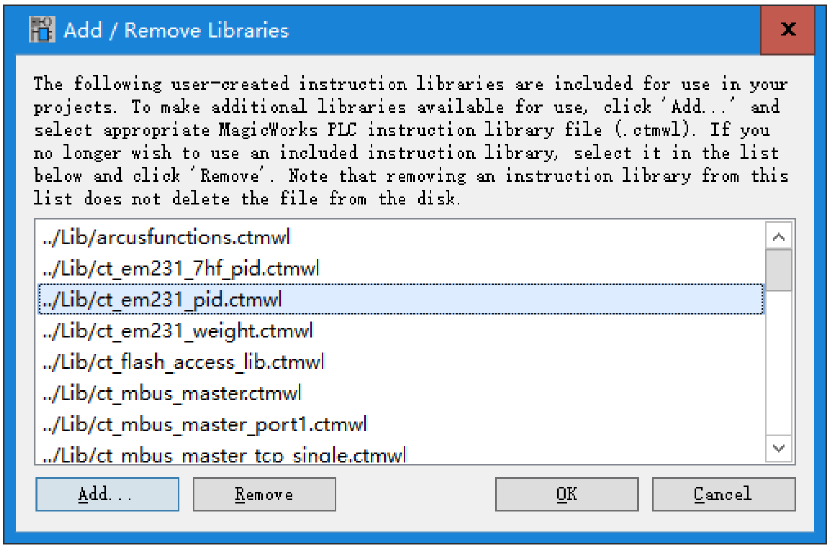
【Installation】

※ Add library file

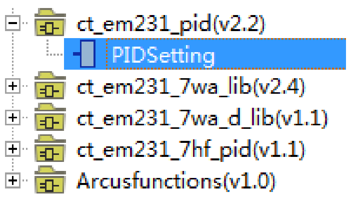
“File” -> “Add/delete library”, find “em231 pid lib.ctmwl”, as the following figure



Find file where you save “em231 pid lib.ctmwl” click “Add”

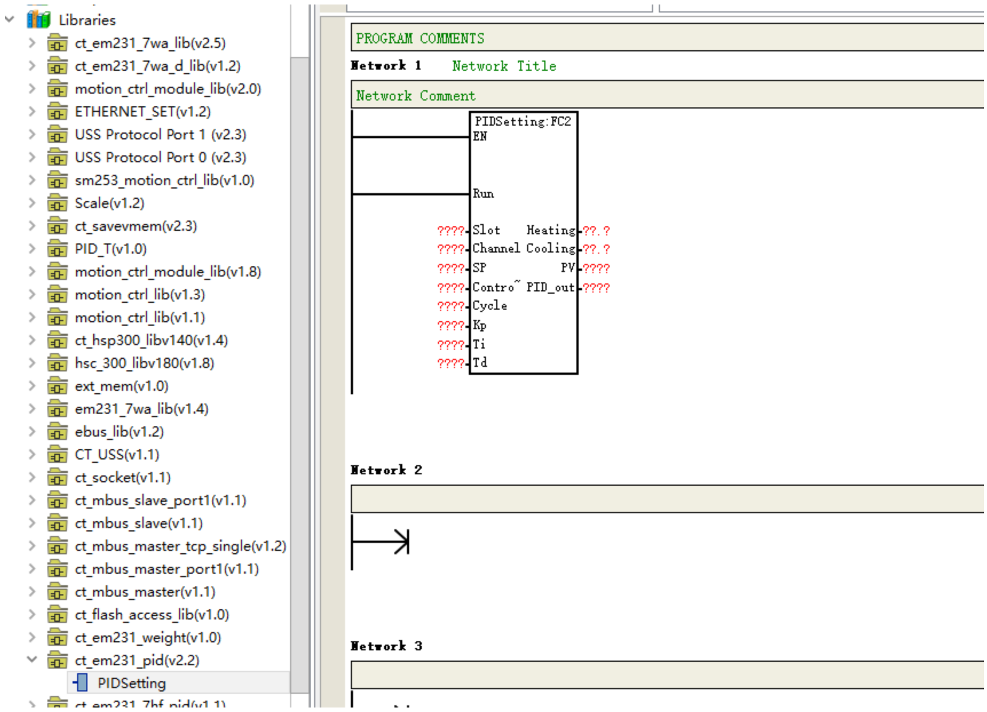


You can see new PID_setting library under directory tree “Add” after installation.



※ Call PID_Setting library

Click “network” to add a function block, then double-click “PID_Setting” to get the function block in “network”. As the following figure:



※ 【PID_setting library function explanation】

Address parameter explanation

Parameter address	Explanation	Type	Value range	Note
Run	Run	Bit	0 or 1	
Slot	Start from slot 0	Word, constant or variable	0~6	
Channel	Channel number	Word, constant or variable	0~7	
SP	Setting value	Word, constant or variable	-2000~32767	Unit: 0.1℃
CTRLByte	Control byte, control PID run	constant or variable		Common control Word: 1) 16#03(Only heating output with adaptive function) 2) 16#07(Only cold output with adaptive function)
Cycle	Pulse output cycle	Word, constant or variable	1~255	Unit: s
Kp	Kp	Word, int, constant or variable		
Ti	Ti	Word, int,	1~3600	Unit: s

		constant or variable		
Td	Td	Word, int, constant or variable	0~3600	Unit: s
Heating	Heating-output	Bit		
Cooling	Cooling-output	Bit		
PV	Measurement(feedback value)	Word, variable	-2000~32767	Unit: 0.1℃
PID_out	PID analog output	Word, int, variable	Only define heat output: 0-32000. Cool output: -32000~32000	

※ Control word bit address

Control word bit address definition:

Bit	Setting	Note
0	0	PID stop
	1	PID run
1	0	Integral always work, Kp do not self-tuning
	1	Integral separate Kp self-tuning
2	0	PID unipolar output
	1	PID bipolar output
3	0	Reserved
	1	Reserved
4	0	Integral works
	1	Integral does not work
5	0	Differential works
	1	Differential does not work
6	-	Reserved
7	-	Reserved

※ PID address and parameter configuration

• PID address calculation formula

Address name	Formula	Note
PID parameter address	A=(2048+S*256)+16*C	S is slot where module is(range: 0~6) C means channel, 231-7HF is 0~7
PID positive pulse output address	X=(2048+S*256)+12	
PID negative pulse output address	Y=(2048+S*256)+13	

• PID parameter output (modules to CPU)

Items	Address	Value set range	Actual value
Actual temperature	VW A	0~13000	0~1300°
Status Word	VW A+2		
PID analog output	VW A+4	-32000~32000	

• PID parameter input(CPU to modules)

Items	Address	Value set range	Actual value
Setting temperature	VW A+128	0~13000	0~1300
Control byte		When VB A+130 bit = 0	
	V(A+130).0	PID stop, no output	PID run
	V(A+130).1	Integral always work, Kp do not self-tuning	Integral separate Kp self-tuning
	V(A+130).2	PID unipolar output, 0~32000	PID bipolar output, -32000~32000, both heat and cool
	V(A+130).3	Unused	
	V(A+130).4	Integral works	
	V(A+130).5	Differential works	
	V(A+130).6	Actual temperature filter, stronger ant jamming capability	Actual temperature do not filter
PID pulse output cycle setting	VW A+132	1~255	
Kp	VW A+134	0~9999	0~999.9
Ti	VW A+136	0~3600	
Td	VW A+138	0~3600	

• Heating pulse output address

Channel 0 pulse output	V X.0
Channel 1 pulse output	V X.1
Channel 2 pulse output	V X.2
Channel 3 pulse output	V X.3
Channel 4 pulse output	V X.4
Channel 5 pulse output	V X.5
Channel 6 pulse output	V X.6
Channel 7 pulse output	V X.7

• Cooling pulse output address

Channel 0 pulse output	V Y.0
Channel 1 pulse output	V Y.1
Channel 2 pulse output	V Y.2
Channel 3 pulse output	V Y.3
Channel 4 pulse output	V Y.4
Channel 5 pulse output	V Y.5
Channel 6 pulse output	V Y.6
Channel 7 pulse output	V Y.7

【Example】

System description

This routine sets the parameters of the first PID loop (channel 0) of SM231-7TD expansion module (slot 0). Call PIDSetting to set the loop parameters, no need to calculate the PID parameter address, just input the slot and channel number where the loop is, and then enable Run to run the loop.

Q0.0 is the positive pulse output; Q0.1 is negative pulse output;

VW0 is the actual temperature; VW2 is PID analog output;

Use other addresses to Modify PID setting parameters;

Set temperature: VW120;

Control Word: VB122;

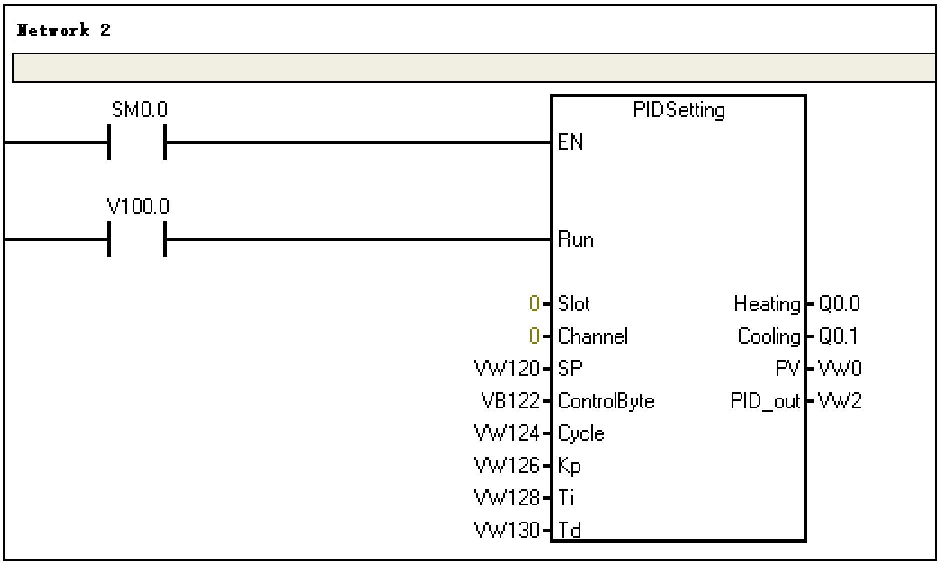
Pulse output cycle: VW124;

Kp: VW126;

Ti: VW128;

Td: VW130;

Application program



Note

For PID modules normal use, please do not occupy following PID V storage when programming.

Address that module occupied in slot 0: VW2048 to VW2298

Address that module occupied in slot 1:VW2304 to VW2554

Address that module occupied in slot 2:VW2560 to VW2810

Address that module occupied in slot 3:VW2816 to VW3066

Address that module occupied in slot 4:VW3072 to VW3322

Address that module occupied in slot 5:VW3328 to VW3578

Address that module occupied in slot 6:VW3584 to VW3834

E Using Motion Control Library “motion_ctrl_lib”

E.1 Introduction

The motion_ctrl_lib library is used for CTH200 PLC. No complex programming, just call and set some simple parameters to use, this series of CPU has multi-axis independent control function, CPU with interpolation can carry out any two axis linear interpolation and circular interpolation, also support linear acceleration and deceleration control.

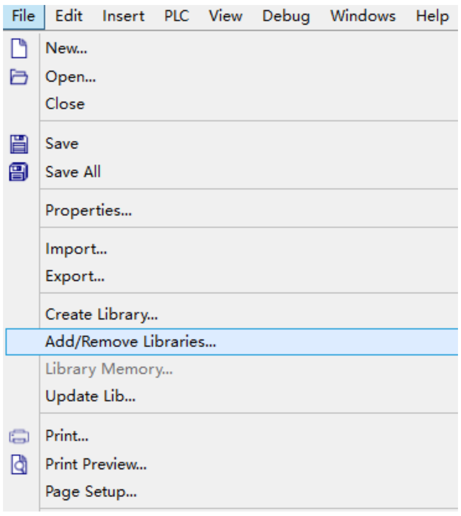
Note

Library download address: http://www.co-trust.com

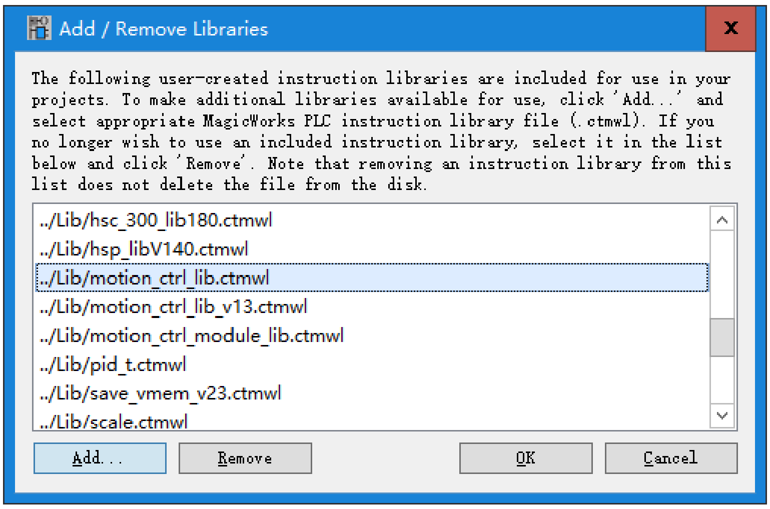
E.2 Installation Instruction

【Add library file】

“File” -> “Add/delete library”, find “motion_ctrl_lib.ctmwl”, as the following figure

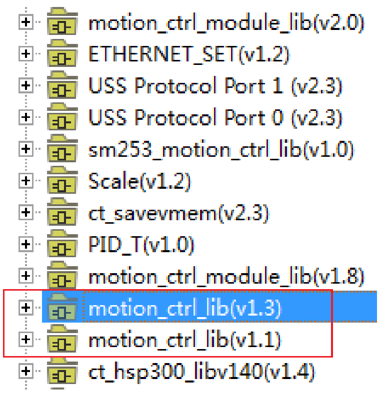


Find file where you save “motion_ctrl_lib.ctmwl” click “Add”.



You can see new motion_ctrl_lib library under directory tree “Libraries” after installation.

V1.1 (at left)/V1.3:



E.3 Instructions Explanation

【Motion axes with the CPU I/O】

CPU type	CPU H224/H226L			CPU H224X/H226XL/H228XL			
Output	Q0.0	Q0.1	Q0.2	Q0.0	Q0.1	Q0.2	Q0.3
Motion control I/O	Pulse_0	Pulse_1	Pulse_2	Pulse_0	Pulse_1	Dir_0	Dir_1

Notice

Pulse_0 --------0 axis pulse output; Dir_0 ------0 axis direction output;

Pulse_1 --------1 axis pulse output; Dir_1 ------1 axis direction output;

Pulse_2 --------2 axis pulse output;

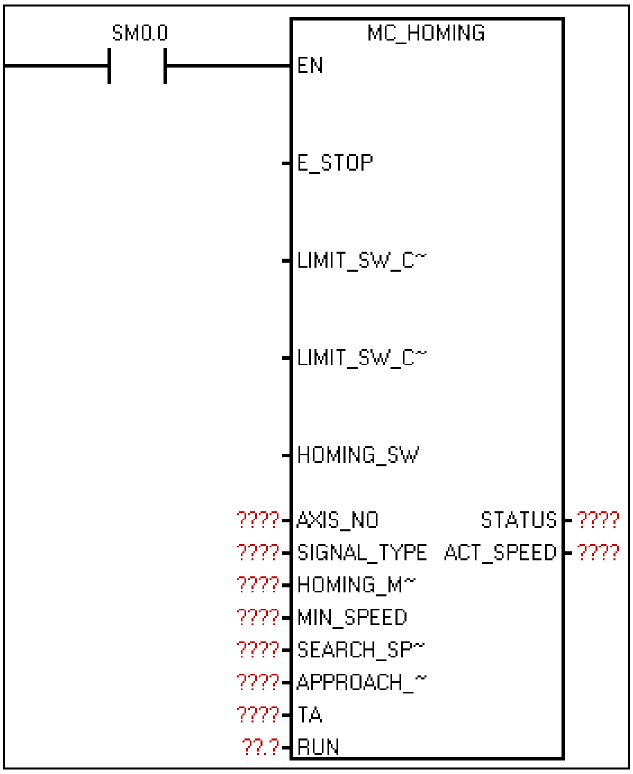
【Motion_ctrl_lib instruction list】

Function name	Instruction name	CTH200 series CPU	
		H224/H226L	**H224X/H226XL/H228XL**
MC_HOMING	Homing instruction	Nonsupport	Support
MC_INIT_DIR	Configure motor direction instruction	Nonsupport	Support
MC_READ_POS	Read position instruction	Nonsupport	Support
MC_EXT_RESET_EN	External reset coordinate enable instruction	Support	
MC_EXT_RESET_EN_EXT	External reset coordinate enable instruction II	Nonsupport	Support
MC_SET_POS_ZERO	Software reset instruction	Nonsupport	Support
MC_SET_POS_PV	Set target position instruction	Nonsupport	
MC_SET_MAX_ACCELE	Set maximum acceleration instruction	Nonsupport	Support
MC_SPEED_CTL	Speed control instruction	Support	
MC_PTP_R	Uniaxial relative motion instruction	Support	
MC_PTP_A	Uniaxial absolute motion instruction	Nonsupport	Support
MC_LINE_R	Two axis linear interpolation relative motion instruction	Nonsupport	
MC_LINE_A	Two axis linear interpolation absolute motion instruction	Nonsupport	Support

MC_CIRCLE_R	Two axis circular interpolation relative motion instruction	Nonsupport	Support
MC_CIRCLE_A	Two axis circular interpolation absolute motion instruction	Nonsupport	Support
MC_SET_CI_MODE	Set continuous interpolation function instruction	Nonsupport	Support

HOMING instruction

① Function name: MC_HOMING



② Function: Find device origin by setting parameters such as homing mode.

The relationship between the axis number and external reset IO signal (such as homing Z pulse):

Axis 0 ——I0.2 (HSC0, SM37.0)   Axis 1 ——I1.0 (HSC1, SM47.0)

Axis 2 ——I1.4 (HSC2, SM57.0)   Axis 3 ——I0.5 (HSC4, SM147.0)

If the homing mode refers to the origin switch (mode 3 or 4), connect origin switch signal (the HOMING_SW parameter of the instruction) to the corresponding point above; otherwise, the origin cannot be found.

③ Parameter

Name	I/O	Description	Type	Value range	Note
E_STOP	IN	Emergency stop bit 1: valid, 0: invalid	BOOL	0~1	1 Run only when RUN =1 and E_STOP =0. 2 When E_STOP=1, RUN reset internal.
LIMIT_SW_CC	IN	CCW counterclockwise	BOOL	0~1	

W		stroke limit input			
LIMIT_SW-CW	IN	CW counterclockwise stroke limit input	BOOL	0~1	
HOMING_SW	IN	Origin signal input	BOOL	0~1	
AXIS_NO	IN	Axis number	BYTE	0~3	Unchangeable in the process
SIGNAL_TYPE	IN	7 6 5 4 3 2 1 0 Signal type: Bit0: Counterclockwise stroke limit input signal type 0—High level 1—Low level Bit1: Clockwise stroke limit input signal type 0—High level 1—Low level Bit2: Origin switch signal type 0—High level 1—Low level	BYTE	0~255	
HOMING_MODE	IN	Homing mode	BYTE	1~14	Please refer to Appendix *E.4 Back to Zero Mode Illustration*
MIN_SPEED	IN	Minimum speed Unit: Hz	DWORD	0~200000	1 Pulse output close when speed < 5Hz. 2 Unchangeable in the process 3 Search speed should not be too large, close speed should be as small as possible
SEARCH_SPEED	IN	Origin search speed Unit: Hz	DWORD	0~200000	
APPROACH_SPEED	IN	Origin close speed Unit: Hz	DWORD	0~200000	
TA	IN	Time of ACC/DEC. Unit: ms	DWORD	0~10000	Changeable in the process
RUN	IN	Run enable bit 1: Valid	BOOL	0~1	1 Run only when RUN =1 and E_STOP =0. 2 RUN reset internal when run complete. 3 When E_STOP=1, RUN reset internal.
STATUS	OUT	7 6 5 4 3 2 1 0 Output status byte: Bit0: Parameter configuration error sign 1—Parameter	BYTE	0~255	Bit0: 1 Judge Only to axis parameter configuration error and homing mode out of range. 2 No error reported for

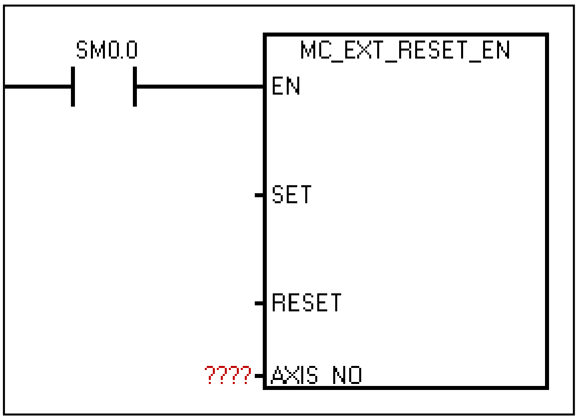
		configuration error 0—Parameter configuration normal Bit1: Run sign 1—Running 0—Do not run Bit2: Completion sign 1—Instruction execution completed 0—Instruction execution do not completed Bit3: Busy sign 1—Valid, the axis occupied by other instruction 0—Invalid, executing instruction or execution done. Bit4: Reserved Bit5: Find origin or not 1—Found origin 0—Not find origin			other parameters, it will set to the nearest reasonable value automatically. 3 Report parameter failure if TA=0 and the maximum acceleration is not set; TD the same
ACT_SPEED	OUT	Current speed	DWORD	0~200000	

④ Explanation

The program detect each input by scanning (the Z phase is not affected by this), so when the switch value changes and process not in time, there may be some delay. If the homing speed (including search and close speed) is too high, the processing delay will be amplified, resulting in inaccurate return.

External reset coordinate enable instruction

① Function name: MC_EXT_RESET_EN



② Function: When call this instruction, set whether enable external IO reset absolute coordinate value.

Note: The relationship between the axis number and external reset signal:

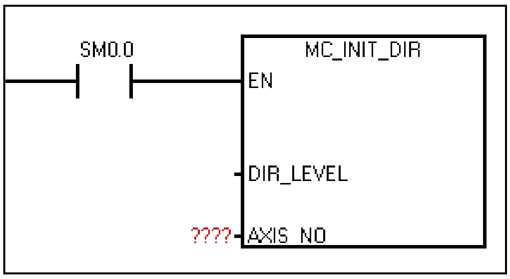
Axis 0 ——I0.2 (HSC0, SM37.0)
Axis 1 ——I1.0 (HSC1, SM47.0)
Axis 2 ——I1.4 (HSC2, SM57.0)
Axis 3 ——I0.5 (HSC4, SM147.0)

③ Parameter

Name	I/O	Description	Type	Value range
SET	IN	SET rising edge, set external reset enable, for each call, first reset SET and then set 1	Bool	Range: 0~1
RESET	IN	RESET rising edge, prohibit external reset enable. For each call, first reset and then set 1	Bool	Range: 0~1
AXIS_NO	IN	Set axis number, 0/1/2/3	Byte	Range: 0~3

Configure motor direction instruction

① Function name: MC_INIT_DIR



② Function: Configure motor direction

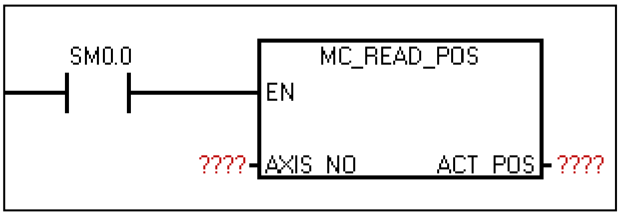
Note: Execute one time for first scan cycle CPU power on

③ Parameter

Name	I/O	Description	Type	Value range	Note
DIR_ LEVEL	IN	Configure effective level when positive direction signal Set motor forward for corresponding direction axis output “1” when DIR_LEVEL is 1. Set motor reverse for corresponding direction axis output “0” when DIR_LEVEL is 0.	Bool	Range: 0~1	Default: 1, Motor forward when direction axis output is“1”
AXIS_NO	IN	Set axis number, 0/1/2/3	Byte	Range: 0~3	

Read position instruction

① Function name: MC_READ_POS



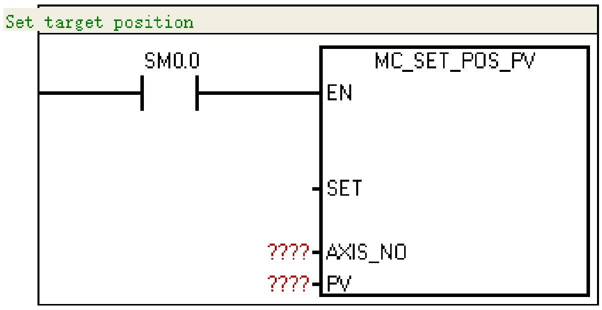
② Function: Read the absolute coordinate value of each axis. Once the origin coordinate is set, the value will be calculated according to the relationship between output pulse and direction: positive rotation output one pulse: +1, and reverse rotation output a pulse: -1. What you end up with is an absolute coordinate with the set point as the origin.

③ Parameter

Name	I/O	Description	Type	Value range	Note
AXIS_NO	IN	Set axis number, 0/1/2/3	Byte	Range: 0~3	
ACT_POS	OUT	Absolute coordinate of current axis (1 pulse 1 units coordinate)	DInt	-2147483647~ +2147483647	This instruction no error and set the axis number correctly.

Set target position instruction

① Function name: MC_SET_POS_PV



② Function: This instruction is used to write the absolute location of the machine to the module. For example, machine can save current position when the power cut off. And this position instead of origin will be written back to the module next time the machine power on, so that the absolute position counting starting point of the module is consistent with the actual starting point of the machine. If this position happens to be the origin, this instruction has the same effect as MC_SET_POS_ZERO.

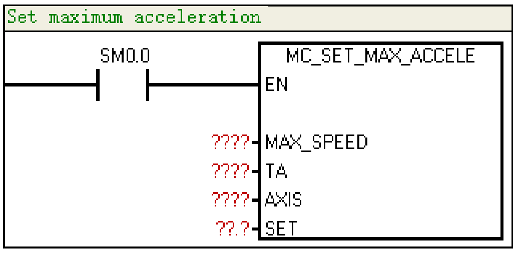
③ Parameter

Name	I/O	Description	Type	Value range
AXIS_NO	IN	Set axis number	BYTE	0~3
SET	IN	SET rising edge, instruction enable, reset and then set 1 for each call.	BOOL	0~1
PV	IN	Set the target position, points plus or minus. The positive output pulse indicates the positive direction along	DINT	-2147483648 ~ +2147483647

		the X-axis, and the negative pulse number indicates the negative direction along the X-axis.		

Set maximum acceleration instruction

① Function name: MC_SET_MAX_ACCELE



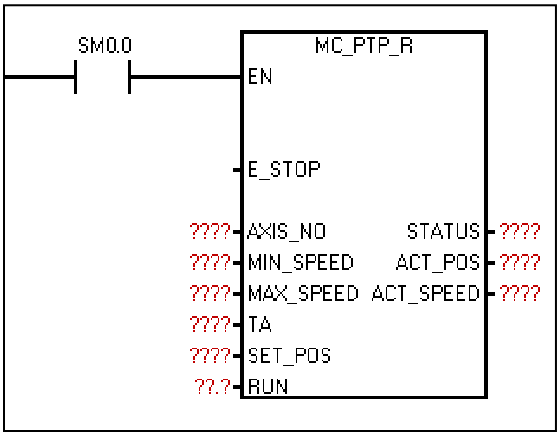
② Function: Set maximum acceleration(= MAX_SPEED/TA)(TA≠0)(no call indicates no setting for acceleration)

③ Parameter

Name	I/O	Description	Type	Value range	Note
MAX_SPEED	IN	Maximum speed Unit: Hz	DWORD	0~200000	Changeable in the process
TA	IN	Time of ACC/DEC, Unit: ms	DWORD	0~10000	Changeable in the process; if TA=0, no settings for acceleration.
AXIS	IN	Set axis number	BYTE	0~3	This instruction no error and set the axis number correctly.
SET	IN	Give SET a rising edge to take effect after above parameter ensure.	BOOL	0~1	

Uniaxial relative motion instruction

① Function name: MC_PTP_R



② Function: Use for single - axis point - to - point control (single - axis fixed - length drive). One call for fixed pulse output, through the maximum and minimum speed and deceleration time setting, the output pulse will gradually accelerate to the largest speed, when the pulse number is ready to run, cut down pulse frequency automatically, to prevent vibration or jammed caused by too much inertia when start or stop the machine.

③ Parameter

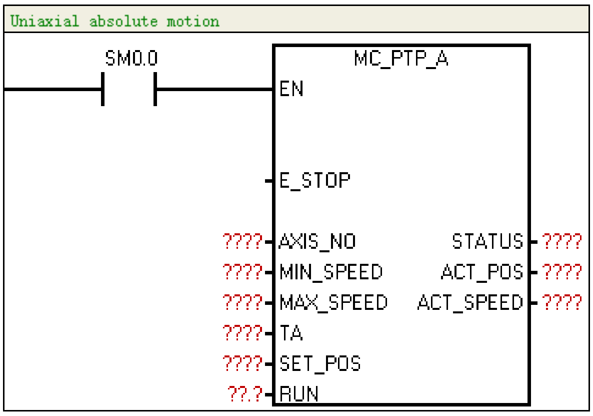
Name	I/O	Description	Type	Value range	Note
E_STOP	IN	Emergency stop bit 1: valid, 0: invalid	Bool	0/1	1. Run only when Run =1 and E_Stop =0 2. RUN reset internal when E_STOP = 1
AXIS_NO	IN	Set axis number, 0/1/2/3	Byte	0~3	Unchangeable in the process
MIN_SPEED	IN	Minimum speed of run or stop. Unit: HZ	Dword	500~200000	1. Set minimum speed less than maximum speed 2. Changeable in the process
MAX_SPEED	IN	Maximum speed of run Unit: HZ	Dword	500~200000	
TA	IN	Time of ACC/DEC, Unit: ms	Dword	0~10000	Changeable in the process; if TA=0, no settings for acceleration.
RUN	IN	Run enable bit 1: valid	BOOL	0~1	1 Run only when RUN =1 and E_STOP =0. 2 RUN reset internal when run complete. 3 when E_STOP=1, RUN reset internal.
STATUS	OUT	7 6 5 4 3 2 1 0 Output status byte: Bit0: Parameter configuration error sign 1—Parameter configuration error 0—Parameter configuration normal Bit1: Run sign 1—Running 0—Do not run Bit2: Completion sign 1—Instruction execution completed 0—Instruction execution do not completed Bit3: Busy sign	BYTE	0~255	Bit0: 1 Judge Only to axis parameter configuration error and homing mode out of range. 2 No error reported for other parameters, it will set to the nearest reasonable value automatically. 3 Report parameter failure if TA=0 and the maximum acceleration is not set; TD the same

		1—Valid, the axis occupied by other instruction 0—Invalid, executing instruction or execution done.			
SET_POS	IN	Output pulse. The positive output pulse indicates the positive direction along the X-axis, and the negative pulse number indicates the negative direction along the X-axis.	DInt	-2147483647 ~ +2147483647	Modifiable during operation. When the new set value is greater than the number of pulses output, the final output pulse will be subject to the new set value. When the new set value is less than the number of output pulses, the pulse output will be stopped immediately
ACT_POS	OUT	Absolute coordinate of current axis (1 pulse 1 units coordinate)	DInt	-2147483647 ~ +2147483647	This instruction no error and set the axis number correctly.
ACT_SPEED	OUT	Current run speed	Dword	500~200000	Note: This value may be a little off actual one, at most 5K, related to acceleration time and setting speed

Note 1
Theoretically DCC/DEC time TA⩽ (MAX_SPEED) – (MIN_SPEED), such as TA>(MAX_SPEED) – (MIN_SPEED), then default TA=(MAX_SPEED) – (MIN_SPEED) inside the motion control instruction

Uniaxial absolute motion instruction

① Function name: MC_PTP_A



② Function: Use for single - axis point - to - point control (fixed-point instead of fixed - length). One call for fixed pulse output, through the maximum and minimum speed and deceleration time setting, the output pulse will gradually accelerate to the largest speed, when the pulse number is ready to run, cut down pulse frequency automatically, to prevent vibration or jammed caused by too much inertia when start or stop the machine.

③ Parameter

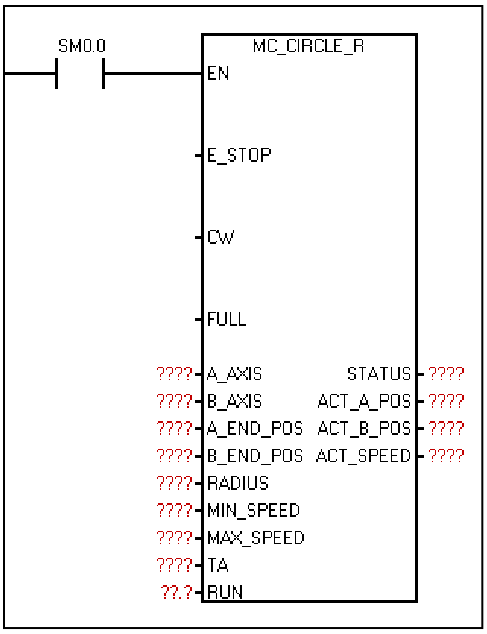
Name	I/O	Description	Type	Value range	Note
E_STOP	IN	Emergency stop bit 1: Valid, 0: Invalid	Bool	0/1	1 Run only when Run =1 and E_Stop =0 2 RUN reset internal when E_STOP = 1
AXIS_NO	IN	Set axis number, 0/1/2/3	Byte	0~3	Unchangeable in the process
MIN_SPEED	IN	Minimum speed of run or stop. Unit: HZ	Dword	0~200000	1 Set minimum speed less than maximum speed 2 Changeable in the process 3 Recommend MIN_SPEED ≥500. Or the pulse output will complete minimum speed limit (usually 500) at the end of the deceleration. 4 Write MAX_SPEED 0 to realize the soft stop function (the output pulse deceleration stops). Then the enable bit RUN no reset; If write speed back to a large one,
MAX_SPEED	IN	Maximum speed of run Unit: HZ	Dword	0~200000	

					output the pulse until it complete.
TA	IN	Time of ACC/DEC, Unit: ms	Dword	0~10000	Changeable in the process; if TA=0, no settings for acceleration.
RUN	IN	Run enable bit 1: Valid	BOOL	0~1	1 Run only when RUN =1 and E_STOP =0. 2 RUN reset internal when run complete. 3 when E_STOP=1, RUN reset internal.
STATUS	OUT	7 6 5 4 3 2 1 0 Output status byte: Bit0: Parameter configuration error sign 1—Parameter configuration error 0—Parameter configuration normal Bit1: Run sign 1—Running 0—Do not run Bit2: Completion sign 1—Instruction execution completed 0—Instruction execution do not completed Bit3: Busy sign 1—Valid, the axis occupied by other instruction 0—Invalid, executing instruction or execution done.	BYTE	0~255	Bit0: 1 Judge Only to axis parameter 2 No error reported for other parameters, it will set to the nearest reasonable value automatically. 3 Report parameter failure if TA=0 and the maximum acceleration is not set; TD the same
SET_POS	IN	Output pulse. The positive output pulse indicates the positive direction along the X-axis, and the negative pulse number indicates the negative direction along the	DInt	-2147483648 ~ +2147483647	Modifiable during operation. When the new set value is greater than the number of pulses output, the final output pulse will be subject to the new

		X-axis.			set value. When the new set value is less than the number of output pulses, the pulse output will be stopped immediately
ACT_POS	OUT	Absolute coordinate of current axis (1 pulse 1 units coordinate)	DInt	-2147483648 ~ +2147483647	This instruction no error and set the axis number correctly.
ACT_SPEED	OUT	Current run speed	Dword	0~200000	

Two axis circular interpolation relative motion instruction

① Function name: MC_CIRCLE_R



② Function: Circular interpolation between any two axes (set points as relative coordinates).

③ Parameter

Name	I/O	Description	Type	Value range	Note
E_STOP	IN	Emergency stop bit 1: Valid, 0: Invalid	BOOL	0~1	1. Run only when RUN =1 and E_STOP =0. 2. When E_STOP=1, RUN reset internal.
CW	IN	Interpolation of CW or CCW 1: CW 0: CCW	Bool	0/1	Unchangeable in the process

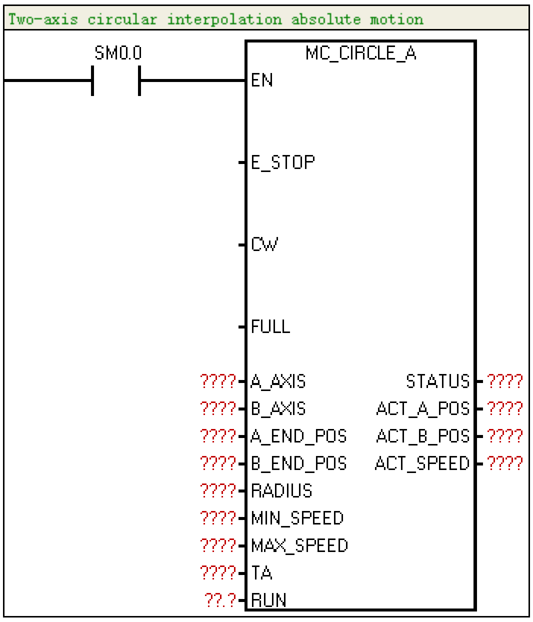
FULL	IN	Full-circle sign 1—Full-circle 0—Arc	Bool	0/1	Unchangeable in the process
A_AXIS	IN	Interpolation A axis number. Interpolation requires two axes, the virtual A-axis and B-axis. Map to the 0, 1, 2, 3 axes of the actual output. This parameter sets which axis the A axis maps to. For example, 3 indicates A axis maps to the three axes (Pulse_3 and Dir_3).	Byte	0~3	Unchangeable in the process
B_AXIS	IN	Interpolation B axis number. Interpolation requires two axes, the virtual A-axis and B-axis. Map to the 0, 1, 2, 3 axes of the actual output. This parameter sets which axis the B axis maps to.	Byte	0~3	Unchangeable in the process
A_END_POS	IN	If FULL = 0, indicates destination coordinate(relative origin) of virtual A-axis If FULL = 1, indicates coordinate (relative starting point) of an arc that is different from the starting point, not the end point.	Dint	-2147483647 ~ +2147483647	1 Default interpolation starting point coordinate(0,0) 2 Unit: pulse, Transform with actual
B_END_POS	IN	If FULL = 0, indicates the coordinates of virtual B axis end point (relative to the starting point). If FULL = 1, indicates the coordinates of the arc at a point different from the starting point, not the ending point.	Dint	-2147483647 ~ +2147483647	1 Interpolation start coordinate default (0,0) 2 Unit: pulse, Transfer with actual displacement Note: Unchangeable in the process

RADIUS	IN	Arc radius 1. There are positive and negative. Positive number: indicates the arc trajectory with radian less than 180°. Negative number: indicates the arc trajectory with a radian greater than 180°. 2. The absolute value of RADIUS represents Arc radius	Dint	|R| <3 x 10^6	1 Unmodifiable in the process 2 Unit: pulse, Transform with actual radius
MIN_ SPEED	IN	Minimum speed of run or stop. Unit: HZ	Dword	500~200000	Unmodifiable in the process
MAX_ SPEED	IN	Maximum speed of run Unit: HZ	Dword	500~200000	Changeable in the process
TA		Time of ACC/DEC, Unit: ms	Dword	0~10000	1 Unmodifiable in the process 2 Calculate acceleration only when startup and TA/TD changes
RUN	IN/OUT	Run enable bit 1: Valid	BOOL	0~1	1 Run only when RUN =1 and E_STOP =0. 2 RUN reset internal when run complete. 3 when E_STOP=1, RUN reset internal.

STATUS	OUT	7 6 5 4 3 2 1 0 Output status byte: Bit0: Parameter configuration error sign 1—parameter configuration error 0—Parameter configuration normal Bit1: Run sign 1—Running 0—Do not run Bit2: Completion sign 1—Instruction execution completed 0—Instruction execution do not completed Bit3: Busy sign 1—Valid, the axis occupied by other instruction 0—Invalid, executing instruction or execution done.	BYTE	0~255	Bit0: 1. Judge Only to axis parameter 2. No error reported for other parameters, it will set to the nearest reasonable value automatically. 3. Report parameter failure if TA=0 and the maximum acceleration is not set; TD the same
ACT_A _POS	OUT	Current position of A axis(relative coordinate, actual output pulse) Indicates relative coordinate of 0	Dint	-2147483647 ~ +2147483647	
ACT_B _POS	OUT	Current position of B axis(relative coordinate, actual output pulse) Indicates relative coordinate of 1	Dint	-2147483647 ~ +2147483647	
ACT _SPEED	OUT	Current run speed	Dword	500~200000	

Two-axis circular interpolation absolute motion instruction

① Function name: MC_CIRCLE_A



② Function: Circular interpolation between any two axes (set points as absolute coordinates).

③ Parameter

Name	I/O	Description	Type	Value range	Note
E_STOP	IN	Emergency stop bit 1: Valid, 0: Invalid	BOOL	0~1	1 Run only when RUN =1 and E_STOP =0. 2 When E_STOP=1, RUN reset internal.
CW	IN	Interpolation of CW or CCW 1: CW 0: CCW	Bool	0/1	Unchangeable in the process
FULL	IN	Full-circle sign 1—Full-circle 0—Arc	Bool	0/1	Unchangeable in the process
A_AXIS	IN	Interpolation A axis number. Interpolation requires two axes, the virtual A-axis and B-axis. Map to the 0, 1, 2, 3 axes of the actual output. This parameter sets which axis the A axis maps to. For example, 3 indicates A axis maps to the three axes (Pulse_3 and Dir_3).	BYTE	0~3	Unchangeable in the process
B_AXIS	IN	Interpolation B axis number. Interpolation requires two	BYTE	0~3	

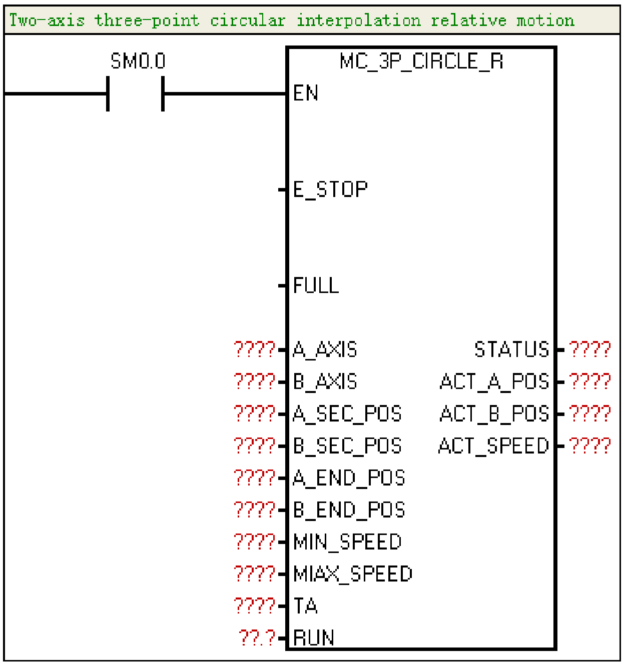
		axes, the virtual A-axis and B-axis. Map to the 0, 1, 2, 3 axes of the actual output. This parameter sets which axis the B axis maps to.			
A_END_POS	IN	If FULL = 0, indicates destination coordinate(relative origin) of virtual A-axis If FULL = 1, indicates coordinate (relative starting point) of an arc that is different from the starting point, not the end point.	DINT	-2147483648 ~ +2147483647	1 Unmodifiable in the process 2 Unit: pulse, Transform with actual displacement
B_END_POS	IN	If FULL = 0, indicates the coordinates of virtual B axis end point (relative to the starting point). If FULL = 1, indicates the coordinates of the arc at a point different from the starting point, not the ending point.	DINT	-2147483648 ~ +2147483647	
RADIUS	IN	Arc radius 1. There are positive and negative. Positive number: indicates the arc trajectory with radian less than 180°. Negative number: indicates the arc trajectory with a radian greater than 180°. 2. The absolute value of RADIUS represents Arc radius	DINT	|R| <3 x 10^6	1 Unmodifiable in the process 2 Unit: pulse, Transform with actual displacement

MIN_SPEED	IN	Minimum speed of run or stop. Unit: HZ	Dword	0~200000	1 Set minimum speed less than maximum speed 2 Changeable in the process 3 Recommend MIN_SPEED ≥500. Or the pulse output will complete minimum speed limit (usually 500) at the end of the deceleration. 4 Write MAX_SPEED 0 to realize the soft stop function (the output pulse deceleration stops). Then the enable bit RUN no reset; If write speed back to a large one, output the pulse until it complete.
MAX_SPEED	IN	Maximum speed of run Unit: HZ	Dword	0~200000	
TA		Time of ACC/DEC, Unit: ms	Dword	0~10000	1 Unmodifiable in the process 2 Calculate acceleration only when startup and TA/TD changes
RUN	IN/OUT	Run enable bit 1: valid	BOOL	0~1	1 Run only when RUN =1 and E_STOP =0. 2 RUN reset internal when run complete. 3 When E_STOP=1, RUN reset internal.

STATUS	OUT	7 6 5 4 3 2 1 0 Output status byte: Bit0: parameter configuration error sign 1—parameter configuration error 0—parameter configuration normal Bit1: Run sign 1—Running 0—Do not run Bit2: Completion sign 1—Instruction execution completed 0—Instruction execution do not completed Bit3: Busy sign 1—Valid, the axis occupied by other instruction 0—Invalid, executing instruction or execution done.	BYTE	0~255	Bit0: 1 Judge Only to axis parameter 2 No error reported for other parameters, it will set to the nearest reasonable value automatically. 3 Report parameter failure if TA=0 and the maximum acceleration is not set; TD the same
ACT_A_POS	OUT	Current position of A axis(relative coordinate, actual output pulse) Indicates relative coordinate of 0 axis when A axis allocate to 0 axis	DINT	-2147483648 ~ +2147483647	
ACT_B_POS	OUT	Current position of B axis(relative coordinate, actual output pulse) Indicates relative coordinate of 1 axis when B axis allocate to 1 axis	DINT	-2147483648 ~ +2147483647	
ACT_SPEED	OUT	Current run speed	Dword	0~200000	

Two-axis three-point circular interpolation relative motion instruction

① Function name: MC_3P_CIRCLE_R



②Function: Circular interpolation between any two axes (set points as relative coordinates).

③ Parameter

Name	I/O	Description	Type	Value range	Note
E_STOP	IN	Emergency stop bit 1: Valid, 0: Invalid	Bool	0~1	1 Run only when RUN =1 and E_STOP =0. 2 When E_STOP=1, RUN reset internal.
FULL	IN	Full-circle sign 1—Full-circle 0—Arc	Bool	0/1	Unchangeable in the process
A_AXIS	IN	Interpolation A axis number. Interpolation requires two axes, the virtual A-axis and B-axis. Map to the 0, 1, 2, 3 axes of the actual output. This parameter sets which axis the A axis maps to. For example, 3 indicates A axis maps to the three axes (Pulse_3 and Dir_3).	BYTE	0~3	Unchangeable in the process
B_AXIS	IN	Interpolation B axis number. Interpolation requires two axes, the virtual A-axis	BYTE	0~3	Unchangeable in the process

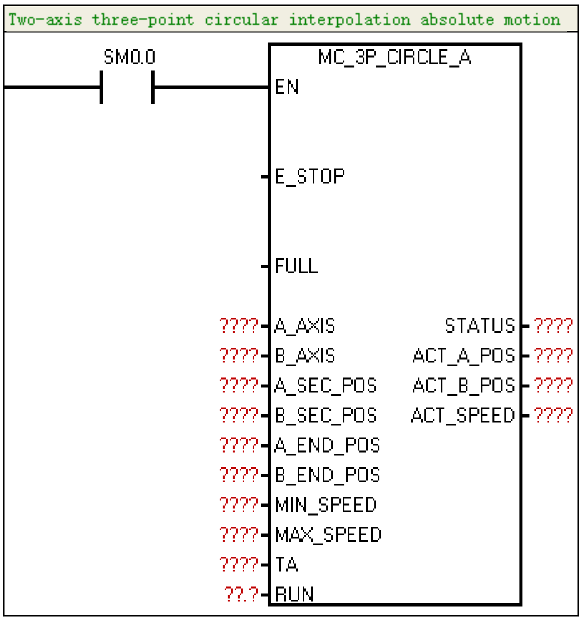
		and B-axis. Map to the 0, 1, 2, 3 axes of the actual output. This parameter sets which axis the B axis maps to.			
A_SEC_POS	IN	Determine arc second point of the A axis coordinate (use for certain the second point in the direction of an arc or circle at three points, relative coordinate)	DINT	-2147483648 ~ +2147483647	1 Interpolation starting coordinate is first point of the arc(relative instruction starting coordinate (0, 0)) 2 Unmodifiable in the process 3 Unit: pulse, Transform with actual replacement 4 Interpolation starting coordinate default(0,0)
B_SEC_POS	IN	Determine arc second point of the B axis coordinate (use for certain the second point in the direction of an arc or circle at three points, relative coordinate)	DINT	-2147483648 ~ +2147483647	
A_END_POS	IN	Determine arc third point of axis A coordinate. If FULL = 0, it is also the terminal coordinate. (used for determine arc or circle at three points, relative coordinate)	DINT	-2147483648 ~ +2147483647	
B_END_POS	IN	Determine arc third point of axis B coordinate. If FULL = 0, it is also the terminal coordinate. (used for determine arc or circle at three points, relative coordinate)	DINT	-2147483648 ~ +2147483647	
MIN_SPEED	IN	Minimum speed of run or stop. Unit: HZ	Dword	0~200000	1 Set minimum speed less than maximum speed 2 Changeable in the process 3 Recommend MIN_SPEED ≥500. Or the pulse output will complete minimum speed limit (usually 500)
MAX_SPEED	IN	Maximum speed of run Unit: HZ	Dword	0~200000	

					at the end of the deceleration. 4 Write MAX_SPEED 0 to realize the soft stop function (the output pulse deceleration stops). Then the enable bit RUN no reset; If write speed back to a large one, output the pulse until it complete.
TA		Time of ACC/DEC, Unit: ms	Dword	0~10000	1 Unchangeable in the process 2 Calculate acceleration only when startup and TA/TD changes
RUN	IN/OUT	Run enable bit 1: valid	Bool	0~1	1 Run only when RUN =1 and E_STOP =0. 2 RUN reset internal when run complete. 3 When E_STOP=1, RUN reset internal.
STATUS	OUT	7 6 5 4 3 2 1 0 Output status byte: Bit0: parameter configuration error sign 1—parameter configuration error 0—parameter configuration normal Bit1: Run sign 1—Running 0—Do not run Bit2: Completion sign 1—Instruction execution completed 0—Instruction execution	BYTE	0~255	Bit0: 1 Judge Only to axis parameter 2 No error reported for other parameters, it will set to the nearest reasonable value automatically. 3 Report parameter failure if TA=0 and the maximum acceleration is not set; TD the same

		do not completed Bit3: Busy sign 1—Valid, the axis occupied by other instruction 0—Invalid, executing instruction or execution done.			
ACT_A_POS	OUT	Current position of A axis(relative coordinate, actual output pulse) Indicates relative coordinate of 0 axis when A axis allocate to 0 axis	DINT	-2147483648 ~ +2147483647	
ACT_B_POS	OUT	Current position of B axis(relative coordinate, actual output pulse) Indicates relative coordinate of 1 axis when B axis allocate to 1 axis	DINT	-2147483648 ~ +2147483647	
ACT_SPEED	OUT	Current actual speed	Dword	0~200000	

Two-axis three-point circular interpolation absolute motion instruction

① Function name: MC_3P_CIRCLE_A



② Function: Circular interpolation between any two axes (set points as absolute coordinates).

③ Parameter

Name	I/O	Description	Type	Value range	Note
E_STOP	IN	Emergency stop bit 1: Valid, 0: Invalid	BOOL	0~1	1 Run only when RUN =1 and E_STOP =0. 2 When E_STOP=1, RUN reset internal.
FULL	IN	Full-circle sign 1—full-circle 0—Arc	Bool	0/1	Unchangeable in the process
A_AXIS	IN	Interpolation A axis number. Interpolation requires two axes, the virtual A-axis and B-axis. Map to the 0, 1, 2, 3 axes of the actual output. This parameter sets which axis the A axis maps to. For example, 3 indicates A axis maps to the three axes (Pulse_3 and Dir_3).	BYTE	0~3	Unchangeable in the process
B_AXIS	IN	Interpolation B axis number. Interpolation requires two axes, the virtual A-axis and B-axis. Map to the 0, 1, 2, 3 axes of the actual output. This parameter sets which axis the B axis maps to.	BYTE	0~3	Unchangeable in the process
A_SEC_POS	IN	Determine arc second point of the A axis coordinate (use for certain the second point in the direction of an arc or circle at three points, relative coordinate)	DINT	-2147483648 ~ +2147483647	1. Interpolation starting coordinate is first point of the arc (relative instruction starting coordinate(0,0)) 2. Unmodifiable in the process 3. Unit: pulse 4. Transform with actual replacement
B_SEC_POS	IN	Determine arc second point of the B axis coordinate (use for certain the second point in the direction of an arc or circle at three points, relative coordinate)	DINT	-2147483648 ~ +2147483647	

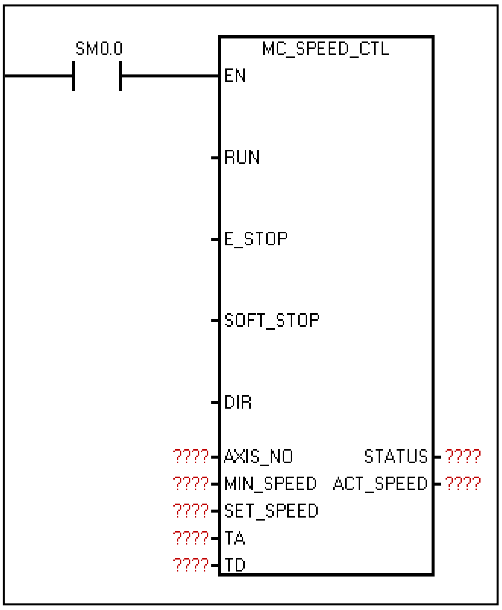
A_END_POS	IN	Determine arc third point of axis A coordinate. If FULL = 0, it is also the terminal coordinate. (used for determine arc or circle at three points, relative coordinate)	DINT	-2147483648 ~ +2147483647	
B_END_POS	IN	Determine arc third point of axis B coordinate. If FULL = 0, it is also the terminal coordinate. (To determine arc or circle at three points, relative coordinate)	DINT	-2147483648 ~ +2147483647	
MIN_SPEED	IN	Minimum speed of run or stop. Unit: HZ	Dword	0~200000	1. Set minimum speed less than maximum speed 2. Changeable in the process 3. Recommend MIN_SPEED ≥500. Or the pulse output will complete minimum speed limit (usually 500) at the end of the deceleration. 4. Write MAX_SPEED 0 to realize the soft stop function (the output pulse deceleration stops). Then the enable bit RUN no reset; if write speed back to a large one, output the pulse until it complete.
MAX_SPEED	IN	Maximum speed of run Unit: HZ	Dword	0~200000	
TA		Time of ACC/DEC, Unit: ms	Dword	0~10000	1. Unmodifiable in the process 2. Calculate acceleration

					only when startup and TA/TD changes
RUN	IN/OUT	Run enable bit 1: Valid	BOOL	0~1	1. Run only when RUN =1 and E_STOP =0. 2. RUN reset internal when run complete. 3. When E_STOP=1, RUN reset internal.
STATUS	OUT	7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 Output status byte: Bit0: parameter configuration error sign 1—Parameter configuration error 0—Parameter configuration normal Bit1: Run sign 1—Running 0—Do not run Bit2: Completion sign 1—Instruction execution completed 0—Instruction execution do not completed Bit3: Busy sign 1—Valid, the axis occupied by other instruction 0—Invalid, executing instruction or execution done.	BYTE	0~255	Bit0: 1. Judge Only to axis parameter 2. No error reported for other parameters, it will set to the nearest reasonable value automatically. 3. Report parameter failure if TA=0 and the maximum acceleration is not set; TD the same
ACT_A_POS	OUT	Current position of A axis(relative coordinate, actual output pulse) Indicates relative coordinate of 0 axis when A axis allocate to 0 axis	DINT	-2147483648 ~ +2147483647	
ACT_B_POS	OUT	Current position of B axis(relative coordinate, actual output pulse)	DINT	-2147483648 ~ +2147483647	

		Indicates relative coordinate of 1 axis when B axis allocate to 1 axis			
ACT_SPEED	OUT	Current actual speed	Dword	0~200000	

Speed control instruction

① Function name: MC_SPEED_CTL



② Function: Control the frequency of uniaxial output pulse, the output pulse frequency (speed) can be changed any time. When receive the soft stop command, it will automatically slow down to stop. When receive the emergency stop command, the pulse output will stop immediately without any deceleration.

③ Parameter

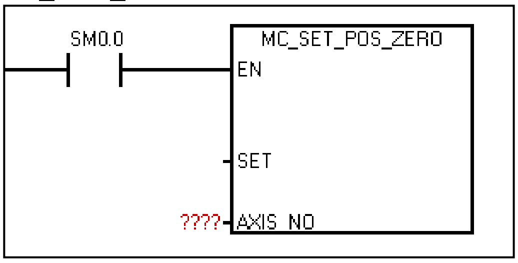
Name	I/O	Description	Type	Value range	Note
RUN	IN	Run enable bit 1: valid	BOOL	0~1	1. Run only when RUN =1 and E_STOP =0. 2. RUN reset internal when run complete. 3. When E_STOP=1, RUN reset internal.

E_STOP	IN	Emergency stop bit 1: valid, 0: invalid	BOOL	0~1	1. Run only when RUN =1 and E_STOP =0. 2. When E_STOP=1, RUN reset internal.
SOFT_STOP	IN	Soft stop bit1: valid 0: invalid Output pulse decelerate to stop when receive the instruction	Bool	0/1	
DIR	IN	Pulse direction bit	Bool	0/1	Unmodifiable in the process
AXIS_NO	IN	Set axis number, 0/1/2/3	Byte	0~3	Unmodifiable in the process
MIN_SPEED	IN	Minimum speed of run or stop. Unit: HZ	Dword	0~200000	1. Minimum speed should less than setting speed 2. Modifiable in the process
SET_SPEED	IN	Setting speed, output pulse will accelerate or decelerate to this speed before stop command	Dword	0~200000	
TA	IN	Time of ACC Unit: ms	Dword	0~10000	Modifiable in the process; Calculate acceleration only when startup and TA/TD changes
TD	IN	Time of ACC, from setting time to minimum time Unit: ms	Dword	0~10000	Unmodifiable in the process

STATUS	OUT	7 6 5 4 3 2 1 0 Output status byte: Bit0: parameter configuration error sign 1—parameter configuration error 0—parameter configuration normal Bit1: Run sign 1—Running 0—Do not run Bit2: Completion sign 1—Instruction execution completed 0—Instruction execution do not completed Bit3: Busy sign 1—Valid, the axis occupied by other instruction 0—Invalid, executing instruction or execution done.	BYTE	0~255	Bit0: 1 Judge Only to axis parameter 2 No error reported for other parameters, it will set to the nearest reasonable value automatically. 3 Report parameter failure if TA=0 and the maximum acceleration is not set; TD the same
ACT _SPEED	OUT	Current actual speed	Dword	0~200000	Current actual speed

Software back to zero instruction

① Function name: MC_SET_POS_ZERO



② Function: Reset absolute coordinate

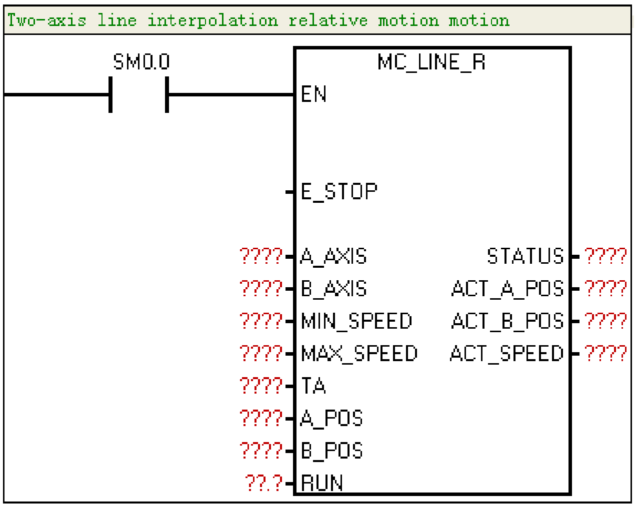
Note: Call the instruction when the machine moves to a certain position, which is equivalent to setting the origin of the axis at that position. Each time you call the "read absolute coordinates" command, you get a coordinate value relative to that point.

③ Parameter

Name	I/O	Description	Type	Value range	Note
SET	IN	Enable reset bit Set absolute coordinate 0 at SET rising edge, set SET bit 0, then set 1 for each call	Bool	Range: 0~1	
AXIS_NO	IN	Set axis number: 0/1/2/3	Byte	Range: 0~3	

Two-axis line interpolation relative motion instruction

① Function name: MC_LINE_R



② Function: Line interpolation between any two axes or any plane area (set points as relative coordinates).

③ Parameter

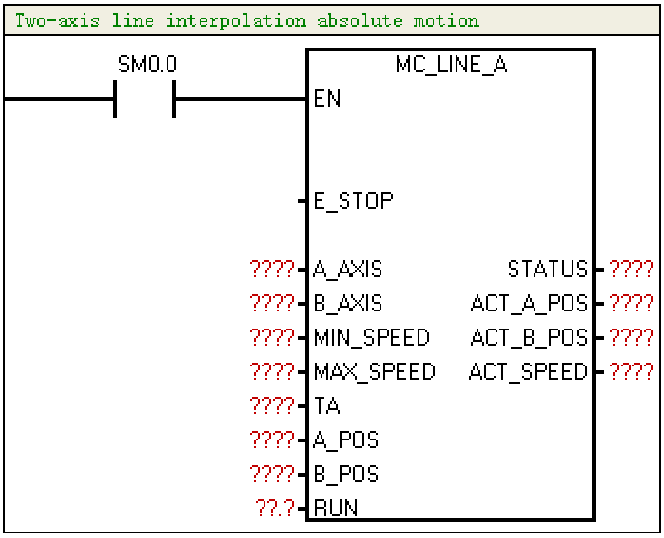
Name	I/O	Description	Type	Value range	Note
E_STOP	IN	Emergency stop bit 1: Valid, 0: Invalid	BOOL	0~1	1 Run only when RUN =1 and E_STOP =0. 2 When E_STOP=1, RUN reset internal.
A_AXIS	IN	Interpolation A axis number. Interpolation requires two axes, the virtual A-axis and B-axis. Map to the 0, 1, 2, 3 axes of the actual output. This parameter sets which axis the A axis maps to. For example, 3 indicates A axis maps to the three axes (Pulse_3 and Dir_3).	BYTE	0~3	Unmodifiable in the process
B_AXIS	IN	Interpolation B axis number. Interpolation requires two axes, the virtual A-axis and B-axis. Map to the 0, 1, 2, 3	BYTE	0~3	

		axes of the actual output. This parameter sets which axis the B axis maps to.			
MIN_SPEED	IN	Minimum speed of run or stop. Unit: HZ	Dword	0~200000	1 Set minimum speed less than maximum speed 2 Changeable in the process 3 Recommend MIN_SPEED ≥500. Or the pulse output will complete minimum speed limit (usually 500) at the end of the deceleration. 4 Write MAX_SPEED 0 to realize the soft stop function (the output pulse deceleration stops). Then the enable bit RUN no reset; If write speed back to a large one, output the pulse until it complete.
MAX_SPEED	IN	Maximum speed of run Unit: HZ	Dword	0~200000	
TA	IN	Time of ACC Unit: ms	Dword	0~10000	Modifiable in the process Calculate acceleration only when startup and TA/TD changes
A_POS	IN	Virtual A axis destination (relative) coordinate.	DINT	-2147483648 ~ +2147483647	1 Unmodifiable in the process 2 Unit: pulse
B_POS	IN	Virtual B axis destination (relative) coordinate.	DINT	-2147483648 ~ +2147483647	
RUN	IN/OUT	Run enable bit 1: valid	BOOL	0~1	1 Run only when RUN =1 and E_STOP =0. 2 RUN reset internal when run complete. 3 When E_STOP=1, RUN reset internal.

STATUS	OUT	| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Output status byte: Bit0: parameter configuration error sign 1—Parameter configuration error 0—Parameter configuration normal Bit1: Run sign 1—Running 0—Do not run Bit2: Completion sign 1—Instruction execution completed 0—Instruction execution do not completed Bit3: Busy sign 1—Valid, the axis occupied by other instruction 0—Invalid, executing instruction or execution done.	BYTE	0~255	Bit0: 1 Judge Only to axis parameter 2 No error reported for other parameters, it will set to the nearest reasonable value automatically. 3 Report parameter failure if TA=0 and the maximum acceleration is not set; TD the same
ACT_A_POS	OUT	Current position of A axis(relative coordinate, actual output pulse) Indicates relative coordinate of 0 axis when A axis allocate to 0 axis	DINT	-2147483648 ~ +2147483647	
ACT_B_POS	OUT	Current position of B axis(relative coordinate, actual output pulse) Indicates relative coordinate of 1 axis when B axis allocate to 1 axis	DINT	-2147483648 ~ +2147483647	
ACT_SPEED	OUT	Current actual speed	Dword	0~200000	

Two-axis line interpolation absolute motion instruction

① Function name: MC_LINE_A



② Function: Line interpolation between any two axes or any plane area (set points as absolute coordinates).

③ Parameter

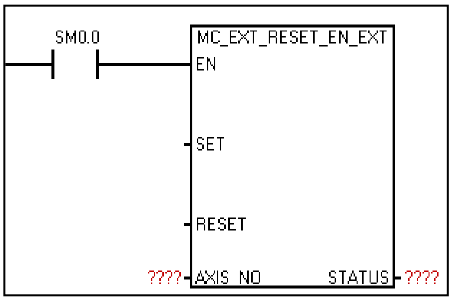
Name	I/O	Description	Type	Value range	Note
E_STOP	IN	Emergency stop bit 1: Valid, 0: Invalid	BOOL	0~1	1 Run only when RUN =1 and E_STOP =0. 2 When E_STOP=1, RUN reset internal.
A_AXIS	IN	Interpolation A axis number. Interpolation requires two axes, the virtual A-axis and B-axis. Map to the 0, 1, 2, 3 axes of the actual output. This parameter sets which axis the A axis maps to. For example, 3 indicates A axis maps to the three axes (Pulse_3 and Dir_3).	BYTE	0~3	Unmodifiable in the process
B_AXIS	IN	Interpolation B axis number. Interpolation requires two axes, the virtual A-axis and B-axis. Map to the 0, 1, 2, 3 axes of the actual output. This parameter sets which axis the B axis maps to.	BYTE	0~3	

MIN_SPEED	IN	Minimum speed of run or stop. Unit: HZ	Dword	0~200000	1 Set minimum speed less than maximum speed 2 Changeable in the process 3 Recommend MIN_SPEED ≥500. Or the pulse output will complete minimum speed limit (usually 500) at the end of the deceleration. 4 Write MAX_SPEED 0 to realize the soft stop function (the output pulse deceleration stops). Then the enable bit RUN no reset; If write speed back to a large one, output the pulse until it complete.
MAX_SPEED	IN	Maximum speed of run Unit: HZ	Dword	0~200000	
TA	IN	Time of ACC Unit: ms	Dword	0~10000	Modifiable in the process Calculate acceleration only when startup and TA/TD changes
A_POS	IN	Virtual A axis destination (relative) coordinate.	DINT	-2147483648 ~ +2147483647	1 Unmodifiable in the process 2 Unit: pulse
B_POS	IN	Virtual B axis destination (relative) coordinate.	DINT	-2147483648 ~ +2147483647	
RUN	IN/OUT	Run enable bit 1: valid	BOOL	0~1	1 Run only when RUN =1 and E_STOP =0. 2 RUN reset internal when run complete. 3 when

					E_STOP=1, RUN reset internal.
STATUS	OUT	7 6 5 4 3 2 1 0 Output status byte: Bit0: parameter configuration error sign 1—Parameter configuration error 0—Parameter configuration normal Bit1: Run sign 1—Running 0—Do not run Bit2: Completion sign 1—Instruction execution completed 0—Instruction execution do not completed Bit3: Busy sign 1—Valid, the axis occupied by other instruction 0—Invalid, executing instruction or execution done.	BYTE	0~255	Bit0: 1 Judge Only to axis parameter 2 No error reported for other parameters, it will set to the nearest reasonable value automatically. 3 Report parameter failure if TA=0 and the maximum acceleration is not set; TD the same
ACT_A_POS	OUT	Current position of A axis(relative coordinate, actual output pulse) Indicates relative coordinate of 0 axis when A axis allocate to 0 axis	DINT	-2147483648 ~ +2147483647	
ACT_B_POS	OUT	Current position of B axis(relative coordinate, actual output pulse) Indicates relative coordinate of 1 axis when B axis allocate to 1 axis	DINT	-2147483648 ~ +2147483647	
ACT_SPEED	OUT	Current actual speed	Dword	0~200000	

External reset coordinate enable instruction II

①Function name: MC_EXT_RESET_EN_EXT



②Function: Set if enable external IO reset absolute coordinate

Note: Correspondence of axis number and external reset signal

Axis 0 ——I0.2 (HSC0, SM37.0)
Axis 1 ——I1.0 (HSC1, SM47.0)
Axis 2 ——I1.4 (HSC2, SM57.0)
Axis 3 ——I0.5 (HSC4, SM147.0)

③ Parameter

Name	I/O	Description	Type	Value range	Note
SET	IN	Enable reset bit Set absolute coordinate 0 at SET rising edge, set SET 0, then set 1 for each call	Bool	0~1	
RESET	IN	RESET rising edge, forbid to enable external reset, RESET then set 0 for each call	Bool	0~1	
AXIS_NO	IN	Set axis number, 0/1/2/3	Byte	0~3	
STATUS	OUT	1: Reset complete 0: Reset uncompleted	Byte	0~1	

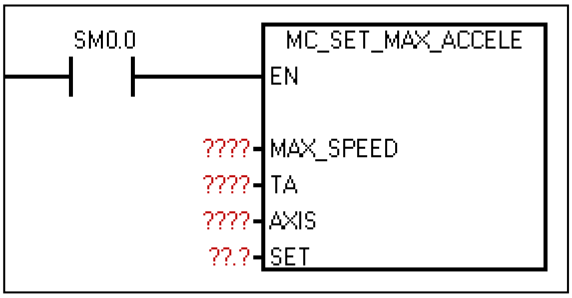
④ Explanation

Use 0 axis to call this instruction. After the SET rising edge enables the external reset function and I0.2 detect the "effective reset signal", the system resets the axis 0 absolute coordinate, and also the STATUS setting reset complete. After the RESET rising edge prohibits the external reset function, even if I0.2 detect the "effective reset signal", the system will not reset the 0 axis absolute coordinate, and the STATUS reset instruct is non-reset state.

Note
The so-called "effective reset signal", the reset signal of each axis corresponding to an external IO, and the corresponding register set its effective level. For example, 0 axis corresponds to I0.2, and HSCO control register SM37.0 set the effective reset level of 0 axis. When set to 0, the effective reset signal of 0 axis is in I0.2 high level. When set to 1, the effective reset signal of axis 0 is in I0.2 low level; This setting is only effective if and only if the corresponding high-speed counter (0 axis corresponds to HSC0) is enabled, otherwise (no high-speed counter is enabled) the system default high level is the effective reset signal. If 0 axis, I0.2 high level is the effective reset signal. In the same way for other axes, the corresponding relation of the relevant control of each axis is shown in ② of this section.

Set maximum acceleration instruction

① Function name: MC_SET_MAX_ACCELE



② Function: Set maximum acceleration (= MAX_SPEED/TA; TA≠0), if the instruction isn't called, then the maximum acceleration has not been set.

③ Parameter

Name	I/O	Description	Type	Value range	Note
MAX_SPEED	IN	Maximum speed of long-axis run Unit: HZ	Dword		Modifiable in the process
TA	IN	Time of ACC Unit: ms	Dword	0~10000	Modifiable in the process Calculate acceleration only when startup and TA/TD changes
AXIS_NO	IN	Set axis number, 0/1/2/3	Byte		
SET	IN	Enable reset bit Set absolute coordinate 0 at SET rising edge, SET set 0, then set 1 for each call	Bool		

④ Explanation

If set the X-axis parameter TA=0, or the X-axis does not call this instruction, it is considered that the X-axis does not set maximum acceleration; Otherwise, MAX_ACCELE=MAX_SPEED/TA is considered to be set as the X-axis maximum acceleration. The significance of this instruction is:

a. Set an appropriate acceleration to limit the acceleration of each instruction on an axis.

Such as PTP instruction, set AXIS_NO=0, MIN_SPEED=1000, MAX_SPEED=11000, TA=500, then in theory the acceleration of PTP motion is 20HZ/ms (= (MAX_SPEED-MIN_SPEED)/TA); if 0 axis call MC_SET_MAX_ACCELE and set maximum acceleration 15HZ/ms, the PTP actual acceleration is 15HZ/ms (the same for MC_LINE_R and MC_CIRCLE_R)

b. An instruction on a certain axis is to obtain the maximum acceleration to run

If the PTP instruction run at the maximum acceleration, call MC_SET_MAX_ACCELE instruction on the coaxes to set the maximum acceleration (the parameter TA of MC_SET_MAX_ACCELE instruction cannot be 0, otherwise cannot obtain the maximum acceleration), and simultaneously set the parameter TA of PTP instruction to 0.

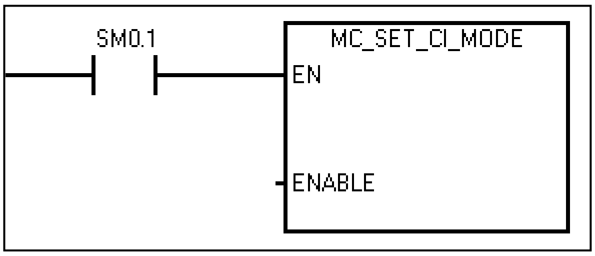
If the maximum acceleration is not set and the PTP instruction TA=0, the alarm parameter fails on the PTP instruction. (Same for MC_LINE_R and MC_CIRCLE_R)

Note: For the biaxial instructions MC_LINE_R and MC_CIRCLE_R, if both set the maximum acceleration, the smaller one will be taken as the maximum acceleration of the biaxial system. If only one axis is set with the maximum acceleration, then take it as the maximum acceleration of the biaxial system. If the maximum acceleration is not set on both axes, the biaxial system has no limitation for acceleration.

Also: After MAX_SPEED, TA, AXIS_NO have been determined, give SET a rising edge, so these parameters are updated and effective.

Set continuous interpolation instruction

① Function name: MC_SET_CI_MODE



② Function: set if enable continuous interpolation function

③ Parameter

Name	I/O	Description	Type	Value range	Note
ENABLE	IN	0: Turn off continuous interpolation 1: Enable continuous interpolation	Bool	0~1	Do not modify in continuous interpolation process

④ Explanation

When need continuous interpolation function, call this instruction in PLC program first to activate the continuous interpolation function, and then call each interpolation instruction successively according to the trajectory sequence. Similarly, the RUN bit of each instruction must be activated according to the trajectory sequence. If not call this instruction, default as discontinuous interpolation movement.

Note
Note1: When one axis call the motion control instruction, the function of recovering normal IO can be restored only when the axis is reenergized and reset or the CPU is stopped. **Note2**: Calculation of line interpolation instruction(A_POS, B_POS), circular interpolation (A_END_POS, B_END_POS, RADIUS): POS-------Pulses needed in interpolation instruction;

L -------Absolute value on coordinate actual point, unit: mm;

S -------Lead screw on motion axis, unit: mm;

M------- The subdivision of the stepping driver or the resolution of the servo drive (the number of pulses required for each turn of the motor);

Formula: POS = L*M÷S(unit: pulse number)

Note3: When TA≠0, acceleration=(MAX_SPEED-MIN_SPEED)/TA(limited to the set maximum acceleration); If TA=0, use instruction MC_SET_MAX_ACCELE to set the maximum acceleration; if it is not set, report the parameter failure. The same for TD. For the biaxial instruction, if both axes set maximum acceleration, take the smaller one as system acceleration.

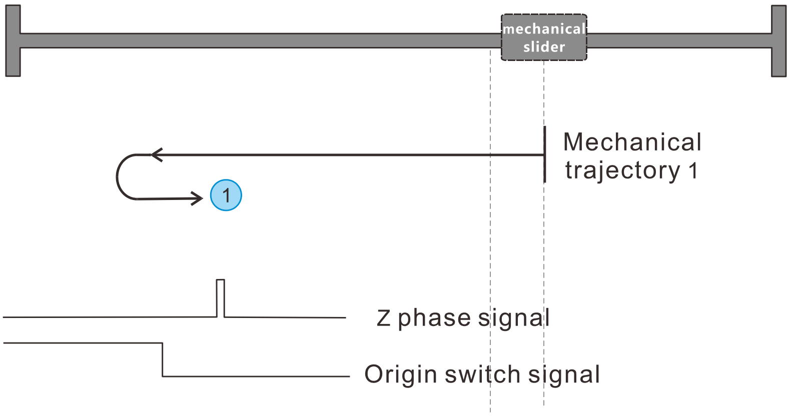
E.4 Homing Mode Illustration

14 origin modes in all, select by your own request to precision and application

Mode	Definition
1	Refer to negative origin switch and Z phase signal origin mode
2	Refer to positive origin switch and Z phase signal origin mode
3	Only refer to origin mode of negative origin switch
4	Only refer to origin mode of positive origin switch
5	Only refer to origin mode of Z phase signal(negative back to zero)
6	Only refer to origin mode of Z phase signal(positive back to zero)
7	Refer to origin mode of origin switch , Z phase signal and positive limit(adopt Z phase signal of positive origin switch left of left edge)
8	Refer to origin mode of origin switch , Z phase signal and positive limit(adopt Z phase signal of positive origin switch right of left edge)
9	Refer to origin mode of origin switch , Z phase signal and positive limit(adopt Z phase signal of positive origin switch left of right edge)
10	Refer to origin mode of origin switch , Z phase signal and positive limit(adopt Z phase signal of positive origin switch right of right edge)
11	Refer to origin mode of origin switch , Z phase signal and negative limit(adopt Z phase signal of positive origin switch right of right edge)
12	Refer to origin mode of origin switch , Z phase signal and negative limit(adopt Z phase signal of positive origin switch left of right edge)
13	Refer to origin mode of origin switch , Z phase signal and negative limit(adopt Z phase signal of positive origin switch right of left edge)
14	Refer to origin mode of origin switch , Z phase signal and negative limit(adopt Z phase signal of positive origin switch left of left edge)

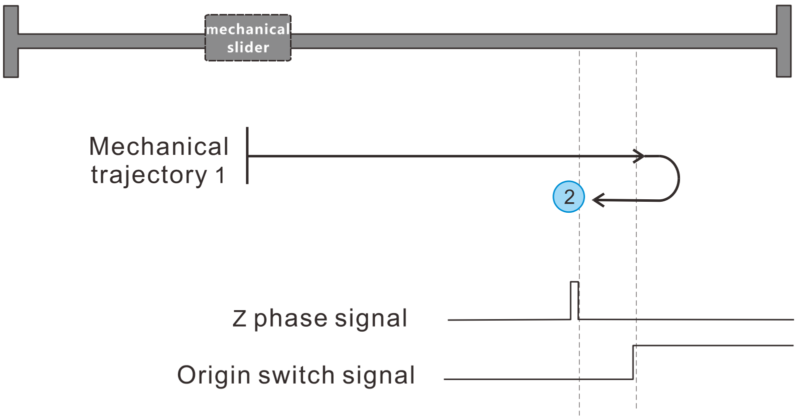
Regardless of the initial position of the machine, when the equipment (origin switch, positive route limit switch, negative route limit switch) installed, the origin of the equipment sought by the servo is always unique. The vertical line “|” in the schematic diagram of the following modes represents the initial position, and the circle“X”represents the origin position.

Origin mode 1: Refer to negative origin switch and Z phase signal origin mode



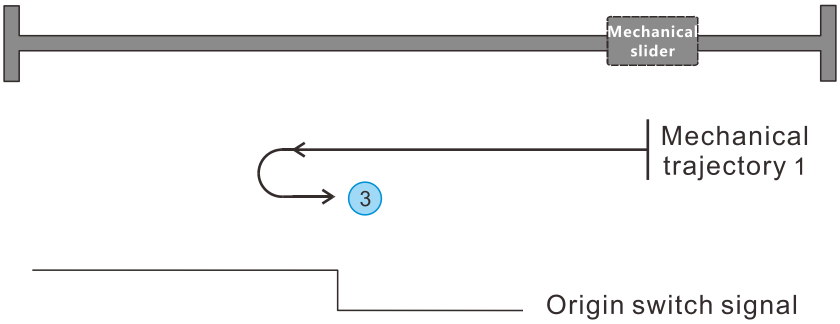
The origin switch located in the negative mechanical direction. The machine moves towards the origin switch, decelerates to stop after detecting the origin switch, then reverses to exit the origin switch, finds the next Z phase signal of the motor and records it as the origin, the motor stops immediately.

Origin mode 2: Refer to positive origin switch and Z phase signal origin mode



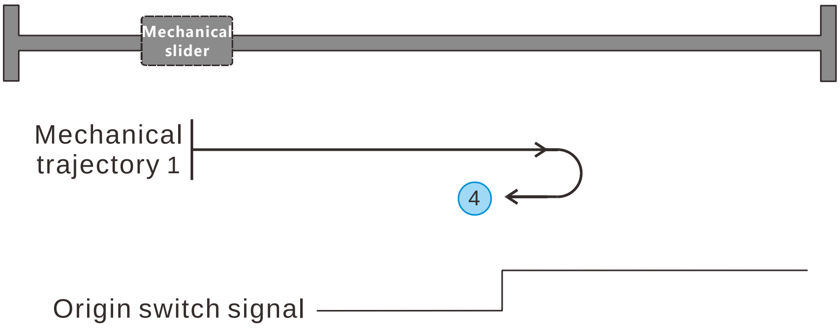
The origin switch located in the positive mechanical direction. The machine moves towards the origin switch, decelerates to stop after detecting the origin switch, then reverses to exit the origin switch, finds the next Z phase signal of the motor and records it as the origin, the motor stops immediately.

Origin mode 3: Only refer to origin mode of negative origin switch



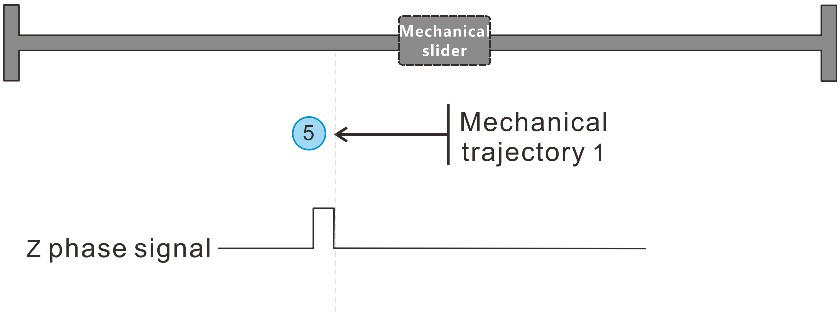
The origin switch located in the negative mechanical direction. The machine moves towards the origin switch, decelerates to stop after detecting the origin switch, and then reverses to exit the origin switch. Find and record origin switch signal falling edge as the origin, and the motor stops immediately.

Origin mode 4: Only refer to origin mode of positive origin switch



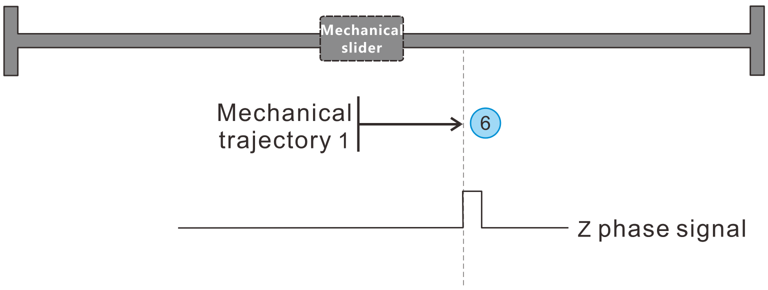
The origin switch located in the positive mechanical direction. The machine moves towards the origin switch, decelerates to stop after detecting the origin switch, and then reverses to exit the origin switch. Find and record origin switch signal falling edge as the origin, and the motor stops immediately.

Origin mode 5: Only refer to origin mode of Z phase signal (negative back to zero)



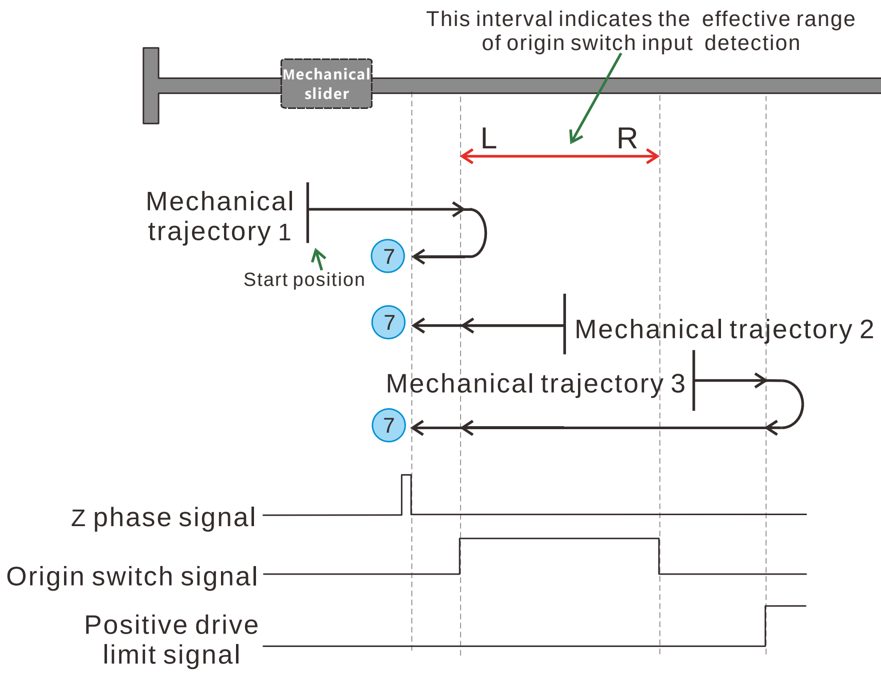
The motor moves along the negative direction from the current position and records it as the origin when the next Z phase signal is found.

Origin mode 6: Only refer to origin mode of Z phase signal (positive back to zero)



The motor moves along the positive direction from the current position and records it as the origin when the next Z phase signal is found.

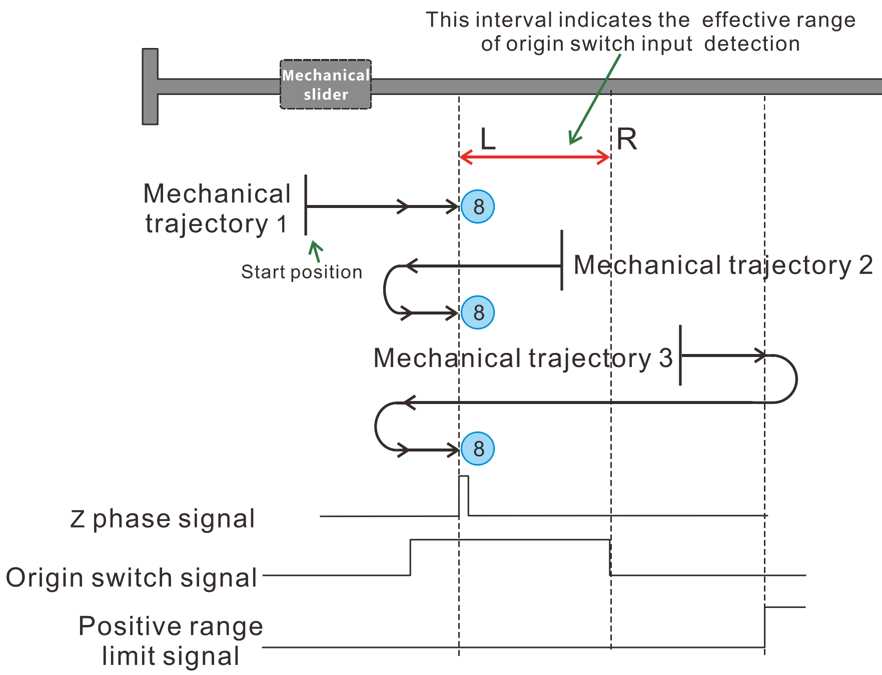
Origin mode 7: Refer to origin mode of origin switch, Z phase signal and positive limit (adopt Z phase signal of positive origin switch left of left edge)



As shown in the figure above, the mechanical slider slides in the positive limit direction (positive direction), and Z in the left position along the left side of the origin switch signal, that is, outside the effective range of the origin switch signal.

When the machine is within the origin switch range (mechanical motion trajectory 2), the origin can be found by directly running in the negative direction. When the machine is outside the origin switch (mechanical motion track 1 and 3), the machine runs in a constant direction (positive direction) toward the limit switch. According to the detected sequence of the origin switch and limit switch, find the movement track and the origin.

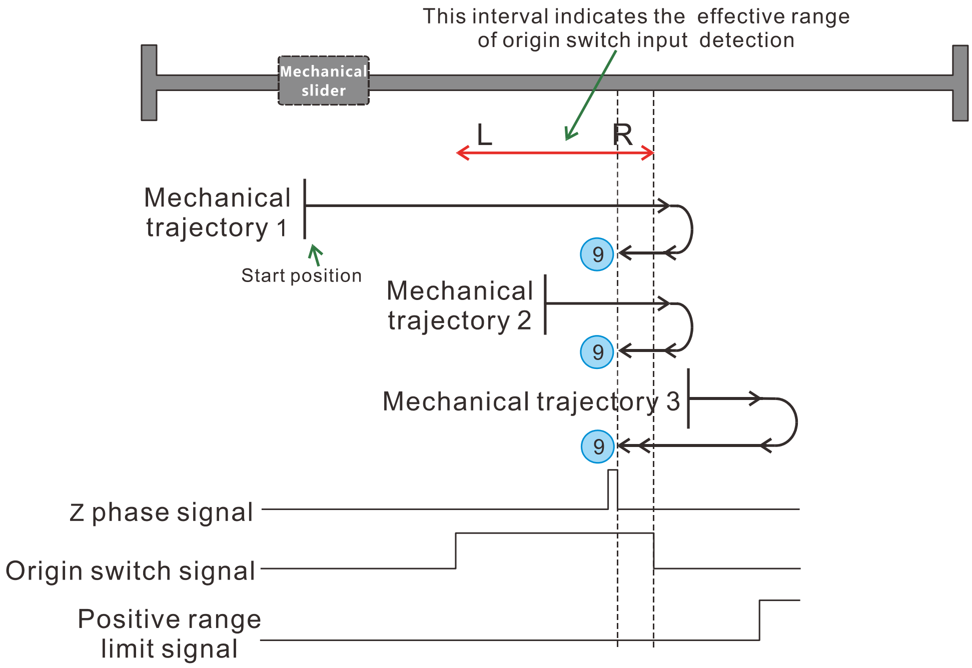
Origin mode 8: Refer to origin mode of origin switch, Z phase signal and positive limit (adopt Z phase signal of positive origin switch right of left edge)



As shown in the figure above, the mechanical slider slides in the positive limit direction (positive direction), and Z in the left position along the right side of the origin switch signal, the effective range of the origin switch signal.

When the machine is within the origin switch range (mechanical motion trajectory 2), the origin can be found by directly running in the negative direction. When the machine is outside of the origin switch (mechanical motion track 1 and 3), the machine runs in a constant direction (positive direction) toward the limit switch. According to the detected sequence of the origin switch and limit switch, find the movement track and the origin.

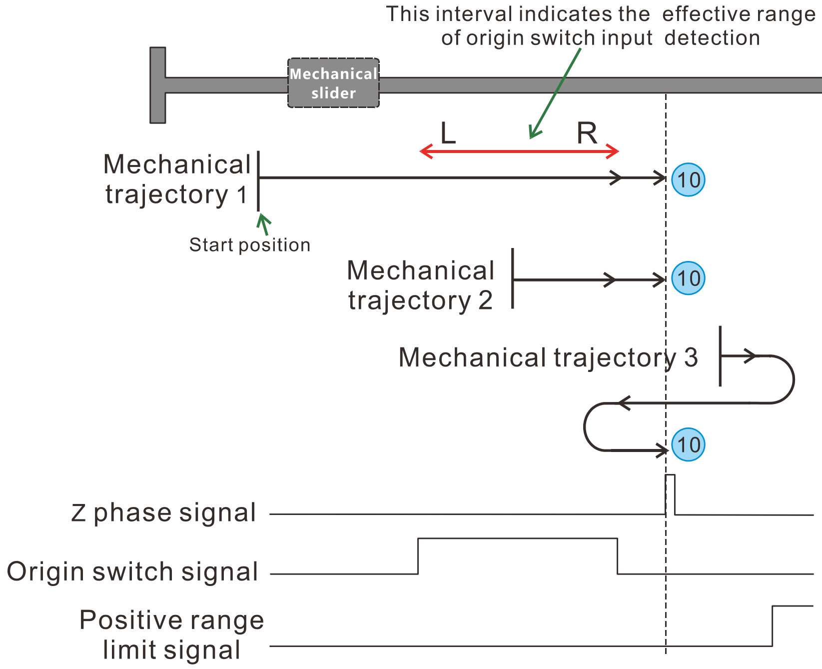
Origin mode 9: Refer to origin mode of origin switch, Z phase signal and positive limit (adopt Z phase signal of positive origin switch left of right edge)



As shown in the figure above, the mechanical slider slides in the positive limit direction (positive direction), and Z in the right position along the left side of the origin switch signal, the effective range of the origin switch signal.

When the machine is within the origin switch range (mechanical motion trajectory 2), the origin can be found by directly running in the negative direction. When the machine is outside of the origin switch (mechanical motion track 1 and 3), the machine runs in a constant direction (positive direction) toward the limit switch. According to the detected sequence of the origin switch and limit switch, find the movement track and the origin.

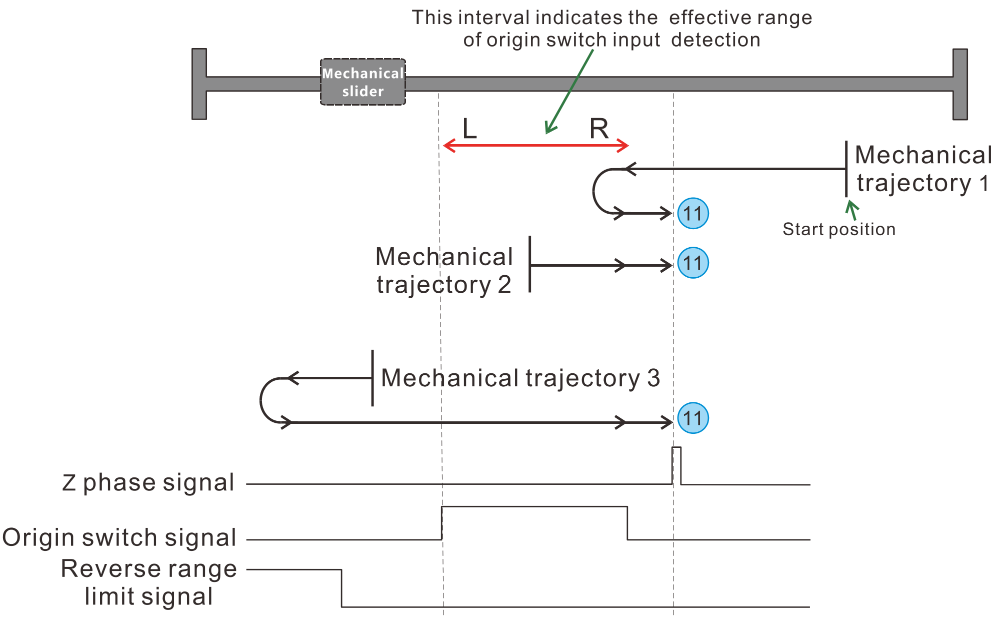
Origin mode 10: Refer to origin mode of origin switch, Z phase signal and positive limit (adopt Z phase signal of positive origin switch right of right edge)



As shown in the figure above, the mechanical slider slides in the positive limit direction (positive direction), and Z in the right position along the right side of the origin switch signal, the effective range of the origin switch signal.

When the machine is within the origin switch range (mechanical motion trajectory 2), the origin can be found by directly running in the negative direction. When the machine is outside of the origin switch (mechanical motion track 1 and 3), the machine runs in a constant direction (positive direction) toward the limit switch. According to the detected sequence of the origin switch and limit switch, find the movement track and the origin.

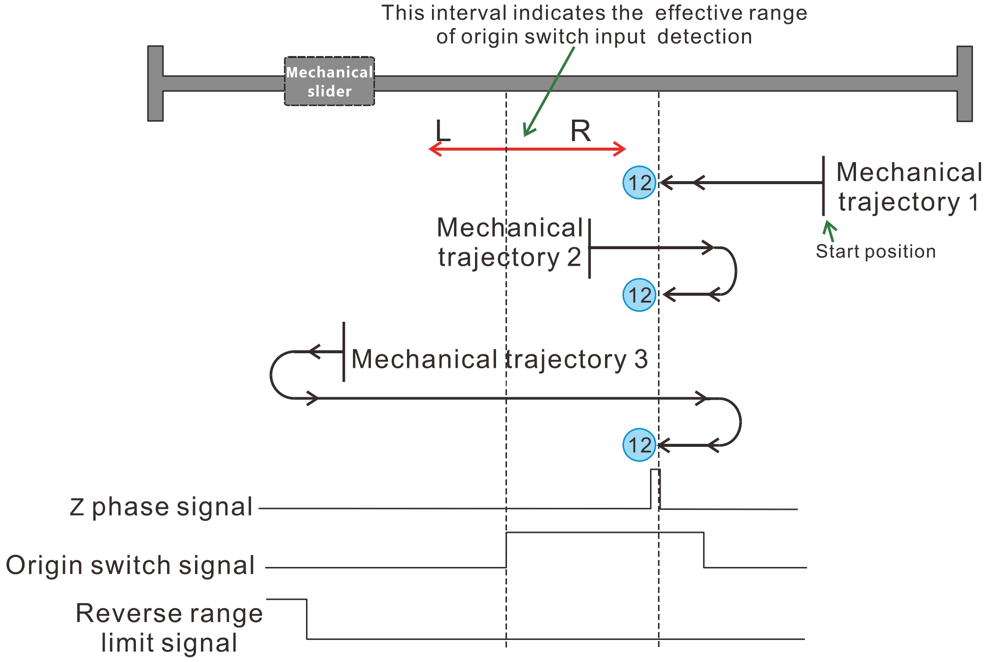
Origin mode 11: Refer to origin mode of origin switch, Z phase signal and negative limit (adopt Z phase signal of positive origin switch right of right edge)



As shown in the figure above, the mechanical slider slides in the negative limit direction (negative direction), and Z in the right position along the right side of the origin switch signal, outside of the effective range of the origin switch signal.

When the machine is within the origin switch range (mechanical motion trajectory 2), the origin can be found by directly running in the negative direction. When the machine is outside of the origin switch (mechanical motion track 1 and 3), the machine runs in a constant direction (positive direction) toward the limit switch. According to the detected sequence of the origin switch and limit switch, find the movement track and the origin.

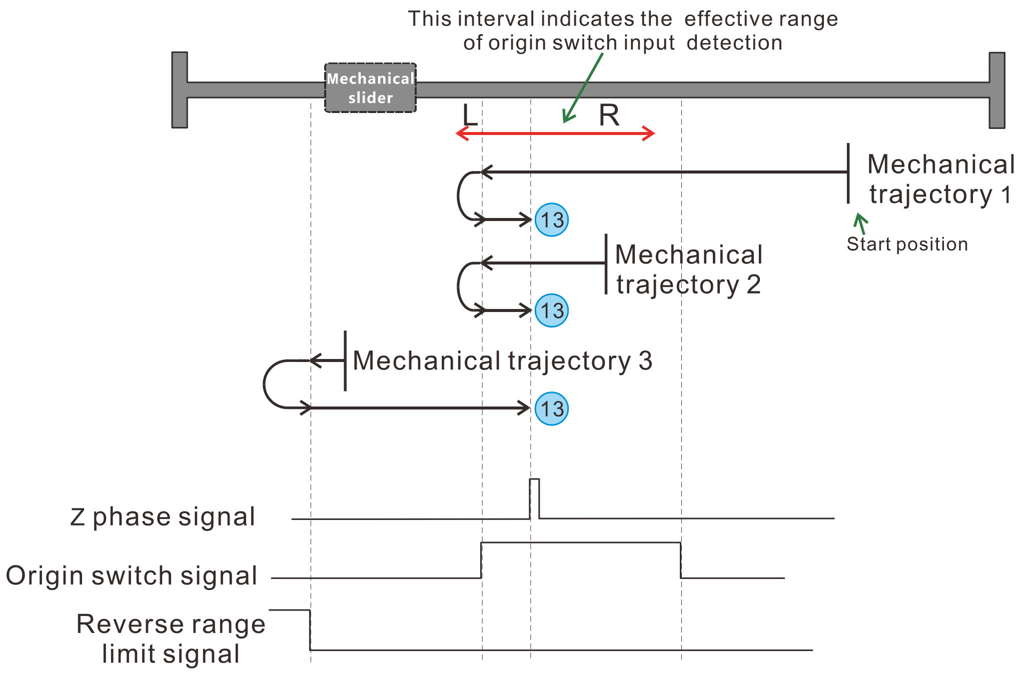
Origin mode 12: Refer to origin mode of origin switch, Z phase signal and negative limit (adopt Z phase signal of positive origin switch left of right edge)



As shown in the figure above, the mechanical slider slides in the negative limit direction (negative direction), and Z in the right position along the right side of the origin switch signal, inside of the effective range of the origin switch signal.

When the machine is within the origin switch range (mechanical motion trajectory 2), the origin can be found by directly running in the positive direction. When the machine is outside of the origin switch (mechanical motion track 1 and 3), the machine runs in a constant direction (negative direction) toward the limit switch. According to the detected sequence of the origin switch and limit switch, find the movement track and the origin.

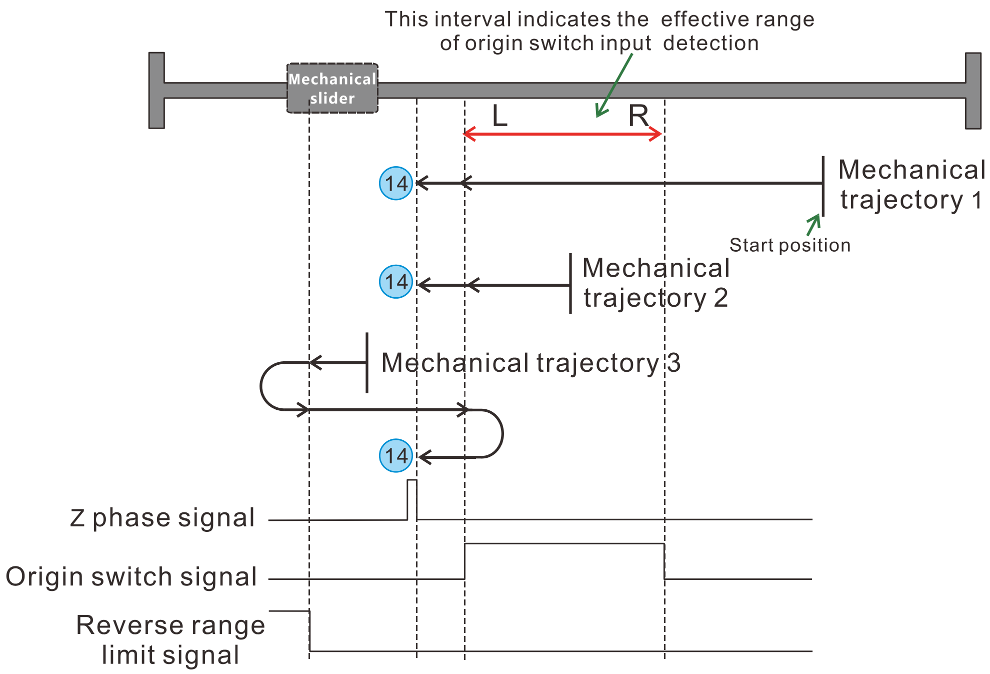
Origin mode 13: Refer to origin mode of origin switch, Z phase signal and negative limit (adopt Z phase signal of positive origin switch right of left edge)



As shown in the figure above, the mechanical slider slides in the negative limit direction (negative direction), and Z in the left position along the right side of the origin switch signal, inside of the effective range of the origin switch signal.

When the machine is within the origin switch range (mechanical motion trajectory 2), the origin can be found by directly running in the negative direction. When the machine is outside of the origin switch (mechanical motion track 1 and 3), the machine runs in a constant direction (negative direction) toward the limit switch. According to the detected sequence of the origin switch and limit switch, find the movement track and the origin.

Origin mode 14: Refer to origin mode of origin switch, Z phase signal and negative limit (adopt Z phase signal of positive origin switch left of left edge)



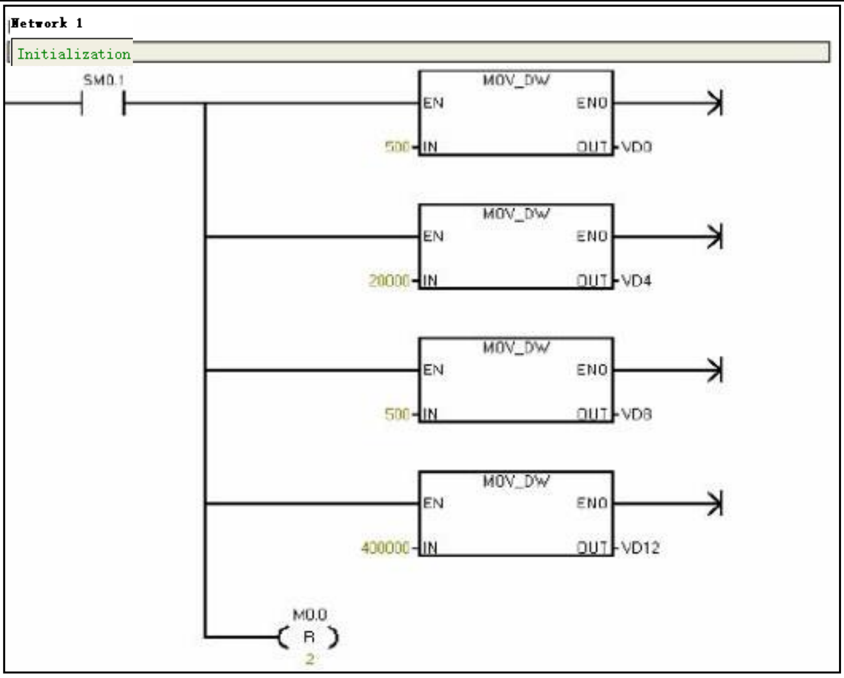
As shown in the figure above, the mechanical slider slides in the negative limit direction (negative direction), and Z in the left position along the left side of the origin switch signal, inside of the effective range of the origin switch signal.

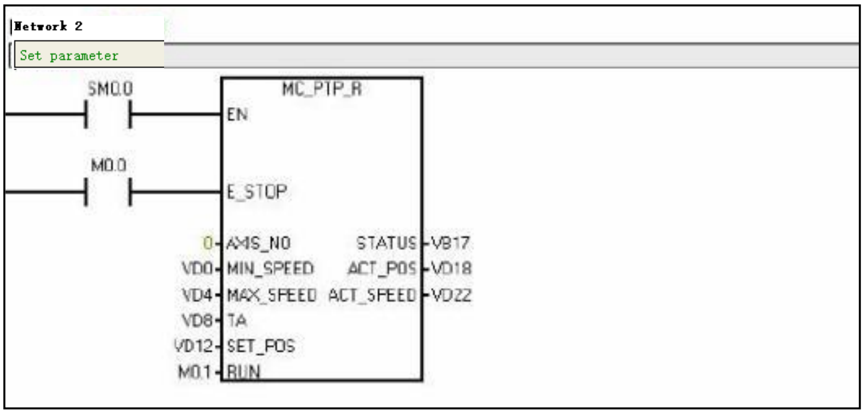
When the machine is within the origin switch range (mechanical motion trajectory 2), the origin can be found by directly running in the negative direction. When the machine is outside of the origin switch (mechanical motion track 1 and 3), the machine runs in a constant direction (negative direction) toward the limit switch. According to the detected sequence of the origin switch and limit switch, find the movement track and the origin.

E.5 Examples

Case 1) Using uniaxial relative motion instruction

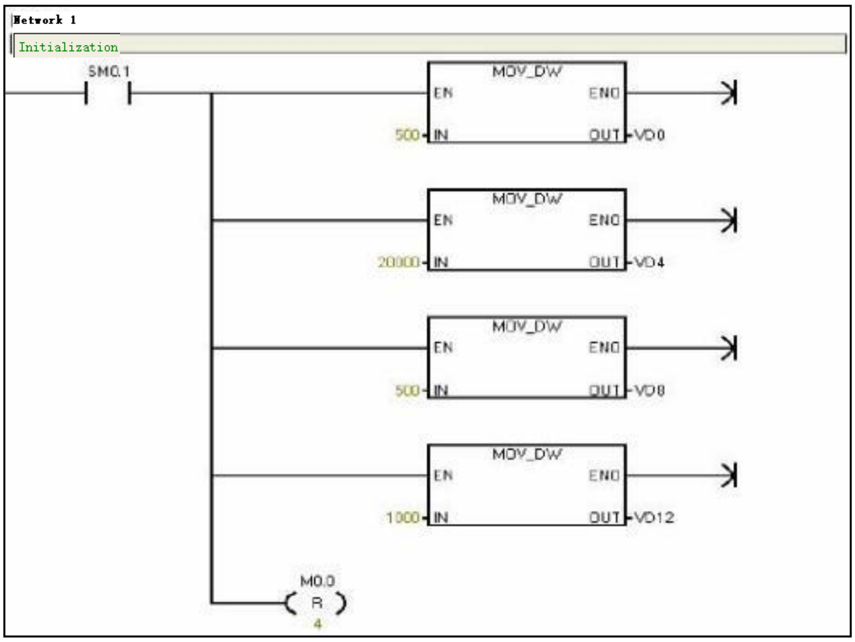
```
PROGRAM COMMENTS
Function:use for uniaxial point to point control(uniaxial fixed length drive)
output fixed pulse with one call, by setting the maximum, minimum speed and
acceleration time, the output pulse will gradually accelerate to the maximum speed
when starting up. When the pulse is about to run out, the pulse frequency will
automatically reduce to prevent vibration or jamming caused by too much inertia of
the machine when starting or stopping.
M0.0----------emergency stop bit
Axis 0--------Q0.0 is pulse output, Q0.1 is direction output
VD0-----------start/stop speed
VD4-----------normal speed after acceleration
VD8-----------acceleration time(ms)
VD12----------pulse number to output
VB17----------output status byte
VD18----------output pulse number
VD22---------current output pulse speed(frequency)
```

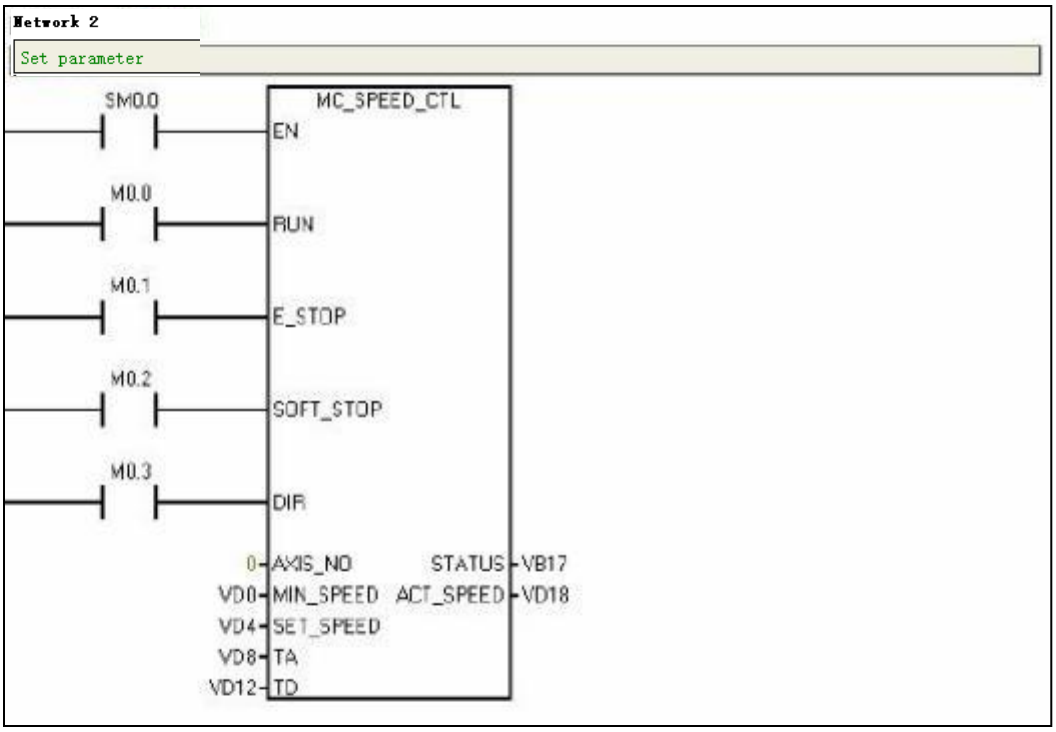




Case 2) Using uniaxial speed motion instruction

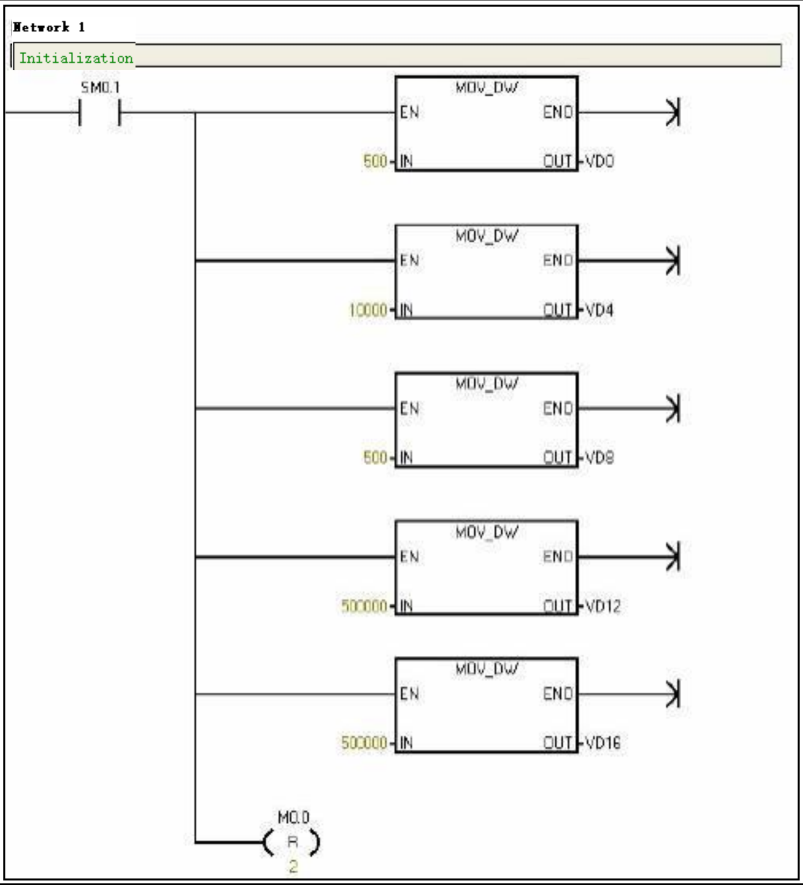
```
PROGRAM COMMENTS
Function:control uniaxial output pulse frequency, you can change the frequency
anytime(speed).
When receiving the soft stop command, it will automatically reduce to stop. when
receiving emergency stopcommand, it will immediately stop to output pulse without
deceleration.
M0.0----------enable run bit
M0.1----------emergency stop bit
M0.2----------soft stop bit
M0.3----------pulse direction bit(0 negative, 1 positive)
Axis 0--------Q0.0 is pulse output, Q0.1 is direction output
VD0-----------start/stop speed
VD4-----------normal speed after acceleration
VD8-----------acceleration time(ms)
VD12---------- deceleration time(ms)
VB17----------output status byte
VD18---------current output pulse speed(frequency)
```

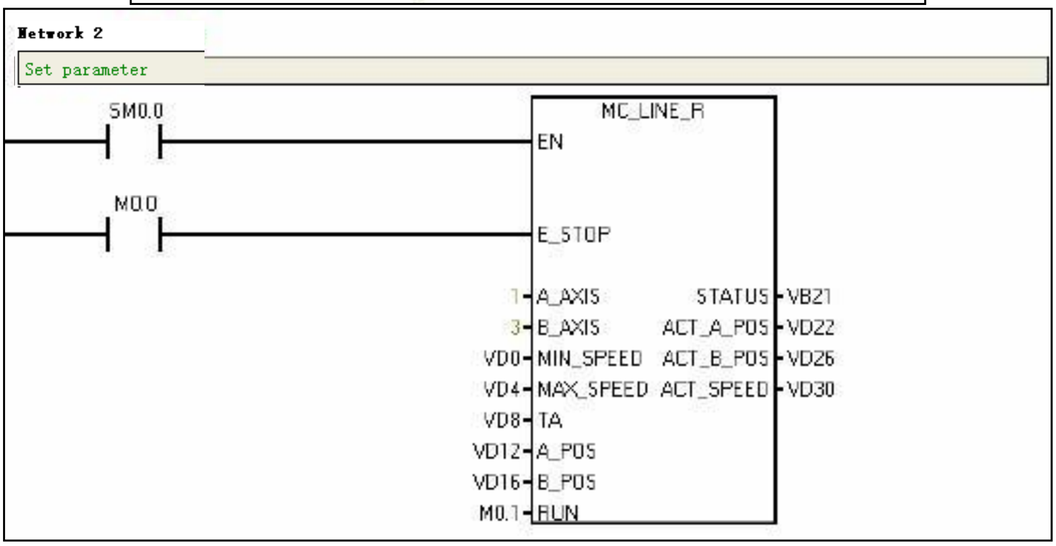




Case 3) Using two-axis line interpolation motion instruction

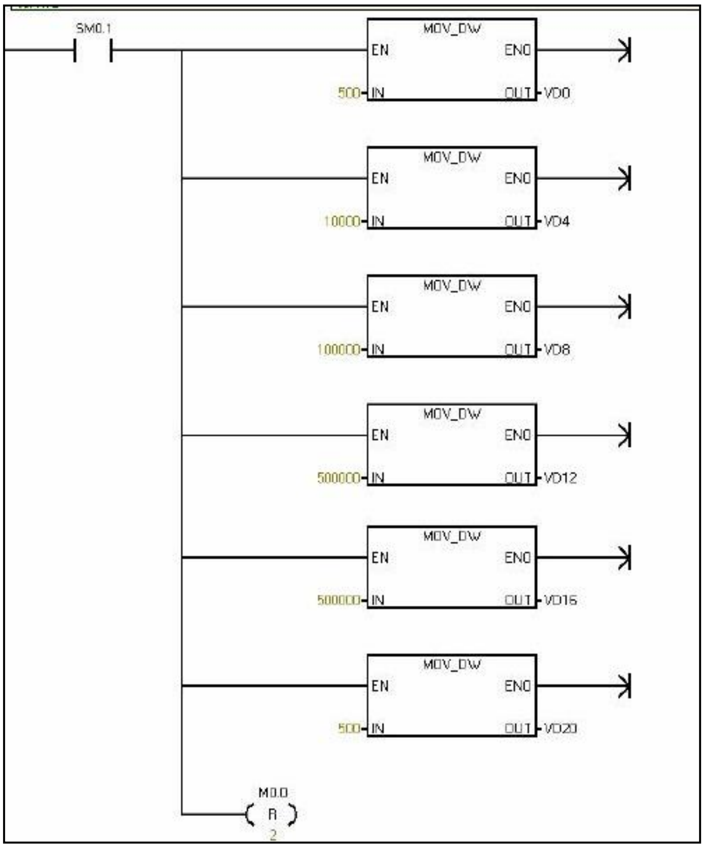
```
PROGRAM COMMENTS
Function:line interpolation in any plane area and any two axes.

M0.0-----------emergency stop bit
Axis 1---------Q0.2 is pulse output, Q0.3 is direction output
Axis 3---------Q0.5 is pulse output, Q0.7 is direction output
VD0------------long axis start/stop speed
VD4------------normal speed after long axis acceleration
VD8------------ACC/DCC time(ms)
VD12----------- virtual A axis terminus (relative) coordinate
VB16-----------virtual B axis terminus (relative) coordinate
M0.1-----------enable run bit
VB21-----------output status byte
VD22----------A axis current position(relative coordinate, actual output pulse number
for this call)
VD26----------B axis current position(relative coordinate, actual output pulse number
for this call)
VD30----------current actual speed(frequency)
```

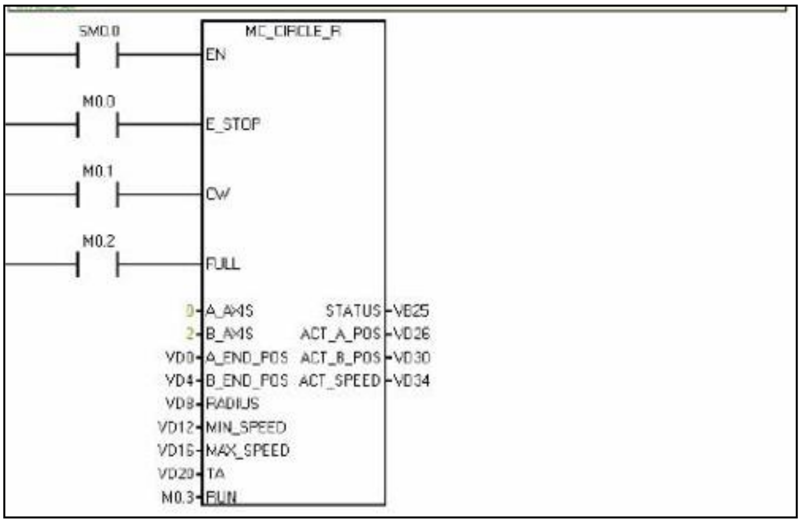




Case 4) Using two-axis arc interpolation motion instruction

```
PROGRAM COMMENTS
Function:circular interpolation in any plane area and any two axes.
M0.0----------emergency stop bit
M0.1----------clockwise or counterclockwise interpolation sign bit
M0.2----------full circle sign bit(1-full circle, 0-arc)
Axis 0--------Q0.0 is pulse output, Q0.1 is direction output
Axis 2--------Q0.4 is pulse output, Q0.5 is direction output
VD0-----------long axis start/stop speed
VD4-----------normal speed after long axis acceleration
VD8-----------arc radius
VD12---------- virtual A axis terminus (relative) coordinate
VB16----------virtual B axis terminus (relative) coordinate
VD20-----------ACC/DCC time(ms)
M0.3----------enable run bit
VB25----------output status byte
VD26---------A axis current position(relative coordinate, actual output pulse number
for this call)
VD30---------B axis current position(relative coordinate, actual output pulse number
for this call)
VD34---------current actual speed(frequency)
```


SM0.1
MOV_DW
EN
ENO
500
IN
OUT
VD0
MOV_DW
10000
VD4
MOV_DW
100000
VD8
MOV_DW
500000
VD12
MOV_DW
500000
VD16
MOV_DW
500
VD20
M0.0
R
2


SM0.0
MC_CIRCLE_R
EN
M0.0
E_STOP
M0.1
CW
M0.2
FULL
0
A_AXIS
STATUS
VB25
2
B_AXIS
ACT_A_POS
VD26
VD0
A_END_POS
ACT_B_POS
VD30
VD4
B_END_POS
ACT_SPEED
VD34
VD8
RADIUS
VD12
MIN_SPEED
VD16
MAX_SPEED
VD20
TA
M0.3
RUN

F Using SM231 Weighing Module and the Library

Weighing module converts the voltage signal of weighing sensor to digital AIW, the function of SM231 weight module is to turn it into actual weight.

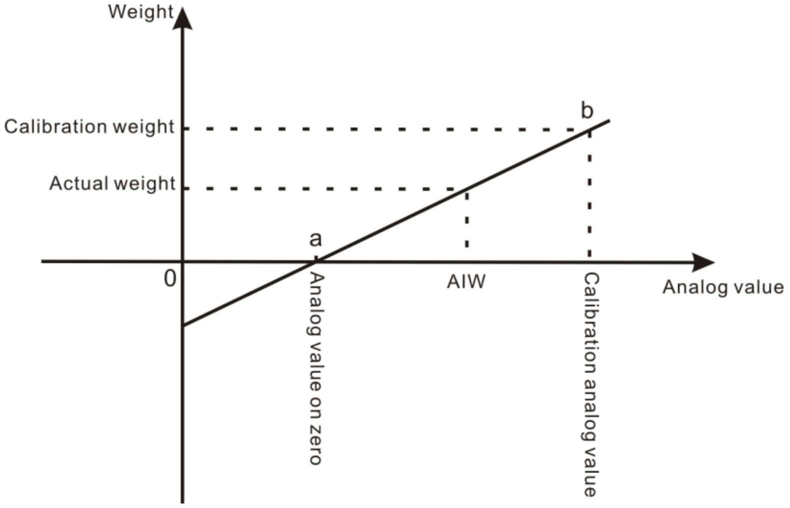


Figure F-1: The principle of measuring weight

You can obtain the linear relationship equation between weight and analog value via point a and b from diagram above, then you can get the actual weight of AIW. So as to get the coordinate value of point a and b, you need adjust zero and calibrate.

Zero Setting and Calibration method:

First suppose the weighting sensor has fixed to the horizontal direction, and the pallet has horizontally fixed on the weighting sensor.

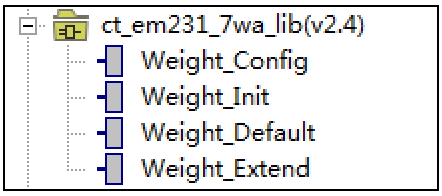
When setting zero, don't put anything on the the pallet, you can get the Zero-analog value while it is stable.

When calibrating, the weights for calibrating are placed to the pallet, you can get the calibration analog value while it is stable.

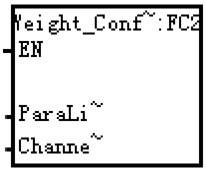
F.1 Weighing Library Function Illustration

Weighing module correspond to weighing library EM231_7WA_LIB (V1.4), library files download address: www.co-trust.com

Weighing library instructions parameter:



Weighing library configuration



① Name: Weight_Config

② Function: Set channel numbers and its starting address

③ Parameter

Name	I/O	Type	Range	Default	Description
EN	IN	BYTE	--	--	Enable end
ParaListBase	IN	DWORD	--	--	Set startup address pointer to store parameter table, refer to parameter table below for detailed info. There must be a “&” before entering operation.
ChannelNum	IN	BYTE	0~255	--	Set total connected channel number of SM231

Note: This function must be called for a time and execute in the first loop cycle. For N channels, ChannelParaNum * N bytes are occupied in all. It is a must to ensure that the V memory parameter table occupied not overrange and not cover space of other program.

72 bytes are occupied each channel when weighing library is called, table below is the parameter definition the storage saved (take VB0 of channel 0):

Parameter table

Parameter	Address	Explanation	Note
Mode	VB0	Mode	Basic/expand mode
Sensitivity	VB1	Sensitivity of sensor	
LimitFreq	VB2	Frequency of LPF	
FilterDepth	VB3	Filter depth	0-255, 0 or 1 means without average filtering
WeightRange	VW4	Max weight range	
FirCalWeight	VW6	First calibration of weight	
SecCalWeight	VW8	Second calibration of weight	
TareInput	VW10	Tare input	
MinWeight	VW12	Minimum weight range	General 20d, d is digital step
Step	VB14	Digital step	Range: 1, 2, 5, 10 and 20
StandstillTime	VW15	Pause time	Unit: ms
StandstillRange	VW17	Pause range	
ZeroValue	VW19	Zero sample value	
FirCalValue	VW21	First calibration of sample	
SecCalValue	VW23	Second calibration of sample	
GWProcessValue	VW25	GWProcessValue	
NWProcessValue	VW27	NWProcessValue	
TWProcessValue	VW29	TWProcessValue	
AnalogValueInit	VW31	Sample value before filter	
AnalogValue	VW33	Sample value after filter	
GrossWeight	VW35	GrossWeight	
NetWeight	VW37	NetWeight	
TareWeight	VW39	TareWeight	

Status_I	VB41	"Weight_Init" status byte	
Status_D	VW42	"Weight_Default" status word	
Status_E	VB44	"Weight_Extend" status byte	
InternalVariable1	VB45	Internal variable 1	
AQWx	VW46	L area backup	
AQWx2	VW48	L area backup	
LB21_D	VB50	Weight_Default L area backup	
LB57_D	VB51	Weight_Default L area backup	
LB58_D	VB52	Weight_Default L area backup	
LB59_D	VB53	Weight_Default L area backup	
LB59_E	VB54	Weight_Extend L area backup	
LW36_E	VW55	Weight_Extend L area backup	
LD40_E	VD57	Weight_Extend L area backup	
Reserved	VB61	Reserved	
ZeroTraceTime	VD62	Zero Trace Time	
StandBeginTime	VD66	Begin time from pause status	
ZeroTraceValue	VW70	Zero Trace Value	

Weight initialization library

① Name: Weight_Init

② Function: Configure sensitivity, low pass filter cutoff frequency and average filter depth of sensors which connect to weighing module.

③ Parameter

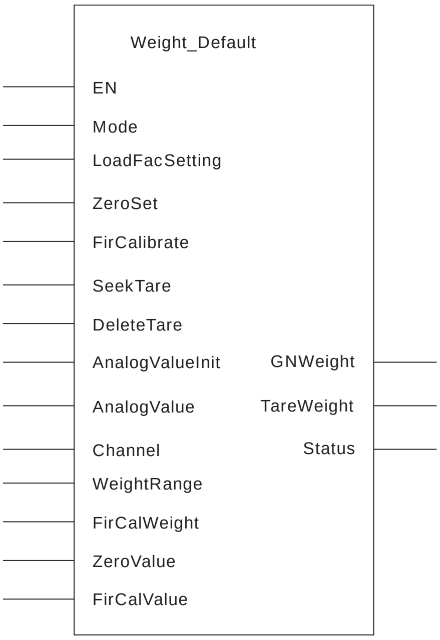
Name	Description	I/O	Type	Value range	Note
Channel	Channel number	IN	BYTE	-	0~6
Sensitivity	Sensor sensitivity	IN/OUT	BYTE	2	1: Eigenvalue range 1mV/V 2: Eigenvalue range 2mV/V 4: Eigenvalue range 4mV/V No other definition
LimitFreq	Low pass filter cutoff frequency	IN/OUT	BYTE	4	3: fg = 5Hz 4: fg = 2Hz 5: fg = 1Hz 6: fg = 0.5Hz 7: fg = 0.2Hz 8: fg = 0.1Hz 9: fg = 0.05Hz No other definition
Filter Depth	Average filter depth	IN/OUT	BYTE	15	0~255, 0 or 1 indicates no average filter
AQWx	First analog output of weighing channel	OUT	WORD	--	Format: "0x53"+"Sensitivity"
AQWx2	Second analog output of weighing channel	OUT	WORD	--	Format: "LimitFreq “+”FilterDepth"
Status	Status byte	OUT	BYTE	--	Bit0: eigenvalue error Bit1: Low pass filter frequency setting error

Note: Call this instruction by SM0.0

Refer to *F.2 Weighing Library Mode Explanation* for parameter explanation about sensitivity, LPS cutoff frequency and average filter depth.

Weighing standard library

① Name: Weight_Default



② Function: Calibration, zero set and measurement

③ Parameter

Name	I/O	Type	Range	Default	Explanation
Mode	IN	BOOL	--		0: basic mode 1: expand mode
LoadFacSetting	IN	BOOL	--		Load factory setting, effective at rising edge
ZeroSet	IN	BOOL	--	5461	Zero setting, effective at rising edge
FirCalibrate	IN	BOOL	--		First calibration, effective at rising edge
SeekTare	IN	BOOL	--		Calculate tare, effective at rising edge
DeleteTare	IN	BOOL	--		Delete tare, effective at rising edge
AnalogValueInit	IN	WORD			Analog value before filter, corresponding to first input of weight channel
AnalogValue	IN	WORD			Analog value after filter, corresponding to second input of weight channel
Channel	IN	BYTE			Channel No. 0-6
WeightRange	I/O	INT		0~32767	Max weight range(default as 2000)
FirCalWeight	I/O	INT	More than 5% of range	2000	Weight for first calibration(default as 2000)
ZeroValue	I/O	WORD			Zero sample value(default as 5461)
FirCalValue	I/O	WORD		60074	First calibration value(default as 60074)
GNWeight	OUT	INT			GNWeight
TareWeight	OUT	INT			TareWeight
Status	OUT	WORD			Status

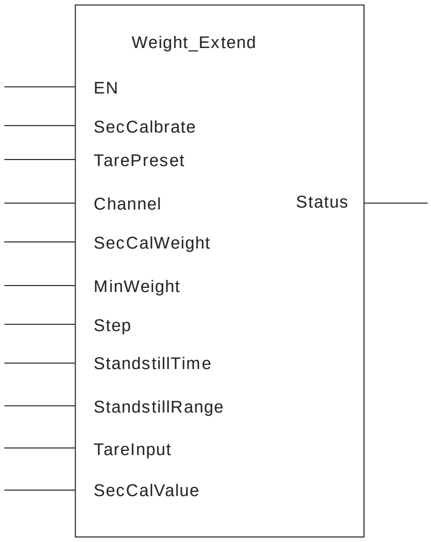
Note: Refer to appendix *F.2 Weighing Library Mode Explanation* for parameter explanation of mode select, first calibration and tare.

Status note:

Bit	Name	Range	Description
Bit0	Power-down alarm	Basic/expansion mode	0: Module power normal, 1: Module no power. When Bit0=1, sampling value is 0xFFFF
Bit1	Break line alarm	Basic/expansion mode	0: Sensor connection normal, 1: sensor disconnect When Bit1=1, sampling value is 0xFFFE
Bit2	Outrange alarm	Basic mode	1: Rough weight≥rated weight, sampling value is 0xFFFD
	Max+9e	Expansion mode	1: Rough weight≥rated weight +9e. e is digital step
Bit3	Tare decided	Basic/expansion mode	1: Tare storage has been occupied(tare process value≠0)
Bit4	Tare preseted	Expansion mode	1: Preset tare
Bit5	1/4d	Expansion mode	1: Rough weight<±0.25d, d is digital step
Bit6	Pause	Expansion mode	1: Ensure pause status
Bit7	Zero be found	Basic/expansion mode	1: Find zero already
Bit8	Calibrated	Basic mode	1: Complete once calibration
		Expansion mode	1: Complete first calibration (second calibration weight 0)or complete first and second calibration
Bit9	Low weights	Expansion mode	1: Current weight <minimum weighing range
Bit10	Execute only in pause status	Expansion mode	1: Find zero and tare must be paused in expansion mode
Bit11	Execute only in zero status	Basic/expansion mode	1: Zero set before first or second calibration
Bit12	Execute only in calibrated status	Basic/expansion mode	1: Calibrated before measure or preset tare
Bit13	Calibrated weight too small	Basic/expansion mode	1: First calibration weight and zero; the weighing difference between second and first calibration cannot be less than 5%FS
Bit14	Illegal tare	Basic/expansion mode	1: Tare cannot be less than 0 or more than maximum weighing range
Bit15			

Weighing extension library

① Weight_Extend



② Function: Realize second calibration, set minimum weight, digital phase step, pause status detection and preset tare.

③ Parameter

Name	Description	I/O	Type	Value range	Note
SecCalibrate	Second calibration, effective in rising edge	IN	BOOL	--	--
TarePreset	Preset tare	IN	BOOL	--	--
Channel	Channel number	IN	BYTE	--	0-6
SecCalWeight	Weight of second calibration	IN/OUT	INT	0	0 indicates no second calibration
MinWeight	Minimum weight	IN/OUT	INT	20	This weight value can only use for calibration records with specified digital steps above the minimum weight. The minimum weight depends on the type of sensor.
Step	Digital phase step	IN/OUT	BYTE	1	Range is 1, 2, 5, 10, 20, no other definition
StandstillTime	Pause time	IN/OUT	INT	1000	Unit: ms
StandstillRange	Pause range	IN/OUT	INT	10	
TareInput	Tare input	IN/OUT	INT	0	Preset tare
SecCalValue	Second calibration analog value	IN/OUT	WORD	0	
Status	Status byte	OUT	BYTE	--	

Status note:

Bit	Name	Note
Bit0	Weight of second calibration is too small	The weighing difference between second and first calibration cannot be more than 5%FS
Bit1	Digital phase step error	Digital phase step can only be 1, 2, 5, 10, 20
Bit2	Pause time error	Pause time must>0

Bit3	Pause range error	Pause range must>0
Bit4	Preset tare outrange	Preseted tare cannot be negative or outrange
Bit5	Reserved	--
Bit6	Pause	Ensure pause status
Bit7	Illegal channel	--

F.2 Weighing Library Mode Explanation

This section shows important parameters of each instruction and operation notes.

1. Weight_Config

Call it by SM0.1, where ParaListBase set the startup address pointer and its memory range of specified parameter list, each channel occupy 72 bytes; ChannelNum connect SM231 channel number which can only more than or equal to actual one for you

2. Weight_Init

Call it by SM0.0, you can modify Sensitivity, LimitFreq and average Filter depth which will be given a value in factory setting(modify sensitivity before calibration, and anytime to modify for the other two).

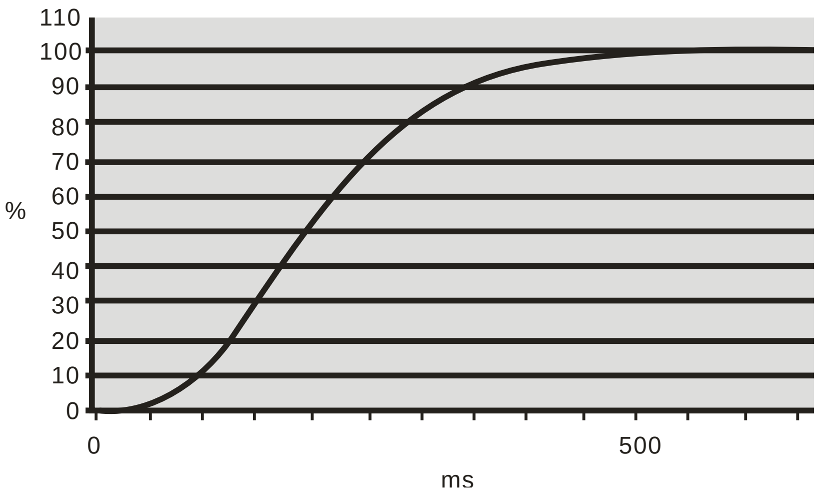
- **Sensitivity**

According to eigenvalue of weighing sensor, 1mV/V, 2mV/V or 4mV/V can be selected.

For example: if the eigenvalue of weighing sensor is 2.85mV/V, the next higher eigenvalue must set as 4mV/V.

- **Cutoff frequency of low pass filter**

Equip a low pass filter conduct electricity in critical condition in modules to barrage jamming. Figure below shows step response of digital lowpass filter at fg=2Hz:



Define appropriate cutoff frequency to resist disturbance. The reaction speed that weighing module to measurement is set by defining limiting frequency.

For example: the weighing module shows quick response at 5Hz cutoff frequency, but much slower at 0.5Hz.

- **Average filter depth**

Filter can smooth the weighing value and resist disturbance. The last weight is an average of n measurements; 50Hz sample frequency means weigh every 20 ms. For example, filter depth n=10 represents taking 10 measurements for average value and recalculate every 20ms.

If the interfering frequency integral multiple of “1/(filter depth * 20 ms)”, then the average filter will realize brilliant damping of cycle disturbance.

3. Weight_Default

Call it by SM0.0, set Mode 1 for second calibration. The parameter will automatically assign value when using 231-7WA module for first time with loading factory setting. Refer to instruction explanation before for details.

- **Mode explanation**

Weighing module SM231-7WA offers 2 usages by configure parameter in “Mode" of "Weight_Default' to select basic mode or expansion mode.

1) Basic mode (Mode=0)

(1) Call “Weight_Config”, "Weight_Init" and "Weight_Default";

(2) Supported following functions:

• Set sensor sensitivity, low pass cutoff frequency and filter depth;

• Load factory parameter;

• Set maximum weighing range;

• Set zero;

• First calibration;

• Measure tare;

• Delete tare;

• Read sampling value both before and after filtering;

• Power-down alarm, line broke alarm, outrange alarm and set zero etc.

2) Expansion mode (Mode=1)

(1) Call "Weight_Config", "Weight_Init", "Weight_Default" and "Weight_Extend";

(2) Support following functions:

• Supported all functions in basic mode;

• Set minimum weighing range;

• Second calibration;

• Preset tare;

• Set Digital phase (set minimum range scale, show in 1×10^k, 2×10^k or 5×10^k)

• Ensure pause status (set pause status when setting zero and tare)

• Newly added status: Max+9e (GB/T 7724-2008), Preseted tare, 1/4d (GB/T 23111-2008), pause, low weight;

• Trace zero

- **Tare**

Packing material Weight of commodity that is the weight of transport package.

For example, GNweight value is the weight of car before weighing which turn to 0 after SeekTare breakover by uprising edge, and Tareweight replace the GNweight value before.

Weighing after loading with cargo, the GNweight value is net weight. Which means GNWeight is rough weight without tare and become net weight with tare.

- **First calibration**

It is electricity signal that the module receive from sensor, which is not corresponding to weight without calibration.

For example, put a 1000g weight on the weighing machine, the actual value of GNweight will set by first calibration. If you put 1000g weight and set FirCalweight 2000(200g), the GNweight will turn every object weighing 1000g into 200g. So the first calibration is very important.

4. Weight_Extend

Call it by SM0.0 only when expansion mode (mode=1), to do second calibration, presetting tare, setting digital steps and starting zero tracing.

Pause time

Pause monitor can be used to identify when the weighing machine would stay at a balance status as well as the static operation (mainly zero sample and tarring). If the change of weight is less than specified deviation range (pause value) in a required time (pause time), then the weighing machine is sure paused.

Digital step

Define digital step as 1, 2, 5, 10 or 20.

Zero tracing

Zero tracing can be used to eradicate zero drift after long time using scale, which will be effective after breakover.

With this function, when the absolute value of rough weight/net weight is less than threshold for a while, write the current analog value after filtering into zero one and save the former analog, the output of rough weight/net weight is 0.

The origin zero analog will be written back after cancel this function. The maximum range for zero tracing is 10d (d is digital step), the overrange value is invalid.

Preset tare

TareInput is input tare which shows value of Tareweight after TarePreset breakover by uprising edge with a written tare value. Tare value turns to 0 after DeleteTare breakover.

For example, TareInput value set as 500 without anything on weighing machine. The TareWeight becomes 500 after breakover of TareInput instruction, the GNweight becomes -500 along with TareWeight turns 0 after deleting theDeleteTare instruction.

The similarity and differentia between preset tare in Weight_Extend and tare in Weight_Default:

Differentia:

TareInput can be random set to be displayed tare value, no matter you put an object or not. GNweight owns value to tarring which is GNweight value only when object put inWeight_Default instruction.

Similarity:

Delete tare by using uprising edge breakover DeleteTare so back to status before tarring. After tarring, value of GNweight is former one subtract tare.

Second calibration

Secselweight is weight value of second calibration (set 1000g weight as 10000).

Note: the difference value of weight between first and second calibration must be more than 5% of range.

Put weight on weighing machine for second calibration when Weight_Extend status Word becomes 64 (ensure pause status) after first calibration.

For more details about weighing modules visit: http://www.co-trust.com

G SM277A Module

【Main feature】

- Adopt a photoelectric isolation technique, high reliability and strong anti-interference ability.
- Integrated terminal resistance, the bus adopt connection mode, no need dedicated network tap.
- The power supply includes anti - access protection and Surge absorption function, it's suitable for bad industrial environment.

【Terms of use】

- The signal wire should use shielded twisted pair wires, and the double end of the shielded wire should connect to the ground.
- The grounding terminal should connect to the earth wire while the system is under good grounding cases.
- If it's the last station, the terminal resistance should set to the ON position.

PROFIBUS-DP network

【Communication】

PROFIBUS-DP network connect to SM277A DP slave station module by its DP communication port, while SM277A DP slave station module connect to CTH200 CPU through serial I/O bus. SM277A use terminal to connect PROFIBUS network without using standard connector.

【Function】

SM277A is a modular slave station device for PROFIBUS DP networks that can connect up to 6 extended modules for digital or analog I/O.

As a DP slave station, the SM 277A module can run on any PROFIBUS baud rate between 9600 and 12M. Also receives a variety of I/O configurations and different amounts of data from the main station and sends data to the main station. The SM277A can also read and write variable data blocks defined in the CTH200 CPU, allows user to exchange any type of data with master station.

MPI network

【Communication】

The SM277A module communicate with other master stations on the same network such as PG/PC programming station, Copanel HMI or S7-300/S7-400CPU as an MPI slave station. Use XGET/XPUT instruction function of S7-300/400 and other main station modules to provide the network communication between MPI main station and CTSC-200. When the SM277A module conducts MPI communication, the MPI master station must use its station address to send information to the CTH200 CPU. The MPI information sent to the SM277A module will be transmitted to the CTH200 CPU via SM277A.

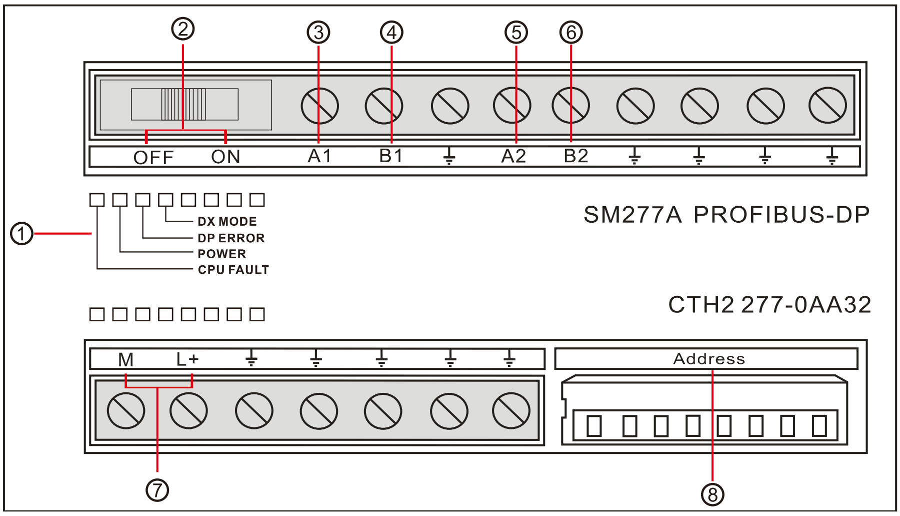
【Function】

SM277A is a slave station device for MPI networks that can connect up to 6 extension modules for digital or analog I/O.

As MPI slave station, the SM277A module can be run at any MPI baud rate between 9.6K and 12M. Also receives a variety of different I/O configurations and different amounts of data from the master station and sends data to the master station, and SM277A can also read and write variable data blocks defined in the CTH200 CPU, allows the user to exchange any type of data with the master station.

Components of SM277A

Status LED is located on the front of the module, address switch, terminal resistance switch ,DP slave interface terminal and user power supply are located on the side of wiring terminal, see as following figure:



- ① Status LED
- ② Terminal resistance switch: ON means with terminal resistance, OFF means without terminal resistance.
- ③ Isolated signal A1
- ④ Isolated signal B1
- ⑤ Isolated signal A2
- ⑥ Isolated signal B2
- ⑦ User Power supply
- ⑧ Address switch: Setup by 8 bits dial switch, and expressed in binary form, the available range is 0~126.

Note

A1/B1 and A2/B2 are both RS-485 electric port, consistent internal electric characters, you can choose to connect one or two group.

PROFIBUS DP communication between S7-300 and SM277A of CTH200

PROFIBUS DP communication between S7-300 and SM 277A of CTH200 need S7-300 station configuration in STEP7, no need to configure and program the communication in CTH200 system, just sort out the data to be communicated and store it in the V storage area corresponding to the hardware I/O address of S7-300 configuration SM277A slave station.

Write program to call FC1 (DP_SEND) and FC2 (DP_RECV) in OB1, DP master can read and write from the slave station data to achieve S7-300 and CTH200 CPU communication. Execute the DP_SEND instruction to output the memory data of CTH200 CPU to the SM277A extension module. By performing DP_RECV, input the SM277A extension module data to the memory of CTH200 CPU.

Figure G-1 shows a PROFIBUS network with CTH200 CPU and SM277A DP slave station module.

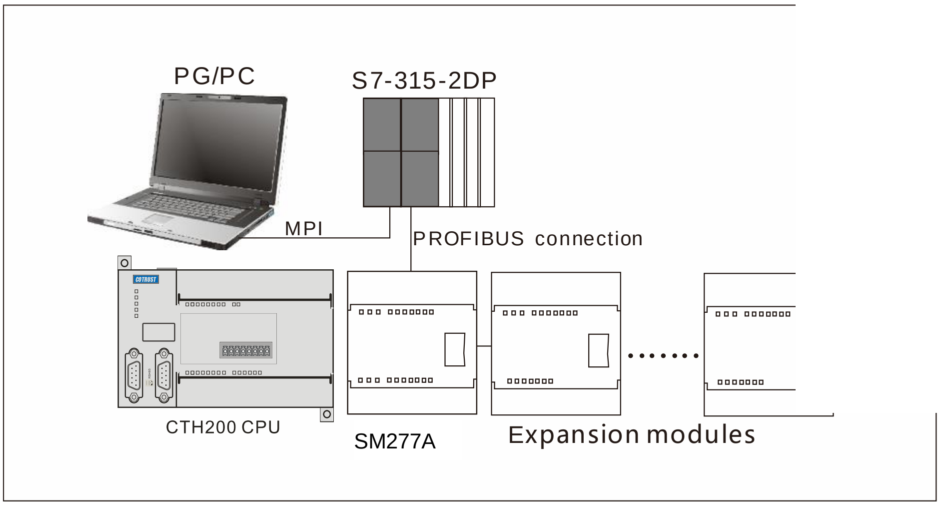


Figure G-1 PROFIBUS network example

- Use S7-300 with CPU 315-2 as the DP master station, and configure with STEP 7 programming software.
- CTH 200 CPU is DP slave station of CPU 315-2 DP master station.
- CPU 315-2 DP master station use DP_SEND and DP_RECV of user program to read and write data from CTH200.

Note: To use SM277A as a DP slave, you must set the DP port address that matches in the master group. The slave station address is set by DIP switch of SM277A module.

MPI communication between S7-300 and SM277A of CTH200

When CTH200 CPU and S7-300 conduct MPI communication, no need to write any program related to communication in CTH200 PLC, just set the data for exchange into continuous V storage area, and in S7-300 need to call system function X_GET (SFC67) and X_PUT (SFC68) in OB1 (or timer interrupt tissue block OB35) to realize the communications between S7-300 and CTH200 CPU, When call SFC67 and SFC68, the VAR_ADDR parameter is filled in the data address area of CTH200, where P# db1.* * * BYTE n corresponds to the data area from VB * * to VB (* * +n) in the CTH200 CPU V storage area.

Figure G-2 shows an MPI network with CTH200 CPU and EM277A DP slave station module.

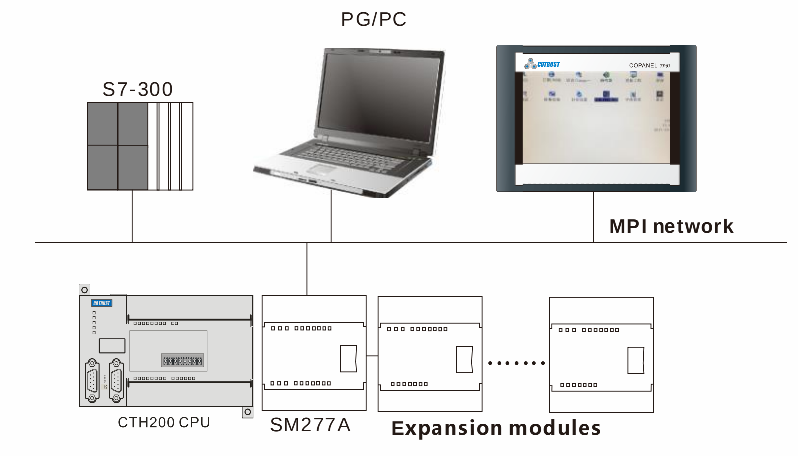


Figure G-2 MPI network example

- Use S7-300 master station module, PG\PC program station and Copanel HMI as MPI master station
- CTH200 CPU is a MPI slave station
- Use XGET / XPUT instruction of S7-300/400 master station module to read and write data from CTH200.

Note: Set the slave station address (NEST_ID) the same with dial code address of SM277A module

H SM277B Module

SM277B, connect with 6 digital or analog I/O expansion modules at most, is a modular slave station device for PROFIBUS DP network, which could communicate with DP master station. With terminal connecting PROFIBUS network instead of standard connectors, the baud rate is automatically adjusted to be consistent with the main station.

【Main feature】

- Photoelectric isolation technique, strong anti-interference ability, high reliability.
- Power supply with reverse connection protection and surge absorption function, suitable for harsh industrial environments.

【Terms for use】

- Signal lines should be shielded twisted-pair cable, and double side grounding.
- Ground the module earthing terminal under the condition of the system has a good grounding, otherwise not grounded.
- As the last station, termination resistors should be ON position

H.1 PROFIBUS-DP Network Structure

Distributed I/O systems contain active (master) and passive (slave) nodes that are interconnected through PROFIBUS-DP.

The following figure shows SM 277B in a typical PROFIBUS-DP network configuration:

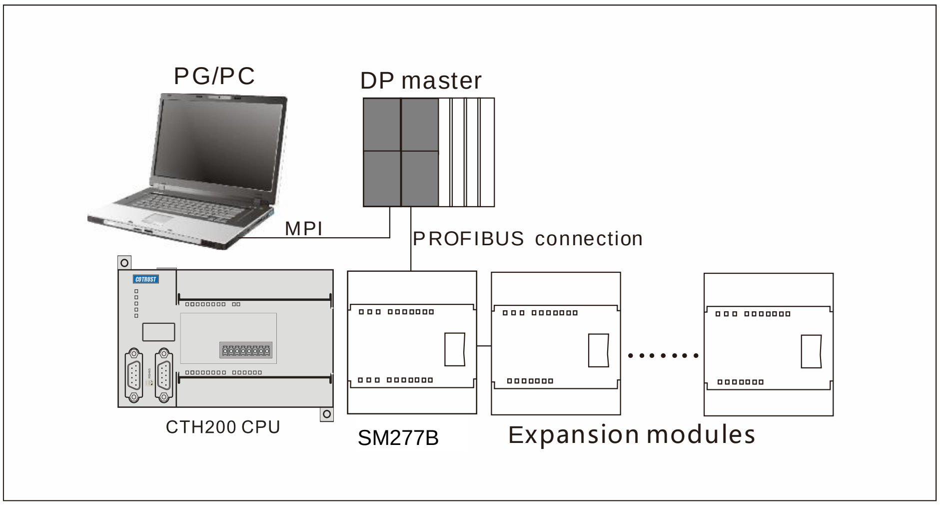


Figure H-1 Construction of SM277B PROFIBUS-DP network

H.2 Components of SM277B

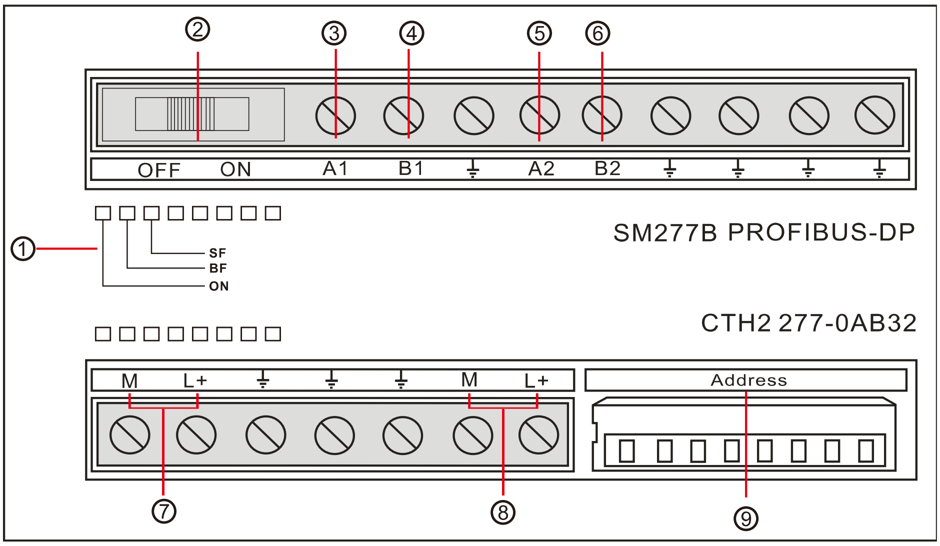


Figure H-2 SM277B structure

① Status LED

② Terminal resistance option switch: ON indicates with terminal resistance, OFF indicates without terminal resistance

③ Isolated signal A (network input)

④ Isolated signal B (network input)

⑤ Isolated signal A (network cascade)

⑥ Isolated signal B (network cascade)

⑦ User power

⑧ Sensor power

⑨ Address switch: set with 8 bit DIP switch, indicates in binary number, effective option range is 1-125

H.3 Operating Guide

This section introduces how to use SM277B, combined with "CPU312-1AE13 and SM277B communication" example gradually introduced SM277B hardware configuration, user programming and debugging as well as system diagnosis.

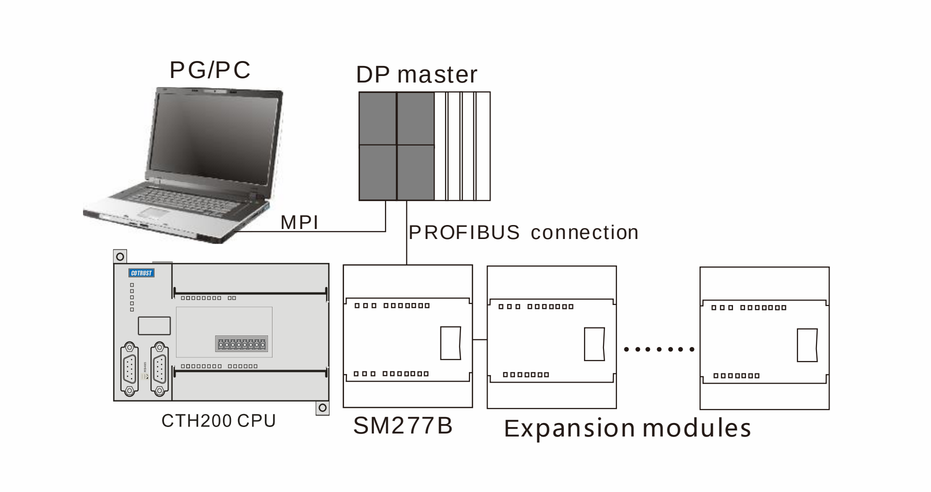


Figure H-3 SM277B network structure

Components in this example

Components	Explanation
A PG/PC with STEP7	Version of STEP 7 must support to configure DP master station
MPI program cable	For programming(download network hardware, program and monitor data)
A DP master station system	A CPU312-1AE13, a CP 342-5(as master)
CTH200 SM 277B	DP slave station device
A PROFIBUS communication cable	PROFIBUS cable must have a standard PROFIBUS connector to connect DP master station
An expansion module connect to SM277B	Such as CTH200 SM223-1BL32 module

H.3.1 Hardware Configuration

Steps:

1) Start SIMATIC Manager

Select “File” -> “New”, enter project name and select catalogue to save project in dialog box.

2) Create a SIMATIC 300 station

Select station in use (such as: SIMATIC 300 station)

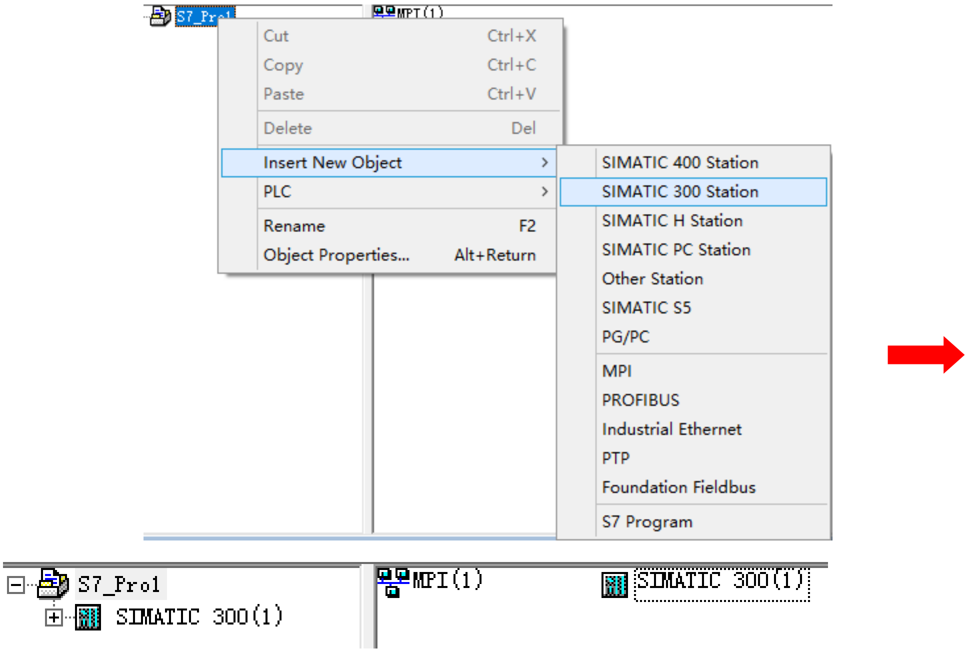


Figure H-4 Insert SIMATIC 300 Station

3) Open HW-Config to configure hardware

Double click the inserted 300 station “SIMATIC 300(1)” in step 2), as shown below:

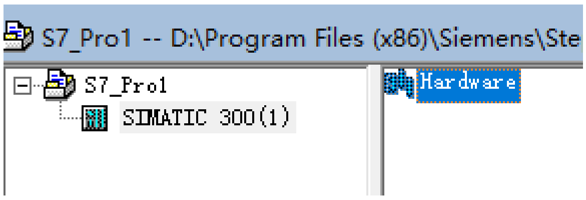


Figure H-5 HW-Config Interface

Double click “Hardware” in picture above to open HW-config interface, set distribute I/O (DP) rack, modules and PROFIBUS connection

4) Add guide rail

First add guide rail for station in use (Rail), use it to install DP master station

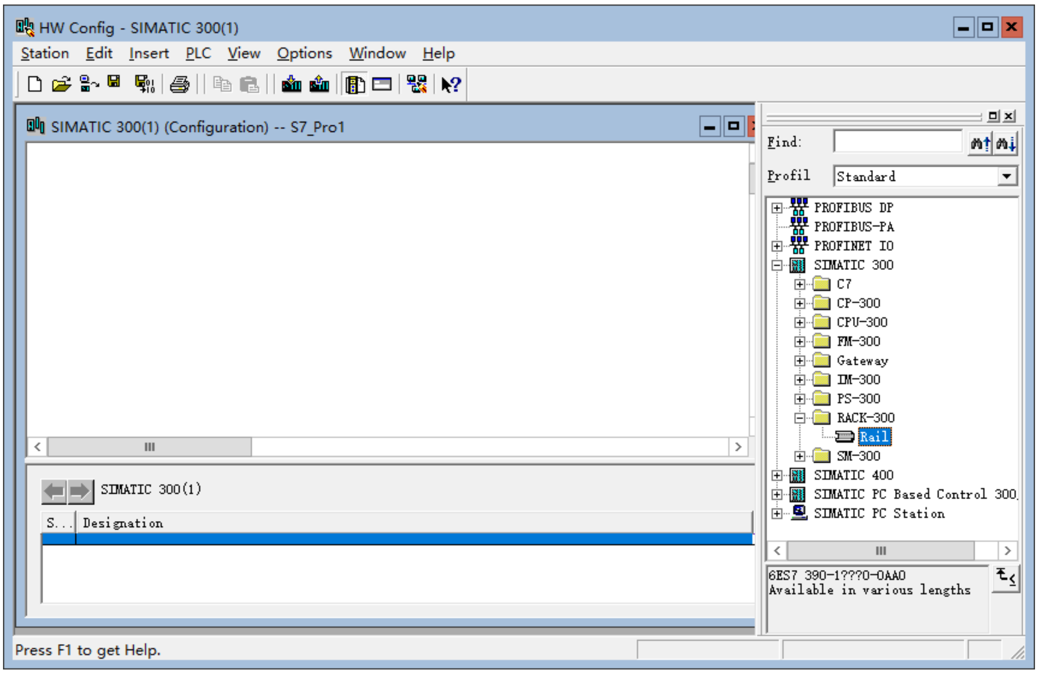


Figure H-6 Add Guide Rail

5) Add Power Supply (optional)

Add power to guide rail. The power can only be placed in the first slot of the rack.

6) Add DP master station CPU and PROFIBUS connection (CP 342-5)

Drag the CPU of the DP master station actually used in the hardware directory to the second slot of the guide rail or double click to add it, then drag the CP 342-5 actually used in the hardware directory to the fourth slot of the guide rail, as shown in the figure below:

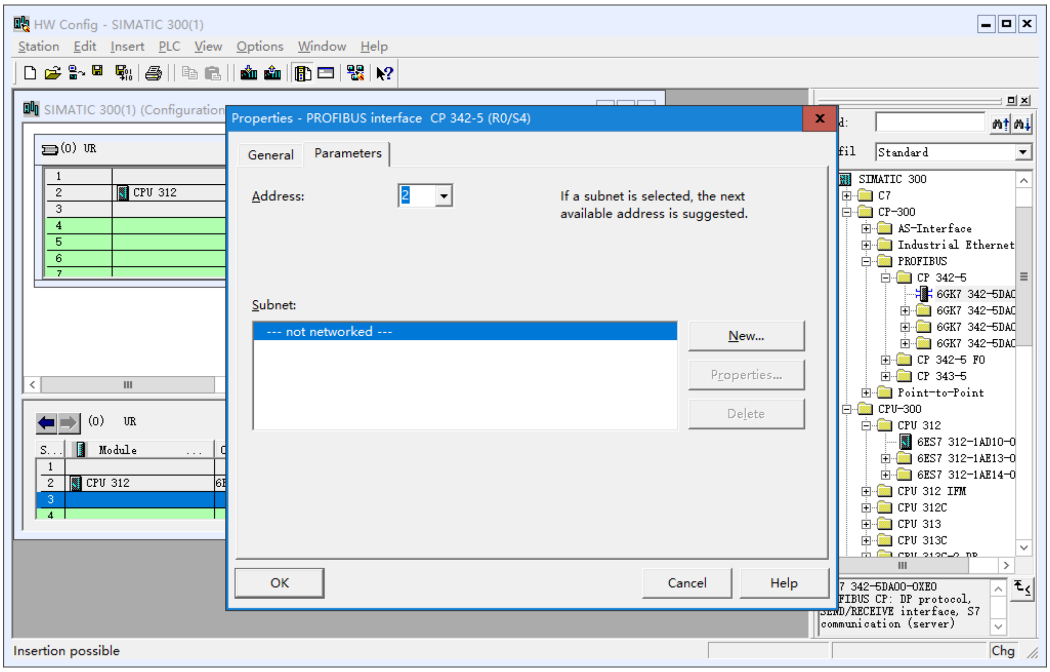


Figure H-7 Add DP Master Station CPU

In the dialog box above, set the PROFIBUS CP 342-5 interface address (as 2), and click the "New" to add a PROFIBUS connection for the DP master station. In the "Properties" dialog box, set the network parameters you need, including the transmission rate (set by default 1.5m by) and the configuration file (select "DP"), as shown below:

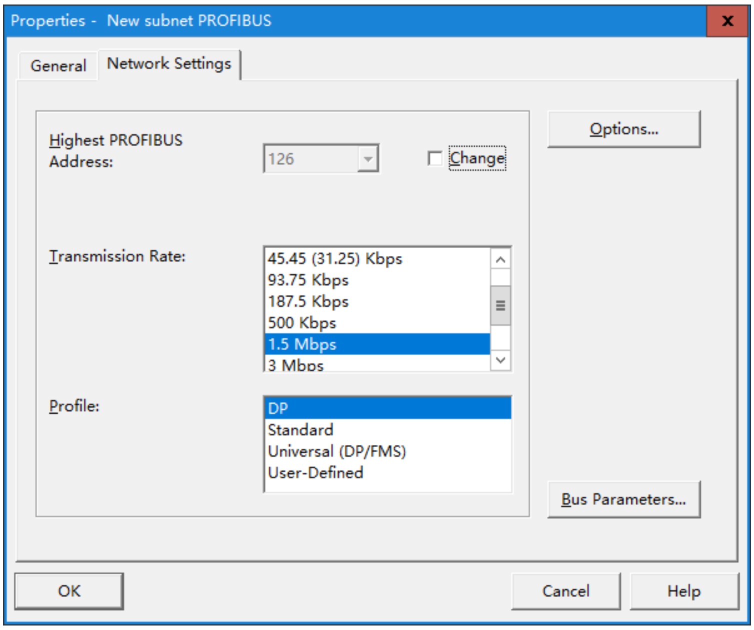


Figure H-8 Master Station Network Setting

Click “OK” to add the CPU and PROFIBUS connections of DP master station.

After confirmation, set the data I/O address and working mode of CP 342-5d. Double-click CP342-5 on the guide rail, and the properties dialog box pops up, as shown below:

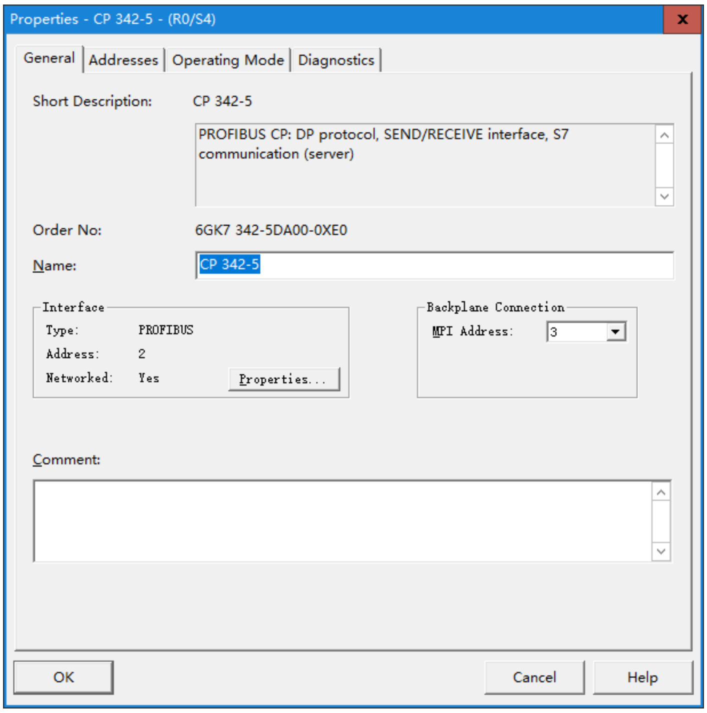


Figure H-9 Set CP 342-5D I/O Address

Set CP 342-5 I/O address

In the 【Address】 of the properties dialog box, the "Input" and "Output" start addresses are set by default (the default is 256 (16#100, the subroutine's CPLADDR parameter use this value).

Set Operating Mode of CP 342-5

Set CP 342-5 “DP master station” in [Operating Mode] of properties dialog box:

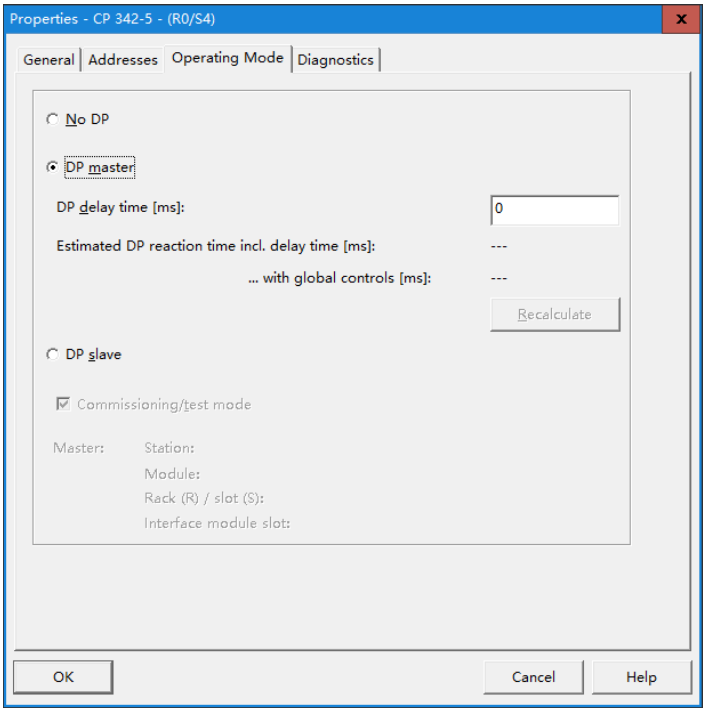


Figure H-10 Set CP 342-5D Work Mode

A dialog box pops up indicating that must call FC1 (DP_SEND) and send FC2 (DP_RECV).

After setting CP342-5 property, a PROFIBUS DP master station system will be automatically added in the blank on the right, as shown in the figure below:

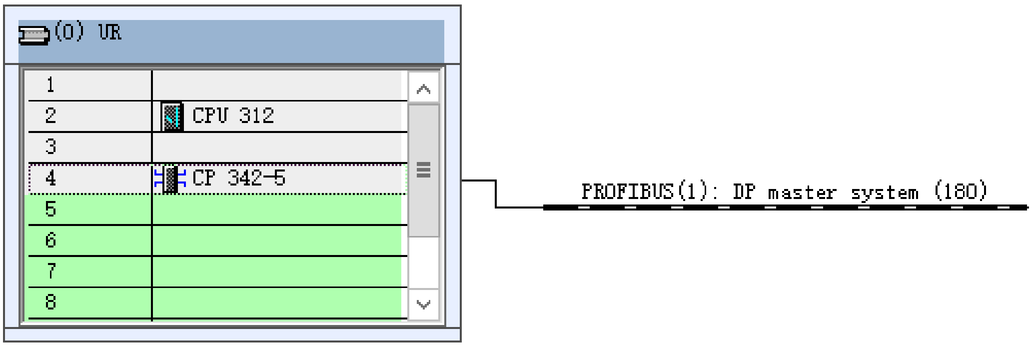


Figure H-11 CP 342-5D Attribute Set

1) Import GSD file of SM277B

First use of SMC277B, you must import GSD file involving device information in STEP 7, otherwise skip this

Install method:

In "HW Config" interface, select menu [Operations] - > [Install GSD Files], click “Browse…” to open the 277b GSD file directory in the “Import GSD files” dialog, then click "OK", show as second figure, select "ct277B. gsd", then click the "Install", after the installation click the "Close" to complete the SM277B GSD file installation.

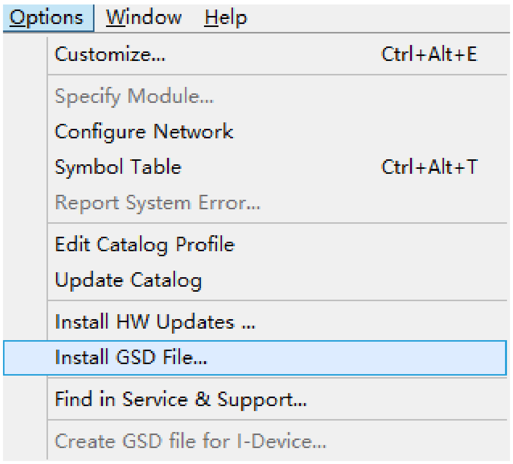


Figure H-12 Import GSD File

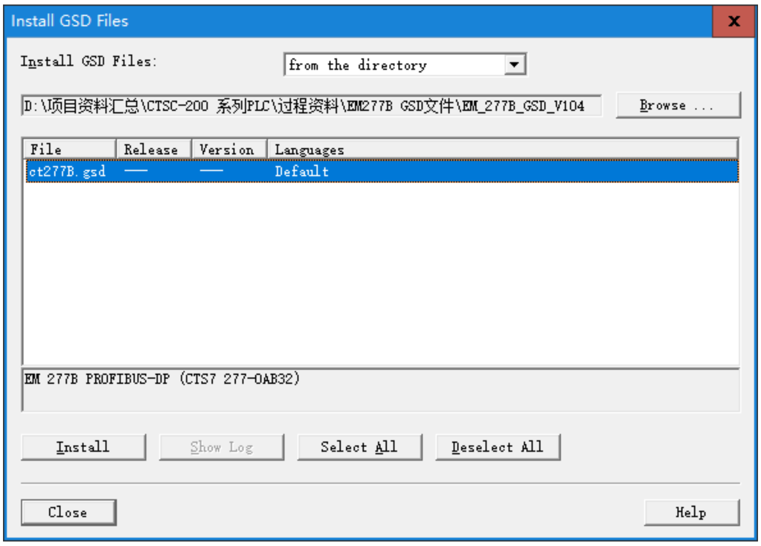


Figure H-13 Install GSD File

Note

Visit http://www.co-trust.com to download GSD file of SM277B.

You can see information of SM277B and its expansion module in “HW-Config” directory as figure below:

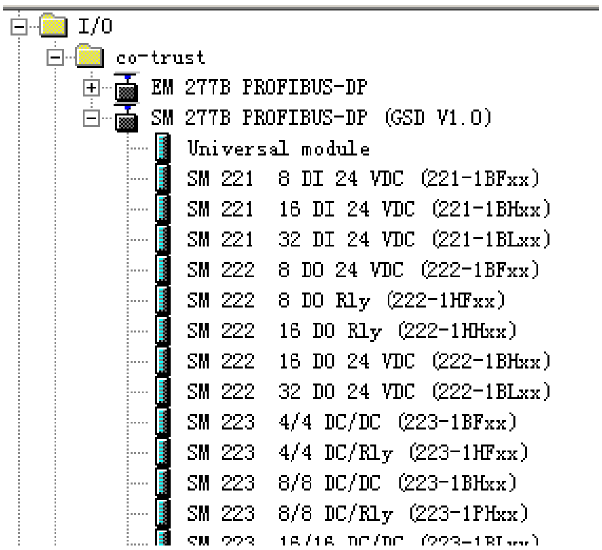


Figure H-14 SM277B information

2) Add SM 277B DP slave station

Drag SM 277 PROFIBUS-DP from the hardware directory to the "PROFIBUS (1): DP master station system (180)". As shown in the figure below, set the SM277B slave station and network in the property dialog box that pops up.

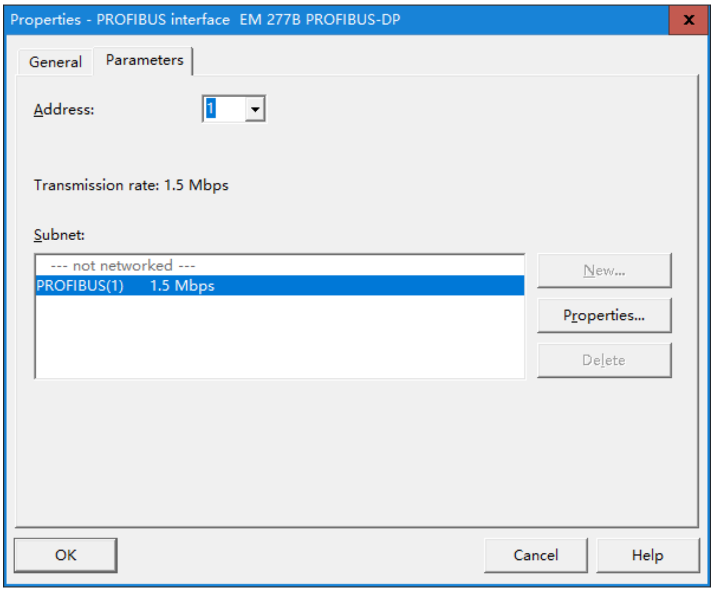


Figure H-15 Add SM 277B DP Slave Station

Set slave station address

Set the SM277B slave address in address under parameters of properties box.

Note

The valid address between 1 and 125, and is unique on PROFIBUS DP, also must be set as the address dialed by the DIP switch of SM277B module.

Set slave station network

Click the "Properties" under [Parameters] of the property box, then pop up a child dialog box as shown in the figure below:

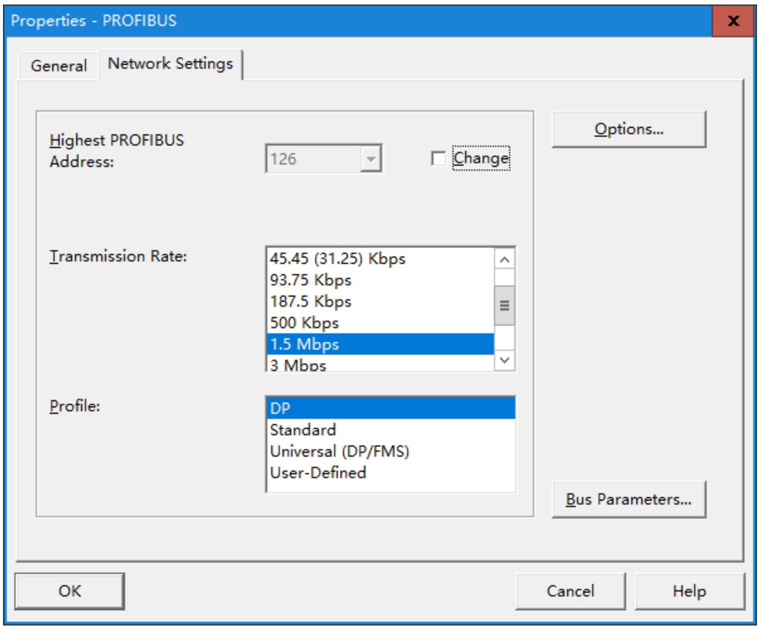


Figure H-16 Set slave station network

Click the [Network Settings] of this dialog box, and set "Transmission Rate" (default as 1.5Mbps) and "Profile" (set as "DP"). Then the addition and configuration of SM277B slave station is completed, as shown in the following figure:

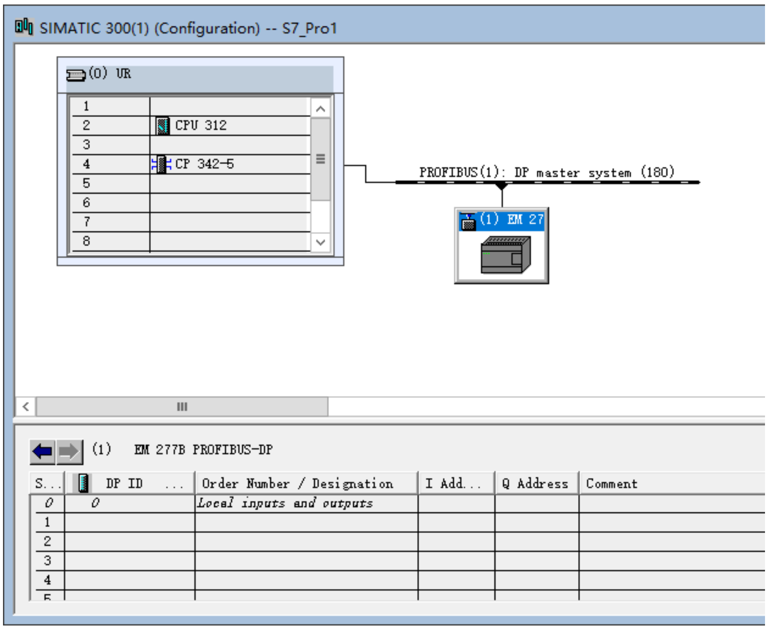


Figure H-17 slave station properties configuration

Note: ensure that the "Hex parameter assignment" of the [Parameter Assignment] of the SM277B slave station property is set by default, as shown in the figure below (00,00,00) (double-click the SM277B icon to see), otherwise there may be a communication failure.

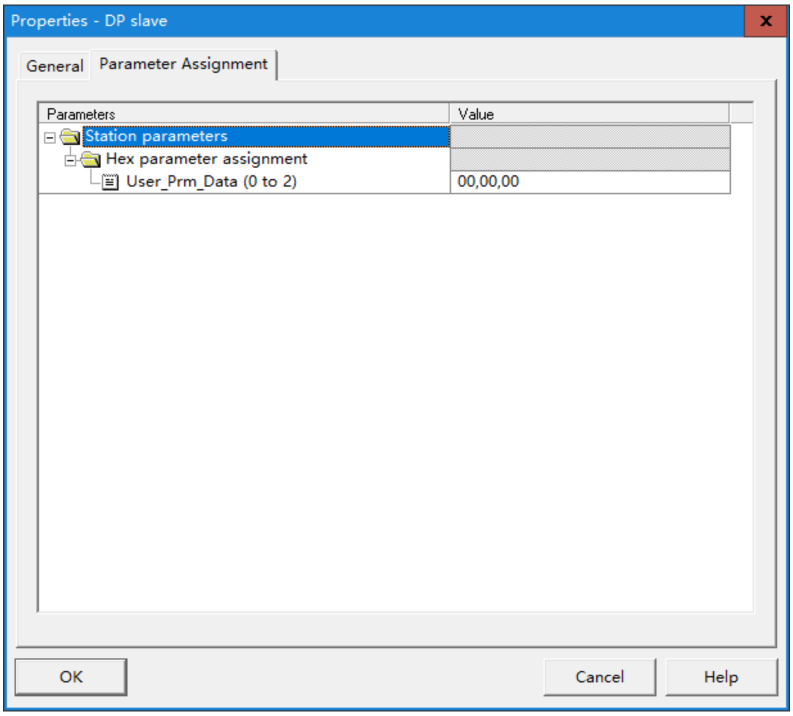


Figure H-18 DP slave station parameter assignment

※ Add expansion modules for SM277B slave station

Drag or double click the required extension module to the EM 277B blank slot number in the order you need from the hardware directory "SM 277B PROFIBUS-DP" to add DP slave station extension module, as shown in the following figure:

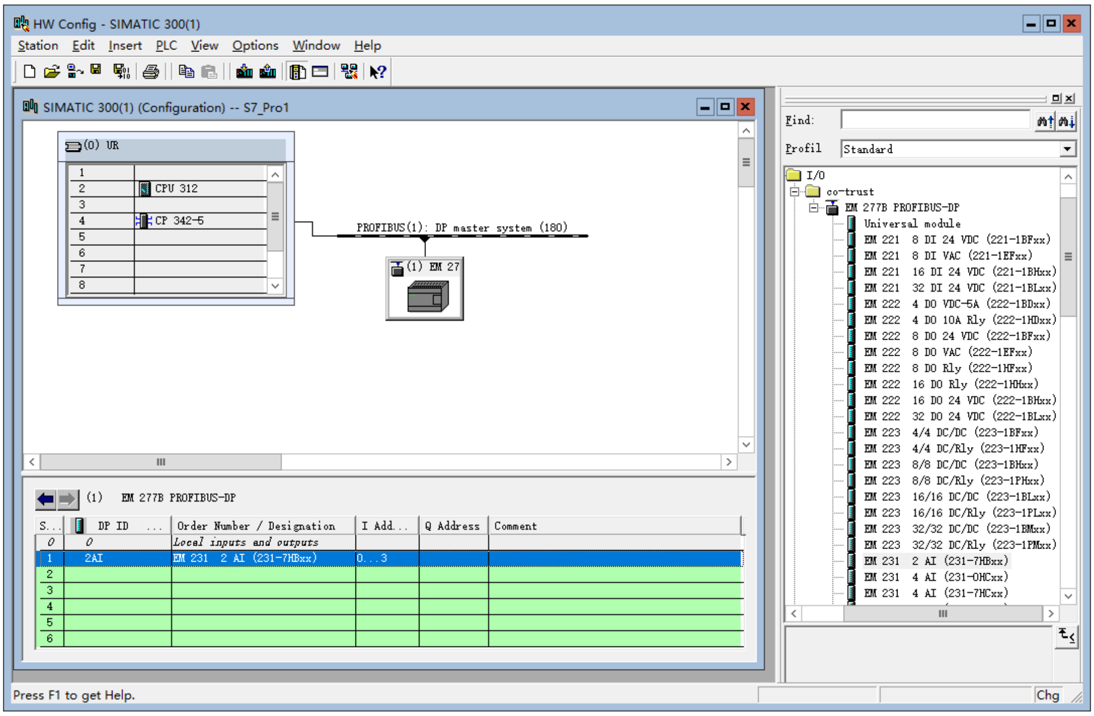


Figure H-19 Add expansion modules for SM 277B

※ Download hardware configuration to CPU of DP master station

Execute "Station" -> "Save and Compile" in of the HW Config interface.

Till now, STEP7 has completed the hardware configuration. Before sending the configuration, confirm that you have established connection between the DP master station and PC/PG (MPI connection), then execute "PLC" -> "Download" in the same interface to download the hardware configuration to the CPU of DP master station.

H.3.2 Programming and Debugging

SM277B reads input of expansion module for DP master which offer SM277B output information to write into expansion module by using LAD/STL/FBD editor to create program in OB1 and download it into CPU of main system.

【Programming】

Steps:

1) Write program to call FC1 (DP_SEND) and FC2 (DP_RECV) in OB1 to realize the DP master station reading and writing 16 bytes of data from slave station, as shown in the following figure:

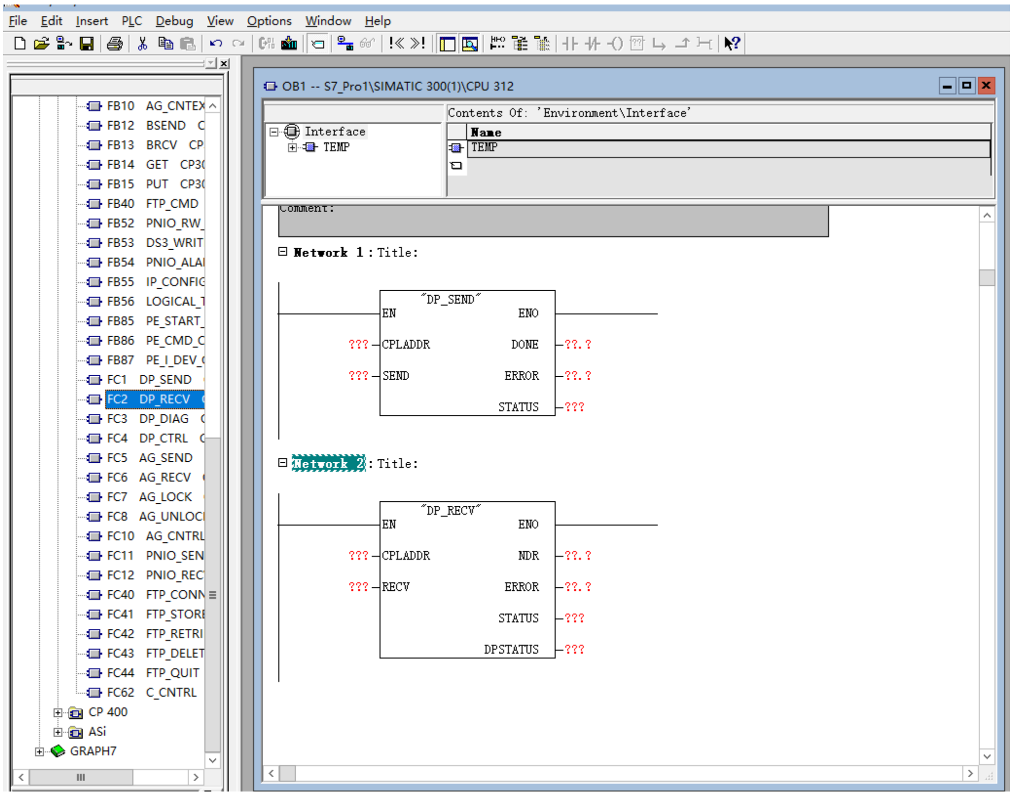


Figure H-20 Program interface

Program function description:

Execute FC1 (DP_SEND) to output CPU312 memory MB0-MB14 data to QB0-QB14 of SM277B extension module; input SM277B extension module IB0-IB14 into the MB15-MB29 memory of CPU312 by performing FC2 (DP_RECV).

CPLADDR: Address 256(16#100) of CP342-5

Note

If you select the CPU with DP communication port (e.g., CPU313C-2DP), there is no need to call FC1 and FC2 but directly access the address assigned by configuration SM277B module (e.g., IB0-IB14, QB0-QB14).

2) Save user program

【Debugging】

Steps:

1) Connect CP 342-5 and SM277B, set SM277B address and middle resistance, and connect the power supply of all devices in the system.

2) STEP7 connect to the main station and download the program to the CPU of the main station system.

3) Set the main station as RUN.

4) Set and monitor the I/O data to be transmitted and the status results of program execution in the STEP7 variable scale, observe the output status of the SM277B extension module and status indicator light (LED)

Some faults may occur in the debugging process. Please refer to Appendix *H.3.3 Diagnosis*

H.3.3 Diagnosis

Diagnose PROFIBUS DP master/slave network through LED status, STEP 7 fault information and specific programming diagnosis.

Diagnose by SM 277B

The LED status for diagnosing SM277B can provide diagnostic information about connection with the main station and self-diagnosis information of SM277B firmware.
After connecting SM 277B power supply, the LED indicator marked "ON" (green) light up.
If the BF and SF LED remain off, SM 277B will function normally.
If the BF or SF LED lights up, indicate an error in the extension module, address setting over range, wiring, or main program configuration.

Function of EM277B LED:

LED	Color	Description
ON	Green	Indicator lights on when SM277B power on, controlled by SM277B hardware
SF(system failure)	Red	If SM 277B find expansion module fault or address setting over range, the LED light up
BF(bus failure)	Red	Flicker when no DP exchange with master station

If main system run without fault, bellows are the possible fault reasons:

- STEP 7 hardware configuration didn’t match completely with equipment in network. You need to change the hardware configuration and download it to the CPU of DP master.
- STEP 7 configure wrong parameter
- The address STEP7 configured is different with SM277B actual PROFIBUS address, or SM277B actual PROFIBUS address set as 126 127(invalid address). If the SM277B actual address is correct, change the PROFIBUS address of the slave station device in STEP 7 and re-download the configuration to CPU of DP master station. If the actual address of SM277B is wrong, change it, after that SM277B needs to be cut off and re-energized.
- Not set the PROFIBUS network terminal resistance correctly. Dial "ON" if you are at the last node in the network, otherwise "OFF".
- PROFIBUS wiring error or PROFIBUS cable damaged. Check wiring and cables to make sure they are correct.

Diagnose by STEP7

【Read diagnosis information in HW-Config】
For all SIMATIC S7/M7 series modules, S7 diagnosis can be called. For the base unit and I/O modules, the structure of S7 diagnostics is the same. To make the diagnosis in STEP 7, read the diagnostic buffer of the main station device or SM 277B.

【Confirm fault】
Confirm faults in HW-Config due to steps below:

- Open project online window through Station >Open ONLINE
- Seek symbols showing diagnostic equipment state and fault condition. Press the F1 to open the help page of the diagnostic symbol to view the comments for the diagnosis.
- Select menu PLC >Faulty Modules to view fault modules list. If there is a failure, the view will not update automatically when open the online view. Select view>update to display the current state.

【Read diagnosis information in master and slave station】

In HW-config, you can open the module information to get detailed diagnostic information. Double-click for displaying the fault modules.

- For DP master, select Diagnostic Buffer symbols to view fault information
- For SM 277B station equipment: select the General symbol to view module status. Select the DP Slave diagnostics TAB, and click the Hex. Format button display the diagnostic bytes of SM277B.

H.3.4 power budget of diagnosis module

SM227B power	5 VDC power	24 VDC power
Supply	SM 277B offer a 5 VDC logic power for all expansion modules in system	SM 277B offer a 24VDC sensor power for expansion modules or anywhere needed.
Maximum usable DC power	The additional electricity sum of all SM277B expansion modules require less than 660mA	Please confirm power demand less than the budget 400mA

Power budget of SM277B, including 5VDC and 24VDC, is to calculate if there is sufficient power for its connected expansion modules.

Warning

- Do not use SM277B DC sensor and another external power to power for one device.
- Do not use two powers on one device
- Connecting an external 24VDC power to SM277B 24VDC sensor power supply will cause conflict. Which may result in shortened service life of the power supply or failure of both power supplies, which may then lead to failure of PLC system operation. Accidental operation may result in serious personal injury or even death, and/or equipment damage.

I Permanent V Memory Library “CT_savevmem”

Introduction

CT_SAVEVMEM serves as a library function to save a segment of V memory data which users need into permanent memory for a long time (about one year).

Note
Apply to save parameters, do not save too often.
Do not set power-off hold in the system block for the permanent storage V memory address to be used.

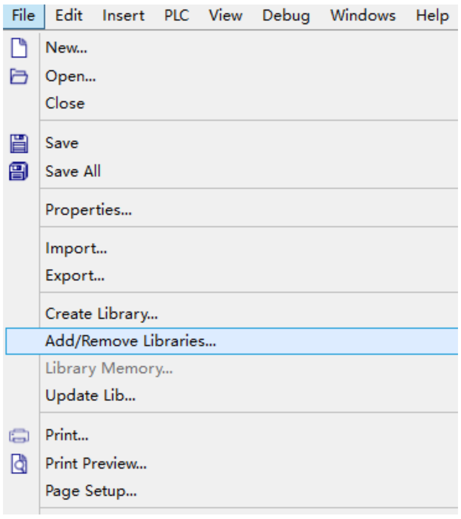
Data storage of CTH200 CPU:

CPU title	Data store space
H224	Basic 8KB, non-extendable
H224X	Basic 8KB, extend to 108KB
H226L	Basic 8KB, non-extendable
H226XL	Basic 10KB, extend to 110KB
H228XL	

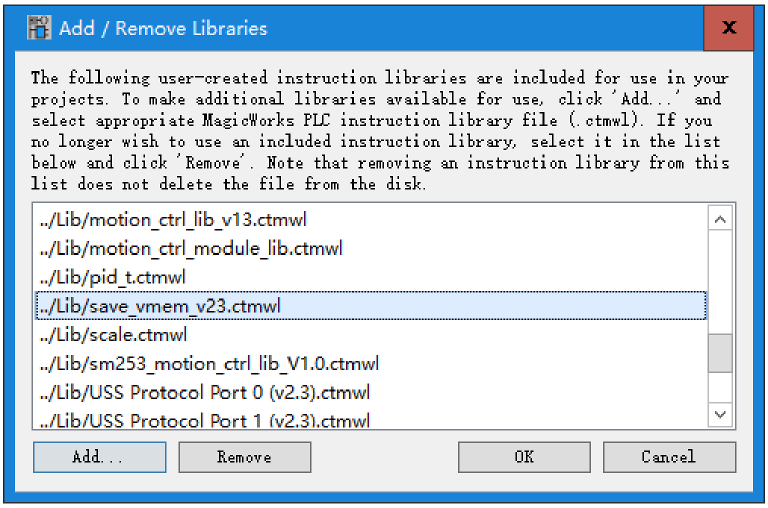
Software Installation

1) Add library file

“File” -> “Add/delete library”, find library file “ct_savevmem” as figure below:



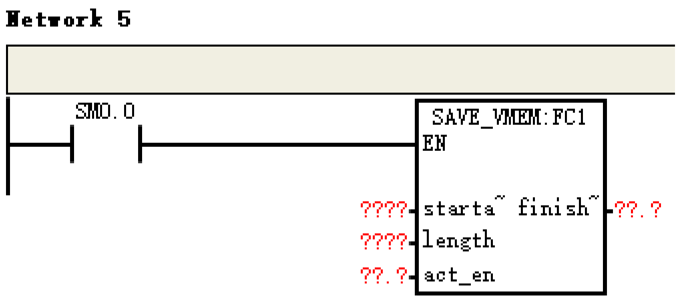
Click “Add” in dialog box popped up, find “ct_savevmem.ctmwl”, click “Save” after you find it, then select “ct_savevmem.ctmwl” in “Add/delete library”, click “OK”



You can see newly added “ct_savevmem” under directory tree after installation:

2) Call CT_SAVEVMEM library

Click “Network” where you add function block, double click “SAVE_VMEM” under “library”, it shows in “Network”



Note

Make sure EN is always ON before completing write, in other Word use sm0.0 or act_en to call.

3) CT_SAVEVMEM library function explanation

Parameter address	Explanation	Type	Note
str_addr	Start address of V memory	WORD	Can be constant or variable. Start address of VW500 is 500
length	Length unit: word	WORD	Permanently store the length of contiguous V memory. Length of VB500-VB4499 is 2000 or 2000 for VW500-VW4498
act_en	Allow write bit	BOOL	When this bit is 1, write permanent memory, automatically reset after write, and the bit is kept at 1 during the write.
finished	End write sign bit	BOOL	Automatically reset at the beginning of writing and set to 1 at the end of writing.
Note: total length of write is integer multiple of the Word.			

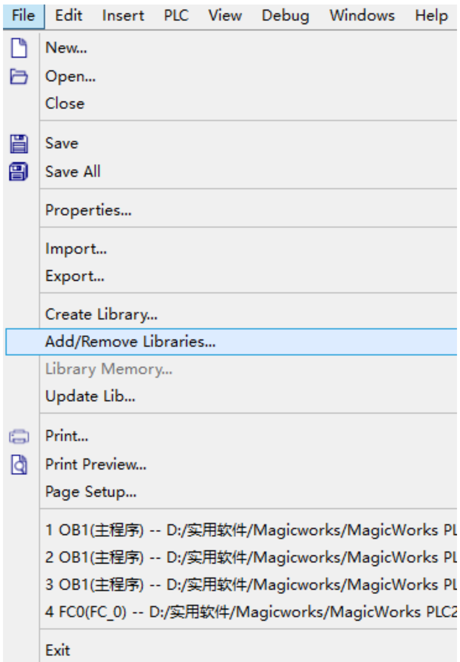
J SM253 Motion Control Library

J.1 Introduction

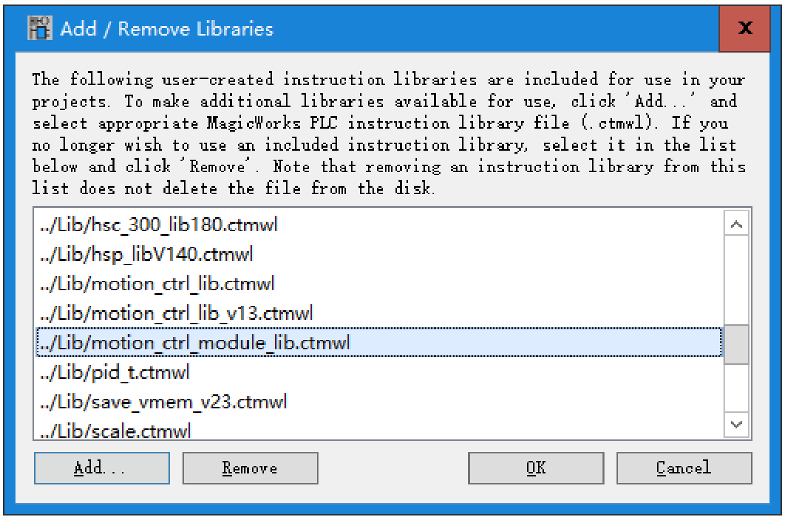
SM253 motion control library is dedicated for SM253 motion control module, able to realize multi-axis independent control.

J.2 Installation Instruction

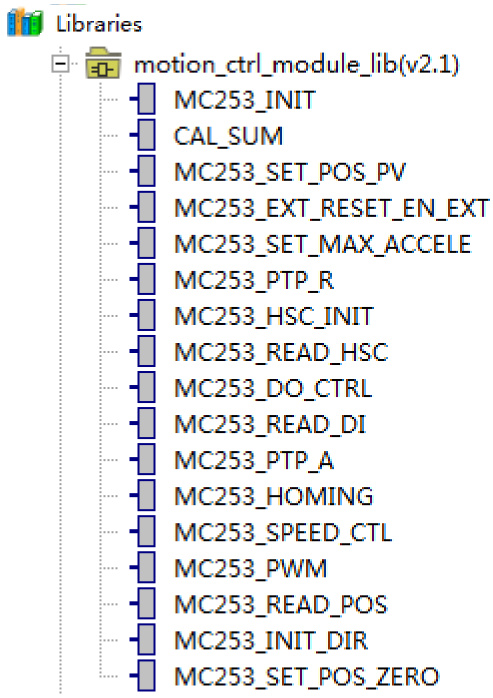
Select menu item “File” -> “Add/Remove Libraries” to find library file “motion_ctrl_module_lib.ctmwl”:



View catalogue where you save library files, select “motion_ctrl_module_lib” and click “Add”:



Click “OK” after addition, check the library node of command tree, you can see the “motion_ctrl_module_lib”:



J.3 sm253_motion_ctrl_lib Instruction List

Table J-1 Library instruction list

Function name	Instruction name	Support or not
MC253_EXT_RESET_EN	External reset coordinate enable instruction	SM253 not support
MC253_INIT_DIR	Configure motor direction instruction	Support
MC253_READ_POS	Read position instruction	Support
MC253_PTP_R	Uniaxial relative motion instruction	Support
MC253_CIRCLE_R	Two-axis arc interpolation motion instruction	SM253 not support
MC253_SPEED_CTL	Speed control instruction	Support
MC253_SET_POS_ZERO	Software back to zero instruction	Support
MC253_SET_POS_PV	Set target location instruction	Support
MC253_LINE_R	Two-axis line interpolation motion instruction	SM253 not support
MC253_EXT_RESET_EN_EXT	External reset coordinate enable instruction II	Support
MC253_SET_MAX_ACCELE	Set maximum acceleration instruction	Support
MC253_SET_CI_MODE	Set continuous interpolation instruction	SM253 not support
MC253_PWM	PWM instruction	Support
MC253_INIT	Initialize motion control module instruction	Support and must be called first

MC253_DO_CTRL	Control module output instruction	Support
MC253_READ_DI	Read module input status instruction	Support
MC253_HSC_INIT	Set module HSC instruction	Support
MC253_READ_HSC	Read module HSC instruction	Support
MC253_PTP_A	Uniaxial absolute motion instruction	Support
MC253_HOMING	Homing instruction	Support

J.4 sm253_motion_ctrl_lib Library Instruction Explanation

For Micro/Win programming, the instruction format of the operation control module is the same as that of CPU, but differ in the instruction library. Prefix of CPU instruction is MC, while the motion control module EM253 is MC253. The axis number of CPU instruction is 0~3, while the motion control module instruction is numbered from 0, which can be continued down and System automatically assigns to each module. For example, the motion control module instruction axis number is 0~1, which corresponds to the 0 and 1 axis of the first module respectively. The motion control module instruction axis number is 2~3, corresponding to the 0 and 1 axis of the second motion control module respectively. By analogy, if the motion control module out of scope, consider the parameter error. For example, there is only one motion control module (with only two axes), motion control module specifies axis 2, but it cannot find the corresponding value, so that it cannot be used.

Similarly, the module HSC is not uniformly numbered with CPU HSC, which must call the module instruction for use. The module HSC numbered from 0 and can be continued down. For example, the HSC of module instruction label 0~1, which corresponds to HSC0 and HSC1 of the first module respectively; HSC of module instruction 2~3 corresponds to HSC0 and HSC1 of the second module respectively.

※ Attention

1 Initialization and library storage area

Using the SM253 motion control module require to initialize the system control variables in the initial call MC253_INIT of the system program (it is needed only once, that is, only be called by SM0.1). Note that the EM253 motion control module use V memory space. Users can specify the library storage area address themselves through "library storage area allocation". Users are not allowed to use the address space occupied by the library in program. (if only one SM253 module hangs after the CPU, the library occupies 218 bytes; 236 bytes for 2. 254 bytes for 3; 272 bytes for 4. 290 bytes for 5.)

2 Analog filter

When using SM253 motion control module, you must clear filtering function at the corresponding channel of SM253 motion control in the upper computer software system block -> input filter -> analog, and download the system block to the PLC. Otherwise, the SM253 motion control module library instruction cannot be use (check the PLC information, where the 4 in and 4 out analog module channel is corresponding to that of the SM253 motion control module.).

3 Communication status bit

SM253 newly increased all the communication bit other than MC253_INIT, the status bit will alarm to indicate communication fault and then you should check if the bus connection is normal or the analog filter of motion control channel has been cleared.

4 Pulse output instruction execution order

Coaxial pulse output instructions (MC253_PTP_R, MC253_SPEED_CTL, MC253_PWM) have up to 3 data buffers, that is, only process 3 instructions at the same time. When there are multiple instructions on the same axis (such as all of them are 0 axis), if the buffer unfell (there are less than 3 enabled output instructions), then the new enabled instructions can be processed in time. Then, the execution order of the instructions is consistent with the time order of the enabled instructions, which is not necessarily related to the instructions position order. If the buffer is full, the enabled output command is more than 3), the new enable instruction cannot be processed in time, and can only be enabled after the end of the instruction and release the buffer before according to place order for processing by the program scan, which is related to order of execution and position, but not necessarily according to the time sequence of instructions. Simply understood as "under 3 by time, full 3 by position" (as shown in example 1).

In order to achieve the trajectory, and if the position order of these motion instructions is consistent with the desired trajectory, all these instructions can be enabled at one time. If not, you must enable the corresponding instruction of the enabled trajectory when the enabled instruction is less than 3 (as shown in example 2).

Case 1) Enable all instructions at a time

The enable instruction (instruction A/B/C) is executed in the enable order when the buffer is less than 3 copies. As following table, enable all the five instructions at one time, and the order is: instruction A--> instruction C--> instruction B--> instruction E-> instruction D, as shown in table 1; Then, the enabled 3 instructions are executed successively in: instruction A--> instruction C--> instruction B, as shown in column 3 of table:

Instruction enable order	Instruction position order	Instruction execute order
1	0—AXIS 1—RUN Instruction A	1
3	0—AXIS 1—RUN Instruction B	3
2	0—AXIS 1—RUN Instruction C	2
5	0—AXIS 1—RUN Instruction D	--

4	0—AXIS 1—RUN Instruction E	--

The remaining enabled instructions (instruction D/E) execute in position order when the buffer is filled with 3 portions. After the execution of instruction A, the buffer is released. As the program scans from top to bottom, instruction D obtains the buffer first. Instruction E obtains the buffer after instruction C finishes executing and releases the buffer.

The instruction enable order: instruction A--> instruction C--> instruction B--> instruction E-> instruction D, as shown in table 1;The order of instruction execution: instruction A--> instruction C--> instruction B--> instruction D-> instruction E, as shown in table 3:

Instruction enable order	**Instruction position order**	**Instruction execute order**
1	0—AXIS 1—RUN Instruction A	1
3	0—AXIS 1—RUN Instruction B	3
2	0—AXIS 1—RUN Instruction C	2
5	0—AXIS 1—RUN Instruction D	4
4	0—AXIS 1—RUN Instruction E	5

Case 2) Enable 3 instructions first, then enable the rest after instructions completely executed to release buffer

The enable instruction (instruction A/B/C) is executed in the enable order when the buffer is less than 3 copies, as following table: enable 3 instructions first, the order is: instruction A -- > instruction C --> instruction B, as shown in table 1, execute in order successively, the execution order is: instruction A -- > instruction C --> instruction B, as shown in column 3 of table:

Instruction enable order	**Instruction position order**	**Instruction execute order**
1	0—AXIS 1—RUN Instruction A	1
3	0—AXIS 1—RUN Instruction B	3

2	0—AXIS 1—RUN Instruction C	2
--	0—AXIS 0—RUN Instruction D	--
--	0—AXIS 0—RUN Instruction E	--

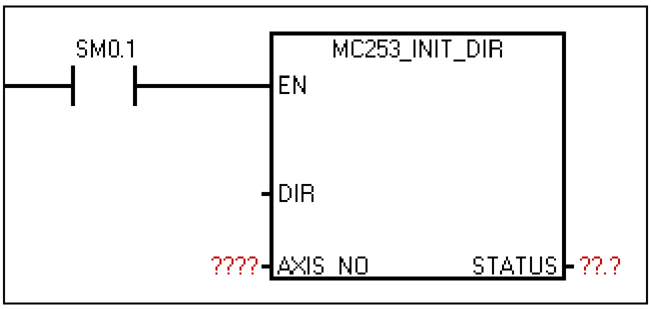
After instruction A completes execution and releases the buffer, enable instruction E first, then it obtains the buffer; Instruction D acquires the buffer after instruction C completes execution and releases the buffer.

The order of instruction enable is: instruction A--> instruction C--> instruction B--> instruction E-> instruction D, as shown in table 1;The order of instruction execution is: instruction A--> instruction C--> instruction B--> instruction E-> instruction D, as shown in column 3 of table:

Instruction enable order	Instruction position order	Instruction execute order
1	0—AXIS 0—RUN Instruction A	1
3	0—AXIS 0—RUN Instruction B	3
2	0—AXIS 1—RUN Instruction C	2
5	0—AXIS 1—RUN Instruction D	5
4	0—AXIS 1—RUN Instruction E	4

Configure motor direction instruction

① Function name: MC253_INIT_DIR



② Function: configure motor direction

Note

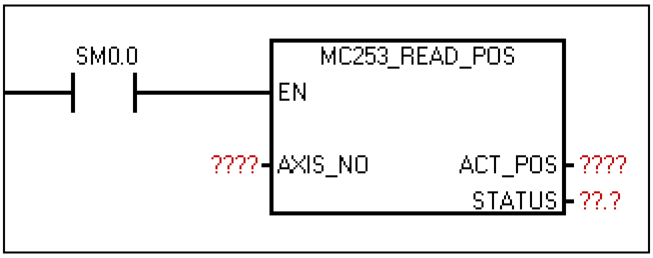
Execute one time on first scan cycle of CPU power on

③ Parameter

Name	I/O	Description	Type	Value range	Note
DIR	IN	Configure effective level when the direction signal is positive. DIR=1, set motor forward when corresponding direction axis output “1” DIR=0, set motor reverse when corresponding direction axis output “0”	Bool	0~1	Default: 1, motor forward when direction axis output
AXIS_NO	IN	Set axis number(2 axes for each EM253 module, axis number range decided by motion control modules number)	Byte	0~255	
STATUS	OUT	Communication status sign bit 1: Communication timeout	Bool	0~1	

Read position instruction

① Function name: MC253_READ_POS



② Function: Read the absolute coordinate value of each axis. Once the origin coordinate is set, the value will calculate according to relationship between the output pulse and direction: positive rotation output a pulse: +1, while reverse output a pulse: -1. What you end up with is an absolute coordinate with the set point as the origin.

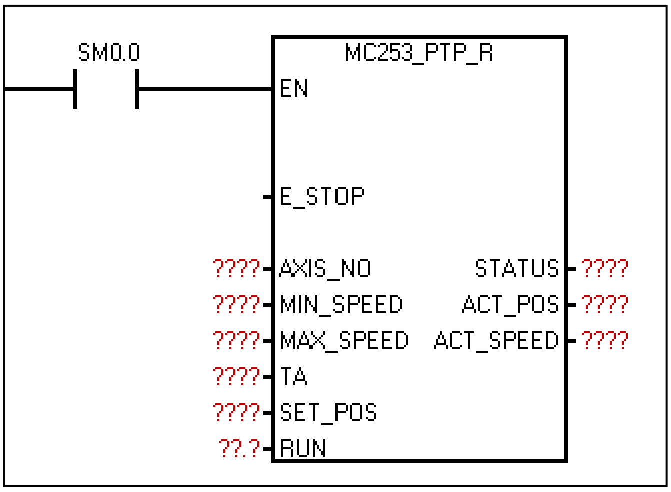
③ Parameter

Name	I/O	Description	Type	Value range	Note
AXIS_NO	IN	Set axis number(2 axes for each EM253 module, axis number range decided by motion control modules number)	Byte	0~255	

ACT_P OS	OUT	Absolute coordinate of current axis(1 pulse indicate 1 unit coordinate)	Dint	-2147483648~ +2147483647	No error status output for this instruct, axis number must set correctly.
STAT US	OUT	Communication status sign bit 1: communication timeout	Bool	0~1	

Uniaxial relative motion instruction

① Function name: MC253_PTP_R



② Function: Used for uniaxial point - to - point control (uniaxial fixed - length drive). Call one time output fixed pulse, by setting the maximum, minimum speed and deceleration time, the output pulse will gradually accelerate to the largest speed, when the pulse number is ready to run, cut down pulse frequency automatically to prevent vibration or jammed due to too much inertia when start or stop the machine.

③ Parameter

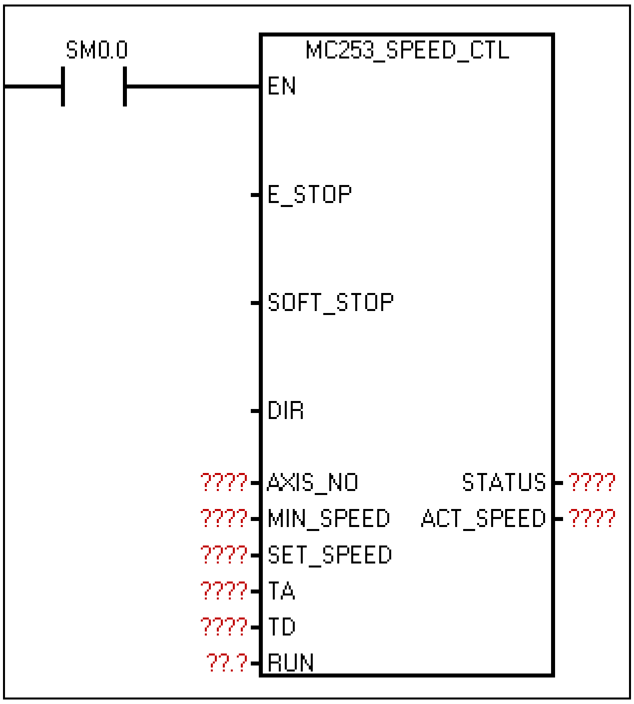
Name	I/O	Description	Type	Value range	Note
E_STOP	IN	Emergency stop bit 1: Valid, 0: Invalid	Bool	0/1	1 Run only when Run =1 and E_Stop =0 2 RUN reset internal when E_STOP = 1
AXIS_NO	IN	Set axis number, 0/1/2/3	Byte	0~3	Unchangeable in the process
MIN_SPE ED	IN	Minimum speed of run or stop. Unit: HZ	Dword	500~200000	1 Set minimum speed less than maximum speed 2 Changeable in the process
MAX_SP EED	IN	Maximum speed of run Unit: HZ	Dword	500~200000	
TA	IN	Time of ACC/DEC, Unit: ms	Dword	0~10000	Changeable in the process; if TA=0, no settings for acceleration.
RUN	IN	Run enable bit	BOOL	0~1	1 Run only when

		1: valid			RUN =1 and E_STOP =0. 2 RUN reset internal when run complete. 3 When E_STOP=1, RUN reset internal.
STATUS	OUT	7 6 5 4 3 2 1 0 Output status byte: Bit0: parameter configuration error sign 1—parameter configuration error 0—parameter configuration normal Bit1: Run sign 1—Running 0—Do not run Bit2: Completion sign 1—Instruction execution completed 0—Instruction execution do not completed Bit3: Busy sign 1—Valid, the axis occupied by other instruction 0—Invalid, executing instruction or execution done.	BYTE	0~255	Bit0: 1 Judge Only to axis parameter configuration error and homing mode out of range. 2 No error reported for other parameters, it will set to the nearest reasonable value automatically. 3 Report parameter failure if TA=0 and the maximum acceleration is not set; TD the same
SET_POS	IN	Output pulse. The positive output pulse indicates the positive direction along the X-axis, and the negative pulse number indicates the negative direction along the X-axis.	DInt	-2147483647 ~ +2147483647	Modifiable during operation. When the new set value is greater than the number of pulses output, the final output pulse will be subject to the new set value. When the new set value is less than the number of output pulses, the pulse output will be stopped immediately
ACT_PO	OUT	Absolute coordinate of	DInt	-2147483647	This instruction no

S		current axis (1 pulse 1 units coordinate)		~ +2147483647	error and set the axis number correctly.
ACT_SPEED	OUT	Current run speed	Dword	500~200000	Note: this value may be a little off actual one, at most 5K, related to acceleration time and setting speed

Speed control instruction

① Function name: MC253_SPEED_CTL



② Function: Control the frequency of uniaxial output pulse, the output pulse frequency (speed) can be changed any time. When receive the soft stop command, it will automatically slow down to stop. When receive the emergency stop command, the pulse output will stop immediately without any deceleration.

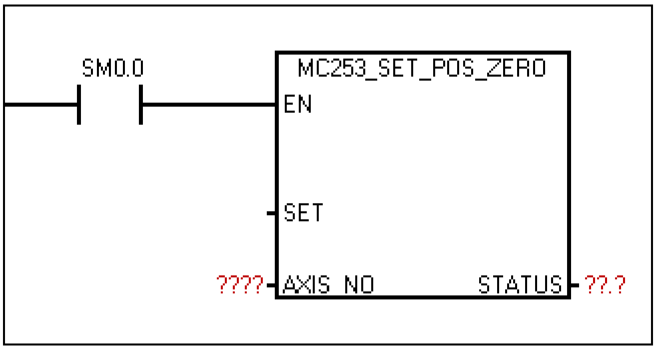
③ Parameter

Name	I/O	Description	Type	Value	Note
RUN	IN	Run enable bit 1: Valid	BOOL	0~1	1 Run only when RUN =1 and E_STOP =0. 2 RUN reset internal when run complete. 3 when E_STOP=1, RUN reset internal.
E_STOP	IN	Emergency stop bit 1: Valid, 0: Invalid	BOOL	0~1	1 Run only when RUN =1 and E_STOP =0. 2 When E_STOP=1, RUN reset internal.
SOFT_STOP	IN	Soft stop bit1: valid 0: invalid Output pulse decelerate to stop when receive the instruction	Bool	0/1	
DIR	IN	Pulse direction bit	Bool	0/1	Unmodifiable in the process
AXIS_NO	IN	Set axis number, 0/1/2/3	Byte	0~3	unmodifiable in the process
MIN_SPEED	IN	Minimum speed of run or stop. Unit: HZ	Dword	0~200000	1 Minimum speed should less than setting speed 2 Modifiable in the process
SET_SPEED	IN	Setting speed, output pulse will accelerate or decelerate to this speed before stop command	Dword	0~200000	
TA	IN	Time of ACC Unit: ms	Dword	0~10000	Modifiable in the process; Calculate acceleration only when startup and TA/TD changes
TD	IN	Time of ACC, from setting time to minimum time Unit: ms	Dword	0~10000	Unmodifiable in the process

STATUS	OUT	7 6 5 4 3 2 1 0 Output status byte: Bit0: parameter configuration error sign 1—Parameter configuration error 0—Parameter configuration normal Bit1: Run sign 1—Running 0—Do not run Bit2: Completion sign 1—Instruction execution completed 0—Instruction execution do not completed Bit3: Busy sign 1—Valid, the axis occupied by other instruction 0—Invalid, executing instruction or execution done.	BYTE	0~255	Bit0: 1 Judge Only to axis parameter 2 No error reported for other parameters, it will set to the nearest reasonable value automatically. 3 Report parameter failure if TA=0 and the maximum acceleration is not set; TD the same
ACT _SPEED	OUT	Current actual speed	DWORD	0~200000	Current actual speed

Software back to zero instruction

① Function name: MC253_SET_POS_ZERO



② Function: Reset absolute coordinate

Note

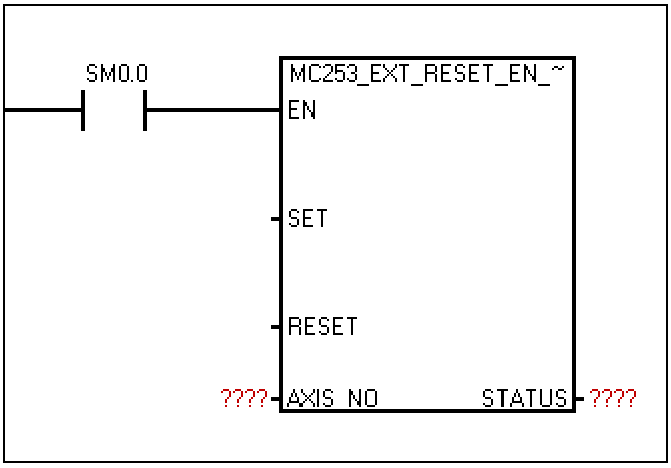
Call the instruction when the machine moves to a certain position, which is equivalent to setting the origin of the axis at that position. Each time you call the "read absolute coordinates" command, you get a coordinate value relative to that point.

③ Parameter

Name	I/O	Description	Type	Value range	Note
SET	IN	Enable reset bit Set absolute coordinate 0 at SET rising edge, set SET 0, then set 1 for each call	Bool	0~1	
AXIS_NO	IN	Set axis number(2 axes for each EM253 module, axis number range decided by motion control modules number)	Byte	0~255	
STATUS	OUT	Communication status sign bit 1: Communication timeout	Bool	0~1	

External reset coordinate enable instruction II

① Function name: MC253_EXT_RESET_EN_EXT



② Function: set if enable external IO reset absolute coordinate

Note

Corresponding relation between axis number and external reset signal:

Axis 0 ——I0.2 (MC253_HSC0)

Axis 1 ——I0.6 (MC253_HSC1)

① Parameter

Name	I/O	Description	Type	Value range	Note
SET	IN	Enable reset bit Set absolute coordinate 0 at SET rising edge, set SET bit 0, then set 1 for each call	Bool	0~1	
RESET	IN	RESET rising edge, forbid to enable external reset, RESET	Bool	0~1	

		then set 0 for each call			
AXIS_NO	IN	Set axis number(2 axes for each EM253 module, axis number range decided by motion control modules number)	Byte	0~255	
STATUS	OUT	Status bit: 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 Bit0: Reset status sign bit 1—Reset complete 0—Reset uncompleted Bit1~Bit6: reserved Bit7: Communication status sign bit 1—Communication timeout 0—No timeout	Byte	0~255	

④ Explanation

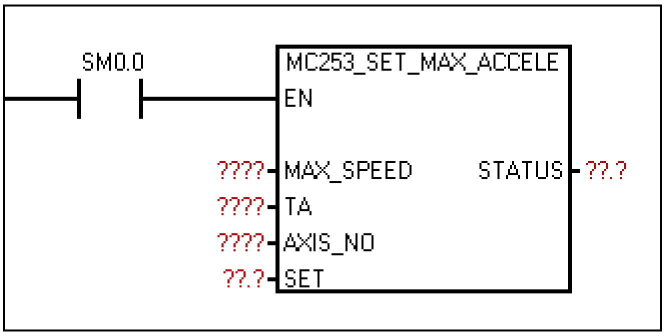
Use 0 axis to call this instruction. After the SET rising edge enables the external reset function and I0.2 detect the "effective reset signal", the system resets the axis 0 absolute coordinate, and also the STATUS setting reset complete. After the RESET rising edge prohibits the external reset function, even if I0.2 detect the "effective reset signal", the system will not reset the 0 axis absolute coordinate, and the STATUS reset instruct is non-reset state.

Note

The so-called "effective reset signal", the reset signal of each axis corresponding to an external IO, and the corresponding register set its effective level. For example, 0 axis corresponds to I0.2, and HSCO control register SM37.0 set the effective reset level of 0 axis. When set to 0, the effective reset signal of 0 axis is in I0.2 high level. When set to 1, the effective reset signal of axis 0 is in I0.2 low level; This setting is only effective if and only if the corresponding high-speed counter (0 axis corresponds to HSC0) is enabled, otherwise (no high-speed counter is enabled) the system default high level is the effective reset signal. If 0 axis, I0.2 high level is the effective reset signal. In the same way for other axes, the corresponding relation of the relevant control of each axis is shown in ② of this section.

Set maximum acceleration instruction

① Function name: MC253_SET_MAX_ACCELE



② Function: set maximum acceleration (= MAX_SPEED/TA) (TA≠0) (if the instruction isn't called, indicate the maximum acceleration has not been set)

③ Parameter

Name	I/O	Description	Type	Value range	Note
MAX_SPEED	IN	Maximum speed of long-axis run Unit: HZ	Dword	0~200000	Modifiable in the process
TA	IN	Time of ACC Unit: ms	Dword	0~10000	Modifiable in the process Calculate acceleration only when startup and TA/TD changes
AXIS_NO	IN	Set axis number(2 axes for each EM253 module, axis number range decided by motion control modules number)	Byte	0~255	
SET	IN	After ensuring parameter above, give SET a rising edge to make it effective.	Bool	0~1	
STATUS	OUT	Communication status sign bit 1: Communication timeout	Bool	0~1	

④ Explanation

If set the X-axis parameter TA=0, or the X-axis does not call this instruction, it is considered that the X-axis does not set maximum acceleration; Otherwise, MAX_ACCELE=MAX_SPEED/TA is considered to be set as the X-axis maximum acceleration. The significance of this instruction is:

a. Set an appropriate acceleration to limit the acceleration of each instruction on an axis.

Such as PTP instruction, set AXIS_NO=0, MIN_SPEED=1000, MAX_SPEED=11000, TA=500, then in theory the acceleration of PTP motion is 20HZ/ms (= (MAX_SPEED-MIN_SPEED)/TA); if 0 axis call MC_SET_MAX_ACCELE and set maximum acceleration 15HZ/ms, the PTP actual acceleration is 15HZ/ms (the same for MC_LINE_R and MC_CIRCLE_R)

b. An instruction on a certain axis is to obtain the maximum acceleration to run

If the PTP instruction run at the maximum acceleration, call MC_SET_MAX_ACCELE instruction on the coaxes to set the maximum acceleration (the parameter TA of MC_SET_MAX_ACCELE instruction cannot be 0, otherwise cannot obtain the maximum acceleration), and simultaneously set the parameter TA of PTP instruction to 0.

If the maximum acceleration is not set and the PTP instruction TA=0, the alarm parameter fails on the PTP instruction. (Same for MC_LINE_R and MC_CIRCLE_R)

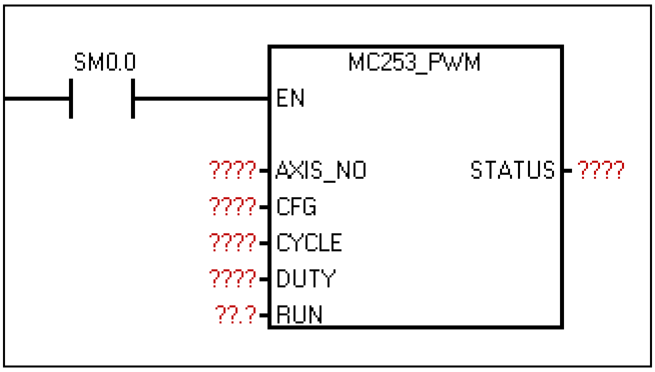
Note

1. For the biaxial instructions MC_LINE_R and MC_CIRCLE_R, if both set the maximum acceleration, the smaller one will be taken as the maximum acceleration of the biaxial system. If only one axis is set with the maximum acceleration, then take it as the maximum acceleration of the biaxial system. If the maximum acceleration is not set on both axes, the biaxial system has no limitation for acceleration.

2. After MAX_SPEED, TA, AXIS_NO have been determined, give SET a rising edge, so these parameters are updated and effective.

PWM instruction

① Function name: MC253_PWM



② Function: Output different period pulse and duty ratios by setting cycle and duty ratio parameters.

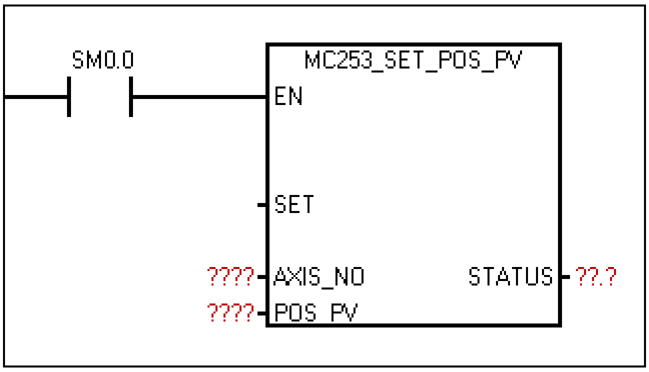
③ Parameter

Name	I/O	Description	Type	Value range	Note
AXIS_NO	IN	Set axis number(2 axes for each EM253 module, axis number range decided by motion control	Byte	0~255	Unmodifiable in the process

		modules number)			
CFG	IN	Reference time unit 0: 1us, 1: 0.5ms	Byte	0~1	Unmodifiable in the process
CYCLE	IN	Pulse period	Word	2~65535	Unmodifiable in the process
DUTY	IN	Pulse duty factor	Word	0~65535	Unmodifiable in the process
RUN	IN	Enable run	Bool	0~1	
STATUS	OUT	Output status byte: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bit0: Parameter configuration error sign 1—Parameter configuration error 0—Parameter configuration normal Bit1: Run sign 1—Running 0—Do not run Bit2: Completion sign 1—Instruction execution completed 0—Instruction execution do not completed Bit3: Busy sign 1—Valid, the axis occupied by other instruction 0—Invalid, executing instruction or execution done. Bit4: Modules Estop sign 1—Sign is valid, the axis is forbidden run by external condition. 0—Sign is invalid Bit5~Bit6: reserved Bit7: Instruction communication status sign 1—Communication timeout 0—No timeout	Byte	0~255	Bit0: ● Judge only to axis parameter configuration error ● CYCLE/DUTY is set to the nearest reasonable value without error reported. ● 253 PLC module is not identified or not connect to the module will prompt parameters configuration errors

Set target position instruction

① Function name: MC253_SET_POS_PV



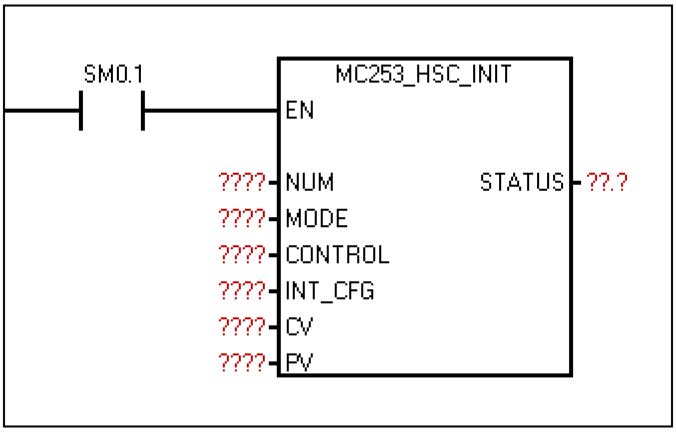
② Function: This instruction is used to write the absolute location of the machine to the module. For example, if the power cut off when machine runs to a certain position, save this position. When next time power up, it can be written back to the module, so that the starting point of counting the module absolute position is consistent with the machine actual starting point, no need to return to the origin. If this position happens to be the origin, this instruction has the same effect as MC253_SET_POS_ZERO.

③ Parameter

Name	I/O	Description	Type	Value range	Note
AXIS_NO	IN	Set axis number(2 axes for each EM253 module, axis number range decided by motion control modules number)	Byte	0~255	
SET	IN	After ensuring parameter above, give SET a rising edge to make it effective.	Bool	0~1	
POS_PV	IN	Set target position, in positive and negative. Output positive pulse indicates along X-axis forward direction, negative pulses indicate reverse.	Dint	-2147483648 ~ +2147483647	
STATUS	OUT	Communication status sign 1—communication timeout	Bool	0~1	

Set module HSC instruction

① Function name: MC253_HSC_INIT



② Function: Configure HSC

Note

Execute one time on first scan cycle of CPU power on

③ Parameter

Name	I/O	Description	Type	Value range	Note
NUM	IN	HSC sign(2 axes for each EM253 module, axis number range decided by motion control modules number)	Byte	0~255	
MODE	IN	Count mode	Byte	0~12	Not support all, refer to attached table below
CONTROL	IN	Control Word: 7 6 5 4 3 2 1 0 Bit0: Reset effective level control bit 1—Low level 0—High level Bit1: Reserved Bit2: Orthogonal counter speed selection 1—1x velocity 0—4x velocity Bit3: Position control bit 1—Count addition 0—Count decrease Bit4: Update direction 1—Update direction 0—No update Bit5: update default	Byte	0~255	

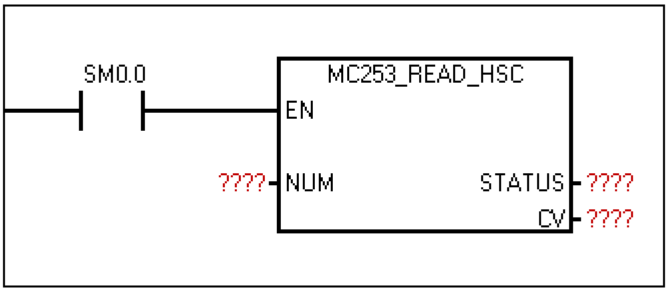
		1—Write new default 0—No update Bit6: Update current value 1—Write new current value 0—No update Bit7: effective bit 1—Effective 0—Ineffective			
INT_CFG	IN	Interrupt configuration(reserved)	Byte	reserved	reserved
CV	IN	New current value	Dword	-2147483648 ~ +2147483647	
PV	IN	New default value	Dword	-2147483648 ~ +2147483647	
STATUS	OUT	Communication status sign 1—Communication timeout	Bool	0~1	

Attached Table

Mode	Description	Input		
	MC253_HSC0	I0.0	I0.1	I0.2
	MC253_HSC1	I0.4	I0.5	I0.6
0	Single phase counter with internal direction control	Clock		
1		Clock		Reset
3	Single phase counter with external direction control	Clock	Direction	
4		Clock	Direction	Reset
9	A/B phase orthogonal counter	Clock A	Clock B	
10		Clock A	Clock B	Reset

Read modules HSC status instruction

① Function name: MC253_READ_HSC



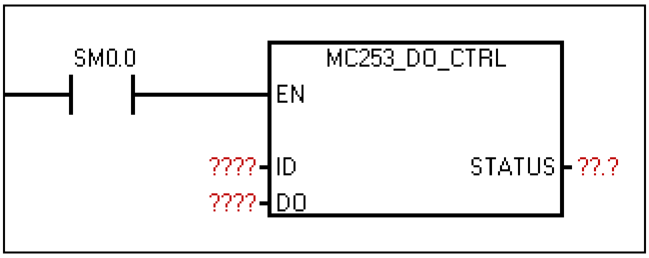
② Function: Read HSC count status and current value

③ Parameter

Name	I/O	Description	Type	Value range
NUM	IN	HSC sign(2 axes for each EM253 module, axis number range decided by motion control modules number)	Byte	0~255
STATUS	OUT	Status Word: 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 Bit0: Communication status bit 1—Communication timeout Bit1~Bit4: reserved Bit5: current count direction bit 1—count addition Bit6: current value equal to default 1—equal Bit7: current value greater than default 1—greater than	Byte	0~255
CV	OUT	New current value	Dword	-2147483648 ~ +2147483647

Control modules output instruction

① Function name: MC253_DO_CTRL



② Function: Control output of motion control module

③ Parameter

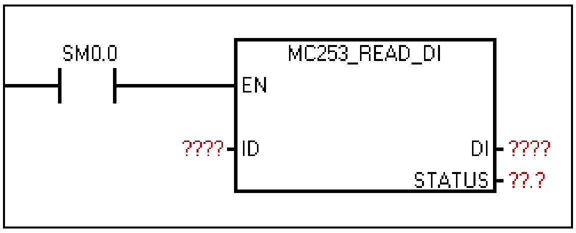
Name	I/O	Description	Type	Value range	Note
ID	IN	Motion control module ID number	Byte	0~255	Apply only to motion control module
DO	IN	Module output port value	Byte	0~255	DO corresponds to 8 Q output, low byte corresponds to Q low bit, such as DO bit0 corresponds to Q0.0
STATUS	OUT	Communication status sign 1—communication timeout	Bool	0~1	

Note

- The number of motion control module is only internally unified. If the PLC has 3 modules, among which module 0 and 2 is the motion control module, module 1 is not, then module 0 is ID0, module 1 is uncontrollable for it is not motion control module, and module 2 is ID1.
- If a single motion control module is attached behind the CPU, the delay from enable to execution of the output instruction is about 780us due to transmission delay. If there are up to 5 motion control modules hanging behind the CPU, the delay is about 930us.

Read modules input status instruction

① Function name: MC253_READ_DI



② Function: Read input of motion control module

③ Parameter

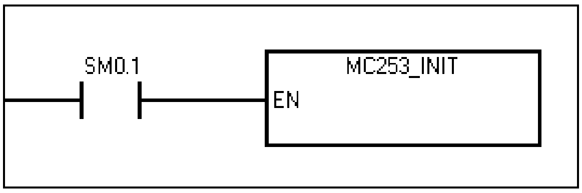
Name	I/O	Description	Type	Value range	Note
ID	IN	Motion control module ID number	Byte	0~255	Apply only to motion control module
DI	OUT	Module output port value	Byte	0~255	DI corresponds to 8 I input, low byte corresponds to I low bit, such as DI bit0 corresponds to I0.0
STATUS	OUT	Communication status sign 1—communication timeout	Bool	0~1	

Note

- The number of motion control module is only internally unified. If the PLC has 3 modules, among which module 0 and 2 is the motion control module, module 1 is not, then module 0 is ID0, module 1 is uncontrollable for it is not motion control module, and module 2 is ID1.
- When an axis call motion control instruction, if confirm to operating conditions, the running process of the axis port output pulse or direction according to the instruction (corresponding port is not controlled by the MC253_DO_CTRL), the port can restore normal IO functions after run (accept instruction MC253_DO_CTRL control); use instruction MC253_DO_CTRL carefully so as not to affect the use of motion control.

Initialize motion control module instruction

① Function name: MC253_INIT

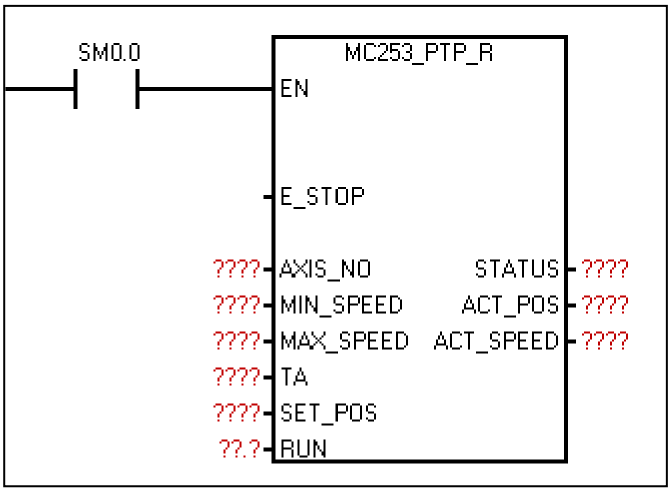


② Function: Initialize the system control variable of motion control module

③ Parameter: NONE

Uniaxial relative motion instruction

① Function name: MC253_PTP_R



② Function: Used for uniaxial point - to - point control (uniaxial fixed - length drive). Call one time output fixed pulse, by setting the maximum, minimum speed and deceleration time, the output pulse will gradually accelerate to the largest speed, when the pulse number is ready to run, cut down pulse frequency automatically to prevent vibration or jammed due to too much inertia when start or stop the machine.

③ Parameter

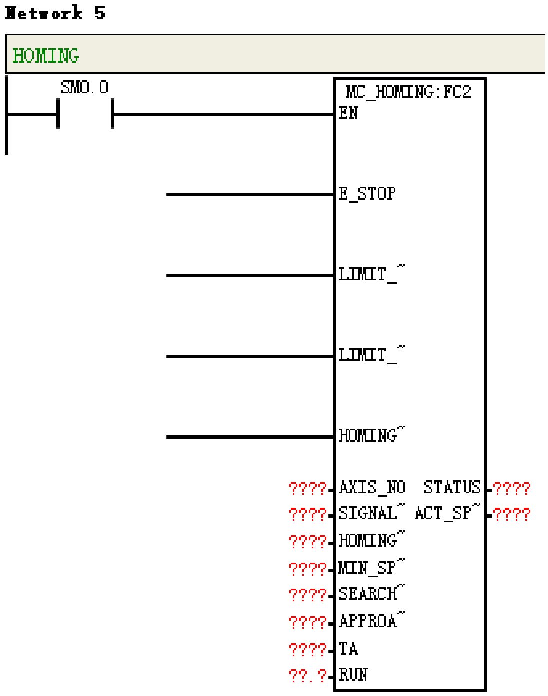
Name	I/O	Description	Type	Value range	Note
E_STOP	IN	Emergency stop bit 1: Valid, 0: Invalid	Bool	0/1	1 Run only when Run =1 and E_Stop =0 2 RUN reset internal when E_STOP = 1
AXIS_NO	IN	Set axis number, 0/1/2/3	Byte	0~3	Unchangeable in the process
MIN_SPEED	IN	Minimum speed of run or stop. Unit: HZ	Dword	500~200000	1 Set minimum speed less than maximum speed 2 Changeable in the process
MAX_SPEED	IN	Maximum speed of run Unit: HZ	Dword	500~200000	
TA	IN	Time of ACC/DEC,	Dword	0~10000	Changeable in the

		Unit: ms			process; if TA=0, no settings for acceleration.
RUN	IN	Run enable bit 1: Valid	BOOL	0~1	1 Run only when RUN =1 and E_STOP =0. 2 RUN reset internal when run complete. 3 when E_STOP=1, RUN reset internal.
STATUS	OUT	7 6 5 4 3 2 1 0 Output status byte: Bit0: Parameter configuration error sign 1—Parameter configuration error 0—Parameter configuration normal Bit1: Run sign 1—Running 0—Do not run Bit2: Completion sign 1—Instruction execution completed 0—Instruction execution do not completed Bit3: Busy sign 1—Valid, the axis occupied by other instruction 0—Invalid, executing instruction or execution done.	BYTE	0~255	Bit0: 1 Judge Only to axis parameter configuration error and homing mode out of range. 2 No error reported for other parameters, it will set to the nearest reasonable value automatically. 3 Report parameter failure if TA=0 and the maximum acceleration is not set; TD the same
SET_POS	IN	Output pulse. The positive output pulse indicates the positive direction along the X-axis, and the negative pulse number indicates the negative direction along the X-axis.	DInt	-2147483647 ~ +2147483647	Modifiable during operation. When the new set value is greater than the number of pulses output, the final output pulse will be subject to the new set value.

					When new value is less than output pulses number, the pulse output will be stopped immediately
ACT_POS	OUT	Absolute coordinate of current axis (1 pulse 1 units coordinate)	DInt	-2147483647 ~ +2147483647	This instruction no error and set the axis number correctly.
ACT_SPEED	OUT	Current run speed	Dword	500~200000	Note: this value may be a little off actual one, at most 5K, related to acceleration time and setting speed

Homing instruction

① Function name: MC253_HOMING



② Function: Find device origin by setting parameters such as homing mode. The relationship between the axis number and external reset IO signal (such as homing Z pulse):

Axis 0 → I0.2 (MC253_HSC0)

Axis 1 → I0.6 (MC253_HSC1)

If the homing mode refers to the origin switch (mode 3 or 4), connect origin switch signal (the HOMING_SW parameter of the instruction) to the corresponding point above; otherwise, the origin cannot be found.

③ Parameter

Name	I/O	Description	Type	Value range	Note
E_STOP	IN	Emergency stop bit 1: Valid, 0: Invalid	BOOL	0~1	1 Run only when RUN =1 and E_STOP =0. 2 When E_STOP=1, RUN reset internal.
LIMIT_SW_CCW	IN	CCW limit switch	BOOL	0~1	
LIMIT_SW-CW	IN	CW limit switch	BOOL	0~1	
HOMING_SW	IN	Origin switch	BOOL	0~1	
AXIS_NO	IN	Axis number	BYTE	0~255	Unchangeable in the process
SIGNAL_TYPE	IN	7 6 5 4 3 2 1 0 Signal type: Bit0: Counterclockwise stroke limit input signal type 0—High level 1—Low level Bit1: Clockwise stroke limit input signal type 0—High level 1—Low level Bit2: Origin switch signal type 0—High level 1—Low level	BYTE	0~255	
HOMING_MODE	IN	Homing mode	BYTE	1~14	Refer to Appendix *E.4 Back to Zero Mode Illustration*
MIN_SPEED	IN	Minimum speed Unit: Hz	DWORD	0~200000	1 Pulse output close when speed < 5Hz. 2 Unchangeable in the process 3 Search speed should not be too large, close speed should be as small as possible
SEARCH_SPEED	IN	Origin search speed Unit: Hz	DWORD	0~200000	
APPROACH_SPEED	IN	Origin close speed Unit: Hz	DWORD	0~200000	

TA	IN	Time of ACC/DEC. Unit: ms	DWORD	0~10000	Changeable in the process
RUN	IN	Enable run bit 1: Valid	BOOL	0~1	1 Run only when RUN =1 and E_STOP =0. 2 When E_STOP=1, RUN reset internal.

④ Explanation

The program detect each input by scanning (the Z phase is not affected by this), so when the switch value changes and process not in time, there may be some delay. If the homing speed (including search and close speed) is too high, the processing delay will be amplified, resulting in inaccurate return.

Note: 1 When TA≠0, acceleration= (MAX_SPEED-MIN_SPEED) /TA(limited by maximum acceleration if it is set before); When TA=0, use MC253_SET_MAX_ACCELE to set maximum acceleration, parameter will report error if not set. Same for TD, for two-axis instruction, if both axes have set maximum acceleration, then adopt smaller one as system acceleration.

2 In theory, acceleration =【(MAX_SPEED - MIN_SPEED) / TA】. If the desired acceleration is too small (less than 1), the instruction default internal acceleration is 1. Users can reasonably set TA/TD according to the expected acceleration.

3 Import "Module emergency stop signal" in module input port: when the module detects this signal, forbid to output pulse, and alarm at the status of pulse output instructions (such as MC253_PTP_R, MC253_SPEED_CTL and MC253_PWM instructions). The corresponding relation between axis number and emergency stop signal:

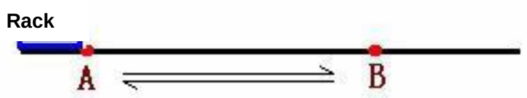
Axis 0 → I0.3

Axis 1 → I0.7

J.5 Debugging Example

Hybrid motion control instruction

Making stepping motor whose subdivision is 1000 and screw lead is 5mm to do to-and-fro movement, a distance of 2000mm.



【System Explanation】

Attach one SM253 module and set it as point-point movement parameter of axis 0 after CPU in this example. Call MC253_PTP_R to set controlling parameter.

I0.2 is hardware zero reset of A point which will be set as mechanical origin under limit switch input.

I1.0 is emergency stop input of system;

Q0.0 is pulse output, Q0.1 is direction output

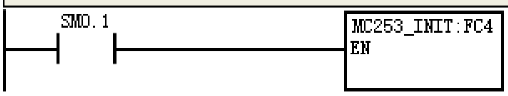
【Program Block】

Program comment:
Use CPU H226XL to control stepping motor whose subdivision is 1000 and screw lead is 5mm to do to-and-fro movement, a distance of 2000mm.
Attach one SM253 module and set it as point-point movement parameter of axis 0 after CPU in this example. Call MC253_PTP_R to set controlling parameter.
I0.2 is hardware zero reset of A point which will be set as mechanical origin under limit switch input.
I1.0 is emergency stop input of system;
Q0.0 is pulse output, Q0.1 is direction output

Program idea:
First, call speed control instruction to make machine back to zero before startup;
Second, call single axis relative motion instruction for point-point movement after zero;
Third, realize to-and-fro movement between A and B by changing the value of SET_POS in MC_PTR_R instruction

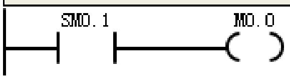
Network 1

Use SM0.1 to call MC253_INIT to initialize system for module MC253



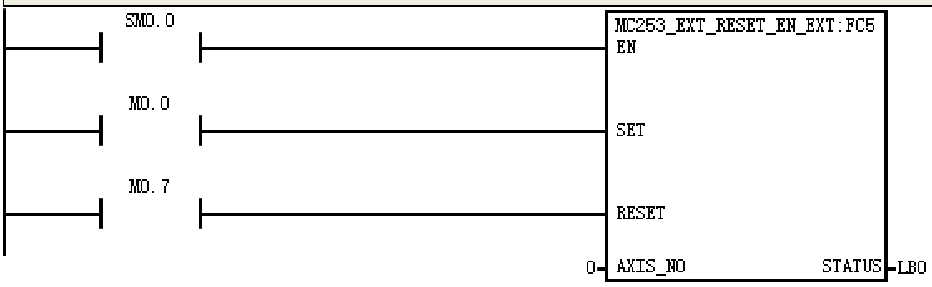
Network 2

Initialize the corresponding bit



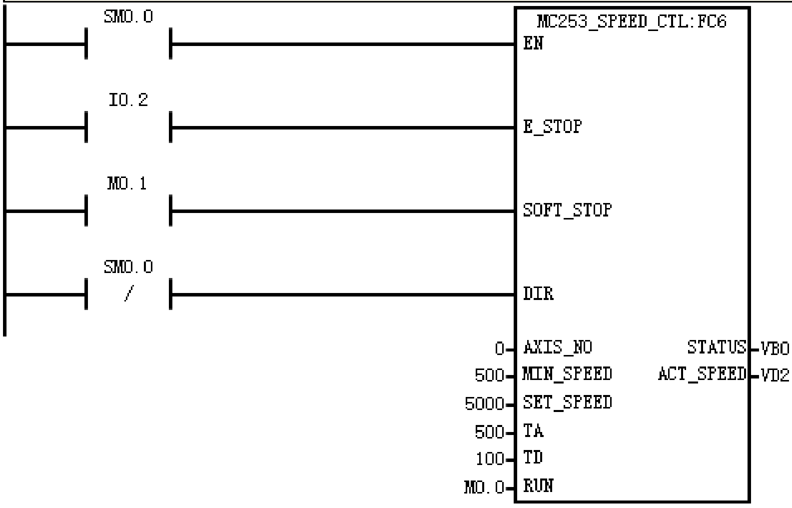
Network 3

Program comment:
Enable external reset coordinate instruction
Function: realize machine zero
M0.0----Run bit of origin(set by HMI), enable external reset bit by rising edge
M0.7----The rising edge prohibit external reset bit



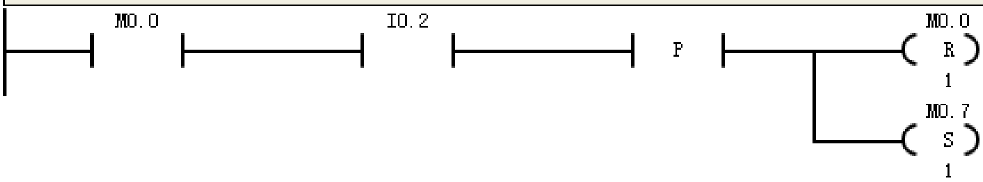
Network 4

Program comment:
Speed control instruction
Function: control axis 0 to do reverse rotate back to mechanical origin, a movement startup of A point.
M0.0----startup enable bit;
I0.2-----travel limit switch, emergency stop switch of instruction ;
M1.0----soft stop bit;
Reverse pulse direction to origin;
Axis number is 0;
The startup/stop speed is 500Hz, the normal speed after acceleration;
Acceleration time is 500ms;
Deceleration time is 100ms;
Put the output status Word into VB0;
Put the current speed(frequency) into VD2



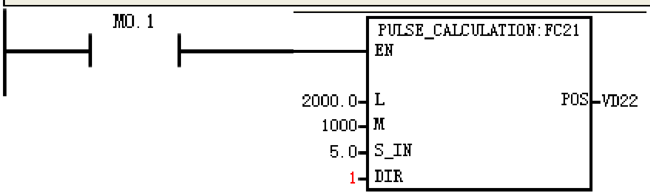
Network 5

Reset the corresponding bit after back to zero



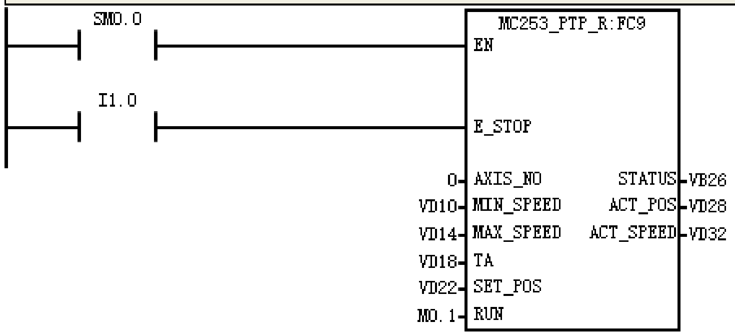
Network 6

M0.1-------movement enable bit from point A to point B
Displacement between A and B is 2000mm
The subdivision of stepping motor is 1000
Screw lead is 5.0mm
DIR is 1(1---corotation; 0---reverse rotation)
Put the pulse into VD22



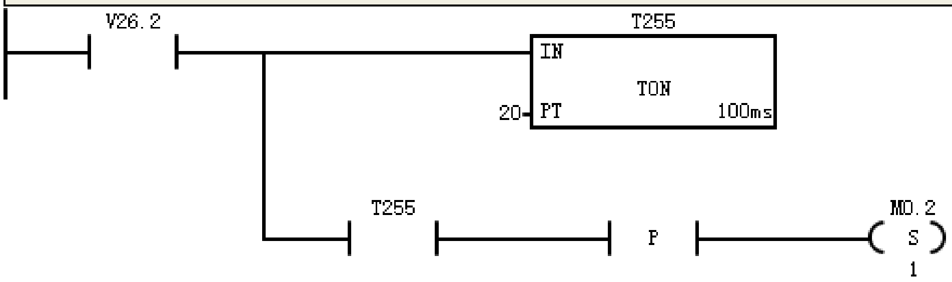
Network 7

Single axis relative motion control instruction
Function: motion control between point A and point B
I1.0--------emergency stop bit
Axis number is 0
VD10------starting/stop speed
VD14------normal operation speed after acceleration
VD18------acceleration/deceleration time
VD22------output pulse
M0.1-------run enable bit
VB26------Output status byte
VD28------output pulse number
VD32------current actual operation speed



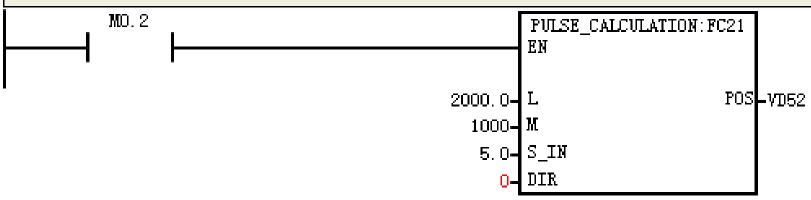
Network 8

Timing 2s to set M0.2 reverse rotation



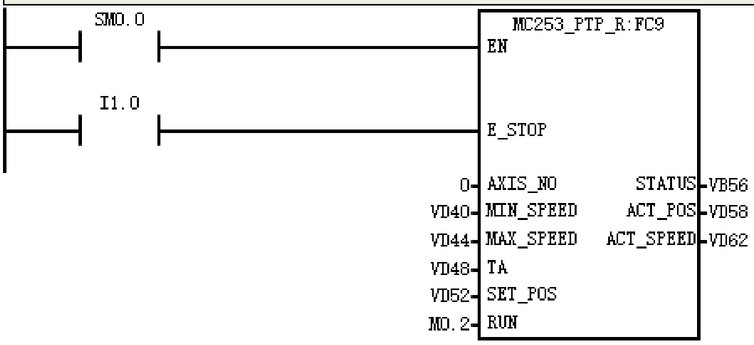
Network 9

Timing 2s to set M0.2 reverse rotation
M0.2-------movement enable bit from point B to point A
Displacement from point B to point A is 2000mm
The subdivision of stepping motor is 1000
Screw lead is 5.0mm
DIR is 0(1---corotation, 0---reverse rotation)
Put the pulse number into VD52



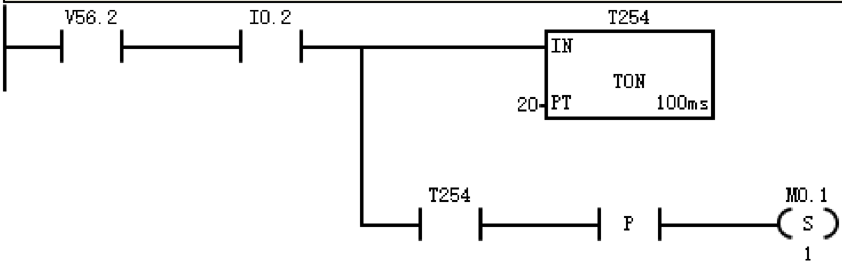
Network 10

Single axis relative motion control instruction
Function: motion control between point A and point B
I1.0--------emergency stop bit
Axis number is 0
VD40------starting/stop speed
VD44------normal operation speed after acceleration
VD48------acceleration/deceleration time
VD52------output pulse
M0.2-------run enable bit
VB56------Output status byte
VD58------output pulse number
VD62------current actual operation speed



Network 11

Timing 2s to set M0.1 corotation



K Instructions of ETHERNET_SET

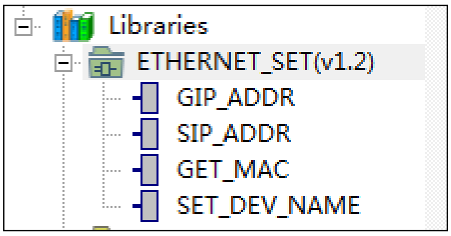
K.1 Instruction

Instructions ETHERNET_SET (V1.2) is used to set the remote communication parameters of MagicWorks PLC system block, you can set and get the IP address, MAC address and device name of the EtherNET PLC without stopping CPU, as well as remote program and monitor PLC via MICO.

K.2 Instruction Explanation

It's available for free download the ETHERNET_set library file from COTRUST official website: www.co-trust.com.

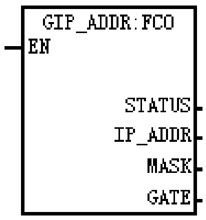
The library file includes the following parameters:



Get IP address library

① **Library name**: GIP_ADDR

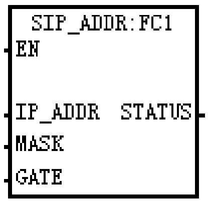
② **Function:** Get IP address.

Parameters	Name	IN/OUT	Data Type	Explanation
	EN	IN	BOOL	Enable, SM0.0 is allowed to invoke.
	STATUS	OUT	BYTE	Status word bit0=1, means successfully acquired the address. bit1=1 means failed to get the address.
	IP_ADDR	OUT	DWORD	IP address, four bytes in total, each byte shows the IP address from low to high.
	MASK	OUT	DWORD	Subnet mask, four bytes in total, each byte shows the subnet mask number from low to high.
	GATE	OUT	DWORD	Gateway, four bytes in total, each byte shows the gateway number from low to high.

Set IP address library

① **Library name:** SIP_ADDR

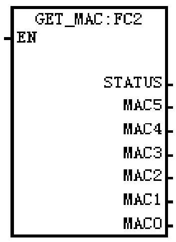
② **Function:** Set IP address.

Parameters	Name	IN/OUT	Data Type	Explanation
	EN	IN	BOOL	Enable, edge triggered. The instruction is not allowed to circularly invoke, cause set IP and EPPROM has limits for number of times, and the EtherNET communication would be failed due to frequently write.
	IP_ADDR	IN	DWORD	IP address, four bytes in total, each byte shows the IP address from low to high.
	MASK	IN	DWORD	Subnet mask, four bytes in total, each byte shows the subnet mask number from low to high.
	GATE	IN	DWORD	Gateway, four bytes in total, each byte shows the gateway number from low to high.
	STATUS	OUT	BYTE	Status word bit0=1, means set successful. bit1=1, means illegal IP address. bit2=1, means IP and mask is not match. Bit3=1, means IP and gateway is not match.

Get MAC address library

① **Library name:** GET_MAC

② **Function:** Get MAC address

Parameters	Name	IN/OUT	Data Type	Explanation
	EN	IN	BOOL	Enable, SM0.0 is allowed to invoke.
	STATUS	OUT	BYTE	Status word bit0=1, means successfully acquired the address.
	MAC0~ MAC5	OUT	BYTE	MAC Address xx:xx:xx:xx:xx:xx MAC5------MAC0

Set device name library

① **Library name:** SET_DEV_NAME

② **Function:** Set device name.

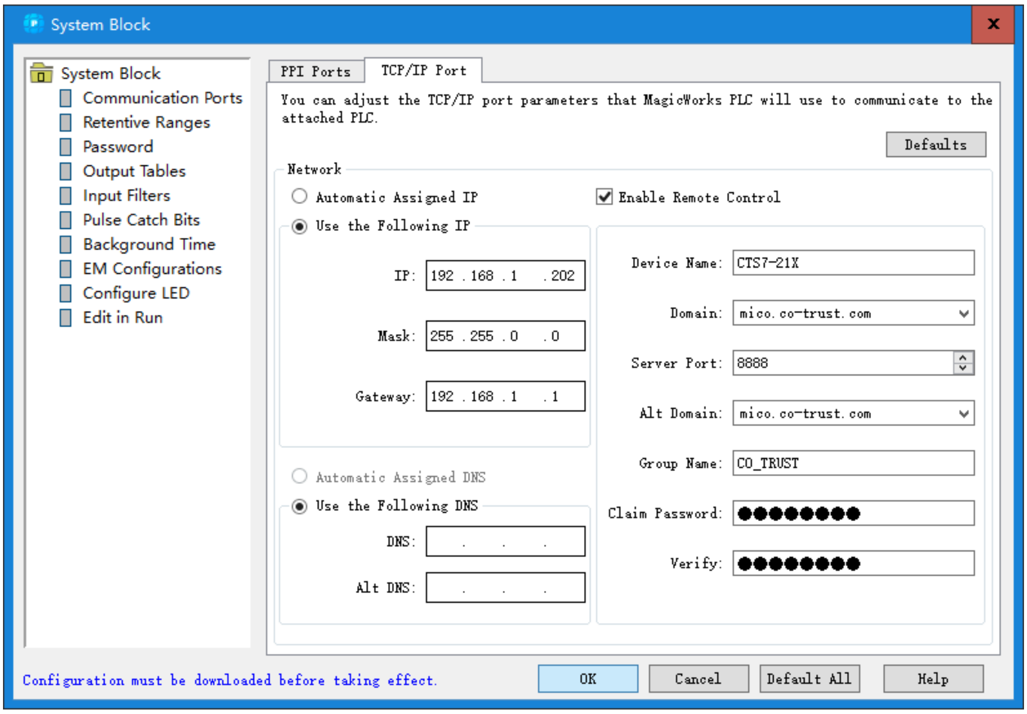
Parameters	Name	IN/OUT	Data Type	Explanation
	EN	IN	BOOL	Enable, edge triggered. The instruction is not allowed to circularly invoke, cause set IP and EPPROM has limits for number of times, and the EtherNET communication would be failed due to frequently write.
	NAME	IN	DWORD	Pointer of device name. The first byte express the length, the whole string contains 32 length bytes at most.

	STATUS	OUT	BYTE	Status word Bit0=1, means set successful. Bit1=1, means length error. Bit2=1, means save failed. Bit3 means illegal character.

Cautions:

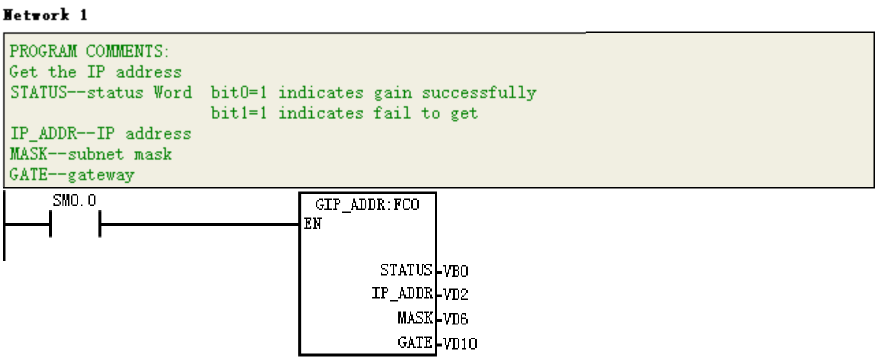
It's available for you to set IP via TD4S except the library. If you use TD4S to set IP, please use static IP in the TCP/IP port of system block, which is not allowed to select automatic acquisition IP address mode, otherwise the setting would not be effect.

If you select library, there is no need to set following system block.



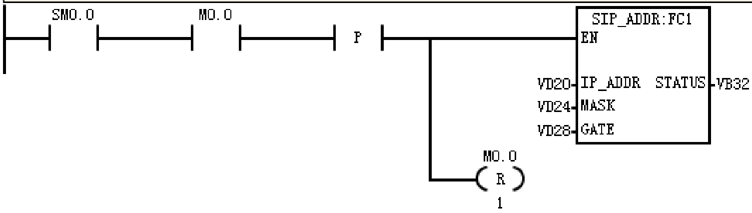
K.3 Example

The sample application of ETHERNET_SET instruction shows as follows:



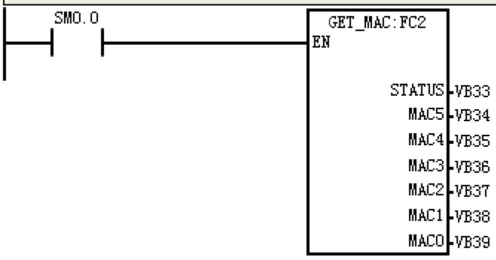
Network 2

```
PROGRAM COMMENTS:
Set IP address
IP_ADDR--IP address
MASK--subnet mask
GATE--gateway
STATUS--status Word  bit0=1 indicates setting completed
                     bit1=1 indicates illegal IP address
                     bit2=1 indicates IP is unmatched with mask
                     bit3=1 indicates IP is unmatched with gateway
                     other bits are not in use
Note:this instruction cannot be called in cycle, for the IP setting has time limit
as EPPROM, with trigger.
```



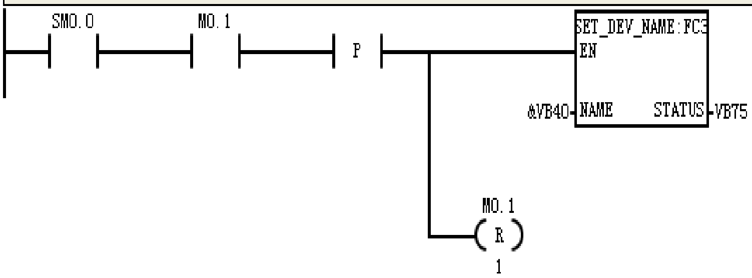
Network 3

```
PROGRAM COMMENTS:
Get MAC address
STATUS--status Word  bit0=1 indicates gain successfully, other bits is not in use
MAC0~5--MAC address
         xx:xx:xx:xx:xx:xx
         MAC5---------MAC0
```



Network 4

```
PROGRAM COMMENTS:
NAME   device name pointer, the pointed first byte of area indicates length. Whole
string including maximum length byte is 32 bytes.
STATUS bit0=1 indicates set successfully
       bit1=1 indicates length error
       bit2=1 indicates fail to save
       bit3 indicates illegal characters
```



L Instruction Set

Table L-1 CTH200 instruction set

Bit logic instruction			
LD	Normally open contact load instruction LD SM0.0	ALD	AND load instruction ALD
A	Normally open contact AND instruction A SM0.0	OLD	OR load instruction OLD
O	Normally open contact OR instruction O SM0.0	LPS	Logic into stack instruction LPS
LDN	Normally close contact load instruction LDN SM0.0	LDS	Load stack instruction LDS 1
AN	Normally close contact AND instruction AN SM0.0	LRD	Logic read instruction LRD
ON	Normally close contact OR instruction ON SM0.0	LPP	Logic out stack instruction LPP
LDI	Normally open contact immediately load instruction LDI I0.0	=	Output instruction = Q0.0
AI	Normally open contact AND immediately instruction AI I0.0	=I	Immediately output instruction =I Q0.0
OI	Normally open contact OR immediately instruction OI I0.0	S	Set instruction S Q0.0 1
LDNI	Normally close contact immediately load instruction LDNI I0.0	SI	Immediately set instruction SI Q0.0 1
ANI	Normally close contact AND immediately instruction ANI I0.0	R	Reset instruction R Q0.0 1
ONI	Normally close contact OR immediately instruction ONI I0.0	RI	Immediately reset instruction RI Q0.0 1
NOT	Logic stack top revere instruction NOT	AENO	Power flow AND instruction AENO

EU	Rising edge detection instruction EU	NOP	Null operation instruction NOP 1
ED	Falling edge detection instruction ED	--	--
Comparison instruction			
LDBx	Byte comparison (=,<,>,<=,>=,<>) load instruction LDB= 1, VB0	OBx	Byte operand comparison (=, <,>,<=,>=,<>) OR instruction OB= 1, VB0
LDWx	Integer comparison (=,<,>,<=,>=,<>) load instruction LDW= 10000, VW0	OWx	Integer comparison (=,<,>,<=, >=,<>) OR instruction OW= 10000, VW0
LDDx	Long integer comparison (=,<,>,<=,>=,<>) load instruction LDD= 100000, VD0	ODx	Double integer comparison (=,<,>,<=,>=,<>) OR instruction OD= 100000, VD0
LDRx	Floating number comparison (=,<,>,<=,>=,<>) load instruction LDR= 1.0, VD0	ORx	Floating number comparison (=, <, >, <=, >=, <>) OR instruction OR= 1.0, VD0
ABx	Byte operand comparison (=,<,>,<=,>=,<>) AND instruction AB= 1, VB0	LDSx	String comparison (=,<>) load instruction LDS="1234567890",VB0
AWx	Integer comparison (=,<,>, <=,>=,<>) AND instruction AW= 10000, VW0	ASx	String comparison (=,<>) AND instruction AS="1234567890", VB0
ADx	Double integer comparison (=,<,>, <=,>=,<>) AND instruction AD= 100000, VD0	OSx	String comparison (=,<>) OR instruction OS="1234567890",VB0
ARx	Floating number comparison (=,<,>, <=,>=,<>) AND instruction AR= 1.0, VD0	--	--
Transmit instruction			
MOVB	Byte move instruction MOVB 1 VB0	SWAP	High and low byte exchange instruction SWAP VW0
MOVW	Word move instruction MOVW 1000,VD0	BIR	Move byte immediately read instruction BIR IB0, VB0

MOVD	Double Word move instruction MOVD 100000,VD0	BIW	Move byte immediately write instruction BIW VB0, QB0
MOVR	Floating number move instruction MOVR 1.0,VD0	SRCP	Recipe to storage card
BMB	Block move byte instruction BMB VB0,VB100, 1	LRCP	Load recipe from storage card
BMW	Block move Word instruction BMW VW0,VW100, 1	SDL	Record data to storage card
BMD	Block move double Word instruction BMD VD0,VD100, 1	--	--
Integer calculate instruction			
+I	Integer addition instruction +I 10000, VW0	MUL	Integer multiply with double Integer MUL 10000, VD0
-I	Integer subtraction instruction -I 10000, VW0	DIV	Integer divide with double Integer DIV VW0, VD0
*I	Integer multiply instruction *I 10000, VW0	INCB	Byte increasing instruction INCB VB0
/I	Integer division instruction /I 10000, VW0	DECB	Byte decreasing instruction DECB VB0
+D	Double Integer addition instruction +D 100000, VD0	INCW	Integer increasing instruction INCW VW0
-D	Double Integer subtraction instruction -D 100000, VD0	DECW	Integer decreasing instruction DECW VW0
*D	Double Integer multiply instruction *D 100000, VD0	INCD	Long Integer increasing instruction INCD VD0
/D	Double Integer division instruction /D 100000, VD0	DECD	Long Integer decreasing instruction DECD VD0
Float-point calculate instruction			
+R	Real addition instruction +R 1.0, VD0	COS	Cosine calculate instruction COS 1.0, VD0
-R	Real addition instruction -R 1.0, VD0	TAN	Tangent calculate instruction TAN 1.0, VD0
*R	Real multiplication instruction *R 1.0, VD0	LN	Nature exponential calculate instruction LN 1.0, VD0
/R	Real division instruction	EXP	Nature exponential calculate

	/R 1.0, VD0		instruction EXP 1.0, VD0
SQRT	Square root instruction	PID	PID circuit calculate instruction PID VB0, 1
SIN	Sine calculate instruction SIN 1.0, VD0	--	--
Convert instruction			
BTI	Convert from byte to Integer BTI 1, VW0	DTR	Convert from Double Integer to real number DTR 100000, VD0
ITB	Convert from Integer to byte ITB 10000, VB0	DTS	Convert from Double Integer to string DTS 100000, VB0, 10
ITD	Convert from Integer to double Integer ITD 10000, VD0	ROUND	Carry integer instruction ROUND 1.0, VD0
ITS	Convert Integer to string ITS 10000, VB0, 10	TRUNC	Cut bit integer instruction TRUNC 1.0, VD0
DTI	Convert from double Integer to Integer DTI 100000, VW0	RTS	Convert from real to string RTS 1.0, VB0, 10
BCDI	Convert from BCD to Integer BCDI VW0	STI	Convert from string to Integer STI"1234567890", 5, VW0
IBCD	Convert from Integer to BCD IBCD VW0	STD	Convert from string to double Integer STD"1234567890", 5, VD0
ITA	Convert from Integer to ASCII ITA 10000,VB0,10	STR	Convert string to real STR"1234567890", 5, VD0
DTA	Convert from double Integer to ASCII DTA 100000, VB0, 10	DECO	Decode instruction DECO 1, VW0
RTA	Convert from real to ASTII instruction	ENCO	Coding instruction ENCO 10000, VB0
ATH	Convert from ASCII to 16 system ATH VB0, VB100, 10	SEG	Segment instruction SEG 1, VB0
HTA	Convert from 16 system to ASCII HTA VB0, VB100, 10	--	--
Real-time clock			
TODR	Read real-time clock TODR VB0	TODRX	Read extend real-time clock TODRX VB0
TODW	Set real-time clock	TODWX	Set extend real-time clock

	TODW VB0		TODWX VB0
Shift cycle instruction			
SLB	Left shift byte SLB VB0, 4	RLW	Left rotation Word RLW VW0, 8
SLW	Left shift Word SLW VW0, 8	RLD	Left rotation double Word RLD VD0, 16
SLD	Left shift double Word SLD VD0, 16	RRB	Right rotation byte RRB VB0, 4
SRB	Right shift byte SRB VB0, 4	RRW	Right rotation Word RRW VW0, 8
SRW	Right shift Word	RRD	Right rotation double Word RRD VD0, 16
SRD	Right shift double Word SRD VD0, 16	SHRB	Shifting register bit instruction SHRB I0.0, V0.0, 8
RLB	Left rotation byte RLB VB0, 4	--	--
Logic calculate instruction			
INVB	Reverse byte instruction INVB VB0	ORB	OR byte instruction ORB 1, VB0
INVW	Reverse Word instruction INVW VW0	ORW	OR Word instruction ORW 10000, VW0
INVD	Reverse double Word instruction INVD VD0	ORD	OR double Word instruction OD 100000, VD0
ANDB	AND byte instruction ANDB 1, VB0	XORB	XOR byte instruction XORB 1, VB0
ANDW	AND Word instruction ANDW 10000, VW0	XORW	XOR Word instruction XORW 10000, VW0
ANDD	AND double Word instruction ANDD 100000, VD0	XORD	XOR double Word instruction XORD 100000, VD0
Counter instruction			
CTU	Count-up instruction CTU C1, 10000	HDEF	HSC definition HDEF 0, 0
CTD	Count-down instruction CTD C1, 10000	HSC	HSC instruction
CTUD	Count down and up instruction CTUD C1, 10000	PLS	Pulse output instruction
Timer instruction			
TON	Open delay timer TON T37, 10000	BITIM	Read start timer BITIM VD0
TONR	Reserved open delay timer TONR T31, 10000	R_UTIM	Read current microsecond value instruction
TOF	Turn off delay timer	CITIM	Calculate interval timer

	TOF T37, 10000		CITIM VD0, VD100
String instruction			
SLEN	Get string length instruction SLEN"1234567890", VB0	SCAT	Concatenation string instruction SCAT"1234567890", VB0
SCPY	Copy string instruction SCPY"1234567890", VB0	SFND	Seek string in string instruction SFND"12345678890", "321", VB0
SSCPY	Copy substring from string instruction SSCPY"1234567890", 1, 10, VB0	CFND	Seek a character in string instruction CFND"12345678890", "a", VB0
Table instruction			
FILL	Memory fill instruction FILL 10000, VW0, 10	FIFO	First in-first out instruction FIFO VW200, VW400
ATT	Add to table instruction ATT 10000, VW0	LIFO	Last in-first out instruction LIFO VW200, VW400
FNDx	Seek table (=, <, >, <=, >=, <>) instruction FND=VW0,9999,VW1000	--	--
Interrupt instruction			
CRETI	Return from interrupt routine instruction	ATCH	Add interrupt instruction
ENI	Start interrupt instruction ENI	DTCH	Separate interrupt instruction
DISI	Forbid interrupt instruction DISI	CEVNT	Clear interrupt event instruction CEVNT 10
Program control instruction			
FOR	FOR-NEXT start loop FOR-NEXT FOR VW0, 1, 10 NEXT	LSCR	Load sequence control interrupt
NEXT	FOR-NEXT loop end	SCRE	Sequence control relay end
JMP	Jump to label instruction	SCRT	Sequence control relay transfer
LBL	Label instruction	CSCRE	Conditional Sequence control relay end
WDR	Monitor reset instruction	CRET	Return from subroutine with condition instruction
DLED	Diagnosis LED instruction DLED 1	END	Conditionally end
CALL	Call subroutine instruction	STOP	Stop instruction
Communication instruction			

XMT	FPORT send instruction	SPA	Set port instruction
RCV	FPORT receive instruction	GPA	Get port address instruction
NETR	Network read instruction	EBUSR	Read module information instruction
NETW	Network write instruction	EBUSW	Write module information instruction
UDPNETR	Ethernet read function	EBUSGETDIA	Get module internal diagnosis information instruction
UDPNETW	Ethernet write function	EBUSSNDCMD	Send module operation command instruction
MBSNDMSG	Send Modbus message	--	--

M Programming Cable

CTH200 PLC programming cable with USB-485 port is used for CPU programing, upload, download and monitor, USB end connect to PC/PG, 485 end connect to CPU PPI port.

Table M-1 Physical features of programming cable

	Item	Contents
	Order No.	CTS7 191-USB30
	Supported OS	Windows2000/XP/7/10
	Baud rate	300bps ~ 1Mbps standard baud rate self-adaption
	Work temperature	0 ~ +55 ℃
	Cable length	2.5 m
	Cable length each PC supported	1

Note
Please install drive before using programming cable, drive download address: www.co-trust.com

Cable appearance:

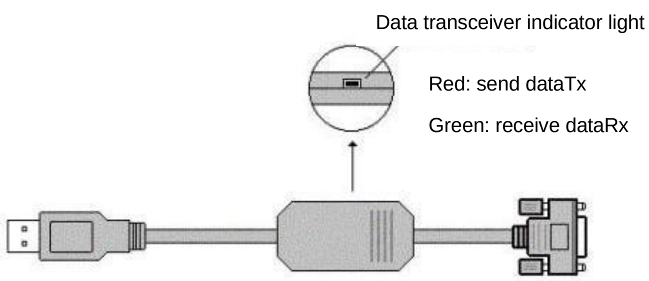


Figure M-1 Programming cable appearance

N How to Use Programming Card

Table N-1 Products specifications

	Specifications	Description
	Title	Programming card
	Order no.	CTS7 291-PC001
	Dimension	(W×H×D) 63 mm ×33 mm × 17 mm

Remarks:

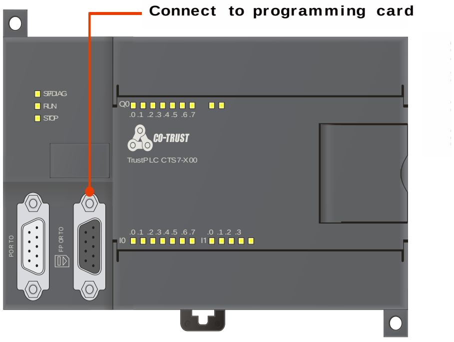
- CTH200 series PLC realize data storage and portable programming by corresponding card.
- CTH200 series PLC do not support third-party memory card.
- H228XL do not support CTSC 291-PC001 V1.0(version 128K)

【Security】

Data in programming card is encrypted to keep away from being maliciously cracked or modified.

【Interface】

Connect the right RS485 port.

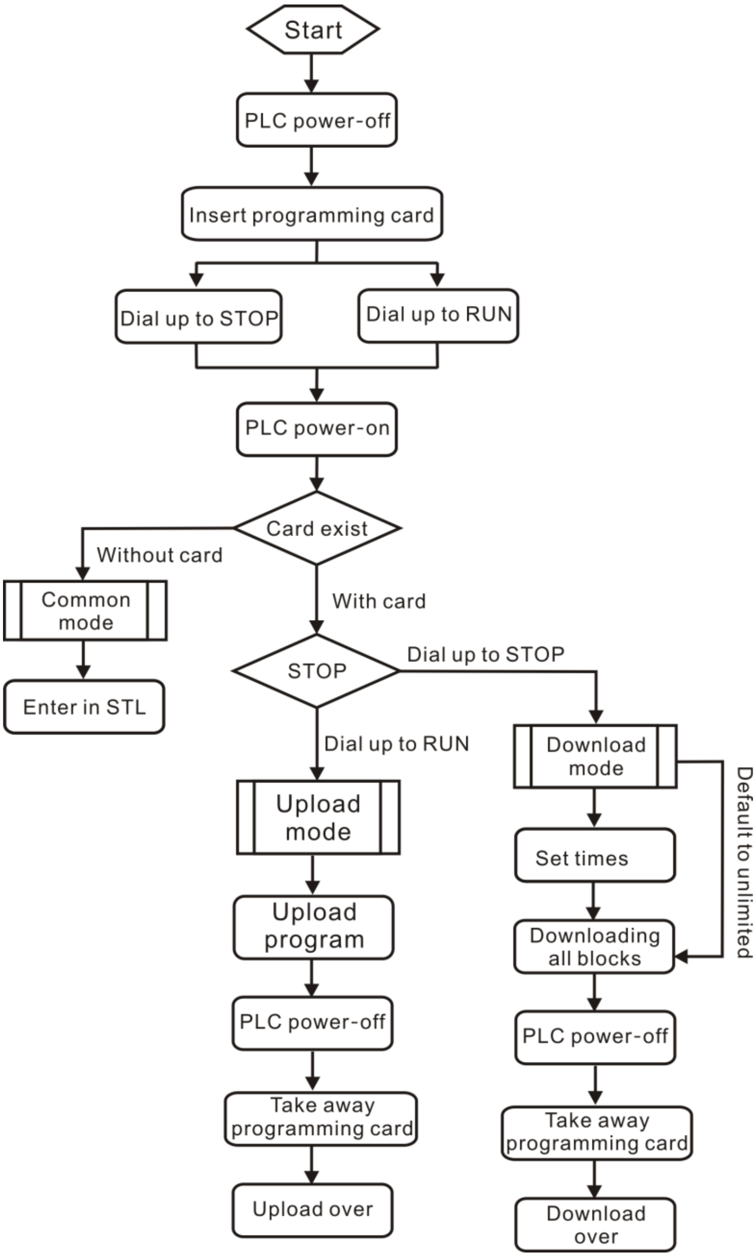


Note

Programming card record CPU’s type while downloading program from it. Uploading program is not allowed if the CPU’s type is inappropriate. There is no need to distinguish relay or transistor for the same type PLC.

Using programming card

【Flow chart】



【Functions】

※ Download

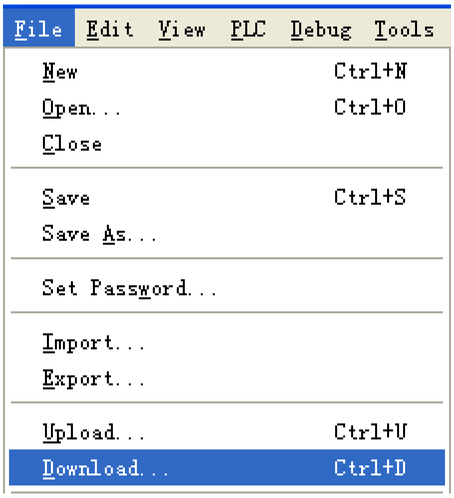
Download the STL programming block and library, system block and data block of microwin project into programming card storage.

Steps:

Step 1: First cut the PLC power off, then insert the programming card to the specific port.

Step 2: Switch the DIP to "STOP", then power on.

Step 3: The system enter "programming card download mode" in default if the card is effective. The orange light always on so just click download. Click icon "" on toolbar or click "file"-> "download" to download the programinto PLC and programming card.



Note: For the block that didn't download to programming card, it will generate a default one which would upload to CPU when needed.

Step 4: If programming card is invalid, the system would enter in common mode.

Step 5: If you need to set the limit use times of programming card, set by modify value of SMB100 before downloading any block for the default is no limit(the value of SMB100 is 0)

Step 6: If you don't want the the programming block or data block of aimed PLC to be covered, set by modify the value of SMB111 for the default is cover all. Refer to table in **upload** for specific definition of SMB111.

Step 7: Download the programming blocks to CPU in turns.

Step 8: Send all the blocks to be downloaded to programming card(the orange light flash during downloading and stays on after all the transmissions).

Step 9: After downloading all the blocks, you can cut the CPU power off when orange light stays on over 10s from flash status, then pull out the programming card. CPU will not work until cut off the power and pull out the programming card if you finish downloading blocks.

Note

- A longer time will need to transmit blocks into programming card if you choose to download under baud rate of 187.5K bps
- CPU will clear all the data of programming card and set limit times of using the card when it confirm the first block has been downloaded.
- If you choose to cover data, set SMB111 before downloading blocks
- For the sake of security, it is recommended to download system block into programming card or it will turn to default as in the new project. Power off in the process of downloading may lead to contents in card lost.

Definition of Indicator lights

Indicator lights	OFF	FLASH	ON
SF	Normal	Error in transmission	---
STOP -orange	---	Data transmitting	Normal
SMB100	Set limit times of programming		
SMB111	Cover function-control byte		

Flash frequency: About 0.5s(accuracy is not required)

※ **Upload**

All the blocks in programming card can be uploaded to CPU to update the EPROM.

Steps:

Steps 1: First cut the PLC power off, then insert the programming card to the specific port.

Steps 2: Switch the DIP to “RUN”, then power on.

Steps 3: The system enter “programming card upload mode” in default if the card is effective. The green light stays on.

Steps 4: The system enter normal mode if the programming card is ineffective.

Steps 5: All the existing blocks would be cleared and then be read into EPROM if CPU detect card.

Note: For the block that didn’t download to programming card, it will generate a default one which would upload to CPU when needed.

Steps 6: RUN green light will flash during the process of uploading. You can cut the PLC power off and pull out the card when green light stays on over 10s from flashing.

Steps 7: Repower the PLC, a new STL program would be executed.

Conditions that may lead to upload failure:

1) The using times for programming card reach limit9(no effect on existing blocks in CPU)

2) Information calibration failure(existing program in CPU may be deleted)

3) Other blocks in programming card calibration failure(blocks without error will be uploaded into CPU)

4) for the CPU with password protection, the uploading will fail so as to keep the security of user program if the programming card haven’t been set as covering blocks(that means not set Bit0 as 0)

The SF red light will flash when upload failure.

<Note >

SMB110 to SMB111 are used for uploading control data, detailed definition as following:

SM address	Function details
SMB110	Enforce to start the upload protection function
SMB111	Programming coverage: 0 – cover program block and data block; 1 –not cover program block, cover data block 2 –cover program block, not cover data block; 3 – not cover program block and data block; 4 –same with 0

Definition of lights status

Lights	OFF	FLASH	ON
RUN light(green)	---	Loading program	Normal
SF light	Normal	Error	---
SMB101	Left times can be used(including this time), display 255 if there is no limit.		

Flash frequency: about 0.5s(accuracy is not required)

※ **Limit of using times for programming card**

You can set the limit of using times for programming card, which minus one every time the uploading complete.

Steps:

Steps 1: When PLC is under “programming card download mode”, set limit of using times by using SM100 and display at SMB101 position before download first block in MicroWIN

Steps 2: The limit of using times for programming card minus one every time the uploading complete(including the third time of uploading failure).

Parameter definition

Value	Explanation
0	No limit
1-255	Times available
Default	0

O Battery Card and Memory Card

CTH200 PLC equip with specific battery card, and the high-performance series CPU (H224X/H226XL/H228XL) support memory card.

Table O-1 Order info

Type	Description	Order No.
Memory card	64K(user program, recipe and data record)	CTS7 291-MC064
	256K(user program, recipe and data record)	CTS7 291-MC256
Battery card	Data-hold time: 200d, typical	CTH2 291-8BA33

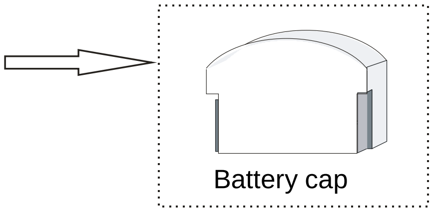
Steps to change battery card or memory card:

1) Unplug the battery cover in power cut off.

2) Take down the existing battery card in vertical direction to avoid damaging slot.

3) Insert the new battery card into slot and cover the cap.

Backup the data before cut the PLC power off to change the battery.



Caution

1. Please change battery card in same specification. Be careful in case to damage other components. And save data before changing.

2. The use method of COTRUST memory card is same as SIMENS'. Use it to store user program, recipe and data record, but for CTH200 the contents in storage card will not copy to PLC automatically after restart S

P Special Function Register

SMB0: System status bit

The 8 bits of SMB0 play a special role in the system. At the end of each scan cycle, the PLC will update the 8 bits. Users can use the special functions of these bits in the program to achieve some common operations, the details shows as following table:

SMB0 bit	Function description(read only)
SM0.0	The bit is always on and can be used as enable bit if you want a section of a program to execute every cycle.
SM0.1	This bit is only 1 at first scan and can be used as the enable bit to initialize segment.
SM0.2	If the power-off hold data is lost, the bit opens a scan cycle. This bit can be used as an error memory bit or to activate enable bits in a particular startup order.
SM0.3	When the power is turned on into RUN mode, the bit is switched on for a scan cycle. And it can provide machine warm-up time before operation.
SM0.4	This bit offers a square wave clock pulse period 1 min, 30s on and 30s off.
SM0.5	This bit offers a square wave clock pulse period 1sec, 0.5s on and 0.5s off.
SM0.6	The bit is the scanning clock, set 1 for this scan and 0 for the next, which also can be used as enable bit of scan counter.
SM0.7	This bit indicates PLC switch (0: TERM, 1: RUN) position, be used as the enable bit of the FPORT communication program segment, when the switch is at the RUN position, the FPORT communication mode is effective, PPI communication is effective when switch to TERM.

SMB1: Instruction execution status bit

The 8 bits of SMB1 are used to indicate errors or special states in the execution of certain instructions, as detailed in the following table:

SMB1 bit	Function description(read only)
SM1.0	Set 1 when instruction executed as 0
SM1.1	Set 1 when instruction execution results overflow or detect illegal value
SM1.2	Set 1 when instruction execution results is negative
SM1.3	Set 1 when the divisor in the instruction is 0
SM1.4	Set 1 if the appended table length exceeds the table range when executing ATT instructions
SM1.5	Set 1 if the table is empty when executing a LIFO or FIFO instruction,
SM1.6	Set 1 when trying to convert a non-BCD number to a binary number
SM1.7	Set 1 When ASCII code cannot be converted to a valid hexadecimal number

SMB2: FPORT receive character

SMB2 is the FPORT receive character buffer, every character received is put here under FPORT communication mode for program easy to read. Detailed description shows as following table:

SM2 bit	Function description(read only)
SMB2	This byte includes every character of port 0 or 1 in FPORT communication

SMB3: FPORT odd-even parity error

SMB3 is used to display FPORT parity errors. In FPORT communication mode, set SM3.0 1 when detecting checksum error in the received character. Normally, use this bit to detect FPORT information characters for transmission errors before the program receives and reads the information character values stored in SMB2. Detailed description is shown in the following table:

SMB3 bit	Function description(read only)
SM3.0	This bit indicates if there is parity error in port 0 and 1 0: No error, 1: Error
SM3.1~SM3.7	Reserved

SMB4: Queue overflow

SMB4 contains interrupt queue overflow bits, whether interrupts allow flag bits, send idle bits, and whether mandatory flag bits exist. Queue overflow indicates either that interrupts occur more frequently than the CPU, or it has been disabled by the global interrupt forbidden instruction. Detailed description is shown in the following table:

SMB4 bit	Function description(read only)
SM4.0	Turned on when the communication interrupt queue overflows. Reset when the queue is empty and control returns to the main program, use only in interrupt routine
SM4.1	Turned on when the input interrupt queue overflows. Reset when the queue is empty and control returns to the main program, use only in interrupt routine
SM4.2	Turned on when the timing interrupt queue overflows. Reset when the queue is empty and control returns to the main program, use only in interrupt routine
SM4.3	Turn on when detecting run time program error
SM4.4	This bit reflects the global interrupt enabled state, turn on when interrupts are enabled.
SM4.5	Turn on when port 0 transmitter in idle
SM4.6	Turn on when port 1 transmitter in idle
SM4.7	Turn on when any memory position is forced

SMB5: I/O status

SMB5 contains the I/O error status bits in the system. Which provide information about the I/O errors, as detailed in the following table:

SMB5 bit	Function description(read only)
SM5.0	Set 1 when I/O error
SM5.1	Set 1 when I/O bus connected too many digital I/O
SM5.2	Set 1 when I/O bus connected too many analog I/O
SM5.3	Set 1 when I/O bus connected too many intelligent I/O modules
SM5.4~SM5.7	Reserved

SMB6: CPU identification register

SMB6 is CPU identification (ID) register, use for identify CPU type, as detailed in the following table:

SM address	Function description(read only)
SMB6	CPU ID register
SM6.0~SM6.3	Reserved
SM6.4~SM6.7	Use for identifying CPU type 0000: CPU212, CPU222; 0010: CPU214, CPU224; 0110: CPU221; 1000: CPU215; 1001: CPU216, CPU226, CPU226XM; 1110: H35-00;

SMB6 format:

MSB LSB

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
x	x	x	x	r	r	r	r
0000: CPU212, CPU222; 0010: CPU214, CPU224; 0110: CPU221; 1000: CPU215; 1001: CPU216, CPU226, CPU226XM; 1110: H35-00				Reserved			

SMB8 to SMB21: I/O module and error register

SMB8 to SMB21 are prepared in byte pair form (two adjacent bytes) for extension modules 0 to 6. The even-bit byte of each pair of bytes is the module identification register, and the odd-bit byte is the module error register. The former marks the module type, I/O type, input and output points; while the latter is the I/O error detected against corresponding module. Detailed description is shown in the following table:

SM address	Function description(read only)
SMB8	Module 0 identification register
SMB9	Module 0 error register
SMB10	Module 1 identification register
SMB11	Module 1 error register
SMB12	Module 2 identification register
SMB13	Module 2 error register
SMB14	Module 3 identification register
SMB15	Module 3 error register
SMB16	Module 4 identification register
SMB17	Module 4 error register
SMB18	Module 5 identification register
SMB19	Module 5 error register
SMB20	Module 6 identification register
SMB21	Module 6 error register

Even byte (module ID register) format:

MSB LSB

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
m	t	t	a	i	i	q	q
m: If there is module 0—Module 1—No module	tt: Module type 00—Non-intelligent I/O module 01—Intelligent module 10—Reserved 11-- Reserved		a: I/O type 0—Digital 1—Analog	ii: Inputs 00—No input 01--2 AI or 8 DI 10--4 AI or 16 DI 11--8 AI or 32 DI		qq: Outputs 00—No output 01--2 AQ or 8 DQ 10--4 AQ or 16 DQ 11--8 AQ or 32 DQ	

Odd byte (module error register) format:

MSB LSB

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
c	0	0	b	r	p	f	t
c: Configuration error 0—No error 1-- Error			b: Bus or parity error 0-- No error 1-- Error	r: Over range error 0-- No error 1-- Error	p: User power error 0-- No error 1-- Error	f: Fuse error 0-- No error 1-- Error	t: Terminal block loose error 0-- No error 1-- Error

SMW22 to SMW26: Scanning time

SMW22, SMW24 and SMW26 offer information of scanning time: minimum scanning time, maximum scanning time and last scanning time calculate by ms

SM address	Function description(read only)
SMW22	Last scanning time
SMW24	Recorded minimum scanning time after entering RUN mode
SMW26	Recorded maximum scanning time after entering RUN mode

SMB28 to SMB29: Analog potentiometer

Special memory bytes 28 and 29 contain values corresponding to analog potentiometer 0 and 1 axes position. The analog potentiometer is located in front of the CPU behind the access gate. Use a screwdriver to adjust the potentiometer (increasing clockwise or decreasing counterclockwise). This type of read-only value can be used by a program to perform various functions, such as updating the current value for a timer or counter, inputting or changing a preset value, or setting limits. It also has a rated range of 0 to 255, and reliable repeats of ±2 counts. Detailed description is shown in the following table:

SM address	Function description(read only)
SMB28	Input of memory analog regulator 0. Updating it for every scan in STOP/RUN mode
SMB29	Input of memory analog regulator 1. Updating it for every scan in STOP/RUN mode

SMB30 and SMB130: FPORT control register

SMB30 controls FPORT 0 communication mode, and SMB130 controls FPORT 1 communication mode. You can write and read between SMB30 and SMB130 which can be used to set operation mode of FPORT communication and provide the choice between the FPORT and the protocols supported by the system. See the following table for detailed description:

SM address	Function description
SMB30	FPORT 0 control register
SMB130	FPORT 1 control register
SM30.0~SM30.1	Port 0 communication protocol selection
SM130.0~SM130.1	Port 1 communication protocol selection
SM30.2~SM30.4	FPORT 0 baud rate
SM130.2~SM130.4	FPORT 1 baud rate
SM30.5	Each character data bit of FPORT 0
SM130.5	Each character data bit of FPORT 1
SM30.6~SM30.7	FPORT 0 parity selection
SM130.6~SM130.7	FPORT 1 parity selection

FPORT control byte format:

MSB LSB

<table>
<tr><th>Bit 7</th><th>Bit 6</th><th>Bit 5</th><th>Bit 4</th><th>Bit 3</th><th>Bit 2</th><th>Bit 1</th><th>Bit 0</th></tr>
<tr><td>p</td><td>p</td><td>d</td><td>b</td><td>b</td><td>b</td><td>m</td><td>m</td></tr>
<tr><td colspan="2">Pp: Parity selection
00—No parity
01—Even parity
10-- No parity
11—Odd parity</td><td>d: Each character data bit
0--8 bits data bit
1--7 bits data bit</td><td colspan="3">bbb: FPORT baud rate
000--38400 bps
001--19200 bps
010--9600 bps
011--4800 bps
100--2400 bps
101--1200 bps
110--115200 bps
111--57600 bps</td><td colspan="2">Mm: Protocol setting
00—Point-point port protocol(PPI/slave station mode)
01—FPORT protocol
10--PPI/ master station mode
11—Reserved(default PPI/slave station mode)</td></tr>
</table>

<Note> When select mm = 10(PPI master station), PLC will become a network master station, able to execute NETR and NETW instruction. Ignore 2 to 7 bit in PPI mode.

SMB31 and SMW32: Permanent storage (EEPROM) write control

With control of the user program, you can store the data of V storage into permanent storage, also known as non-volatile storage. First, store the address into SMW32, then store the saved command into SMB31. Once you issue the storage command, you cannot change the value of V storage until the CPU completes the storage operation and SM31.7 is set to 0.

At the end of each scan cycle, the CPU checks for a command to store data in the permanent storage area. If so, store data into permanent storage. Detailed description is shown in the following table:

SM address	Function description
SMB31	Permanent storage command register
SM31.0~SM31.1	Data size to be saved 00 =Byte; 01 =Byte; 10 =Word; 11 =Double Word
SM31.2~SM31.6	Preserved, always 00000
SM31.7	Store into permanent storage, S7-200 reset the bit after each storage operation. 0 = No request of executing storage 1 = User program applies to permanent storage for data storage
SMW32	The V memory address of stored data, which is offset from V0. When executing a storage command, store data into the corresponding location in permanent storage.

SMB34 and SMB35: Timing interrupt interval register

SMB34 and SMB35 define the interval between timed interrupts 0 and 1. You can specify the time interval (incremented by 1ms) from 1ms to 255ms. If the corresponding timed interrupt event is connected to an interrupt service routine, the CTH200CPU gets the interval value. To change the interval, you must either reallocate timed interrupt events to the same or another interrupt program, or terminate timed interrupt events by interrupt detach.

SM address	Function description
SMB34	Time interval value of timing interrupt 0
SMB35	Time interval value of timing interrupt 1

SMB36 to SMB65: HSC0, HSC1 and HSC2 register

SMB36 to SMB65 are used to monitor and control operation of HSC0, HSC1 and HSC2. Detailed description is shown in the following table:

SM address	Function description
HSC0	
SMB36	HSC0 counter status
SM36.0~SM36.4	Reserved
SM36.5	HSC0 current count direction bit: 1=addition count
SM36.6	HSC0 current value equal to preset value bit: 1=equal
SM36.7	HSC0 current value more than preset value bit: 1=more than
SMB37	HSC0 control byte
SM37.0	Reset valid control bit: 0= high level reset valid, 1=low level reset valid
SM37.1	Reserved
SM37.2	Count speed selection of orthography counter: 0=4x count speed; 1=1x count speed
SM37.3	HSC0 direction control bit: 1= addition count
SM37.4	HSC0 update direction: 1= update direction
SM37.5	HSC0 update preset value: 1=write new preset value to HSC0
SM37.6	HSC0 update current value: 1=write new initial value to HSC0
SM37.7	HSC0 valid bit: 1=valid

SMD38	HSC0 new initial value, set SM37.6 as 1 after write new value into SMD38, then the current value will be updated
SMD42	HSC0 new preset value, set SM37.5 as 1 after write new value into SMD42, then the current value will be updated
HSC1	
SMB46	HSC1 counter status
SM46.0~SM46.4	Reserved
SM46.5	HSC1 current count direction bit: 1=addition count
SM46.6	HSC1 current value equal to preset value bit: 1=equal
SM46.7	HSC1 current value more than preset value bit: 1=more than
SMB47	HSC1 control byte
SM47.0	Reset valid control bit: 0= high level reset valid, 1=low level reset valid
SM47.1	HSC1 start valid level control bit: 0= high level, 1=low level
SM47.2	Count speed selection of orthography counter: 0=4x count speed; 1=1x count speed
SM47.3	HSC1 direction control bit: 1= addition count
SM47.4	HSC1 update direction: 1= update direction
SM47.5	HSC1 update preset value: 1=write new preset value to HSC0
SM47.6	HSC1 update current value: 1=write new initial value to HSC0
SM47.7	HSC1 valid bit: 1=valid
SMD48	HSC1 new initial value, set SM47.6 as 1 after write new value into SMD48, then the current value will be updated
SMD52	HSC1 new preset value, set SM47.5 as 1 after write new value into SMD52, then the current value will be updated
HSC2	
SMB56	HSC2 counter status
SM56.0~SM56.4	Reserved
SM56.5	HSC2 current count direction bit: 1=addition count
SM56.6	HSC2 current value equal to preset value bit: 1=equal
SM56.7	HSC2 current value more than preset value bit: 1=more than
SMB57	HSC2 control byte
SM57.0	Reset valid control bit: 0= high level reset valid, 1=low level reset valid
SM57.1	HSC1 start valid level control bit: 0= high level, 1=low level
SM57.2	Count speed selection of orthography counter: 0=4x count speed; 1=1x count speed
SM57.3	HSC2 direction control bit: 1= addition count
SM57.4	HSC2 update direction: 1= update direction
SM57.5	HSC2 update preset value: 1=write new preset value to HSC0
SM57.6	HSC2 update current value: 1=write new initial value to HSC0
SM57.7	HSC2 valid bit: 1=valid
SMD58	HSC2 new initial value, set SM57.6 as 1 after write new value into SMD58, then the current value will be updated
SMD62	HSC2 new preset value, set SM57.5 as 1 after write new value into SMD62, then the current value will be updated

<Note>

1）The counter status bit is only valid in interrupt program triggered by HSC event.

2）Do not reset new current value in program when using the HSC external reset interrupt event, and re-enable the counter in the interrupt program connected to the event, which will result serious errors.

SMB66 to SMB85: PTO/PWM register

SMB66 to SMB85 are used to monitor and control the pulse chain output and PWM functions of PLC (pulse) commands. Detailed description is shown in the following table:

SM address	Function description
PTO0	
SMB66	PTO0 status byte
SM66.0~SM66.3	Reserved
SM66.4	PTO0 envelope termination: 0=no error, 1=terminate for incremental error
SM66.5	PTO0 envelope termination: 0=terminate not by user command; 1=terminate by user command
SM66.6	PTO0 pipe overflow(clear by system when using external envelope, or by user program): 0=no overflow, 1=overflow
SM66.7	PTO0 free bit: 0=PTO busy, 1=PTO free
SMB67	Control pulse string output and PWM of Q0.0
SM67.0	PTO0/PWM0 update cycle: 1=write new cycle
SM67.1	PWM0 update pulse width: 1=write new pulse width
SM67.2	PTO0 update pulse quantity: 1=write new pulse quantity
SM67.3	PTO0/PWM0 reference time unit: 0=1 μs/grid, 1=1 ms/grid
SM67.4	Synchronous update PWM0: 0=asynchronous update, 1=synchronous update
SM67.5	PTO0 operation: 0= single segment operation(cycle and pulse store in SM storage), 1=multiple segment operation(envelope table involved in V storage area)
SM67.6	PTO0/PWM0 mode selection: 0=PTO, 1=PWM
SM67.7	PTO0/PWM0 valid bit: 1=valid
SMW68	PTO0/PWM0 cycle(2~65, 535 time standard)
SMW70	PWM0 pulse width (0~65, 535 time standard)
SMD72	PTO0 pulse count (1~232 --1)
PTO1	
SMB76	PTO1 status byte
SM76.0~SM76.3	Reserved
SM76.4	PTO1 envelope termination: 0=no error, 1=terminate for incremental error
SM76.5	PTO1 envelope termination: 0=terminate not by user command; 1=terminate by user command
SM76.6	PTO1 pipe overflow(clear by system when using external envelope, or by user program): 0=no overflow, 1=overflow

SM76.7	PTO1 free bit: 0=PTO busy, 1=PTO free
SMB77	Control pulse string output and PWM of Q0.1
SM77.0	PTO1/PWM1 update cycle: 1=write new cycle
SM77.1	PWM1 update pulse width: 1=write new pulse width
SM77.2	PTO1 update pulse quantity: 1=write new pulse quantity
SM77.3	PTO1/PWM1 reference time unit: 0=1 μs/grid, 1=1 ms/grid
SM77.4	Synchronous update PWM1: 0=asynchronous update, 1=synchronous update
SM77.5	PTO1 operation: 0= single segment operation(cycle and pulse store in SM storage), 1=multiple segment operation(envelope table involved in V storage area)
SM77.6	PTO1/PWM1 mode selection: 0=PTO, 1=PWM
SM77.7	PTO1/PWM1 valid bit: 1=valid
SMW78	PTO1/PWM1 cycle(2~65, 535 time standard)
SMW80	PWM1 pulse width (0~65, 535 time standard)
SMD82	PTO1 pulse count (1~232 --1)

SMB86 to SMB94 and SMB186 to SMB194: receiving information control

SMB86 to SMB94 and SMB186 to SMB194 are used to control and read status about receiving information. Detailed description is shown in the following table:

SM address	Function description
P0 received information	
SMB86	P0 receive information status byte
SMB87	P0 receive information control byte
SMB88	P0 start of information character
SMB89	P0 end of information character
SMW90	The P0 free row interval is given in ms. The first character received at the end of free line time is the beginning of new information.
SMW92	P0 inter-character/inter-message timer timeout value (represented by ms). If the timeout, stop receiving message.
SMB94	Maximum characters P0 received(1~255 byte) <Note> This area must set as maximum buffer you want, even if character count information is not used to terminate it.
P0 received information	
SMB186	P1 receive information status byte
SMB187	P1 receive information control byte
SMB188	P1 start of information character
SMB189	P1 end of information character
SMW190	The P1 free row interval is given in ms. The first character received at the end of free line time is the beginning of new information.
SMW192	P1 inter-character/inter-message timer timeout value (represented by ms). If the timeout, stop receiving message.
SMB194	Maximum characters P1 received(1~255 byte) <Note> This area must set as maximum buffer you want, even if character count information is not used to terminate it.

Receiving information status byte format:

MSB LSB

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n	r	E	0	0	t	c	p

n = 1: Receive user order to stop receiving information

r = 1: Input parameter error or no start or end condition result in information receiving end

e = 1: Receive end character

t = 1: Receive information end: timeout

c = 1: Receive information end: over maximum character

p = 1: Receive information end: parity error

Receiving information control byte format:

MSB LSB

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
en	sc	ec	il	c/m	tmr	bk	0

en: 0 = Forbid receiving information function

1 = Allow receiving information function

Allow/forbid receive information bit each time execute RCV instruction

sc: 0 = Ignore SMB88 or SMB188

1 = Use value of SMB88 or SMB188 to detect start information

ec: 0 = Ignore SMB89 or SMB189

1 = Use value of SMB89 or SMB189 to detect start information

il: 0 = Ignore SMW90 or SMW190

1 = Use value of SMW90 or SMW190 to detect free status

c/m: 0 =Internal character timer

1 = Inform timer

tmr: 0 = Ignore SMW92 or SMW192

1 = Terminate receiving when SMW92 or SMW192 timing timeout.

bk: 0 = Ignore interrupt condition

1 = Use interrupt condition as start of information detection

SMW98: Expansion I/O bus error

SMW98 offer information about expansion I/O bus error. Detailed description is shown in the following table:

SM address	Function description
SMW98	Each time a checksum error occurs on the extended bus, it increases by 1. When the system power on or the user program writes zero, it can be reset.

SMB110 to SMB111: programming card upload control

SMB110 to SMB111 Use for programming card load control:

SM address	Function description
SMB110	Start enforcing load protection function
SMB111	Programming card cover item: 0 – Cover programming block and data block; 1 – Not cover programming block, cover data block 2 - Cover programming block, not cover data block; 3 - Not cover programming block and data block 4 – Same when ADD is 0

SMB114 to SMB115: expansion board information

SMB114: Expansion board module type

SMB114 = 0x20: CAN

0x19: 2AI/1AQ

0x1E: 4AI/2AQ

Detailed description is shown in the following table:

<table>
<tr><th>Bti7</th><th>Bti6</th><th>Bti5</th><th>Bti4</th><th>Bti3</th><th>Bti2</th><th>Bti1</th><th>Bti0</th></tr>
<tr><td></td><td>E</td><td colspan="2">TT</td><td>t</td><td colspan="2">II</td><td>OO</td></tr>
<tr><td>E
If there is module</td><td colspan="7">1: Module do not exist
0: Module exist</td></tr>
<tr><td rowspan="4">TT
(module type)</td><td>00</td><td colspan="6">Non intelligent module</td></tr>
<tr><td>01</td><td colspan="6">Intelligent module</td></tr>
<tr><td>10</td><td colspan="6">Special non intelligent module(7PF)</td></tr>
<tr><td>11</td><td colspan="6">cotrust intelligent module(PID)</td></tr>
<tr><td rowspan="4">t
(module subtype)</td><td rowspan="2">Non intelligent module</td><td colspan="6">0: Digital</td></tr>
<tr><td colspan="6">1: Analog</td></tr>
<tr><td rowspan="2">Intelligent module</td><td colspan="6">0: Normal intelligent</td></tr>
<tr><td colspan="6">1: Cotrust intelligent module(PID)</td></tr>
<tr><td rowspan="5">II(inputs)</td><td rowspan="4">Non intelligent module</td><td>00</td><td colspan="5">No input</td></tr>
<tr><td>01</td><td colspan="5">AI/8DI</td></tr>
<tr><td>10</td><td colspan="5">2AI/16DI</td></tr>
<tr><td>11</td><td colspan="5">4AI/32DI</td></tr>
<tr><td>Intelligent module</td><td colspan="6">Reserved</td></tr>
<tr><td rowspan="5">OO
(outputs)</td><td rowspan="4">Non intelligent module</td><td>00</td><td colspan="5">No output</td></tr>
<tr><td>01</td><td colspan="5">1AO/8DO</td></tr>
<tr><td>10</td><td colspan="5">2AO/8DO</td></tr>
<tr><td>11</td><td colspan="5">4AO/32DO</td></tr>
<tr><td>Intelligent module</td><td colspan="6">Reserved</td></tr>
</table>

SMB115: Expansion board status

SMB115 = 0x0: Module no error

0x255: Expansion board visit error

Others: Expansion board internal diagnosis

SMW116 to SMW126: Expansion board analog mapping area

Corresponding relation of mapping area:

Expansion board	Analog mapping area
2AI/1AQ	SMW116 corresponds to AIW0; SMW118 corresponds to AIW2; SMW124 corresponds to AQW0
4AI/2AQ	SMW116 corresponds to AIW0; SMW118 corresponds to AIW2; SMW120 corresponds to AIW4 SMW122 corresponds to AIW6; SMW124 corresponds to AQW0; SMW126 corresponds to AQW2

SMB131 CAN expansion board access cycle setting

CAN expansion board visit cycle (ms), default 1ms when set as 0

SMB136 to SMB165: HSC3, HSC4 and HSC5 register

SMB136 to SMB165 use for monitor and control HSC3, HSC4 and HSC5operation. Detailed description is shown in the following table:

SM address	Function description
HSC3	
SMB136	HSC3 counter status
SM136.0~4	Reserved
SM136.5	HSC3 current count direction bit: 1=addition count
SM136.6	HSC3 current value equal to preset value bit: 1=equal
SM136.7	HSC3 current value more than preset value bit: 1=more than
SMB137	HSC3 control byte
SM137.0~2	Reserved
SM137.3	HSC3 direction control bit: 1= addition count
SM137.4	HSC3 update direction: 1= update direction
SM137.5	HSC3 update preset value: 1=write new preset value to HSC3
SM137.6	HSC3 update current value: 1=write new initial value to HSC3
SM137.7	HSC3 valid bit: 1=valid
SMD138	HSC3 new initial value, set SM137.6 as 1 after write new value into SMD138, then the current value will be updated
SMD142	HSC3 new preset value, set SM137.5 as 1 after write new value into SMD142, then the current value will be updated
HSC4	
SMB146	HSC4 counter status
SM146.0~4	Reserved
SM146.5	HSC4 current count direction bit: 1=addition count
SM146.6	HSC4 current value equal to preset value bit: 1=equal

SM146.7	HSC4 current value more than preset value bit: 1=more than
SMB147	HSC4 control byte
SM147.0	Reset valid control bit: 0= high level reset valid, 1=low level reset valid
SM147.1	Reserved
SM147.2	Count speed selection of orthography counter: 0=4x count speed; 1=1x count speed
SM147.3	HSC4 direction control bit: 1= addition count
SM147.4	HSC4 update direction: 1= update direction
SM147.5	HSC4 update preset value: 1=write new preset value to HSC4
SM147.6	HSC4 update current value: 1=write new initial value to HSC4
SM147.7	HSC4 valid bit: 1=valid
SMD148	HSC4 new initial value, set SM147.6 as 1 after write new value into SMD148, then the current value will be updated
SMD152	HSC4 new preset value, set SM147.5 as 1 after write new value into SMD152, then the current value will be updated
HSC5	
SMB156	HSC5 counter status
SM156.0~4	Reserved
SM156.5	HSC5 current count direction bit: 1=addition count
SM156.6	HSC5 current value equal to preset value bit: 1=equal
SM156.7	HSC5 current value more than preset value bit: 1=more than
SMB157	HSC5 control byte
SM157.0~2	Reserved
SM157.3	HSC5 direction control bit: 1= addition count
SM157.4	HSC5 update direction: 1= update direction
SM157.5	HSC5 update preset value: 1=write new preset value to HSC5
SM157.6	HSC5 update current value: 1=write new initial value to HSC5
SM157.7	HSC5 valid bit: 1=valid
SMD158	HSC5 new initial value, set SM157.6 as 1 after write new value into SMD168, then the current value will be updated
SMD162	HSC5 new preset value, set SM157.5 as 1 after write new value into SMD162, then the current value will be updated

<Note>

1 The counter status bit is only valid in the interrupt program triggered by HSC event.

2 When you use the HSC external reset interrupt event, do not reset the new current value in the program, and then re-enable the counter in the interrupt program connected to the event, this operation will cause serious errors.

SM address	Function description
PTO0	
SMB66	PTO0 status byte
SM66.0~SM66.3	Reserved

SM66.4	PTO0 envelope termination: 0=no error, 1=terminate for incremental error
SM66.5	PTO0 envelope termination: 0=terminate not by user command; 1=terminate by user command
SM66.6	PTO0 pipe overflow(clear by system when using external envelope, or by user program): 0=no overflow, 1=overflow
SM66.7	PTO0 free bit: 0=PTO busy, 1=PTO free
SMB67	Control pulse string output and PWM of Q0.0
SM67.0	PTO0/PWM0 update cycle: 1=write new cycle
SM67.1	PWM0 update pulse width: 1=write new pulse width
SM67.2	PTO0 update pulse quantity: 1=write new pulse quantity
SM67.3	PTO0/PWM0 reference time unit: 0=1 μs/grid, 1=1 ms/grid
SM67.4	Synchronous update PWM0: 0=asynchronous update, 1=synchronous update
SM67.5	PTO0 operation: 0= single segment operation(cycle and pulse store in SM storage), 1=multiple segment operation(envelope table involved in V storage area)
SM67.6	PTO0/PWM0 mode selection: 0=PTO, 1=PWM
SM67.7	PTO0/PWM0 valid bit: 1=valid
SMW68	PTO0/PWM0 cycle(2~65, 535 time standard)
SMW70	PWM0 pulse width (0~65, 535 time standard)
SMD72	PTO0 pulse count (1~232 --1)

SMB166 to SMB185: PTO0, PTO1 envelope definition table

SMB166 to SMB185 use for display quantity of envelope step and envelope table, also the V storage table address (H224/H226L not support; H224X/H226XL/H228XL support) Detailed description is shown in the following table:

SM address	Function description
PTO0	
SMB166	PTO0 envelope step current count value
SMB167	Reserved
SMW168	PTO0 envelope table V storage address(offset fromV0)
SMB170	Linear PTO0 status byte
SMB171	Linear PTO0 result byte
SMD172	Specifies the frequency of the linear PTO0 generator generates when operating in manual mode. The frequency is a bit integer in Hz.
PTO1	
SMB176	PTO1 envelope step current count value
SMB177	Reserved
SMW178	PTO1 envelope table V storage address(offset fromV0)
SMB180	Linear PTO1 status byte
SMB181	Linear PTO1 result byte
SMD182	Specifies the frequency of the linear PTO0 generator generates when operating in manual mode. The frequency is a bit integer in Hz.

SM195: Communication status

SM195.6	SM195.6=1 after CPU connect to remote server, or set as 0
SM195.7	Control whether heartbeat packets are sent in TCP_Modbus communication over time. SM195.7=1, heartbeat packet sent out after communication timeout; Otherwise do not send; Default is 0.

SMB200 to SMB549: Intelligent module status

SMB200 to SMB549 reserve storage intelligent expansion module information, detailed description is shown in the following table:

0 slot intelligent module	1 slot intelligent module	2 slot intelligent module	3 slot intelligent module	4 slot intelligent module	5 slot intelligent module	6 slot intelligent module	Description
SMB200 ~215	SMB250 ~265	SMB300 ~315	SMB350 ~365	SMB400 ~415	SMB450 ~465	SMB500 ~515	Module name(16 ASCII characters)
SMB216 ~219	SMB266 ~269	SMB316 ~319	SMB366 ~369	SMB416 ~419	SMB466 ~469	SMB516 ~519	S/W revise number (4 ASCII characters)
SMW220	SMW270	SMW320	SMW370	SMW420	SMW470	SMW520	Error code
SMB222 ~249	SMB272 ~299	SMB322 ~349	SMB372 ~399	SMB422 ~449	SMB472 ~499	SMB522 ~549	Information related to specific module

Q Product Oder Info.

Table P-1 Products Oder info

Specifications	Order No.
CPU	
CPU H224 12KB program space/8KB data space, 24VDC, 14DI/10DO transistor source outputs, 0.5A, one PPI, one Freeport, one EtherCAT port, 3*50KHz motion control output	CTH2 214-1AD33-0X24
CPU H224 12KB program space/8KB data space, 220VAC, 14DI/ 10DO relay outputs, 2A, one PPI, one Freeport, one EtherCAT port	CTH2 214-1BD33-0X24
CPU H226L 12KB program space/8KB data space, 24VDC, 24DI/16DO transistor source outputs, 0.5A, two PPI/Freeport, one EtherCAT port, 2*50KHz motion control output	CTH2 216-2AD33-0X40
CPU H226L 12KB program space/8KB data space, 220VAC, 24DI/16DO relay outputs, 2A, two PPI/ Freeport, one EtherCAT port	CTH2 216-2BD33-0X40
CPU H224X 16KB program space /108KB data space, 24VDC, 14DI/10DO transistor source outputs, 0.5A, one PPI, one Freeport, one EtherCAT port, 2*50KHz output(Pulse/Dir or PTO/PWM)	CTH2 214-1AX33-0X24
CPU H224X 16KB program space /108KB data space, 220VAC, 14DI/10DO relay outputs, 2A, one PPI, one Freeport, one EtherCAT port	CTH2 214-1BX33-0X24
CPU H226XL 72KB program space /110KB data space, 24VDC, 24DI/16DO transistor source outputs, 0.5A, two PPI/ Freeport, one EtherCAT port , 2*50KHz output(Pulse/Dir or PTO/PWM)	CTH2 216-2AX33-0X40
CPU H226XL 72KB program space /110KB data space, 220VAC, 24DI/16DO relay outputs, 2A, two PPI/ Freeport, one EtherCAT port	CTH2 216-2BX33-0X40
CPU H228XL 96KB program space /10KB data space, 220VAC, 36DI/24DO relay outputs, 2A, two PPI/ Freeport, one EtherCAT port	CTH2 218-3BX33-0X60
Expansion module	
SM221 digital input module, 8 inputs, 24VDC	CTH2 221-1BF32
SM221 digital input module, 16 inputs, 24VDC	CTH2 221-1BH32
SM221 digital input module, 32 inputs, 24VDC	CTH2 221-1BL32
SM222 digital output module, 8 transistor source outputs, 24VDC, 0.5A(output protection)	CTH2 222-1BF32
SM222 digital output module,16 transistor source outputs, 24VDC, 0.5A(output protection)	CTH2 222-1BH32
SM222 digital output module,32 transistor source outputs, 24VDC, 0.5A(output protection)	CTH2 222-1BL32
SM222 digital output module, 8 relay outputs, 2A	CTH2 222-1HF32
SM222 digital output module, 8 relay outputs, 2A	CTH2 222-1HH32
SM223 digital input/output module, 4*24VDC inputs, 4 transistor source outputs, 24VDC, 0.5A(output protection)	CTH2 223-1BF32
SM223 digital input/output module, 8*24VDC inputs, 8 transistor	CTH2 223-1BH32

source outputs, 24VDC, 0.5A(output protection)	
SM223 digital input/output module, 16*24VDC inputs, 16 transistor source outputs, 24VDC, 0.5A(output protection)	CTH2 223-1BL32
SM223 digital input/output module, 4*24VDC inputs, 4 relay outputs, 2A	CTH2 223-1HF32
SM223 digital input/output module, 8*24VDC inputs, 8 relay outputs, 2A	CTH2 223-1PH32
SM223 digital input/output module, 16*24VDC inputs, 16 relay outputs, 2A	CTH2 223-1PL32
SM231 analog input module, 4 inputs, 0~20 mA current or ±5V, ±2.5V, 0~10V, 0~5V voltage inputs, isolated 12 bits precision	CTH2 231-0HC32
SM231 high precision analog input module, 8 inputs, voltage input, opto-isolated, 16 bits precision	CTH2 231-0HF32
SM231 high precision analog input module, 8 inputs, current input, opto-isolated, 16 bits precision	CTH2 231-1HF32
SM231 analog voltage input module, 8 inputs, ±2.5V, 0~10V, 0~5V voltage input, two channels available 0~20mA current input, isolated 12 bits precision	CTH2 231-5HF32
SM231 thermal resistance temperature input module, 2*RTD, isolated 16 bits precision	CTH2 231-7PB32
SM231 thermal resistance temperature input module, 4*RTD, isolated 16 bits precision	CTH2 231-7PC32
SM231 thermocouple temperature input module, 4*TC, J/K/R/S/T/E/N, isolated 16 bits precision	CTH2 231-7PD32
SM231 thermocouple temperature input module, 8*TC, J/K/R/S/T/E/N, isolated 16 bits precision	CTH2 231-7PF32
SM231 thermocouple PID module, 4*J/K type, with intelligent PID, isolated 16 bits precision	CTH2 231-7TD32
SM231 thermocouple PID module, 8*J/K type, with intelligent PID, isolated 16 bits precision	CTH2 231-7TF32
SM231 8 current inputs, 0-20mA/4-20mA with intelligent PID, isolated 16 bits precision	CTH2 231-7HF32
SM231 hybrid temperature input module, 2*NTC or PT100, 2*0~20mA current or ±5V, ±10V, 0~10V, 0~5V voltage input, isolated 16 bits precision	CTH2 231-7ND32
SM231 thermocouple temperature input module, 8NTC/PT100, isolated 16 bits precision	CTH2 231-7NF32
SM231 weighing module, one sensor input, sampling frequency 50Hz, 0.01% module precision, 6VDC 150MA excitation power output each, isolated 16 bits precision	CTH2 231-7WA32
SM232 analog output module, 2 inputs, ±10V current or 0~20mA current output, isolated voltage 12 bits precision or current 11 bits precision	CTH2 232-0HB32
SM232 analog output module, 4 inputs, ±10V voltage or 0~20mA current output, isolated voltage 12 bits precision or current 11 bits	CTH2 232-0HD32

precision	
SM235 analog input/output module, 4 voltage or current input /1 voltage or current output, isolated voltage 12 bits precision or current 11 bits precision	CTH2 235-0KD32
SM253 motion control module, 2 ways single phase or AB phase HSC inputs, 200KHz, 2-axis PTO/PWM output, 200KHz, COTRUST motion control library	CTH2 253-1BH32
SM277A Profibus DP slave interface module, 12M communication rate, opto-isolated	CTH2 277-0AA32
SM277B Profibus DP slave module, 1.5M communication rate, opto-isolated	CTH2 277-0AB32
SM277C CAN slave module, with 8DI/6DO, opto-isolated, 7 modules can be extended	CTH2 277-0AC32
SM277 Profinet slave module, 100M communication rate, 8 modules can be extended	CTH2 277-0PN32
Expansion board	
EBH AMS-03 analog I/O expansion board, 2 *12 bits precision voltage inputs, 1 *12 bits precision voltage/current output	CTH2 AMS-03S1-EB
EBH-AMS-06 analog I/O expansion board, 4 *12 bits precision voltage inputs, 2 *12 bits precision voltage output	CTH2 AMS-06S1-EB
EBH-AMS-06 analog I/O expansion board,4 *12 bits precision voltage inputs, 2 *12 bits precision current output	CTH2 AMS-06S2-EB
EBH CAN-01 master communication expansion board, 1Mbps, opto-isolated	CTH2 CAN-01S1-EB
EBH PWM 04 charge-guided expansion board, double-channel voltage inputs, single-channel PWM output	CTH2 PWM-04S1-EB
Fittings	
PLC lithium battery	CTH2 291-8BA33
RS485 programming cable	CTS7 191-USB30
64K storage card(only for CPU H224X/H226XL/H228XL)	CTS7 291-MC064
256K storage card(only for CPU H224X/H226XL/H228XL)	CTS7 291-MC256
Programming card	CTS7 291-PC001

Address: Room 209, 210 at IC Design Industrial Park of
XiLi Chaguang Road, Nanshan District, Shenzhen
Hotline: 400-700-4858
E-mail: sales@co-trust.com
Http: //www.co-trust.com
SHENZHEN CO-TRUST TECHNOLOGY CO., LTD.

Scan for more info