



CTH200 系列可编程逻辑控制器



用户手册

权利声明

版权声明

深圳市合信自动化技术有限公司版权所有，并保留对本手册及本声明的最终解释权和修改权。未得到深圳市合信自动化技术有限公司的书面许可，任何单位及个人不得以任何方式或形式对本手册内的任何部分进行复制、摘录、备份、修改、传播、汇编、翻译成其它语言、将其全部或部分用于商业用途。

商标声明

、TrustPLC®、MagicWorks®、COTRUST®、MagiCampus®、COMOTION®、®均为深圳市合信自动化技术有限公司或其关联公司的注册商标，受法律保护，侵权必究。未经深圳市合信自动化技术有限公司或其关联公司的书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、永久下载、修改、传播、抄录或与其他产品捆绑销售，或注册为其域名或无线网址名称，或域名或无线网址的主要部分。

免责声明

本手册依据现有信息制作，其内容如有更改，恕不另行通知。深圳市合信自动化技术有限公司在编写该手册的时候已尽最大努力保证其内容准确可靠，但深圳市合信自动化技术有限公司不对本手册中的遗漏、不准确或印刷错误导致的损失和损害承担责任。

前言

感谢您购买深圳市合信自动化技术有限公司的 CTH200 系列可编程逻辑控制器产品！

在使用我公司 CTH200 系列 PLC 前，请您仔细阅读本手册，以便更清楚地掌握产品的特性，更安全地应用，充分利用本产品丰富的功能。

本手册主要描述 CTH200 系列可编程控制器和扩展模块的硬件规格、特性及安装使用方法，另外还有用户指令集汇总，产品订货号等供参考。

产品特性

CTH200 系列 PLC 是合信面向不断发展的中国 OEM 市场正式推出的自动化控制产品。该产品凝聚了合信工程师多年的丰富经验并及时响应工业 4.0 的市场需求，具备机型丰富、扩展多样、以太网互动、性能卓越、使用便捷等特点。为 OEM 客户提供高性价比的小型自动化解决方案，也将助力加快中国工业 4.0 的发展进程。

CPU 模块本体标配以太网接口，集成强大的以太网功能，可通过以太网接口与其他 CPU 模块、触摸屏或计算机进行通信，可快速便捷地组网。CTH200 CPU 同时也支持远程编程、调试、监控及数据交换，可通过以太网接口安全、快速、便捷地连接其他设备，实现设备的远程监控与维护。

适用对象

本手册提供关于 CTH200 系列 PLC 的安装和调试信息，面向工程师、安装人员、维护人员和具有自动化常识的电工等对象。

服务支持

深圳市合信自动化技术有限公司建立了售后维修服务中心，并提供电话热线服务。客户在产品使用过程中遇到问题时，可随时与深圳市合信自动化技术有限公司各地的服务支持联系。各地服务支持热线电话请到 <http://www.co-trust.com/Company/Contact/index.html> 获取。此外，客户还可通过深圳市合信自动化技术有限公司网站 <http://www.co-trust.com> 或官方微信及时了解最新产品动态，以及下载需要的技术文档。

安全注意事项

在开始使用之前，请认真阅读用户手册的注意事项，以避免意外事故的发生。

所负责产品安装、操作的人员必须经过严格培训，遵守相关行业的安全规范，严格遵守该手册提供的相关设备注意事项和特殊安全指示，按正确的操作方法进行设备的各项操作。

为防止对人的危害和对财产的损害，对务必遵守的事项特做以下说明。对错误使用本产品而可能带来的危害和损害程度见相关符号说明。



警告

该标记表示

“由于没有按要求操作造成的危险，可能导致人身伤亡”



注意

该标记表示

“由于没有按要求操作造成的危险，可能会导致人身轻度或中度伤害和设备损坏”



提示

该标记表示

“对操作的描述进行必要的补充或说明”

使用中的注意事项

使用中必须有安全电路，保证当外部电源掉电或可编程控制器故障时，可编程控制器的应用系统能安全工作。在使用的设计中应考虑方面包括：

必须在可编程控制器的外部电路中添加紧急制动电路、保护电路、正反转操作互锁电路和防止机器损坏的位置上限、下限互锁开关。

为确保使设备能安全运行，对重大事故相关的输出信号，请务必设计外部保护电路和安全机构。

可编程控制器的 CPU 检测到系统异常可能会导致所有输出关闭；当控制器的部分电路故障时，可能导致其输出不受控制，为保证设备正常运转，需加设合适的外部控制电路。

可编程控制器的继电器、晶体管等输出单元损坏时，会使其输出无法控制为 ON 或 OFF 状态。

电源系统应加有防雷装置，确保雷击过电压不会加在可编程控制器的电源输入与信号输入、控制输出端等端口，以免损坏设备。

安装时的注意事项

请勿在下列地方使用可编程控制器：有灰尘、油烟、导电性尘埃、腐蚀性气体、可燃性气体、有振动、冲击的地方，不要将控制器暴露在高温、结露、风雨的地方。电击、火灾、误操作也会导致产品损坏和恶化。

在扭紧螺丝和接线时，金属屑和电线头掉入控制器的通风孔时会引起火灾、故障、误操作等，请注意不要让金属屑和电线头掉入控制器内。

可编程控制器安装结束后，需保证通风面上没有杂物，否则可能导致运行时散热不畅，引起火灾、故障、误操作等。

要避免在带电状态下接线、插拔电缆插头，否则容易导致电击或电路损坏。

插拔通信插头时，必须先断电。带电插拔有可能烧坏通信端口而导致通信异常。

安装和接线必须牢固可靠，接触不良将可能导致误动作。

在干扰严重的应用场合，高频信号的输入与输出电缆应使用屏蔽电缆，来提高系统的抗干扰能力。

布线时的注意事项

必须在将外部电源全部切断后，方可进行安装、接线等操作，否则可能会引起触电或设备损坏。

安装布线完毕后，请立即清除杂物，通电前请盖好产品的端子盖板，以免引起触电。

按本手册中的提示接入电源。将电源接入其它端子，会烧毁可编程控制器。

在接 PLC 的输入、输出信号线时请不要与其它强电或强干扰线路并排布线，以减少干扰，应分开单独布线于分线槽中。

请将主模块的接地端子（PG）与强电系统的地分开，不要共地。

运行和保养时的注意事项

请不要在通电时触摸端子，以免引起电击、误操作等。

请务必在关闭电源后方可进行清扫和端子的扭紧等工作，在通电状态时这些操作可能会引起触电。

请务必在关闭电源后才能进行通信信号电缆的连接或拆除、扩展模块或控制单元的电缆连接或拆除等操作，否则会引起设备损坏、误操作等。

请不要拆解控制器，以免损坏内部元器件。

请务必仔细阅读熟悉本手册，充分确认安全后，再进行程序的变更、试运行、启动和停止等操作。

产品报废时的注意事项

电路板上的电解电容器焚烧时可能会发生爆炸。

可编程控制器主体为塑料件，焚烧时可能会产生有毒气体。

请按工业废弃物进行处理或者按当地环境保护规定处理。

版本修订记录

发行日期	修订后版本	修订内容
2021 年 9 月	V1.54	<ul style="list-style-type: none"> • 增加 MC_STOP_CTRL、MC_SET_PWM 指令描述, 参见章节 E.2 motion_ctrl_lib (v1.5) 库指令一览表 • 增加 avdPID 指令描述, 参见章节 D.1 CPU 嵌入式 PID_T 库的使用 • 完善特殊内存 SMB114 相关内容, 参见章节 W 特殊存储器功能说明 SMB114 至 SMB115
2021 年 8 月	V1.53	<ul style="list-style-type: none"> • 完善定时器具体使用范围, 参见章节 4.2.1 CPU 规范
2021 年 7 月	V1.52	<ul style="list-style-type: none"> • 完善高速计数器规格, 参见章节 4.2.1 CPU 规范 • 新增章节 F motion_ctrl_lib(v1.5)库文件介绍, G 回原功能, H 位置控制, I 速度控制, J 直线插补, K 圆弧插补。
2021 年 6 月	V1.51	<ul style="list-style-type: none"> • 新增 H226IM/H226IL/H226IH 规格, 参见章节 1 产品概述, 3.2 安装尺寸说明, 4.2.1 CPU 规范, 4.2.2 CPU 结构示意图和接线图。 • 增加 Trace 追踪功能 9 Trace 追踪功能。 • 订货单号信息完善, 参见章节 S 订货信息
2020 年 7 月	V1.50	<ul style="list-style-type: none"> • 新增 NPN 型模块 CTH2 222-1DF32 的规格参数。
2020 年 5 月	V1.40	<ul style="list-style-type: none"> • 增加章节 6.4 西门子 S7 协议通信举例 • 完善章节 5.8 西门子 S7 协议通信、S 订货信息
2020 年 4 月	V1.30	<ul style="list-style-type: none"> • 增加西门子 S7 协议通信, 参见章节 5.8 西门子 S7 协议通信 • 增加章节 M CANfree 通信指令库介绍, 高性能系列 CPU 支持 canfree 通信 • 增加硬件组态 CANopen 通信方式, 参见章节 6.1 CANopen 通信 • 增加 CTH2-CAN-01S2-EB 和新版编程卡订货信息说明
2019 年 11 月	V1.21	<ul style="list-style-type: none"> • 增加高性能升级版型 H226XM/H226XL/H224X 产品信息以及 RS485 扩展板订货信息 • 增加章节 L 以太网“ct_socket”通信库的使用
2019 年 6 月	V1.20	<ul style="list-style-type: none"> • 删掉原 5.1 章节 • 增加章节 5.5 以太网通信、5.6 Profibus-DP 通信和 5.7 PROFINET 通信 • 增加章节 4.7.4 SM277 PN 从站模块规范, 并在附录添加其订货信息, 同时附录还增加 CTH200 系列配件订货信息 • 增加附录 N 编程电缆, 删掉原 3.4 章节并新增附录 O 编程卡使用说明替换其原有内容 • 删掉 SDO 指令说明

目 录

权利声明.....	II
前 言.....	III
安全注意事项	IV
版本修订记录	VI
目 录.....	VII
1 产品概述.....	1
1.1 CPU 简介.....	1
1.2 扩展模块	4
1.3 系统最大配置	6
1.4 适用面板	7
1.5 网络架构	7
1.6 产品使用标准及规范	8
2 使用入门.....	10
2.1 连接 CTH200 PLC	10
2.2 与 CTH200 建立通信	12
2.2.1 选择串口通信方式.....	12
2.2.2 选择以太网通信方式.....	13
2.2.3 在系统块中修改 IP 地址.....	15
2.3 创建程序段.....	15
2.3.1 编辑程序	16
2.3.2 编译程序	17
2.3.3 下载程序	18
2.3.4 运行 PLC	18
2.4 CPU 在线升级固件.....	19
3 安装.....	21
3.1 安装注意事项	21
3.2 安装尺寸说明	23
3.3 安装方法	24
3.4 接地和布线.....	25
3.5 抑制电路	26
4 技术规范.....	27
4.1 通用规范	27
4.2 CPU 规范和接线图.....	27
4.2.1 CPU 规范.....	27
4.2.2 CPU 结构示意图和接线图	39
4.2.3 通信接口信号定义.....	46
4.2.4 拨码开关定义	47
4.2.5 制作标准网线.....	47

4.3	数字量扩展模块规范	47
4.3.1	数字量输入模块规范	48
4.3.2	数字量输出模块规范	49
4.3.3	数字量输入输出模块规范	53
4.4	模拟量扩展模块规范	57
4.4.1	模拟量输入模块规范	58
4.4.2	模拟量输出模块规范	65
4.4.3	模拟量输入输出模块规范	67
4.5	温度测量模块规范	70
4.5.1	热电阻测量模块规范	71
4.5.2	热电偶测量模块规范	76
4.5.3	混合温度输入模块规范	80
4.6	PID 控制模块规范	85
4.7	通信模块规范	96
4.7.1	SM277A DP 从站接口模块规范	97
4.7.2	SM277B DP 从站模块规范	99
4.7.3	SM277C CANopen 从站模块规范	102
4.7.4	SM277 PN 从站模块规范	104
4.8	称重模块规范	106
4.9	运动控制模块规范	108
4.10	BD 扩展板规范及安装方法	110
4.10.1	模拟量扩展板规范	111
4.10.2	CAN 扩展板规范	113
4.10.3	导引电路扩展板规范	114
4.10.4	RS485 扩展板规范	118
4.10.5	BD 扩展板的安装	118

5 网络通信协议及组网方式 120

5.1	PPI 通信	120
5.2	MPI 通信	121
5.3	自由口通信	122
5.4	CANopen 通信	122
5.5	以太网通信	123
5.6	Profibus-DP 通信	124
5.7	PROFINET 通信	125

6 应用举例 126

6.1	CANopen 通信	126
6.1.1	示例组件	126
6.1.2	网络连接	127
6.1.3	操作步骤	127
6.1.4	导入第三方 EDS 文件	141
6.1.5	诊断	143
6.2	Modbus TCP 通信	145
6.2.1	示例组件	146
6.2.2	网络连接	146

6.2.3	操作步骤	146
6.2.4	ModBus TCP 从站地址映射	148
6.3	UDP_PPI 通信	149
6.3.1	示例组件	149
6.3.2	网络连接	150
6.3.3	操作步骤	150
6.3.4	UDP_PPI 通信从站地址映射	154
6.4	西门子 S7 协议通信举例	154
6.4.1	示例组件	155
6.4.2	网络连接	155
6.4.3	西门子产品与 CTH2 系列高性能升级版型 PLC 的网口通讯	155
6.4.4	其它产品与 CTH2 系列高性能升级版型 PLC 的网口通讯	168
6.5	模拟量输入输出扩展板应用	172
6.5.1	安装注意事项	172
6.5.2	扩展板访问	173
6.6	配方和数据记录应用	173
6.6.1	使用配方	173
6.6.2	使用数据记录	181

7 电源预算 188

7.1	电源要求	188
7.2	5VDC 电源	189
7.3	24VDC 电源	190
7.4	电源预算计算示例	191

8 故障诊断 192

8.1	通过 MagicWorks PLC 进行诊断	192
8.2	通过 CTH200 系列主控模块进行诊断	195

9 Trace 追踪功能 196

附录 199

A	CT-MODBUS 主站和从站库的使用	199
A.1	CT_MODBUS 库介绍	199
A.2	安装说明	199
A.3	CT_MODBUS 库功能说明	201
B	CTH200 的 CPU 扩展程序空间使用	207
B.1	功能介绍	207
B.2	使用说明	207
C	CTH200 系列 PLC 扩展 100K 数据块库的使用	211
C.1	功能介绍	211
C.2	安装说明	211
C.3	Ext_mem 库功能说明	213
C.4	应用例子	214
D	CTH200 系列 PLC PID 控制库的使用	215
D.1	CPU 嵌入式 PID_T 库的使用	215
D.2	CTH200 系列 PLC 热电偶型温度 PID 模块控制库使用	222
E	motion_ctrl_lib(v1.5)库文件介绍	228

E.1	添加库文件	228
E.2	motion_ctrl_lib (v1.5) 库指令一览表	230
F	回原功能	263
F.1	回原模式	263
F.2	回原动作	263
G	位置控制	271
G.1	位置控制类型	271
G.2	单轴相对运动指令使用	272
H	速度控制	273
H.1	速度控制类型	273
H.2	单轴速度运动指令使用	274
I	直线插补	276
I.1	两轴直线插补运动指令使用	276
J	圆弧插补	277
J.1	两轴圆弧插补运动指令使用	277
K	SM231 称重模块及对应库函数的使用	279
K.1	称重库函数说明	280
K.2	称重库说明	284
L	SM277A 模块的使用	288
M	SM277B 模块的使用	291
M.1	PROFIBUS-DP 网络结构	291
M.2	SM277B 的组件	291
M.3	使用指南	292
N	永久保存 V 内存功能库“CT_savevmem”的使用	304
O	SM253 运动控制模块库的使用	306
O.1	功能介绍	306
O.2	安装说明	306
O.3	sm253_motion_ctrl_lib 库指令一览表	308
O.4	sm253_motion_ctrl_lib 库指令详解	308
O.5	调试示例	330
P	以太网设置指令库	334
P.1	功能介绍	334
P.2	指令详解	334
P.3	应用示例	337
Q	以太网“ct_socket”通信库的使用	338
Q.1	功能介绍	338
Q.2	使用说明	339
R	CANfree 通信指令库介绍	343
R.1	功能介绍	343
R.2	添加库文件	343
R.3	库指令说明	345
S	指令集	349
S.1	指令速查	349
S.2	指令详解	354
T	编程电缆	445
T.1	安装驱动	446

T.2	更改 COM 端口号	451
T.3	释放 COM 口	452
U	编程卡使用说明.....	454
V	电池卡和存储卡使用说明	458
W	特殊存储器功能说明	459
X	订货信息	472

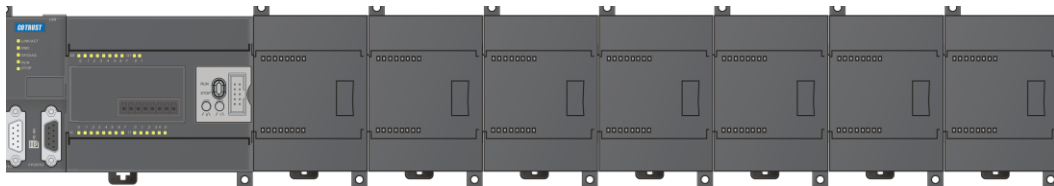
1 产品概述

CTH200 系列 PLC 是合信面向不断发展的中国 OEM 市场正式推出的自动化控制产品。该产品凝聚了合信工程师多年的丰富经验并及时响应工业 4.0 的市场需求，具备机型丰富、扩展多样、以太网互动、性能卓越、使用便捷等特点。为 OEM 客户提供高性价比的小型自动化解决方案，也将助力加快中国工业 4.0 的发展进程。

CPU 模块本体标配以太网接口，集成强大的以太网功能，可通过以太网接口与其他 CPU 模块、触摸屏或计算机进行通信，可快速便捷地组网。CTH200 CPU 同时也支持远程编程、调试、监控及数据交换，可通过以太网接口安全、快速、便捷地连接其他设备，实现设备的远程监控与维护。

CTH200 是该系列所有主机产品型号的统称，具体包括 CPU H224、CPU H226L、CPU H226M、CPU H224X、CPU H226XL、CPU H226IM、CPU H226IL、CPU H226IH、CPU H226XM、CPU H228XL 共 24 款 CPU 产品（CPU H228XL 为继电器输出，标准型每款包括继电器输出和晶体管输出两个型号）；CTH200 系列 CPU 可以支持多类型的扩展模块，其模块类型包括但不限于：数字量输入输出、模拟量输入输出、温度采集、DP 通信模块及 profinet 从站模块等，同时也支持模拟量及 CAN 扩展板，方便您在解决工业自动化问题时灵活选用。

CTH200 系列 PLC 通过以太网通信口或 RS485 通信口与上位机进行通信，使用 MagicWorks PLC（MagicWorks PLC V2.08 及以上版本）或 MicroWin 编程（网口不适用），两种编程平台可互相切换，简单易用。CTH200 系统可以由一个单独的 CPU 组成，也可由一个 CPU 外加最多 7 个各种型号的可选扩展模块组成。



1.1 CPU 简介

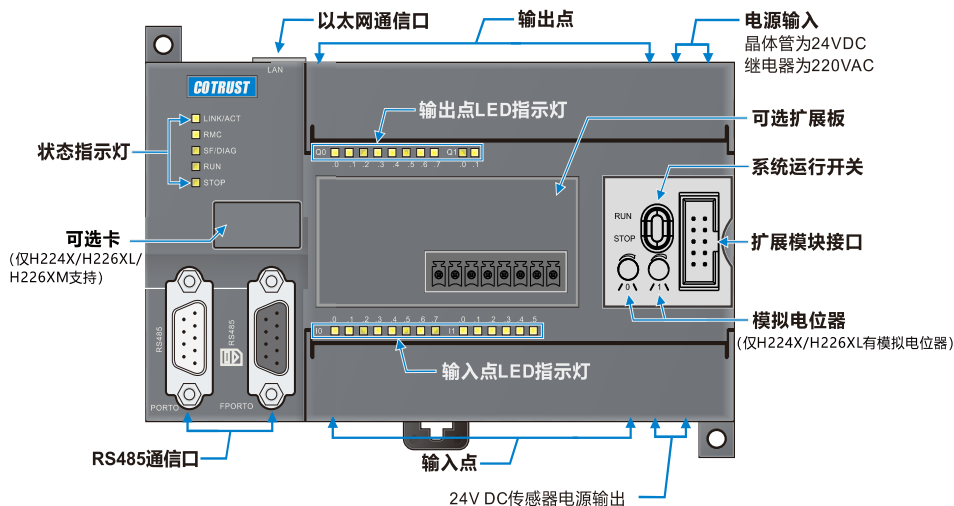


图 1-1 CTH200 系列 PLC（24 点）结构示意图

表 1-1 CTH200 系列 CPU 主控模块

产品名称与规格	产品订货号
基本型	
CPU H224 12KB 程序空间/8KB 数据空间, 220VAC 电源, 14DI/10DO 继电器输出, 2A, 1 个 PPI, 1 个自由通讯口, 1 个以太网通讯口, 不支持模块和扩展板	CTH2 214-1BA33-0X24
CPU H226L 12KB 程序空间/8KB 数据空间, 220VAC 电源, 24DI/16DO 继电器输出, 2A, 2 个 PPI/自由通讯口, 1 个以太网通讯口, 不支持模块和扩展板	CTH2 216-1BA33-0X40
标准型	
CPU H224 12KB 程序空间/8KB 数据空间, 24VDC 电源, 14DI/10DO 晶体管源型输出, 0.5A, 1 个 PPI, 1 个自由通讯口, 1 个以太网通讯口, 3 路 50KHz 运控输出	CTH2 214-1AD33-0X24
CPU H224 12KB 程序空间/8KB 数据空间, 220VAC 电源, 14DI/10DO 继电器输出, 2A, 1 个 PPI, 1 个自由通讯口, 1 个以太网通讯口	CTH2 214-1BD33-0X24
CPU H226L 12KB 程序空间/8KB 数据空间, 24VDC 电源, 24DI/16DO 晶体管源型输出, 0.5A, 2 个 PPI/自由通讯口, 1 个以太网通讯口, 3 路 50KHz 运控输出	CTH2 216-2AD33-0X40
CPU H226L 12KB 程序空间/8KB 数据空间, 220VAC 电源, 24DI/16DO 继电器输出, 2A, 2 个 PPI/自由通讯口, 1 个以太网通讯口	CTH2 216-2BD33-0X40
CPU H226M 12KB 程序空间/8KB 数据空间, 24VDC 电源, 14DI/10DO 晶体管源型输出, 0.5A, 2 个 PPI/自由口, 1 个以太网口, 3 路 50KHz 运控输出	CTH 216-1AD33-0X24
CPU H226M 12KB 程序空间/8KB 数据空间, 220VAC 电源, 14DI/10DO 继电器输出, 2A, 2 个 PPI/自由口, 1 个以太网口	CTH 216-1BD33-0X24
高性能型	
CPU H224X 16KB 程序空间/108KB 数据空间, 24VDC 电源, 14DI/10DO 晶体管源型输出, 0.5A, 1 个 PPI, 1 个自由通讯口, 1 个以太网通讯口, 2 路 50KHz 输出 (Pulse/Dir 或 PTO/PWM)	CTH2 214-1AX33-0X24
CPU H224X 16KB 程序空间/108KB 数据空间, 220VAC 电源, 14DI/10DO 继电器输出, 2A, 1 个 PPI, 1 个自由通讯口, 1 个以太网口	CTH2 214-1BX33-0X24
CPU H226XL 72KB 程序空间/110KB 数据空间, 24VDC 电源, 24DI/16DO 晶体管源型输出, 0.5A, 2 个 PPI/自由通讯口, 1 个以太网通讯口, 2 路 50KHz 输出 (Pulse/Dir 或 PTO/PWM)	CTH2 216-2AX33-0X40
CPU H226XL 72KB 程序空间/110KB 数据空间, 220VAC 电源, 24DI/16DO 继电器输出, 2A, 2 个 PPI/自由通讯口, 1 个以太网通讯口	CTH2 216-2BX33-0X40
CPU H228XL 96KB 程序空间/10KB 数据空间, 220VAC 电源, 36DI/24DO 继电器输出, 2A, 2 个 PPI/自由通讯口, 1 个以太网通讯口	CTH2 218-3BX33-0X60
高性能型升级版	
CPU H224X 16KB 程序空间/108KB 数据空间, 24VDC 电源, 14DI/10DO 晶体管源型输出, 0.5A, 1 个 PPI, 1 个自由通讯口, 1 个以太网通讯口, 2 路 20KHz (PTO/PWM) 或 2 路 100KHz 输出 (Pulse/Dir), 数据永久保存, 支持模拟量和 CAN 扩展板	CTH2 214-1AX35-0X24
CPU H224X 16KB 程序空间/108KB 数据空间, 220VAC 电源, 14DI/10DO 继电器输出, 2A, 1 个 PPI, 1 个自由通讯口, 1 个以太网口, 数据永久保存, 支持模拟量和 CAN 扩展板	CTH2 214-1BX35-0X24

<p>CPU H226XM 72KB 程序空间/110KB 数据空间，24VDC 电源，14DI/10DO 晶体管源型输出，0.5A，2 个 PPI/自由口，1 个以太网口，两路 20KHz（PTO/PWM），3 路 100KHz 输出（Pulse/Dir），数据永久保存，支持模拟量、CAN 和 RS485 扩展板</p>	<p>CTH2 216-1AX35-0X24</p>
<p>CPU H226XM 72KB 程序空间/110KB 数据空间，220VAC 电源，14DI/14DO 继电器输出，2A，2 个 PPI/自由口，1 个以太网口，数据永久保存，支持模拟量、CAN 和 RS485 扩展板</p>	<p>CTH2 216-1BX35-0X24</p>
<p>CPU H226XL 72KB 程序空间/110KB 数据空间，24VDC 电源，24DI/16DO 晶体管源型输出，0.5A，2 个 PPI/自由通讯口，1 个以太网通讯口，2 路 20KHz（PTO/PWM）或 2 路 100KHz 输出（Pulse/Dir），数据永久保存，支持模拟量、CAN 和 RS485 扩展板</p>	<p>CTH2 216-2AX35-0X40</p>
<p>CPU H226XL 72KB 程序空间/110KB 数据空间，220VAC 电源，24DI/16DO 继电器输出，2A，2 个 PPI/自由通讯口，1 个以太网口，数据永久保存，支持模拟量、CAN 和 RS485 扩展板</p>	<p>CTH2 216-2BX35-0X40</p>
<p>CPU H226IM 256KB 程序空间/164KB 数据空间，24VDC 电源，14DI/10DO 晶体管源型输出，0.5A，2 个 PPI/自由口，1 个以太网口，2 路 20KHz（PTO/PWM），3 路 100KHz 输出（Pulse/Dir），数据永久保存，支持模拟量、CAN 和 RS485 扩展板，支持西门子 S7 协议</p>	<p>CTH2 216-1AD46-0X24</p>
<p>CPU H226IM 256KB 程序空间/164KB 数据空间，220VAC 电源，14DI/10DO 继电器输出，2A，2 个 PPI/自由口，1 个以太网口，数据永久保存，支持模拟量、CAN 和 RS485 扩展板，支持西门子 S7 协议</p>	<p>CTH2 216-1BD46-0X24</p>
<p>CPU H226IL 256KB 程序空间/164KB 数据空间，24VDC 电源，24DI/16DO 晶体管源型输出，0.5A，2 个 PPI/自由口，1 个以太网口，2 路 20KHz（PTO/PWM），3 路 100KHz 输出（Pulse/Dir），数据永久保存，支持模拟量、CAN 和 RS485 扩展板，支持西门子 S7 协议</p>	<p>CTH2 216-2AD46-0X40</p>
<p>CPU H226IL 256KB 程序空间/164KB 数据空间，220VAC 电源，24DI/16DO 继电器输出，2A，2 个 PPI/自由通讯口，1 个以太网口，数据永久保存，支持模拟量、CAN 和 RS485 扩展板，支持西门子 S7 协议</p>	<p>CTH2 216-2BD46-0X40</p>
<p>CPU H226IH 256KB 程序空间/164KB 数据空间，24VDC 电源，14DI/10DO 晶体管漏型输出，0.5A，2 个 PPI/自由口，1 个以太网口，4 路 200KHz 输出（Pulse/Dir）数据永久保存，支持模拟量、CAN 和 RS485 扩展板，支持西门子 S7 协议</p>	<p>CTH2 216-1AH46-2B24</p>

CPU 特性:

稳定、可靠

- 抗 ESD、抗 EFT，三防处理，经过严格的工业环境适应性测试，通过 CE 认证
- 用户程序、数据永久保存；内置实时时钟
- 多重密码保护，更有核心程序单向下载功能，永久保密
- 通信接口防雷处理，可靠性高

通信能力强

- 支持以太网通信、MPI、PPI、自由口、MODBUS、profinet、Profibus-DP 等多种通信协议
- 网络读写一次可交换 200 Bytes，组网通信能力强
- 支持远程编程、调试、监控及数据交换
- 支持西门子 S7 协议从站功能

智能化程度高

- 集成参数自整定模糊逻辑算法温控 PID 库，温控准确，动态性能好
- 丰富的内置集成功能，使编程更简单、控制更灵活
- 高速闭环能力能满足一些高速系统的应用
- 高性能的运动控制功能及丰富的应用，非常方便的实现一些同步和定位功能

高速、大容量

- 逻辑和浮点运算速度快，高性能 CPU 分别可达 0.15 μ s 和 8 μ s
- 程序容量大，最高可达 256KB；数据空间大，最高可达 164KB
- 高性能 X 系列 CPU 拥有多达 6 个高速输入计数器
- 满足高速脉冲输出，其中 H226XM 为三轴 100KHz
- 支持网口升级固件，方便快捷

超强系统扩展

- 超强模拟量 I/O，最高可达 194AI/194AQ，标准型达 32AI/32AQ
- 超强数字量 I/O，最高可达 640DI/640DQ，标准型达 128DI/128DQ
- 最多可扩展 7 个 I/O 模块和 1 个扩展板（基本型不支持扩展）

1.2 扩展模块

CTH200 系列 CPU 本机提供一些 I/O，也可借助扩展模块提供扩展更多 I/O 及通信功能。CTH200 系列产品的扩展模块种类非常多，主要包括数字量输入输出扩展模块、模拟量输入输出模块、温度测量模块和通信模块等类别，每个类型模块点数多样，可以灵活配置各种 I/O 规模并且具有很高的性价比。所有模块采用 DIN35 标准导轨安装方式，安装非常方便。

表 1-2 扩展模块及规格

模块名称与规格	订货号
SM221 数字量输入模块，8 点输入，24VDC	CTH2 221-1BF32
SM221 数字量输入模块，16 点输入，24VDC	CTH2 221-1BH32
SM221 数字量输入模块，32 点输入，24VDC	CTH2 221-1BL32
SM222 数字量输出模块，8 点晶体管输出，24VDC，0.5A(输出保护)	CTH2 222-1BF32
SM222 数字量输出模块，8 点晶体管 NPN 型输出，24VDC，0.5A	CTH2 222-1DF32
SM222 数字量输出模块，16 点晶体管输出，24VDC，0.5A(输出保护)	CTH2 222-1BH32
SM222 数字量输出模块，32 点晶体管输出，24VDC，0.5A(输出保护)	CTH2 222-1BL32
SM222 数字量输出模块，8 点继电器输出，2A	CTH2 222-1HF32
SM222 数字量输出模块，16 点继电器输出，2A	CTH2 222-1HH32
SM223 数字量输入/输出模块，4 点 24VDC 输入，4 点晶体管输出，24VDC，0.5A(输出保护)	CTH2 223-1BF32
SM223 数字量输入输出模块，8 点 24VDC 输入，8 点晶体管输出，24VDC，	CTH2 223-1BH32

0.5A(输出保护)	
SM223 数字量输入输出模块, 16 点 24VDC 输入, 16 点晶体管输出, 24VDC, 0.5A(输出保护)	CTH2 223-1BL32
SM223 数字量输入/输出模块, 4 点 24VDC 输入, 4 点继电器输出, 2A	CTH2 223-1HF32
SM223 数字量输入输出模块, 8 点 24VDC 输入, 8 点继电器输出, 2A	CTH2 223-1PH32
SM223 数字量输入输出模块, 16 点 24VDC 输入, 16 点继电器输出, 2A	CTH2 223-1PL32
SM231 模拟量输入模块, 4 点, 0~20 mA 电流或±5V, ±2.5V, 0~10V, 0~5V 电压输入, 隔离型 12 位精度	CTH2 231-0HC32
SM231 高精度模拟量输入模块, 8 点输入, 电压输入, 光偶隔离 16 位精度	CTH2 231-0HF32
SM231 高精度模拟量输入模块, 8 点输入, 电流输入, 光偶隔离 16 位精度	CTH2 231-1HF32
SM231 模拟量电压输入模块, 8 点, ±2.5V, 0~10V, 0~5V 电压输入且两通道可选 0~20 mA 电流输入, 隔离型 12 位精度	CTH2 231-5HF32
SM231 热电阻温度输入模块, 2 点 RTD, 隔离型 16 位精度	CTH2 231-7PB32
SM231 热电阻温度输入模块, 4 点 RTD, 隔离型 16 位精度	CTH2 231-7PC32
SM231 热电偶温度输入模块, 4 点 TC, J/K/R/S/T/E/N, 隔离型 16 位精度	CTH2 231-7PD32
SM231 热电偶温度输入模块, 8 点 TC, J/K/R/S/T/E/N, 隔离型 16 位精度	CTH2 231-7PF32
SM231 热电偶 PID 模块, 4 点 J/K 型, 带智能 PID, 隔离型 16 位精度	CTH2 231-7TD32
SM231 热电偶 PID 模块, 8 点 J/K 型, 带智能 PID, 隔离型 16 位精度	CTH2 231-7TF32
SM231, 8 路电流输入, 0-20mA/4-20mA, 带智能 PID, 隔离型 16 位精度	CTH2 231-7HF32
SM231 混合温度输入模块, 2 点 NTC 或 PT100, 2 点 0~20mA 电流或±5V, ±10V, 0~10V, 0~5V 电压输入, 隔离型 16 位精度	CTH2 231-7ND32
SM231 热电阻温度输入模块, 8NTC/PT100, 隔离型 16 位精度	CTH2 231-7NF32
SM231 称重模块, 1 路传感器信输入, 采样频率 50Hz, 0.01%模块精度, 6VDC 每路 150MA 激励电源输出, 隔离型 16 位精度	CTH2 231-7WA32
SM232 模拟量输出模块, 2 点, ±10V 电压或 0~20mA 电流输出, 隔离型电压 12 位精度或电流 11 位精度	CTH2 232-0HB32
SM232 模拟量输出模块, 4 点, ±10V 电压或 0~20mA 电流输出, 隔离型电压 12 位精度或电流 11 位精度	CTH2 232-0HD32
SM235 模拟量输入输出模块, 4 点电压或电流输入/1 点电压或电流输出, 隔离型电压 12 位精度或电流 11 位精度	CTH2 235-0KD32
SM253 运动控制模块, 2 路单相或 AB 相 HSC 输入, 200KHz, 2 轴 PTO/PWM 输出, 200KHz, 合信运动控制库	CTH2 253-1BH32
通信模块	
SM277A Profibus DP 从站接口模块, 12M 通信速率, 光电隔离	CTH2 277-0AA32
SM277B Profibus DP 从站模块, 1.5M 通信速率, 光电隔离	CTH2 277-0AB32
SM277C CAN 从站模块, 自带 8DI/6DO, 光电隔离, 可扩展 7 个模块	CTH2 277-0AC32
SM277 Profinet 从站模块, 100M 通信速率, 可扩展 8 个模块	CTH2 277-0PN32

表 1-3 BD 扩展板

模块名称	规格	订货号
EBH AMS-03	模拟量输入输出扩展板, 2 点*12 位精度电压输入, 1 点*12 位精度电压/电流输出	CTH2 AMS-03S1-EB
EBH-AMS-06	模拟量输入输出扩展板, 4 点*12 位精度电压输入, 2 点*12 位精度电压输出	CTH2 AMS-06S1-EB
EBH-AMS-06	模拟量输入输出扩展板, 4 点*12 位精度电压输入, 2 点*12 位精度电流输出	CTH2 AMS-06S2-EB
EBH CAN-01	CAN 主站通信扩展板, 1Mbps, 光电隔离, 支持 EasyCAN 组态	CTH2 CAN-01S1-EB
EBH CAN-02	CAN 主站通信扩展板, 1Mbps, 光电隔离, 支	CTH2 CAN-01S2-EB

	持 CANfree 和 CAN 硬件组态及第三方 EDS 文件导入	
EBH PWM-04	充电导引扩展板，双通道电压输入，单通道可调脉宽 PWM 输出	CTH2 PWM-04S1-EB
EBH-RS485-01	一路 RS485 通信扩展板隔离型	CTH2 485-01S1-EB

**注意**

CTH200 系列中，H224/H226L/H226M 不支持 CAN-01 主站扩展板。

扩展模块技术特性：**可带扩展模块数**

CTH200 系列 CPU 最多可支持七个扩展模块。

滤波技术

所有的模拟量模块本身都集成 CPU，采用了先进的滤波技术，所以采样的稳定性很高。

数字量模块

数字量模块的所有输入输出都采用光电隔离技术和抗扰动技术，可靠性高。

温度采集模块

采用总线、电源、通道间都隔离的三隔离技术，24 位采样 A/D，抗干扰能力非常强、采样精度非常高。而且具有比较智能的故障诊断功能，安全性高。

智能模块

主要有温度控制 PID 等模块，大大加快程序执行的响应时间。

通信模块

Profibus-DP 从站模块、CANopen 从站模块及 PROFINET 从站模块，显著提升系统的互连性及通信能力

CAN 主站通信扩展板

CAN 主站通信扩展板，显著提升系统的通信能力

模拟量输入输出扩展板

模拟量输入输出扩展板，2 点*12 位精度电压输入，1 点*12 位精度电压/电流输出

1.3 系统最大配置

模块数量：最多 7 个扩展模块。

CPU 型号	数字 I/O 映象区	模拟量 I/O 映象区
H224/H226L/H226M	128DI/128DQ	32AI/32AQ
H224X/H226XL/H226XM/H228XL H226IM/H226IL/H226IH	128DI/128DQ (含 CAN 扩展可达 640DI/640DQ)	32AI/32AQ (含 CAN 扩展可达 194AI/194AQ)

1.4 适用面板

CTH200 系列 PLC 支持 COPANEL 系列显示面板。

显示面板具有文本显示和人机交互功能，可与 PLC 建立通信，方便用户查看、编辑和监控应用程序中的过程变量，帮助 PLC 编程和运行。

1.5 网络架构

CTH200 系统的典型网络架构如下图所示：

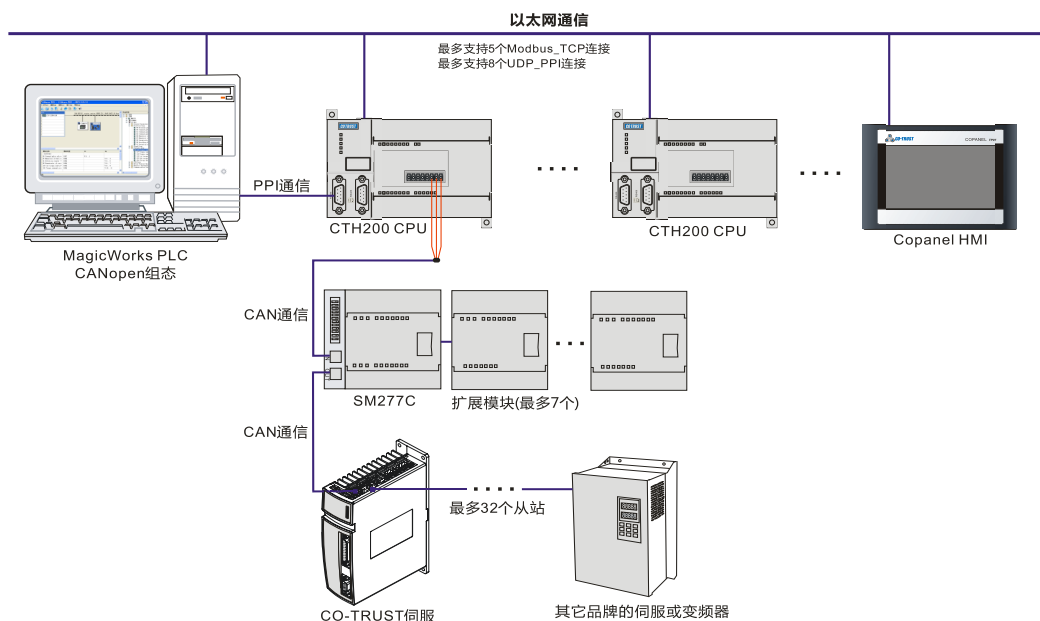


图 1-2 CTH200 系统网络架构

注意：

1、PLC 与 PLC 通信

- H224/H226L/H226M：8 个 UDP_PPI 连接，不分主从，每个连接最大通信 200 个字节。5 个 MODBUS_TCP 连接，不分主从，每个连接最大通信 240 个字节；4 个 socket 连接，2 个 UDP，2 个 TCP。
- （高性能型）H224X/H226XL/H228XL：8 个 UDP_PPI 连接，其中 4 个主站和 4 个从站，每个连接最大通信 200 个字节。4 个 MODBUS_TCP 连接，其中 2 个主站和 2 个从站，每个连接最大通信 240 个字节。
- （高性能型升级版）H224X/H226XM/H226XL/H226IM/H226IL/H226IH：8 个 UDP_PPI 连接，不分主从；8 个 Modbus_TCP 连接，不分主从，4 个 socket 连接，2 个 UDP，2 个 TCP。

2、PLC 与 HMI（或第三方上位机软件）连接，其中 HMI 或第三方上位机软件作为主站

UDP_PPI: H224/H226L/H226M 单台 CPU 最多连接 8 个 HMI; H224X/H226XL/H226XM/H228XL 最多连接 4 个 HMI。

MODBUS_TCP: H224/H226L/H226M 单台 CPU 最多连接 5 个 HMI; H224X/H226XL/H226XM 最多连接 2 个 HMI; H228XL 最多连接 3 个 HMI。

3、CPU 作为主站访问其他设备，相当于占用了该 CPU 一个主站连接。

4、CPU 作为从站，被其他设备访问，相当于占用该 CPU 的一个从站连接。

1.6 产品使用标准及规范

CTH200 系列 PLC 经过多项国际和行业标准测试，其物理与运行环境条件等规范要求如表 1-4:

表 1-4 CTH200 系列 PLC 产品标准和规范

环境条件		
运输和存储		
温度	-25°C ~ 70°C	
大气压	1080hPa~660hPa (对应高度为-1000m ~ +3500m)	
相对湿度	5%~95%，非结露	
跌落	1m, 5 次, 运输包装	
工作		
温度	水平安装位置	0°C~55°C
	垂直安装位置	0°C~45°C
大气压	1080hPa~795hPa (对应高度为-1000m~+2000m)	
相对湿度	10%~95%，非结露	
恶劣环境污染物浓度	较低盐雾、潮湿、尘雾等环境 SO ₂ <0.5ppm, 相对湿度<60%, 非结露 H ₂ S<0.1ppm, 相对湿度<60%, 非结露	
电磁兼容性 - 抗扰度		
静电放电 IEC61000-4-2	接触放电: ±4KV (等级 A) 空气放电: ±8KV (等级 A)	
电快速瞬变脉冲群 IEC61000-4-4	电源线: 2KV, 5KHz (等级 A) 信号线: 2KV, 5KHz (I/O 耦合夹) (等级 A) 1KV, 5KHz (通信耦合夹) (等级 A)	
浪涌 IEC61000-4-5	电源线: 2KV (非对称), 1KV (对称) (等级 B)	
射频电磁场辐射 IEC61000-4-3	80MHz~1GHz, 10V/m, 80%AM (1KHz) (等级 A) 1.4GHz~2GHz, 3V/m, 80%AM (1KHz) 2GHz~2.7GHz, 1V/m, 80%AM (1KHz)	
射频场感应传导干扰 IEC61000-4-6	0.15MHz~80MHz, 10V/m, 80%AM (1KHz) (等级 A) 15KHz~150KHz, 10V/m, 80%AM (1KHz)	
直流电源输入端口短时中断和电压变化 IEC61000-4-29	短时中断: 10ms 电压变化: 80%~120%, 100ms	
抗阻振荡波性 IEC61000-4-12	电源线: 1KV 数字量 I/O (24V 或者更高): 1KV	
电磁兼容性 - 传导与辐射发射		
辐射 (噪声) 电磁干扰 EN55011, A 类 1 组	测量距离为: 10m 30MHz~230MHz, < 40dB (uV/m) 峰值 230MHz~1000MHz, < 47dB (uV/m) 峰值	
传导 (噪声) 交流电源干扰 EN55011, A 类 1 组	测量距离为: 10m 0.15~0.5MHz, < 79dB (uV/m) 峰值; < 66dB (uV/m) 平均值	

	0.5~5MHz, < 73dB (uV/m) 峰值; < 60dB (uV/m) 平均值 5~30MHz, < 73dB (uV/m) 峰值; < 60dB (uV/m) 平均值
环境测试指标	
高温运行 IEC60068-2	60°C 16 小时
低温运行 IEC60068-2	-10°C 16 小时
高温启机 IEC60068-2	60°C 2 小时
低温启机 IEC60068-2	-10°C 2 小时
高低温循环运行 IEC60068-2	-10°C~60°C 下驻留时间 3 小时, 温升速率 1°C/min, 2 个循环
高温存储 IEC60068-2	70°C 72 小时
低温存储 IEC60068-2	-40°C 72 小时
冷热冲击 IEC60068-2	-40°C~70°C 下驻留时间 3 小时, 温变时间<1 分钟, 5 个循环
高温高湿 IEC60068-2	40°C 48 小时
交变湿热 IEC60068-2	25°C~55°C 95%, 2 个循环
正弦振动 (裸机) IEC60068-2	5~150Hz, 0.05G ² /Hz 150Hz~500Hz -3dB/oct, 1 小时/轴, X、Y、Z 总共 3 轴
冲击 (裸机) IEC60068-2	15G, 11ms 脉冲, 3 次/方向
流动混合气体腐蚀试验 IEC60068-2-60	H2S: 0.1ppm, NO2: 0.2ppm, CL2: 0.02ppm, 温度: 30°C, 湿度: 75%, 周期: 4 天
高压绝缘测试指标	
24V/5V 标称电路间	500 VAC
110V/220V 电路对地	1500 VAC
110V/220V 电路间	1500 VAC
110V/220V 对 24V/5V 电路	1500 VAC

2 使用入门

本章主要展示如何在 MagicWorks PLC 编程软件中对 CTH200 系列 PLC 进行连接、编程及程序的运行。

2.1 连接 CTH200 PLC

- 在连接 CTH200 PLC 时，先用通信电缆 RS485（订货号：CTS7 191-USB30）将 CTH200 PLC 连接到编程设备，然后再给 CTH200 CPU 供电。
- CTH200 系列 PLC 支持以太网通信，选用以太网通信时，请使用标准网线将 CTH200 PLC 连接到编程设备。

给 CTH200 CPU 供电

下图列出了两种 CPU 模块供电的接线方式：

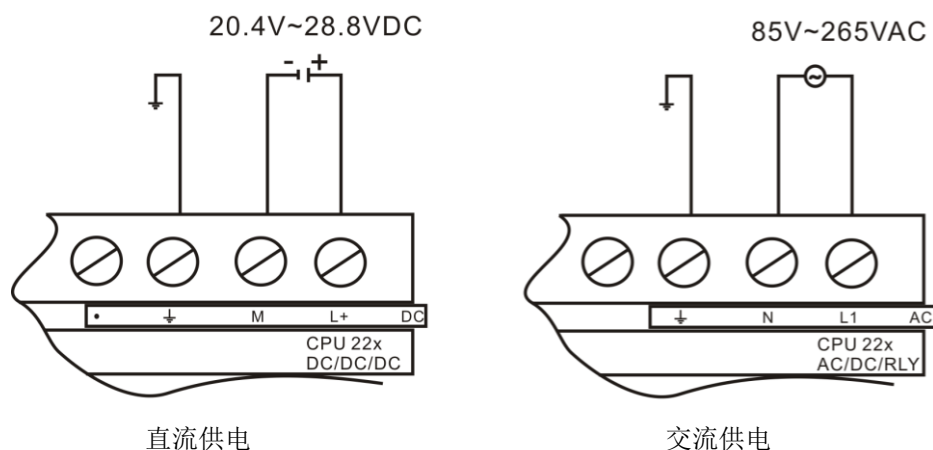


图 2-1 CTH200 电源供电



警告

请勿在带电情况下对 CTH200 系列 PLC 及相关设备进行安装或接线，如错误操作将有可能导致机械毁坏、严重人身伤害甚至人员死亡。在安装和拆除任何电气设备之前，必须确认该设备的电源已断开。

使用 RS485 电缆连接 PLC

图 2-2 所示为使用 RS485 编程电缆连接 CTH200 CPU 与编程设备。



注意

请参考 [L 指令集](#) 安装 PLC 编程电缆驱动。

- 1、编程电缆的 USB 接口连接到编程设备(PC)的通信口上（生成虚拟串口）。

2、编程电缆的 RS485 接口连接到 CTH200 系列 CPU 的 Port0 或 Port1。

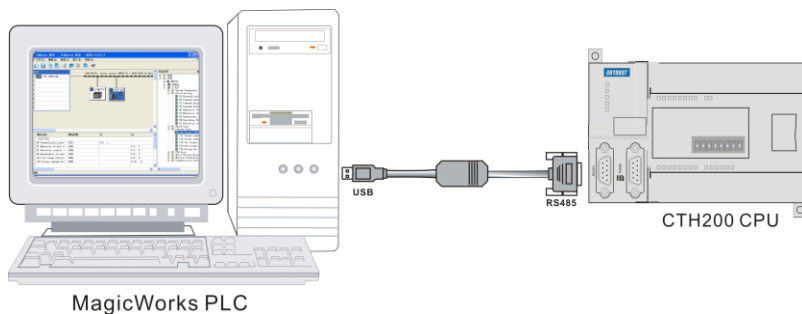


图 2-2 编程电缆连接 CTH200 PLC 与编程设备

使用标准网线连接 PLC

图 2-3 所示为使用标准网线连接 CTH200 CPU 与编程设备。

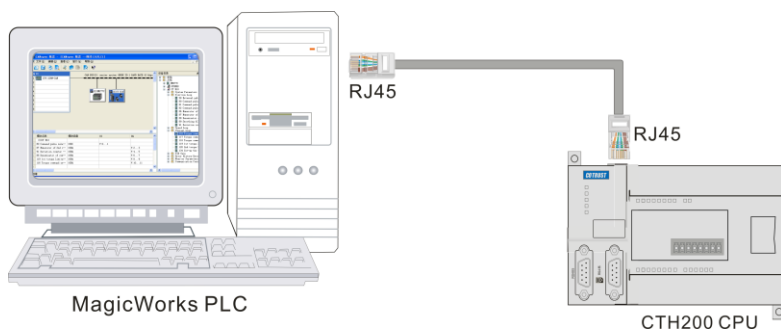


图 2-3 标准网线连接 CTH200 PLC 与编程设备

使用 MagicWorks PLC 软件

双击 MagicWorks PLC 图标启动 MagicWorks PLC 软件，选择“文件 -> 新建”即可创建一个新项目。图 2-4 为一个项目管理器界面，单击项目树中的图标可以打开 MagicWorks PLC 项目中的组件。

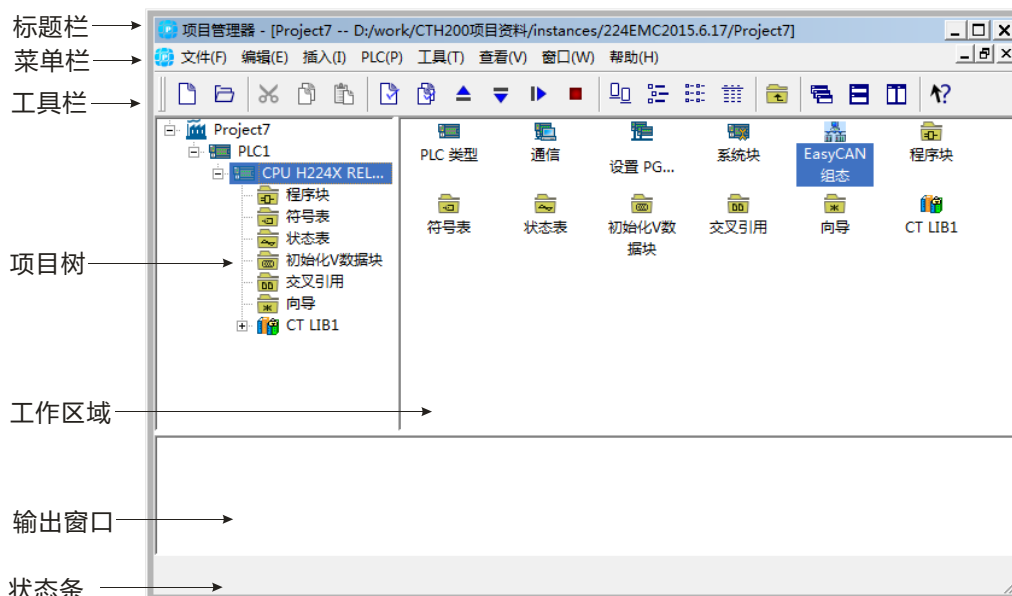


图 2-4 MagicWorks PLC 操作视图

2.2 与 CTH200 建立通信

2.2.1 选择串口通信方式

在“设置 PG/PC 接口”属性对话框中，可以选择并设置 CTH200 支持的串口通信网络，所选的网络将被用作一个接口，MagicWorks PLC 能够访问的默认串口网络通信接口为“PG/PC Cable (PPI)”，具体步骤如下：

- 1) 在通信设置窗口中双击图标或点击“设置 PG/PC 接口”。
- 2) 选择 MagicWorkss 接口选项，默认为“PC/PPI Cable (PPI)”。

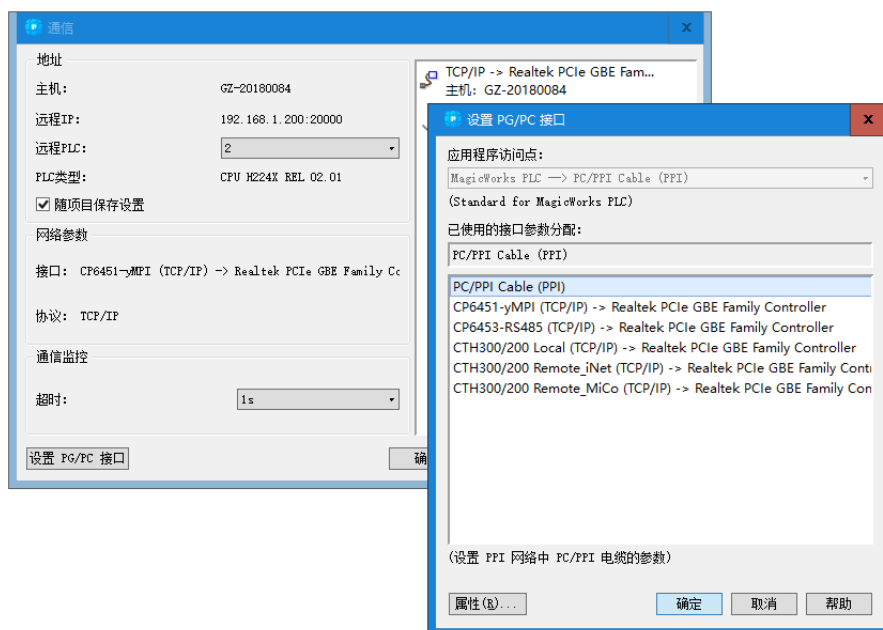


图 2-5 MagicWorks PLC 串口网络通信接口

在“设置 PG/PC 接口”对话框中选择“PC/PPI Cable (PPI)”，然后点击对话框中的“属性”按钮，进行通信参数设置。



- 1) PC/PPI 电缆的通信地址设为 0。
- 2) 接口使用 COM1。
- 3) 传输波特率用 9.6Kbps。

图 2-6 PC/PPI Cable (PPI) 属性设置界面

设置完成后，点击“确定”按钮，返回至通信主界面，然后双击“双击刷新”即可成功搜索到 PLC。

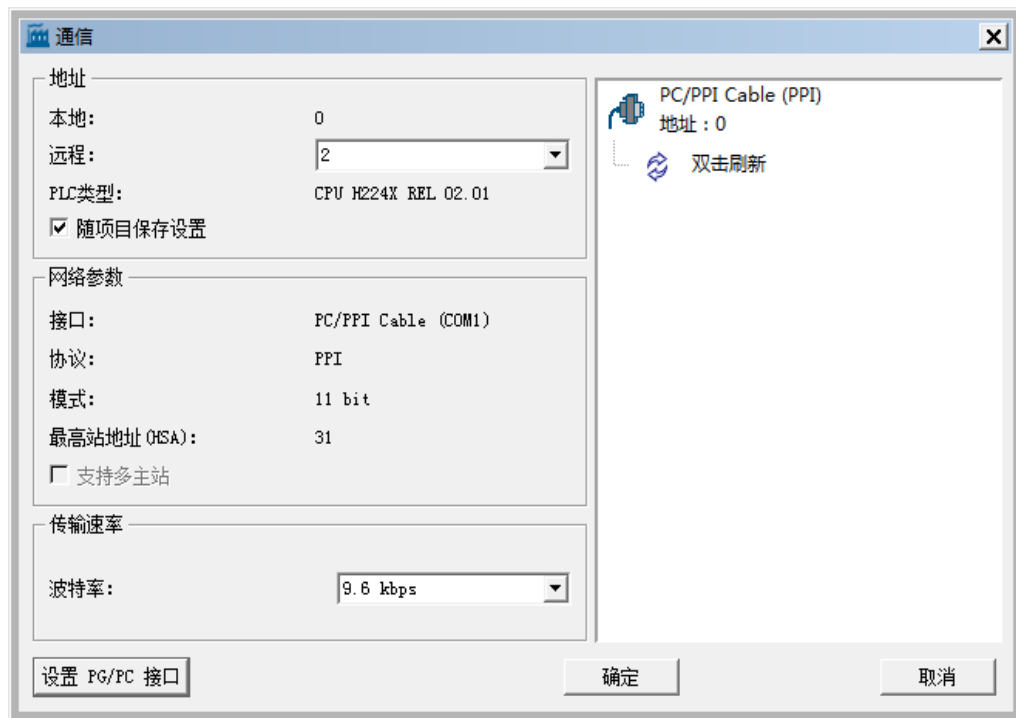


图 2-7 通信主界面

若通信不成功，请按以下步骤排查具体原因：

1) 检查接线

确保接口和电缆完好，请使用合信公司提供的编程电缆（CTS7-191-USB），注意编程电缆接到 PLC 的白色口（如果之前更改过通讯协议，请把 PLC 开关拨到 STOP）。

2) 检查驱动

如选用合信提供的编程电缆，请安装对应版本的驱动，相关驱动安装包下载地址：

<http://www.co-trust.com>

3) 检查通讯设置

- 请选择 PPI 协议(PC/PPI Cable)。
- 请选择对应的 COM 口，确保这个 COM 口未被其他程序占用。
- 选择相应的波特率。

2.2.2 选择以太网通信方式

在“设置 PG/PC 接口”属性对话框中，可以选择并设置 CTH200 支持的串口通信网络，所选的网络将被用作一个接口，可选的网络接口如下：

- CTH300/200 Local(TCP/IP)系列
- CTH00/200 Remote(TCP/IP)系列

以 CTH300/200 Local(TCP/IP)->Realtek PCIe GBE Family Controller 为例，具体操作步骤如下：

(1) 在通信设置窗口中双击图标或点击“设置 PG/PC 接口”

(2) 在列出的网络接口选项中选择“CTH300/200 (TCP/IP) Local-> Realtek PCIe GBE Family Controller”

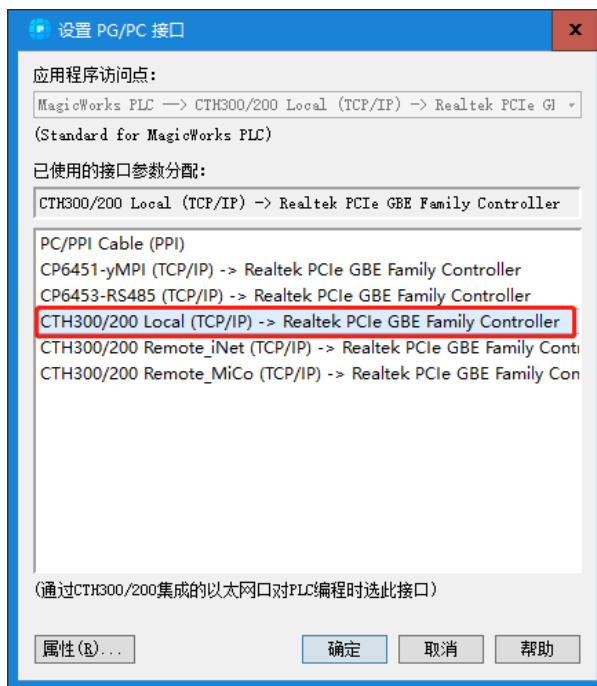


图 2-8 MagicWorks PLC 以太网通信接口

使用通信对话框与 CTH200 建立通信：

- 1) 在通信对话框中双击刷新图标。MagicWorks PLC 搜寻并显示所连接的 CTH200 站的 CPU 图标。

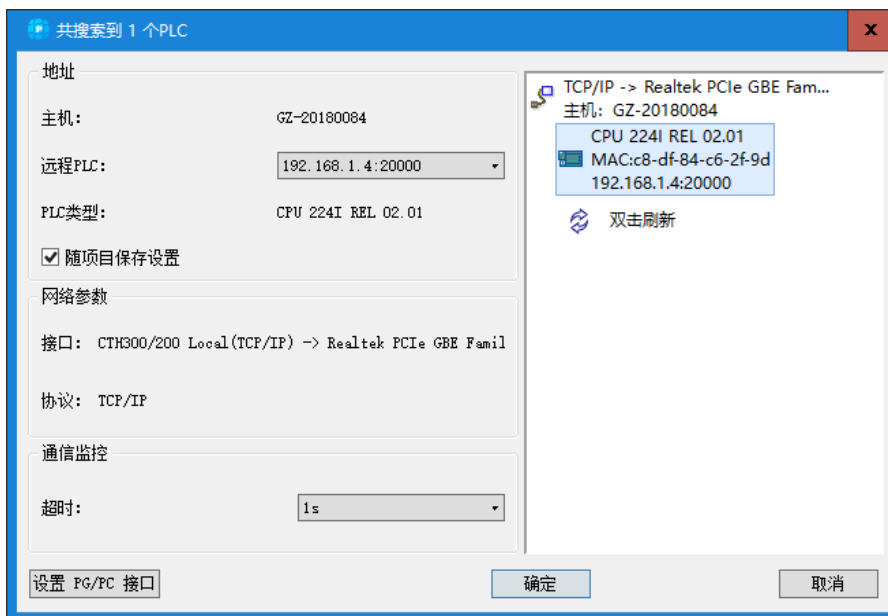


图 2-9 搜索 PLC

- 2) 搜索完成后，选择 CTH200 站并点击确定。如果 MagicWorks PLC 未找到 PLC，请先检查线缆是否连接良好，然后检查通信参数的设置，并重复以上步骤。与 CTH200 建立通信之后，将电脑 IP 设置与 PLC 的 IP 在同一网段，但二者 IP 不能相同，如上图搜索到 PLC 的 IP 为 192.168.1.202，那么电脑 IP 也应为 192.168.1.XXX（其中 XXX 范围 1--255）这样即可编程并下载程序，IP 的具体设置过程如下：

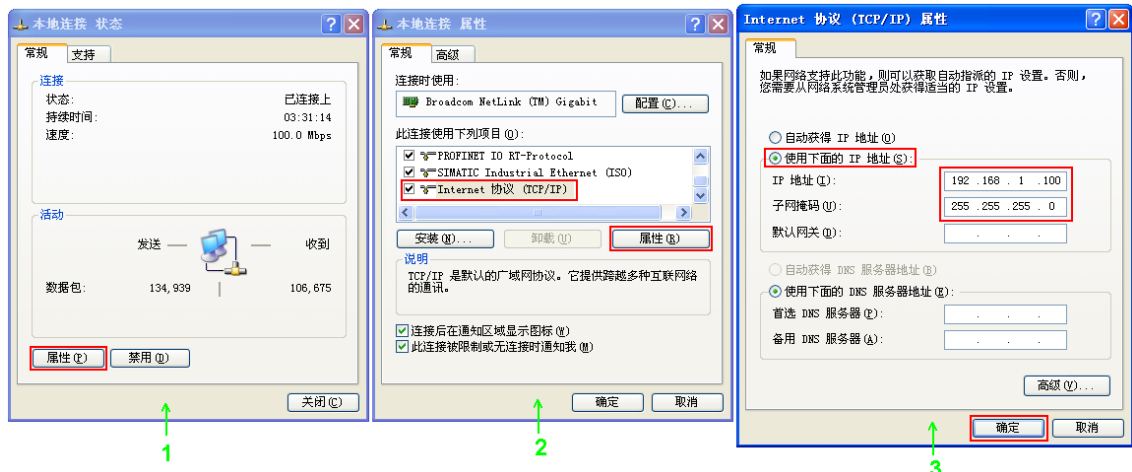


图 2-10 操作系统网络属性设置

2.2.3 在系统块中修改 IP 地址

CTH200 建立通讯后，若需要修改 CPU 的 IP 地址，请打开 MagicWorks PLC 中的系统块，修改其中的 IP 地址，修改后在下载界面选择下载系统块后即可生效，IP 修改成功后需要重新搜索 CPU 建立连接（参考章节 2.2.2 选择以太网通信方式）

修改 IP 如下图所示：

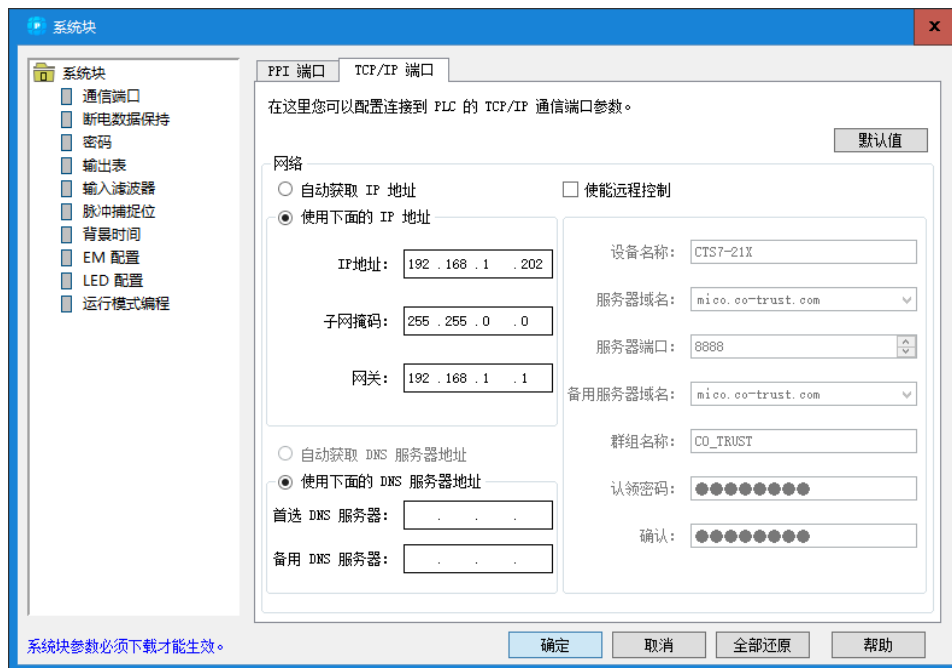


图 2-11 系统块 IP 设置界面



提示

若自动获取 IP 不成功，系统将为 PLC 分配一个默认 IP：192.168.1.202。

2.3 创建程序段

本节举例说明一个 PLC 程序的创建、下载、和运行过程，帮助您了解并使用 CTH200。以下程序

示例在一个程序段中用了 3 条指令，其功能是将传输到 QB0 中的数值字节以 1s 时间间隔左移一位。本例中使用梯形图编辑器来录入程序，并介绍了程序编译、下载和运行过程。

下面是梯形图和语句表程序，语句表中列出的各段程序注释，详细解释了程序的逻辑关系。

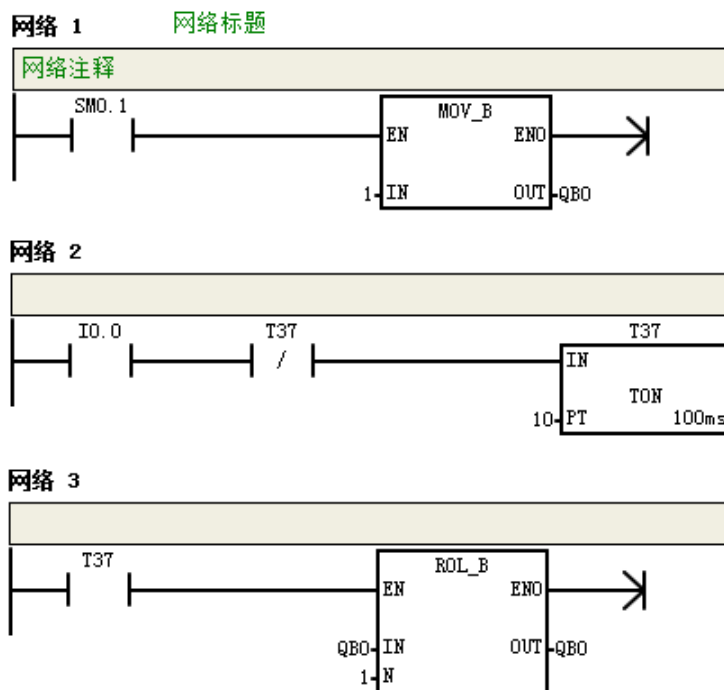


图 2-12 CTH200 程序示例

程序段注释：

```

网络 1:
LD SM0.1 // 通过 SM0.1 首次扫描使能
MOVB 1, QB0 // 将 1 移送到 QB0 内
网络 2:
LD IO.0 // 使能信号
AN T37 // 定时器信号循环
TON T37, 10 // 设定定时器 T37, 计时时间为 100ms x 10 = 1s
网络 3:
LD T37 // 定时器给定使能脉冲
RLB QB0, 1 // QB0 左移一位, PLC 表现为各输出点指示灯每隔一秒循环点亮

```

2.3.1 编辑程序

在 Magicworks PLC (V2.08 以上版本) 中选定 PLC 并设定相关通信参数，然后单击程序块图标打开程序编辑器，见图 2-13。

您可以用拖拽或双击的方式将指令树中的梯形图指令插入到程序编辑器中。在工具栏图标中有一些命令的快捷方式。编写和编译程序之后，即可下载程序到 CTH200 系列 PLC 中。

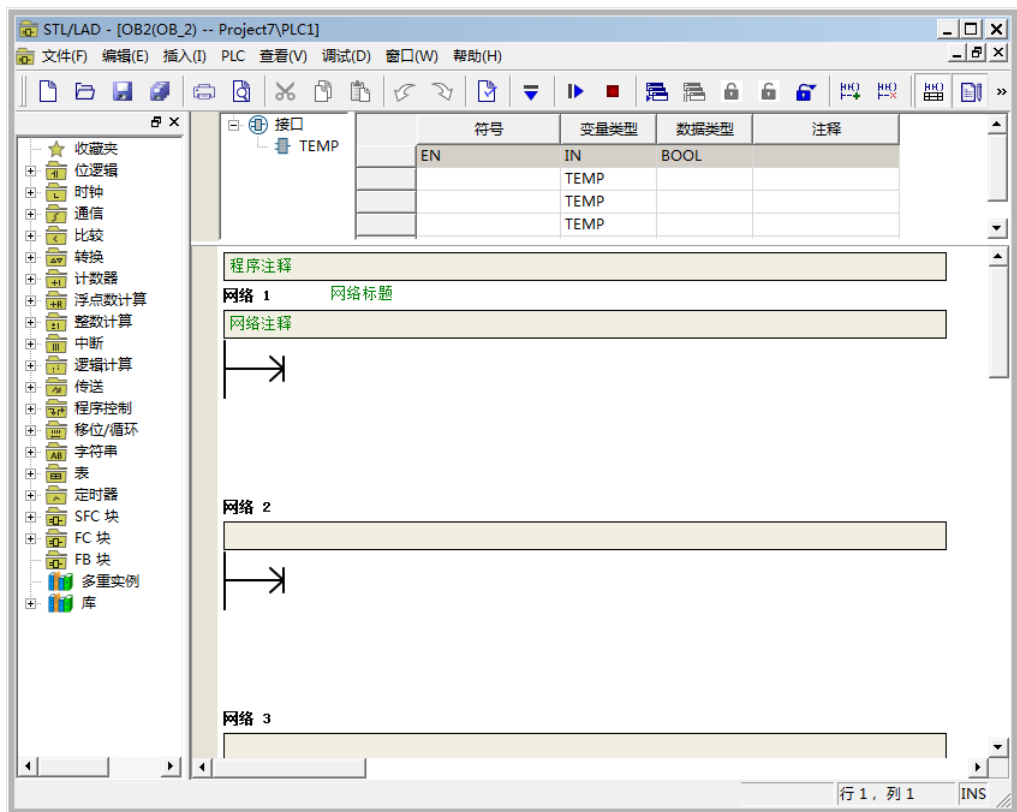



图 2-13 程序编辑器

2.3.2 编译程序

程序段输入完成后，需要编译程序：

- 1) 选择菜单命令“PLC -> 编译”或点击“编译”按钮
- 2) 底部的输出窗口将显示编译状态，如提示编译有误，请点击窗口提示查看具体详情，如编译无误，窗口显示如下：

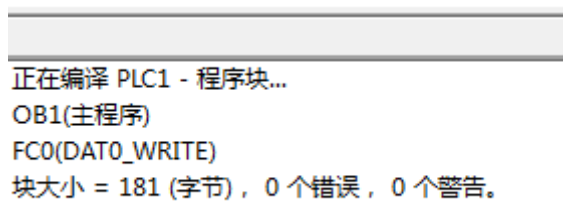


图 2-14 程序编译结果窗口

保存项目：

- 1) 在菜单条中选择菜单命令“文件 -> 另存为”。
- 2) 在“命名”对话框中输入项目名。
- 3) 选择存储位置，即可下载程序到 CTH200 系列 PLC。

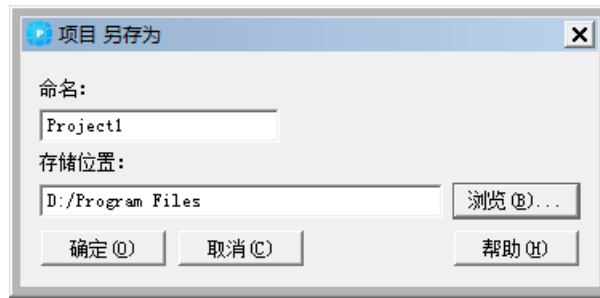


图 2-15 存储程序

2.3.3 下载程序


点击工具条中的下载图标或者在命令菜单中选择“PLC -> 下载”可以下载程序。如图 2-16 和 2-17。点击“确定”下载程序到 CTH200。CTH200 可以在线执行读写操作，即使处于运行模式，也可以将指令或参数值持续写入其中，并不需要将 CTH200 置于停止模式。



图 2-16 串口通信下载程序




图2-17 以太网通信下载程序

2.3.4 运行 PLC

CTH200 系列 PLC 的模式开关设置为 RUN 时，MagicWorks PLC 软件才可以将 CTH200 系列 PLC

转入 RUN 模式。当 CTH200 处于 RUN 模式时，执行程序：

- 1) 单击工具条中的运行图标  或者在命令菜单中选择 PLC -> RUN。
- 2) 点击“是”进入运行模式。当 CTH200 转入运行模式后，CPU 将执行程序，Q0.0-Q0.7 将从左到右以 1s 间隔分别循环点亮，表示正在对 Q 内存中的数值执行循环左移动作。

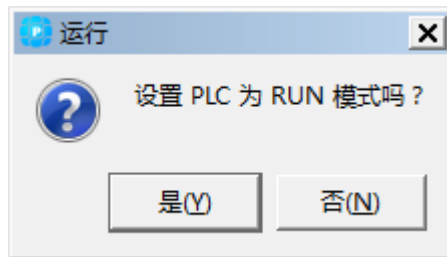

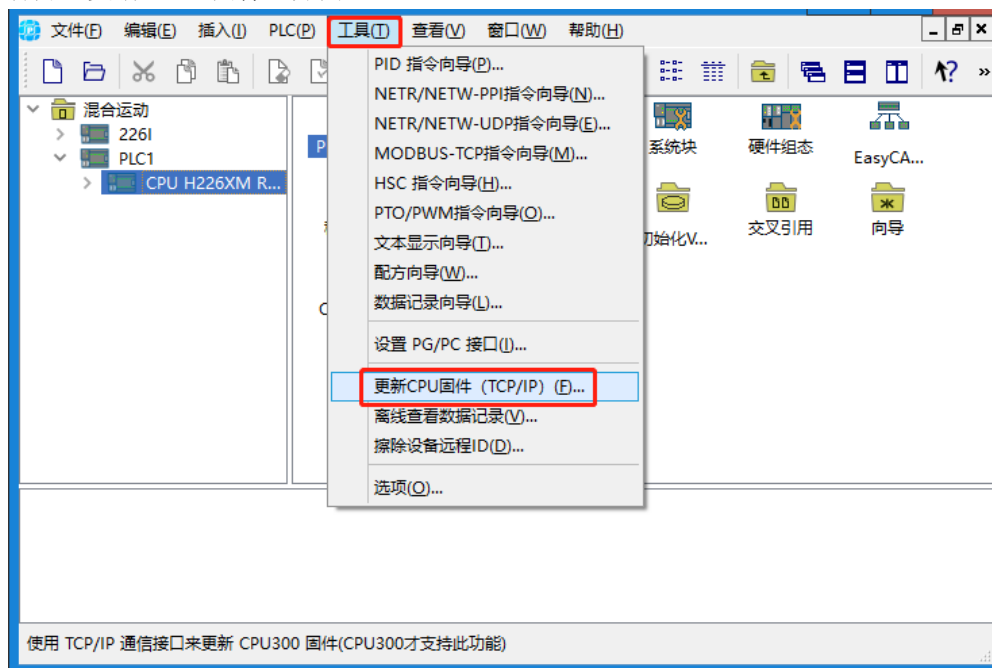


图 2-18 运行程序

您可以通过选择项目树->状态表来监控程序。MagicWorks PLC 显示执行结果。要想终止程序运行，可以单击图标  或选择菜单命令“PLC -> 停止”将 CTH200 置于停止模式。

2.4 CPU 在线升级固件

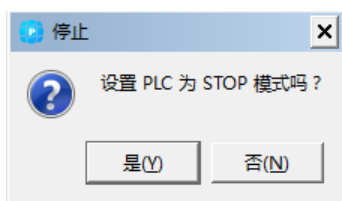
高性能升级版型 CTH200 系列 PLC 支持在线升级固件，进行在线更新固件前，需保证 CPU 已连入网络，如下图所示，在编程软件项目管理器界面选择“工具”→“更新 CPU 固件（TCP/IP）”打开“更新 CPU 固件”界面。



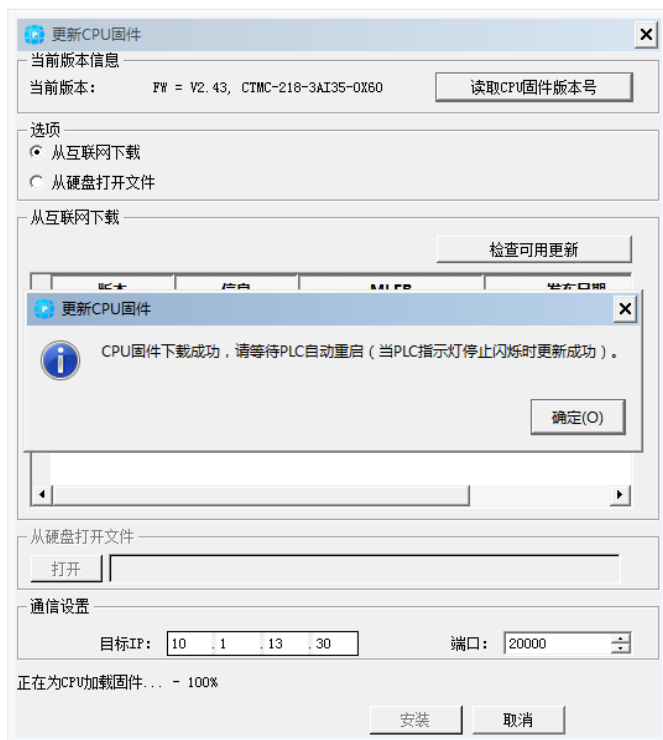
在“更新 CPU 固件”界面选择“从互联网下载”，点击“检查可用更新”，信息窗口将显示当前 PLC 可选用的固件，如下图所示，选中需更新的固件点击“安装”，固件将自动从互联网上下载。



下载完成后将出现如下弹窗，选择“是”，然后 CPU 将自动加载固件。



固件加载完成后，将出现以下提示，点击“确定”。注意，PLC 自动重启完成前切勿对 PLC 进行操作或断开电源！



3 安装

CTH200 系列 PLC 紧凑的外形结构为您有效地分配空间，便于安装。本章将提供 CTH200 系列 PLC 的安装和接线指导。

3.1 安装注意事项

CTH200 系列 PLC 既可以安装在控制柜背板上，也可以安装在标准 DIN 导轨上；兼具水平安装和垂直安装的特点。安装须注意事项如下：

将 CTH200 与加热装置、高电压和电子噪声隔离开

按照一般惯例，在安装设备器件时，总是把产生高电压和高电子噪声的设备与诸如 CTH200 系列 PLC 这样的低压电子型设备分隔开。

在控制柜的背板上排布 CTH200 时，应考虑把电子器件安排在控制柜中温度较低的区域。电子器件长期在高温环境下工作会缩短其使用寿命。

要考虑控制柜的背板布线，尽量避免把交流供电线、高能量、开关频率很高的直流信号线与低压信号线、通信电缆设计在同一个线槽中。

为散热和接线留出适当的空间

CTH200 系列 PLC 的设计采用自然通风散热，在模块的上下方都必须留有至少 30mm 的空间以便正常散热。前面板与背板的板间距离至少应保持 80mm。



注意

在垂直安装时，其允许的最高环境温度要比水平安装时低 10°C，而且 CPU 应安装在所有扩展模块的下方。

导线尽量短并且保证线粗能够满足电流要求。端子排适合的线粗为 2mm²到 0.3mm²（14AWG 到 22AWG），使用屏蔽电缆可以得到最佳的抗电子噪声特性，通常将屏蔽层接地可以得到最佳效果。

大多数的 CTH200 模块有可拆卸的端子排。为了防止连接松动，要确保端子排插接牢固，同时也要确保导线牢固地连接在端子排上。为了避免损坏端子排，螺钉不要拧得太紧。螺钉连接的最大扭矩为 0.56N·m（5 inch-pounds）。

在安装 CTH200 系列 PLC 时，应留出足够空间用于接线和连接通信电缆。

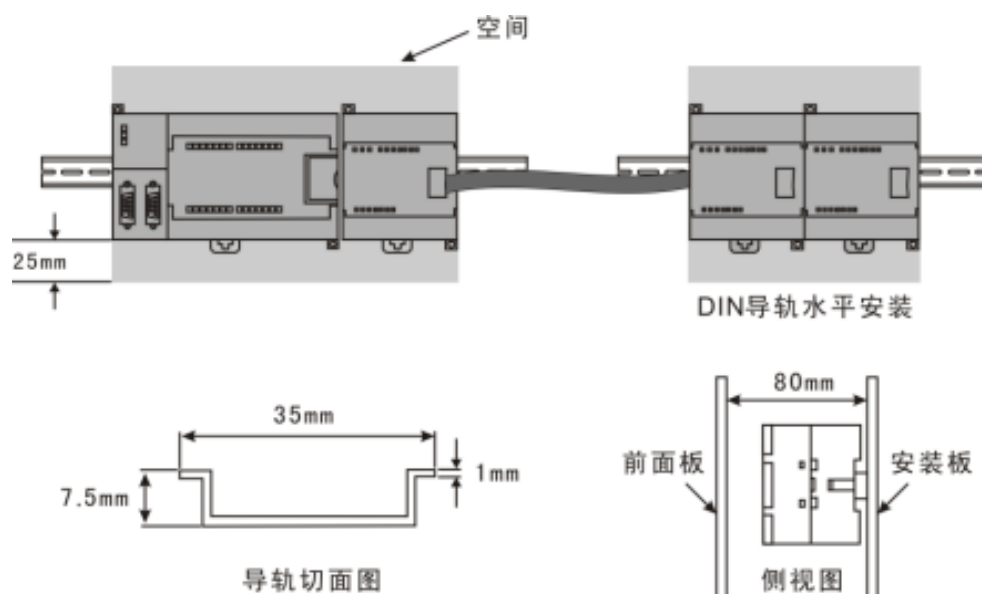


图 3-1 安装尺寸图

电源预算

所有的 CTH200 系列 PLC 都有一个内部电源，为 CPU 自身、扩展模块和其它用电设备提供 24V 直流电源。

CTH200 CPU 为系统中的所有扩展模块提供 5V 直流逻辑电源。您必须特别注意系统配置以确保 CPU 所提供的 5VDC 电源能够满足您所选择的所有扩展模块的需要。若您的配置要求超出 CPU 的供电能力，则需要去掉一些模块或者选择一个供电能力更强的 CPU。

CTH200 系列大多数 CPU 模块也提供 24V 直流传感器供电，此 24VDC 可以为输入点、扩展模块上的继电器线圈或者为其它设备供电。如果设备用电量超过了传感器供电预算，必须为系统另配一个外部 24VDC 供电电源。更多有关电源预算详细内容请参见章节 7 电源预算。

如果您使用了外部 24VDC 供电电源，要确保该电源没有与 CTH200 CPU 上的传感器电源并联使用。为了加强电子噪音保护，建议将不同电源的公共端（M）连在一起。



警告

将外部 24V 电源与 CTH200 CPU 的 24V 传感器供电电源并联，会造成两路供电之间的冲突，每一路电源都试图建立自己的输出电位。这种冲突的结果会缩短电源寿命，或者一路或多路电源立即损坏，使得 PLC 系统产生一系列不确定操作。这种不确定的操作会造成设备损坏或严重的人身伤害甚至人员死亡。

CTH200 系列 CPU 的 24VDC 传感器供电不应和任何外部供电电源同时向同个点供电。

3.2 安装尺寸说明

CTH200 系列 PLC 和扩展模块都有安装孔，可以很方便地安装在背板上，以下是各个 PLC 及扩展模块的安装尺寸图和详细尺寸列表。

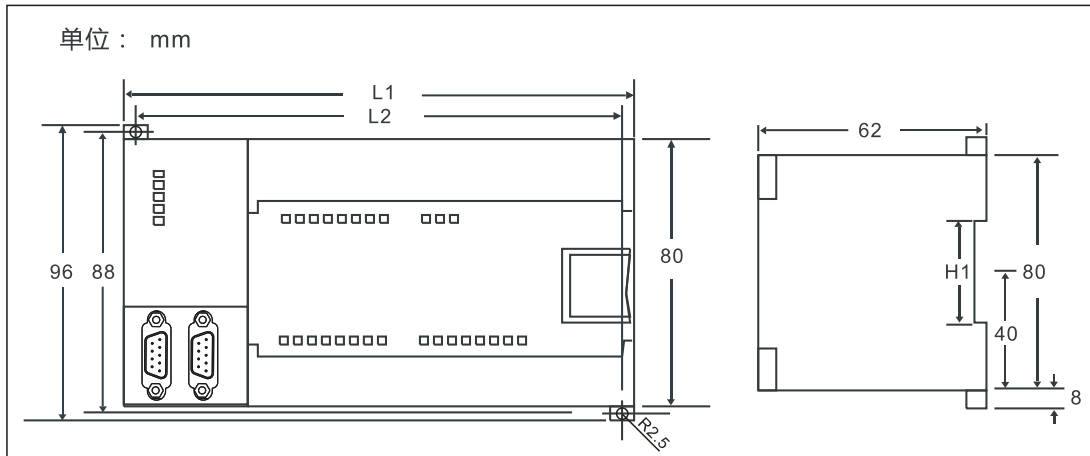


图 3-2 CTH200 系列 PLC 安装尺寸示意图

表 3-1 CTH200 系列 PLC 及扩展模块尺寸

PLC 型号	L1(mm)	L2(mm)
CPU H224/H224X/H224X/H226M/H226IM/ H226IH/H226XM	137	129
CPU H226L/ H226IL/H226XL/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、 CTH2 222-1DF32	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、CTH2 277-0PN32	71.3	62.3
CTH2 277-0AC32	90	82

3.3 安装方法

CTH200 系列 PLC 可以安装在标准的 DIN35 导轨上，或者安装在面板上。

先决条件

在安装和拆卸 CTH200 系列 PLC 及其相关设备之前，要确保 PLC 及与其相连设备的供电均已被切断。



警告

若未切断所有电源而在带电情况下安装或拆卸 CTH200 系列 PLC 及其相关设备有可能导致电击或是设备误动作，从而造成设备损坏或是严重的人身伤害甚至人员死亡。

在更换或安装 CTH200 系列 PLC 时，要确定使用了正确或等同的模块。在更换 CTH200 器件时，除了要使用相同的模块外，还要确保安装的方向和位置是正确的。



注意

如果您安装了不正确的模块，CTH200 系列 PLC 的程序可能会产生错误的功能。

如果未能使用相同的模块按照相同的方向和顺序替换 CTH200 器件，有可能导致设备损坏或严重的人身伤害甚至人员死亡。

CPU 和扩展模块的安装与拆卸

请按照以下方法安装或拆卸 CTH200

• 安装面板

- 1) 按照图 3-1 的尺寸要求定位打孔；
- 2) 用合适的螺钉将模块固定在背板上；
- 3) 如果使用了扩展模块，将扩展模块的扁平电缆连到前盖下面的扩展口。

• 安装 DIN 导轨

- 1) 将导轨固定在背板上，保持间距 80mm。
- 2) 打开模块底部的 DIN 夹子，将模块背部卡在 DIN 导轨上。
- 3) 如果使用了扩展模块，将扩展模块的扁平电缆连到前盖下面的扩展口。
- 4) 旋转模块贴近 DIN 导轨，合上 DIN 夹子。
- 5) 仔细检查模块上 DIN 夹子与 DIN 导轨是否紧密契合。
- 6) 为避免模块损坏，不要直接按压模块正面，可以按压安装孔部分

• 安装端子排

- 1) 打开端子排安装位置的上盖。
- 2) 确保模块上的插针与端子排边缘的小孔对正。
- 3) 将端子排向下压入模块。确保端子块对准了位置并锁住。



注意

当 CTH200 在震动较大的环境下使用或者采用垂直安装方式时，应该使用 DIN 导轨挡块。如果系统处于高震动环境中，使用背板安装方式可以得到较高的震动保护等级。

• 拆卸 CPU 或者扩展模块

按照以下步骤拆卸 CPU 或者扩展模块：

- 1) 拆除 CTH200 的电源。
- 2) 拆除模块上的所有连线 and 电缆。
- 3) 如果有其它扩展模块连接在您所拆卸的模块上，请打开前盖，拔掉相邻模块的扩展扁平电缆。
- 4) 拆掉安装螺钉或者打开 DIN 夹子。
- 5) 拆下模块，拆卸和安装端子排。

• **拆卸端子排**

- 1) 打开端子排安装位置的上盖。
- 2) 把螺丝刀插入端子块中央的槽口中。
- 3) 用力下压并撬出端子排。

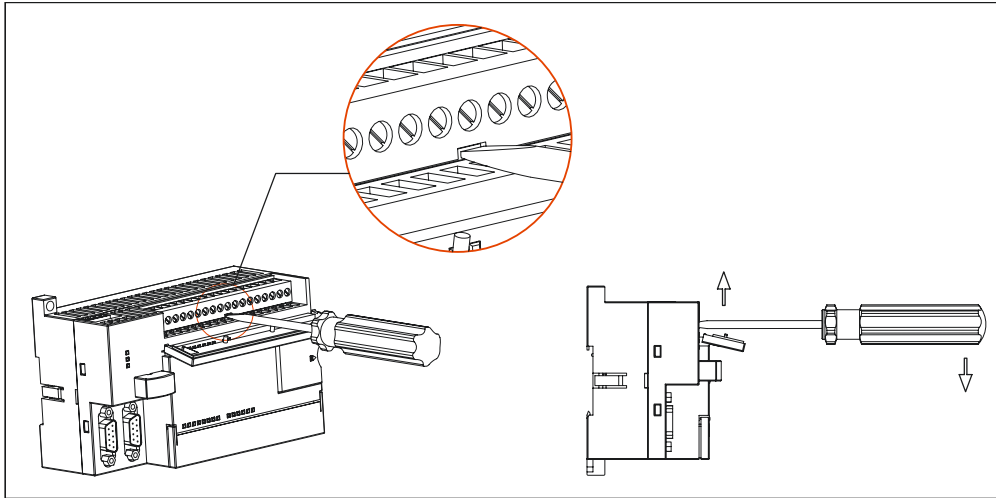


图 3-3 端子排拆卸

3.4 接地和布线

对所有的电器设备合理的接地和接线是非常重要的，它能够确保您的系统具备最优的操作特性，同时能够为您的系统提供更好的电子噪声保护。

在接地和接线之前，必须先确保设备的电源已被切断，也要保证与该设备相关的设备电源已被切断。在对 CTH200 系列 PLC 及其相关设备接线时，必须确保遵从所有有效的电气编码规则。安装和操作所有设备要符合所有有效的国家或地区标准。同地区的权威保持联系，以确定哪些标准符合您的特殊需要。



警告

试图在带电情况下进行接地或接线，有可能造成设备损坏或严重的人身伤害甚至死亡。

在设计 CTH200 系列 PLC 的接地和接线时必须考虑安全因素，否则有可能造成设备的误动作。因此，您应该遵行所有的安全规定以避免人员伤害和设备损坏。



警告

控制设备有可能造成它所控制设备的误操作。这种误操作有可能导致死亡或者严重的人身伤害和设备损坏。因此 CTH200 系列 PLC 中必须具备独立于本机的急停功能、机电互锁或者其它冗余的安全措施。

3.5 抑制电路

在使用感性负载时，需加入抑制电路来限制输出关断时电压上升。抑制电路可以保护输出点不因高感抗开关电流而过早损坏。另外，抑制电路还可以限制感性负载开关产生的电噪声。



注意

抑制电路的有效性取决于应用，您应该调整其参数以适应您的特殊应用。要确保所有器件参数与实际应用相符合。

晶体管输出和控制直流负载的继电器输出

晶体管输出有内部保护，可以适应多种应用。由于继电器型输出既可以连接直流负载，又可以连接交流负载，因而没有内部保护。

图 3-4 给出了直流负载抑制电路的一个实例。在大多数的应用中，用附加的二极管 A 即可，但如果您的应用中要求更快的关断速度，则推荐您加上齐纳二极管 B。确保齐纳二极管能够满足输出电路的电流要求。

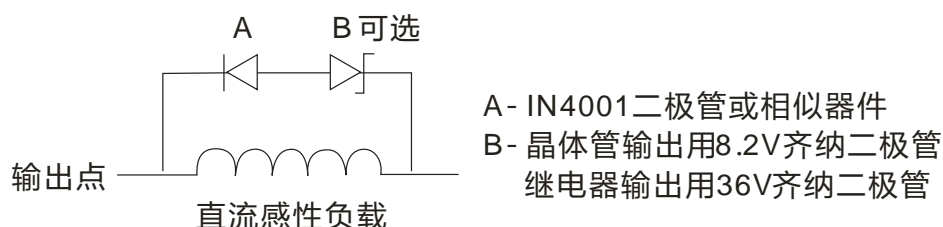


图 3-4 直流负载的抑制电路

交流输出和控制交流负载的继电器输出

交流输出对大部分应用都有内部保护，因为继电器可用于 DC 或 AC 负载，所以不提供内部保护。

图 3-5 给出了交流负载抑制电路的一个实例。在大多数的应用中，附加的金属氧化物可变电阻 (MOV) 可以限制峰值电压，从而保护 CTH200 系列 PLC 的内部电路。要确保 MOV 的工作电压比正常的线电压至少高出 20%。

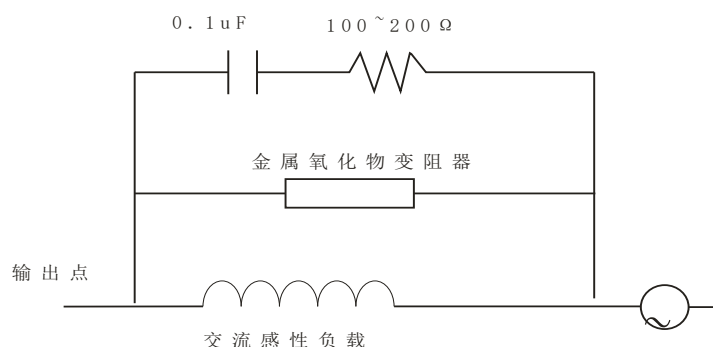


图 3-5 交流负载的抑制电路

4 技术规范

本章介绍 CTH200 系列 CPU 及其扩展模块和扩展板的技术规范。

4.1 通用规范

如下表所示，CTH200 系列 PLC 的各项物理性能经过各项测试具有以下规范：

表 4-1-1 CTH200 系列 PLC 各项物理性能规范

项目	参数
额定电源输入	AC: 220V, DC: 24V
电压范围	AC: 85 V~265V, DC: 20.4 ~ 28.8V
运行环境温度	0 ~ 55°C
储存环境温度	-25 ~ 70°C
空气流动性	在设备上下 50.8mm 位置以内有流动空气
湿度	5 ~ 95% (无结露)
抗电气干扰	脉宽 50ns, 重复频率 5kHz, 2,000V 电压峰值
抗振性	频率: 10~57Hz, 幅度: 0.1mm, 频率: 57~150Hz, 加速度: 1.0g, 三维方向各 10 次
抗冲击性	15g, 持续 11ms, 3 维方向冲击 3 次
耐高压绝缘性	端对地 2200VDC, I/O 接口端对其它端 1500VAC, 持续 1 分钟
接地方式	3 类接地 (多分支单点接地)
运行环境	防尘、非腐蚀性环境
倒下	100mm, 4 次落体, 未包装
自由落下	1 米, 5 次, 运输包装

4.2 CPU 规范和接线图

CTH200 系列 CPU 的各项功能规范如下文所述，请遵照需要进行选型。

4.2.1 CPU 规范

本节介绍 CTH200 系列 CPU 的各项规范和特性

表 4-2-1 CTH200 系列 CPU 性能规范

功能名称	描述
尺寸 (W×H×D)	137×80×62 (24 点) ; 196.5×96×71.3mm (40 点) 200×100×69.5mm (60 点)

用户程序空间	H224/H226L/H226M: 8 (普通) +4KB (ct_lib1) H224X: 12 (普通) +4KB (ct_lib1) H226XL/H226XM: 24 (普通) +48KB (ct_lib1) H228XL: 48 (普通) +24 (ct_lib1) +24KB (ct_lib2) H226IM/H226IL/H226IH: 128 (普通) +64 (ct_lib1) +64KB (ct_lib2)		
数据空间	H224/H226L/H226M: 8KB (普通) H224X: 8 (普通) +100KB (扩展) H226XL/H226XM/H228XL: 10 (普通) +100KB (扩展) H226IM/H226IL/H226IH: 64 (普通) +100KB (扩展)		
模拟电位器	H224X/H226XL/H228XL: 2 个, 8 位分辨率 H224/H226L/H226M/H226XM/H226IM/H226IL/H226IH 无		
位存储区 (M)	256 位		
局部存储区 (L)	64 字节		
顺序控制继电器 (S)	256 位		
脉冲捕捉输入	14, H228XL 为 36, H226IL 为 24		
定时中断	2 个, 1ms 分辨率		
边沿中断	4 个上升沿/4 个下降沿		
实时时钟	内置, 精度±120s/m		
编程卡	支持 (下载工程到 PLC 时, 可选择“覆盖/不覆盖”相应“程序块/数据块”)		
电池卡	支持		
计数器	256 个 (掉电保持)		
最大扩展模块数	7 (基本型 CPU 不支持扩展)		
BD 扩展板数	1 个		
	H224/H226L/H226M	H224X/H226XL/H226XM/H228XL/ H226IM/H226IL/H226IH	
位指令处理速度	0.22μs	0.15μs	
浮点指令处理速度	12μs	8μs	
掉电保持	最大 8KB 永久保持	最大 10KB, 200 小时 其中高性能型升级版 H226IM/H226IL/H226IH 最大 64KB+100KB, 永久保持。	
掉电保持方式	FlashROM	超级电容+锂电池 (高性能型升级版 FlashROM 永久保存)	
数字 I/O 映像区	128DI/128DQ (不含 CAN 通信映像区)	128DI/128DQ (含 CAN 扩展可达 640DI/640DQ)	
模拟 I/O 映像区	32AI/32AQ (不含 CAN 通信映像区)	32AI/32AQ (含 CAN 扩展可达 194AI/194AQ)	
高速计数器	总数	4(不支持 HSC4 和 HSC5)	6
	单相	4x50KHz (仅支持模式 0)	6, 每个 50KHz; 每个 200KHZ(仅 H226IH)
	双相	1x30KHz (仅支持模式 9)	4, 每个 30KHz; 每个 100KHZ(仅 H226IH)

高速计数器输入点		输入点			
模式	中断描述				
	HSC0	I0.0	I0.1	I0.2	
	HSC1	I0.6	I0.7	I1.0	I1.1
	HSC2	I1.2	I1.3	I1.4	I1.5
	HSC3	I0.1			
	HSC4	I0.3	I0.4	I0.5	
	HSC5	I0.4			
0	支持内部方向控制的单相计数器	时钟			
1		时钟		复位	
2		时钟		复位	启动
3	支持外部方向控制的单相计数器	时钟	方向		
4		时钟	方向	复位	
5		时钟	方向	复位	启动
6	带有增减计数时钟的双相计数器	增时钟	减时钟		
7		增时钟	减时钟	复位	
8		增时钟	减时钟	复位	启动
9	A/B 相正交计数器	时钟 A	时钟 B		
10		时钟 A	时钟 B	复位	
11		时钟 A	时钟 B	复位	启动
12	只有 HSC0 和 HSC3 支持模式 12 HSC0 计数 Q0.0 输出脉冲数 HSC3 计数 Q0.1 输出的脉冲数				
高速计数输入电压		18~26V			
高速脉冲输出（晶体管输出）	运动控制输出：3 路 50KHz 输出 Q 点：Q0.0~Q0.2	H224X/H226XL：2 路 100KHz； H226XM：3 路 100KHz （Pulse(Q0.0~Q0.2), Dir(Q0.3~Q0.5)） PTO/PWM：2 路 20KHz； 输出 Q 点：Q0.0~Q0.1 H226IM/H226IL 运动控制输出：3 路 100KHz； 输出 Q 点：Q0.0~Q0.2； PTO/PWM 输出：2 路 20KHz； 输出 Q 点：Q0.0, Q0.1。 H226IH 运动控制输出：4 路 200KHz。 输出 Q 点：Q0.0, Q0.2, Q0.4, Q0.6。			
CAN 扩展板	不支持	支持			
存储卡	不支持	支持			
定时器	总数	256 个	512 个		
	1ms	4 个	4 个		
	10ms	16 个	272 个		
	100ms	236 个	236 个		

表 4-2-2 CPU 电源规范及系统指示灯

功能名称	描述
输入电压范围	85~264VAC (47-63HZ), 20.4~28.8VDC, 有防反接保护
冲击电流	8A @ 264VAC, 6A @ 28.8VDC
隔离 (现场与逻辑)	AC供电: 1500VAC DC供电: 不隔离
保持时间 (掉电)	120/240VAC: 10ms/20ms, 24VDC: 10ms
扩展总线+5V	660Ma
BD 扩展板+5V	200Ma
通信口+5V	10Ma, 100R 电阻
传感器电源 24VDC	<=300Ma, 纹波噪声 (<10MHZ) <1V PP, 晶体管版本决定于输入电源的质量
系统指示灯	SF 指示灯 (红色): ON=系统错误, OFF=正常
	RUN 指示灯 (绿色): ON=运行, OFF=停止
	STOP 指示灯 (橙色): ON=停止, OFF=运行
	RMC 指示灯 (绿色): ON=远程控制允许, OFF=禁止
	LINK/ACT 指示 (绿色): ON=连接, 闪烁=传输, OFF=断开

表 4-2-3 CTH200 系列 PLC 通信功能规范

RS485 通信		
通讯接口数及协议	常规	H224 /H224X: 2个 RS485通讯口, 一个 PPI, 一个自由口 H226L /H226M /H226XL /H226XM /H228XL /H226IM /H226IL /H226IH: 2个 PPI/自由口
	扩展	高性能型升级版: 插上 RS485扩展板并重新启动后, PORT0固定为 PPI0, PORT1为 PPI/自由口, 自由口 FPORT0配置到 RS485扩展板
PPI 波特率	波特率: 9.6Kbps、19.2Kbps、187.5Kbps	
自由口波特率	自由口波特率: 1.2Kbps~115.2Kbps	
每段最大电缆长度	使用隔离中继器: 187.5Kbps 时 1000 米, 38.4Kbps 时 1200 米 不带隔离中继器: 50米	
最大站点数	每段32个站, 每个网络126个站	
最大主站数	32	
物理接口	2个RS485接口	
隔离	不隔离	
以太网通信		
接口类型	一个标准以太网口	
通信标准	符合IEEE802.3标准	
传输速度	10Mbps/100Mbps自适应	
自适应交叉连接	H224X/H226XL/H228XL 不支持, H224/ H226L/ H226M/ H226L/ H226M/H224X /H226XM /H226IM /H226IL /H226IH支持	
工业以太网接口 (10/100Mbps)	RJ45插座	
固件升级	高性能型升级版支持网口在线升级固件功能	
协议类型	支持UDP_PPI, MODBUS_TCP、Socket, S7协议, 支持PLC间以太网通信, 支持MiCo以太网编程。	
配置方式	支持上位机通过网口搜索PLC, 支持网口下载系统配置。	

应用接口	支持 MiCo 服务、UDP_PPI、Modbus_TCP、S7 协议、Socket 编程		
最大连接数	8 个 UDP_PPI，不分主从 8 个 Modbus_TCP，不分主从 4 个 Socket 连接，2 个 UDP，2 个 TCP 8 个 S7 协议，不分主从，合信 PLC 只作为从站		
	H224/H226M/H226L	H224X/H226XL/H228XL	高性能型升级版
最大 IT 连接数	8个UDP_PPI连接，不分主从；5个Modbus_TCP连接，不分主从；4个Socket连接，2个UDP，2个TCP。	8个UDP_PPI连接，4主4从；4个Modbus_TCP连接，2主2从。	8个UDP_PPI连接，不分主从；8个Modbus_TCP连接，不分主从；4个Socket连接，2个UDP，2个TCP；西门子S7协议最多支持8个连接，合信CPU只做从站。
用户数据数量	最多200个字节用于UDP_PPI传输。 最多240个字节用于Modbus_TCP传输。 最多512个字节用于Socket传输 最多200个字节用于S7传输		
端口	本地：1~65535 MiCo远程服务器地址：mico.co-trust.com；端口：8888		
启动时间或复位后重启时间	约3s		
指示灯	RMC 指示灯（绿色）：亮=远程控制生效，灭=远程控制失效		
	以太网指示灯（绿色）：亮=连接，闪烁=传输，灭=未连接		
最大电缆段长度	100m		
隔离	通信口隔离		
电缆类型	以太网：CAT5e屏蔽电缆		

表 4-2-4 CTH200 系列 CPU 数字量输入特性

数字量输入特性			
型号	H224X/H226XM/H226IM/ H226IH	H226L/H226XL/H226XL /H226IL	H228XL
本机集成数字量输入点数	14	24	36
输入类型	漏型/源型		
额定电压	24VDC		
输入电压范围	20.4~28.8VDC		
浪涌电压	35VDC, 持续 0.5s		
逻辑 1 信号 (最小)	15 VDC, 2.5Ma		
逻辑 0 信号 (最大)	5 VDC, 1Ma		
允许的最大漏电流	1Ma		
输入滤波	可配置, H224/H226L/H226M 支持 3.4ms 和 6.4ms; H224X/H226XL/H226XM/H228XL/H226IM/H226IL/H226IH 支持 0.2ms, 0.4ms, 0.8ms, 1.6ms, 3.2ms, 6.4ms, 12.8ms, 默认 为 6.4ms		
隔离 (现场与逻辑) 隔离组	500VAC, 1 分钟 见接线图		
可同时接通的输入	14	24	36
最大电缆长度 屏蔽 非屏蔽	500m (标准输入), 50m (高速计数器输入) 300m (标准输入)		

表 4-2-5 CTH200 系列 CPU 数字量输出特性

输出类型		晶体管		继电器
类型		固态 MOSFET, 源型		干触点
额定电压		24VDC		24VDC 或 110V/220VAC
电压范围		20.4~28.8VDC		5 ~ 30VDC 或 5 ~ 250VAC
浪涌电流 (最大值)		8A, 100ms		5A, 4s@10%占空比
每点额定电流 (最大值)		0.5A		2.0A
每个公共端的额定电流 (最大值)		4A		8A
照明负载 (最大值)		5W		DC 30W, AC 200W
导通电阻 (Ω)		典型 0.3, 最大 0.6		0.2 (新使用时最大值)
隔离 (现场到逻辑)		500VAC, 1 分钟		1500VAC, 1 分钟
		H224/H226M/H226L/ H226IM/H226IL /H226XM	H226IH	H224X/H224X/H226XL/ H226XL
输出延迟 (晶体管)	接通到断开	10μs(Q0.0, Q0.1, Q0.2), 130μs (其它)	0.2us (Q0.0-Q0.7), 50us (其他)	10μs(Q0.0, Q0.1), 130μs (其它)
	断开到接通	2μs(Q0.0, Q0.1, Q0.2), 15μs (其它)	0.2us (Q0.0-Q0.7), 50us (其他)	2μs(Q0.0, Q0.1), 15μs (其它)
使用寿命 (机械寿命)		--		10,000,000 (无负载)
使用寿命 (电气寿命)		100,000 (额定负载 2A)		
同时接通的输出点数		所有输出点		
并行连接两个输出		是, 仅限输出同组时		
最大电缆长 度	屏蔽	500m		
	非屏蔽	其余: 150m		

用户程序需求及数据存储器规范

表 4-2-6 CTH200 系列 CPU 编程规范

功能名称	描述	
基本指令集	CTH200基本指令集	
编程工具	MagicWorks PLC/Step7 MicroWIN (注: 若需使用网口则只能使用MagicWorks PLC进行编程)	
编程接口	RS485/以太网接口, 支持通过以太网口在线升级固件	
在线编程	H224/H226L/H226M不支持; H224X/H226XL/H226XM/H228XL/H226IM/H226IL/H226IH支持	
编程语言	语句表 STL/梯形图 LAD	
POU	类型/数量	主程序: 1 (OB1)
		子程序: 128 (0-127)
		中断程序: 126 (2-127); OB0 预留, OB1 为主程序
	嵌套深度	从主程序: 8 级子程序
从中断程序: 1 级子程序		
累加器	4 个	
内置库	MODBUS RTU 主从站通信库; MODBUS TCP 通信库; PID_T 通信库 (H224/H226L/H226M 支持 16 路; 其他型号支持 64 路)	

表 4-2-7 数据存储区规格

型号	H224/H226L/H226M	H224X/H226XL/H226XM/H228XL/H226IM/H226IL/H226IH	H224/H226L/H226M	H224X/H226XL/H226XM/H228XL/H226IM/H226IL/H226IH
	数字输入映像区 (I)		数字输出映像区 (Q)	
位地址范围	I0.0~I15.7	I0.0~I79.7	Q0.0~Q15.7	Q0.0~Q79.7
字节地址范围	IB0~IB15	IB0~IB79	QB0~QB15	QB0~QB79
字地址范围	IW0~IW14	IW0~IW78	QW0~QW14	QW0~QW78
双字地址范围	ID0~ID12	ID0~ID76	QD0~QD12	QD0~QD76
访问属性	支持立即访问/直接访问/间接访问			
保持属性	不支持掉电保持			
	模拟输入映像区 (AI)		模拟输出映像区 (AQ)	
每通道字长	16 bits			
字地址范围	AIW0~AIW62	AIW0~AIW386	AQW0~AQW62	AQW0~AQW386
访问属性	支持立即访问/直接访问/间接访问			
保持属性	不支持掉电保持			
变量内存区 (V)				
型号	H224/H226L/H226M	H224X	H226XL/H226XM/H228XL/H226IM/H226IL/H226IH	
容量 (字节)	8K	8K(可扩展至108K)	10K(可扩展至110K)	
位地址范围	V0.0~V8191.7	V0.0~V8191.7	V0.0~V10239.7	
字节地址范围	VB0~VB8191	VB0~VB8191	VB0~VB10239	
字地址范围	VW0~VW8190	VW0~VW8190	VW0~VW10238	

双字地址范围	VD0~VD8188	VD0~VD8188	VD0~VD10236
访问属性	支持直接访问/间接访问		
保持属性	可配置为全部或部分掉电保持		
特殊内存区 (SM)			
型号	H224/H226L/H226M	H224X/H226XL/H226XM/H228XL/H226IM/H226IL/H226IH	
容量	550bytes	650bytes	
位地址范围	SM0.0~SM549.7	SM0.0~SM649.7	
字节地址范围	SMB0~SMB549	SMB0~SMB649	
字地址范围	SMW0~SMW548	SMW0~SMW648	
双字地址范围	SMD0~SMD546	SMD0~SMD646	
访问属性	前30字节的访问属性为只读，支持直接访问/间接访问		
特殊寄存器 SM (具体功能说明请参见附录 T 指令集)			
SMB0	系统状态位		
SMB1	指令执行状态位		
SMB2	自由口接收字符		
SMB3	自由口校验错误		
SMB4	中断队列溢出、运行时间程序错误、中断启用、自由口变送器被强制		
SMB5	I/O 状态		
SMB6	CPU 标识寄存器		
SMB8~SMB21	I/O 模块识别和错误寄存器		
SMW22~SMW26	扫描时间		
SMW28	模拟电位器 (仅 H224X/H226XL/H228XL 支持)		
SMB30, SMB130	自由口控制寄存器		
SMB31	永久性内存 (EEPROM) 写入控制		
SMW32	需要保存的 V 内存位置地址		
SMB34	定时中断0的时间间隔数值		
SMB35	定时中断1的时间间隔数值		
SMB36~SMB65	HSC0、HSC1和 HSC2寄存器		
SMB66~SMB85	PTO/PWM 寄存器		
SMB86~SMB94 SMB186~SMB194	自由口接收信息控制		
SMW98	I/O 扩展总线通信错误		
SMB110	用于启动强制上载保护功能		
SMB111	编程卡覆盖选项： 0 — 覆盖程序块和数据块；1 — 不覆盖程序块，覆盖数据块； 2 — 覆盖程序块，不覆盖数据块；3 — 不覆盖程序块和数据块； 4 — 与为0时相同		
SMB112	始终正确显示当前 CPU 版本信息 (SMB112=16#82, 即表示 CTH200系列；SMB112=16#85, 表示高性能型升级版)		
SMB113	始终正确显示当前 CPU 型号		
	16#02	H224/H224X RLY	
	16#03	H224/H224X PNP	
	16#08	H226L/H226XL /H226XM RLY	
	16#09	H226L/H226XL /H226XM PNP	

	16#18	H228XL RLY
SMB114		扩展板模块类型 0x19: 2AI/1AQ (CTH2-AMS-03S1-EB) 0x1E: 4AI/2AQ (CTH2-AMS-06S1-EB/ CTH2-AMS-06S2-EB) 0x8A: RS485 (CTH2-485-01S1-EB) 0x8D: CAN V2 (CTH2-CAN-01S2-EB)
SMB115		扩展板状态 0x0: 模块没有错误 0x255: 扩展板访问错误 其它: 扩展板内部诊断
SMW116~SMW122		扩展板模拟量输入映射区
SMW124~SMW126		扩展板模拟量输出映射区
SMB131		CAN 扩展板访问周期 (毫秒), 设为0时默认为1毫秒
SMB136~SMB165		HSC3、HSC4和 HSC5寄存器
SMB166~SMB185		PTO0, PTO1包络定义表 H224/H226L/H226M 不支持; H224X/H226XL/H226XM/H228XL/H226IM/H226IL/H226IH 支持
SM195		通信状态
SMB200~SMB549		智能模块状态
SMB550~SMB649		用于 CAN 通信
内部内存区 (M)		
容量 (字节)		32 bytes
位地址范围		M0.0~M31.7
字节地址范围		MB0~MB31
字地址范围		MW0~MW30
双字地址范围		MD0~MD28
访问属性		支持直接访问/间接访问
保持属性		可配置为全部或部分掉电保持
局部变量区 (L)		
容量 (字节)		64 bytes
位地址范围		L0.0~L63.7
字节地址范围		LB0~LB63
字地址范围		LW0~LW62
双字地址范围		LD0~LD60
访问属性		支持直接访问
保持属性		开机到停机对同一调用位置子程序保持(限 H224X/H226XL/H226XM /H228XL 支持), 无掉电保持
累加器寄存器 (AC)		
容量		4个
位地址范围		不支持
字节地址范围		AC0~AC3
字地址范围		AC0~AC3
双字地址范围		AC0~AC3
访问属性		支持直接访问

保持属性	不支持掉电保持			
顺序控制继电器(S)				
容量（字节）	32			
位地址范围	S0.0~S31.7			
字节地址范围	SB0~SB31			
字地址范围	SW0~SW30			
双字地址范围	SD0~SD28			
访问属性	支持直接/间接访问			
保持属性	不支持掉电保持			
定时器(T)				
H224/H226L/H226M				
类型	分辨率	数目	编号	最大定时值
有记忆接通延时定时器 TONR	1ms	2	T0, T64	32.767s
	10ms	8	T1~T4, T65~T68	327.67s
	100ms	54	T5~T31, T69~T95	3276.7s
接通延时定时器 TON/关断延时定时器 TOF	1ms	2	T32, T96	32.767s
	10ms	8	T33~T36, T97~T100	327.67s
	100ms	182	T37~T63, T101~T255	3276.7s
H224X/H226XL/H226XM/H228XL/H226IM/H226IL/H226IH				
类型	分辨率	数目	编号	最大定时值
有记忆接通延时定时器 TONR	1ms	2	T0, T64	32.767s
	10ms	8	T1~T4, T65~T68	327.67s
	100ms	54	T5~T31, T69~T95	3276.7s
接通延时定时器 TON/关断延时定时器 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
访问属性	计时寄存器可直接/间接访问，状态位仅可直接访问			
保持属性	可配置当前计时值的掉电保持属性，状态位不能保持			
计数器（C）				
数目	256			
计数方式	向上计数/向下计数/向上向下计数			
最大计数值	32767			
访问属性	计数寄存器可直接/间接访问，状态位仅可直接访问			
保持属性	可配置当前计数值的掉电保持属性，状态位不能保持			

表 4-2-8 CTH200 支持的数据类型

数据类型	大小	说明	取值范围
布尔	1位	布尔值	0~1
字节	8位	无符号字节	0~255
字	16位	无符号整数	0~65535
整数	16位	有符号整数	-32768~+32767
双字	32位	无符号双整数	0~4294967295
双整数	32位	有符号双整数	-2147483648~+2147483647

实数	32位	IEEE 32位浮点数	+1.175495E-38~+3.402823E+38 -1.175495E-38~-3.402823E+38
字符串	1~255字节	ASCII 字符串照原样存储在 PLC 内存中，形式为：1字节字符串长度+ASCII 字符	无

密码级别及权限控制

表 4-2-9 CTH200 密码级别及权限控制

操作说明	权限级别1	权限级别2	权限级别3	权限级别4
读写用户数据	允许	允许	允许	允许
运行/停止/上电复位	允许	允许	允许	允许
读写实时时钟	允许	允许	允许	允许
STOP 模式下写 Q 点	允许	需验证密码	需验证密码	需验证密码
强制数据	允许	需验证密码	需验证密码	需验证密码
上载程序块/数据块/硬件配置块	允许	允许	需验证密码	不允许
下载程序块/数据块/硬件配置块	允许	需验证密码	需验证密码	需验证密码（不允许下载硬件配置块）
清除程序块/数据块/硬件配置块	允许	需验证密码	需验证密码	需验证密码（不允许删除硬件配置块，允许三块一起删除）
运行时编辑	允许	需验证密码	需验证密码	不允许
首次或多次扫描	允许	需验证密码	需验证密码	需验证密码
刷新扫描周期	允许	需验证密码	需验证密码	需验证密码
项目比较	允许	允许	需验证密码	不允许
程序状态监控（时间戳比较通过）	允许	允许	允许	允许
程序状态监控（时间戳比较不通过）	允许	需验证密码	需验证密码	不允许

实时时钟及事件

表 4-2-10 CTH200 实时时钟功能

出厂设置	未设置，固定值90年1月1日00:00:00 星期日	
掉电保持时间	约100小时（25℃典型值）	
精度	每月偏差<120秒	
读取时钟功能	通过指令 TODR/TODRX 或通过软件读取	
设置时钟功能	通过指令 TODW/TODWX 或通过软件设置	
常规时钟格式（8字节）		
T 字节	说明	字节数据
0	年（0-99）	当前年份（BCD 值）
1	月（1-12）	当前月份（BCD 值）
2	日期（1-31）	当前日期（BCD 值）
3	小时（0-23）	当前小时（BCD 值）
4	分钟（0-59）	当前分钟（BCD 值）
5	秒（0-59）	当前秒（BCD 值）
6	0	保留，始终设置为00
7	星期几（1-7）	当前是星期几，1=星期日（BCD 值）
扩展时钟格式（19字节）		
0	年（0-99）	当前年份（BCD 值）
1	月（1-12）	当前月份（BCD 值）
2	日期（1-31）	当前日期（BCD 值）
3	小时（0-23）	当前小时（BCD 值）
4	分钟（0-59）	当前分钟（BCD 值）
5	秒（0-59）	当前秒（BCD 值）
6	0	保留，始终设置为00
7	星期几（1-7）	当前是星期几，1=星期日（BCD 值）
8	时区	00H-03H, 08H, 10H-13H, FFH
9	修正小时数（0-23）	修正数量，小时（BCD 值）
10	修正分钟数（0-59）	修正数量，分钟（BCD 值）
11	开始月份（1-12）	夏时制的开始月份（BCD 值）
12	开始日期（1-31）	夏时制的开始日期（BCD 值）
13	开始小时（0-23）	夏时制的开始小时（BCD 值）
14	开始分钟（0-59）	夏时制的开始分钟（BCD 值）
15	结束月份（1-12）	夏时制的结束月份（BCD 值）
16	结束日期（1-31）	夏时制的结束日期（BCD 值）
17	结束小时（0-23）	夏时制的结束小时（BCD 值）
18	结束分钟（0-59）	夏时制的结束分钟（BCD 值）

下表列出了 CTH200 支持的中断事件，供您参考。

表 4-2-11 CTH200 支持的中断事件

优先级别群组	中断事件号码	优先级别组	说明
通讯和诊断事件 (最高优先级)	8	0	端口0: 接收字符
	9	0	端口0: 传输完成
	23	0	端口0: 接收信息完成
	24	0	端口1: 接收信息完成
	25	0	端口1: 接收字符
	26	0	端口1: 传输完成
	36	0	模块诊断事件中断
IO 中断及高速计数器中断 (中等优先级)	0	1	上升边缘, I0.0
	2	1	上升边缘, I0.1
	4	1	上升边缘, I0.2
	6	1	上升边缘, I0.3
	1	1	下降边缘, I0.0
	3	1	下降边缘, I0.1
	5	1	下降边缘, I0.2
	7	1	下降边缘, I0.3
	12	1	HSC0 CV=PV
	27	1	HSC0方向改变
	28	1	HSC0外部复原 / Zphase
	13	1	HSC1 CV=PV
	14	1	HSC1方向改变
	15	1	HSC1外部复原
	16	1	HSC2 CV=PV
	17	1	HSC2方向改变
	18	1	HSC2外部复原
	19	1	PTO 0 完成中断
	20	1	PTO 1 完成中断
	32	1	HSC3 CV=PV
29	1	HSC4 CV=P	
30	1	HSC4方向改变	
31	1	HSC4外部复原 / Z phase	
33	1	HSC5 CV=PV	
定时 (最低优先级)	10	2	定时中断0
	11	2	定时中断1
	21	2	定时器 T32 CT=PT 中断
	22	2	定时器 T96 CT=PT 中断

4.2.2 CPU 结构示意图和接线图

CTH200 系列 CPU 及相关配件与模块的接线方式有所不同，本节列出各型号 CPU 的具体接线图。在 CTH200 系列 PLC 中，为方便起见，H224/H226XM/H226M/H224X/H226IM 与 H226IH，H226L/H226IL 与 H226XL 的外观结构一致（H224X/H226XL/H228XL 有模拟电位器），结构示意图如下：

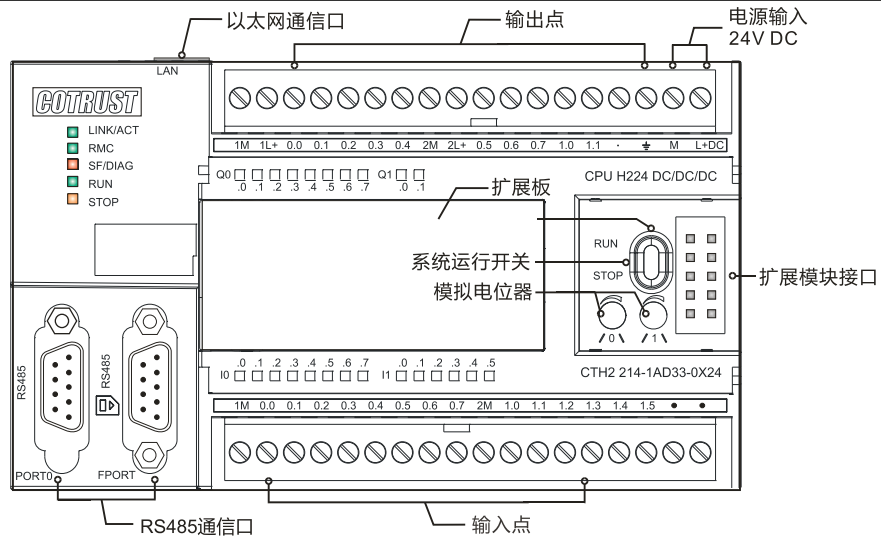


图 4-1 CTH2 214-1AD33-0X24 结构示意图

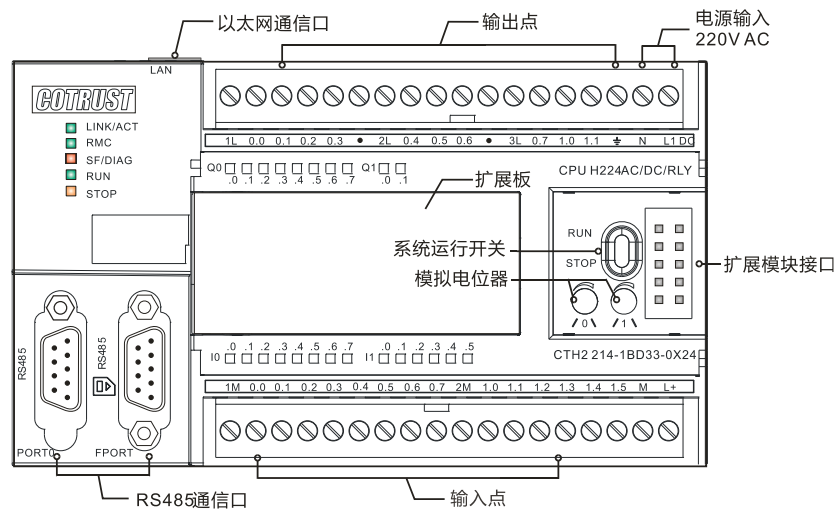


图 4-2 CTH2 214-1BD33-0X24 结构示意图

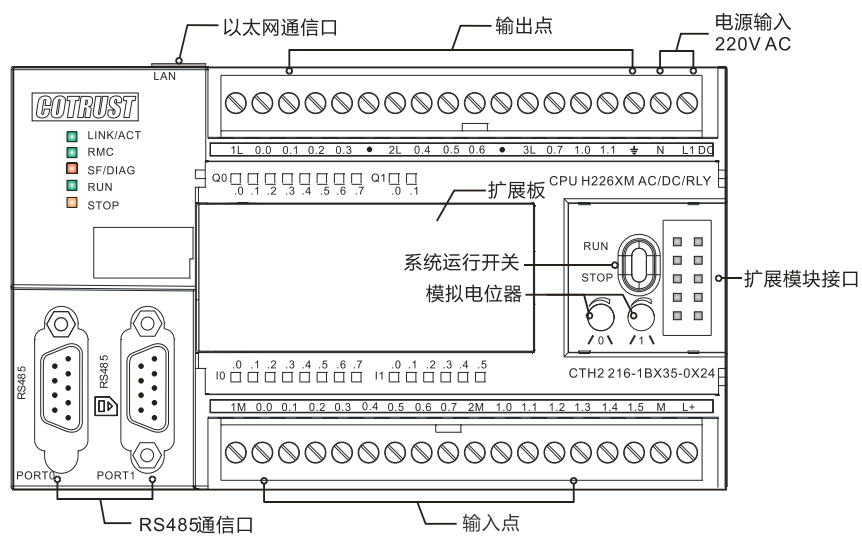


图 4-3 CTH2 216-1BX35-0X24 结构示意图

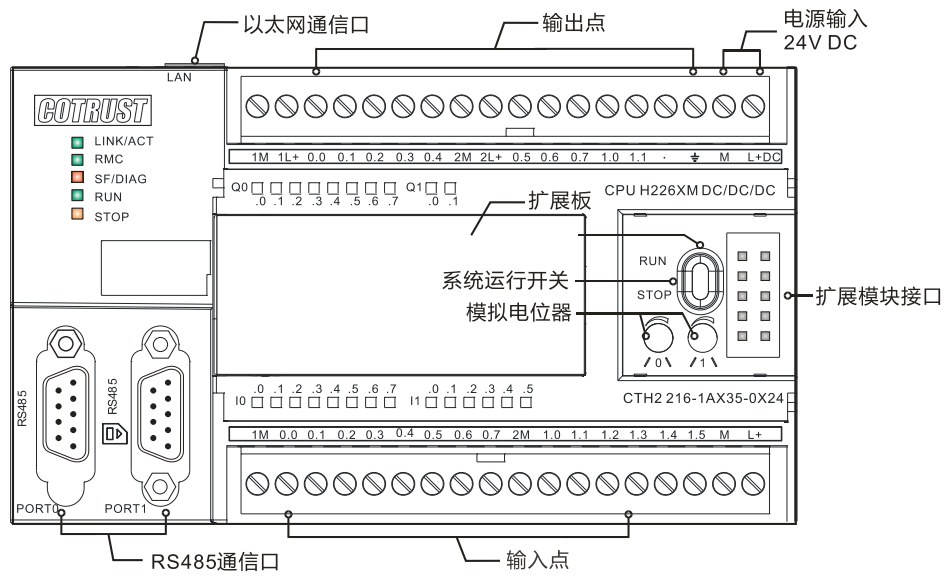


图 4-4 CTH2 216-1AX35-0X24 结构示意图

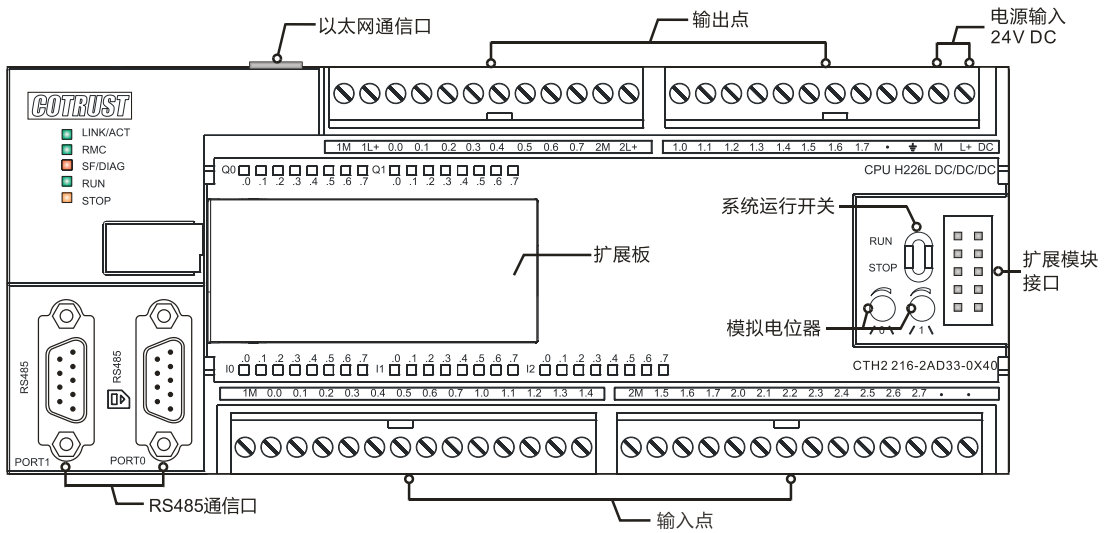


图 4-5 CTH2 216-2AD33-0X40 结构示意图

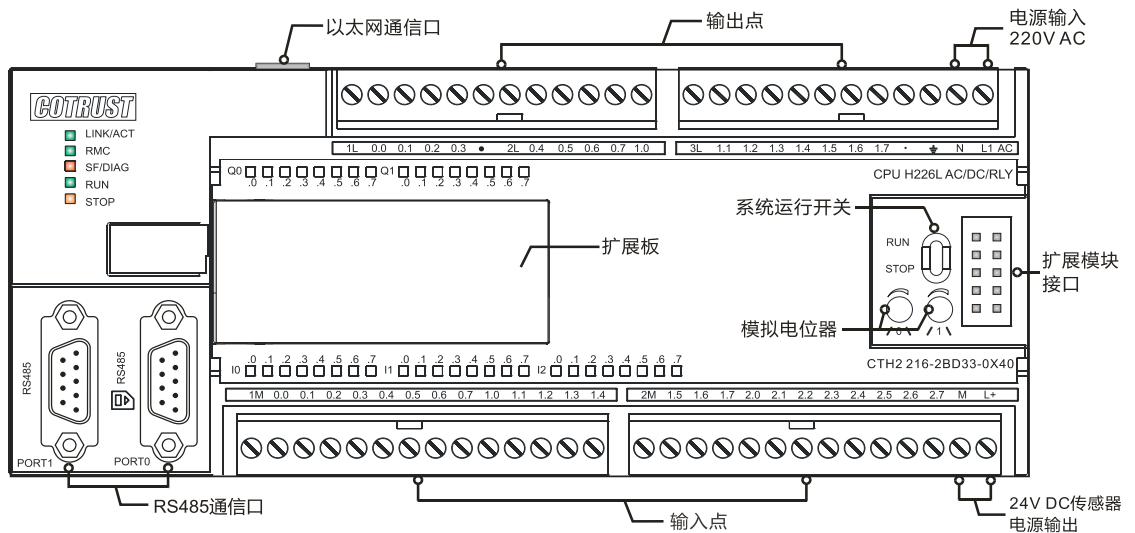


图 4-6 CTH2 216-2BD33-0X40 结构示意图

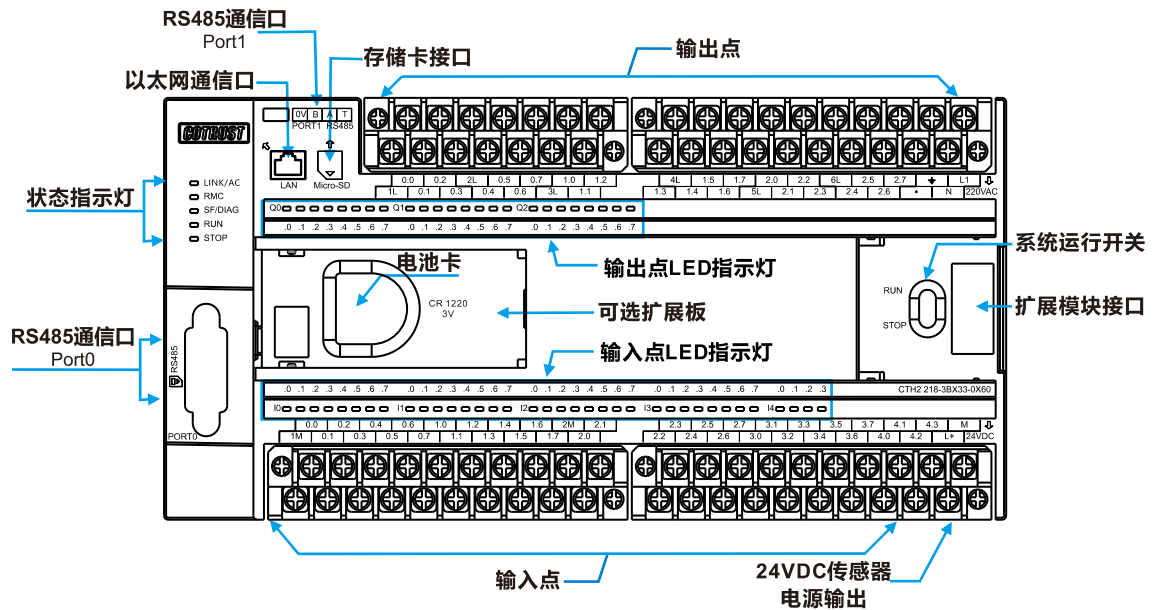


图 4-7 CTH2 218-3BX33-0X60 结构示意图

接线图

CTH200 系列 CPU 的端子接线图如下所示：

➤ H224/H226M 端子接线图相同

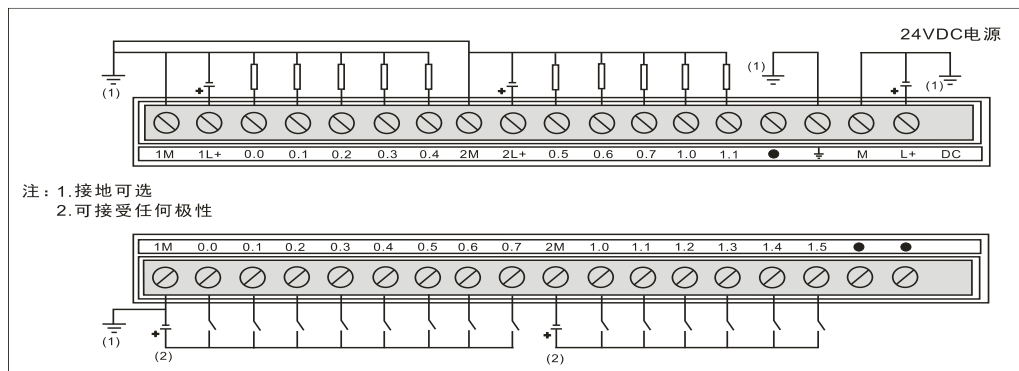


图 4-8 CTH2 214-1AD33-0X24/CTH 216-1AD33-0X24 晶体管输出型接线图

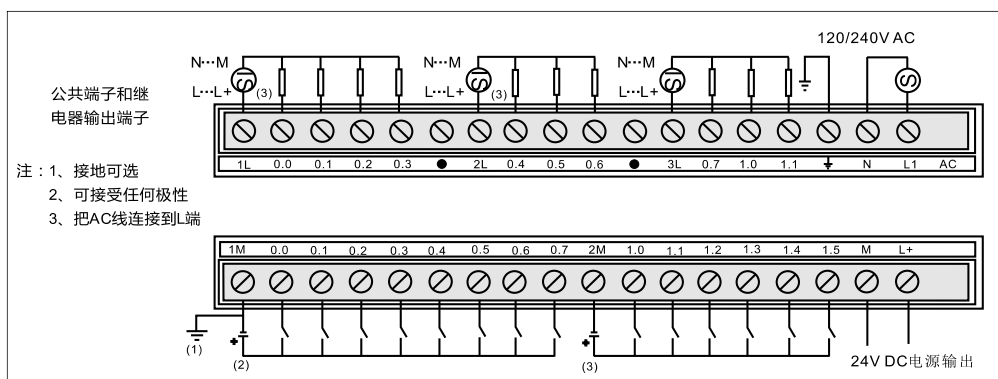


图 4-9 CTH2 214-1BA33-0X24/CTH2 214-1BD33-0X24/CTH 216-1BD33-0X24 继电器输出型接线图

➤ H226L 端子接线图

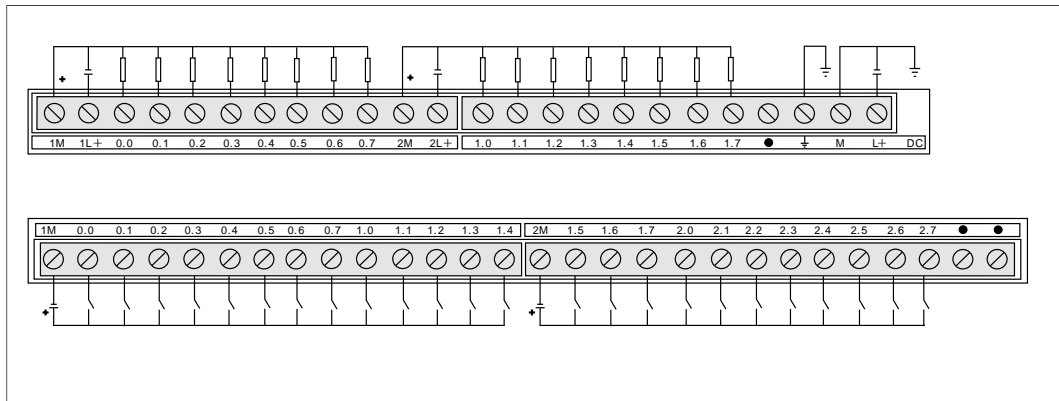


图 4-10 CTH2 216-2AD33-0X40 晶体管输出型接线图

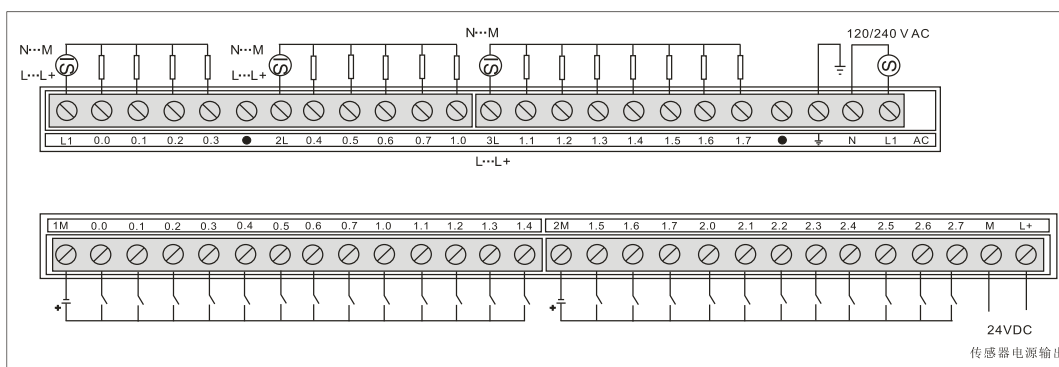


图 4-11 CTH2 216-2BD33-0X40/CTH2 216-1BA33-0X40 继电器输出型接线图

➤ H226XM/H224X/H226IM 端子接线图相同

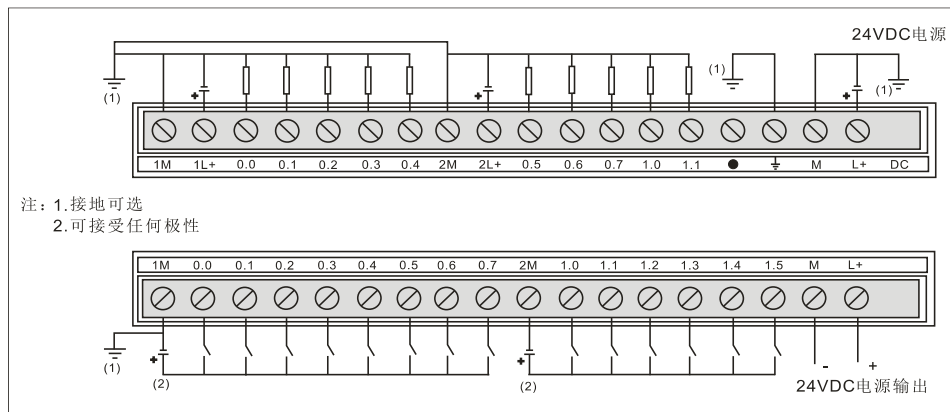


图 4-12 CTH2 216-1AX35-0X24/CTH2 214-1AX33-0X24/CTH2 214-1AX35-0X24/CTH2 216-1AD46-0X24 晶体管输出型接线图

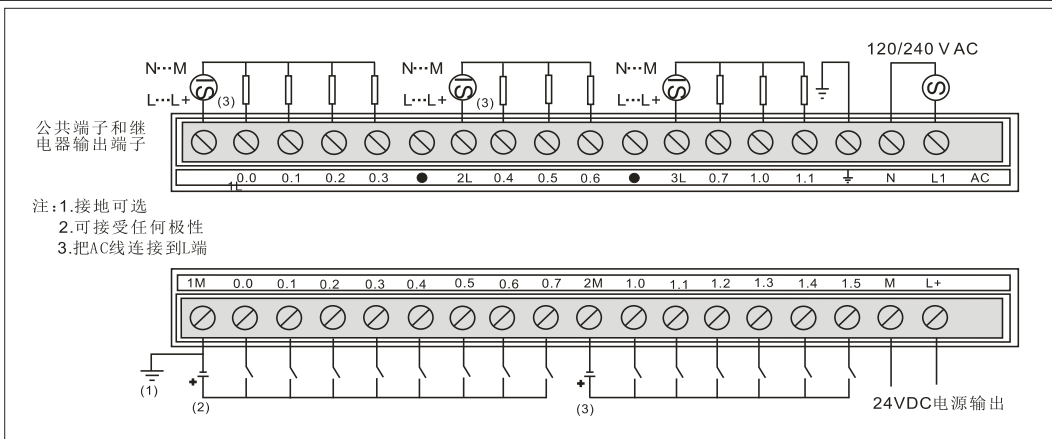


图 4-13 CTH2 216-1BX35-0X24/CTH2 214-1BX33-0X24/CTH2 214-1BX35-0X24/CTH2 216-1BD46-0X24 继电器输出型接线图

➤ H226IL 与 H226XL 端子接线图相同

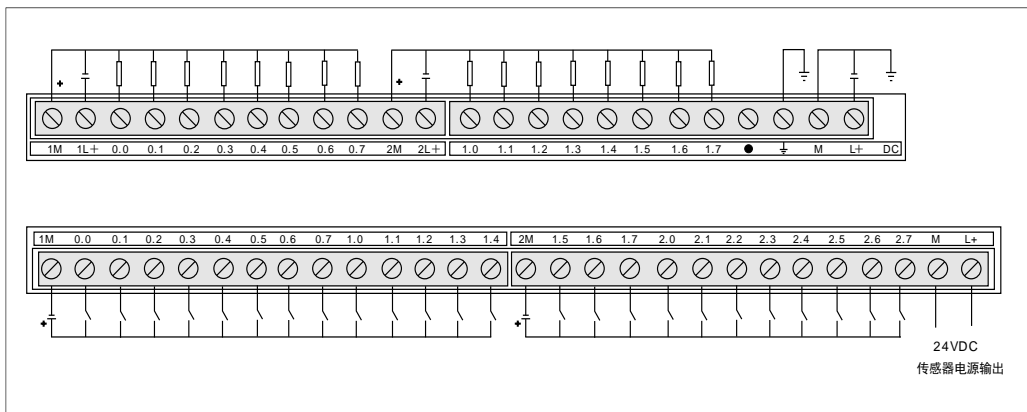


图 4-14 CTH2 216-2AD46-0X40/CTH2 216-2AX33-0X40/CTH2 216-2AX35-0X40 晶体管输出型接线图

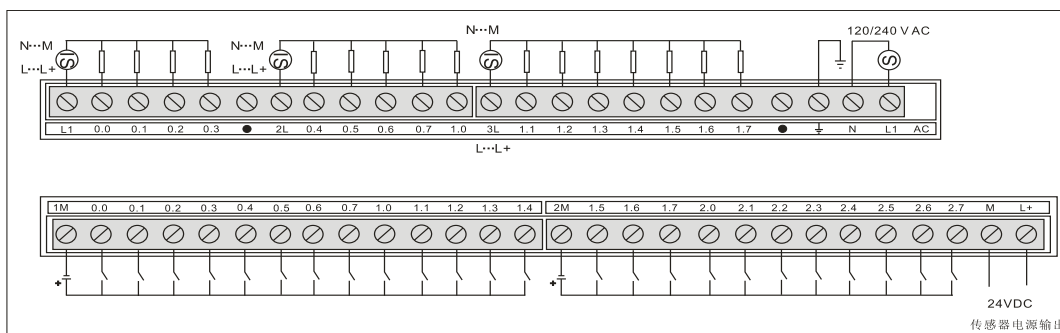


图 4-15 CTH2 216-2BD46-0X40/CTH2 216-2BX33-0X40/CTH2 216-2BX35-0X40 继电器输出型接线图

➤ H228XL 端子接线图

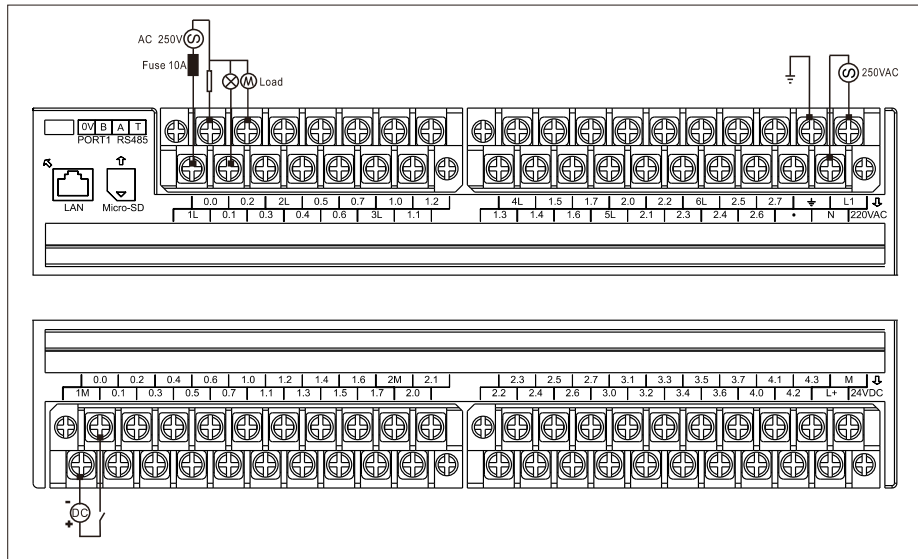
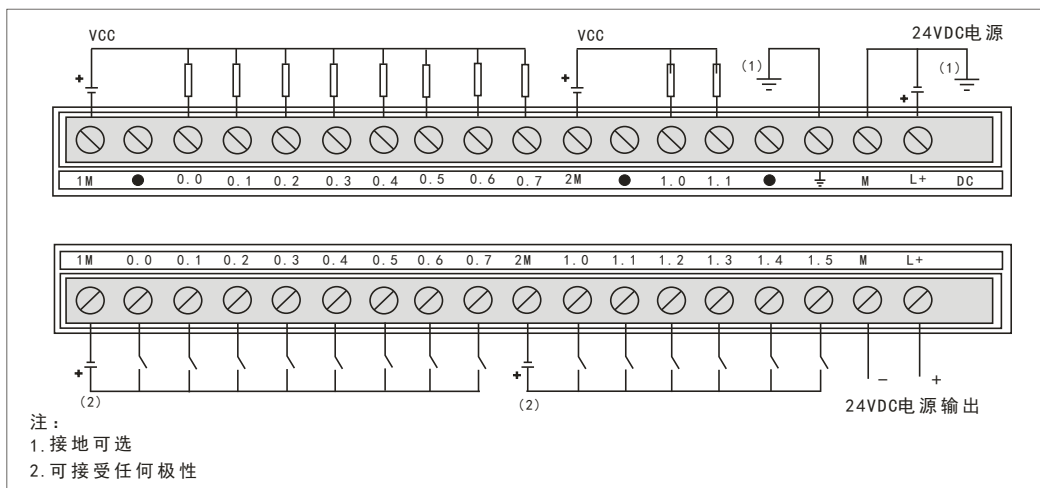


图 4-16 CTH2 218-3BX33-0X60 继电器输出型接线图

➤ H226IH 端子接线图



CTH2 216-1AH46-2B24 晶体管输出型接线图

4.2.3 通信接口信号定义

CTH200 系列 PLC 支持串口通信和以太网通信，机身配有 RS485 接口和标准以太网口，这两种接口的定义如表 4-2-12 到 4-2-14 所示。

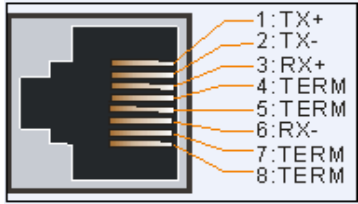
表 4-2-12 标准 RS485 通信接口定义

连接器	插针号	PORT1 (RS485)	PORT0 (RS485)
	1	机壳接地	机壳接地
	2	+24V 地	+24V 地
	3	RS485 信号 B/+	RS485 信号 B/+
	4	发送申请	发送申请
	5	+5V 地	+5V 地
	6	+5V, 100Ω	+5V, 100Ω
	7	+24V	+24V
	8	RS485 信号 A/-	RS485 信号 A/-
	9	NC	NC
	连接器外壳		机壳接地

表 4-2-13 4pin 端子 RS485 通信接口定义（仅限 CPU H228XL）

连接器	端子号	信号	信号定义
	1	0V	逻辑地
	2	B	RS485 信号 B/+
	3	A	RS485 信号 A/-
	4	T	终端电阻，连接端子 3

表 4-2-14 RJ45 以太网通信接口定义

RJ45 通信口	位号	信号	信号定义
	1	TX+	数据发送正端
	2	TX-	数据发送负端
	3	RX+	数据接收正端
	4	TERM	--
	5	TERM	--
	6	RX-	数据接收负端
	7	TERM	--
	8	TERM	--



提示

一些仪表的 RS485 信号定义为 A 正，B 负，接线的时候需要注意对应正负，正对正，负对负（全局通用）。

4.2.4 拨码开关定义

表 4-2-15 拨码开关定义

两态开关	位号	拨码方向	信号定义
	ON	向上	系统运行
	OFF	向下	系统停止
		在 2 秒内快速拨动 6 次	复位 IP 配置 (RUN->STOP 计一次; STOP->RUN 计一次)

4.2.5 制作标准网线

CTH200 CPU 通信所需使用的网线建议采用标准网线，标准网线有平行线，交叉线两种形式。平行线两头同为 568A 标准或 568B 标准；交叉线一头为 568A 标准，一头为 568B 标准。568A/568B 标准线序颜色如下所示：

568B 标准线序：白橙，橙，白绿，蓝，白蓝，绿，白棕，棕。

568A 标准线序：白绿，绿，白橙，蓝，白蓝，橙，白棕，棕。

标准网线制作形式，如下图所示：



图 4-17 标准网线线序

4.3 数字量扩展模块规范

CTH200 系列有多个配套的数字量扩展模块，包括输入模块、输出模块以及输入输出模块。您可以根据具体需要选择使用。下表是数字量扩展模块的订货数据，后续章节分类逐一介绍了各个模块。

表 4-3-1 CTH200 系列 PLC 数字量扩展模块订货数据

规格参数	订货号
SM221数字量输入模块，8点输入，24VDC	CTH2 221-1BF32
SM221数字量输入模块，16点输入，24VDC	CTH2 221-1BH32
SM221数字量输入模块，32点输入，24VDC	CTH2 221-1BL32
SM222数字量输出模块，8点晶体管输出，24VDC，0.5A(输出保护)	CTH2 222-1BF32
SM222数字量输出模块，8点晶体管 NPN 型输出，24VDC，0.5A	CTH2 222-1DF32
SM222数字量输出模块，16点晶体管输出，24VDC，0.5A(输出保护)	CTH2 222-1BH32
SM222数字量输出模块，32点晶体管输出，24VDC，0.5A(输出保护)	CTH2 222-1BL32

SM222数字量输出模块, 8点继电器输出, 2A	CTH2 222-1HF32
SM222数字量输出模块, 16点继电器输出, 2A	CTH2 222-1HH32
SM223数字量输入/输出模块, 4点24VDC输入, 4点晶体管输出, 24VDC, 0.5A(输出保护)	CTH2 223-1BF32
SM223数字量输入/输出模块, 8点24VDC输入, 8点晶体管输出, 24VDC, 0.5A(输出保护)	CTH2 223-1BH32
SM223数字量输入/输出模块, 16点24VDC输入, 16点晶体管输出, 24VDC, 0.5A(输出保护)	CTH2 223-1BL32
SM223数字量输入/输出模块, 4点24VDC输入, 4点继电器输出, 2A	CTH2 223-1HF32
SM223数字量输入/输出模块, 8点24VDC输入, 8点继电器输出, 2A	CTH2 223-1PH32
SM223数字量输入/输出模块, 16点24VDC输入, 16点继电器输出, 2A	CTH2 223-1PL32

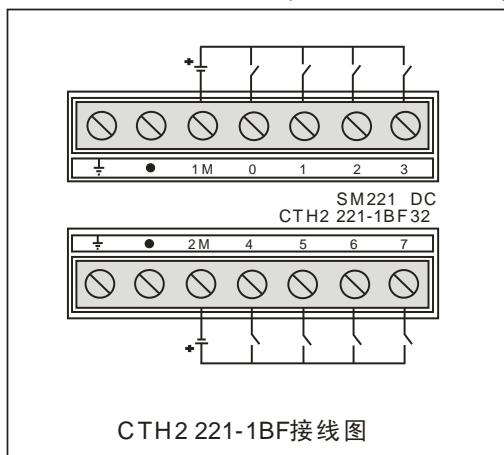
4.3.1 数字量输入模块规范

表 4-3-2 数字量输入模块规范

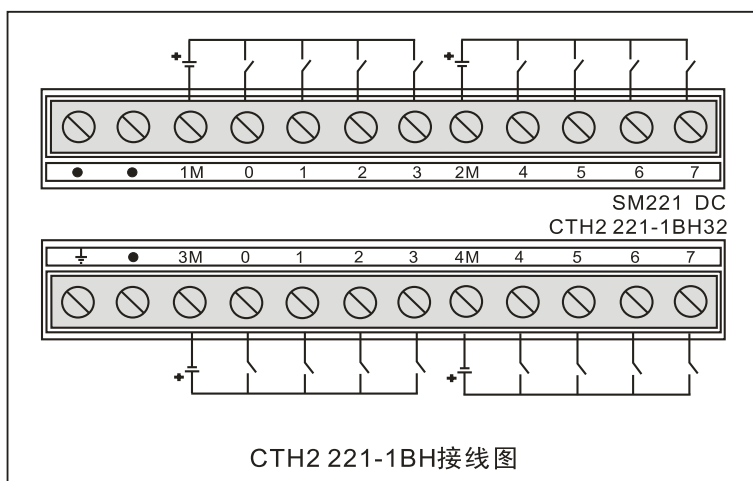
规范	8点输入, 24VDC	16点输入, 24VDC	32点输入, 24VDC
订货号	CTH2 221-1BF32	CTH2 221-1BH32	CTH2 221-1BL32
物理特性			
尺寸(宽×高×深)	46 × 96 × 62mm	71.3 × 96 × 62mm	138 × 96 × 62mm
电源损耗			
功率损耗	2W	3W	
+5VDC 消耗电流	57Ma	79Ma	179Ma
输入特性			
输入点数	8	16	32
输入类型	漏型/源型		
输入电压			
额定值	24VDC		
最大允许电压	30VDC		
浪涌电压	35VDC, 0.5s		
信号"1"	15~30V		
信号"0"	0~5V		
隔离			
光电隔离	500VAC, 持续时间 1 分钟以内		
每组隔离点数	4点	8点	
最长输入时延	4.5ms		
最大漏电流	1Ma AC		
电缆长度(最大)			
非屏蔽	300m		
屏蔽	500m		
可同时接通的输入点数			
40°C	8	16	32
50°C	8	16	32

接线图

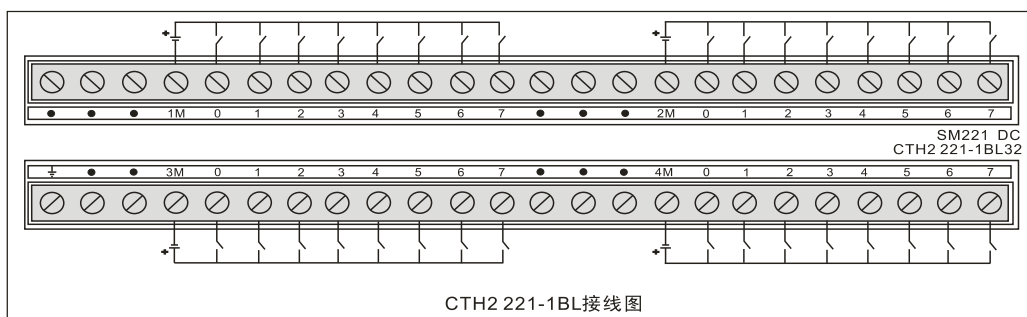
SM221 数字量输入模块(CTH2 221-1BF32)



SM221 数字量输入模块(CTH2 221-1BH32)



SM221 数字量输入模块(CTH2 221-1BL32)



4.3.2 数字量输出模块规范

晶体管输出

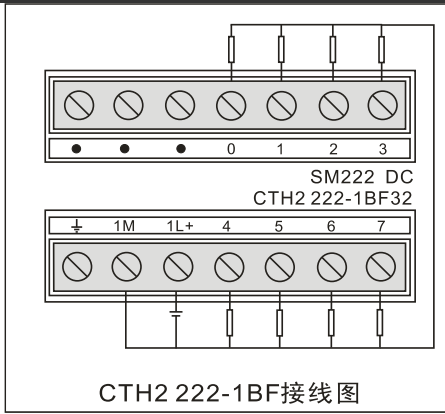
表 4-3-3 数字量输出模块规范

特性	8点输出, 24VDC	16点输出, 24VDC	32点输出, 24VDC
订货号	CTH2 222-1BF32 CTH2 222-1DF32	CTH2 222-1BH32	CTH2 222-1BL32
物理特性			
尺寸 (宽×高×深)	46 × 96 × 62mm	71.3 × 96 × 62mm	138 × 96 × 62mm
电源损耗			

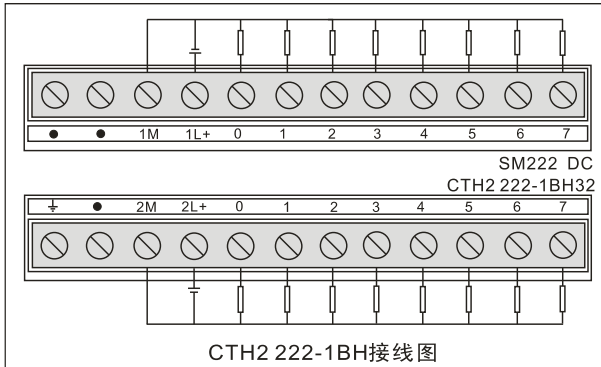
功率损耗	CTH2 222-1BF32: 2W CTH2 222-1DF32: 1W	3W	
+5VDC 消耗电流	CTH2 222-1BF32: 57Ma CTH2 222-1DF32: 110Ma	79Ma	174Ma
输出特性			
输出点数	8	16	32
输出类型	CTH2 222-1BF32: 固态MOSFET (PNP 源型) CTH2 222-1DF32: 固态MOSFET (NPN 漏型)	固态MOSFET (PNP源型)	
输出电压			
额定负载电压	24VDC		
允许电压范围	20.4~28.8VDC		
信号“1”	最小20VDC		
信号“0”	最大0.1VDC (10K Ω 负载)		
输出电流			
信号“1”	0.5A		
输出组数	1	2	4
每组输出点数	8		
同时接通输出点数	8	16	32
每组最大电流	4A		
灯载	5W		
接触电阻	0.3 Ω		
每点最大漏电流	10Ma		
浪涌电流 (最大)	8A, 100ms		
隔离			
隔离	光耦隔离, 500VAC, 持续时间1分钟以内		
每组隔离点数	8点		
输出延时(RL = 50Ω)			
关—开	最大50 μ s		
开—关	最大200 μ s		
电缆长度 (最大)			
非屏蔽	150m		
屏蔽	500m		

接线图

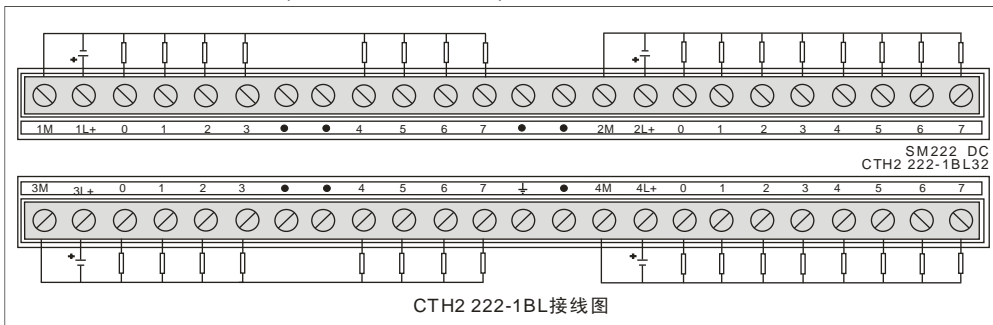
SM222 晶体管输出模块(CTH2 222-1BF32)



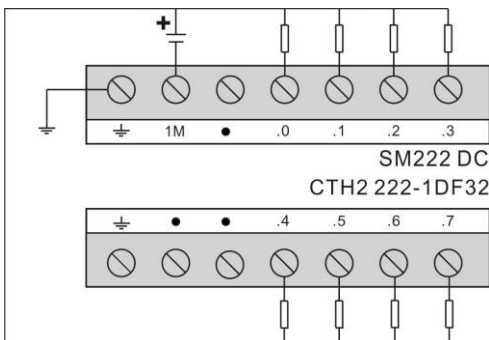
SM222 晶体管输出模块(CTH2 222-1BH32)



SM222 晶体管输出模块(CTH2 222-1BL32)



SM222 晶体管 NPN 型输出模块(CTH2 222-1DF32)



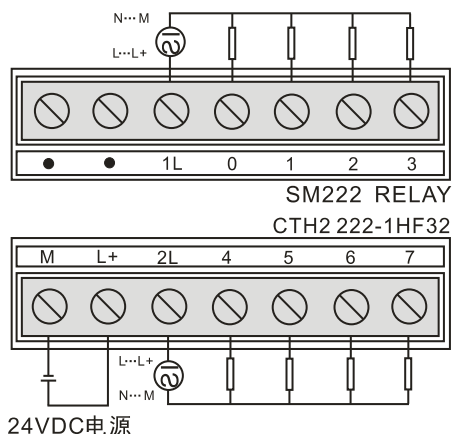
继电器输出

表 4-3-4 数字量输出模块规范

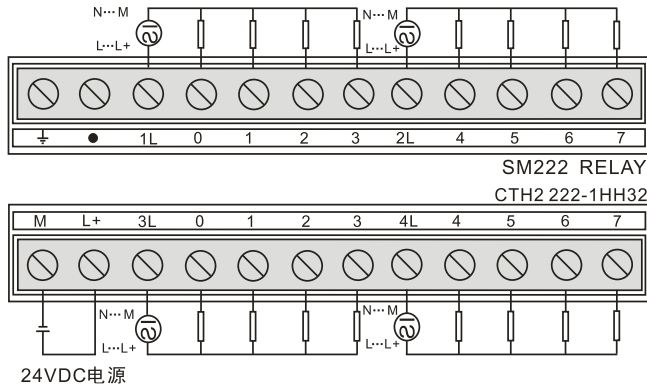
特性	8点输出, 继电器	16点输出, 继电器
订货号	CTH2 222-1HF32	CTH2 222-1HH32
物理特性		
尺寸 (宽×高×深)	46 × 96 × 62mm	71.3 × 96 × 62mm
电源损耗		
功率损耗	2W	3W
从背板总线 (+5VDC)	68Ma	115Ma
输出特性		
输出点数	8	16
输出类型	继电器—干触点	
输出电压范围	DC: 5~30V, AC: 5~250V	
输出电流		
信号“1”	2A	
输出组数	2	
每组输出点数	4	
同时为 ON 的输出点数	8	16
每组最大电流	8A	
灯载	5W	
接触电阻	0.2Ω	
浪涌电流 (最大)	7A, 触点关闭时	
短路保护	外部提供	
隔离		
每组隔离点数	4 点	
线圈和逻辑电源之间	无	
线圈和触点之间	1500VAC, 持续 1 分钟	
电阻 (线圈和触点之间)	最小 100MΩ	
继电器特性		
开关延时	15ms	
开关频率 (最大)	1Hz	
机械开关次数 (空载)	30,000,000	
触点寿命 (额定负载)	300,000	
电缆长度 (最大)		
非屏蔽	150m	
屏蔽	500m	

接线图

SM222 继电器输出模块(CTH2 222-1HF32)



SM222 继电器输出模块(CTH2 222-1HH32)



4.3.3 数字量输入输出模块规范

晶体管输出

表 4-3-5 数字量输入输出模块规范

特性	4点I/O 24VDC	8点I/O 24VDC	16点I/O 24VDC
订货号	CTH2 223-1BF32	CTH2 223-1BH32	CTH2 223-1BL32
物理特性			
尺寸 (宽×高×深)	46 × 96 × 62mm	71.3 × 96 × 62mm	138 × 96 × 62mm
电源损耗			
功率损耗	2W	3W	
+5VDC 消耗电流	57Ma	73Ma	115Ma
输入特性			
输入点数	4	8	16
输入类型	漏型/源型		
输入电压			
额定值	24VDC		
最大允许电压	30VDC		
浪涌电压	35VDC, 0.5s		
信号“1”	15~30V		
信号“0”	0~5V		
隔离			
光耦隔离	500VAC, 持续时间1分钟以内		
每组隔离点数	4点	8点	
最长输入时延	4.5ms		
最大漏电流	1Ma AC		
电缆长度 (最大)			
非屏蔽	300m		
屏蔽	500m		
可同时接通的输入点数			
40°C/50°C	4	8	16
输出特性			
输出点数	4	8	16
输出类型	固态MOSFET		
输出电压			
额定负载电压	24VDC		
允许电压范围	20.4~28.8VDC		
信号“1”	最小20VDC		

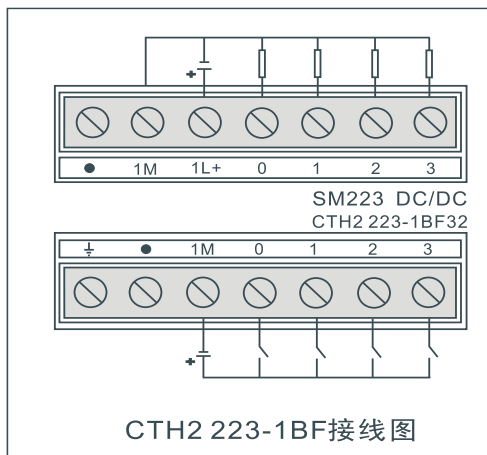
信号"0"	最大0.1VDC (10K Ω 负载)		
输出电流			
信号"1"	0.5A		
输出组数	1	2	
每组输出点数	4	8	
同时接通的输出点数	4	8	16
每组最大电流	2A	4A	
灯载	5W		
接触电阻	0.3 Ω		
每点峰值电流	10Ma		
浪涌电流(最大)	8A, 100ms		
隔离	光耦隔离, 500VAC, 持续时间1分钟以内		
每组隔离点数	4点	4/4/8点	
输出延时(RL = 50Ω)			
关-开	最大50ms		
开-关	最大200ms		
电缆长度(最大)			
非屏蔽	150m		
屏蔽	500m		

**注意**

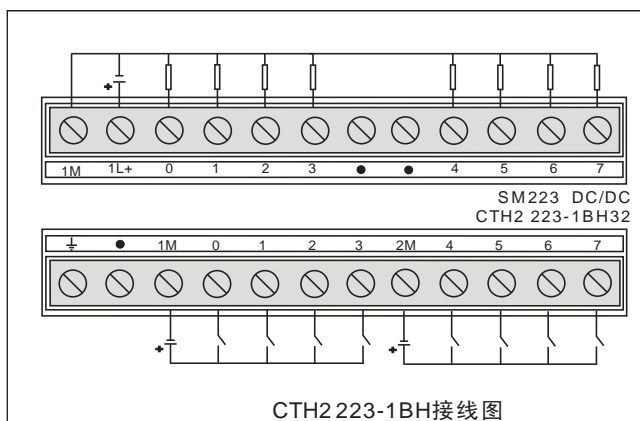
晶体管输出型数字量扩展模块具有输出短路、过流和过压保护功能。

接线图

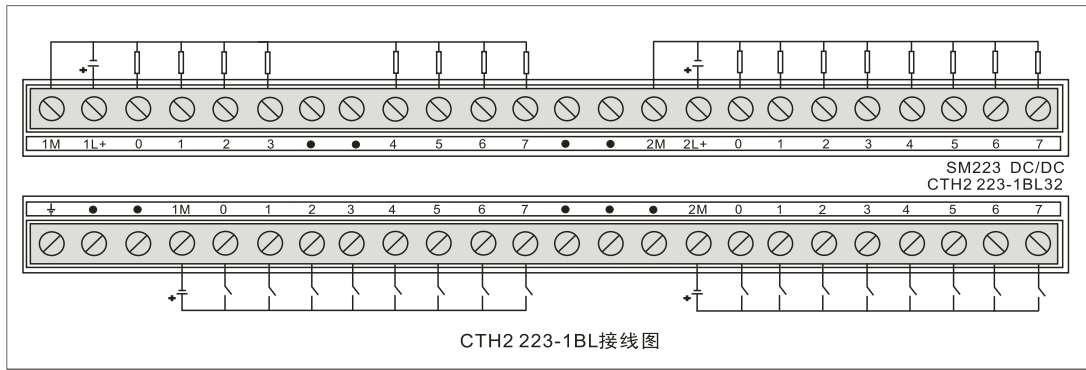
SM223 数字量 I/O 模块(CTH2 223-1BF32)



SM223 数字量 I/O 模块(CTH2 223-1BH32)



SM223 数字量 I/O 模块(CTH2 223-1BL32)



继电器输出

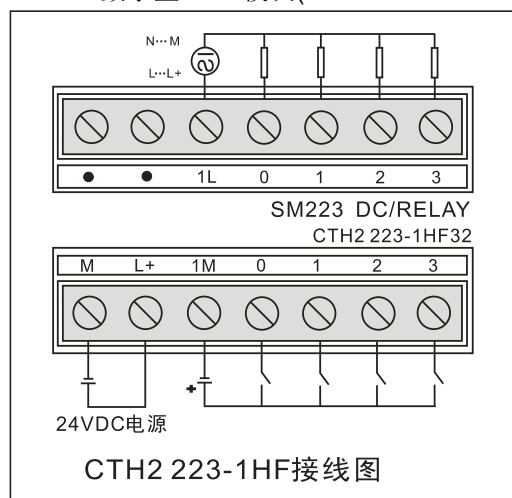
表 4-3-6 数字量输入输出模块规范

特性	4DI×24VDC 4DO×继电器	8DI×24VDC 8DO×继电器	16DI×24VDC 16DO×继电器
订货号	CTH2 223-1HF32	CTH2 223-1PH32	CTH2 223-1PL32
物理特性			
尺寸 (宽×高×深)	46 × 96 × 62mm	71.3 × 96 × 62mm	138 × 96 × 62mm
电源损耗			
功率损耗	2W	3W	6W
+5VDC 消耗电流	58Ma	89Ma	150Ma
L+	接通时每个输出9Ma		
L+线圈电压范围	20.4—28.8VDC		
输入特性			
输入点数	4	8	16
输入类型	漏型/源型		
输入电压			
额定值	24VDC		
最大允许电压	30VDC		
浪涌电压	35VDC, 0.5s		
信号"1"	15~30V		
信号"0"	0~5V		
隔离			
光耦隔离	500VAC, 持续时间 1 分钟以内		
每组隔离点数	4点	8 点	
最长输入时延	4.5ms		
最大漏电流	1Ma AC		
电缆长度 (最大)			
非屏蔽	300m		
屏蔽	500m		
可同时接通的输入点数			
40°C/50°C	4	8	16
输出特性			
输出点数	4	8	16
输出类型	继电器一干触点		

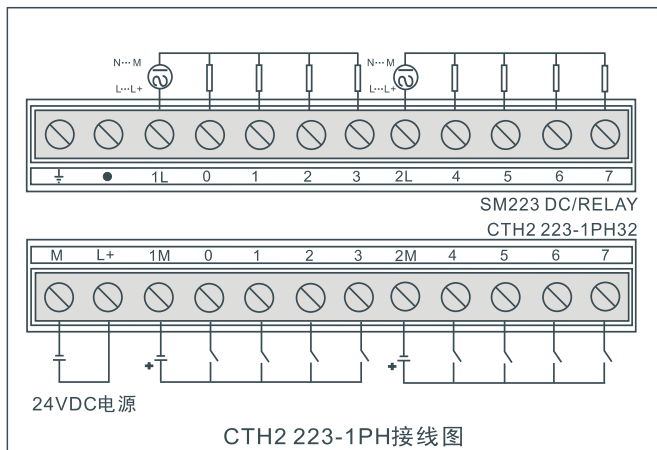
输出电压范围	DC: 5~30V, AC: 5~250V		
输出电流			
信号“1”	2A		
输出组数	1	2	4
每组输出点数	4		
同时接通输出点数	4	8	16
每组最大电流	8A		
灯载	DC: 30W, AC: 200W		
接触电阻	0.2Ω		
浪涌电流（最大）	7A, 触点关闭		
短路保护	外部提供		
隔离			
每组隔离点数	4点		
线圈—逻辑电源间	无		
线圈—触点	1500VAC, 持续1分钟		
电阻（线圈和触点之间）	最小100MΩ		
继电器特性			
开关延时	15ms		
开关频率（最大）	1Hz		
机械开关次数（空载）	30,000,000		
触点寿命（额定负载）	300,000		
电缆长度（最大）			
非屏蔽	150m		
屏蔽	500m		

接线图

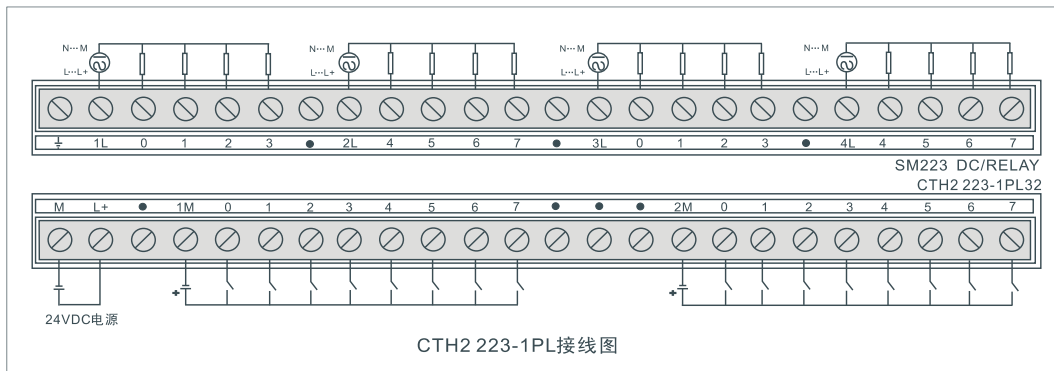
SM223 数字量 I/O 模块(CTH2 223-1HF32)



SM223 数字量 I/O 模块(CTH2 223-1PH32)



SM223 数字量 I/O 模块(CTH2 223-1PL32)



4.4 模拟量扩展模块规范

表 4-4-1 模拟量扩展模块订货数据

规格参数	订货号
SM231 模拟量输入模块, 4 点, 0~20Ma 电流或 ±5V, ±2.5V, 0~10V, 0~5V 电压输入, 隔离型 12 位精度	CTH2 231-0HC32
SM231 高精度模拟量输入模块, 8 点输入, 电压输入, 光偶隔离 16 位精度	CTH2 231-0HF32
SM231 高精度模拟量输入模块, 8 点输入, 电流输入, 光偶隔离 16 位精度	CTH2 231-1HF32
SM231 模拟量电压输入模块, 8 点, ±2.5V, 0~10V, 0~5V 电压输入, 两通道可选 0~20Ma 电流输入, 隔离型 12 位精度	CTH2 231-5HF32
SM232 模拟量输出模块, 2 点, ±10V 电压或 0~20Ma 电流输出, 隔离型电压 12 位精度或电流 11 位精度	CTH2 232-0HB32
SM232 模拟量输出模块, 4 点, ±10V 电压或 0~20Ma 电流输出, 隔离型电压 12 位精度或电流 11 位精度	CTH2 232-0HD32
SM235 模拟量输入输出模块, 4 点电压或电流输入/1 点电压或电流输出, 隔离型电压 12 位精度或电流 11 位精度	CTH2 235-0KD32

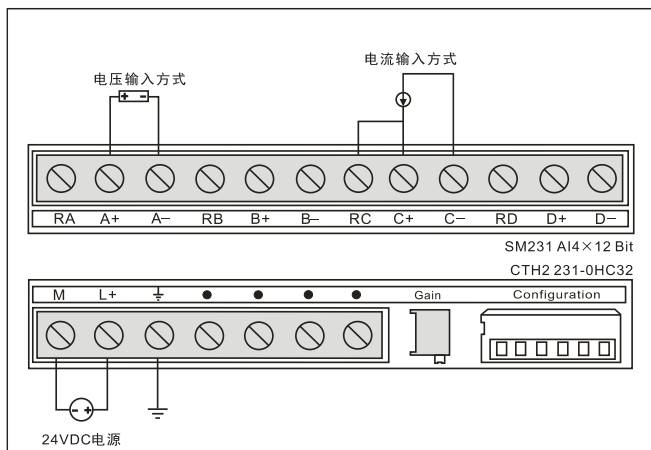
4.4.1 模拟量输入模块规范

表 4-3-2 SM231-0HC 规范

物理特性	
尺寸 (宽×高×深)	71.3×96×62mm
LED 指示灯	24V 电源指示灯, 亮: 表示电源正常, 灭: 表示电源故障
电源损耗	
+5VDC 消耗电流	87Ma
L+	17Ma
L+线圈电压范围	20.4~28.8VDC
功率损耗	2W
模拟量输入特性	
输入点数	4
隔离 (现场与逻辑)	光耦隔离: 500VAC, 1分钟
输入类型	差分输入
量程范围	
电压输入 (单极性)	0~10V, 0~5V
电压输入 (双极性)	±5V, ±2.5V
电流输入	0~20Ma
数据字格式	
单极性, 全量程	0~32000
双极性, 全量程	-32000~32000
输入分辨率	
电压输入 (单极性)	2.5Mv (0~10V量程); 1.25Mv (0~5V量程)
电压输入 (双极性)	2.5Mv (±5V量程); 1.25Mv (±2.5V量程)
电流输入	5Ma (0~20Ma量程)
模数转换时间	小于300μs
模拟量输入响应时间	1.5ms
共模抑制	40Db, DC 60Hz
共模电压	信号电压+共模电压 (必须小于等于12V)
输入阻抗	不小于10MΩ
最大输入电压	30V
最大输入电流	30Ma

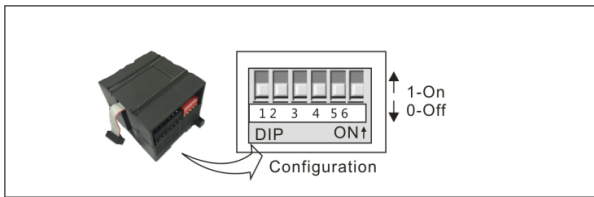
接线图

SM231 模拟量输入模块(CTH2 231-0HC32)



校准与量程选择

【校准与量程选择开关位置】



【输入校准】

由于校准调节影响到模拟多路开关后的运放，因而将影响所有的用户输入通道。另外，由于多路开关前的各输入通道的元件参数可能存在差异，因而即使在校准后，同一输入信号在不同通道上的读数也就会存在轻微的差异。

模块内部对输入已经做了滤波处理，测量结果比较稳定。如果要求更好的性能参数，可以启动用于模块所有输入的模拟输入滤波器，计算平均值时，选择 64 次以上的采样次数。

校准步骤如下：

- 1) 切断模块外部电源，配置 DIP 开关以选择所需要的输入量程。
- 2) 接通 PLC 系统的 CPU 和模块电源，等待 15 分钟以上
- 3) 用一个变送器、一个电压源或一个电流源，将满刻度值信号加到一个输入通道
- 4) 读取 CPU 中该输入通道的测量值读数
- 5) 调节 GAIN（增益）电位器，直到读数为 32000。

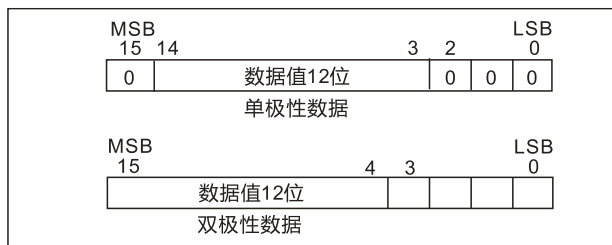
【量程选择】

下表所示为如何用 DIP 开关设置 SM231 模拟量输入模块的量程。开关 1、2 和 3 可选择模拟量输入范围。所有的输入设置成相同的模拟量输入范围。下表中，ON 为接通，OFF 为断开。需要特别注意的是，未使用的 DIP 开关 SW4~SW6，必须设置到 OFF 的位置。

表 4-4-3 SM231-0HC 的 DIP 开关配置

单极性			满量程输入	分辨率
SW1	SW2	SW3		
ON	OFF	ON	0 — 10V	2.5Mv
ON	ON	OFF	0 — 5V	1.25Mv
ON	ON	OFF	0 — 20Ma	5Ma
双极性			满量程输入	分辨率
SW1	SW2	SW3		
OFF	OFF	ON	±5V	2.5Mv
OFF	ON	OFF	±2.5V	1.25Mv

输入数据格式



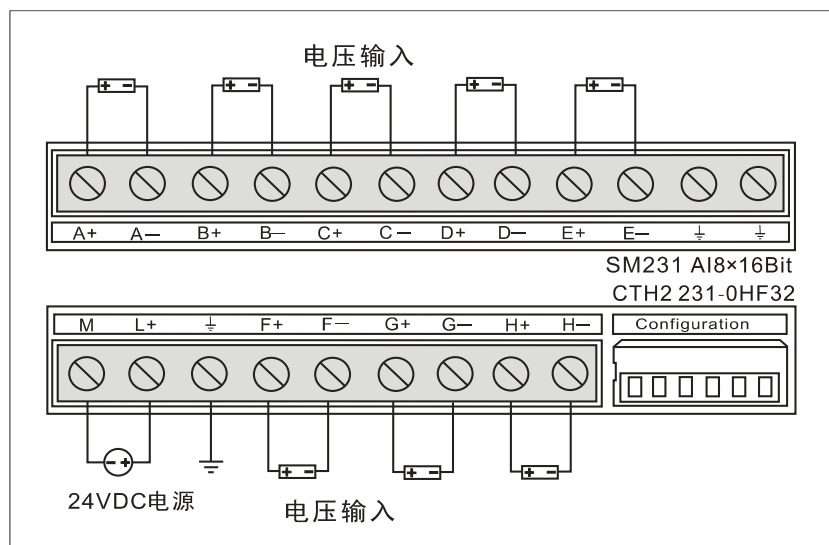
提示

模数转换器（ADC）的 12 位读数，其数据格式是左端对齐的。最高有效位是符号位（0 表示正数），对单极性格式，3 个连续的 0 使得 ADC 计数值每变化 1 个单位，则数据字的变化是以 8 为单位变化的。对双极性格式，4 个连续的 0 使得 ADC 计数值每变化 1 个单位，则数据字的变化是以 16 为单位变化的。

表 4-4-5 SM231-0HF 模拟量输入模块规范

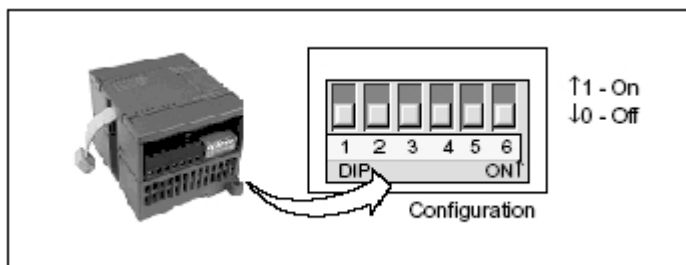
物理特性	
尺寸(宽×高×深)	71×80×62mm
LED指示灯	24V 电源指示灯, 亮: 表示电源正常, 灭: 表示电源故障
电源损耗	
+5VDC消耗电流	87 Ma
L+	31Ma
L+ 线圈电压范围	20.4~28.8V DC
功率损耗	2.5 W
模拟量输入特性	
输入点数	8
隔离 (现场与逻辑)	光耦隔离
输入类型	差分输入
量程范围	
电压输入(单极性)	0~10V, 0~5V
电压输入(双极性)	±5V, ±2.5V
数据字格式	
单极性, 全量程	0~32000
双极性, 全量程	-32000~32000
输入分辨率	
电压 输入(单极性)	300Mv (0~10V 量程); 150Mv (0~5V 量程)
电压输入(双极性)	150Mv (±5V 量程); 75Mv (±2.5V 量程)
模数转换时间	约10ms
八通道转换时间	100ms
共模抑制	40Db, DC — 60Hz
共模电压	信号电压+共模电压(必须小于等于12V)
输入阻抗	不小于10MΩ
最大输入电压	30V
最大输入电流	30Ma
AD转换器分辨率	16位

接线图



量程选择与软件配置

【量程选择开关位置】



【量程选择】

下表所示为如何用 DIP 开关设置 SM231 8AI 模块的量程。开关 1、2 和 3 可选择模拟量输入范围。所有的输入设置成相同的模拟量输入范围。下表中，ON 为接通，OFF 为断开。需要特别注意的是，未使用的 DIP 开关 SW4~SW6，必须设置到 OFF 的位置。

表 4-4-6 SM231-0HF 的 DIP 开关配置

单极性			满量程输入	分辨率
SW1	SW2	SW3		
ON	OFF	ON	0-10V	300Mv
ON	ON	OFF	0 — 5V	150Mv
双极性			满量程输入	分辨率
SW1	SW2	SW3		
OFF	OFF	ON	± 5V	150Mv
OFF	ON	OFF	± 2.5 V	75Mv

【软件配置】

对于 SM231 8AI×16 位模拟量输入模块，其读数是在 VW 中，而不是 AIW，模块所处的相对位置不同，对应的地址也不同。地址计算公式如下：

$$x(VWx) = \text{槽位号} \times 64 + \text{输入通道号} \times 2$$

槽位号对应模块的安装位置，紧靠 CPU 的第一个扩展模块槽位号为 0，第二个扩展模块槽位号为 1，依此类推。输入通道共 8 路，从 A 至 G，对应的编号为 0 至 7。表 4-4-7 为基于以上公式计算得到的地址速查表。

表 4-4-7 地址速查表

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



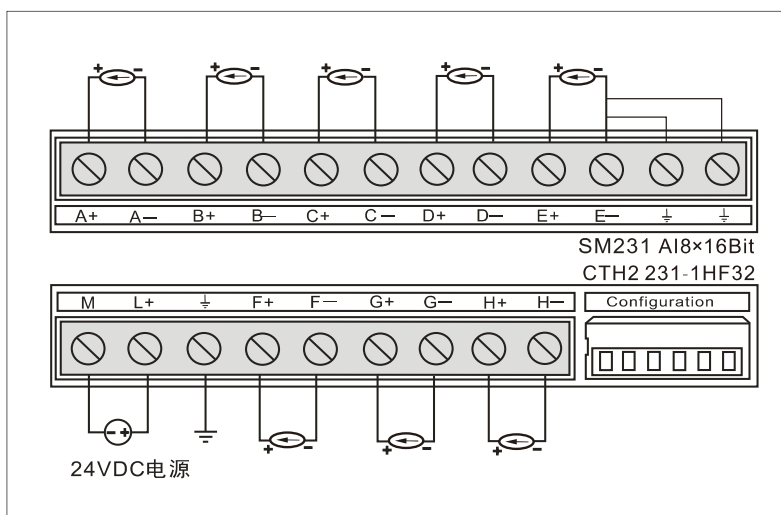
提示

由于 COTRUST 的 TD2X 文本显示器和 SIEMENS 的 TD200 文本显示面板所分配的地址固定为 VW0，因而如果你的系统中需要同时使用到 TD2X 或 TD200 和 SM231 8AI 模块，则 SM231 8AI 模块不能安装在第一个位置(Slot 0)，否则将不能正常工作。

表 4-4-8 SM231-1HF 模拟量输入模块规范

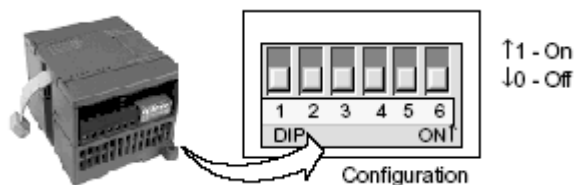
物理特性	
尺寸 (W×H×D), mm	120.5×80×62
重量	210 g
插入式 I/O 端子	否
电源消耗	
总线 (5V DC)	87Ma
L+	30MA
L+电压范围	20.4~28.8V DC
功率损耗	1W
模拟量输入特性	
输入点数	8
输入类型	电流
量程范围	0~20Ma; 4~20Ma
数据字格式	0~32000
分辨率	0.000625 Ma (0~20Ma) ; 0.0005 Ma (4~20Ma)
测量精度	0.1%
输入阻抗	250 Ω
更新时间	100ms (8 通道)
共模电压	120V AC
共模抑制	90Db, 60Hz @ DC
输入滤波衰减	-3db @ 325 HZ
诊断	
诊断程序	+24VDC 电源指示灯, 亮: 模块供电正常, 灭: 模块无电源, 所有通道数据变为 32766 拨码开关设置错误, 所有通道数据变为 32767
隔离特性	
现场侧—逻辑	500V AC
现场侧—直流 24V	500V AC
直流 24V—逻辑	500V AC

接线图



量程选择与软件配置

【量程选择开关位置】



【量程选择】

下表所示为如何用 DIP 开关设置 SM231 8AI 模块的量程。开关 1、2 和 3 可选择模拟量输入范围。所有的输入设置成相同的模拟量输入范围。

下表中，ON 为接通，OFF 为断开。

需要特别注意的是，未使用的 DIP 开关 SW4~SW6，必须设置到 OFF 的位置。

表 4-4-9 SM231-1HF 的 DIP 开关配置

单极性			满量程输入	分辨率
SW1	SW2	SW3		
OFF	OFF	OFF	0-20Ma	0.000625Ma
OFF	OFF	ON	4-20Ma	0.0005Ma

【软件配置】

对于 SM231 8AI×16 位模拟量输入模块，其读数是在 VW 中，而不是 AIW，模块所处的相对位置不同，对应的地址也不同。地址计算公式如下：

$$x(VWx) = \text{槽位号} \times 64 + \text{输入通道号} \times 2$$

槽位号对应模块的安装位置，紧靠 CPU 的第一个扩展模块槽位号为 0，第二个扩展模块槽位号为 1，依此类推。输入通道共 8 路，从 A 至 G，对应的编号为 0 至 7。表 4-4-9 为基于以上公式计算得到的地址速查表。

表 4-4-9 地址速查表

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



提示

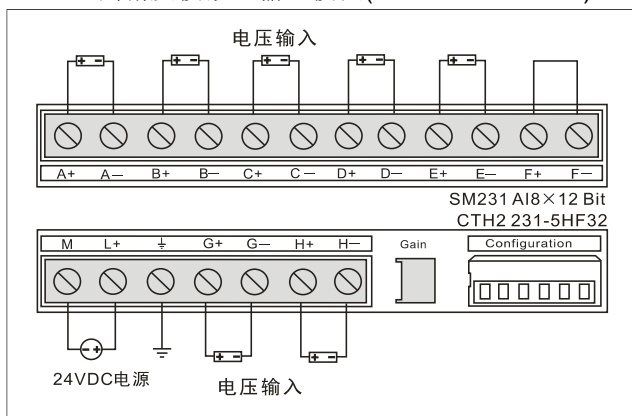
由于 COTRUST 的 TD2X 文本显示器和 SIEMENS 的 TD200 文本显示面板所分配的地址固定为 VW0，因而如果你的系统中需要同时使用到 TD2X 或 TD200 和 SM231 8AI 模块，则 SM231 8AI 模块不能安装在第一个位置(Slot 0)，否则将不能正常工作。

表 4-4-10 SM231-5HF 规范

物理特性	
尺寸 (宽×高×深)	71.3 × 96 × 62mm
LED指示灯	24V电源指示灯, 亮: 表示电源正常, 灭: 表示电源故障
电源损耗	
+5VDC消耗电流	87Ma
L+	50Ma
L+ 线圈电压范围	20.4~28.8VDC
功率损耗	2.5W
模拟量输入特性	
输入点数	8
隔离(现场与逻辑电路间)	光耦隔离
输入类型	差分输入, 可为电流选择两个通道
量程范围	
电压输入 (单极性)	0~10V, 0~5V
电压输入 (双极性)	±2.5V (通道0~7)
电流	0~20Ma (通道6~7)
数据字格式	
单极性, 全量程	0~32000
双极性, 全量程	-32000~32000
输入分辨率	
电压输入 (单极性)	2.5Mv (0~10V量程); 1.25Mv (0~5V量程)
电压输入 (双极性)	2.5Mv (±5V量程); 1.25Mv (±2.5V量程)
模数转换时间	约10ms
八通道转换时间	100ms
共模抑制	40Db, DC — 60Hz
共模电压	-12V ≤ 信号电压 + 共模电压 ≤ +12V
输入阻抗	电压: ≥2MΩ 电流: 250Ω
最大输入电压	30VDC
最大输入电流	32Ma
A/D转换器分辨率	单极性: 12bit; 双极性: 11bit+符号位
测量原理	逐次逼近
测量误差	0.5% (最大)
隔离 (现场至逻辑)	500V AC
模拟输入寻址	AIW 寻址

接线图

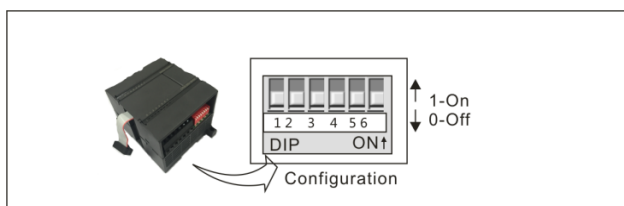
SM231 高精度模拟量输入模块(CTH2 231-5HF32)



<注> 上图中将未使用的输入短接, 如 F+和 F-。

量程选择与软件配置

【量程选择开关位置】



【量程选择】

下表所示为如何用 DIP 开关设置 SM231 8AI 模块的量程。开关 3、4 和 5 选择模拟量输入范围（见表 4-4-11），使用开关 1 和 2 来选择电流模式输入，开关 1 为 ON 时选择通道 6 的电流输入模式；为 OFF 时选择电压模式。开关 2 为 ON 时选择通道 7 的电流输入模式；为 OFF 时选择电压模式。

表 4-4-11 SM231-5HF 的 DIP 开关配置

单极性			满量程输入	分辨率
SW3	SW4	SW5		
ON	OFF	ON	0-10V	2.5Mv
ON	ON	OFF	0-5V	1.25Mv
			0-20Ma	5Ma
双极性			满量程输入	分辨率
SW3	SW4	SW5		
OFF	OFF	ON	±5V	2.5Mv
OFF	ON	OFF	±2.5V	1.25Mv

4.4.2 模拟量输出模块规范

表 4-4-12 模拟量输出模块规范

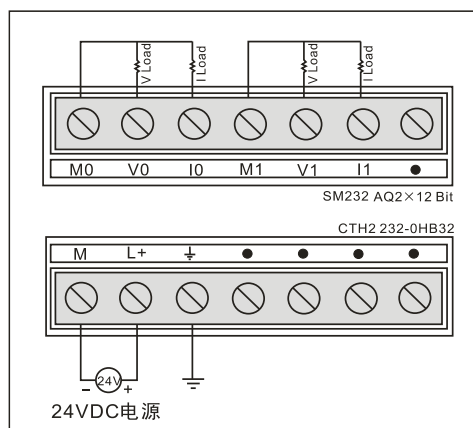
特性	SM232: 2 AQ × 12位	SM232: 4 AQ × 12位
订货号	CTH2 232-0HB32	CTH2 232-0HD32
物理特性		
尺寸（宽×高×深）	46 × 96 × 62mm	71.3 × 96 × 62mm
LED指示灯	24V电源指示灯，亮：电源正常，灭：电源故障	
电源损耗		
+5VDC 消耗电流	87Ma	
L+	61Ma	112Ma
L+线圈电压范围	20.4~28.8VDC	
功率损耗	2W	
模拟量输出特性		
输出点数	2	4
隔离（模拟到数字）	光耦隔离：500VAC，持续1分钟	
输出范围		
电压输出	±10V	
电流输出	0~20Ma	
输出分辨率		
电压输出	12位	

电流输出	11位
数据字格式	
电压输出	-32000~+32000
电流输出	0~32000
测量误差	典型情况: 满量程 $\pm 0.5\%$; 最坏情况: 满量程 $\pm 2\%$
稳定时间	
电压输出	100 μ s
电流输出	2ms
最大驱动@24V用户电源	
电压输出	最小5000 Ω
电流输出	最大500 Ω

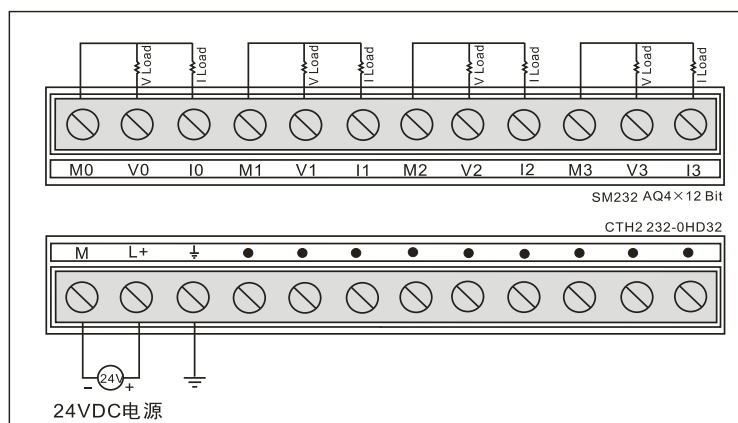
CPU 为该模块保留两个模拟输出点，即 235 模块模拟量输出点会占用两个 AQW 输出地址。

接线图

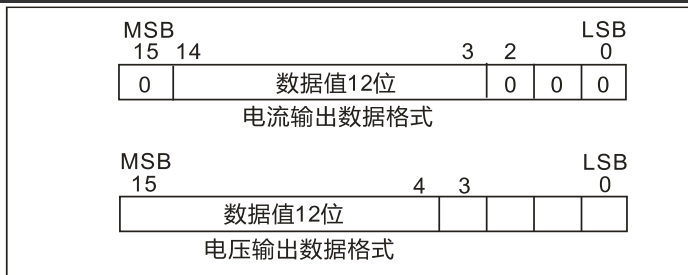
SM232 模拟量输出模块(CTH2 232-0HB32)



SM232 模拟量输出模块(CTH2 232-0HD32)



输出数据格式



提示

数模转换器（DAC）的 12 位读数，其输出数据格式是左端对齐的，最高有效位是符号位（0 表示是正数），数据在装载到 DAC 寄存器之前，4 个连续的 0 是被裁断的，这些位不影响输出信号值。

4.4.3 模拟量输入输出模块规范

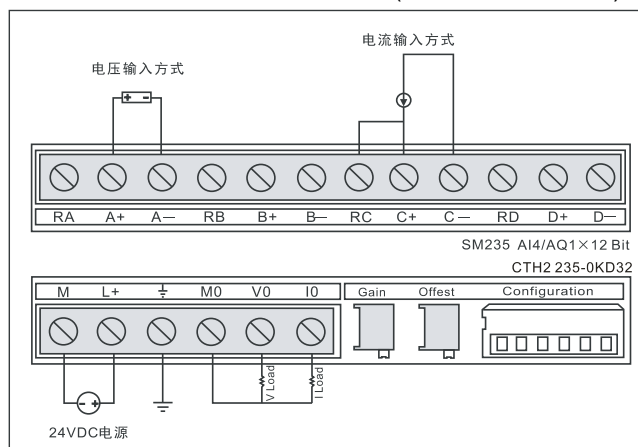
表 4-4-13 模拟量输入输出模块规范

物理特性	
尺寸（宽×高×深）	71.3 × 96 × 62mm
LED指示灯	24V电源指示灯，亮：表示电源正常，灭：表示电源故障
电源损耗	
+5VDC 消耗电流	87Ma
L+	48Ma
L+线圈电压范围	20.4~28.8VDC
功率损耗	2W
模拟量输入特性	
输入点数	4
隔离（现场与逻辑电路间）	光耦隔离：500VAC，持续1分钟
输入类型	差分输入
量程范围	
电压输入（单极性）	0-10V, 0-5V, 0-1V, 0-500Mv, 0-100Mv, 0-50Mv
电压输入（双极性）	±10V,±5V,±2.5V,±1V,±500Mv, ±250Mv, ±100Mv,±50Mv,±25Mv
电流输入	0~20Ma
数据字格式	
单极性，全量程	0~32000
双极性，全量程	-32000~32000
输入分辨率	
电压输入（单极性）	2.5Mv（0~10V量程）； 1.25Mv（0~5V量程）
电压输入（双极性）	2.5Mv（±5V量程）； 1.25Mv（±2.5V量程）
电流输入	5Ma（0~20Ma量程）
模数转换时间	小于300μs
模拟量输入响应时间	1.5ms
共模抑制	40Db, DC~60Hz
共模电压	信号电压+共模电压（必须小于等于12V）
输入阻抗	不小于10MΩ
最大输入电压	30V

最大输入电流	30Ma
A/D转换器分辨率	12位
模拟量输出特性	
输出点数	1
电压输出	$\pm 10V$
电流输出	0~20Ma
输出分辨率	
电压输出	12位
电流输出	11位
数据字格式	
电压输出	-32000~+32000
电流输出	0~32000
精度	
电压输出	典型情况：满量程 $\pm 0.5\%$ ，最坏情况：满量程 $\pm 2\%$
电流输出	
稳定时间	
电压输出	100 μs
电流输出	2ms
最大驱动@24V用户电源	
电压输出	最小5000 Ω
电流输出	最大500 Ω

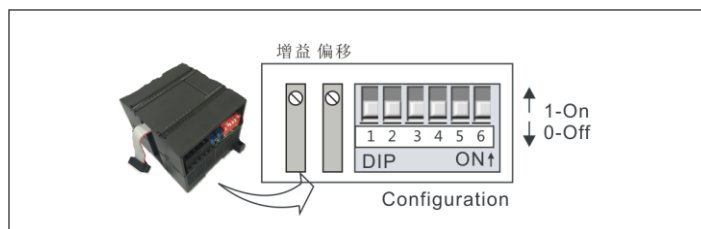
接线图

SM235 模拟量混合输入输出模块(CTH2 235-0KD32)



校准、量程与增益选择

【校准与量程选择开关位置】



【输入校准】

由于校准调节影响到模拟多路开关后的运放，因而将影响所有的用户输入通道。另外，由于多路

开关前的各输入通道的元件参数可能存在差异，因而即使在校准后，同一输入信号在不同通道上的读数也就会存在轻微的差异。

模块内部对输入已经做了滤波处理，测量结果比较稳定。如果要求更好的性能参数，可以启动用于模块所有输入的模拟输入滤波器，计算平均值时，选择 64 次以上的采样次数。

校准步骤如下：

- 1) 切断模块外部电源，配置 DIP 开关以选择所需要的输入量程。
- 2) 接通 PLC 系统的 CPU 和模块电源，等待 15 分钟以上。
- 3) 用一个变送器、一个电压输入源或一个电流输入源，将零值信号加到模块的一个输入端。
- 4) 在 CPU 中读取该输入端的测量值读数。
- 5) 调节 OFFSET（偏移）电位器，直到该读数为零，完成调零校准。
- 6) 接着，在该输入端输入一个满刻度值信号，读出 CPU 中的测量值读数。
- 7) 调节 GAIN（增益）电位器，直到该读数为 32000。
- 8) 必要时，重复步骤 3~7。

【量程与增益选择】

表 4-4-14 和表 4-4-15 所示为如何用 DIP 开关设置 SM235 模块。开关 SW1 到 SW6 可选择模拟量输入范围和分辨率。所有的输入设置成相同的模拟量输入范围和格式。表 4-4-16 所示为如何选择单/双极性（SW6）、增益（开关 SW4 和 SW5）和衰减（开关 SW1、SW2 和 SW3）。下列表中，ON 为接通，OFF 为断开。

表 4-4-14 SM235-0KD 的 DIP 开关配置（单极性）

单极性						满量程输入	分辨率
SW1	SW2	SW3	SW4	SW5	SW6		
ON	OFF	OFF	ON	OFF	ON	0 — 50Mv	12.5Mv
OFF	ON	OFF	ON	OFF	ON	0 — 100Mv	25Mv
ON	OFF	OFF	OFF	ON	ON	0 — 500Mv	125Mv
OFF	ON	OFF	OFF	ON	ON	0 — 1V	250Mv
ON	OFF	OFF	OFF	OFF	ON	0 — 5V	1.25Mv
ON	OFF	OFF	OFF	OFF	ON	0 — 20Ma	5Ma
OFF	ON	OFF	OFF	OFF	ON	0 — 10V	2.5Mv

表 4-4-15 SM235-0KD 的 DIP 开关配置（双极性）

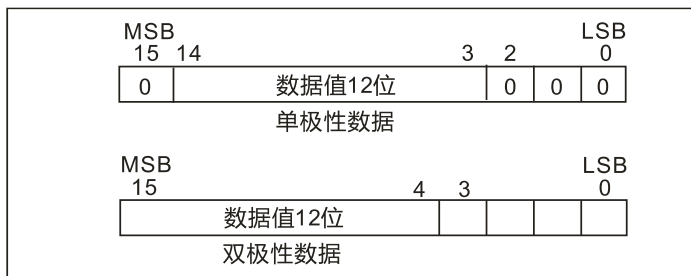
双极性						满量程输入	分辨率
SW1	SW2	SW3	SW4	SW5	SW6		
ON	OFF	OFF	ON	OFF	OFF	±25Mv	12.5Mv
OFF	ON	OFF	ON	OFF	OFF	±50Mv	25Mv
OFF	OFF	ON	ON	OFF	OFF	±100Mv	50Mv
ON	OFF	OFF	OFF	ON	OFF	±250Mv	125Mv
OFF	ON	OFF	OFF	ON	OFF	±500Mv	250Mv
OFF	OFF	ON	OFF	ON	OFF	±1V	500Mv
ON	OFF	OFF	OFF	OFF	OFF	±2.5V	1.25Mv
OFF	ON	OFF	OFF	OFF	OFF	±5V	2.5Mv
OFF	OFF	ON	OFF	OFF	OFF	±10V	5Mv

表-4-4-16 SM235-0KD 单/双极性、增益和衰减开关配置

SM235 开关						单/双极性选择	增益选择	衰减选择
SW1	SW2	SW3	SW4	SW5	SW6			
---	---	---	---	---	ON	单极性	---	---

---	---	---	---	---	OFF	双极性	---	---
---	---	---	OFF	OFF	---	---	x1	---
---	---	---	OFF	ON	---	---	x10	---
---	---	---	ON	OFF	---	---	x100	---
---	---	---	ON	ON	---	---	无	---
ON	OFF	OFF	---	---	---	---	---	0.8
OFF	ON	OFF	---	---	---	---	---	0.4
OFF	OFF	ON	---	---	---	---	---	0.2

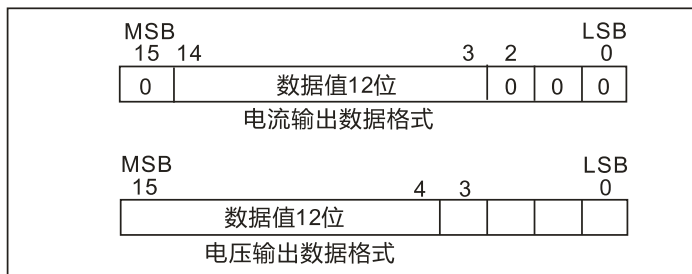
输入数据格式



提示

模数转换器（ADC）的 12 位读数，其数据格式是左端对齐的。最高有效位是符号位（0 表示正数），对单极性格式，3 个连续的 0 使得 ADC 计数值每变化 1 个单位，则数据字的变化是以 8 为单位变化的。对双极性格式，4 个连续的 0 使得 ADC 计数值每变化 1 个单位，则数据字的变化是以 16 为单位变化的。

输出数据格式



提示

数模转换器（DAC）的 12 位读数，其输出数据格式是左端对齐的，最高有效位是符号位（0 表示是正数），数据在装载到 DAC 寄存器之前，4 个连续的 0 是被裁断的，这些位不影响输出信号值。

4.5 温度测量模块规范

CTH200 系列 PLC 支持混合温度输入模块，具体信息如下：

表 4-5-1 混合温度输入模块订货数据

规格参数	订货号
SM231 热电阻温度输入模块，2 点 RTD，隔离型 16 位精度	CTH2 231-7PB32
SM231 热电阻温度输入模块，4 点 RTD，隔离型 16 位精度	CTH2 231-7PC32

SM231 热电偶温度输入模块, 4 点 TC, J/K/R/S/T/E/N, 隔离型 16 位精度	CTH2 231-7PD32
SM231 热电偶温度输入模块, 8 点 TC, J/K/R/S/T/E/N, 隔离型 16 位精度	CTH2 231-7PF32
SM231 混合温度输入模块, 2NTC/PT100, 2 点 0~20Ma 电流或±5V, ±10V, 0~10V, 0~5V 电压输入, 隔离型 16 位精度	CTH2 231-7ND32
SM231 热电阻温度输入模块, 8NTC/PT100, 隔离型 16 位精度	CTH2 231-7NF32

4.5.1 热电阻测量模块规范

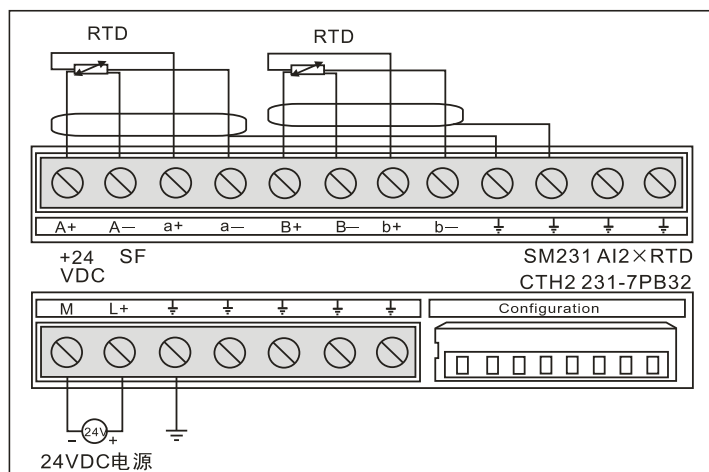
表 4-5-2 RTD 模块技术规范

特性	SM231: 2AI×RTD	SM231: 4AI×RTD
订货号	CTH2 231-7PB32	CTH2 231-7PC32
物理特性		
尺寸 (宽×高×深)	71.3 × 96 × 62mm	
LED 指示灯	24VDC 电源指示灯, ON=无错, OFF=无 24VDC 电源; SF 指示灯, ON=模块故障; 闪烁=超量程或断线, OFF=无错	
电源损耗		
+5VDC 消耗电流	87Ma	
L+	34Ma	
L+线圈电压范围	20.4~28.8VDC	
功率损耗	1.7W	
输入特性		
输入类型	模块参考接地热电阻	
输入点数	2	4
输入范围	热电阻类型 (选一种): 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	
温度测量范围	Pt-100Ω, 200Ω, 500Ω, 1000Ω: -200°C~850°C Pt-10000Ω: -200°C~600°C NI-0.00672: -80°C~260°C NI-0.006178: -60°C~300°C Cu-0.004270: -200°C~260°C <备注> 测量结果超范围则报错, 诊断详情参考表4-31。	
隔离		
现场至逻辑	500VAC	
现场至 24VDC	500VAC	
24V 至逻辑	500VAC	
共模输入范围 (输入通道至输入通道)	0	
共模抑制	>120Db@120VAC	

采样特性		
温度分辨率	0.1°C/0.1°F	
电压分辨率	15位加符号位	
测量原理	Σ-Δ	
模块更新时间(所有通道)	425ms	825ms
到传感器的导线长度	最大100米	
导线回路电阻	20Ω, 铜线型2.7Ω	
噪声抑制	85Db@ 50Hz/60Hz/400Hz	
数据字格式		
输入阻抗	>10MΩ	
最大输入电压	30VDC (检测), 5VDC (源)	
分辨率	15位+符号位	
输入滤波衰减	-3Db@21kHz	
基本误差	0.1% FS (电阻)	
重复性	0.05% FS	
模拟输入寻址方式	AIW寻址	

接线图

SM231 热电阻输入模块 2AI x RTD (CTH2 231-7PB32)



SM231 热电阻输入模块 4AI x RTD (CTH2 231-7PC32)

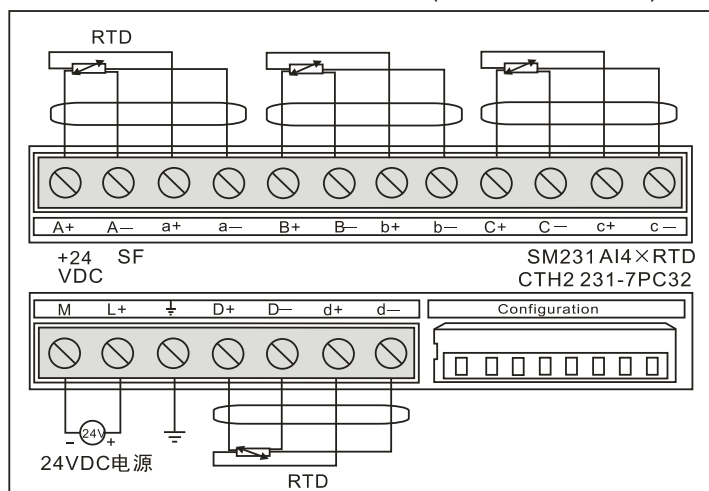


表 4-5-3 热电阻模块诊断信息

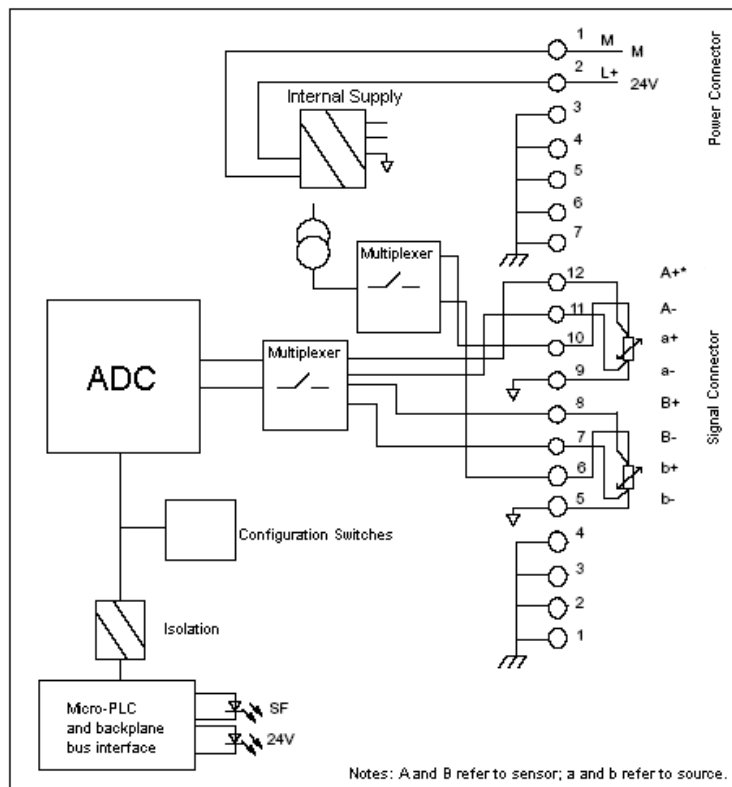
出错类型	通道数据	SF 指示灯	24V 指示灯	范围状态位	24V 电源故障
无模块电源	32766	熄灭	熄灭	0	1
断线	32767 (正向标定) -32768 (负向标定)	闪烁	亮起	1	0
超出温度范围	32767 (正向标定) -32768 (负向标定)	闪烁	亮起	1	0

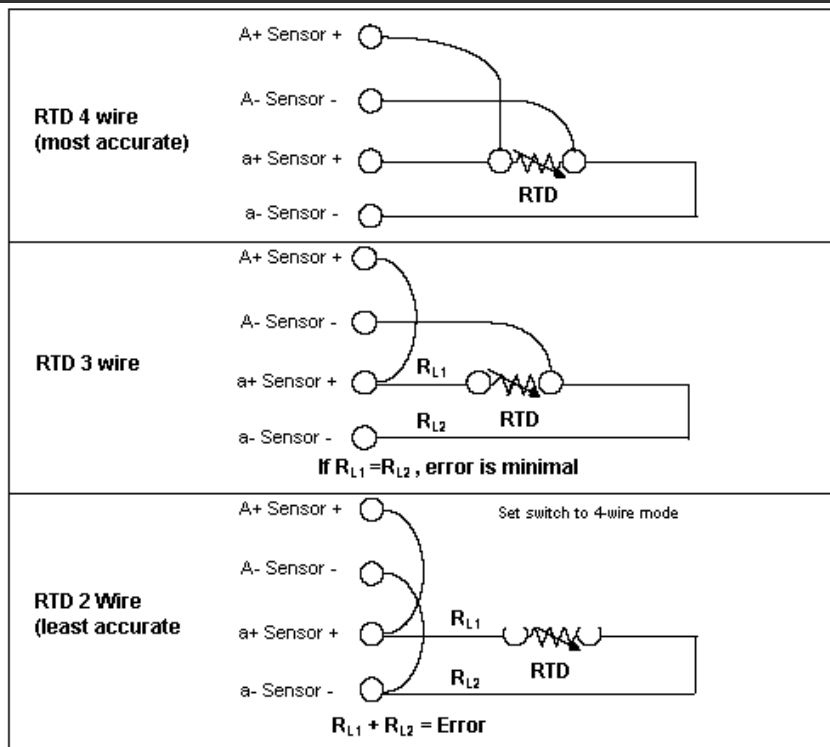
<备注> 范围状态位是模块出错寄存器字节中的位，可以根据模块顺序查看 SMB8~21，即可查询对应模块的错误代码。

表 4-5-4 SMB8~21 诊断信息

SMB8	模块 1 标识寄存器
SMB9	模块 1 错误寄存器
SMB10	模块 2 标识寄存器
SMB11	模块 2 错误寄存器
SMB12	模块 3 标识寄存器
SMB13	模块 3 错误寄存器
SMB14	模块 4 标识寄存器
SMB15	模块 4 错误寄存器
SMB16	模块 5 标识寄存器
SMB17	模块 5 错误寄存器
SMB18	模块 6 标识寄存器
SMB19	模块 6 错误寄存器
SMB20	模块 7 标识寄存器
SMB21	模块 7 错误寄存器

RTD 与传感器的接线





Notes: A refers to sensor; a refers to source

用户可以直接将热电阻传感器直接接到 CTH200 的 SM231 热电阻模块上，也可使用扩展接线方式。使用屏蔽线可达到最好的抗噪性，如果用户使用屏蔽线，应将屏蔽线接到信号连接器的 1 至 4 针接地点上。该接地点与电源连接器的 3 至 7 针共地。如果有的热电阻输入通道没有使用，用户应将一个电阻器与未使用的输入通道相连，以防止由于浮地输入信号产生的误差影响有效通道的错误显示。

用户需将电源连到电源连接器的 1 和 2 针上。用户必须将电源连接器的针 3 接到附近的机壳地。用户可任选三种接线方式之一将热电阻模块与传感器相连。精度最高的是 4 线接法，精度最低的是 2 线接法，建议只有在用户应用中不在乎导线误差时才用 2 线接法。

【选择热电阻类型】

SM231 热电阻模块使 CTH200 系列 PLC 能方便地与多种热电阻传感器连接。用户可以通过 DIP 开关来选择热电阻的类型、接线方式、测量单位和开路故障的方向。连接到同一个模块上的所有热电阻必须是同类型热电阻。

DIP 选择开关位于模块的右下部，如下图所示。为使 DIP 开关设置起作用，用户需要给 PLC 或用户 24V 电源断电再通电。

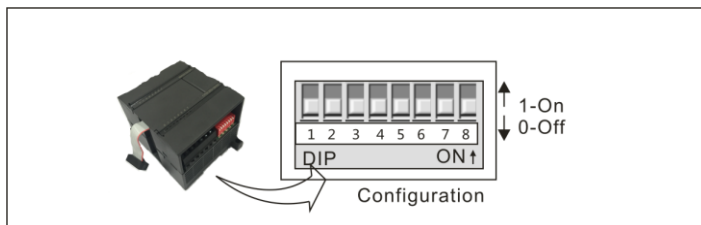


表 4-5-5 SM231 热电阻模块 DIP 开关配置

热电阻类型	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
保留	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	

表 4-5-6 断线检测标定方向、测量单位和热电阻接线配置

SW6	标定方向	SW7	测量单位	SW8	接线方式
0	正标定 (+3276.7 度)	0	摄氏度 (°C)	0	3 线
1	负标定 (-3276.8 度)	1	华氏度 (°F)	1	2 线或 4 线

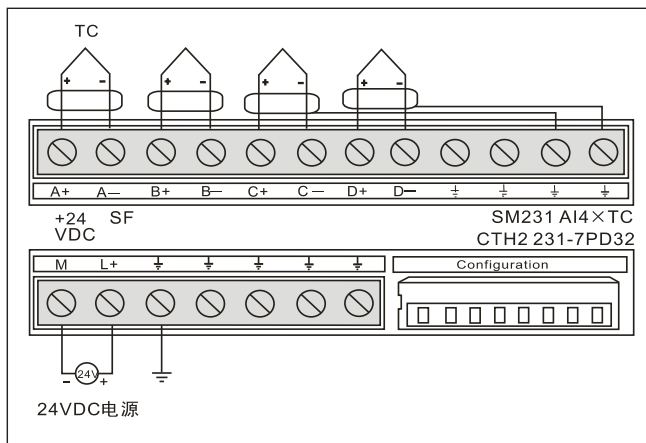
4.5.2 热电偶测量模块规范

表 4-5-7 SM231 热电偶输入模块规范

规格项目	SM231: 4AI×TC	SM231: 8AI×TC
物理特性		
尺寸（宽×高×深）	71.3 × 96 × 62mm	
LED 指示灯	24VDC电源指示灯，ON=无错，OFF=无24VDC电源；SF指示灯，ON=模块故障；闪烁=超量程或断线，OFF=无错	
电源损耗		
+5VDC 消耗电流	87Ma	
L+	30Ma	
L+线圈电压范围	20.4~28.8VDC	
功率损耗	1.7W	
输入特性		
输入类型	悬浮型热电偶	
输入点数	4	8
输入范围	热电偶类型（选一种）：S, T, R, E, N, K, J 电压范围：+/-80mV <备注> 有关温度测量范围，请参考章节4.5.2 热电偶测量模块规范。测量结果超量程则报错，诊断详情参考表4-5-8。	
隔离特性		
现场至逻辑	500VAC	
现场至24VDC	500VAC	
24V至逻辑	500VAC	
共模输入范围（输入通道至输入通道）	120VAC	
共模抑制	>120Db@120VAC	
采样特性		
温度分辨率	0.1°C/0.1°F	
电压分辨率	15位加符号位	
转换原理	Σ-Δ	
模块更新时间(所有通道)	425ms	825ms
到传感器的导线长度	最大100m	
导线回路电阻	最大100Ω	
噪声抑制	85Db@ 50Hz/60Hz/400Hz	
数据字格式	电压：-27648 至+27648	
输入阻抗	>1MΩ	
最大输入电压	30VDC	
分辨率	15位加符号位	
输入滤波衰减	-3Db@ 21kHz	
基本误差	0.1% FS（电压）	
重复性	0.05% FS	
冷接点误差	±1.5°C	
模拟输入寻址方式	AIW寻址	VW寻址

接线图

SM231 热电偶输入模块(CTH2 231-7PD32) x 4TC



SM231 热电偶输入模块(CTH2 231-7PF32) x 8TC

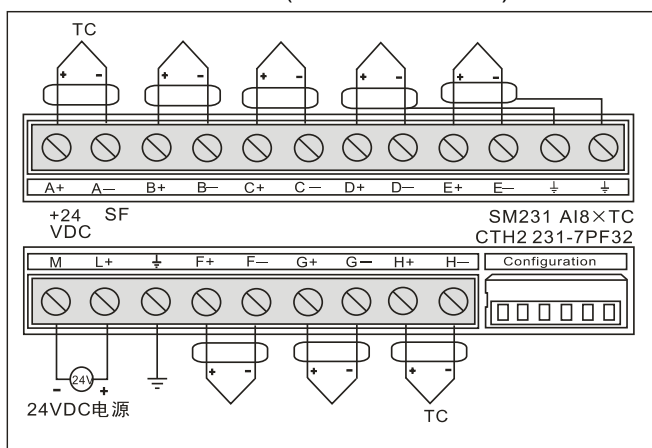


表 4-5-8 热电偶模块诊断信息

出错类型	通道数据	SF 指示灯	24V 指示灯	范围状态位	24V 电源故障
无模块电源	32766	熄灭	熄灭	0	1
断线	32767 (正向标定) -32768 (负向标定)	闪烁	亮起	1	0
超出温度范围	32767 (正向标定) -32768 (负向标定)	闪烁	亮起	1	0

<备注> 范围状态位是模块出错寄存器字节中的位，可以根据模块顺序查看 SMB8~21，即可查询对应模块的错误代码。

表 4-5-9 SMB8~21 诊断信息

模块 1 标识寄存器	SMB8/SMB9
模块 2 标识寄存器	SMB10/SMB11
模块 3 标识寄存器	SMB12/SMB13
模块 4 标识寄存器	SMB14/SMB15
模块 5 标识寄存器	SMB16/SMB17
模块 6 标识寄存器	SMB18/SMB19
模块 7 标识寄存器	SMB20/SMB21

各种类型热电偶的温度范围 (°C) 和准确度

数据字 (1个数字位=0.1°C)		类型J	类型K	类型T	类型E	类型R,S	类型N	±80mV	
十进制	十六进制								
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
:	:							80.0029mV	
27649	6C01							80mV	
27648	6C00								
:	:								
17680	4510					1768.0°C			
:	:								
13720	3598		1372.0°C 超出范围						
:	:								
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	
:	:								
-500	FE0C					低于范围 -50.0°C			
-1500	FA24	-150.0°C							
:	:								
-2000	F830	低于范围	-200.0°C						
:	:								
-2100	F7CC	-210.0°C	低于范围						
:	:								
-2550	F60A			-255.0°C	-255.0°C				
:	:			低于范围	低于范围				
-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
全量程范围的精度		S0.1%	S0.3%	S0.6%	S0.1%	S0.6%	S0.1%	S0.1%	
精度(无冷端补偿的额定范围)		S1.5°C	S1.7°C	S1.4°C	S1.3°C	S3.7°C	S1.6°C	S0.10°C	
冷端误差		S1.5°C	S1.5°C	S1.5°C	S1.5°C	S1.5°C	S1.5°C	NA	

*OF=上溢 OR=超出范围 NR=额定范围 VR=低于范围 UF=下溢

↑表示所有大于该值但小于断线阈值的模拟量都报告为上溢值, 32767 (10x7FFF)

↓表示所有小于该值但大于断线阈值的模拟量都报告为下溢值, -32768 (0x8000)

图 4-16 热电偶测量范围

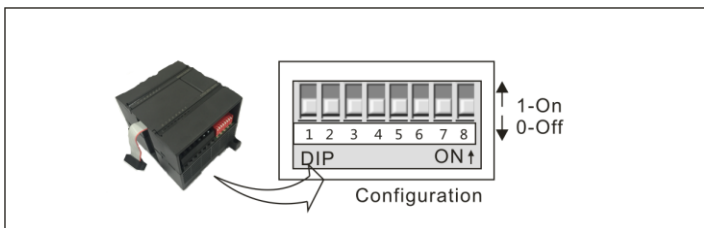
开关组态与软件配置

● **DIP 开关配置**

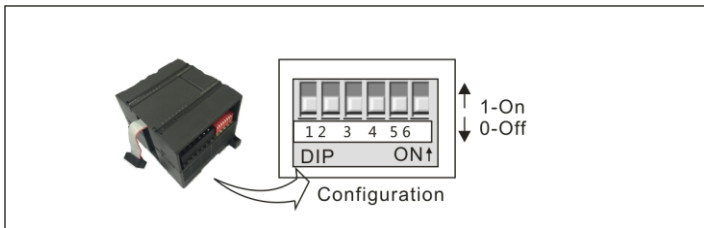
SM231 热电偶模块提供一个方便的，隔离的接口，用于七种热电偶类型：J，K，E，N，S，T，和 R 型，它也允许连接微小的模拟量信号（±80Mv 范围），所有连到模块上的热电偶必须是相同类型，且最好使用隔离型热电偶传感器。

热电偶模块需要用户通过 DIP 开关进行选择的有：热电偶的类型、断线检查、测量单位、冷端补偿和开路故障方向，用户可以很方便地通过位于模块下部（如下图所示）的组态 DIP 开关进行以上选择。

对于 SM231 4TC 模块，SW1~SW3 用于选择热电偶类型，SW4 没有使用（要求设置到 OFF 的位置），SW5 用于选择断线检测方向，SW6 用于选择是否进行断线检测，SW7 用于选择测量单位，SW8 用于选择是否进行冷端补偿。



对于 SM231 8TC 模块，SW1~SW3 用于选择热电偶类型，SW4 用于选择断线检测方向，SW5 用于选择测量单位，SW6 用于选择是否进行冷端补偿。而是否进行断线检测则固定设置为是，无需用户设置。



为了使 DIP 开关设置生效，用户需要给 PLC 和/或用户的电源断电再重新上电。

表 4-5-10 热电偶模块 DIP 配置：

热电偶类型	SW1	SW2	SW3
J（默认）	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

表 4-5-11 SM231 热电偶模块其它特性

项目	SM231 4TC		SM231 8TC	
	开关位置	设置	开关位置	设置
断线检测方向	SW5	0: 正标定 (+3276.7 度)	SW4	0: 正标定 (+3276.7 度)
		1: 负标定 (-3276.8 度)		1: 负标定 (-3276.8 度)
是否进行断线检测	SW6	0: 是, 1: 否	固定为进行断线检测	
测量单位选择	SW7	0: 摄氏度, 1: 华氏度	SW5	0: 摄氏度, 1: 华氏度
是否进行冷端补偿	SW8	0: 是, 1: 否	SW6	0: 是, 1: 否

● 软件配置

对于 SM231 8AIxTC 热电偶输入模块, 其读数是在 VW 中, 而不是 AIW, 模块所处的相对位置不同, 对应的地址也不同。地址计算公式如下:

$$x(VWx) = \text{槽位号} \times 64 + \text{输入通道号} \times 2$$

槽位号对应模块的安装位置, 紧靠 CPU 的第一个扩展模块槽位号为 0, 第二个扩展模块槽位号为 1, 依此类推。输入通道共 8 路, 从 A 至 H, 对应的编号为 0 至 7。表 4-5-12 为基于以上公式计算得到的地址速查表。

表 4-5-12 地址速查表

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



提示

由于 COTRUST 的 TD2X 文本显示器和 SIEMENS 的 TD200 文本显示面板所分配的地址固定为 VW0, 因而如果您的系统中需要同时使用到 TD2X 或 TD200 和 SM231 8TC 模块, 则 SM231 8TC 模块不能安装在第一个位置 (Slot 0), 否则将不能正常工作。

4.5.3 混合温度输入模块规范

CTH2 231-7ND32 模块规范

SM231 NTC 混合模拟量输入扩展模块是 CTH200 系列 PLC 的模拟量扩展模块, 提供 4 通道模拟量采集, 其中两通道用于连接热敏电阻 NTC 温度传感器或热电阻 PT100 温度传感器, 另外两个通道用于采集电压/电流信号输入, 所有通道的输入精度 (含符号位) 均为 16BIT。主要用于灭菌设备或中央空调设备等既有温度测量需求又有压力信号测量需求的场合。

表 4-5-13 SM231-7ND 模块技术规范

订货号	CTH2 231-7ND32
物理特性	
尺寸（宽×高×深）	71.3 × 96 × 62mm
LED指示灯	24VDC电源指示灯，ON=无错，OFF=无24VDC电源； SF指示灯，ON=模块故障；闪烁=超量程或断线，OFF=无错
电源损耗	
+5VDC 消耗电流	87Ma
L+	60Ma
L+线圈电压范围	20.4~28.8VDC
功率损耗	1.7W
输入信号	
热电阻/热敏电阻输入范围	热电阻类型（任选一种）： Pt-100（3850ppm，3920ppm，3850.55ppm，3916ppm，3902ppm） NTC（R25=10K ω /B=3950，R25=10K ω /B=3435）
温度测量范围	Pt-100：-50℃~850℃ NTC（R25=10K，B=3950）：-40℃~120℃ NTC（R25=10K，B=3435）：-40℃~150℃ <备注> 若超出测量范围则报错，诊断详情参考表4-5-8。
电压输入	0V~5V，0V~10V， \pm 5V， \pm 10V
电流输入	0~20Ma
输入点数	2PT100/2NTC和2AI
隔离特性	
现场至逻辑	500VAC
现场至 24VDC	500VAC
24V 到逻辑	500VAC
共模抑制	>120Db@120VAC
采样特性	
温度分辨率	0.1℃/0.1°F
电压分辨率	15位+符号位
转换原理	Σ - Δ
模块更新时间（所有通道）	425ms
到传感器的导线长度	最大100m
导线回路电阻	最大20 Ω
噪声抑制	85Db@50Hz/60Hz/400Hz
数据字格式	温度（NTC：R25=10K ω ，B=3950K）：-400~1200（仅限通道1、2） 温度（NTC：R25=10K ω ，B=3435K）：-400~1500（仅限通道1、2） 温度（PT100）：-500~2000（仅限通道1、2） 电压/电流：单极性0~32000，双极性-32000~+32000（仅限通道3、4）
输入阻抗	电压输入>10M Ω ；电流输入=250 Ω ；NTC输入>10M Ω
最大输入电压	30VDC（检测），5VDC（源）
分辨率	0.1℃/0.1°F
输入滤波衰减	-3Db@21kHz
基本误差	0.1%FS（电阻）
重复性	0.05%FS
模拟输入寻址方式	AIW寻址

应用环境

- 工作温度：水平安装 0-55℃，垂直安装 0-45℃
- 工作湿度：95%非冷凝湿度

使用方法

SM231 NTC 模块用于扩展 CTH200 CPU 的模拟量测量能力，通过总线接口与 CPU 模块连接。

表 4-5-14 SM231-7ND 模块配置设置

输入类型	SW1	SW2	SW3	SW4	SW5
	通道1、2有效			通道3、4有效	
100ΩPT0.003850 (出厂设置)	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=10K ω , B=3950K	1	0	1	—	—
NTC :R25=10K ω , B=3435K	1	1	0	—	—
禁用	1	1	1	—	—
0—5V	—	—	—	0	0
0—20Ma	—	—	—	0	0
0—10V	—	—	—	0	1
-10V—10V	—	—	—	1	0
-5V—5V	—	—	—	1	1

SW6	标定方向 (所有通道有效)	SW7	测量单位	SW8	接线方式
0	正标定 (+3276.7度)	0	摄氏度 (°C)	0	3线
1	负标定 (-3276.8度)	1	华氏度 (°F)	1	2线/4线

【启动步骤】

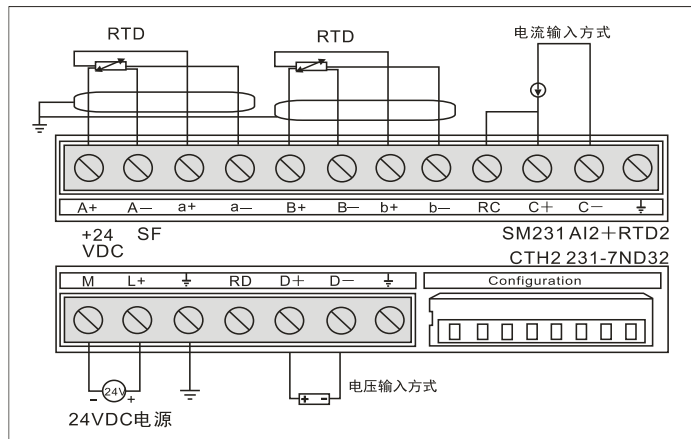
- 1) 按照接线端子图所示在 L+和 M 端子接入 24VDC 电源。
- 2) 通过扩展 I/O 总线连接到 CPU 模块。
- 3) 连接 CPU 电源和通信端口。
- 4) 按照接线端子图连接输入信号，空闲通道的端子应悬空，接地端子 (EARTH) 应按接地规范连接大地 (单点接地)。
- 5) 按照配置设置要求配置传感器类型、标定方向等。
- 6) 开启 CPU 和模块的供电电源。

【获取输入值】

- 用户可在用户程序或编程监控软件 (如: 本公司的 MagicWorks PLC 软件或西门子的组态软件 MicroWIN) 中读取每个通道 2 字节的输入数据:
- NTC 满量程读数为-400~1200 (R25=10K B=3950), 对应温度范围: -40.0~120.0℃; 或者 -400~1500 (R25=10K B=3435), 对应温度范围: -40.0~150.0℃; PT100 满量程读数为-500~2000, 对应温度范围: -50.0~200.0℃; 超量程或断线时返回读数由标定方向配置开关决定 (-32768 或 32767)。
- 电压/电流满量程读数为-32000~32000, 超量程直到-32767/32765 仍为有效读数。
- 无用户电源所有通道读数均为 32766。

接线图

SM231 混合输入模块(CTH2 231-7ND32)



CTH2 231-7NF32 模块规范

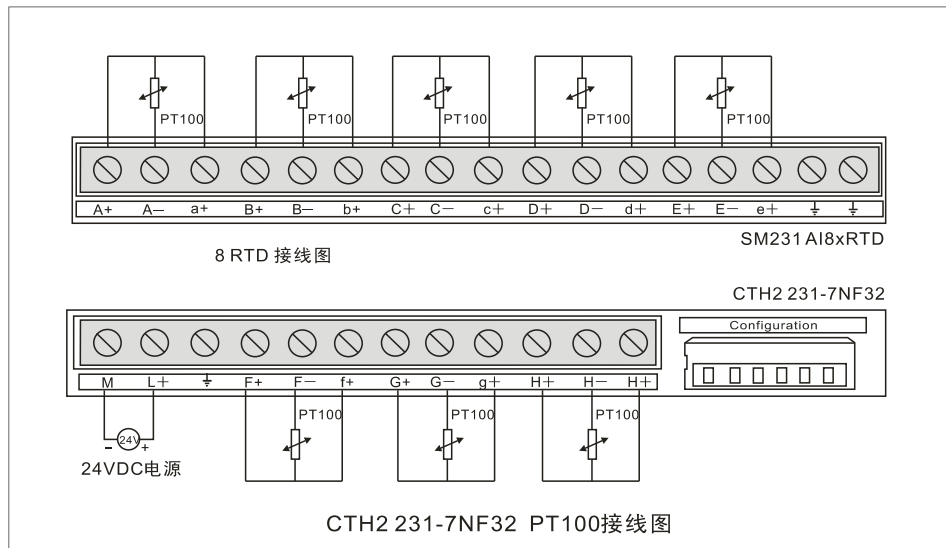
表 4-5-15 SM231-7NF 模块规范

项目	CTH2 231-7NF32
尺寸 (W x H x D)	71.3 x 96 x 62mm
LED指示灯	24VDC电源指示灯, ON=24VDC供电正常, OFF=无24VDC供电; SF指示灯, ON=模块故障, 闪烁=超量程或断线, OFF=无错;
电源损耗	
+5VDC消耗电流	87Ma
L+	32.5Ma
L+电压范围	20.4~28.8VDC
功率损耗	1.8W
输入信号	
输入类型	Pt100: $\alpha=3850PPm/3920PPM/3850.55PPM/3916PPM/3902PM$ NTC: R25=10k/B=3950或R25=10k/B=3435
温度测量范围	Pt100: -50°C~800°C NTC (R25=10k/B=3950): -40°C~120°C NTC (R25=10k/B=3435): -40°C~150°C <备注> 若超出测量范围则报错, 诊断详情参考表4-5-2。
输入点数	8PT100/8NTC
插入式 I/O 端子	是
电源是否隔离	是
现场侧-逻辑	500VAC
现场侧-直流 24 V	500VAC
直流 24 V-逻辑	500VAC
共模抑制	120Db@120VAC
采样特性	
温度分辨率	0.1°C/0.1°F
测量原理	$\Sigma-\Delta$
模块更新时间 (所有通道)	825ms
最大电缆长度	100m
最大电缆环路电阻	20 Ω
噪声抑制	85Db@50/400 Hz
输入阻抗	>1K Ω

测量精度	±0.3℃
断线检测	Pt100检测3条导线（NTC检测2条导线），检测时间最长3分钟

接线图

SM231 温度输入模块 Pt100 (CTH2 231-7NF32)



SM231 温度输入模块 NTC (CTH2 231-7NF32)

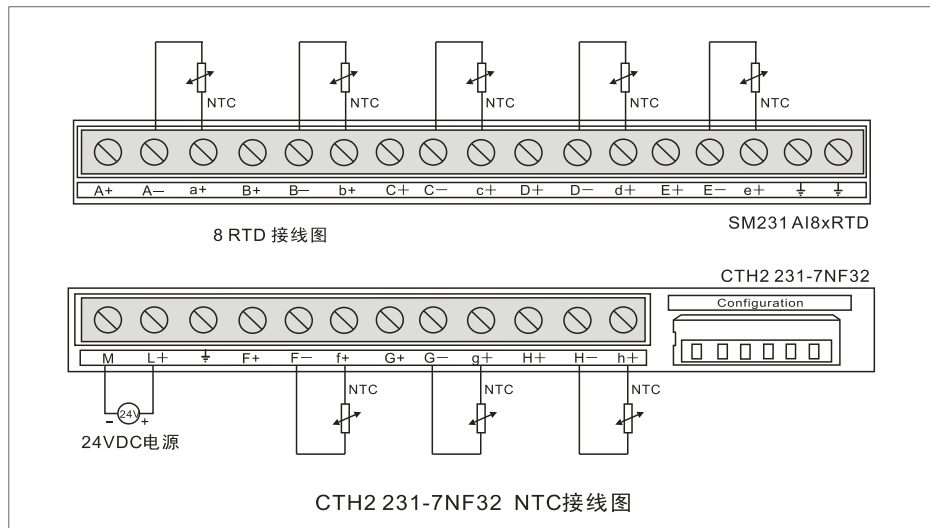


表 4-5-16 SM231-7NF DIP 开关配置

输入类型	SW1	SW2	SW3
100Ω Pt 0.003850（默认）	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
禁用	1	1	1

SW4	标定方向	SW5	测量单位	SW6	备用
0	正标定（+3276.7度）	0	摄氏度（℃）	0	无影响

1	负标定 (-3276.8 度)	1	华氏度 (°F)	1	无影响
---	-----------------	---	----------	---	-----

软件配置

对于 SM231 8AIx16 位模拟量输入模块，其读数是在 VW 中，而不是 AIW，模块所处的相对位置不同，对应的地址也不同。地址计算公式如下：

$$x(VWx) = \text{槽位号} \times 64 + \text{输入通道号} \times 2$$

槽位号对应模块的安装位置，紧靠 CPU 的第一个扩展模块槽位号为 0，第二个扩展模块槽位号为 1，依此类推。输入通道共 8 路，从 A 至 G，对应的编号为 0 至 7。表 4-5-17 为基于以上公式计算得到的地址速查表。

表 4-5-17 地址速查表

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



提示

由于 COTRUST 的 TD2X 文本显示器和 SIEMENS 的 TD200 文本显示面板所分配的地址固定为 VW0，因而如果您的系统中需要同时使用到 TD2X 或 TD200 和 SM231 8AI 模块，则 SM231 8AI 模块不能安装在第一个位置（Slot 0），否则将不能正常工作。

4.6 PID 控制模块规范

表 4-6-1 PID 控制模块订货数据

规格参数	订货号
SM231 热电偶 PID 模块，4 点 J/K 型，带智能 PID，隔离型 16 位精度	CTH2 231-7TD32
SM231 热电偶 PID 模块，8 点 J/K 型，带智能 PID，隔离型 16 位精度	CTH2 231-7TF32
SM231，8 路电流输入，0-20Ma/4-20Ma，带智能 PID，隔离型 16 位精度	CTH2 231-7HF32

【主要特性】

- 总线、电源、通道间全隔离，可靠性高，抗干扰能力强。
- 16 位采样精度，采用硬件滤波技术，测量值更加准确稳定。
- 供电电源有反接保护和浪涌吸收功能，适用于恶劣的工业环境。
- 集成先进的模糊逻辑控制算法，不占用 CPU 资源，无需编程即可实现准确的温度控制，动态性能好。
- PID 控制输出可以是 PWM 或模拟量，双极性输出，可以控制加热和冷却。

【使用规范】

- 应采用绝缘型热电偶以获得理想的抗干扰能力，提高可靠性。
- 信号线应采用屏蔽线，屏蔽线需单端接地。

- 在系统有良好接地的情况下模块的接地端应接到地线上，否则不接地。
- 未使用到的通道请短接以消除断线故障告警。

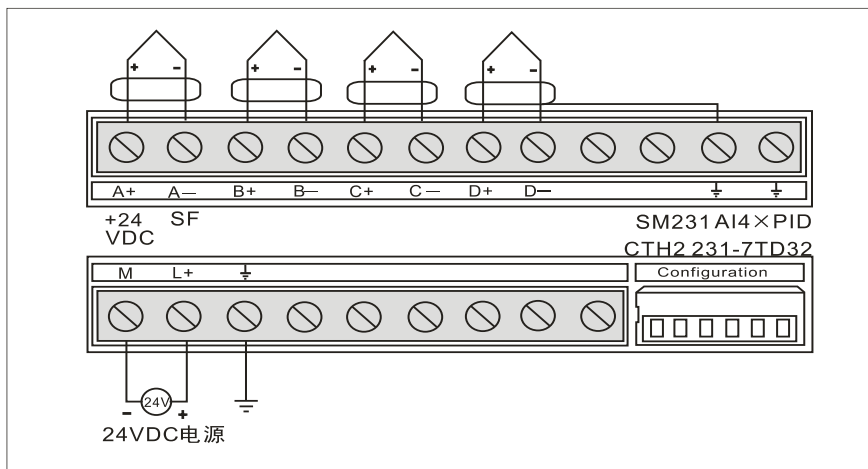
表 4-6-2 SM231-7TD/7TF 热电偶 PID 模块技术规范

特性	SM231 4AIxTC PID	SM231 8AIxTC PID
订货号	CTH2 231-7TD32	CTH2 231-7TF32
物理特性		
尺寸（宽×高×深）	71.3 × 96 × 62mm	71.3 × 96 × 62mm
LED 灯指示	24VDC电源指示灯，ON=无错，OFF=无24VDC电源；SF指示灯，ON=模块故障，闪烁=超量程或断线，OFF=无错	
电源损耗		
+5VDC消耗电流	87Ma	87Ma
L+电流	34Ma	39Ma
L+线圈电压范围	20.4~28.8VDC	
功率损耗	1.8W	1.8W
输入特性		
输入类型	悬浮型热电偶	
输入范围	K 型热电偶	
输入点数	4	8
隔离特性		
现场至逻辑	500VAC	
现场至 24VDC	500VAC	
24V至逻辑	500VAC	
共模输入范围（输入通道至输入通道）	120VAC	
共模抑制	>120Db@120VAC	
采样性能		
温度分辨率	0.1°C/0.1°F	
电压分辨率	15位加符号位	
测量原理	Σ-Δ	
模块更新时间（所有通道）	405ms	825ms
到传感器的导线长度	最大100m	
导线回路电阻	最大100Ω	
噪声抑制	85Db@ 50Hz/60Hz/400Hz	
数据字格式	电压：-27648~+27648	
输入阻抗	>1MΩ	
最大输入电压	30VDC	
分辨率	15 位+符号位	
输入滤波衰减	-3Db@ 21kHz	
基本误差	0.1% FS（电压）	
重复性	0.05% FS	
冷接点误差	±1.5°C	
诊断程序	LED: EXTF, SF	
PID特性		
PID算法	PID+FUZZY参数自调整	
采样时间	1s	
输出最小脉宽	10ms	
PID类型	P、PI、PD、PID型	
PID输出类型	模拟量或PWM脉宽控制	

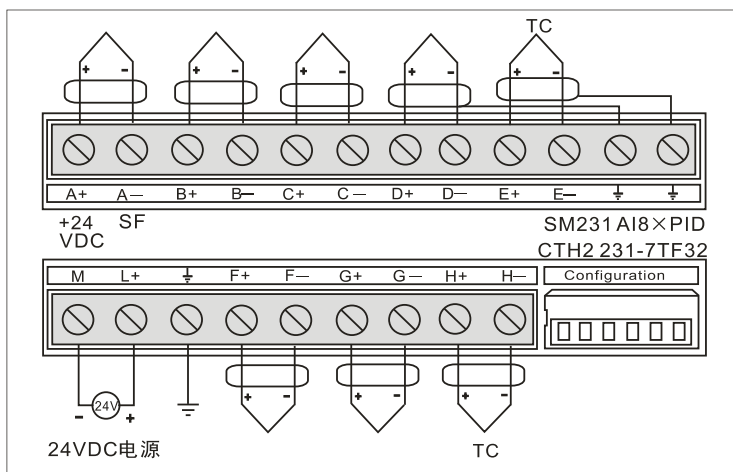
PID输出极性	双极或单极
---------	-------

接线图

SM231 热电偶 PID 模块(CTH2 231-7TD32)



SM231 热电偶 PID 模块(CTH2 231-7TF32)



PID 地址与参数配置

表 4-6-3 PID 地址计算公式

地址名称	计算公式	备注
PID 参数地址	$A = (2048 + S * 256) + 16 * C$	S为模块所在的槽号（范围：0~6），C为通道号，SM231-7TF为0~7，SM231-7TD为0~3
PID 正向脉冲输出地址	$X = (2048 + S * 256) + 12$	
PID 负向脉冲输出地址	$Y = (2048 + S * 256) + 13$	

表 4-6-4 PID 参数输出部分（模块到 CPU）

内容	地址	数值设置范围	实际对应数值
实际温度	VW A	-2000~13000	-200~1300度
状态字	VW A+2		-
PID 模拟量输出	VW A+4	-32000~32000	-

表 4-6-5 PID 参数输入部分 (CPU 到模块)

内容	地址	数值设置范围	实际对应数值
设定温度	VW A+128	-2000~13000	-200~1300度
控制字节		VB A+130位等于0时	VB A+130位等于1时
	V (A+130) .0	PID不运行, 无输出	PID运行
	V (A+130) .1	积分一直起作用, 比例系数 Kp不自动调整	积分分离及比例系数自动调整
	V (A+130) .2	PID 单极输出, 0 ~ 32000	PID 双极输出, -32000 ~ 32000, 具有加热和冷却功能
	V (A+130) .3	未使用	
	V (A+130) .4	积分起作用	积分不起作用
	V (A+130) .5	微分起作用	微分不起作用
	V (A+130) .6	实际温度值滤波, 抗干扰更强	实际温度值不滤波
PID 脉冲输出周期设定	VW A+132	1~255	1~255s
Kp (比例系数)	VW A+134	0~9999	0~999.9
Ti (积分时间)	VW A+136	0~3600	0~3600s
Td (微分时间)	VW A+138	0~3600	0~3600s
Range(量程范围)	VW A+140	0~13000	0~1300 度

表 4-6-6 正向脉冲输出地址

脉冲输出通道	脉冲输出地址	
0	V X.0	V Y.0
1	V X.1	V Y.1
2	V X.2	V Y.2
3	V X.3	V Y.3
4	V X.4	V Y.4
5	V X.5	V Y.5
6	V X.6	V Y.6
7	V X.7	V Y.7

表 4-6-7 负向脉冲输出地址

脉冲输出通道	脉冲输出地址
0	V Y.0
1	V Y.1
2	V Y.2
3	V Y.3
4	V Y.4
5	V Y.5
6	V Y.6
7	V Y.7

应用举例：

计算第二个扩展模块上的 SM231-7TF 的最后一个 PID 回路的地址。

首先计算地址：S=1，C=7

A 地址， $A=2048 + 1 * 256 + 16 * 7 = 2416$

X 地址， $X=2048 + 1 * 256 + 12 = 2316$

Y 地址， $Y=2048 + 1 * 256 + 13 = 2317$

然后根据以下参数地址说明设定或读出参数内容：

VW2544	//设定温度
VB2546	//控制字（参数自调整、双极输出）
VW2548	//脉冲输出周期
VW2550	//Kp比例系数
VW2552	//Ti积分时间（秒）
VW2554	//Td微分时间（秒）
VW2416	//实际温度
VW2418	//状态字
VW2420	//PID模拟量输出
V2316.7	//正向脉冲输出
V2317.7	//负向脉冲输出

为了保证 PID 模块能正常使用，编写其他程序块时请一定不要使用您所用的 PID 模块占用的 V 存储区。可调用 SM231 PID 专用的参数配置程序库。

模块在第 0 号插槽所占用的地址为：VW2048 到 VW2298

模块在第 1 号插槽所占用的地址为：VW2304 到 VW2554

模块在第 2 号插槽所占用的地址为：VW2560 到 VW2810

模块在第 3 号插槽所占用的地址为：VW2816 到 VW3066

模块在第 4 号插槽所占用的地址为：VW3072 到 VW3322

模块在第 5 号插槽所占用的地址为：VW3328 到 VW3578

模块在第 6 号插槽所占用的地址为：VW3584 到 VW3834

参数配置梯形图如下：

PID模块的参数地址占用CPU的V存储区。

为了保证PID模块能正常使用，编写其他程序块时请一定不要使用您所用的PID模块占用的V存储区。

模块在第0号插槽所占用的地址为：VW2048到VW2298

模块在第1号插槽所占用的地址为：VW2304到VW2554

模块在第2号插槽所占用的地址为：VW2560到VW2810

模块在第3号插槽所占用的地址为：VW2816到VW3066

模块在第4号插槽所占用的地址为：VW3072到VW3322

模块在第5号插槽所占用的地址为：VW3328到VW3578

模块在第6号插槽所占用的地址为：VW3584到VW3834

网络 1

设置第1个扩展模块（插槽号为0）上的EM231 PID模块第1个PID回路（通道号为0）的参数：

假设该回路调试完成后需要再改变PID参数，我们调用PIDSetting来设定该回路的参数。

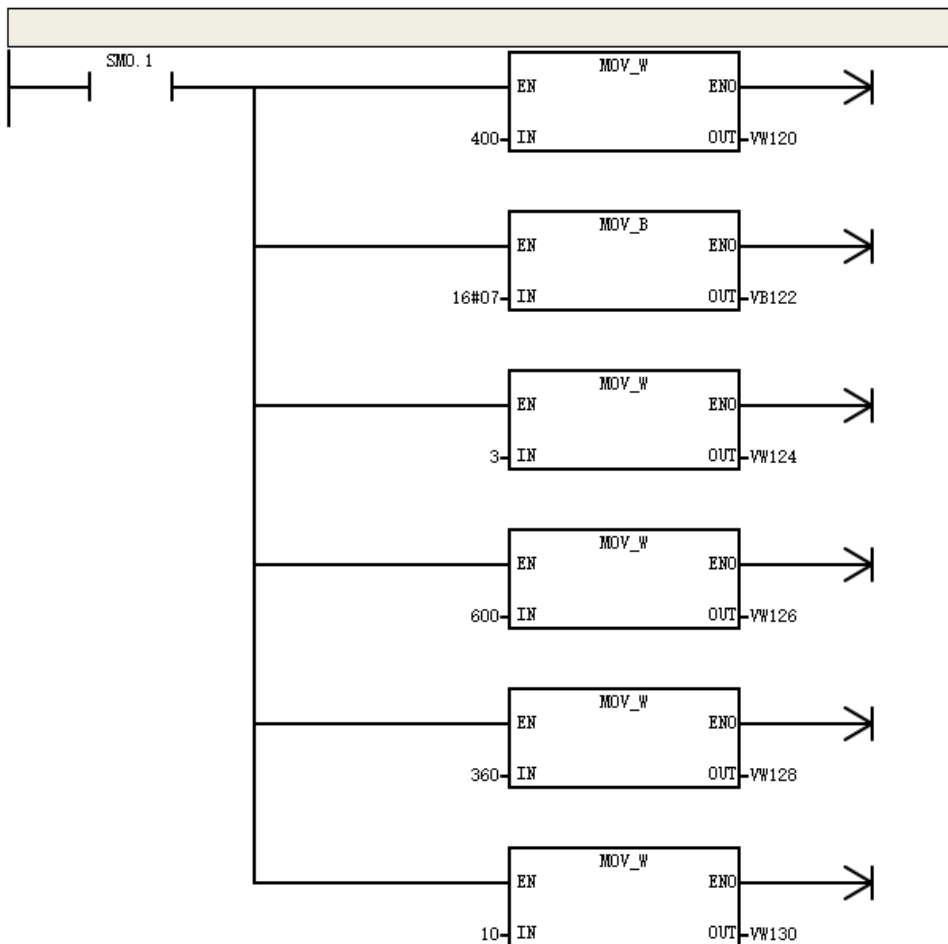
无需计算PID参数地址，只需输入回路所在的插槽号和通道号。

使能Run来运行该回路。

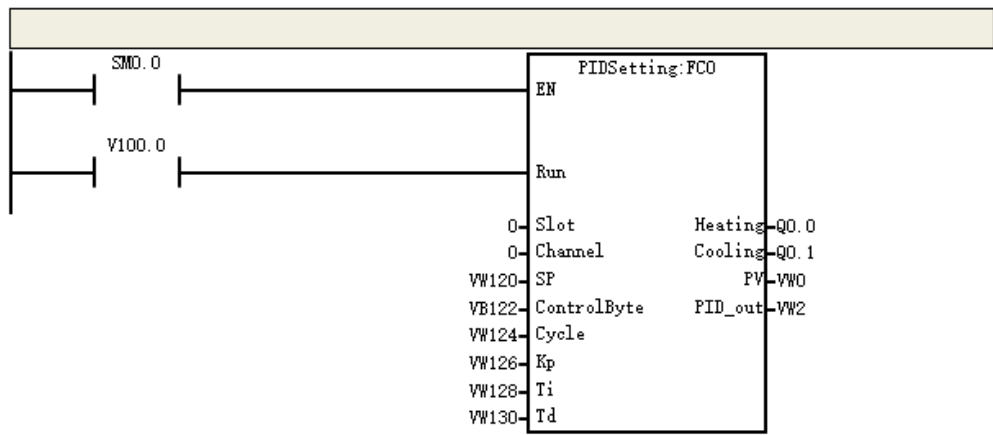
本例包含网络1和网络2。

Q0.0为正向脉冲输出，Q0.1为负向脉冲输出。VW0为实际温度，VW2为PID模拟量输出。

修改PID设定参数使用其他地址，本例设定温度对应地址使用VW120，控制字：VB122，脉冲输出周期：VW124，比例系数：VW126，积分时间：VW128，微分时间：VW130



网络 2



DIP 开关配置

SM231 热电偶温控模块支持 J/K 型热电偶，模块提供了一个 6 位的 DIP 开关用于选择测量单位、冷端补偿和开路故障方向，对于是否进行断线检测，与 SM231 8TC 一样，SM231 热电偶温控模块强制进行断线检测。

6 位 DIP 开关位于模块的下方，如下图所示。其中 SW1~SW2 没有使用，SW3 用于选择热电偶类型，SW4 用于选择断线检测方向，SW5 用于选择测量单位，SW6 用于选择是否进行冷端补偿。为了使 DIP 开关设置起作用，用户需要给 PLC 电源断电再通电。

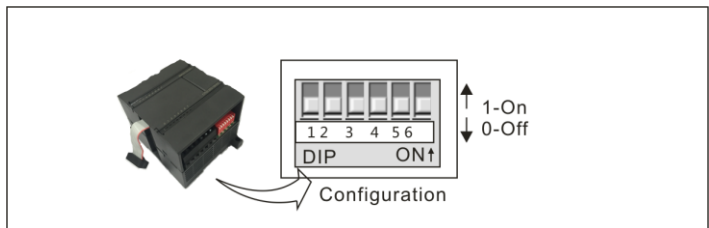


表 4-6-8 SM231 开关配置

设置项目	开关位置	设置
热电偶类型选择	SW3	0: J型; 1: K型
断线检测方向	SW4	0: 正标定 (+3276.7度); 1: 负标定 (-3276.8度)
测量单位选择	SW5	0: 摄氏度; 1: 华氏度
是否进行冷端补偿	SW6	0: 是; 1: 否

各种类型热电偶的温度范围 (°C) 和准确度

数据字 (1个数字位=0.1°C)		类型J	类型K	类型T	类型E	类型R,S	类型N	±80mV	
十进制	十六进制								
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
:	:							80.0029mV	
27649	6C01							80mV	NR
27648	6C00								
:	:								
17680	4510					1768.0°C			
:	:								
13720	3598		1372.0°C 超出范围						
:	:								
13000	32C8		1300.0°C				1300.0°C		
:	:								
12000	2EE0	1200.0°C							
:	:								
10000	2710								
:	:								
4000	0FA0			-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	
:	:								
-500	FE0C					400.0°C			
:	:								
-1500	FA24	-150.0°C							
:	:								
-2000	F830	低于范围	-200.0°C						
:	:								
-2100	F7CC	-210.0°C							
:	:								
-2550	F60A		低于范围						
:	:								
-2700	F574			-255.0°C	-255.0°C				
:	:			低于范围	低于范围				
-27648	9400			-270.0°C	-270.0°C				
:	:								
-27649	93FF							-80mV	
:	:								
-32512	8100							-80.0029mV	
:	:								
#	#							-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
全量程范围的精度		S0.1%	S0.3%	S0.6%	S0.1%	S0.6%	S0.1%	S0.1%	
精度(无冷端补偿的额定范围)		S1.5°C	S1.7°C	S1.4°C	S1.3°C	S3.7°C	S1.6°C	S0.10°C	
冷端误差		S1.5°C	S1.5°C	S1.5°C	S1.5°C	S1.5°C	S1.5°C	NA	

*OF=上溢 OR=超出范围 NR=额定范围 VR=低于范围 UF=下溢
 †表示所有大于该值但小于断线阈值的模拟量都报告为上溢值, 32767 (10x7FFF)
 ‡表示所有小于该值但大于断线阈值的模拟量都报告为下溢值, -32768 (0x8000)

图 4-17 热电偶测量范围

表 4-6-9 231-7HF32 电流型 PID 模块规范

物理特性	
尺寸 (W x H x D), mm	120.5 x 80 x 62
重量	210 g
插入式 I/O 端子	不是
电源消耗	
总线 (5 V DC)	87 Ma
L+	37 Ma
功率损耗	1.8 W
输入特性	
输入数目	8, 模拟量
输入范围 / 输入阻抗	电流型输入
电压输入端允许的最大输入电	30 V DC

压	
电隔离	是
现场侧-逻辑	500 V AC
现场侧-直流 24 V	500 V AC
直流 24 V-逻辑	500 V AC
更新时间	825 ms (所有通道)
测量原理	SIGMA-DELTA
分辨率	15 bit + 符号
温度	0.1 °C / 0.1 °F
噪声频率上的噪声抑制	85 Db
对噪声频率	50 / 60 / 400 Hz
共模电压	120 V AC
共模抑制, 最小	120 Db @ 120 V AC
基本误差	0.1% FS (电流)
再现性	0.05% FS
诊断程序	LED : EXTf, SF
电缆长度, 最大	100 m —传感器
电缆环路电阻, 最大	10 Ω
PID 特性	
PID算法	PID+FUZZY参数自调整
采样时间	1秒
输出最小脉宽	10ms
PID类型	P、PI、PD、PID型
PID输出类型	模拟量或PWM脉宽控制
PID输出极性	双极或单极

SM231 电流型 PID 模块 (CTH2 231-7HF32)

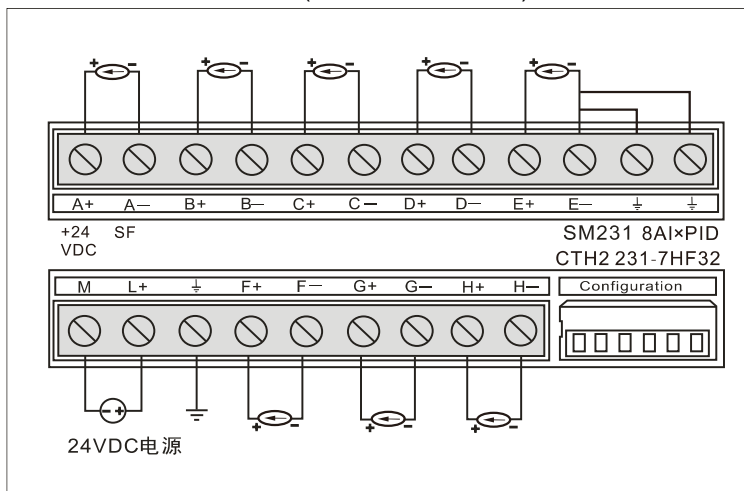


表 4-6-10 DIP 开关配置

Sw1	Sw2	Sw3	Sw4	Sw5	Sw6	量程	分辨率
ON	ON	OFF	OFF	OFF	OFF	0~20Ma	0.1°C/F
ON	ON	ON	OFF	OFF	OFF	4~20Ma (default)	0.1°C/F

PID 地址与参数配置

表 4-6-11 PID 地址计算公式

地址名称	计算公式	备注
PID 参数地址	$A = (2048 + S * 256) + 16 * C$	S 为模块所在的槽号 (范围 0~6) C 为通道号 231-7HF 为 0~7
PID 正向脉冲输出地址	$X = (2048 + S * 256) + 12$	
PID 负向脉冲输出地址	$Y = (2048 + S * 256) + 13$	

表 4-6-12 PID 参数输出部分 (模块到 PLC)

内容	地址	数值范围	实际值
实际温度	VW A	-2000~13000	-200~1300 度 (根据用户设定的 Range 参数而定, 默认是 1300)
状态字	VW A+2	-	-
PID 模拟量输出	VW A+4	双极: -32000~+32000 单极: 0~+32000	-

表 4-6-13 PID 参数输入部分 (PLC 到模块)

内容	地址	数值范围	实际值
设定温度	VW A+128	-2000~13000	-200~1300 度
控制字节	VB A+130	位为“0”时	位为“1”时
	V (A+130).0	PID 不运行, 没输出	PID 运行
	V (A+130).1	积分一直起作用, 比例系数 Kp 不自动调整	积分分离及比例系数自动调整
	V (A+130).2	PID 单极输出	PID 双极输出, 具有加热和冷却功能
	V (A+130).3	无作用	无作用
	V (A+130).4	积分起作用	积分不起作用
	V (A+130).5	微分起作用	微分不起作用
	V (A+130).6	实际温度值滤波, 抗干扰更强	实际温度值不滤波
PID 脉冲输出周期设定	VW A+132	1-255	1-255 秒
Kp(比例系数)	VW A+134	0~9999	0~999.9
Ti(积分时间)	VW A+136	0~3600	0~3600 秒
Td(微分时间)	VW A+138	0~3600	0~3600 秒
Range(量程范围)	VW A+140	0~13000	0~1300 度

表 4-6-14 PID 脉冲输出地址计算

脉冲输出通道	脉冲输出地址	
	正向脉冲输出	负向脉冲输出
0	V X.0	V Y.0
1	V X.1	V Y.1
2	V X.2	V Y.2
3	V X.3	V Y.3
4	V X.4	V Y.4
5	V X.5	V Y.5
6	V X.6	V Y.6
7	V X.7	V Y.7

正向脉冲输出地址的计算： $X=(2048+S*256)+12$;

负向脉冲输出地址的计算： $Y=(2048+S*256)+13$

应用例子：

计算第二个扩展模块上的 231-7HF 的最后一个 PID 回路的地址，Kp=1200，Ti=360，Td=50 脉冲输出周期设定=2s，第二个模块最后一个通道，所以：S=1，C=7	
A地址	A=2048+1*256+16*7=2416
X地址	X=2048+1*256+12=2316
Y地址	y=2048+1*256+13=2317
DB BLOCK的参数块如下:// S=1，C=7	
VW2544 500	//设定温度
VB2546 16#07	//控制字(PID 使能、参数自调整、双极输出
VW2548 2	//脉冲输出周期
VW2550 1200	//Kp 比例系数
VW2552 360	//Ti 积分时间 (秒)
VW2554 50	//Td 微分时间 (秒)
VW2556 13000	//量程范围 1300 度 (默认值)
VW2416 0	//实际温度
VW2418 0	//状态字
VW2420 0	//PID 模拟量输出

PID模块的参数地址占用CPU的V存储区。

为了保证PID模块能正常使用，编写其他程序块时请一定不要使用您所用的PID模块占用的V存储区。

模块在第0号插槽所占用的地址为：VW2048到VW2298

模块在第1号插槽所占用的地址为：VW2304到VW2554

模块在第2号插槽所占用的地址为：VW2560到VW2810

模块在第3号插槽所占用的地址为：VW2816到VW3066

模块在第4号插槽所占用的地址为：VW3072到VW3322

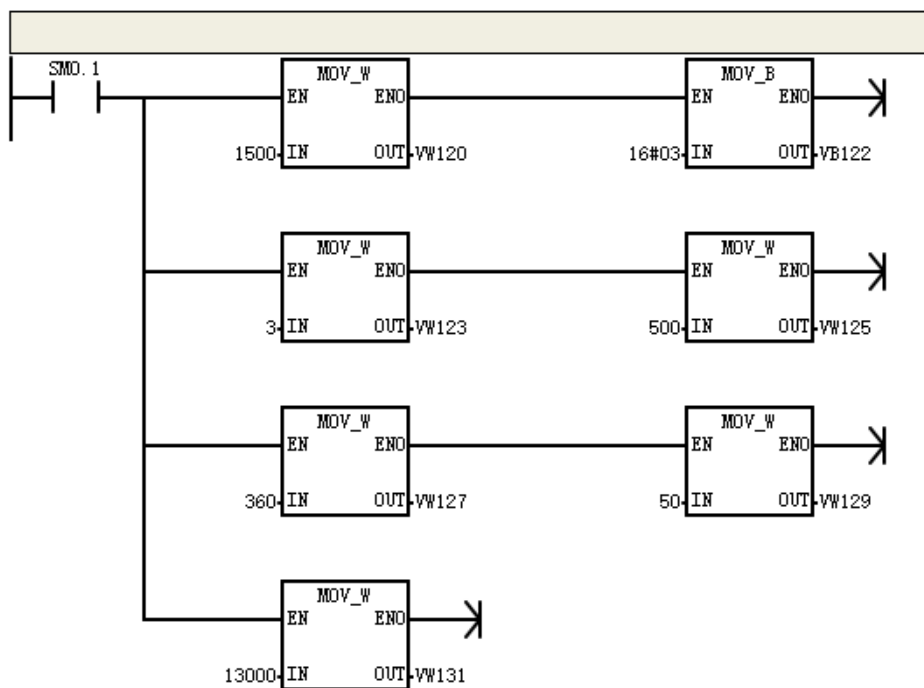
模块在第5号插槽所占用的地址为：VW3328到VW3578

模块在第6号插槽所占用的地址为：VW3584到VW3834

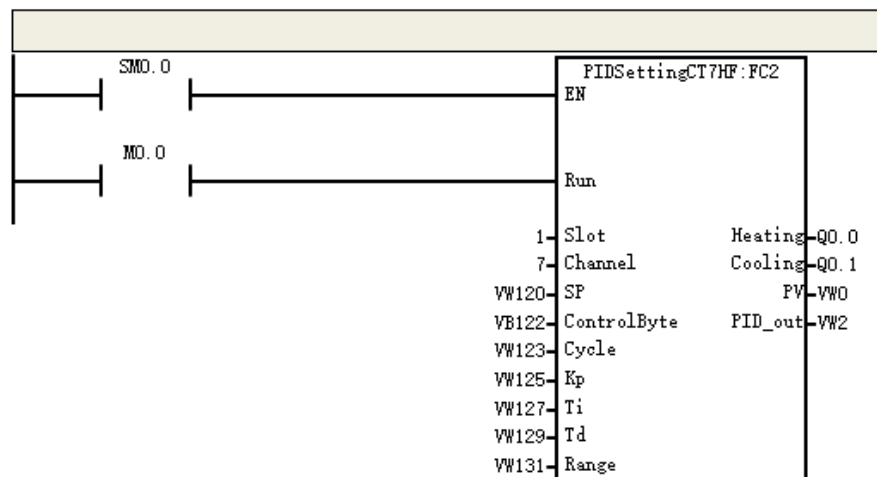
网络 4

设置第一个扩展模块（插槽号为0）的SM231-7HF PID模块第一个PID回路（通道号为0）的参数：
 假设该回路调试完成后需要再改变PID参数，调用PID settingCT7HF来设定该回路的参数（需先添加em231-7hf_pid库）
 无需计算PID参数地址，只需输入回路所在的插槽号和通道号，使能Run来运行该回路。
 本例包含网络1和网络2，网络1初始化PID参数,网络2运行PID
 Q0.0为正向脉冲输出，Q0.1为负向脉冲输出，VW0为实际温度，VW2为PID模拟量输出
 也可修改PID设定参数使用其他地址，本例对应地址为：
 设定温度--VW120；控制字--VB122；脉冲输出周期--VW123；比例系数--VW125
 积分时间--VW127；微分时间--VW129；温度传感器量程--VW131

网络 4



网络 5



4.7 通信模块规范

表 4-7-1 订货数据

规格参数	订货号
SM277A Profibus DP从站接口模块, 12M通信速率, 光电隔离	CTH2 277-0AA32
SM277B Profibus DP从站模块, 1.5M通信速率, 光电隔离	CTH2 277-0AB32
SM277C CAN 从站模块, 自带 8DI/6DO, 光电隔离, 可扩展 7 个模块	CTH2 277-0AC32
SM277 PN 从站模块, 100Mbps 通信速率, 允许扩展 8 个 I/O 模块	CTH2 277-0PN32

在通信模块中, SM277A 和 SM277B 均采用光电隔离技术, 可靠性高, 抗干扰能力强。这两种模块均自带终端电阻, 总线采用接线方式, 无需专用网络接头。其供电电源具有反接保护和浪涌吸收功能, 可适用于恶劣的工作环境。

不过, SM277A 需要与 CPU 配合使用, 而 SM277B 则可作为独立从站并带有 6 个扩展模块, 从站的 I/O 规模可达 128DI/128DO, 48AI/24AQ。

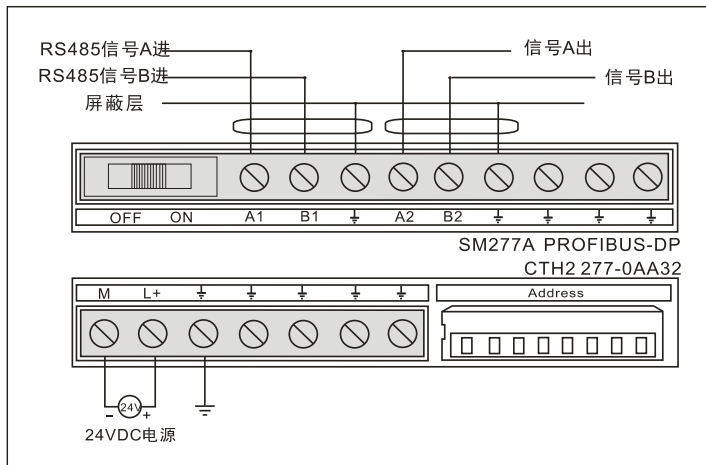
4.7.1 SM277A DP 从站接口模块规范

表 4-7-2 SM277A 模块参数

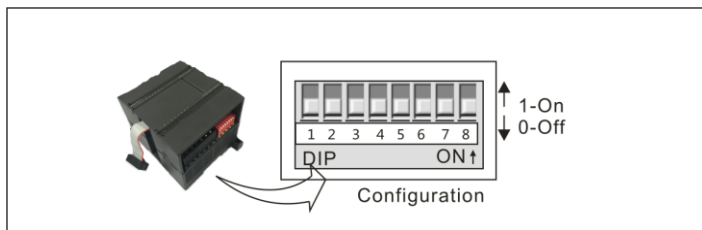
常规				
订货号	CTH2 277-0AA32			
尺寸 (宽×高×深)	71.3 × 96 × 62mm			
端口数	1			
电气接口	隔离RS485			
本机 I/O	0			
最大扩展模块数	6 个			
PROFIBUS-DP/MPI波特率 (自动设置)	9.6、19.2、45.45、93.75、187.5和500Kbps; 1、1.5、3.6和12Mbps			
协议	PROFIBUS-DP从属装置和MPI从属装置			
LED	OFF	红色	红灯闪烁	绿色
DX MODE	不在数据交换模式	--	--	在数据交换模式
DP ERROR	没有错误	脱离数据交换模式	参数化/组态错误	--
POWER	没有24V用户电源	--	--	24VDC用户电源良好
CPU FAULT	模块良好	内部模块故障	--	--
电缆长度 (最大)				
低于93.75Kbps波特率	1200m			
187.5Kbps波特率	1000m			
500Kbps波特率	400m			
1 ~ 1.5Mbps波特率	200m			
3 ~ 12Mbps波特率	100m			
网络性能				
站地址设置	0至126 (由拨码开关进行设置)			
每段最大站数	32			
每个网络最大站数	126、至多125个SM277站点			
MPI连接	共6个, 2个保留 (1个用于PG, 1个用于OP)			
24VDC输入电源要求				
电压范围	20.4 ~ 28.8VDC (2类或PLC传感器电源)			
功耗	2.3W			
最大电流 (模块通信端口的激活)	70Ma			
纹波噪声 (<10Mhz=)	<1V峰间值 (最大值)			

接线图

SM277A DP 从站接口模块(CTH2 277-0AA32)



地址开关设置



DIP 开关 SW1-8, 用二进制数表示, 其中 SW1 为最低位, SW8 为最高位 (必须打到“OFF”), SW1-SW7 打到“ON”表示“1”, 打到“OFF”表示“0”具体计算如下:

$$\text{地址} = \text{SW1} \times 2^0 + \text{SW2} \times 2^1 + \text{SW3} \times 2^2 + \text{SW4} \times 2^3 + \text{SW5} \times 2^4 + \text{SW6} \times 2^5 + \text{SW7} \times 2^6$$

表 4-7-8 地址开关配置速查表

地址	SW1~SW8	地址	SW1~SW8	地址	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		

表 4-7-9 终端电阻设置

终端电阻	含义
ON	表示处于网络最后一个节点
OFF	表示处于网络非最后一个节点

4.7.2 SM277B DP 从站模块规范

表 4-7-10 SM277B 电气和环境规范

运行环境温度	0 ~ 55°C
储存环境温度	-25 ~ 70°C
空气流动性	在设备上下50.8mm位置以内有流动空气
湿度	5 ~ 95% (无结露)
抗电气干扰	脉宽50ns, 重复频率5kHz, 2000V电压峰值
抗振性	频率10~57Hz, 幅度0.1mm, 频率57~150Hz, 加速度1.0g, 3维方向各10次
抗冲击性	15g, 持续11ms, 3维方向冲击3次
耐高压绝缘性	端对地施加DC2200V 的电压, I/O 接口端对其他端施加AC1500V, 持续1分钟
接地方式	3类接地 (多分支单点接地)
运行环境	隔尘非腐蚀性环境
倒下	100mm, 4次落体, 未包装
自由落下	1米, 5次, 运输包装

3类接地方法: 将每个设备的接地端子单独接到接地装置上

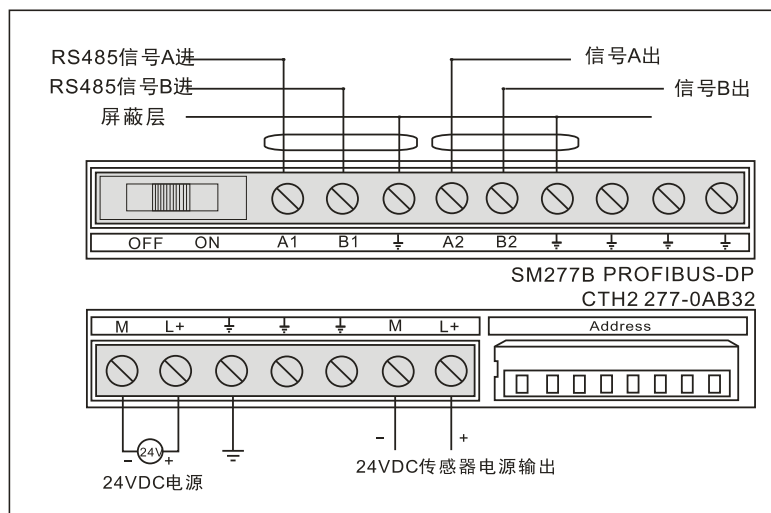
表 4-7-11 SM277B 模块规范

常规	
订货号	CTH2 277-0AB32
尺寸 (宽×高×深)	71.3 × 96 × 62mm
端口数	1
电气接口	RS485隔离 (500VAC)
PROFIBUS-DP 波特率 (自动设置)	9.6、19.2、45.45、93.75、187.5和500Kbps; 1、1.5Mbps

协议	PROFIBUS-DP V0从设备
本机IO	无
扩展IO	
允许的最大扩展模块数	6
允许的最大智能模块数	无
最大 I/O (映射)	256 (128 In/128 Out)
LED指示灯	
ON (绿色)	通电后亮起
SF (红色)	如果SM277B自检时发现错误(包括地址设定超范围), 或扩展IO模块有故障, 该灯亮起。
BF (红色)	没有与主站进入DP交换时, 该灯闪烁。
电缆长度(最大)	
不超过93.75Kbps波特率	1200m
187.5Kbps波特率	1000m
500Kbps波特率	400m
1至1.5Mbps	200m
网络性能	
站地址设置	0~126 (由拨码开关进行设置)
每段最大站数	32
每个网络最大站数	126, 至多125个SM277B站
24VDC输入电源要求	
电压范围	20.4~28.8VDC (2类或PLC的传感器电源)
隔离	无
功耗	8W
为传感器提供的输出电流 (24VDC)	
电压范围	20.4~28.8VDC (2类或PLC传感器电源)
24V输出最大电流	400Ma
为扩展模块提供的输出电流 (5VDC)	
5V输出最大电流	660Ma

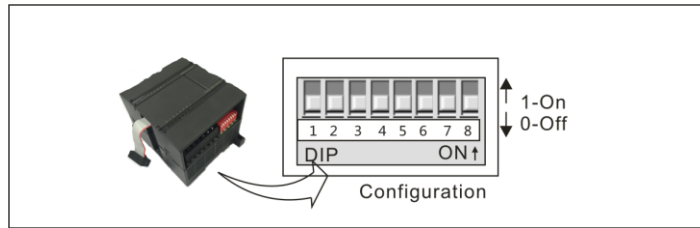
接线图

SM277B PROFIBUS DP 从站模块(CTH2 277-0AB32)



设置 PROFIBUS 地址

通过 DIP 开关设置 PROFIBUS 地址，更改 PROFIBUS 地址之后，需要将 SM277B 断电并重新上电。



DIP 开关 SW1-8，用二进制数表示，其中 SW1 为最低位，SW8 为最高位（必须打到“OFF”），SW1-SW7 打到“ON”表示“1”，打到“OFF”表示“0”，具体计算如下：

$$\text{地址} = \text{SW1} \times 2^0 + \text{SW2} \times 2^1 + \text{SW3} \times 2^2 + \text{SW4} \times 2^3 + \text{SW5} \times 2^4 + \text{SW6} \times 2^5 + \text{SW7} \times 2^6$$

表 4-7-12 地址开关配置速查表

地址	SW1~SW8	地址	SW1~SW8	地址	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		

表 4-7-13 终端电阻

终端电阻	含义
ON	表示处于网络最后一个节点
OFF	表示处于网络非最后一个节点

4.7.3 SM277C CANopen 从站模块规范

CANopen 从站接口模块采用光电隔离，可靠性强，抗干扰能力强。其自带两个 RJ45 CANopen 通信口，与 CAN 主站系统组成分布式控制系统，该模块可扩展 7 个 CTH200 扩展模块（不支持 PID 智能模块），自带 8DI/6DO 晶体管输入输出和 12 Pin 拨码开关，方便选择站地址、波特率及终端电阻。

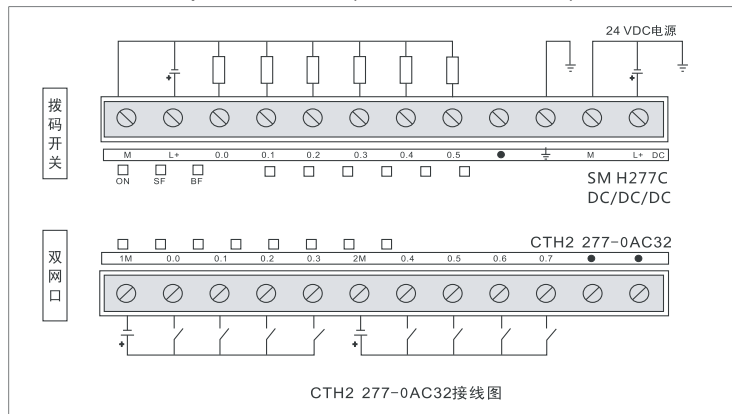
表 4-7-14 SM277C 模块规范

常规		
订货号	CTH2 277-0AC32	
尺寸（宽×高×深）	71.3 × 96 × 62 mm	
数字量输入		
端口数	2 个 RJ45 CANopen 通信接口	
额定输入电压	24VDC	
类型	源型/漏型（IEC 1 类源点）	
输入电压	典型值 24VDC	
允许的最大连续电压	30VDC	
浪涌电压	35VDC，持续 0.5s	
输入延迟	6.4ms（最短 6.3ms）	
隔离（现场到逻辑电路）	是	
同时接通输入点数	8（全部为 55°C）	
电缆最大长度	屏蔽	500m 标准输入
	非屏蔽	300m 标准输入
数字量输出		
类型	固态-MOSFET（源型）	
额定电压	24VDC	
输出电压范围	20.4~28.8VDC	
输出电流（信号 1，最大值）	0.5A	

输出组数	1	
每组输出点数	6	
同时接通输出点数	6	
单组最大电流	4.5A	
浪涌电流	8A, 持续 100ms	
隔离	光耦隔离, 500VAC, 持续时间 1 分钟以内	
输出延迟	关-开	最大 15 μ s
	开-关	最大 130 μ s
最大电缆长度	屏蔽	500m 标准输出
	非屏蔽	150m 标准输出

接线图

SM277C CANopen 从站模块(CTH2 277-0AC32)



地址开关设置

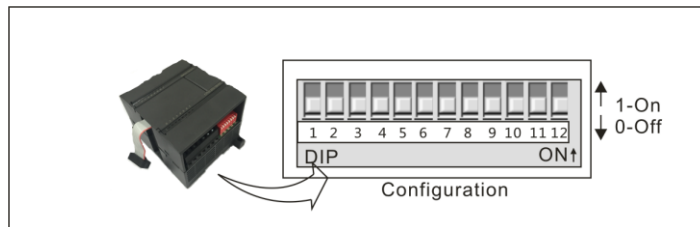
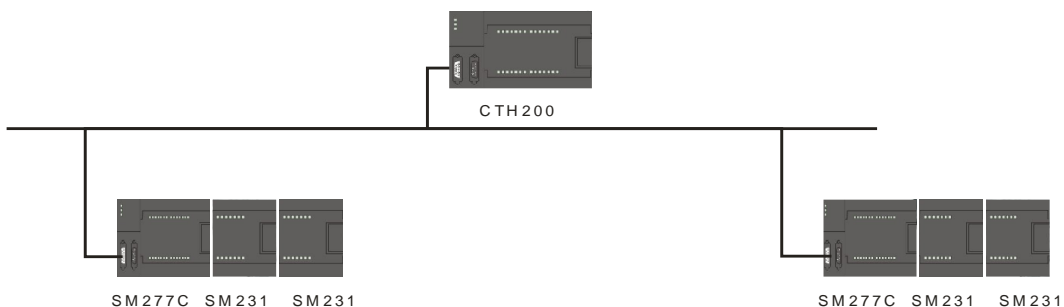


表 4-7-15 DIP 开关 SW1-12 配置

SM277-0AC32 拨码开关									
拨码号		使用方法							
SW1	接点地址	二进制数表示: 1 为低位, 7 为高位。 注: 0 是全局地址, 使用时禁止设为 0 地址。							
SW2									
SW3									
SW4									
SW5									
SW6									
SW7									
SW8	波特率	二进制数表示: 8 为低位, 10 为高位							
SW9		拨码	111	110	101	100	011	010	001
SW10		波特率 Kbps	1000	800	500	250	125	50	20
		最大长度(m)	25	50	100	250	500	1000	2500
SW11	终端电阻	设备处在网络端口处需拨到“ON”位置, 其他拨到“OFF”位置							
SW12	保留	—							

网络架构



硬件诊断

可以通过 SM 277C 的通信指示灯状态来诊断 CANopen 主从站网络。

SM277C 的 LED 灯状态

接通 SM 277C 的电源之后，标示为“ON”（绿）的 POWER（电源）LED 指示灯将亮起。如果 BF 和 SF LED 保持熄灭状态，则说明 SM 277C 正常运作。如果 BF 或 SF LED 亮起或闪烁，则说明硬件组态或者接线有错误。如果主系统的运行无故障，则错误的原因可参考下表：

表 4-7-16 SM 277C 状态 LED 灯功能说明

LED 灯	亮	灭	闪烁	备注
ON	有电源	无电源	---	电源指示
SF	扩展 IO 模块有故障	扩展 IO 模块无故障	---	系统故障
BF	没检测到 CAN 网络	检测到 CAN 网络	组态配置不一致	总线故障

4.7.4 SM277 PN 从站模块规范



提示

SM277 PN 可搭配第三方 CPU 主站使用。具体使用方法及示例请参见《CTH200 系列 profinet 从站模块使用手册》。手册下载地址：<http://www.co-trust.com>

表 4-7-17 SM277-PN 模块遵循的环境标准及规范

环境条件-运输和存贮	
温度	-40℃~+70℃
相对湿度	10%~95%，非结露
环境条件-工作	
温度	0℃~60℃ 0℃~40℃
水平安装位置	
垂直安装位置	
相对湿度	10%~95%，非结露
电磁兼容性-抗扰度	
静电放电 IEC61000-4-2	接触放电：±4KV 空气放电：±8KV
电快速瞬变脉冲群 IEC61000-4-4	电源线：2KV，5KHz 信号线：2KV，5KHz（I/O 耦合夹） 1KV，5KHz（通讯耦合夹）
浪涌 IEC61000-4-5	电源线：2KV（非对称），1KV（对称）
射频电磁场辐射 IEC61000-4-3	80MHz~1GHz，10V/m，80%AM（1KHz）

射频场感应传导干扰 IEC61000-4-6	0.15MHz~80MHz, 10V/m, 80%AM (1KHz)
直流电源输入端口短时中断和 电压变化 IEC61000-4-29	短时中断: 10ms
高温运行 IEC60068-2 低温运行 IEC60068-2	65°C, 驻留时间: 24h, 温变率: ≤1°C/min, 常温恢复时间: ≥1h -20°C, 驻留时间: 24h, 温变率: ≤1°C/min, 常温恢复时间: ≥1h
高温启机 IEC60068-2 低温启机 IEC60068-2	65°C, 2 小时, 温度稳定后, 不连续开机 3 次 -20°C, 2 小时, 温度稳定后, 不连续开机 3 次
高低温循环运行 IEC60068-2	-10°C~60°C, 驻留时间 3h, 温变速率 1°C/min, 5 个循环, 常温 恢复时间: 1h
高温存储 IEC60068-2 低温存储 IEC60068-2	75°C, 存储时间: 72h, 温变率: ≤1°C/min, 常温恢复时间: ≥1h -40°C, 存储时间: 72h, 温变率: ≤1°C/min, 常温恢复时间: ≥1h
冷热冲击 IEC60068-2	-40°C~75°C, 驻留时间 3h, 温变时间: 2min, 3 个循环
恒定高湿 IEC60068-2	存储温度: 40°C, 存储时间: 48h, 存储湿度: 93%
交变湿热 IEC60068-2	25°C~55°C, 湿度: 95%, 单个周期循环: 24h, 2 个循环
正弦振动 (裸机) IEC60068-2	5~150Hz, 0.05G ² /Hz 150Hz~500Hz, -3Db/oct, 1h/轴, X、Y、Z 总共 3 轴
机械冲击 (裸机) IEC60068-2	15G, 11ms 脉冲, 3 次/方向
跌落	1m, 10 次, 运输包装

表 4-7-18 SM277 PN 模块的常规特性

物理特性	
尺寸 (W×H×D)	71×80×62 mm
电源损耗	
额定输入电压	24V DC
输入电压范围	20.4V~28.8V DC
输入电流	0.8A
总线电源电压	+5V DC
总线电源电流	1.1A
隔离	外部电源与系统电源隔离
电源保护	供电电源端提供反接保护功能及浪涌吸收功能
LED 指示灯特性	
24V 电源指示灯 (绿色)	ON = 24VDC 供电正常
	OFF = 无 24VDC 供电
SF 指示灯 (红色)	ON = 扩展 I/O 总线故障或者 PROFINET 模块故障
	OFF = 无错
BF 指示灯 (红色)	ON = PROFINET 总线通信故障 (未连接交换机, 未检测到网络)
	Blinking = 组态配置不一致
	OFF = 无错
MT 指示灯 (黄色) (维护指示灯)	ON = 丢失同步
	OFF = 无错
P1R/P2R 口指示灯 (绿色)	ON = 有到交换机/PN 主站的连接
	OFF = 没有到交换机/PN 主站的连接
P1R/P2R 口指示灯 (黄色)	ON = 有到交换机/PN 主站的数据收发
	OFF = 没有到交换机/PN 主站的数据收发

表 4-7-19 SMS277 PN 模块的功能特性

扩展 I/O 能力	
每从站支持最多模块数	8（数字量模块、模拟量模块、温度模块和 PID 控制模块，不包括其他类型的模块）
协议类型	CTH200 PLC 自定义 4MHZ 总线协议
每从站 IO 最大配置	模拟量 IO 最大配置可达 64AI/32AQ 数字量 IO 最大配置可达 256DI/256DQ
拓扑结构	支持星形、树形、线型、环形
PROFINET 通讯口	
通讯接口	1 个双 RJ45 口
数据传输速率	以太网传输率为 10Mbps
	PROFINET 传输率为 100Mbps，全双工
支持的以太网服务	ping、arp、网络诊断(SNMP)/MIB-2、LLDP
发送周期	250us~4ms
第三方 PROFINET 主站	支持 S7-300/400、SMART CPU、S7-1200/1500
从站间通信距离最长	100m（100BASE-TX）
拓扑结构	支持星形、树形、线型、环形拓扑结构
隔离	通信口隔离
硬件组态功能	
需导入的文件	PROFINET GSD 文件 XML 格式
PN 从站	添加 CTH2 277-0PN32 模块后支持 8 个槽位的扩展
	扩展模块可添加数字量、模拟量、温度模块和 PID 控制模块，其他的模块不能添加

4.8 称重模块规范

表 4-8-1 SM231 称重模块性能参数

订货号	CTH2 231-7WA32
LED 指示灯	电源指示灯，ON=24VDC 供电正常，OFF=无 24VDC 供电 SF 指示灯，闪烁=超量程或断线，OFF=无错
电源损耗	
+5VDC 消耗电流	<140Ma
L+消耗电流	<100Ma
L+线圈电压范围	20.4~28.8VDC
功率损耗	5W
输入特性	
输入类型	应变仪，采用 4 线或 6 线技术接线方式
输入范围	0~1Mv/V
	0~2Mv/V
	0~4Mv/V
输入点数	一路称重传感器
称重传感器电阻	无防爆接口时：40 Ω < R < 4010 Ω
	有防爆接口时：87 Ω < R < 4010 Ω

隔离特性	
现场至逻辑	500VAC
现场至 24VDC	500VAC
24V 到逻辑	500VAC
共模抑制	>120Db@120VAC
输入分辨率	
零点温度系数	≤±0.1mV/K
测量原理	Σ-Δ
到传感器的导线长度	最大 500 米
噪声抑制	85db@50Hz/60Hz
数据字格式	电压：0~65535
分辨率	16 位
基本误差	0.05%Fs
线性误差	0.01%Fs

端子连接

称重模块是一款高精度的测量模块，它必须可靠的测量低信号电平（大约 1.5mV）。因此，要获得无故障的操作，合适的组装和电缆连接是至关重要的。

在连接称重传感器时，应遵守如下规则：

1) 如果准备连接一个以上传感器（称重传感器必须并联），则必须使用一个接线盒。若称重传感器到模块或到接线盒的距离大于称重传感器可用电缆的长度，应该使用专门的延长盒。

2) 电缆的屏蔽始终应通到接线盒或延长盒的电缆压盖处。

3) 规定的线路上应使用扭绞线对电缆，而且应该屏蔽：

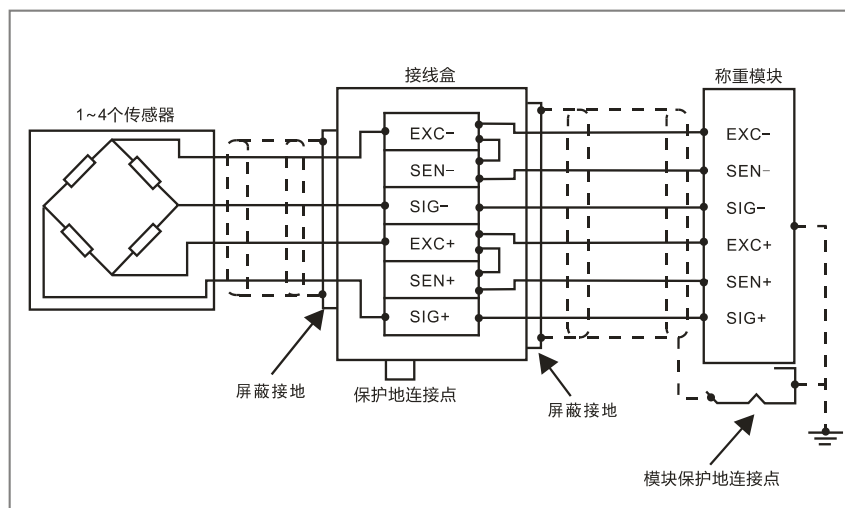
传感器线 SEN+/SEN-；

测量电压线 SIG+/SIG-；

电源电压线 EXC+/EXC-。

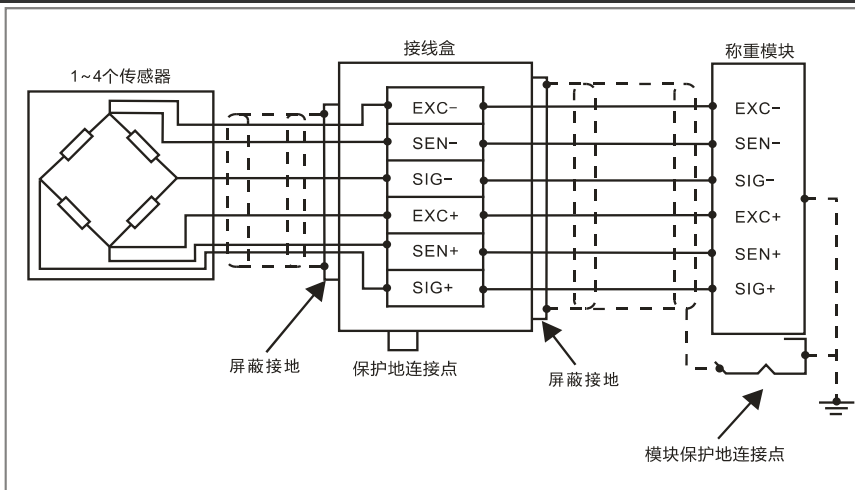
4) 屏蔽线必须接到屏蔽接头卡子上。

1、称重传感器四线制接法



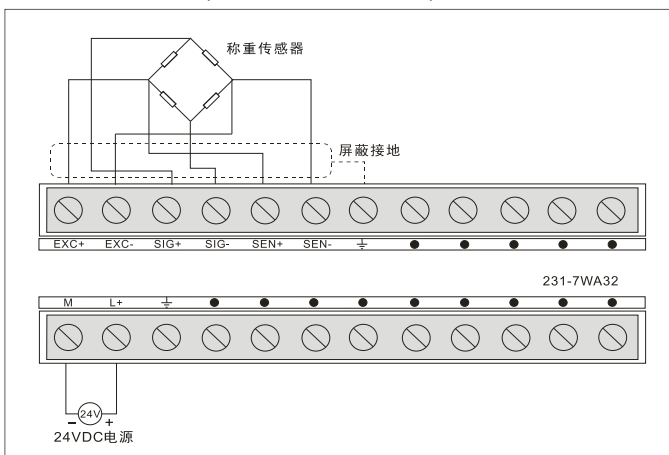
连接四线制传感器时，一定要在接线盒中将 EXC+与 SENS+短接，EXC-与 SENS-短接，否则称重模块可能无法正常工作。

2、称重传感器六线制接法



接线图

SM231 称重模块(CTH2 231-7WA32)



4.9 运动控制模块规范

表 4-9-1 运动控制模块数字量输入特性

常规	24VDC 输入
本机集成数字量输入点数	8 点输入
类型	漏型/源型 (IEC 类型 1/漏型)
额定电压	5Ma 典型值时为 24 VDC
允许的最大连续电压	30 VDC
浪涌电压	35 VDC 持续 0.5 s
逻辑 1 (最小)	在 2.72Ma 时为 15.6 VDC (I0.0、I0.1、I0.2、I0.4、I0.5、I0.6) 在 2.55Ma 时为 12.8VDC (I0.3、I0.7)
逻辑 0 (最大)	在 2.69Ma 时为 15.4VDC (I0.0、I0.1、I0.2、I0.4、I0.5、I0.6) 在 2.51Ma 时为 12.6 VDC (I0.3、I0.7)
输入延迟	< 1.1us (I0.0、I0.1、I0.2、I0.4、I0.5、I0.6) < 1ms (I0.3、I0.7)
邻近传感器的两条导线的连接 (Bero) 允许的泄漏电流 (最大值)	1Ma
隔离 (现场到逻辑电路)	是

光学（电流的）隔离群组	500 VAC, 1 分钟, 参见接线图	
高速输入速率 高速计数器逻辑 1=16~26 VDC	200KHz（单双相）（I0.0、I0.4） 200KHz（A/B 相）（I0.0 和 I0.1、I0.4 和 I0.5） （输入波形占空比 40%~60%）	
同时接通输入	所有	
最大电缆长度	屏蔽	500 米标准输入, 50 米 HSC 输入（对 HSC 输入请使用屏蔽双绞线且屏蔽层接大地）
	非屏蔽	300 米标准输入
高速输入, 计数器支持的最大变向频率	50KHz	

表 4-9-2 运动控制模块数字量输出特性

常规	晶体管输出	
本机集成数字量输出点数	8 输出	
类型	固态 - MOSFET（漏型）	
额定电压	24 VDC	
输出电压范围	5~28.8 VDC	
浪涌电流（最大）	8A 持续 100ms	
逻辑 1（最大）	0.5V	
逻辑 0（最小）	VCC-0.5V	
每点额定电流（最大）	0.5A	
每个公共端的额定电流（最大）	2.0A	
漏电流(最大)	10Ma	
照明负载（最大）	3.5W	
感应钳位电压	L+ 减 48 VDC, 1W 功耗	
接通电阻（接点）	0.3 Ω 典型值（0.6Ω 最大值）	
隔离 光学（电化, 现场到逻辑电路） 逻辑电路到接点 电阻（逻辑电路到接点） 隔离群组	500 VAC, 1 分钟 参见接线图	
延迟（最大） 断开到接通	0.2μs（Q0.0、Q0.1、Q0.2、Q0.3），50μs（Q0.4、Q0.5、Q0.6、Q0.7）	
接通到断开	0.2μs（Q0.0、Q0.1、Q0.2、Q0.3），130μs（Q0.4、Q0.5、Q0.6、Q0.7）	
脉冲频率（最大）	200KHz（Q0.0、Q0.2）	
机械寿命周期	/	
触点寿命	/	
同时打开输出	55°C 时, 输出所有	
两个输出并联	是, 仅输出同组时	
最大电缆长度	屏蔽	500 米
	非屏蔽	150 米

**注意**

由于通信传输，DO 输出指令的执行有一定的延时：若 CPU 后面挂单个 SM253 运动控制模块，则由于传输延时，输出指令从使能到执行延时约为 780us；若 CPU 后面挂最多 5 个 SM253 运动控制模块，则输出指令从使能到执行延时约为 930us。

SM253 模块面向运动控制应用，自身带有 2 个独立的高速计数器 MC253_HSC0 和 MC253_HSC1（单双相均为 200KHZ）；2 路独立的 200KHz 高速脉冲输出，支持运动控制库 motion_ctrl_module_lib 的 MC253_PTP/SPEED_CTL/PWM 等指令。有关运动控制库的具体使用信息，请参见附录 P SM253 运动控制模块库的使用。

接线图

SM253 运动控制模块(CTH2 253-1BH32)

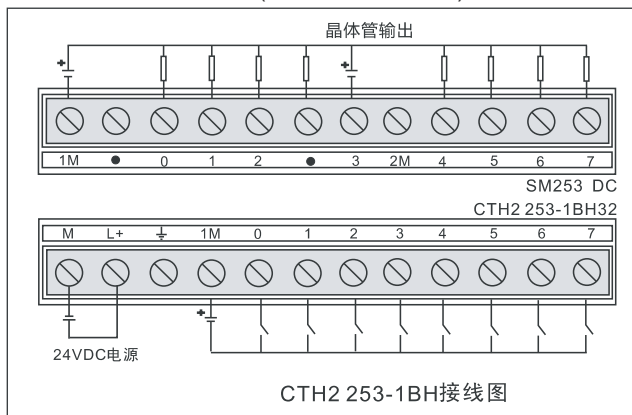


表 4-9-3 I/O 功能定义

IO.0	IO.1	IO.2	IO.3	IO.4	IO.5	IO.6	IO.7
MC253_HSC0 脉冲输入	MC253_HSC0 外部方向信号	MC253_HSC0 复位信号	运动轴 0 急停信号	MC253_HSC1 脉冲输入	MC253_HSC1 外部方向信号	MC253_HSC1 复位信号	运动轴 1 急停信号
Q0.0	Q0.1	Q0.2	Q0.3	Q0.4	Q0.5	Q0.6	Q0.7
轴 0 脉冲输出	轴 0 方向信号	轴 1 脉冲输出	轴 1 方向信号	普通输出口			

4.10 BD 扩展板规范及安装方法

表 4-10-1 BD 扩展板订货数据

规格参数	订货号
EBH AMS-03 模拟量输入输出扩展板，2 点*12 位精度电压输入，1 点*12 位精度电压/电流输出	CTH2 AMS-03S1-EB
EBH-AMS-06 模拟量输入输出扩展板，4 点*12 位精度电压输入，2 点*12 位精度电压输出	CTH2 AMS-06S1-EB
EBH-AMS-06 模拟量输入输出扩展板，4 点*12 位精度电压输入，2 点*12 位精度电流输出	CTH2 AMS-06S2-EB
EBH CAN 主站通信扩展板，1Mbps，光电隔离	CTH2 CAN-01S1-EB
EBH PWM 04 充电导引扩展板，双通道电压输入，单通道可调脉宽 PWM 输出	CTH2 PWM-04S1-EB

4.10.1 模拟量扩展板规范

CTH200 系列可搭配模拟量扩展板 EBH-AMS-03/EBH-AMS-06 使用，其具体规范如下表所示：

表 4-10-2 环境条件规范

项目	参数
运行环境温度	0 ~ 55°C
储存环境温度	-25 ~ 70°C
空气流动性	在设备上下50.8mm位置以内有流动空气
湿度	5 ~ 95%（无结露）
抗电气干扰	具体参考GB17626.2、GB17626.3、GB17626.4、GB17626.5、GB17626.6
抗振性	频率10~57Hz，幅度0.1mm，频率57~150Hz，加速度1.0g，3维方向各10次
抗冲击性	15g，持续11ms，3维方向冲击3次
耐高压绝缘性	端对地施加DC2000V的电压，I/O接口端对其他端施加AC1500V，持续1分钟
接地方式	3类接地
运行环境	隔尘非腐蚀性环境
倒下	100毫米，4次跌落，未包装
自由落下	1米，5次，运输包装

表 4-10-3 模拟量扩展板特性

项目	规范			
	EBH-AMS-03	EBH-AMS-06S1	EBH-AMS-06S2	
基本特性	物理尺寸（W×H×D）	24×14.4×68.2mm		
	总线电源电压	+5VDC		
	总线电源电流	60Ma	60Ma	60~250Ma
模拟量输入特性	电源指示灯	电源指示良好 ON = 5VDC 供电正常，OFF = 无5VDC 供电；		
	SF 指示灯	校准状态指示灯，闪烁=校准失败	校准状态指示灯	
	输入点数	2	4	
	输入类型	单端电压输入		
	输入范围	±10V		
	满量程范围数据字格式	-32000到+32000		
	DC 输入阻抗	>100KΩ		
	最大输入电压	30VDC		
	分辨率	11bit 加1个符号位		
	隔离	无		
	精度			
	最差情况(0-55 °)	±2.5%满量程		
	典型情况(25 °)	±1.0%满量程		
	重复性	±0.05%满量程		
	模拟到数字转换时间	125ms		
转换原理	Σ-Δ			
阶跃响应	最大250ms			
噪音抑制	典型值：-20Db@50Hz			

表 4-10-4 模拟量扩展板特性

项目		规范		
		EBH-AMS-03	EBH-AMS-06S1	EBH-AMS-06S2
模拟量输出特性	输出点数	1	2	
	信号范围	电压: 0-10V	电压: 0-10V	--
		电流: 0-20Ma	--	电流: 0-20Ma
	满量程范围数据字格式	0到+32000		
	分辨率	12位		
	LSB 值	电压: 2.50Mv	电压: 2.50Mv	--
		电流: 5.00Ma	--	电流: 5.00Ma
	隔离	无		
	精度			
	最差情况 (0-55 °)	电压: $\pm 2\%$ 满量程	电压: $\pm 2\%$ 满量程	--
电流: $\pm 3\%$ 满量程		--	电流: $\pm 3\%$ 满量程	
典型情况 (25 °)	电压: $\pm 1\%$ 满量程	--电压: $\pm 1\%$ 满量程	--	
	电流: $\pm 1\%$ 满量程	--	电流: $\pm 1\%$ 满量程	
稳定时间 • 带有阻性负载 • 带有容性负载 • 带有感性负载	0.1ms		0.2 ms 3.3 ms 0.5 ms (1Mh)	
最大输出驱动	$\geq 5000\Omega$	$\leq 500\Omega$		

端子定义

1、AMS-03 模拟量输入输出扩展板的结构图和端子示意图如下所示，后续表格中列出了各个端子位的定义，安装和连接扩展板时，请严格遵照端子位定义进行连接。

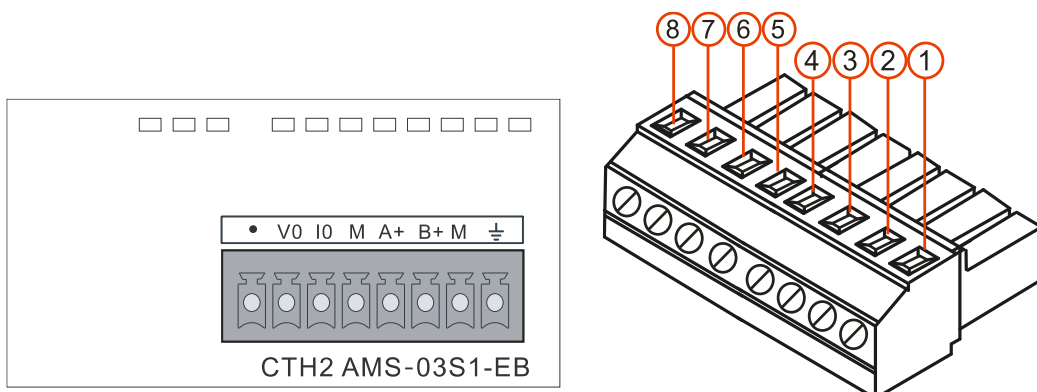


表 4-10-5 AMS-03 端子位定义

端子位	信号定义	端子位	信号定义
1	悬空	5	模拟输入 A+
2	电压输出 VO	6	模拟输入 B+
3	电流输出 IO	7	地 GND-M
4	地 GND-M	8	大地

2、AMS-06 模拟量输入输出扩展板的结构图和端子示意图如下所示，后续表格中列出了各个端子位的定义，安装和连接扩展板时，请严格遵照端子位定义进行连接。

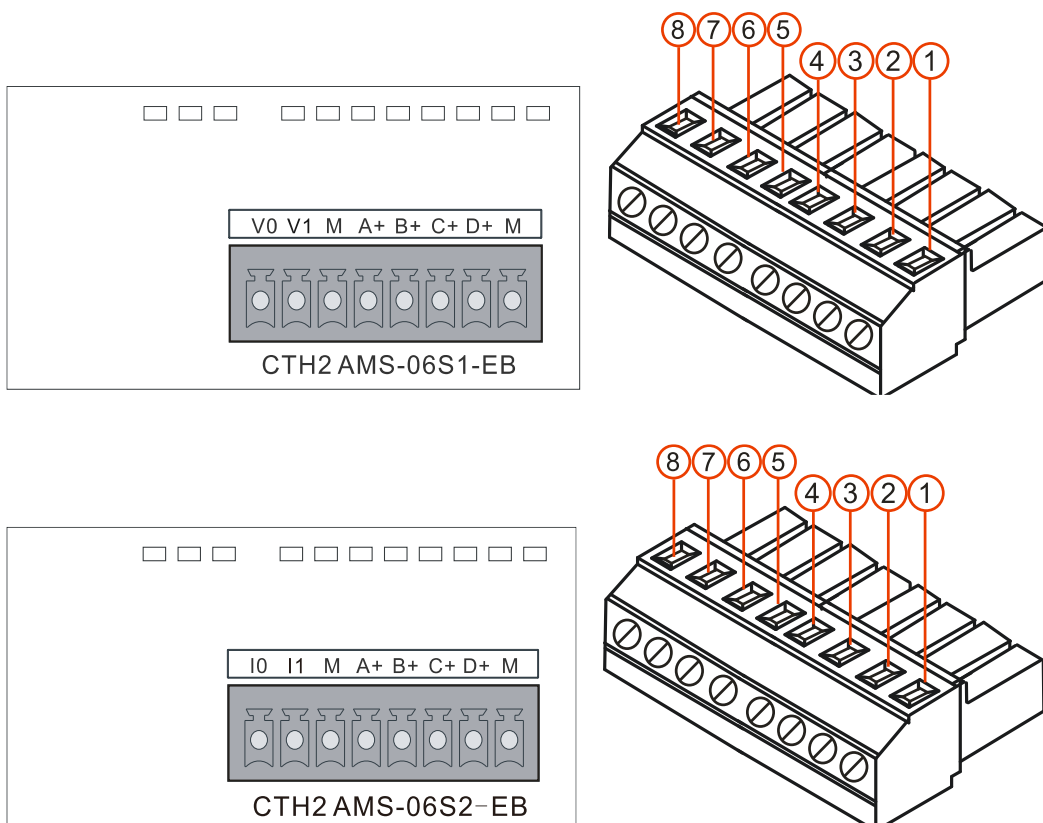


表 4-10-6 AMS-06 端子位定义

端子位	信号定义	端子位	信号定义
1	输出 V0/I0	5	模拟输入 B+
2	输出 V1/I1	6	模拟输入 C+
3	地 GND-M	7	模拟输入 D+
4	模拟输入 A+	8	地 GND-M

<注意> A+、B+、C+、D+为电压输入端，可输入正电压或负电压，外部接线需将端子 V1 连接到 3 号接地端；将模拟输入 D+连接到 8 号接地端。M 为公共端，禁止输出端直接与 M 相连造成输出短路。

4.10.2 CAN 扩展板规范

CTH200 系列高性能 CPU 支持 CAN 扩展板的使用，目前共两款 CAN 扩展板，CTH2-CAN-01S1-EB 和 CTH2-CAN-01S2-EB，前者支持 EasyCAN 组态，后者支持 CANfree 和 CAN 硬件组态以及第三方 EDS 文件的导入。

表 4-10-7 CAN 扩展板规范

物理特性	
尺寸（宽×高×深）	137 x 96 x 62mm
适用产品	CPU H224X/H226XL/H226XM
功耗	9W
CANopen 通信功能	
通信接口	1个（8-Pin 端子）

传输率 (kbps)	1000	800	500	250	125	50	20
最大长度 (m)	25	50	100	250	500	1000	2500
最大站地址	127						
站点地址范围	1-127						
最大主站带从站数	32						
最大数字量访问		字节数		内存起始地址			
	输入	64		IB16			
	输出	64		QB16			
最大模拟量访问		通道数		内存起始地址			
	输入	8		AIW64			
	输出	8		AQW64			



提示

CTH200 标准型 CPU (H224/H226M/H226L) 不支持 CAN 扩展板。

CAN 扩展板的接线图及端子示意图如下所示，后续表格中列出了各个端子位的定义，安装和连接扩展板时，请严格遵照端子位定义进行连接。

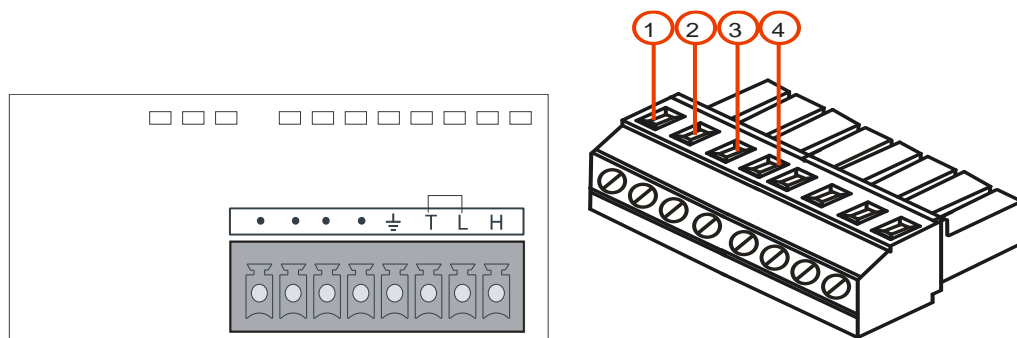


表 4-10-8 端子位定义

端子位	信号定义
1	CAN_H
2	CAN_L
3	匹配终端电阻时，请将 2 与 3 短接
4	地

4.10.3 导引电路扩展板规范

导引扩展板是 CTH200 系列 PLC 系统扩展模块中的一员，用于检测现场充电桩与充电枪甚至与充电车的连接状态以及充电状态，从而控制充电速率，并适用于交流电和直流电两种充电方式。

表 4-10-9 导引电路扩展板常规特性

订货号	CTH2 PWM-04S1-EB
物理特性	
尺寸 (W×H×D)	28×14.4×68.2 mm
电源损耗	
总线电源电压	+5V DC
总线电源电流	< 200Ma
LED 指示灯特性	
电源指示灯	电源指示: ON = 5VDC 供电正常, OFF = 无供电

SF 指示灯	校准状态指示：一直闪烁 = 校准失败，只亮一次 = 校准成功	
CC 指示灯	CC 断线检测：ON = 连接成功，OFF = 断线	
功能类别		
扩展功能	提供总线扩展功能	
输出保护	输出端提供 TVS 管保护功能	
滤波功能	采用硬件滤波与软件滤波相结合的方式	
电源功能	模块采用 5V DC 供电	
输入特性		
输入类型	单端输入（单/双极性）	
输入点数	2	
输入满量程电压	0 ~ +12V	
最大输入电压	+30V DC	
输入阻抗	≥100K	
数据格式	0 ~ +32000	
输入阶跃响应	最大 200ms	
模数转换时间	50ms	
隔离	无	
分辨率	12bit	
噪音抑制	典型值：-20Db@50HZ	
精度	最差(0°C-55°C)	±2%满量程
	典型（25°C）	±1%满量程
重复性	±0.05%满量程	
输出特性		
输出类型	单/双极性	
输出点数	2	
电压输出	+ 12V ± 5%	
PWM 输出	±12V ± 5%	
P W M 参 数	PWM 量程数据格式	0 ~ 1000
	满量程	0(-12V); 1000(+12V)
	输出占空比公差	±0.5%满量程
	PWM 频率范围	1000Hz ± 3%
	PWM 电压输出时间	< 3us
	PWM 上升时间 Tr	< 2us
	PWM 下降时间 Tf	< 2us
	PWM 最大输出驱动	≥1000Ω
+12V 允许输出电流	12Ma (RL=0Ω)	
	6Ma (RL=1KΩ)	
	4Ma (RL=2KΩ)	

端子定义

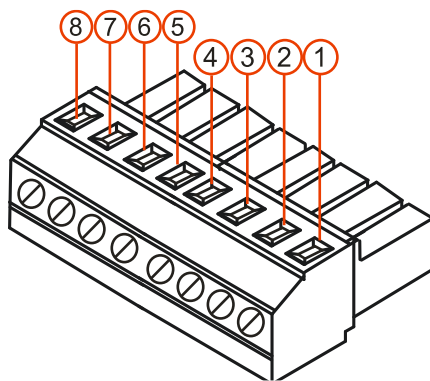


表 4-10-10 端子信号定义

端子位号	信号定义	端子位号	信号定义
1	NC	5	CP
2	NC	6	PWM
3	NC	7	+12V
4	CC	8	PE

接线图

CTH2 PWM-04S1-EB 扩展板适用于交流和直流充电导引

1) 交流充电导引功能，接线图如下图所示：

+12V 为输出端口(已接 1K 电阻)；

PWM 输出正负 12V 可调脉宽的 PWM；

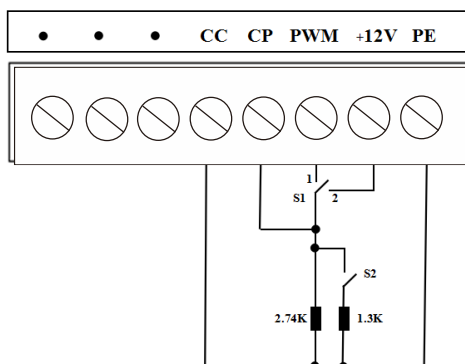
CC 检测电缆连接状态(0X12—断线，0X00—连接成功)；

CP 通过开关切换 S1 检测 PWM 输出和空闲时接+12V；

PE 为参考大地。

正常控制接线说明：

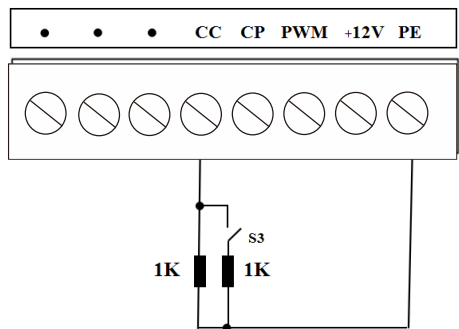
- ◆ 无负载时，开关 S1 闭合到位置 2，CP 检测到 12V 电压，CP 检测输出状态(状态值—0X01)；
- ◆ 当与车辆连接成功后，接入 2.74K 电阻负载，CP 检测到 9V，进入充电状态 2(状态值—0X02)；
- ◆ 然后 S1 切换到位置 1，输出正负 9V 的 PWM 可调脉冲；
- ◆ 最后闭合 S2，接入 1.3K 电阻与 2.74K 并联，此时外接电阻总值约 1K，CP 检测到 6V 的 PWM，进入充电状态 3(状态值—0X03)；



交流充电导引接线图

2) 直流充电导引功能，接线图如下图所示：

只有 CC 检测直流充电电缆连接状态，检测到 12V 为断线状态(0X12)，当接入 1K 负载时，CC 检测到 6V（状态 0X06）；当 S3 闭合后，CC 检测到 4V（状态为 0X04）。



直流充电导引接线图

访问方式

扩展板的访问方式为 SM 区地址访问，访问的地址范围为 SMB114—SMW124，具体某个地址对应的定义如下表 4-68 所示。

表 4-10-11 映射访问地址表

功能	映射访问地址	功能	映射访问地址
CC 输入	SMW116	PWM 输出	SMW124
CP 输入	SMW118	预留输出	SMW126
导引状态	SMW120	模块类型	SMB114
预留输入	SMW122	模块状态	SMB115

当 PLC 访问扩展模块时，该模块的类型与状态的具体定义分别显示在对应的地址中，由下表 4-69 给出不同的值对应功能的详细描述。

表 4-10-12 模块状态定义表

名称		功能	数值	备注说明
模块类型		有模块	0x1E	上电只访问一次
		无模块	0x00	
导引状态	直流充电	CC 断线空闲	0x12	对应 CC 检测电压值 12V
		CC 连接状态 1	0x06	对应 CC 检测电压值 6V
		CC 连接状态 2	0x04	对应 CC 检测电压值 4V
	交流充电	CC 断线空闲	0x12	同直流
		CC 连接成功	0x00	对应 CC 检测电压值 0V
		充电状态 1	0x01	对应 CP 检测电压值 12V
		充电状态 2	0x02	对应 CP 检测电压值 9V
	充电状态 3	0x03	对应 CP 检测电压值 6V	
导引异常	导引检测异常	0Xff	CC/CP 检测到非标准电压	
模块状态	运行正常		0x00	扩展板正常运行
	通信失败		0x01	
	校准失败		0x02	校准失败，SF 灯闪烁
	访问超时		0Xff	无法访问扩展板

4.10.4 RS485 扩展板规范

端子定义

RS485 扩展板的结构图和端子示意图如下所示，后续表格中列出了各个端子位的定义，安装和连接扩展板时，请严格遵照端子位定义进行连接。

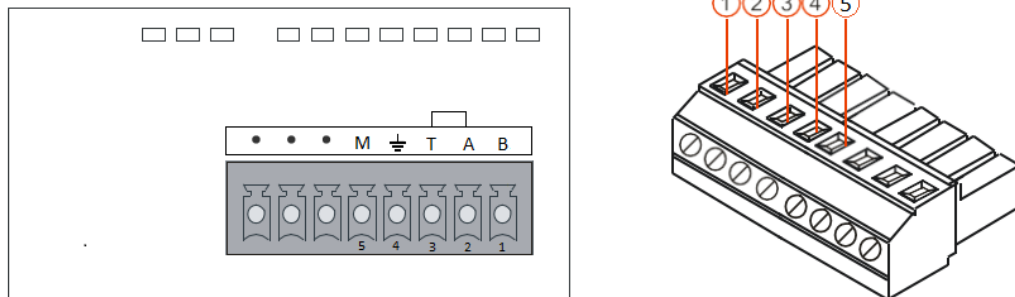


表 4-10-13 RS485 端子位定义

端子位	信号定义
1	RS485信号 B/+
2	RS485信号 A/-
3	终端电阻
4	机壳接地
5	RS485信号 GND

备注：需要使用终端电阻时，将 2 脚和 3 脚短接。

4.10.5 BD 扩展板的安装

扩展板的具体安装操作步骤如下：

- 1、确保 CPU 和所连设备均已断电，平放 CPU 主控模块；
- 2、沿扩展板左侧取下盖板；
- 3、将扩展板模块插入板槽中的插座上，确保插针与插孔紧密贴合；
- 4、将扩展板盖板右侧凸缘倾斜插入主控模块上的扩展板槽侧面插口；
- 5、斜向下压扩展板盖左侧，将其重新装回主控模块上。



提示

- CTH200 系列 CPU 产品全部支持模拟量输入输出扩展板和导引电路扩展板，不过，仅 H224X/H226XL/H226XM/H228XL 支持 CAN-01 主站扩展板。
- CPU 可搭载 CAN-01 主站扩展板直接使用，但是模拟量输入输出扩展板必须通过调

用专用的**扩展板指令库**才能使用，具体用法请参见本文档章节 [6.4 模拟量输入输出扩展板应用](#)。

- 请勿带电插拔扩展板。

5 网络通信协议及组网方式

CTH200 系列 PLC 支持多种网络通信方式，包括常见的串口网络通信以及复杂的以太网通信。支持 PPI、MPI、自由口、CANopen、以太网、profibus-DP 和 PROFINET 七种通信协议。以下为各协议的定义，可根据需要进行选择。具体通信方式和连接请参见本文档章节 [2.2 与 CTH200 建立通信](#)。

5.1 PPI 通信

PPI 是一种主-从协议。工作时主站发送要求到从站，从站响应，主站不发送要求，从站处于等待状态。CTH200 支持 PPI 协议，可通过普通的两芯屏蔽双绞电缆进行联网。该协议支持 3 种波特率：9.6kbps、19.2kbps 和 187.5kbps。CTH200 系列 CPU 上集成的编程口同时也是 PPI 通信联网接口。PPI 通信简单方便，只用 NETR 和 NETW 两条语句即可进行数据信号的传递，不需再额外配置模块或软件。PPI 通信网络是一个令牌传递网，在不加中继器的情况下，最多可以由 31 个 CTH200 系列 PLC、Copanel 系列 HMI、西门子 OP/TP 面板或上位机插 MPI 卡为站点构成 PPI 网。

单主站 PPI 网络

下图给出了一个主站带一个从站的网络示例，编程站 PG\PC 作为主站，CTH200 作为从站。



下图中 Copanel HMI 作为主站，CTH200 作为从站。

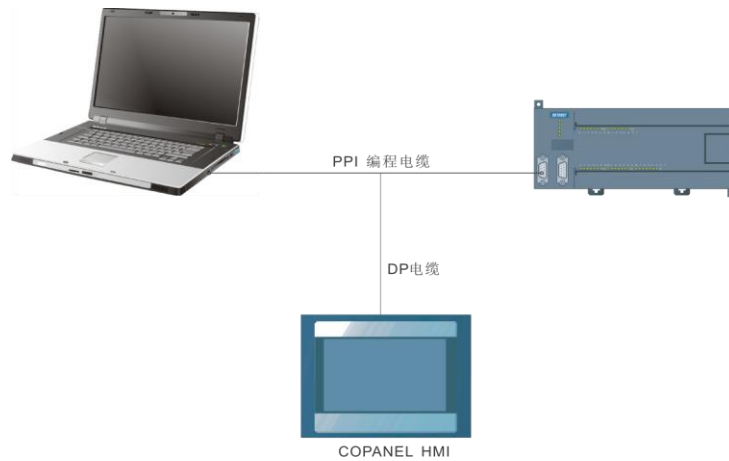


图 5-1 单主站 PPI 网络

注：PPI 编程电缆推荐使用本公司的 PLC 编程电缆。不同的 HMI 请选用对应的通信电缆，以免损坏连接设备。（电缆订货号：CTS7 191-USB30）

多主站 PPI 网络

下图给出了两个主站带一个从站的网络示例，编程站 PG\PC 和 HMI 作为主站，CTH200 作为从站。



下图给出了多个主站带多个从站的网络示例，编程站 PG\PC 和 HMI 作为主站，CTH200 作为从站。

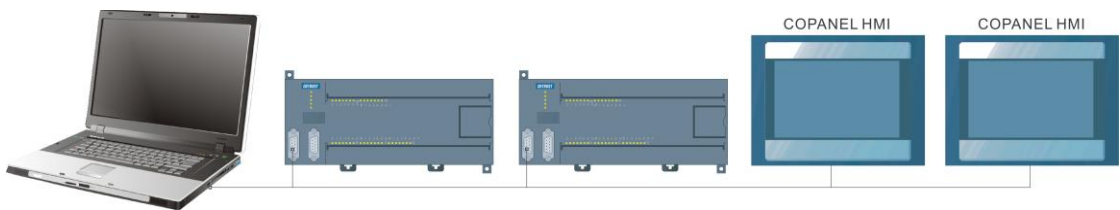


图 5-2 多主站 PPI 网络

下图给出了多个 CPU 互连的网络示例，多个 CTH200 CPU 互连，通过网络读写指令 NETR/NETW 实现多 CPU 之间的网络通信。网络通信过程中每个 CTH200 CPU 既是主站也是从站。各个 CPU 通过 PPI 口进行通信连接。

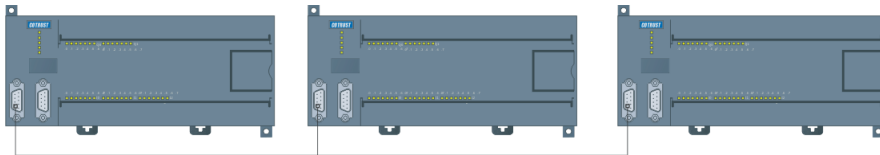


图 5-3 多 CPU 互连网络

5.2 MPI 通信

MPI 支持主-从、主-主通信方式。

CTH200 可以通过内置接口连接到 MPI 网络上，波特率为 19.2Kbps、187.5Kbps，它可与西门子公司 SIMATIC S7-300/S7-400 CPU 进行通信。CTH200 CPU 在 MPI 网络中只能作为从站。

如图 S7-300 和 Copanel HMI 作为 MPI 网络中的主站，CTH200 CPU 作为从站设备。S7-300 主站 PLC 通过 XGET 和 XPUT 指令来读写 CTH200 的数据。

多主站 MPI 网络

下图给出了多个主站带一个从站的网络示例，编程站 PG\PC、S7-300 和 HMI 作为主站，CTH200 作为从站。

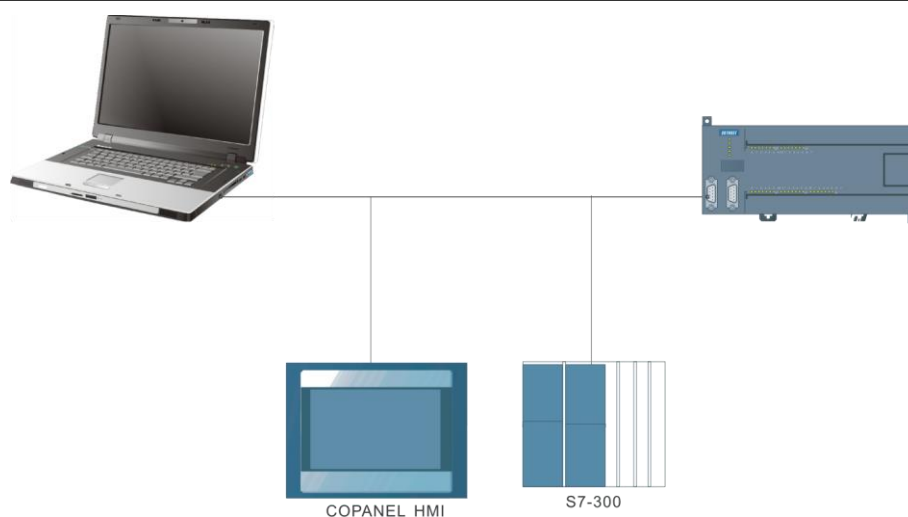


图 5-4 多主站 MPI 网络

5.3 自由口通信

自由口通信方式功能的实现使得 CTH200 可以与任何通信协议公开的第三方设备进行通信，即 CTH200 可以由用户自己定义通信协议。第三方设备例如数据采集模块、控制器、打印机或条形码阅读器、变频器、调制解调器和上位 PC 机等等。

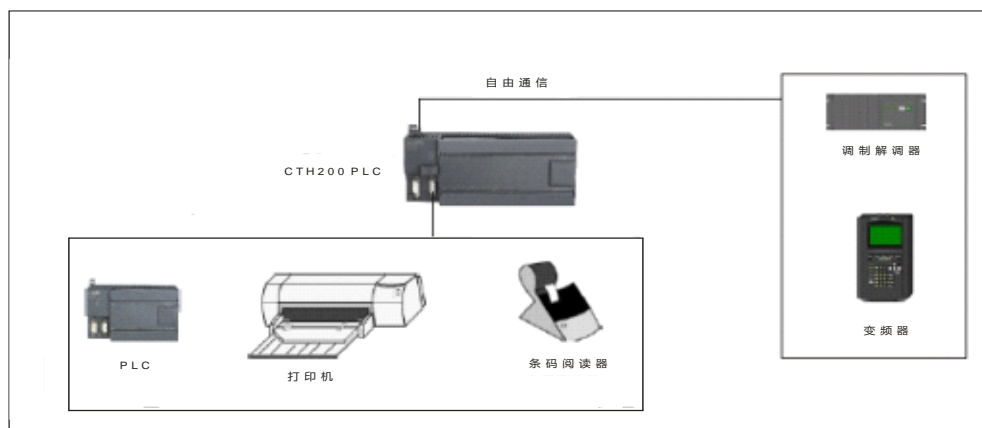


图 5-5 自由口通信方式

5.4 CANopen 通信

CANopen 总线通信是一种主从网络，速度快，可靠性高。在 CTH200 的 CANopen 通信系统中，H224X/H226XL/H226XM/H228XL 本体上扩展一个 CAN 主站通信扩展板做 CAN 主站，可通过 EasyCAN 或 CAN 硬件组态实现 CANopen 通信，对远端 CAN 从站 CTH2 277-0AC32 及其后面扩展的 CTH200 扩展板进行数据交换，主站也可以直接与其他支持 CANopen 通信的伺服驱动器（如合信 E10）、变频器等设备进行 CANopen 通信。

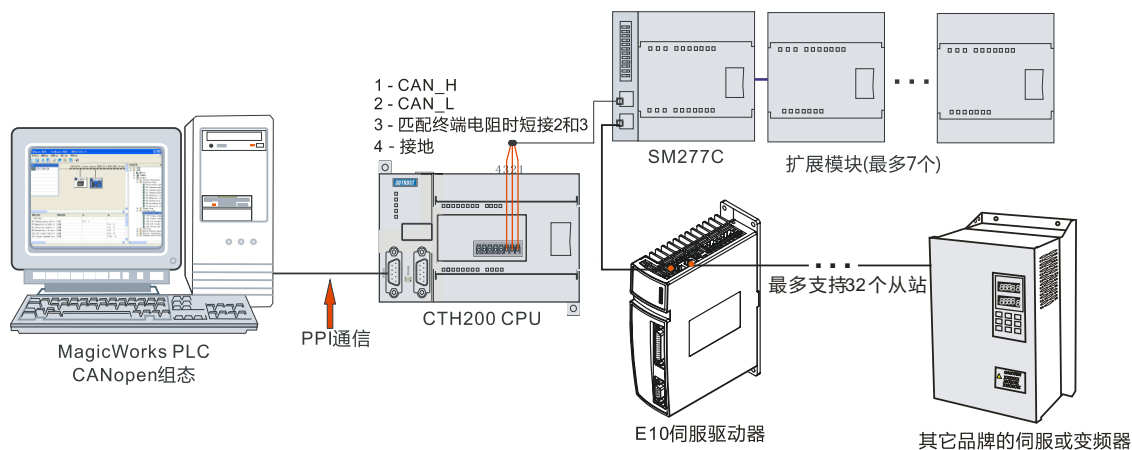


图 5-6 CANopen 通信网络架构

在上述网络架构中，CTH200 通过 CAN-01 扩展板连接 SM277 CANopen 通信从站模块，随后挂接配套的 CTH200 扩展模块，即可对伺服系统进行控制。

扩展板的相关端子和接线操作信息请参见章节 4.10.2 CAN 扩展板规范。

5.5 以太网通信

CTH200 系列 PLC 支持以太网通信，支持 Modbus TCP、UDP_PPI 和 S7 通信协议，也可使用合信云服务器 Mico 进行远程通信、监控和数据交换。

CTH200 系列 PLC 通过本体标配以太网口可连接其他 CPU 模块、触摸屏或是计算机进行通信，可快速便捷地完成组网。支持 Modbus/TCP, UDP/PPI、S7 协议实现以太网本地通信，同时也支持远程编程、调试、监控及数据交换，用户可通过 Mico 云平台实时查看远程设备状况。Modbus TCP 通信应用举例请参考章节 6.2 Modbus TCP 通信，UDP/PPI 通信应用举例请参考章节 6.3 UDP PPI 通信，S7 协议通信应用举例请参考章节 6.4 西门子 S7 协议通信举例，有关远程编程和监控详细操作方法请参考手册《Mico 远程监控系统手册》和《如何使用 Mico 对远程 PLC 进行编程维护和数据监控》，手册下载地址：<http://www.co-trust.com/Download/index.html>

CTH200 系列 PLC 可以通过以太网通信端口接入网络，与接入相同网域的上位机进行通信，使用 MagicWorks PLC 可以在网络中对 PLC 进行编程调试和监控。其典型的网络架构如下图所示：

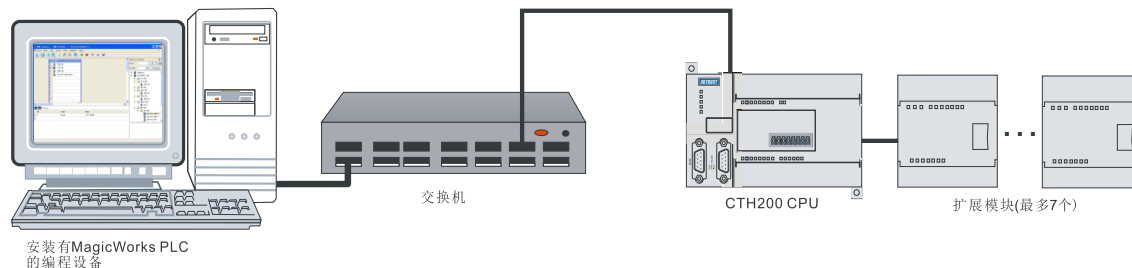


图 5-7 以太网本地通信网络架构

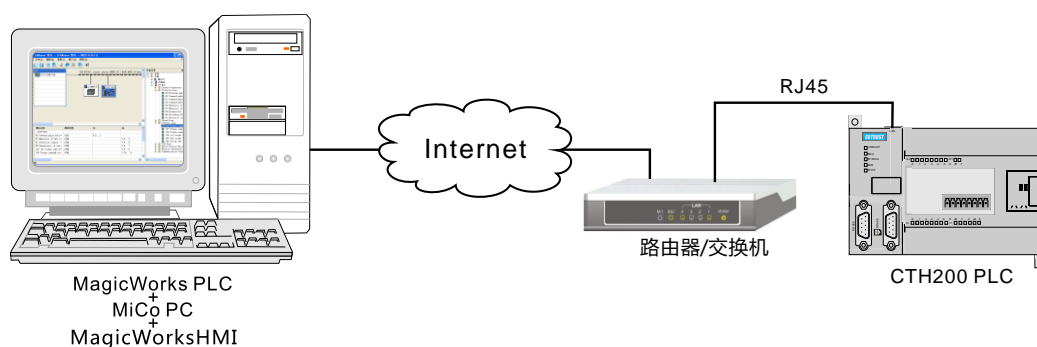


图 5-8 以太网远程通信网络架构

- 波特率：10/100Mbps 自适应；
- 协议类型：UDP/TCP 协议，支持 UDP_PPI，MODBUS_TCP，Socket，S7 协议，支持 PLC 间以太网通信，支持 MiCo 以太网编程。
- 通过上位机进行通信；
- 高性能型升级版 CPU 支持的最大连接数：8 个 S7 协议连接，不分主从，合信 CPU 只做从站，最多支持 8 个 UDP_PPI，不分主从；8 个 Modbus_TCP，不分主从；4 个 Socket 连接，2 个 UDP，2 个 TCP；
- 原平台 CTSC200 CPU 支持的最大连接数：最多支持 8 个 UDP_PPI 连接，其中 4 个主站和 4 个从站；4 个 Modbus_TCP 连接，2 个主站和 2 个从站。
- 支持以太网 PPI 读写指令 UDPNETR、UDPNETW 和以太网 MODBUS 读写指令 TCP_MBUS_MSG；

5.6 Profibus-DP 通信

通过 DP 从站模块，CPU 可以作为某个主站的从站连接到 Profibus 现场总线。既能快速实时地交换数据又能独立运行，具有较高可靠性。SM277A 和 SM277B DP 从站 profibus-DP 网络结构分别如下图 5-9 和 5-10 所示。

有关 SM277A 和 SM277B DP 从站模块具体使用方法请参见附录 G [SM277A 模块的使用](#)和附录 H [SM277B 模块的使用](#)。

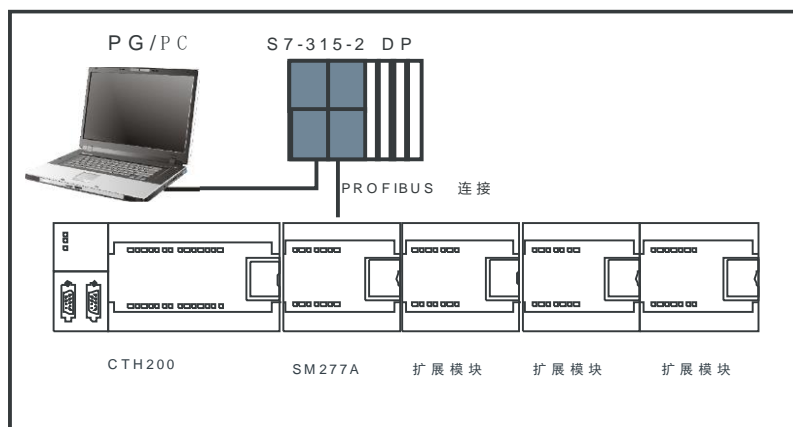


图 5-9 SM277A Profibus-DP 网络的结构

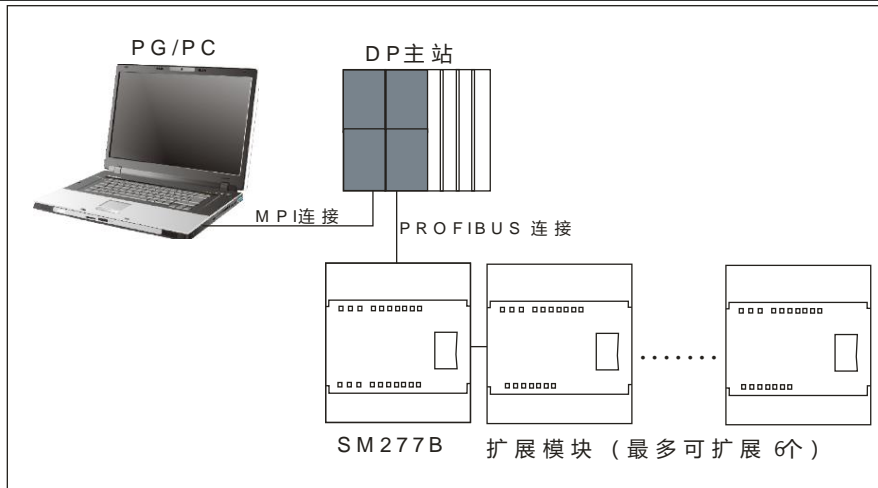


图 5-10 SM277B Profibus-DP 网络的结构

5.7 PROFINET 通信

扩展模块可通过 SM277PN 从站模块连接到 PN 主站，作为其远程 I/O 扩展。每个从站模块最多可扩展 8 个 I/O 模块。PROFINET 通信典型网络架构如下图所示。

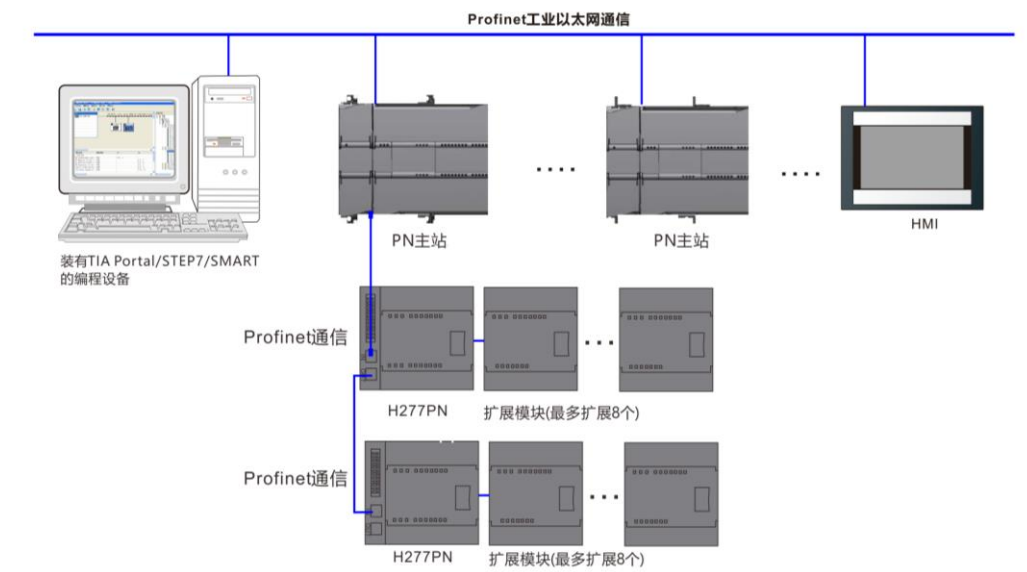


图 5-11 PROFINET 网络架构

有关 PROFINET 通信模块详细信息和应用示例请参考《CTH200 系列 Profinet 从站模块使用手册》，手册下载地址：<http://www.co-trust.com/Download/index.html>

6 应用举例

本章节通过一些具体示例帮助您进一步了解 CTH200 系列 CPU 各项网络通信功能。

6.1 CANopen 通信

本节将通过具体实例来展示实现 CANopen 通信的详细操作步骤，使用 CTH200 系列 CAN 主站扩展板搭配 CPU 组成 CANopen 网络实现 CANopen 通信功能。

6.1.1 示例组件

CANopen 网络示例组件

表 6-1 示例组件

组件	描述
一台装有 MagicWorks PLC 的 PG/PC	请使用 MagicWorks PLC V2.8 及以上版本
一条 PLC 编程电缆	连接编程站与主站
一个 CTH200 主控模块	CTH200 系列 PLC 作为 CANopen 主站
一个 CAN 主站扩展板	CAN-01 主站扩展板
一个从站设备	SM277C
一条 CANopen 总线电缆	带屏蔽的网线或双绞线
三个可以连接 SM277C 的扩展模块	两个 CTH200 系列数字量模块和一个模拟量模块
两个伺服驱动器	一个 E10 伺服驱动器和一个其他品牌伺服

示例组件说明：

PG/PC

请确认 PG/PC 编程站已经与 CANopen 主站相连接（通过 PPI 方式）。本例中编程软件须使用 MagicWorks PLC V2.8 及以上版本。

PLC 编程电缆

连接 PG/PC 编程站和 CANopen 主站 CPU 的通信电缆，用来下载网络硬件配合工程程序，监控数据。

CANopen 主站

本例中采用 CTH200 系列 CPU H226XL 主控模块搭载 CAN-01 主站扩展板作为 CANopen 主站。

CAN 主站扩展板

CTH200 系列主站扩展板，搭配同系列 CPU H224X/H226XL/H226XM/H228XL 使用，可将 CPU 扩展为具有 CANopen 主站功能。

CANopen 总线电缆

CANopen 总线电缆可采用带屏蔽的网线或双绞线。

SM277C

在本例中，采用了配备有 CAN 双通信口和波特率拨码开关的 SM277C 作为 CANopen 从站。

扩展模块

SM277C 搭配扩展模块使用，最多可连接 7 个扩展模块。所有扩展模块均属于 CTH200 产品系列。

伺服驱动器

合信 E10 等伺服驱动器，也可搭配其他品牌伺服使用。

6.1.2 网络连接

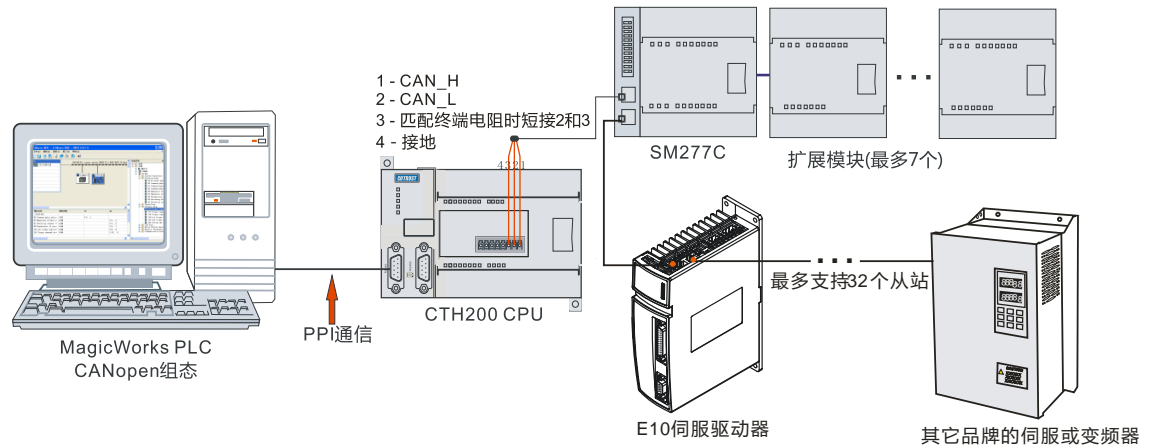


图 6-1 CANopen 通信网络架构

CANopen 总线连接 CPU 的端子接线如下图所示，网线接法请参见章节 4.2.5 制作标准网线。

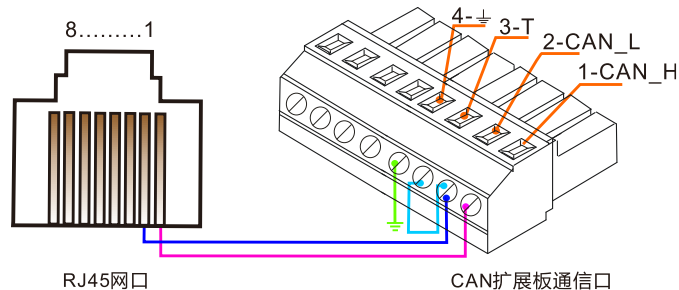


图 6-2 CANopen 总线连接

注：匹配终端电阻时，请将 2 与 3 短接。

6.1.3 操作步骤

编程软件 MagicWorks PLC 提供两种 CANopen 通信组态方式，CPU 搭配不同的 CAN 扩展板可进行不同的 CAN 通信。

方式一：通过 EasyCAN 组态 CANopen 通信（使用扩展板 CTH2-CAN-01S1-EB）

1、打开 EasyCAN 组态界面

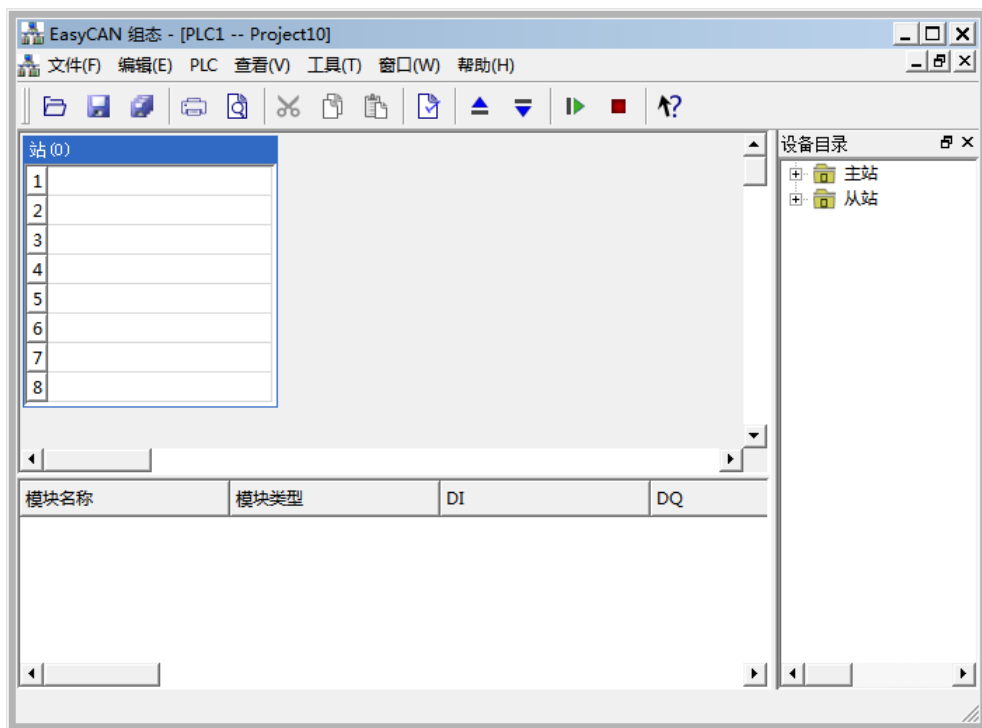


图 6-3 EasyCAN 组态页面

2、添加主站

CANopen 总线编辑器如下所示，在设备目录树的主站列表下选择与当前实际设备相符的 CTH200 系列 PLC（这里以 CPUH226XL 为例），鼠标选中双击或拖拽放入主站“站 0”下的“插槽 1”中。

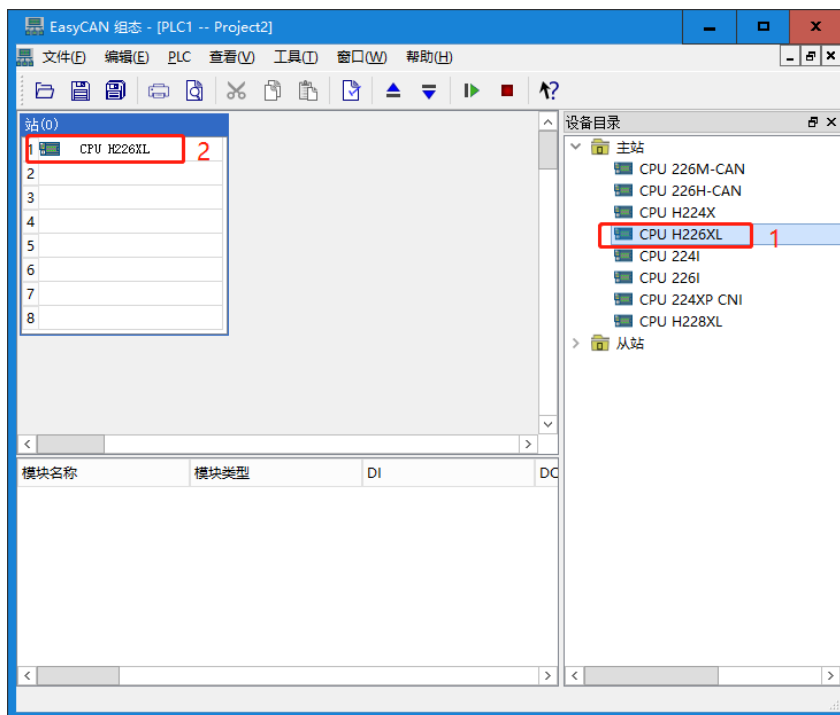


图 6-4 添加主站

然后，双击插槽 1 中的主站 CPU H226XL，会弹出主站信息对话框，如下图：

在此，我们设置通信速率为 1000kbps，选择 CPU 作为主站类型，点击“节点 ID”下拉框选择 1，勾选“使用 CAN 总线”。



图 6-5 主站信息配置

3、配置从站

1) 添加从站

鼠标单击展开右边的设备目录树，打开从站节点，选中与你当前实际设备相符的从站设备型号，鼠标选中拖拽进入组态界面 CAN BUS 区域放下，系统会自动将从站与 CAN BUS 总线相连。如图：

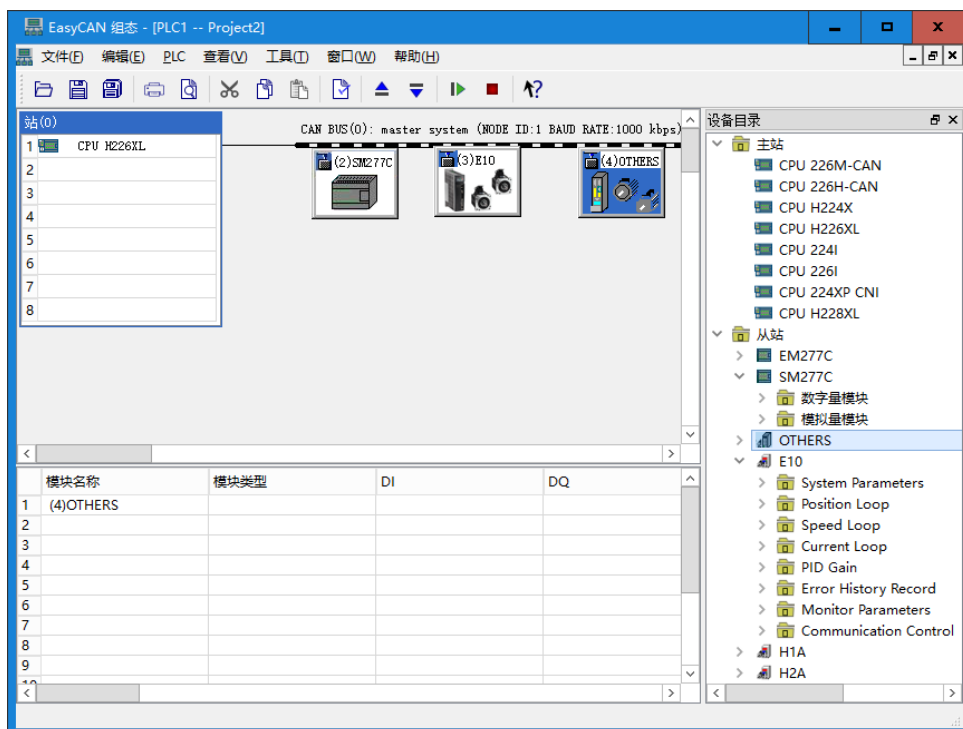


图 6-6 添加从站

2) 设置从站参数

双击 CAN BUS 下连接的从站图标，即弹出从站信息对话框，如下图：

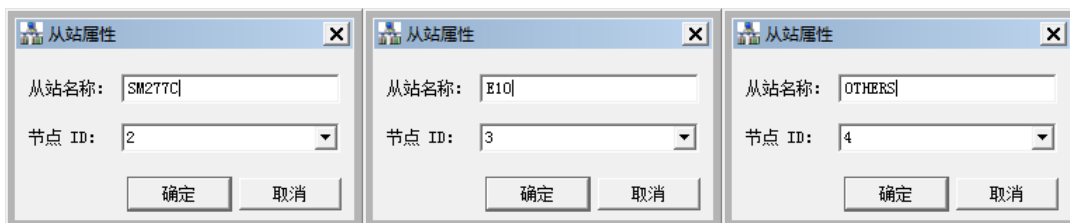


图 6-7 从站属性设置

在此，您可以设置如下从站参数：

- 从站名称：从站的名称（默认为 SM277C、CT E10 或 OTHERS）。
- 节点 ID：范围从 1~127。

3) 为从站 SM277C 组态扩展模块

鼠标单击展开右边的设备目录树，打开从站 SM277C 节点下的数字量模块或模拟量模块节点，选中与你当前实际设备相符的模块型号，鼠标选中拖拽进入模块插槽放下，模块即被添加到当前从站下。您可以通过双击设备目录树上的模块节点将其按顺序插入到当前从站下。模块插入后，系统会自动重排其对应的 I/Q 地址。

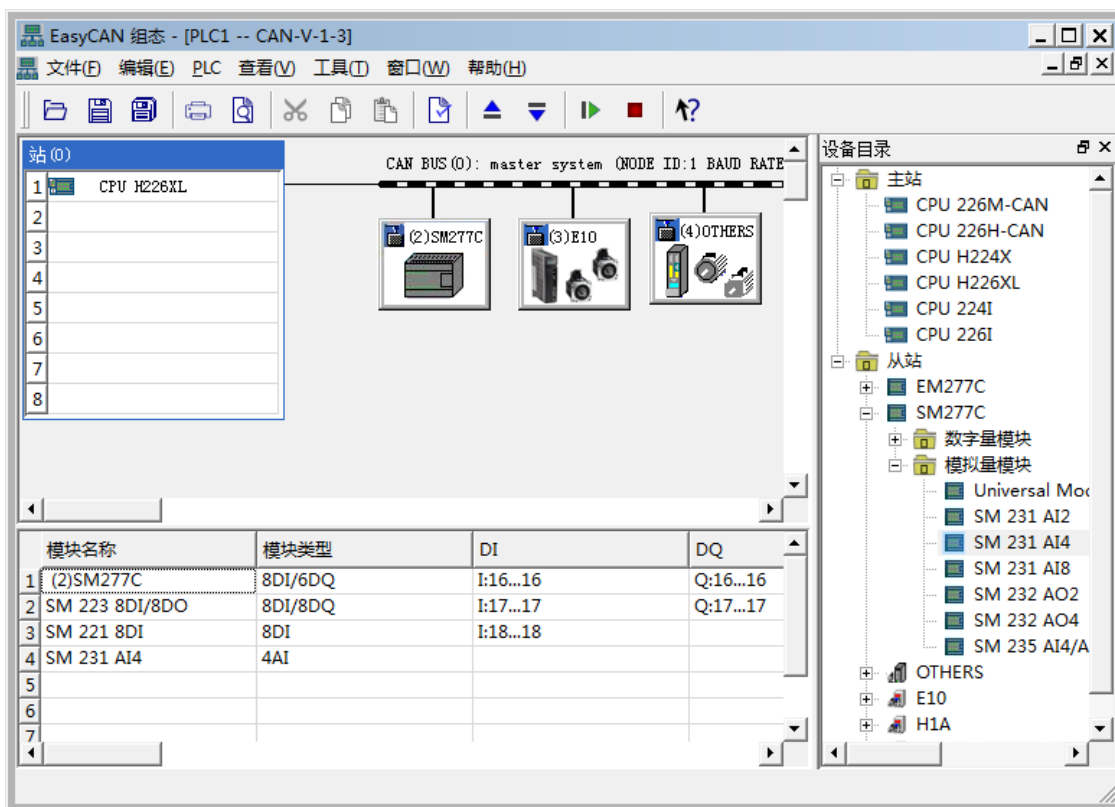


图 6-8 添加扩展模块

修改 IO 配置

MagicWoks PLC 组态软件 V2.08 及以上的版本支持用户自定义 IO 的起始地址，并在映射内存中增加了 V 内存，允许用户自己选择映射内存。双击模块列表中的非空白行，会弹出如下对话框：

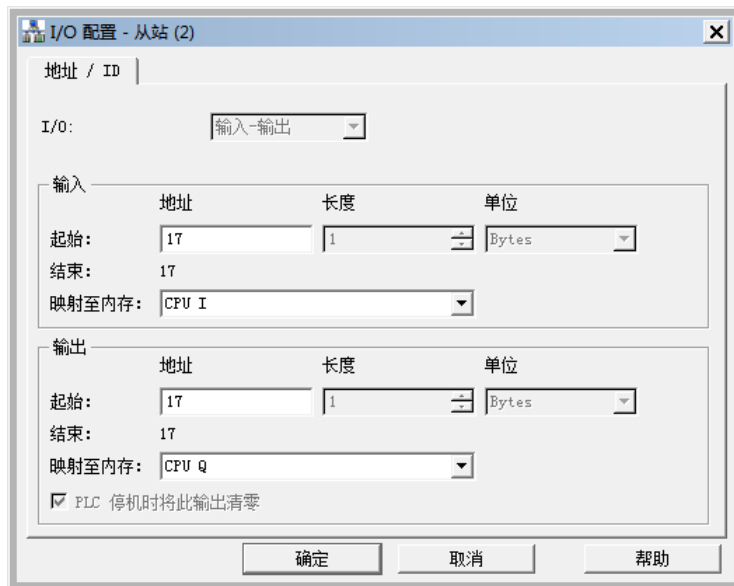


图 6-9 I/O 配置

在此，您可以修改模块对应 IO 的起始地址和映射内存，可选的 CPU 映射内存与当前所选参数的读写类型有关：

数字量输入：可映射到 CPU 的 I 内存和 V 内存；

数字量输出：可映射到 CPU 的 Q 内存和 V 内存；

模拟量输入：可映射到 CPU 的 AI 内存和 V 内存；

模拟量输出：可映射到 CPU 的 AQ 内存和 V 内存；

表 6-2 各内存有效的地址范围

内存	地址范围
I	IB16~IB79
Q	QB16~QB79
AI	AIW64~AIW386
AQ	AQW64~AQW386
V	VB0~Vmax

特别说明：对确定类型的模块，I/O 类型和输入/输出长度是不可修改的，除非您编辑的模块类型是 Universal Module，点击确定时，系统会自动检查所选地址的合法性。如果 SM277C 节点下找不到相应型号的模块，可添加 Universal Module 并进行自定义。

4) 为从站 CT E10 组态参数

鼠标单击展开右边的设备目录树，打开 CT E10 从站节点下的各类参数，选择您需要的参数类型并展开，鼠标双击或拖动该参数进入从站列表放下，该参数即被添加到当前从站下。参数插入后，系统会自动给出最小可用的默认 I/Q 地址，用户可以双击该参数行修改配置。

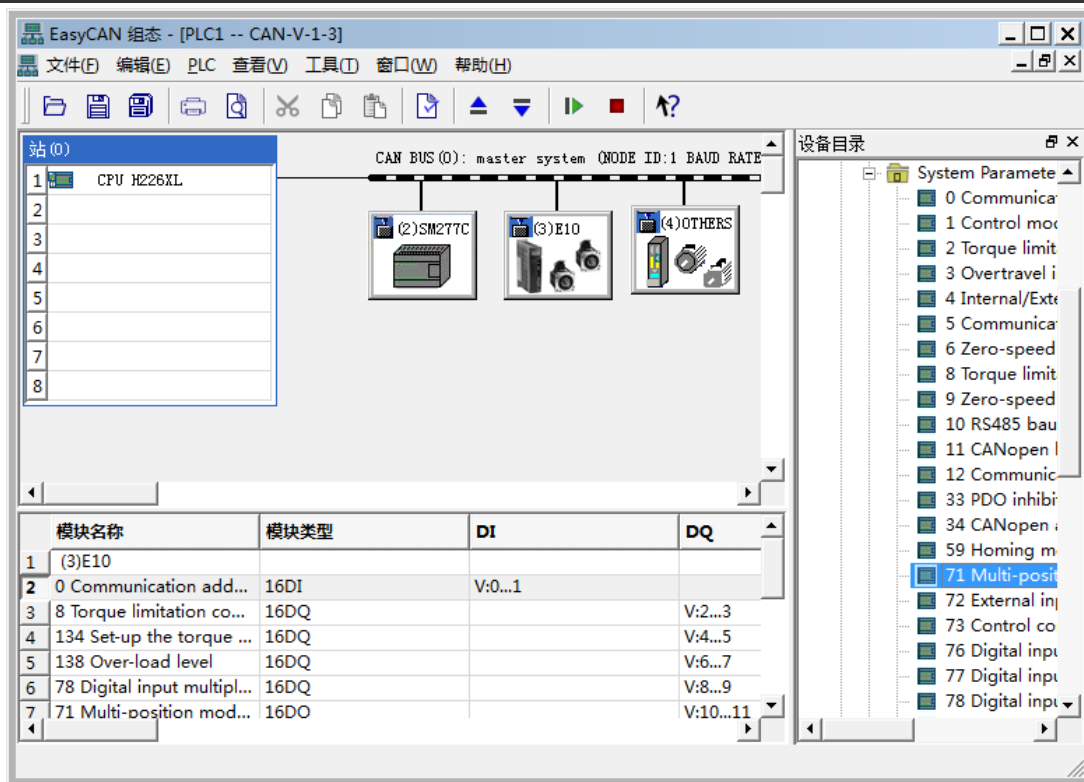


图 6-10 为从站 E10 配置参数

修改 IO 配置

双击模块列表中的任一参数，会弹出如下对话框：

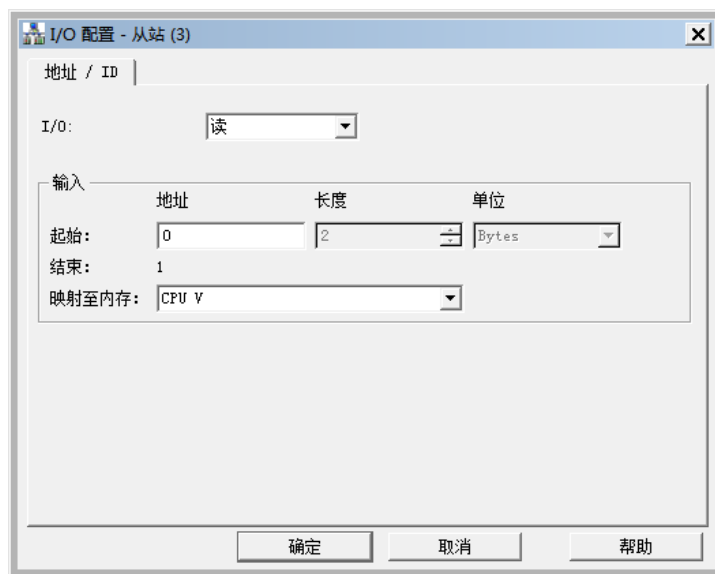


图 6-11 I/O 配置界面

I/O: 选择参数读/写类型。

起始

- 地址：设置参数的起始地址。
- 长度：使用默认值，不可编辑。
- 单位：使用默认值，不可编辑。

结束: 采用系统计算值。

映射至内存: 选择映射到 CPU 的内存类型，可选内存类型与当前所选参数的读写类型有关。

5) 为第三方从站组态参数

MagicWorks PLC 的 V1.73 版本开始允许组态第三方 CANopen 从站，鼠标单击展开右边的设备目录树，打开 OTHERS 从站节点下参数，鼠标双击或拖拽参数进入从站列表放下，该参数即被添加到当前从站下。参数插入后，系统会自动给出最小可用的默认 I/Q 地址，用户可以双击该参数行对其配置进行修改。

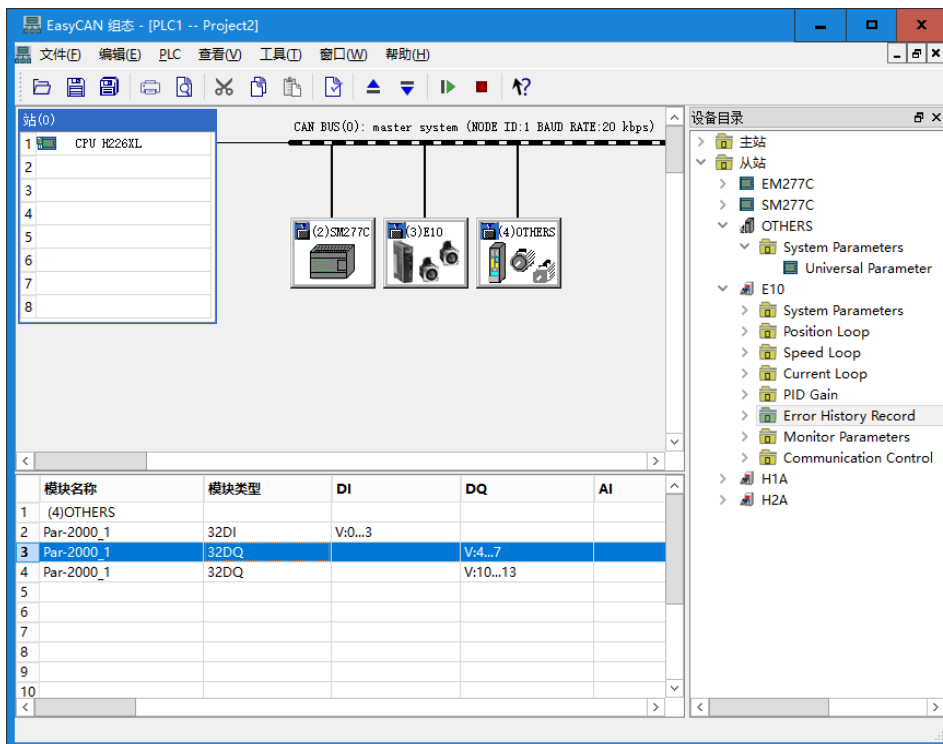


图 6-12 配置第三方从站

修改 IO 配置

双击模块列表中的任一参数，会弹出如下对话框：

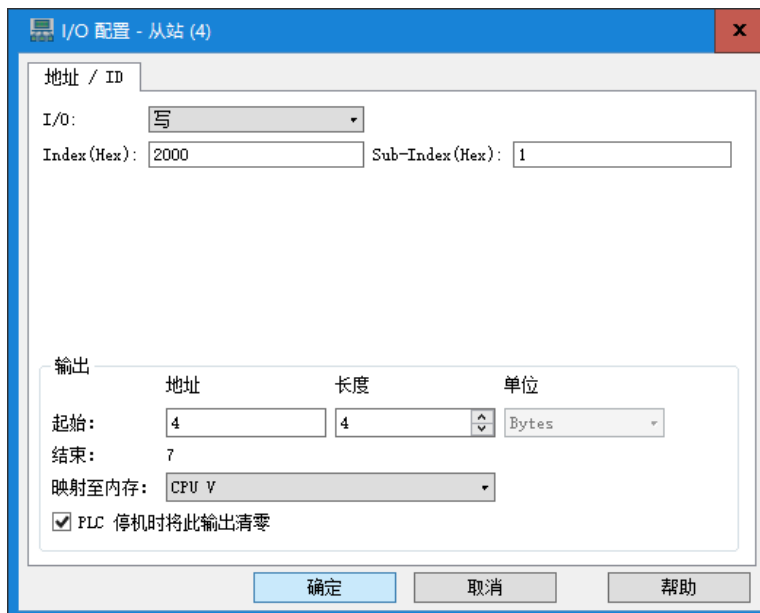


图 6-13 I/O 配置

I/O: 选择参数读/写类型。

Index (Hex) : 所选从站参数相应的主索引。

Sub-Index (Hex) : 所选从站参数相应的子索引。

起始

- 地址：设置参数的起始地址。
- 长度：要求组态的读/写数据的总长度是 4 的倍数（即最小为 4 个字节）。
- 单位：使用默认值，不可编辑。

结束：采用系统计算值。

映射至内存：选择映射到 CPU 的内存类型，有 V、Q 两种可选。

PLC 停机时将此输出清零：勾选此项，可以将映射地址中的内容清零，若不勾选，该输出地址内容保持不变。



提示

- 内存有效的地址范围请参考表 6-2；
- 实际连接时，如果第三方从站检测到错误，在其对应的 SMB 信息区会显示：0x7 组态参数错误。
- 有关合信伺服产品的相关使用，请访问合信官网获取信息：<http://www.co-trust.com>

4、硬件连接和配置

1) 用通信电缆连接 CPU H226XL 与 PC（电缆的 USB 端连接 PC 的 USB 插口，电缆的 RS485 端连接 CPU H226XL 的 RS485 通信口）。

2) 使用通信电缆连接主控模块上的 CAN-01 主站扩展板 CAN 通信口到 SM277C 的 CAN 通信网口。

3) 设置拨码开关（通信速率为 1000kbps，节点 ID 为 4），按拨码开关参照表拨动 SM277C 的拨码开关，如下图所示：



图 6-14 DIP 开关选择

DIP7-DIP1（节点地址）：二进制数表示，DIP7 为高位，DIP1 为低位。

DIP10-DIP8（波特率）：二进制数表示，DIP10 为高位，DIP8 为低位。



提示

0 是全局地址，使用时禁止设为 0 地址。

表 6-3 拨码开关配置

DIP10	DIP9	DIP8	通信速率 (kbit/s)	最大通信距离 (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（终端电阻）：设备处于网络端头处时需要拨到“ON”的位置，其他拨到“OFF”的位置

DIP12：保留

1) 依次将数字量模块 SM223 8DI/8DO、SM221 8DI，以及模拟量模块 SM231 AI2*16BIT 挂接到 SM277C 的扩展口后面。

2) 设备连接完成后，检查接线并确认无误。

3) 接通系统各设备的电源。

5、下载程序

CANopen 组态完成后，保存当前工程，接着在 EasyCAN 组态界面编译已经保存的组态，编译成功后，下载 CANopen 配置到 PLC。

6、监控状态表

CPU 扩展了 64Byte I / 64Byte Q / 162Words AI / 162 Words AQ 内存给 CANopen 作为各模块的地址映射。EasyCAN 组态中的模块的 I/Q 地址如下图所示，此时可在状态表中监控各模块的输入输出。

模块名称	模块类型	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			
7			

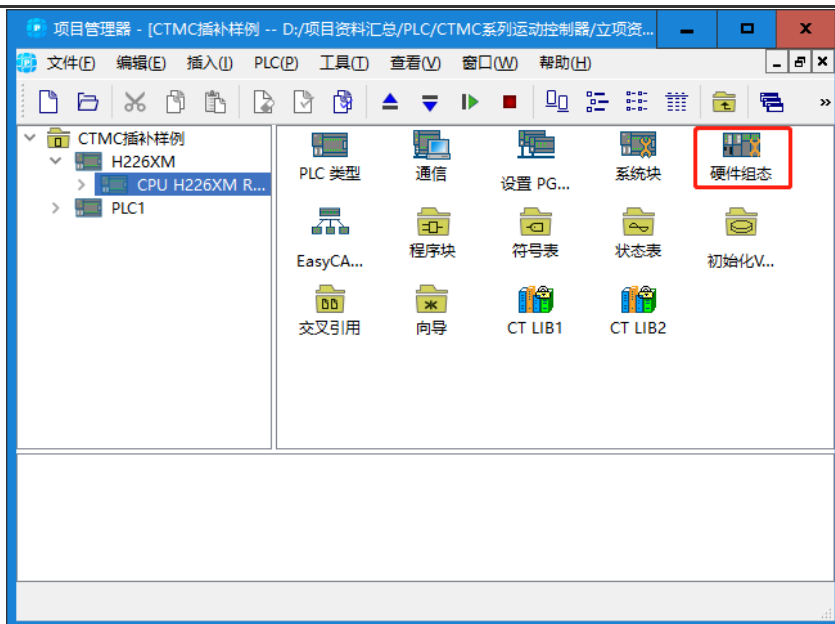
组态完成后，可在状态表中监控 SM277C、SM221 8DI、SM223 8DI/8DO 和 SM231 AI4*12Bit/16Bit 的输入输出情况，如下图：

地址	格式	当前值	
1 Q16.0	位	2#1	SM 277C的6路输出(Q16.0~Q16.5)
2 Q16.1	位	2#1	
3 Q16.2	位	2#1	
4 Q16.3	位	2#1	
5 Q16.4	位	2#1	
6 Q16.5	位	2#1	
7 Q17.0	位	2#0	SM 223 8DI/8DO的8路输出(Q17.0~Q17.7)
8 Q17.1	位	2#1	
9 Q17.2	位	2#1	
10 Q17.3	位	2#0	
11 Q17.4	位	2#0	
12 Q17.5	位	2#1	
13 Q17.6	位	2#1	
14 Q17.7	位	2#1	
15 Q18.0	位	2#0	SM 231 8DI的8路输入(I18.0~I18.7)
16 Q18.1	位	2#0	
17 Q18.2	位	2#0	
18 Q18.3	位	2#0	
19 Q18.4	位	2#0	
20 Q18.5	位	2#0	
21 Q18.6	位	2#0	
22 Q18.7	位	2#0	
23 AIW64	有符号	-32768	SM 231 4AI的4路输入(AIW64~70)
24 AIW66	有符号	-32768	
25 AIW68	有符号	-32768	
26 AIW70	有符号	-32768	

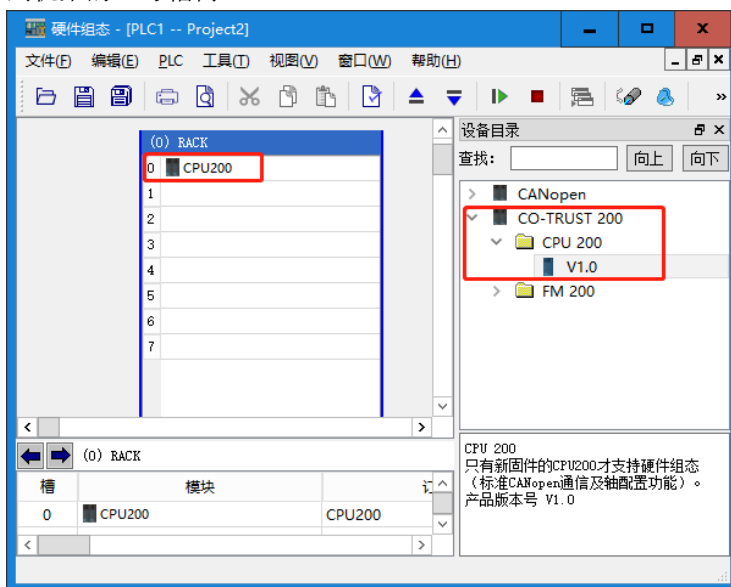
图 6-15 监控表中模块输入输出状态

方式二：通过硬件组态方式组态 CANopen 通信（使用扩展板 CTH2-CAN-01S2-EB）

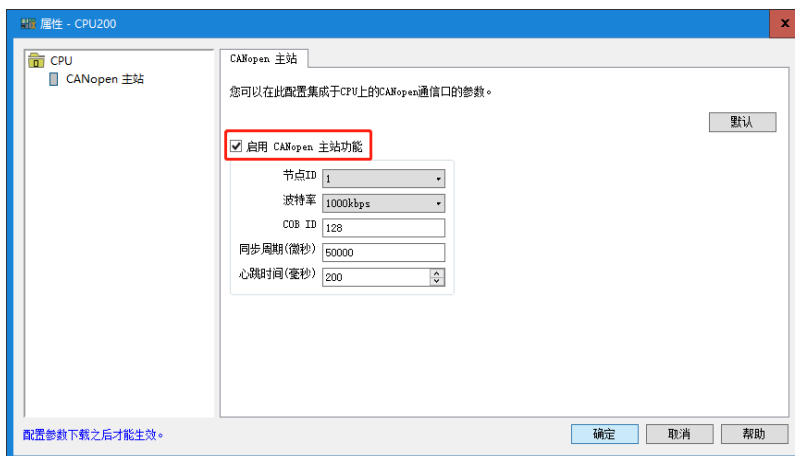
1、在项目管理器界面双击“硬件组态”图标，打开硬件组态界面。



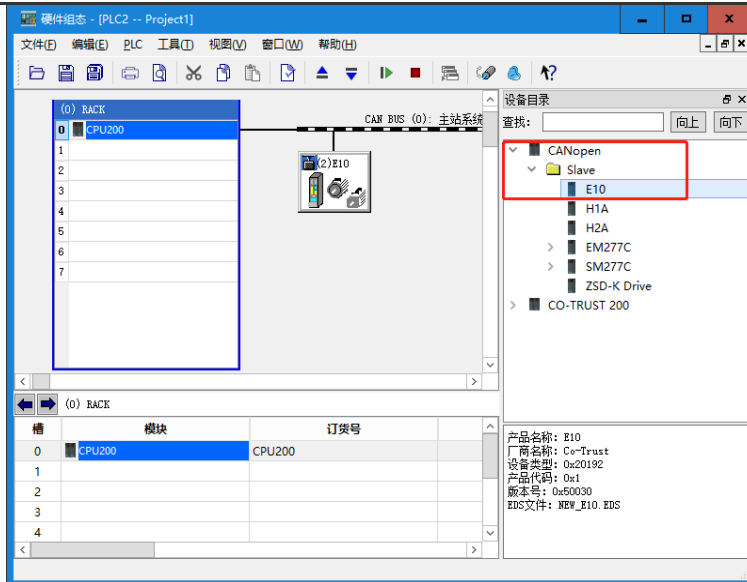
2、在硬件组态界面“设备目录”下依次展开“CO-TRUST 200”→“CPU 200”，将 CPU 拖拽到机架的 0 号槽内。



3、双击机架上的“CPU200”可进行 CPU 属性配置，勾选“启用 CANopen 主站功能”将在 CPU 后自动生成 CAN 总线。

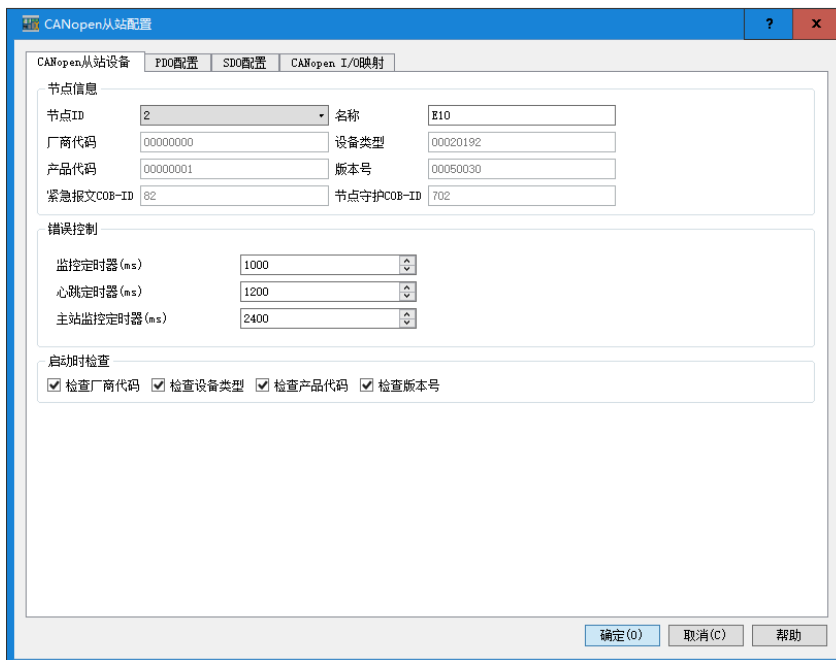


4、在硬件组态界面“设备目录”下依次展开“CANopen”→“Slave”选择目标从站拖拽至 CAN 总线上。



● 为伺服类从站组态参数

双击 CAN 总线上的伺服图标，打开其属性配置界面。



(1) “CANopen 从站设备” 选项卡

在“CANopen 从站配置”对话框的“CANopen 从站设备”选项卡中可以设置节点 ID、错误控制参数及 EDS 一致性检查，参考如下描述：

节点信息

参数	说明
节点 ID	节点在 CANopen 网络上的地址，用户可修改此节点 ID
名称	节点的设备名称，只读参数
产商代码	提供商的 ID 代码，只读参数
设备类型	节点的设备类型代码，只读参数
产品代码	节点的产品代码，其中 1 表示 E10 伺服驱动器、2 表示 H1A 伺服驱动器、3 表示 H2A 伺服驱动器，只读参数
版本号	节点 CANopen 功能的修订版本，只读参数
紧急报文 COB-ID	用来标识 CANopen 网络中各节点的紧急报文。节点紧急报文

	COB-ID=0x80+节点站号，只读参数
节点守护 COB-ID	用来标识 CANopen 网络中各节点的节点管理报文，只读参数

错误控制协议：设置错误控制的方式，支持 Heartbeat 协议。参数说明如下：

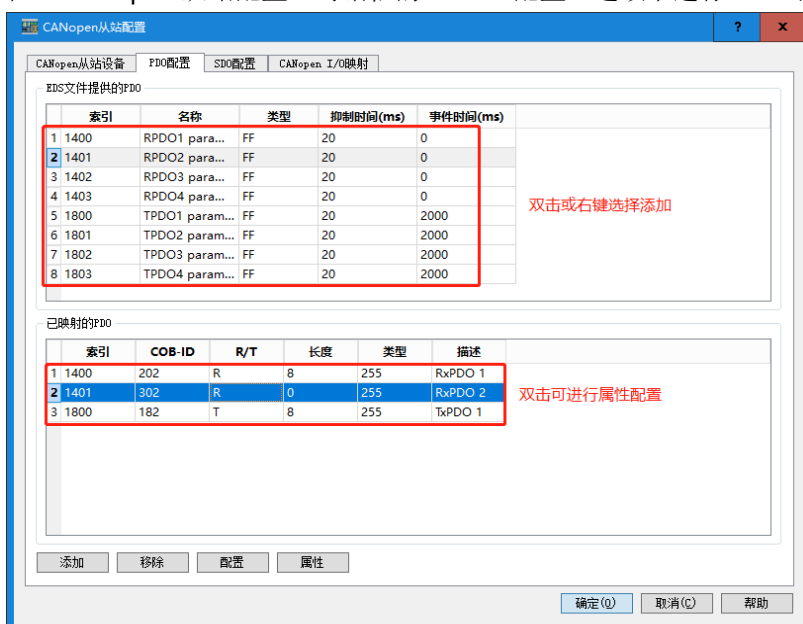
参数 (ms)	说明
监控定时器	从站监控时间，指从站在本监控时间内没有接到主站的心跳即认为掉线
心跳定时器	从站心跳时间，从站按“从站心跳时间”周期发送 Heartbeat 报文给主站
主站监控定时器	主站监控时间，如果主站在“主站监控时间”内没有收到从站的 Heartbeat 报文，主站就认为该从站掉线

选择 EDS 参数一致性检查

您可以选择要不要进行厂商代码、设备类型、产品代码和版本号的一致性检查，默认要检查。如果这四项有勾选，组态下载至 PLC 后，CANopen 从站会将组态中（EDS 文件中）的这四个参数与从站固件中相应参数进行对比，如果不一致，会停止 CANopen 通信，同时在 SM 内存给出错误信息。如果去掉勾选，组态下载后不会进行检查，这样即使组态软件中使用的 EDS 文件版本与固件中不一致，通信成功但可能会导致读写参数的结果不对。

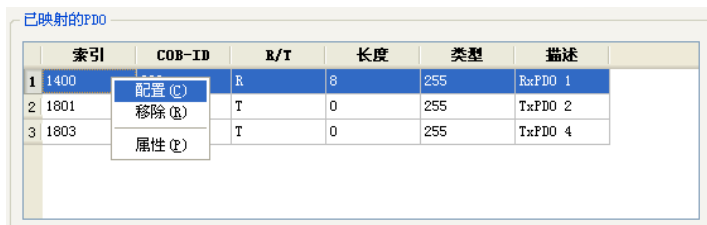
(2) “PDO 配置”选项卡

在“CANopen 从站配置”对话框的“PDO 配置”选项卡进行 PDO 配置，请按图示步骤操作：



<第 1 步>：在 PDO 配置对话框中通过双击或右键选择“添加”即可将所选 PDO 添加到“已映射的 PDO”列表中。

<第 2 步>：在“已映射的 PDO”列表中编辑 PDO 属性或配置 PDO。

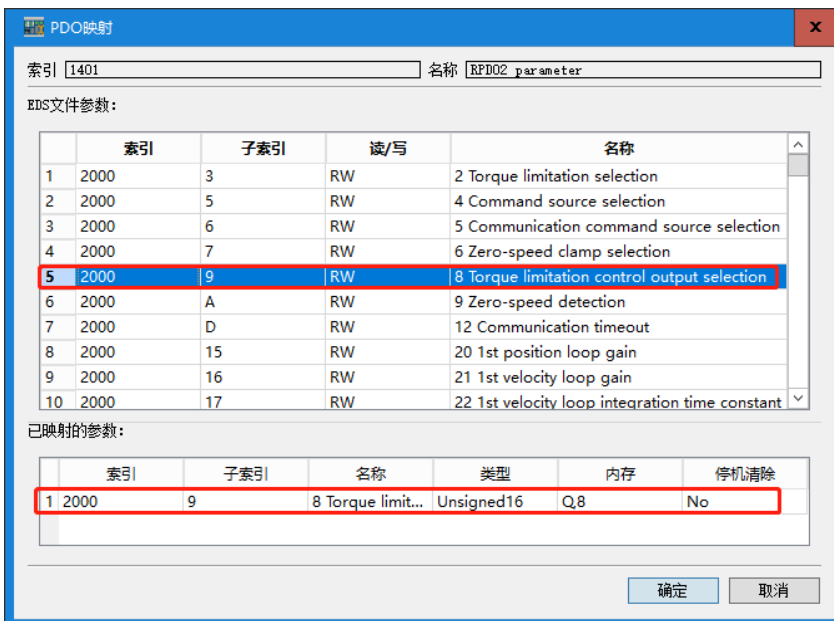


- 索引 1400 ~ 1403：接收 PDO 通信参数
- 索引 1800 ~ 1803：传送 PDO 通信参数
- PDO 映射：从站节点中需要和主站进行通信的参数
- RxPDO：CANopen 主站去写从站映射在 RxPDO 中的参数

- TxPDO: CANopen 主站去读从站映射在 TxPDO 中的参数

“配置”操作

在“已映射的 PDO”中选择一个对象后右键选择“配置”或点击该对话框下方的“配置”按钮即可打开该项的配置对话框，在该对话框中可以配置该组的参数如下图所示：



PDO 配置完成后，您可以在 CANopen I/O 映射页面查看所有 PDO 参数的映射地址。

“移除”操作

在“已映射的 PDO”中选择一个对象后右键选择“移除”或点击该对话框下方的“移除”按钮即可将该项从列表中删除。

“属性”操作

在“已映射的 PDO”中选择一个对象后右键选择“属性”或点击该对话框下方的“属性”按钮即可打开该项的属性对话框，如下所示：



COB ID: CAN 对象标识符地址信息。

事件定时器: 设定事件触发定时器的时间。

抑制定时器: 设定两个连续调用数据对象的传输服务之间的最小时间间隔。

传送类型: 选择 PDO 数据传输类型 (参考下表)。

传输类型	触发 PDO 的条件 (B = both needed, O = one or both)			PDO 传输
	SYNC	RTR	Event	
0	B		B	同步, 非循环
1~240	O		-	同步, 循环

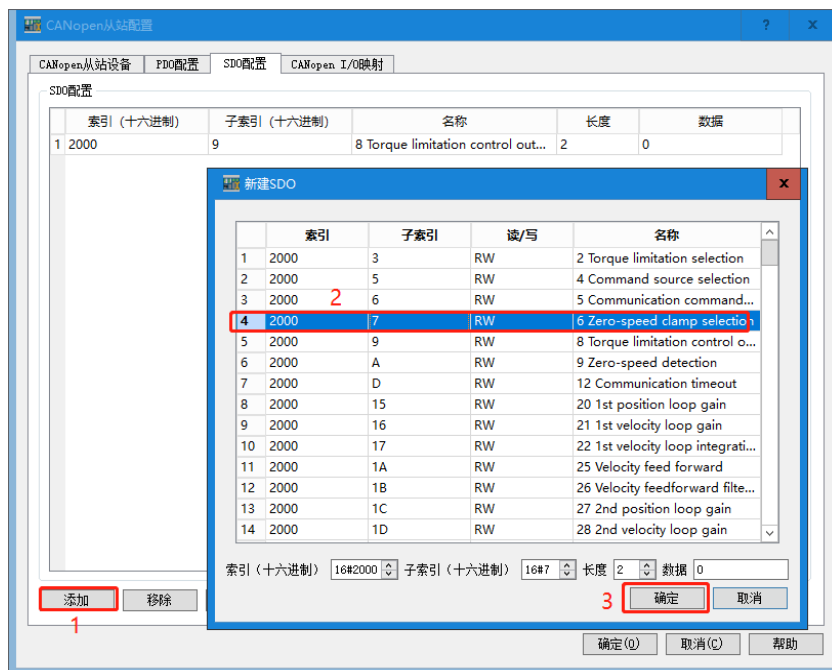
241~251	-		-	Reserved
252	B	B	-	同步，在 RTR 之后
253	-	O		异步，在 RTR 之后
254	-	O	O	异步，制造商特定事件
255	-	O	O	异步，设备子协议特定事件

说明：
SYNC — 接收到 SYNC-object。
RTR — 接收到远程帧。
Event — 例如数值改变或者定时器中断。
 传输类型为 1 到 240 时，该数字代表两个 PDO 之间的 SYNC 对象的数目。

(3) “SDO 配置” 选项卡

自动 SDO 配置：在 CANopen 组态中可以编辑 SDO 报文，下载至主站后，在每次初始化时完成固有的初始化流程后，紧接着就会执行自动 SDO。

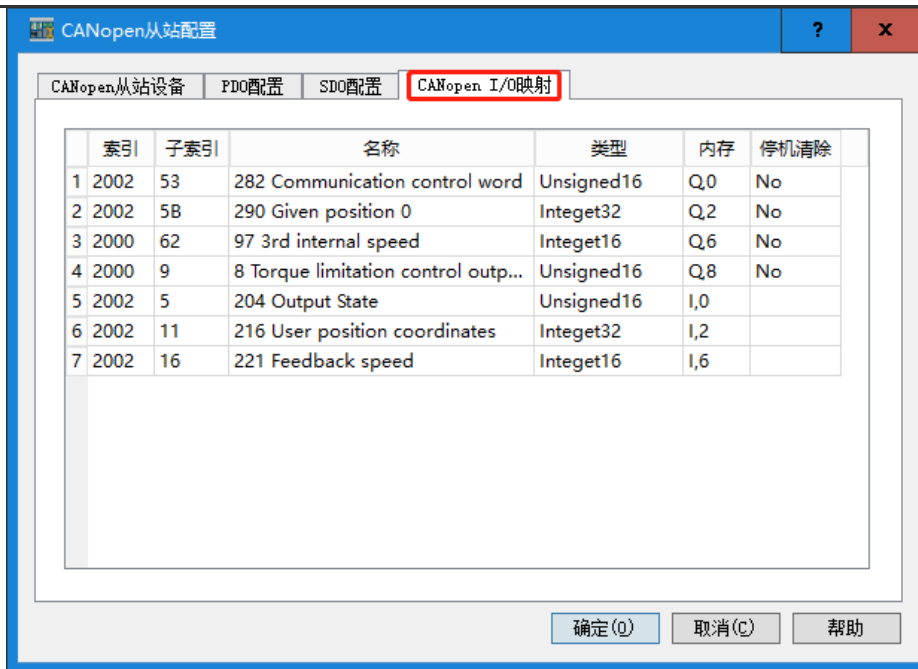
在“CANopen 从站配置”对话框的“SDO 配置”选项卡中可以配置 SDO，操作如下：



点击“添加”在弹出的界面中选中要添加的 SDO，点击“确定”即可添加一条 SDO，点击“编辑”按钮可以对选中的自动 SDO 进行修改，点击“移除”按钮可以删除配置的 SDO。自动 SDO 只能写参数，不能读参数，自动 SDO 只在从站由预运行状态进入运行状态前对从站写一次。此功能常用来初始化从站参数。

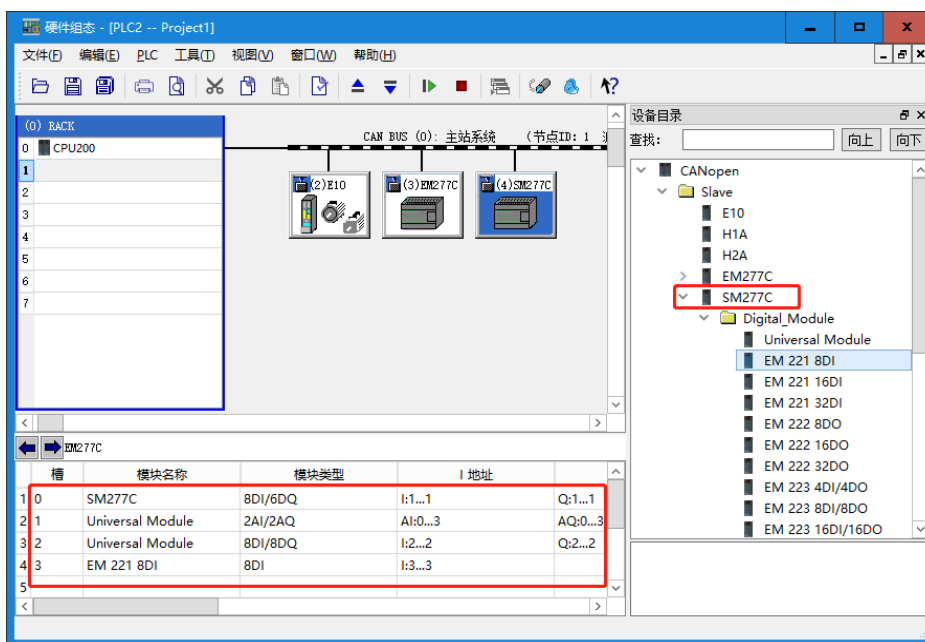
(4) “CANopen I/O 映射” 选项卡

“CANopen 从站配置”对话框的“CANopen I/O 映射”选项卡中将显示已经配置好的 IO/参数及映射的内存。



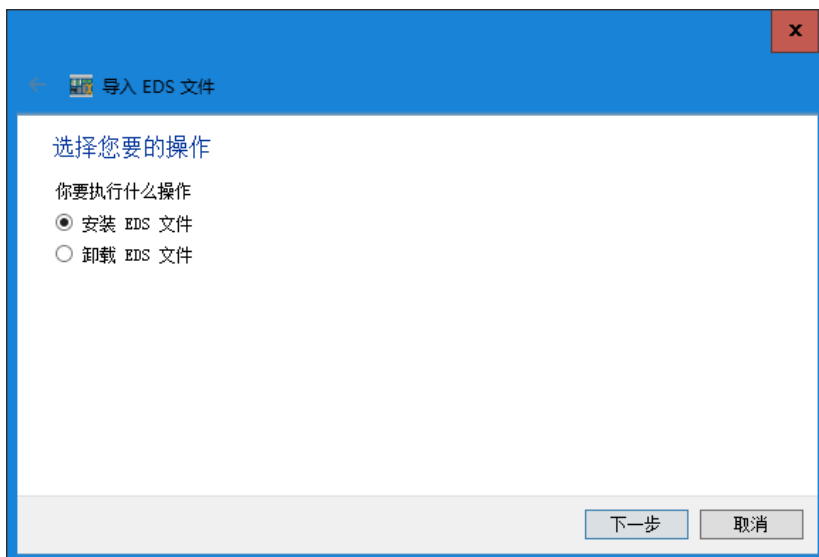
● 组态模块类从站

展开设备目录树，打开所选从站 EM277C 或 SM277C 节点下的数字量模块或模拟量模块节点，选择与你当前实际设备相符的模块型号，使用鼠标双击或拖拽进入模块插槽放下，模块即被添加到当前从站下。您也可以通过双击设备目录树上的模块节点将模块按顺序插入到当前从站下。如果您在模块列表找不到对应的模块型号，您可以选择 Universal Module，它允许用户自己定义模块的 IO。

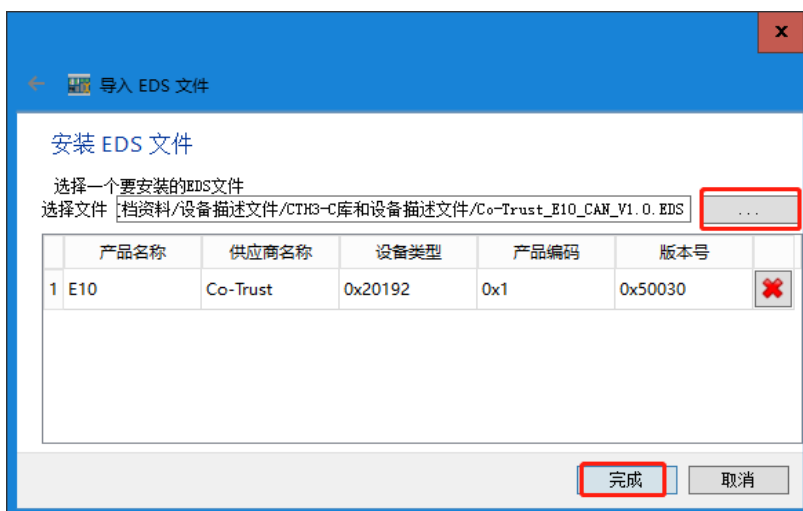


6.1.4 导入第三方 EDS 文件

安装 CANopen 从站设备的 EDS 文件：选择菜单项“工具”→“导入 EDS 文件 (E)” 打开如下对话框：

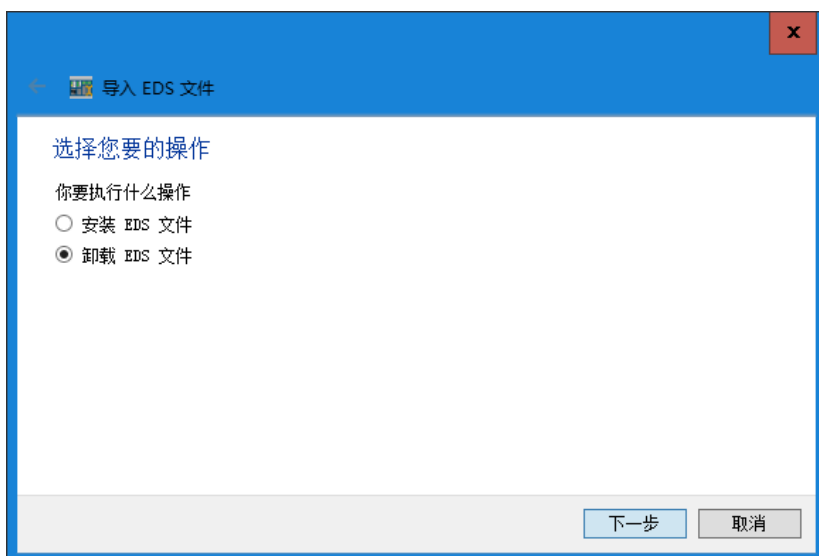


在以上窗口中选择“安装 EDS 文件”，然后点击“下一步”：



点击“...”选择需要安装的 EDS 文件，然后点击“添加”按钮，即可完成 EDS 文件安装。

卸载 CANopen 从站设备的 EDS 文件：选择菜单项“工具”→“导入 EDS 文件（E）”，打开如下对话框，在该对话框中选中“卸载 EDS 文件”，然后点击“下一步”：



勾选需要删除的 EDS 文件，如下图所示，然后点击“完成”按钮即可成功删除该 EDS 文件。



6.1.5 诊断

可以通过 SM 277C / E10 的通讯指示灯状态和 MagicWorks PLC 的 SMB 状态字来诊断 CANopen 主从站网络。

通过硬件进行诊断

SM277C 的 LED 灯状态

接通 SM 277C 的电源之后，标示为“ON”（绿）的 POWER（电源）LED 指示灯将亮起。

如果 BF 和 SF LED 保持熄灭状态，则说明 SM 277C 正常运作。

如果 BF 或 SF LED 亮起或闪烁，则说明硬件组态或者接线有错误。如果主系统的运行无故障，则错误的原因可参考下表：

表 6-4 SM 277C 状态 LED 灯功能说明

LED 灯	亮	灭	闪烁	备注
ON	有电源	无电源	---	电源指示
SF	扩展 IO 模块有故障	扩展 IO 模块无故障	---	系统故障
BF	没检测到 CAN 网络	检测到 CAN 网络	组态配置不一致	总线故障

E10 伺服面板查看状态

在 E10 伺服面板上选择参数菜单，调至参数 P236，点击确认按钮，即可看到 P236 的当前值，参数值对应状态参见表 6-5。

通过软件进行诊断

MagicWorks tuner 软件状态诊断

在 MagicWorks tuner 软件里写入 236 号参数，则当前值即显示 CANopen 状态机状态。

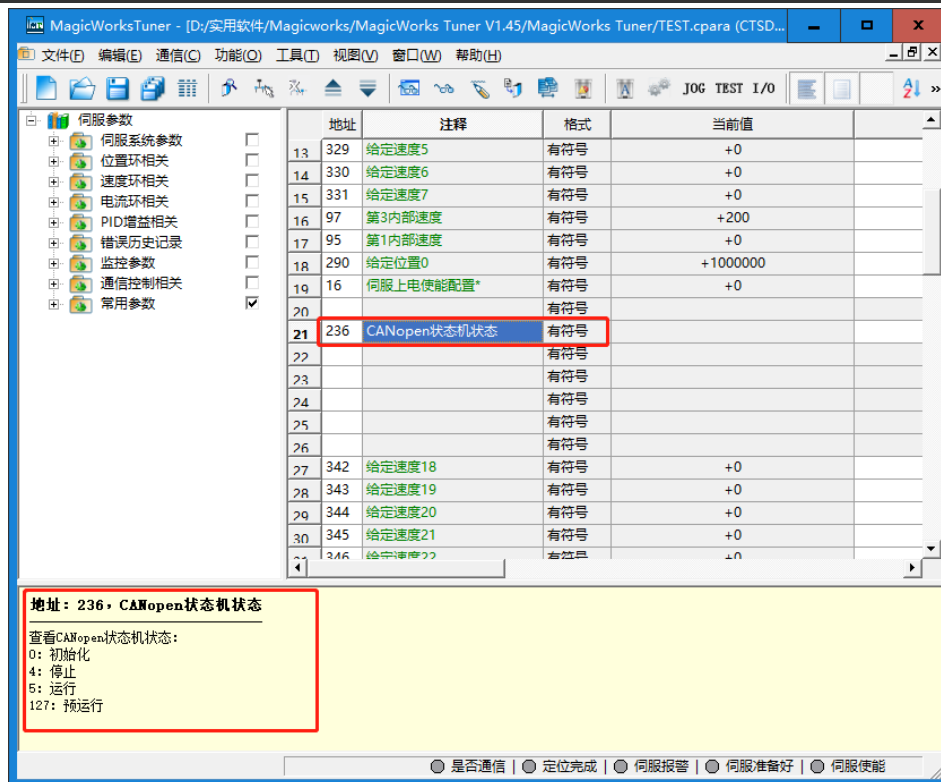


图 6-16 MagicWorks tuner 中查看 CANopen 状态机状态

参数值对应状态如下表所示：

表 6-5 CANopen 状态机状态

参数值	状态机状态
0	初始化
4	停止
5	运行
127	预运行

MagicWorks PLC 软件状态诊断

CTH200 系列 CPU 分配有 100 个字节的 CAN 专用存储区（SM），用户可通过查看状态字节来获取错误信息，CAN 站点状态字意义见下表：

表 6-6 CAN 站点状态表

	字节数	地址分配	状态值说明
SMB (状态)	100	SMB550: 主站 (CPU 本身) 的 CAN 通讯状态	0x00: 初始化 0x01: 断开连接 0x04: 停止 0x05: 运行 0x7f: 预运行 0xff: 组态数据出错
		SMB551-SMB582: 第 1~32 个从站 (按照 Node ID 从小到大的顺序排列) 的 CAN 通讯状态	0x00: 初始化 0x01: 断开连接 0x04: 停止 0x05: 运行 0x7f: 预运行 0x7: 组态参数出错
		SMB583-SMB614: 第 1~32 个从站 (按照 Node ID 从小到大的顺序排列) 的运行告警状态	bit7: 组态 IO 信息同 277C 实际情况的比较结果: 0: 无错误 1: 有错误 bit6-bit0: 277C 扩展 IO 模块的诊断新消息 bit0 代表第 1 个模块 bit6 代表第 7 个模块: 0: 无告警 1: 有告警 (以上为 277C 从站时的定义, 其他从站另行定义)
		SMB615-SMB649	保留



提示

SMB550-SMB649 为只读存储区, 用户不能写入。

6.2 Modbus TCP 通信

本节通过一个具体实例来展示实现 Modbus TCP 通信功能的详细操作步骤。

在 Modbus TCP 通信网络中, CPU 作为从站, 通讯独立于整个大循环周期, 即收即回; CPU 作为主站, 其收发由用户程序控制。

使用 CTH200 的 EtherNET 通信口进行 Modbus TCP 通信时, 无需任何配置, CPU 即可作为 Modbus TCP 从站实现 Modbus TCP 通信。MODBUS-TCP 通信默认端口号为 502, 如需更改 IP 地址, 请参见章节 2 使用入门。当 CTH200 作为 Modbus TCP 主站与其他从站进行通信时, 需要使用 Modbus TCP 向导或 Modbus TCP 单条读写指令库配置主站通信。



提示

请参考《MagicWorks PLC 用户使用手册》查看 TCP 读写指令向导相关使用说明。相关手册和库文件请访问合信官网, 网址: <http://www.co-trust.com>

6.2.1 示例组件

表 6-6 Modbus TCP 通信的示例组件

组件	功能
编程设备 PG\PC	安装有 MagicWorks PLC 软件（V2.08 以上版本），对 CTH200 可编程控制器进行组态、编程和调试。
CTH200 主控模块	本例使用两个 CTH200，一个作为 Modbus_TCP 主站，另一个作为 Modbus_TCP 从站，它们通过 EtherNET 接口进行 Modbus TCP 通讯。
标准网线	<ul style="list-style-type: none"> • 连接CTH200与编程设备 • 连接CTH200（Modbus_TCP主站）与CTH200（Modbus_TCP从站）

6.2.2 网络连接

使用标准网线连接编程设备与 CTH200，然后在编程设备上使用 COTRUST 提供的 CT_Modbus_TCP 库对 Modbus 主站（CTH200）进行编程：

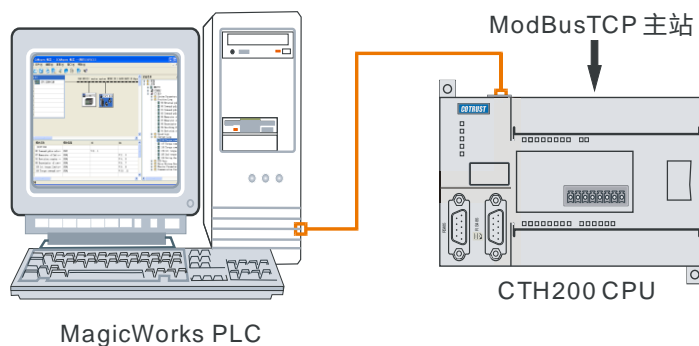


图 6-17 Modbus TCP 通信连接

使用标准网线连接 Modbus_TCP 主站和 Modbus_TCP 从站，Modbus_TCP 主站将进行一个写操作，将指定地址中的数据写入到 Modbus_TCP 从站中，然后，通过 Modbus_TCP 读取从站数据，继而实现 Modbus TCP 通信：

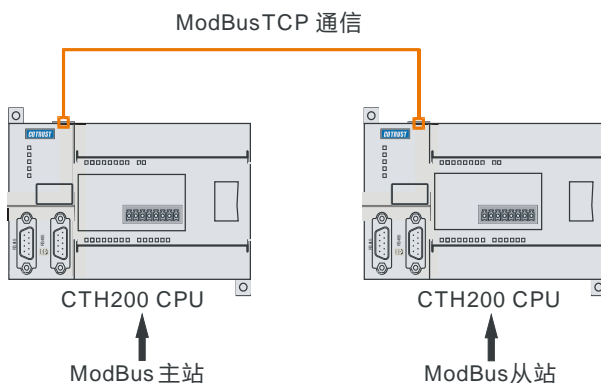


图 6-18 主站与主站通信



提示

CPU H224/H226L/H226M/H228XL 不支持自适应交叉，因此，PLC 直连进行 ModBus_TCP 通信时需所选标准网线应采用交叉线。

6.2.3 操作步骤

步骤 1：连接电缆

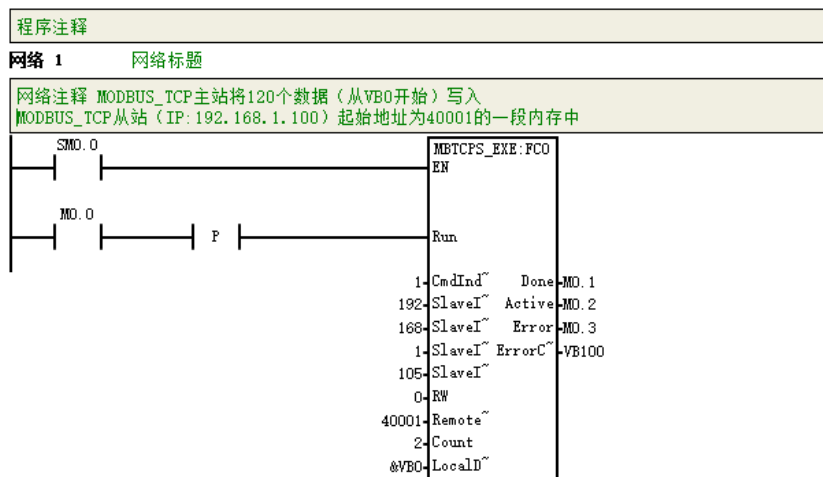
使用一条标准网线参考上图连接 PC 与 Modbus_TCP 主站。

步骤 2：设置通信

在 MagicWorks PLC 中新建一个工程，在该工程中添加 CTH200 站点，然后参考 2.2 与 CTH200 建立通信将 CTH200 与 PC 进行通信连接。

步骤 3：对 Modbus 主站（CTH200）进行编程

1) 在 MagicWorks PLC 的程序块中为 Modbus_TCP 主站编程（可以通过 Modbus TCP 库或 Modbus TCP 向导配置通信，本例使用 Modbus TCP 库进行编程），以下指令为“Ct_Mbus_master_tcp_single”库文件中的“MBTCPS_EXE”库指令：



<备注> 如果您想配置多项读写操作，建议使用 MagicWorks PLC 的 Modbus TCP 向导。具体操作请访问合信官网参见《MagicWorks PLC 用户手册》。

2) 编程完成后，进行编译并将程序下载到 Modbus_TCP 主站设备中。

下表为“MBTCPS_EXE”库指令定义：

符号	变量类型	数据类型	注释
RUN	IN	BOOL	通信启动位，请使用沿触发
CmdIndex	IN	BYTE	调用 MBTCPS_EXE 的序号，每处调用序号都不能重复，范围 1~255
SlaveIP0	IN	BYTE	从站地址的第一个字节，例如，从站地址为 192.168.1.202，第一个字节为 192
SlaveIP1	IN	BYTE	从站地址的第二个字节
SlaveIP2	IN	BYTE	从站地址的第三个字节
SlaveIP3	IN	BYTE	从站地址的第四个字节
RW	IN	BYTE	读写标志，读 = 0；写 = 1
RemoteAddr	IN	DWORD	远程数据地址（如 40001）
Count	IN	WORD	读写单元数，有效范围：1~120 个字或 1-1920 位
LocalDataPtr	IN	DWORD	本地数据指针（如&VB0）
Done	OUT	BOOL	完成位，0=还未完成或未启动；1=操作已完成
Active	OUT	BOOL	激活位，0=未激活或已完成；1=操作已激活且未完成

Error	OUT	BOOL	错误位，0=无错误，1=有错误
ErrorCode	OUT	BYTE	错误代码，完成位为1时有效，详情如下表

ErrorCode 输出端的错误代码定义如下：

错误代码	说明
0	无错误
1	连接数达到最大值
2	正在建立连接
3	接收超时，从站无响应
4	请求的参数有错误
5	指令没有使能
6	连接正在忙（如同时激活多条 MBTCPS_EXE）

步骤 4：Modbus TCP 主站与 Modbus TCP 从站互通通信

组态 Modbus TCP 从站时，从站 IP 必须对应于主站编程设定的 IP，如需更改从站 IP，请打开 MagicWorks PLC 中的系统块，修改其中的 IP 地址，如下图所示。另外，PLC 之间进行 ModBus TCP 通信时，从站端口号应为 1024 以上，此为系统默认规定。

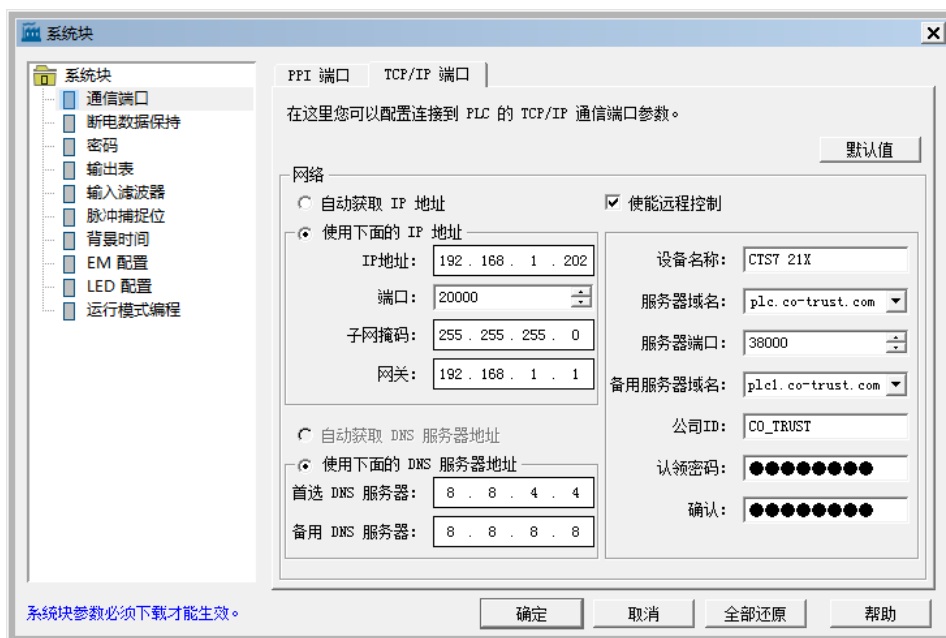


图 6-19 系统块设置

步骤 5：调试

按顺序执行以上步骤后，使用标准网线连接 Modbus 主站与从站，然后通过 MagicWorks PLC 软件的状态表读取 Modbus_TCP 从站以 40001 起始的 120 个数据，若该段内容与 Modbus_TCP 主站中对应内存中的数据一致，则表示 Modbus TCP 通信成功。

6.2.4 ModBus TCP 从站地址映射

ModBus 地址通常是包含数据类型和偏移量的 5 个或 6 个字符值。前一个或两个字符决定数据类型，后几个字符是一个符合当前数据类型的相应值。



提示

Modbus TCP 无法映射 M 内存区。

ModBus-TCP 从站支持以下地址：

TCP 协议对应的从站地址映射

ModBus 从站地址	CTH200 地址
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 通信

本节将通过一个具体实例展示实现 UDP_PPI 通信（EtherNET 通信口）的详细操作步骤。

在 UDP_PPI 通信网络中，CPU 作为从站，通讯独立于整个大循环周期，即收即回；CPU 作为主站，由收发由用户程序控制。

CTH200 可以作为主站或从站，使用网络读写指令 UDP_NETR/UDP_NETW 或 NETW/NETR 指令向导与局域网内的其他设备进行 UDP_PPI 通信。同时，也可以 HMI 作为 UDP_PPI 主站，以 CTH200 作为从站进行 UDP_PPI 通信。



提示

请参考《MagicWorks PLC 用户手册》查看 UDP 读写指令向导相关使用说明。相关手册和库文件请访问合信官网，网址：<http://www.co-trust.com>

6.3.1 示例组件

表 6-6 UDP_PPI 通信的示例组件

组件	功能
编程设备 PG\PC	安装有 MagicWorks PLC 软件（V2.08 以上版本），对 CTH200 可编程控制器进行组态、编程和调试

CTH200 主控模块	本例使用两个 CTH200, 一个作为 UDP_PPI 主站, 另一个作为 UDP_PPI 从站, 它们通过 EtherNET 接口进行 UDP_PPI 通讯
标准网线	<ul style="list-style-type: none"> • 连接 CTH200 与编程设备 • 连接 CTH200 (UDP_PPI 主站) 与 CTH200 (UDP_PPI 从站)

6.3.2 网络连接

使用标准网线连接编程设备与 CTH200, 通过编程设备对 UDP_PPI 主站 (CTH200) 进行编程:

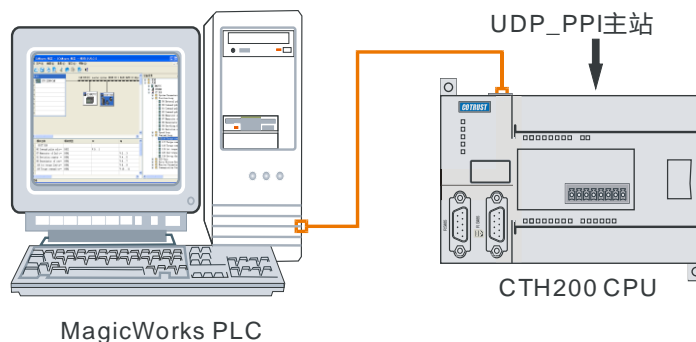


图 6-20 UDP_PPI 通信连接

使用标准网线连接 UDP_PPI 主站和 UDP_PPI 从站, UDP_PPI 主站将进行一个写操作, 将指定地址中的数据写入到 UDP_PPI 从站中, 然后通过 UDP_PPI 读取从站数据, 继而实现 UDP_PPI 通信:

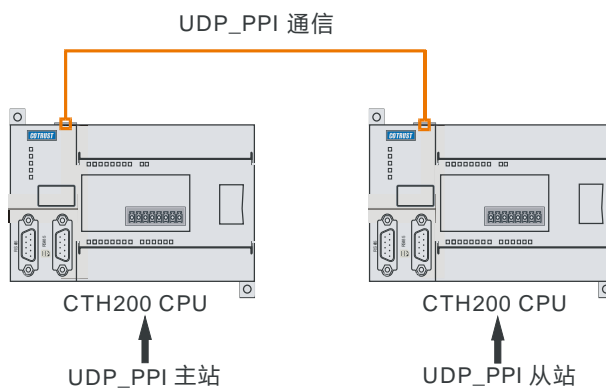


图 6-21 主站间通信



提示

CPU H224/H226L/H226M/H228XL 不支持自适应交叉, 因此, PLC 直连进行 UDP_PPI 通信时需所选标准网线应采用交叉线。

6.3.3 操作步骤

步骤 1: 连接电缆

使用一条标准网线参考上图连接 PC 与 UDP_PPI 主站(CTH200)。

步骤 2: 设置通信

在 MagicWorks PLC 中新建一个工程, 在该工程中添加 CTH200 站点, 然后参考 [2.2 与 CTH200 建立通信](#) 将 CTH200 与 PC 进行通信连接。

步骤 3: 对 UDP_PPI 主站 (CTH200) 进行编程

以 UDP 网络读写为例，您可以通过以下两种方法之一对 UDP_PPI 主站进行网络读写操作。

1) 通过 UDP 网络读写指令 UDP_NETR/UDP_NETW 为 UDP_PPI 主站编程

UDP_NETR/UDP_NETW 指令的 TABLE 参数表：

D	A	E	0	错误代码	0
IP 地址第 1 个字节					1
IP 地址第 1 个字节					2
IP 地址第 1 个字节					3
IP 地址第 1 个字节					4
端口号高字节					5
端口号低字节					6
指向远程站指针<I, Q, M, V, DB> (共 4 个字节) 第一个字节					7
指向远程站指针第 2 个字节					8
指向远程站指针第 3 个字节					9
指向远程站指针第 4 个字节					10
数据长度					11
数据字节 0					12
数据字节 1					13
...					...
数据字节 199					211

D: 完成 (功能完成), 0=未完成、1=完成

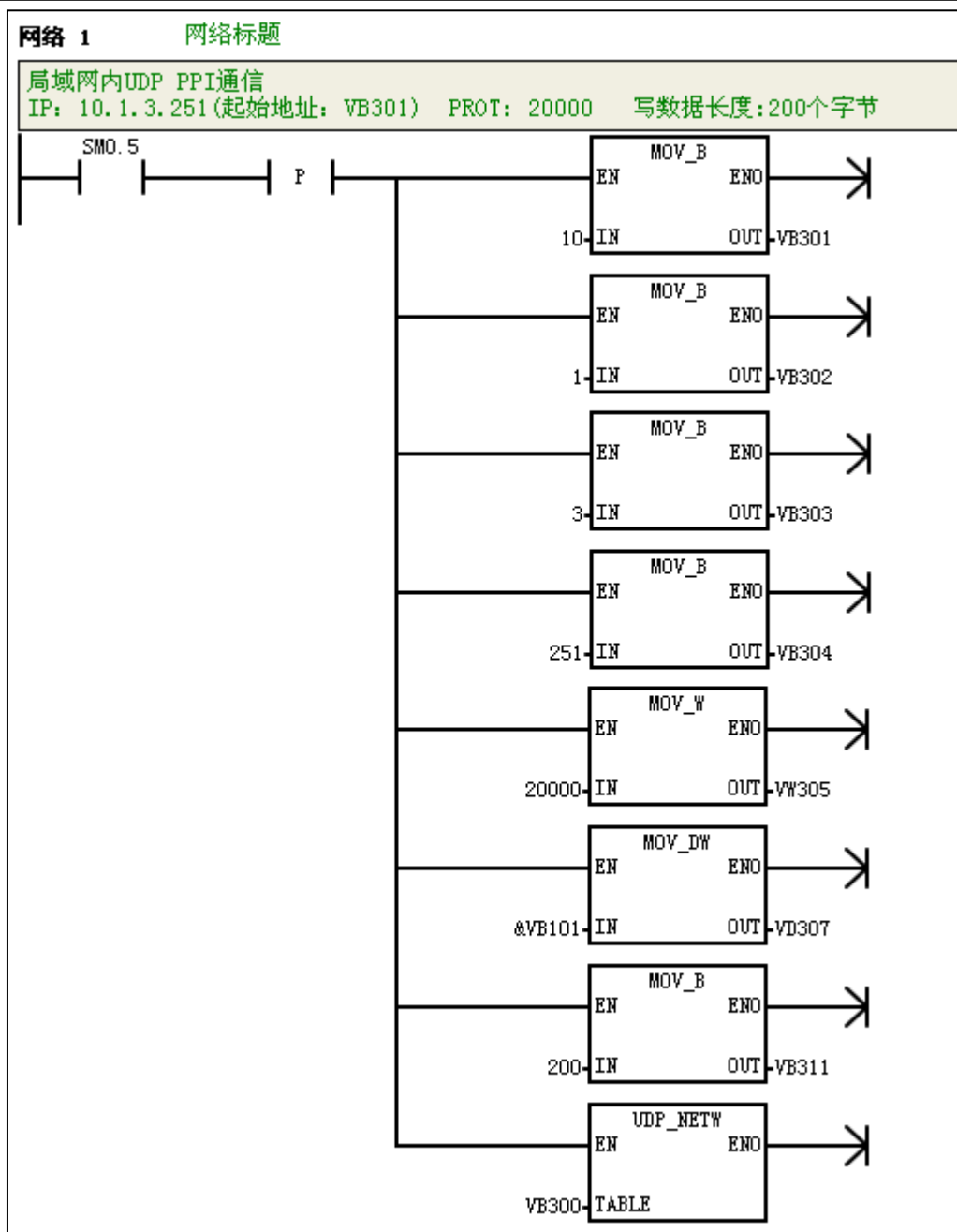
A: 现用 (功能入队), 0=非现用、1=现用

E: 错误, 0=无错、1=错误

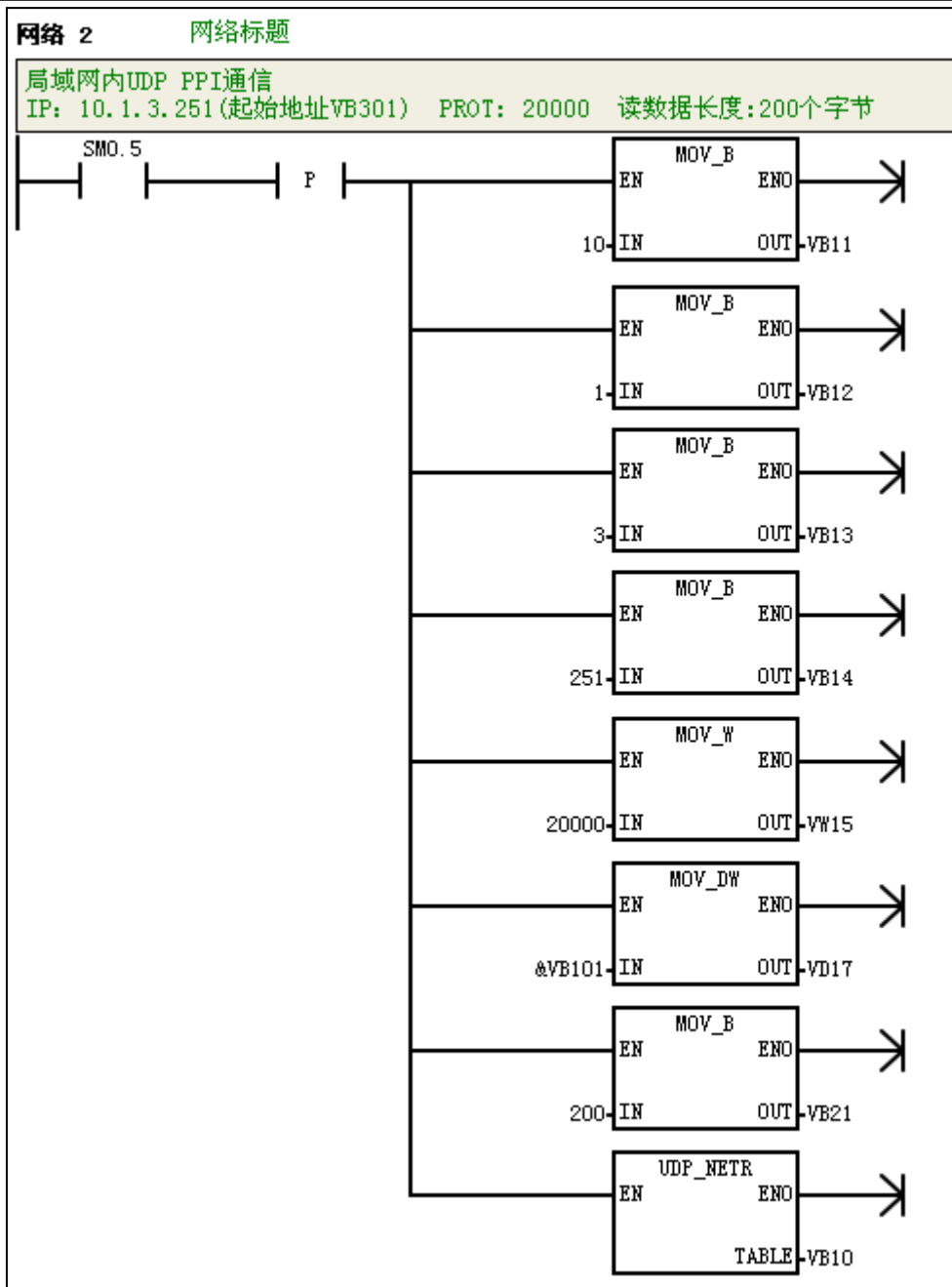
第一个字节的低四位为错误代码, 定义如下:

错误代码	说明
0	无错
1	超时错误; 远程站不应答
2	接收错误; 应答中存在校验、帧或校验和错误
3	脱机错误; 重复站址或故障硬件引起的冲突
4	队列溢出错误; 8 个以上 UDP_NETR/UDP_NETW 方框被激活
6	非法参数; UDP_NETR/UDP_NETW 表格包含一个非法或无效数值
7	无资源; 远程站繁忙 (正在上载或下载序列)
8	第 7 层错误; 违反应用程序协议
9	信息错误; 数据地址错误或数据长度不正确

网络 1: 将 UDP_PPI 主站 (起始地址: VB312) 中的 200 个字节数据写入到 UDP_PPI 从站 (IP: 10.1.3.251、起始地址: VB101) 中。

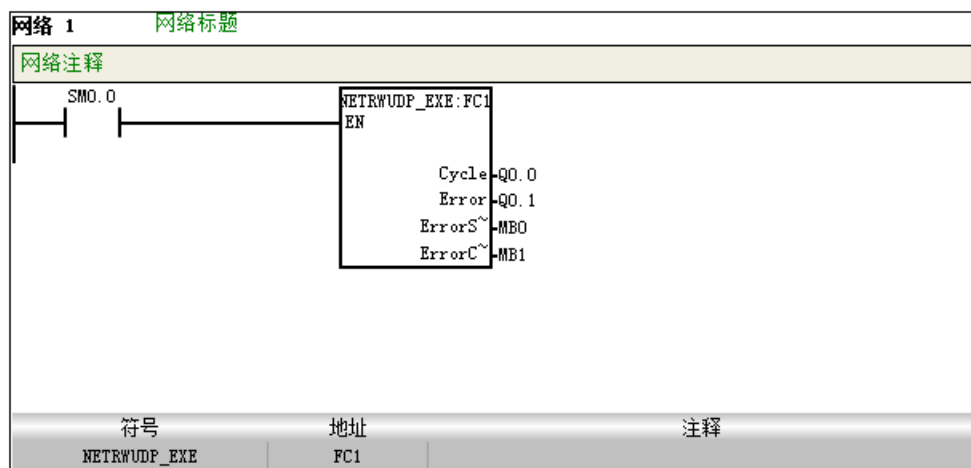


网络 2: 将 UDP_PPI 从站 (IP: 10.1.3.251、起始地址: VB101) 中的 200 个字节数据读入到 UDP_PPI 主站 (起始地址: VB22) 中。



2) 通过 NETR/NETW-UDP 指令向导为 UDP_PPI 主站编程

在项目树中展开 CTH200，在子选项中选择“向导”，然后在右侧的工作区域双击“NETR/W-UDP 向导”打开向导对话框随后即可逐步配置 NETR/NETW-UDP 指令向导。配置完成后，在 OB1 的网络中调用该 FC，如下图所示：



3) 按照上述方法 1) 或 2) 完成程序编辑后, 将程序编译并下载到 UDP_PPI 主站设备中 (必须用 SM0.0 调用)。

步骤 4: UDP_PPI 主站与 UDP_PPI 从站互连通信

- 1) 给 UDP_PPI 主站、从站接通电源。
- 2) 使用标准网线连接 UDP_PPI 主站与从站的 EtherNET 通信口, 随后它们即可进行 UDP_PPI 通信。

6.3.4 UDP_PPI 通信从站地址映射

UDP_PPI 通信采用普通网络读写, 执行读写操作时可组态多种寄存器, 从站地址映射为直接映射方式。

6.4 西门子 S7 协议通信举例

合信 CTH2 系列高性能升级版型 PLC (H224X、H226XM 和 H226XL) 支持西门子 S7 协议从站功能, 西门子 PLC 或屏或支持 S7 协议的上位机软件作为主站, CTH2 系列高性能升级版型 PLC 作为从站, 即可进行 S7 协议通信, 最多支持 8 个连接 (不分主从), 其中合信 CPU 只做从站。

支持西门子 S7 协议的 CTH2 高性能升级版型 PLC 型号如下:

型号	订货号	是否支持西门子 S7 协议
H224X	CTH2-214-1AX35-0X24	是
	CTH2-214-1BX35-0X24	是
H226XM	CTH2-216-1AX35-0X24	是
	CTH2-216-1BX35-0X24	是
H226XL	CTH2-216-2AX35-0X40	是
	CTH2-216-2BX35-0X40	是

本节将通过具体实例来展示合信 CTH2 系列高性能升级版型 PLC 与西门子触摸屏/PLC 及第三方支持西门子 S7 协议的触摸屏与上位机通讯的参数配置操作。

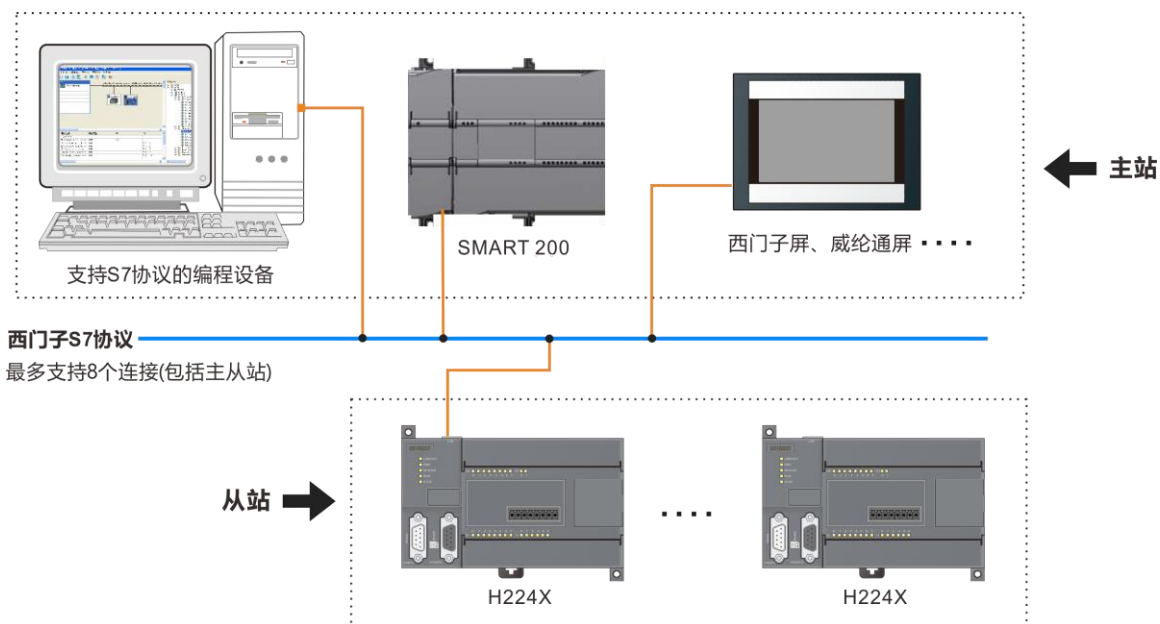
6.4.1 示例组件

表 6-7 示例组件

组件	功能
编程设备	安装有 STEP 7-MicroWin SMART 软件，对 CTH2 系列高性能升级版 PLC 进行组态、编程和调试。
CTH2 系列高性能升级版 PLC	H224X/H226XM/H226XL 作为 S7 协议从站，与 S7 协议主站进行通信
西门子 SMART200 PLC	作为 S7 协议主站，与 S7 协议从站站进行通信
组态王	软件版本 V6.55，作为 S7 协议主站，与 S7 协议从站站进行通信
昆仑通态触摸屏	作为 S7 协议主站，与 S7 协议从站站进行通信
西门子触摸屏	SMART 700 IE V3，作为 S7 协议主站，与 S7 协议从站站进行通信
威纶通触摸屏	作为 S7 协议主站，与 S7 协议从站站进行通信

6.4.2 网络连接

如下为 CTH2 系列高性能升级版 H224X 配合西门子等设备或上位机软件进行 S7 协议通信的典型连接：



6.4.3 西门子产品与 CTH2 系列高性能升级版 PLC 的网口通讯

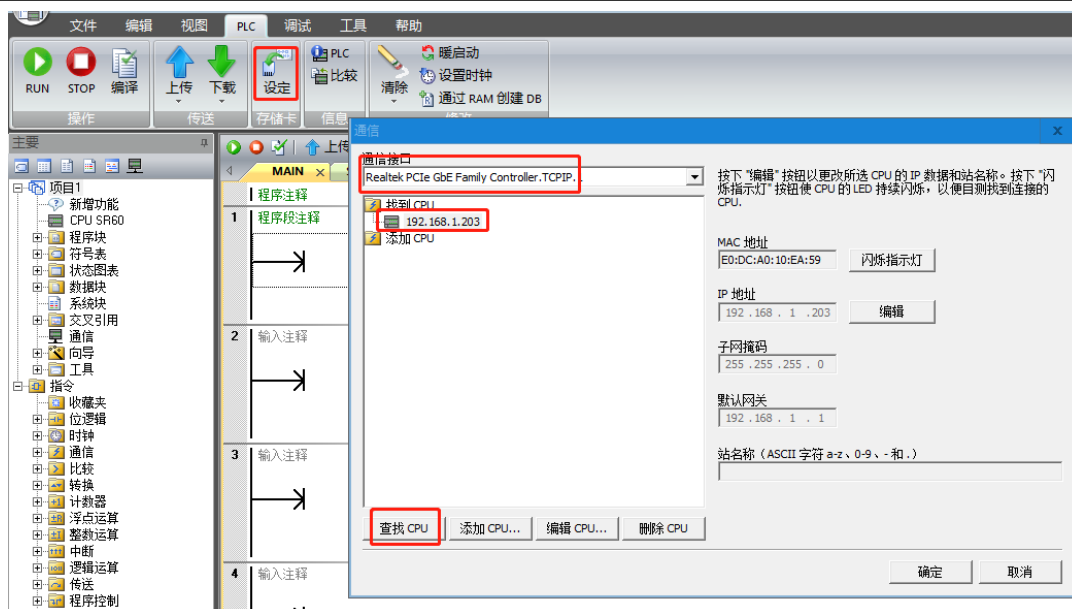
本例以西门子 SMART200 CPU、S7-1200/1500 CPU、触摸屏为主站，CTH2 系列高性能升级版 PLC（H224X、H226XM 和 H226XL）做从站，进行 S7 协议的通信。

示例 1：SMART200 与 CTH2 系列高性能升级版 PLC 的 S7 协议通信

备注： 西门子触摸屏也可以作为主站与 CTH2 系列高性能升级版 PLC 进行 S7 协议的通信。

1、SMART200 PLC 与 STEP 7-MicroWin SMART 编程软件通信设置

打开 STEP 7-MicroWin SMART 软件，将 SMART200 PLC 与电脑用 RJ45 网线连接（电脑与 PLC 保持在同一个网段内），打开 SMART 软件设置界面，选择电脑对应的本地网卡，然后点击查找 CPU 即可找到 PLC。

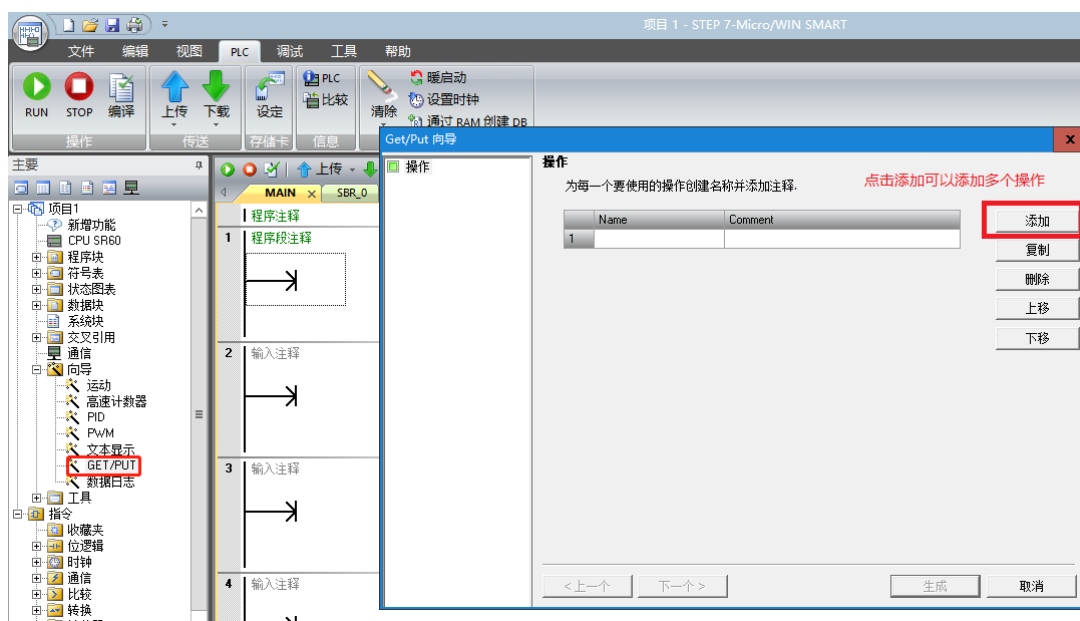


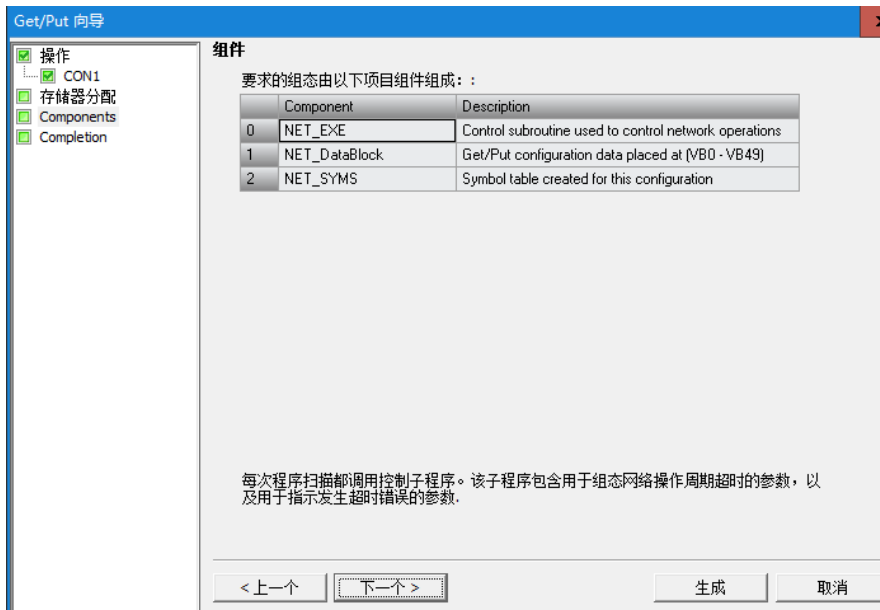
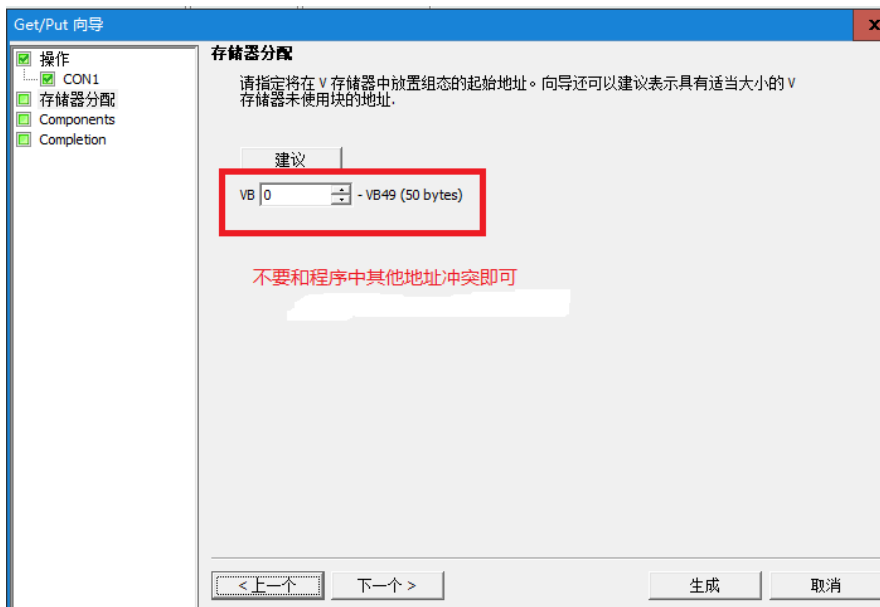
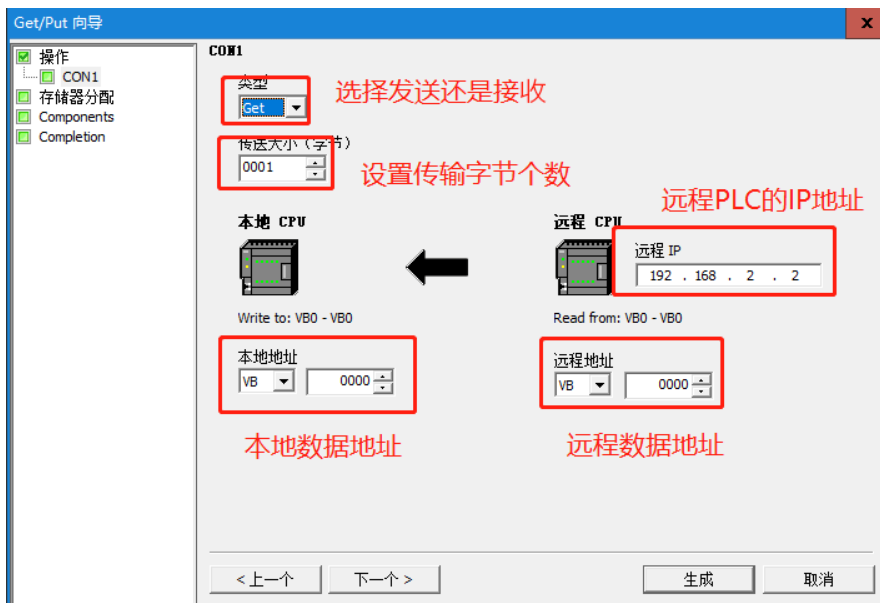
2、SMART200PLC 做主站，CTH2 系列高性能升级版型 PLC 做从站程序设置。

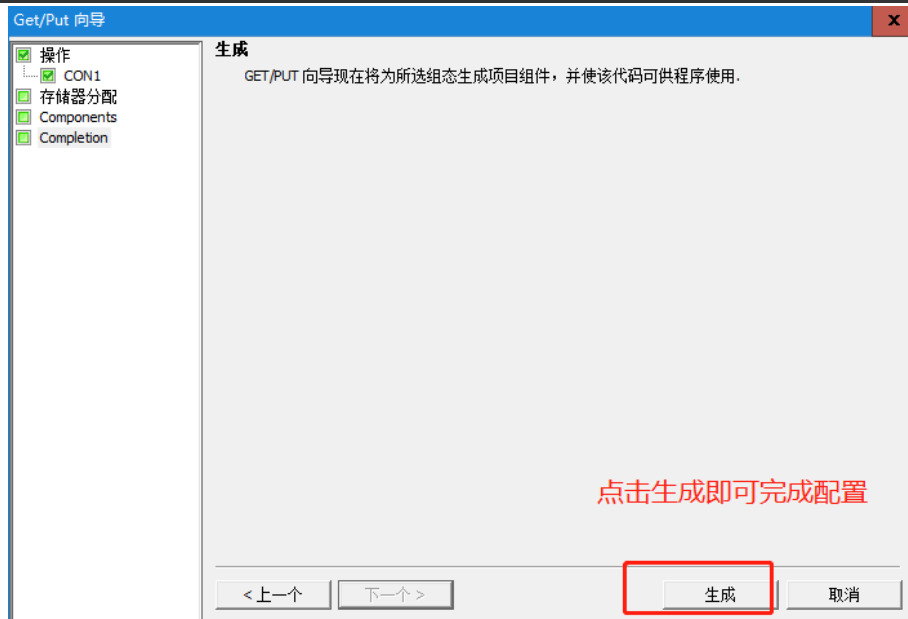
主站配置

1) 建立 GET/PUT 向导

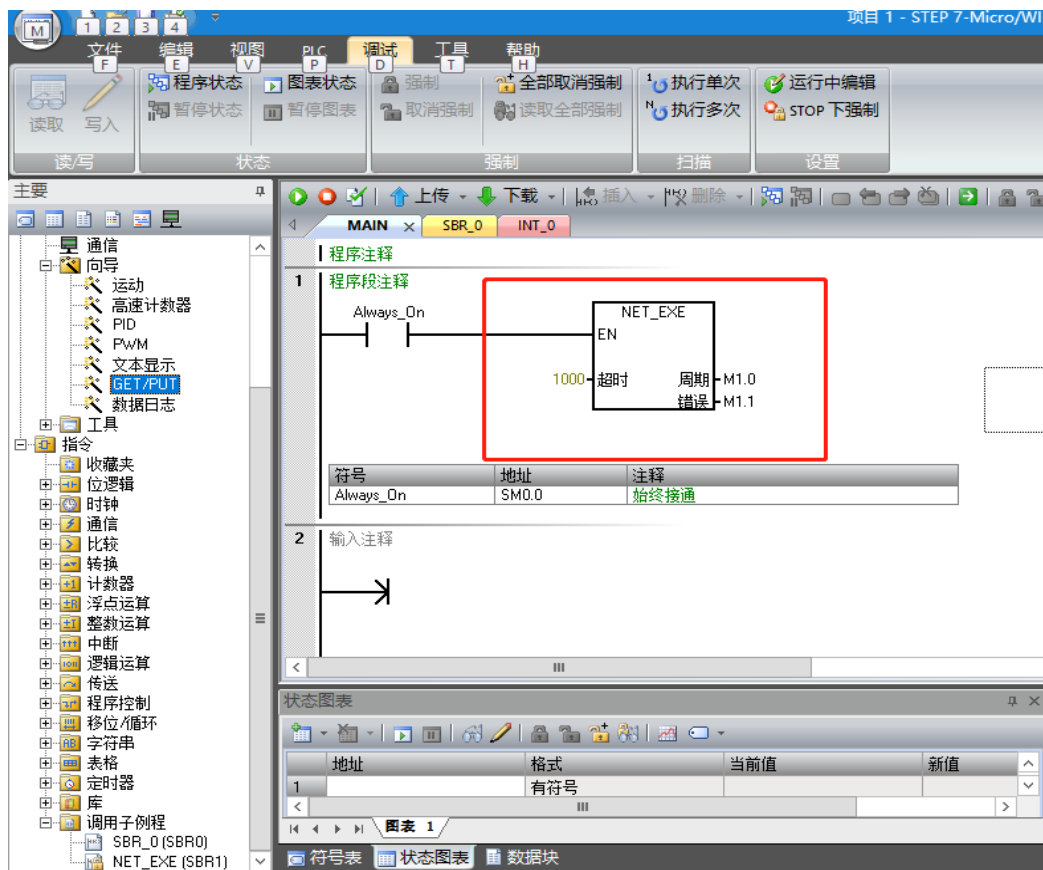
在 STEP 7-MicroWin SMART 软件中打开 GET/PUT 向导，并进行如下设置：

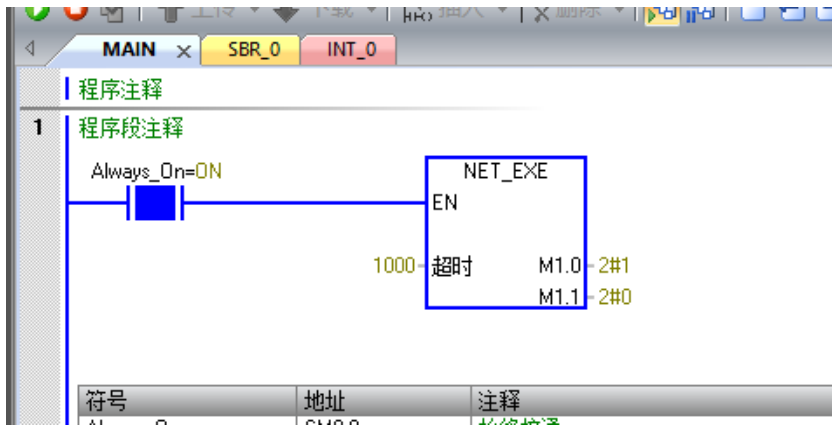






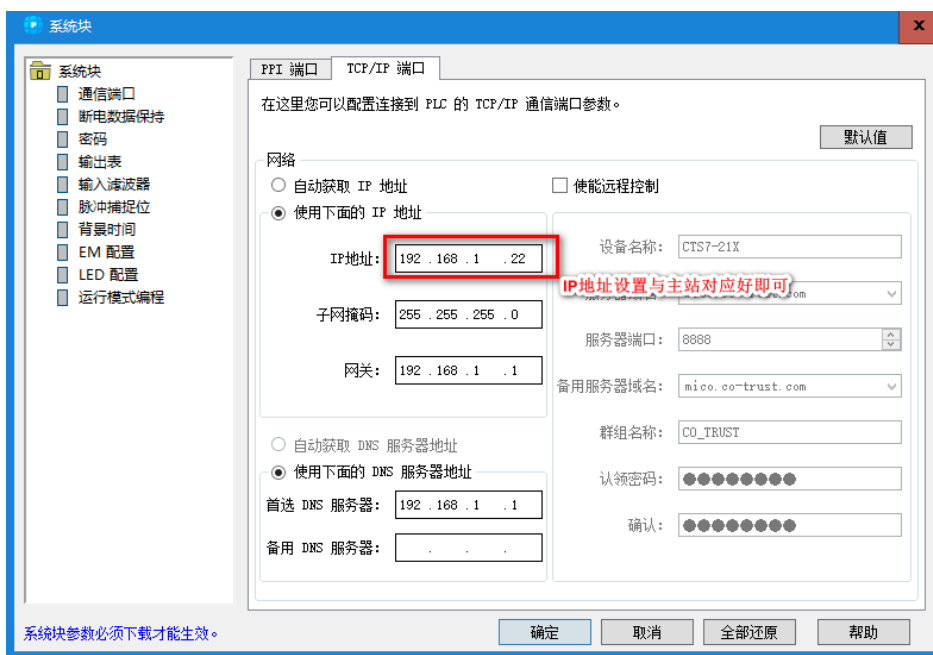
2) 在 SMART 软件的程序里面用 sm0.0 调用向导生成的块





从站配置

在 CTH2 系列高性能升级版型 PLC 的系统块中配置好 IP 地址即可，操作示意如下：

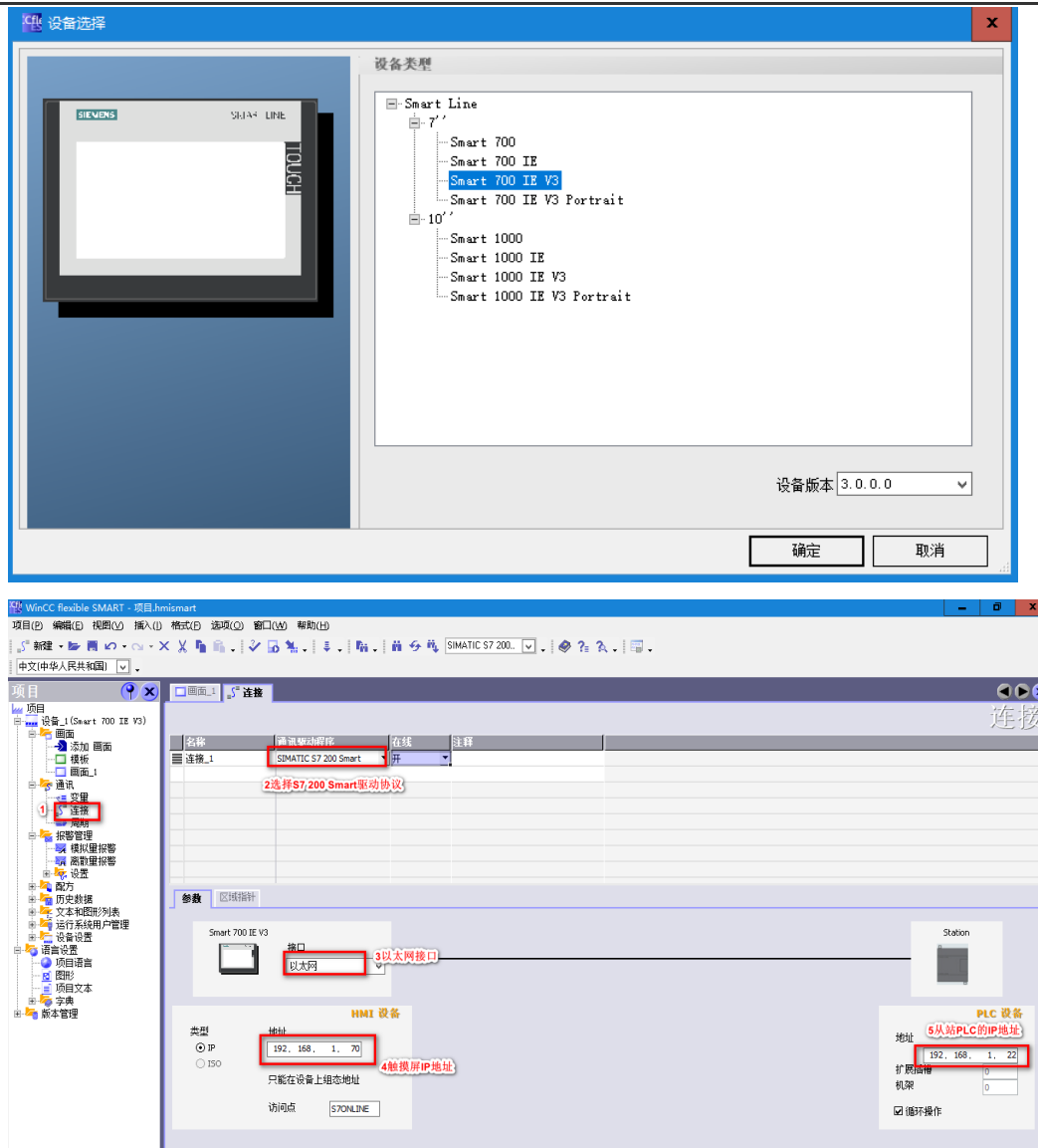


通过以上步骤将 S7 协议主从站设置完成后，即可实现西门子 SMART200 与 CTH2 系列 高性能升级版型 PLC 进行的 S7 协议通信。

示例 2：西门子触摸屏与 CTH2 系列高性能升级版型 PLC 的 S7 协议通信

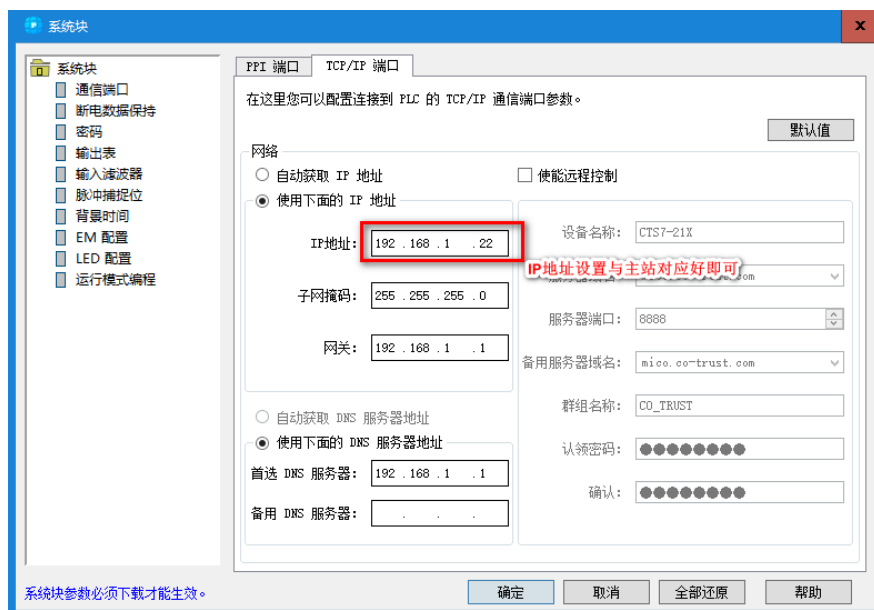
本例以西门子触摸屏为主站，CTH2 系列高性能升级版型 PLC 做从站，进行 S7 协议的通信。

主站配置



从站配置

在 CTH2 系列高性能升级版型 PLC 的系统块中配置好 IP 地址即可：



示例 3: S7-1200/1500 与 CTH2 系列高性能升级版型 PLC 的 S7 协议通信

S7-1200/1500 CPU 与 CTH2 系列高性能升级版型 PLC 的 S7 协议通信 (S7-1200/1500 作为主站)，合信 CTH2 系列高性能升级版型 PLC (H224X、H226XM 和 H226XL) 支持西门子 S7 协议从站功能，西门子 PLC 或屏或支持 S7 协议的上位机软件作为主站，CTH2 系列高性能升级版型 PLC 作为从站，即可进行 S7 协议通信，最多支持 8 个连接 (不分主从)，其中合信 CPU 只做从站。

硬件和软件需求及所完成的通信任务	
硬件	<ul style="list-style-type: none"> ● S7-1200 CPU 硬件版本 V2.0 或更高 ● H226XM (固件版本 2.46 以上) ● PC (带以太网卡) ● TP 以太网电缆 ● 工业交换机
软件	<ul style="list-style-type: none"> ● TIA portal V15 ● MagicWorks PLC V2.19

所完成的通信任务:

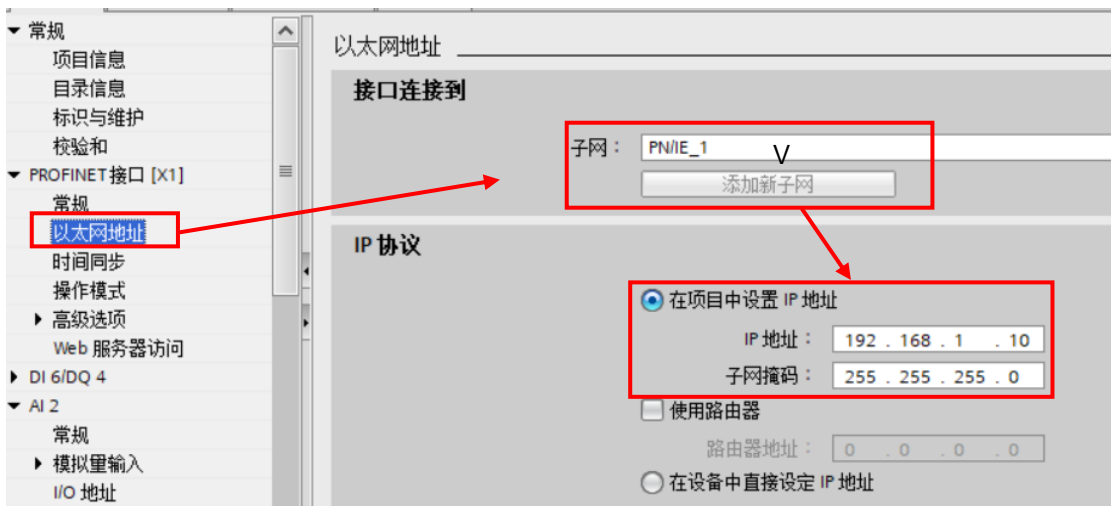
S7-1200 将通信数据区 DB1 中的 100 个字节发送到 H226XM 的 VB 数据区，S7-1200 读取 H226XM 中的 VB 数据区存储到 S7-1200 的数据区 DB1 数据区中。

S7-1200 M 区中的 MB0-MB9 连续 10 个字节发送到 H226XM 的 M 区中 MB0-MB9，S7-1200 读取 H226XM 区中的 MB10-MB19 连续 10 个字节到 S7-1200 的 M 区中 MB10-MB19。

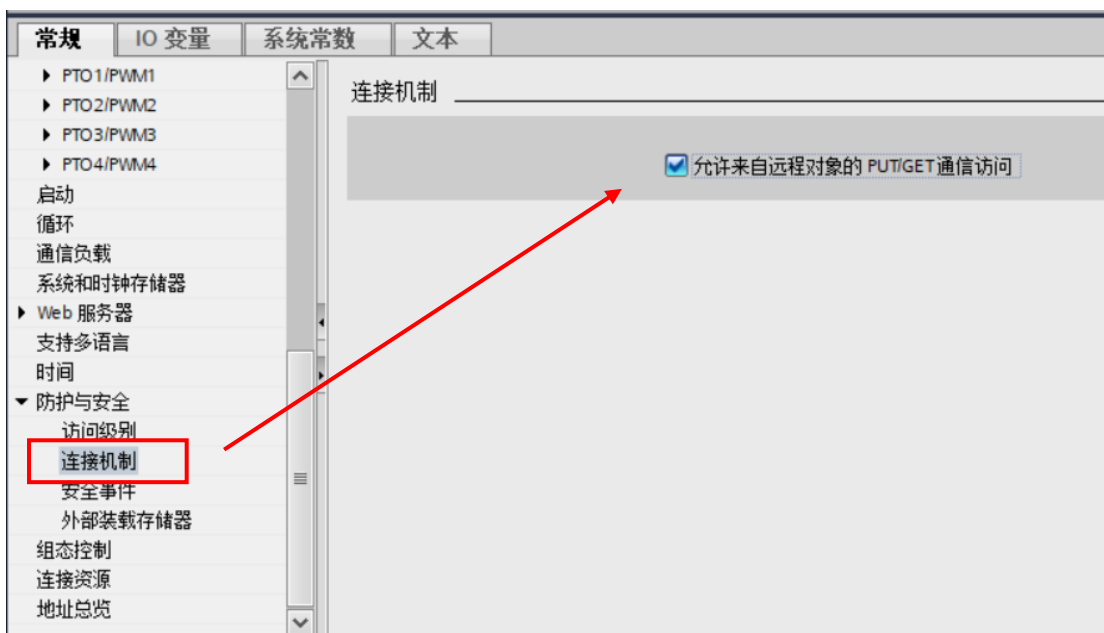
S7-1200 主站的配置编程

步骤 1: 使用 TIA portal V15 软件新建一个项目并完成硬件配置，网络组态

1) 新建子网 PN/IE_1，修改 IP 地址与 PC 和 H226XM 所在同一个网段

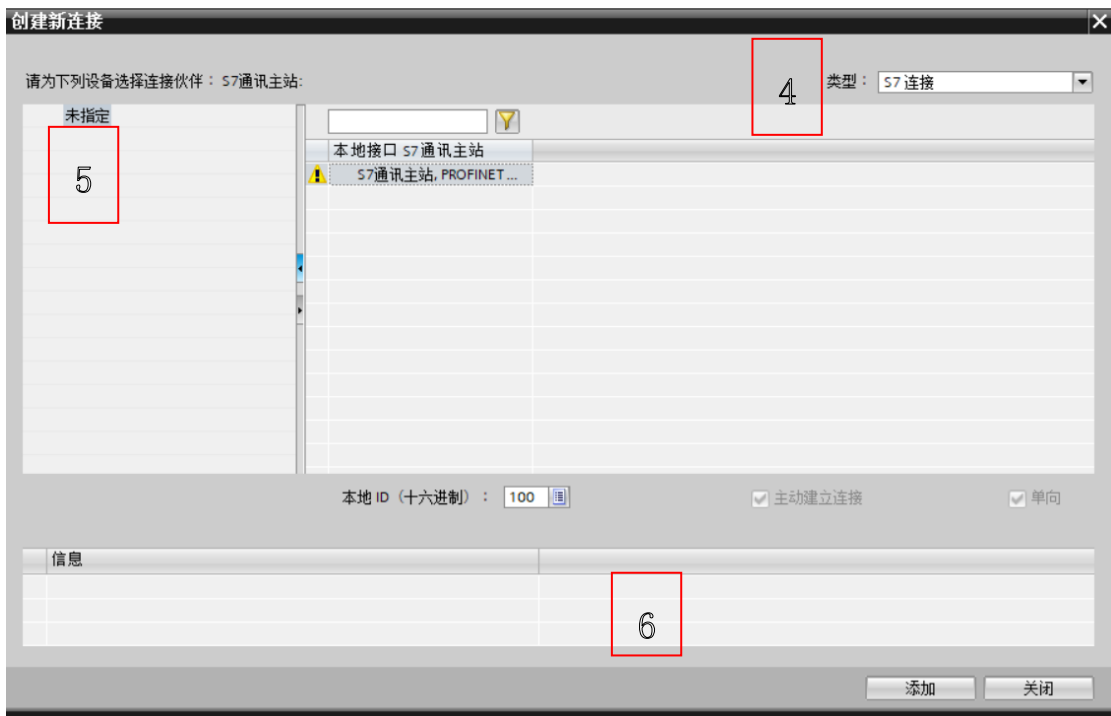


2) 将“防护与安全” > “连接机制” > “允许来自远程对象的 PUT/GET 通信访问”勾选

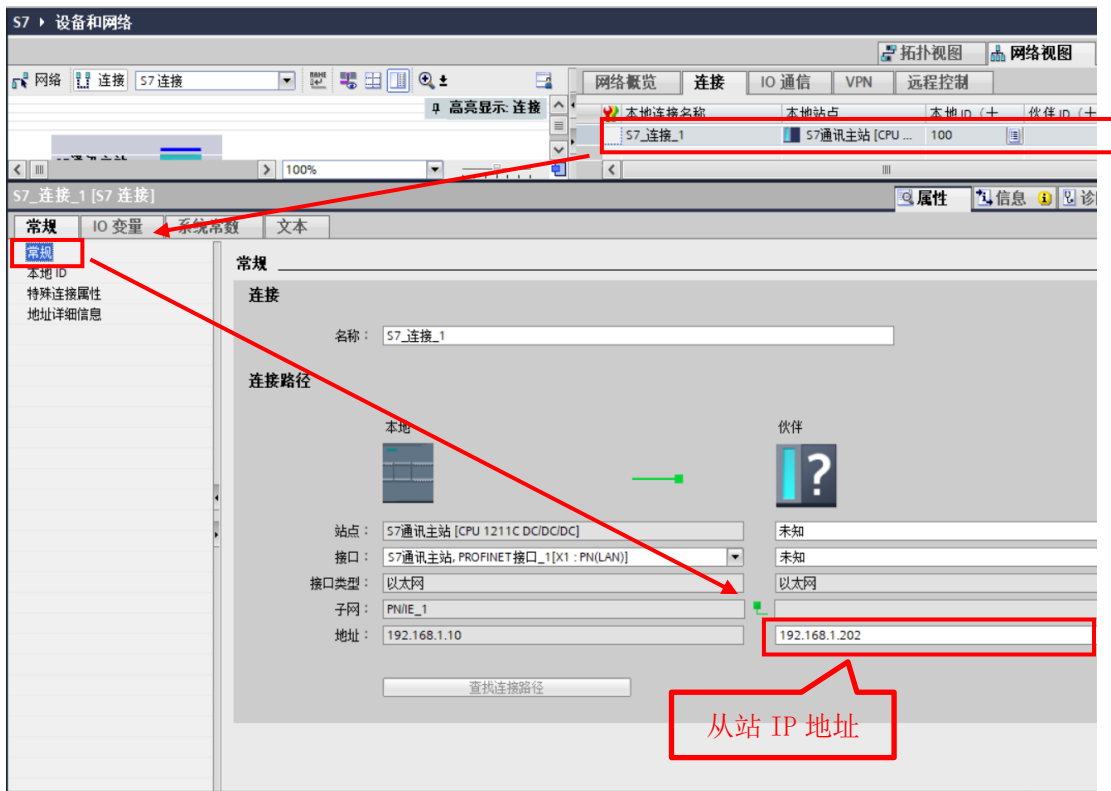


3) 在“项目树”>“设备组态”>“网络视图”下，按如下图片中 1~6 的步骤建立 S7 连接

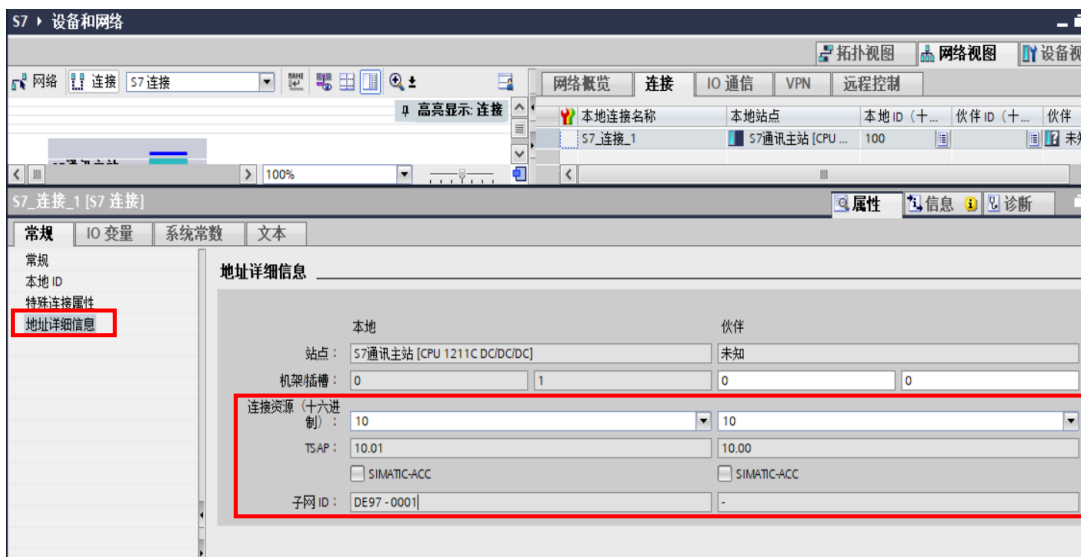




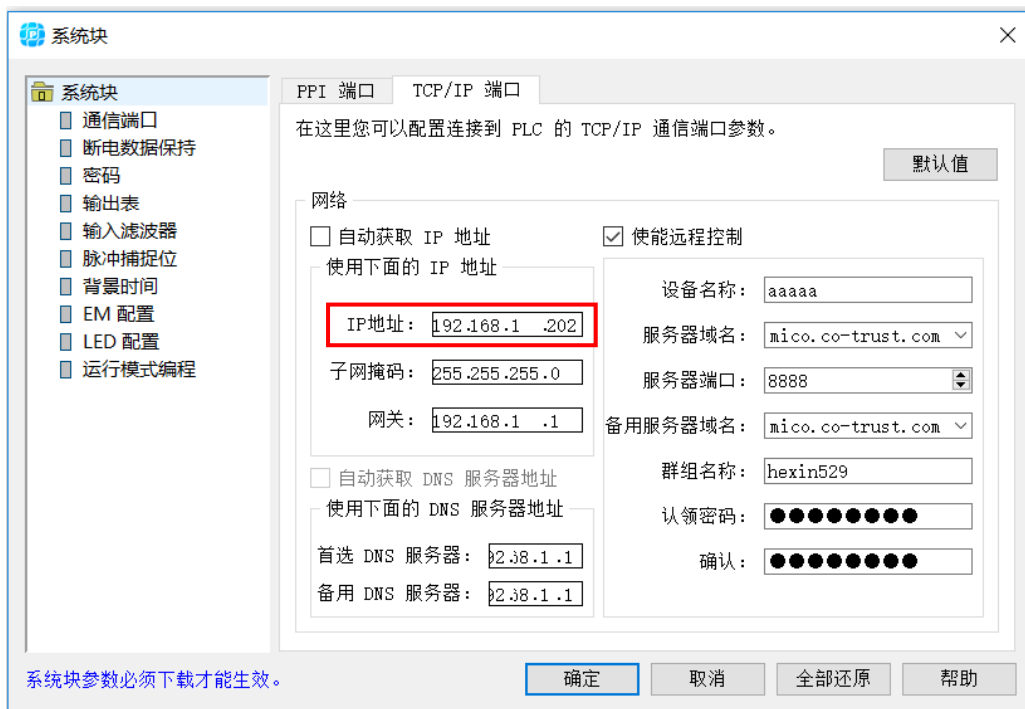
4) 按如下图片中的步骤填写连接参数



5) 在“地址详细信息”中设置通信伙伴的 TSAP 地址

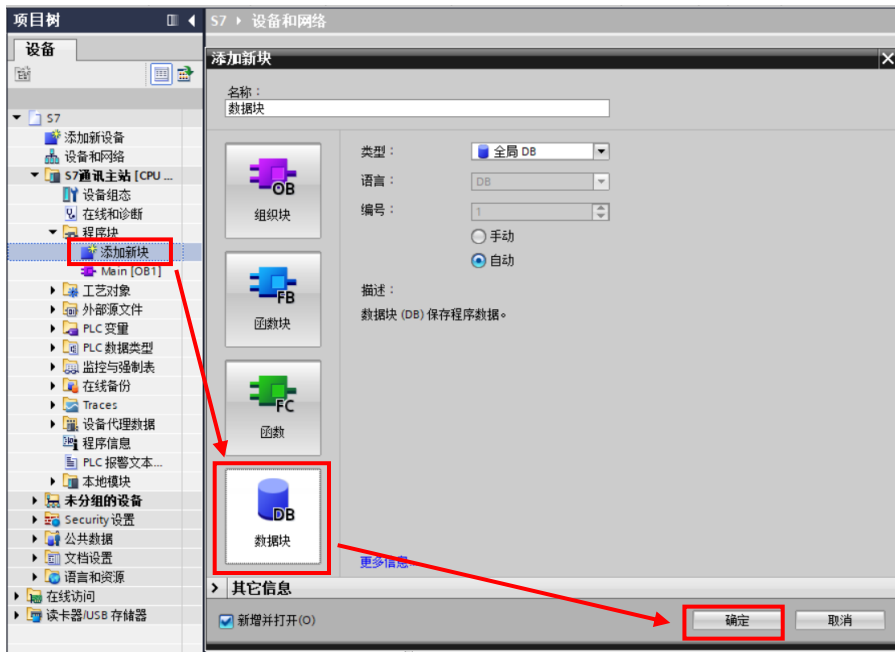


6) 在 CTH2 系列高性能升级版型 PLC 的系统块中配置好 IP 地址即可，操作示意如下

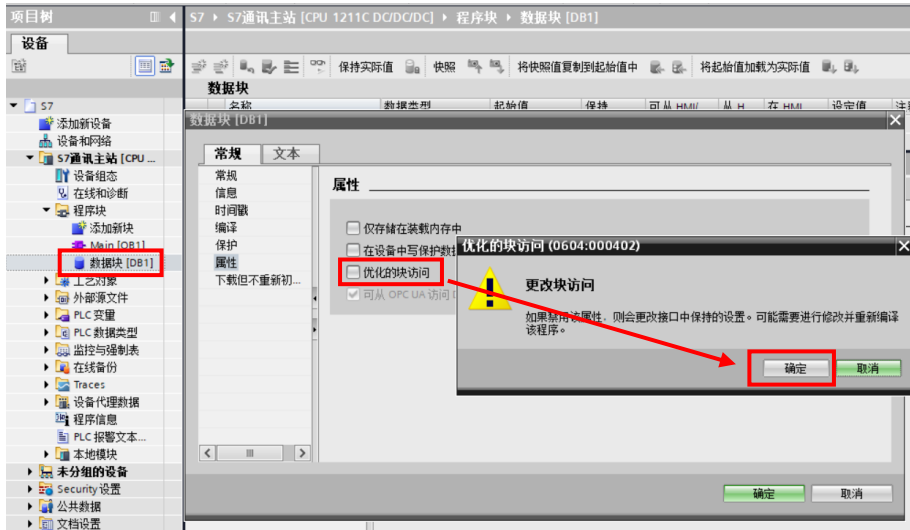


步骤 2：软件编程

1) 创建数据块 DB1,定义两组为 100 个字节的数组



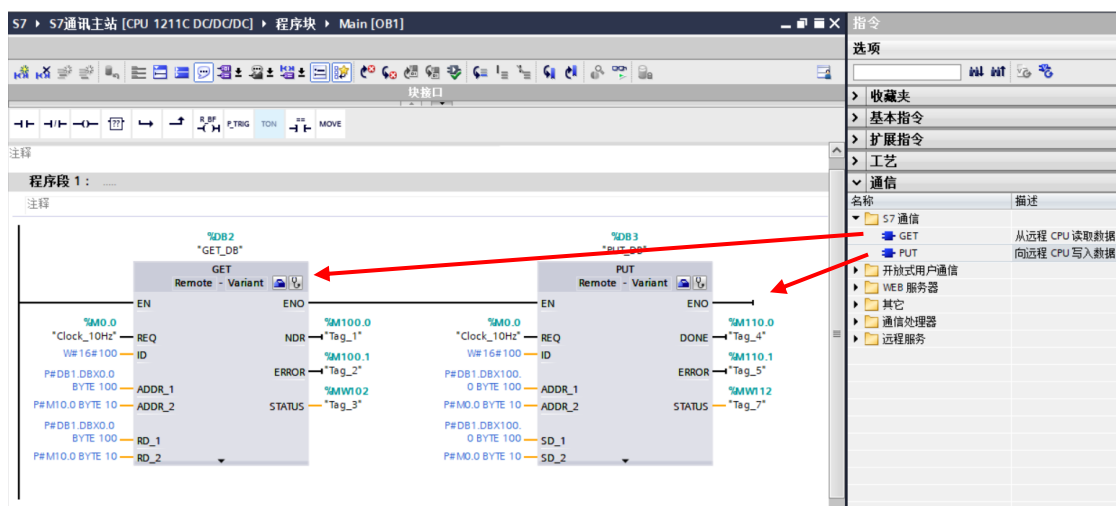
2) 在“项目树”>“数据块”>“属性”下，将优化块访问勾选去掉。



3) 创建数据块 DB1 的写数据和读数据的缓冲区，定义两组为 100 个字节的数组



4) 在 OB1 中，从“指令”→“S7 通信”下，调用 Get、Put 通信指令，程序编写如下：



```

CALL "GET" ,%DB2           //调用 PUT, 使用背景 DB 块: DB2
REQ      : =%M0.0          //系统时钟 0.1 秒脉冲
ID       : =W#16#0100      //连接号, 要与连接配置中一致, 创建连接时的连接号
DONE     : =%M100.0        //为 1 时, 发送完成
ERROR    : =%M100.1        //为 1 时, 有故障发生
STATUS   : =%MW102         //状态代码
ADDR_1   : =P#DB1.DBX0.0 BYTE 100 //从通信伙伴数据区读取数据的地址
ADDR_2   : =P#M10.0 BYTE 10 //从通信伙伴数据区读取数据的地址
SD_1     : =P#DB1.DBX0.0 BYTE 100 //本地接收数据地址
SD_2     : = P#M10.0 BYTE 10 //本地发送数据区

CALL "PUT" ,%DB3           //调用 GET, 使用背景 DB 块: DB3
REQ      : =%M0.0          //系统时钟 0.1 秒脉冲
ID       : =W#16#0100      //连接号, 要与连接配置中一致, 创建连接时的连接号
NDR      : =%M110.0        //为 1 时, 接收到新数据
ERROR    : =%M110.1        //为 1 时, 有故障发生
STATUS   : =%MW112         //状态代码
ADDR_1   : =P#DB1.DBX100.0 BYTE 100 //发送到通信伙伴数据区的地址
ADDR_1   : =P#M0.0 BYTE 10 //发送到通信伙伴数据区的地址
RD_1     : =P#DB1.DBX100.0 BYTE 100 //本地发送数据区
RD_1     : =P#M0.0 BYTE 10 //本地发送数据区

```

步骤 3: 监控结果

通过在 S7-1200 侧编程进行 S7 通信, 实现两个 CPU 之间的数据交换, 监控结果:

S7 > PLC_1 [CPU 1511-1 PN] > 监控与强制表 > 监控表_1

状态表 - [用户定义0 -- Project2\PLC1]

名称	地址	显示格式	监视值	地址	格式	当前值	新值
%MB0		带符号十进制	1	MB0	有符号	+1	
%MB1		带符号十进制	22	MB1	有符号	+22	
%MB2		带符号十进制	33	MB2	有符号	+33	
%MB3		带符号十进制	44	MB3	有符号	+44	
%MB4		带符号十进制	55	MB4	有符号	+55	
%MB5		带符号十进制	66	MB5	有符号	+66	
%MB6		带符号十进制	77	MB6	有符号	+77	
%MB7		带符号十进制	88	MB7	有符号	+88	
%MB8		带符号十进制	99	MB8	有符号	+99	
%MB9		带符号十进制	100	MB9	有符号	+100	
%MB10		带符号十进制	1	MB10	有符号	+1	
%MB11		带符号十进制	2	MB11	有符号	+2	
%MB12		带符号十进制	3	MB12	有符号	+3	
%MB13		带符号十进制	4	MB13	有符号	+4	
%MB14		带符号十进制	5	MB14	有符号	+5	
%MB15		带符号十进制	6	MB15	有符号	+6	
%MB16		带符号十进制	7	MB16	有符号	+7	
%MB17		带符号十进制	8	MB17	有符号	+8	
%MB18		带符号十进制	9	MB18	有符号	+9	
%MB19		带符号十进制	10	MB19	有符号	+10	

1200 > H226XM (MB0-MB9)
1200 < H226XM (MB10-MB19)

S7 > PLC_1 [CPU 1511-1 PN] > 程序块 > 数据块 [DB1]

名称	数据类型	偏移量	起始值	监视值	地址	格式	当前值	新值
Static								
读字节[1]	Array[1..100] of Byte	0.0	16#0	16#01	VB0	有符号	+1	
读字节[2]	Byte	1.0	16#0	16#02	VB1	有符号	+2	
读字节[3]	Byte	2.0	16#0	16#03	VB2	有符号	+3	
读字节[4]	Byte	3.0	16#0	16#04	VB3	有符号	+4	
读字节[5]	Byte	4.0	16#0	16#05	VB4	有符号	+5	
读字节[6]	Byte	5.0	16#0	16#06	VB5	有符号	+6	
读字节[7]	Byte	6.0	16#0	16#07	VB6	有符号	+7	
读字节[8]	Byte	7.0	16#0	16#08	VB7	有符号	+8	
读字节[9]	Byte	8.0	16#0	16#09	VB8	有符号	+9	
读字节[10]	Byte	9.0	16#0	16#00	VB9	有符号	+0	
读字节[11]	Byte	10.0	16#0	16#00	VB10	有符号	+0	
读字节[12]	Byte	11.0	16#0	16#00	VB11	有符号	+0	

1200 < H226X

S7 > PLC_1 [CPU 1511-1 PN] > 程序块 > 数据块 [DB1]

名称	数据类型	偏移量	起始值	监视值	地址	格式	当前值	新值
Static								
读字节	Array[1..100] of Byte	0.0	16#0	16#00	VB12	有符号	+0	
写字节	Array[1..100] of Byte	100.0	16#0	16#00	VB13	有符号	+0	
写字节[1]	Byte	100.0	16#0	16#01	VB14	有符号	+0	
写字节[2]	Byte	101.0	16#0	16#00	VB15	有符号	+0	
写字节[3]	Byte	102.0	16#0	16#03	VB16	有符号	+0	
写字节[4]	Byte	103.0	16#0	16#04	VB17	有符号	+0	
写字节[5]	Byte	104.0	16#0	16#05	VB18	有符号	+0	
写字节[6]	Byte	105.0	16#0	16#06	VB19	有符号	+0	
写字节[7]	Byte	106.0	16#0	16#07	VB20	有符号	+0	
写字节[8]	Byte	107.0	16#0	16#08	VB100	有符号	+1	
写字节[9]	Byte	108.0	16#0	16#09	VB101	有符号	+0	
写字节[10]	Byte	109.0	16#0	16#00	VB102	有符号	+3	
写字节[11]	Byte	110.0	16#0	16#00	VB103	有符号	+4	
写字节[12]	Byte	111.0	16#0	16#00	VB104	有符号	+5	
写字节[13]	Byte	112.0	16#0	16#00	VB105	有符号	+6	
写字节[14]	Byte	113.0	16#0	16#00	VB106	有符号	+7	
写字节[15]	Byte	114.0	16#0	16#00	VB107	有符号	+8	
写字节[16]	Byte	115.0	16#0	16#00	VB108	有符号	+9	
写字节[17]	Byte	116.0	16#0	16#00	VB109	有符号	+0	
写字节[18]	Byte	117.0	16#0	16#00	VB110	有符号	+0	

1200 >> H226X

注意：H226XM 中 V 区对应于 DB1，即在 PUT 指令中使用的通信伙伴数据区 ADDR_1=P#DB1.DBX.0 BYTE 100 在 H226XM 中对于为 VB0~VB99。即在 GET 指令中使用的通信伙伴数据区 ADDR_1=P#DB1.DBX100.0 BYTE 100 在 H226XM 中对于为 VB100~VB199。

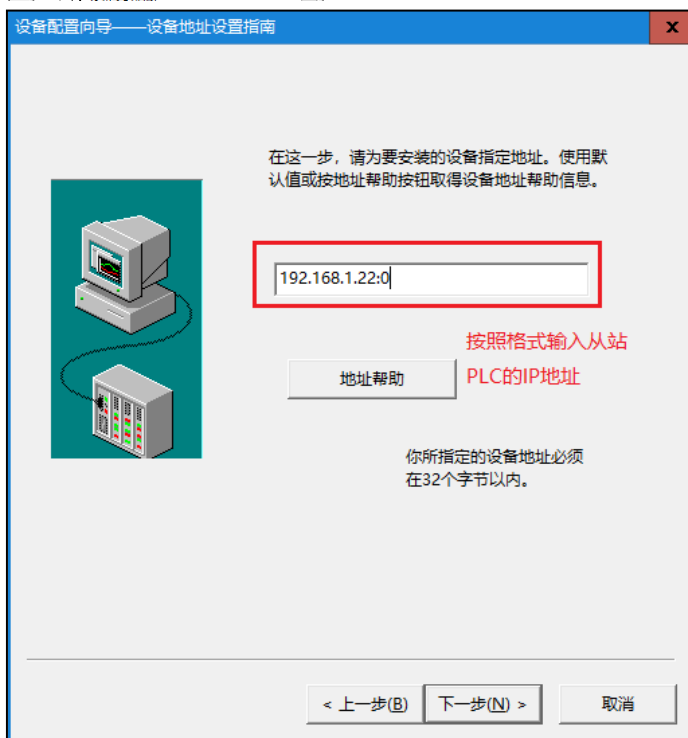
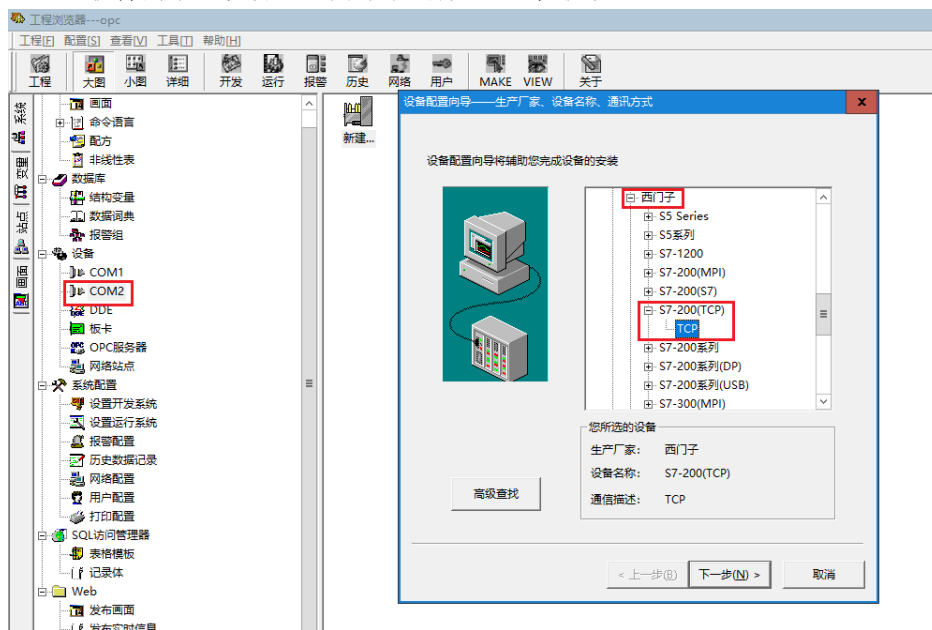
6.4.4 其它产品与 CTH2 系列高性能升级版型 PLC 的网口通讯

示例 1：组态王与 CTH2 系列高性能升级版型 PLC 的 S7 协议通信

本例以组态王为主站，CTH2 系列高性能升级版型 PLC 做从站，进行 S7 协议的通信。

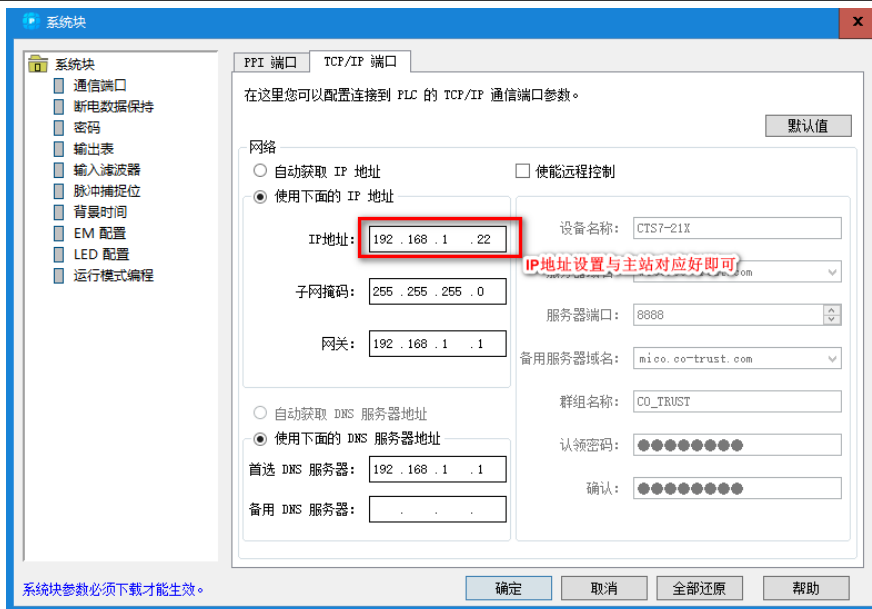
主站配置

组态王软件的串口设备配置向导中选择 S7 通信协议：



从站配置

在 CTH2 系列 高性能升级版型 PLC 的系统块中配置好 IP 地址即可，操作示意如下：



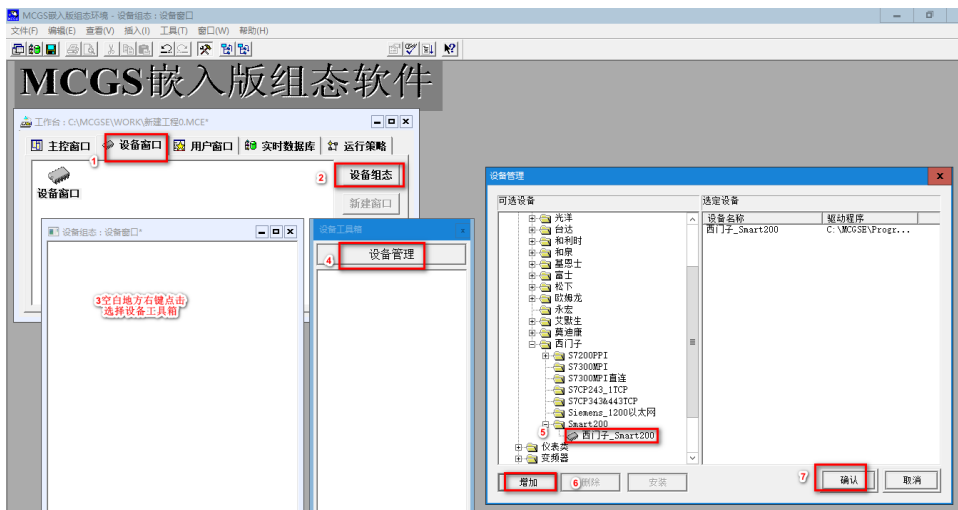
示例 2：昆仑通态触摸屏与 CTH2 系列 高性能升级版型 PLC 的 S7 协议通信

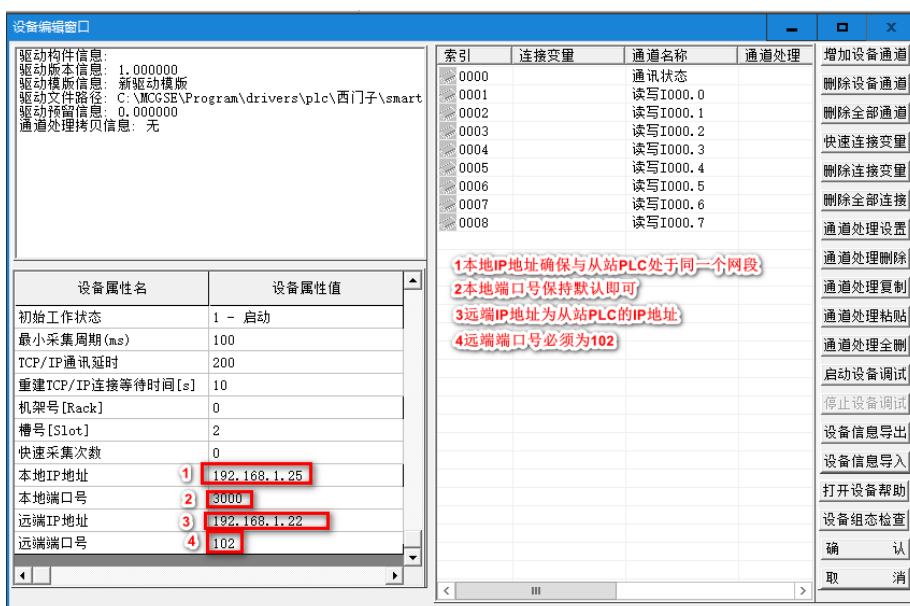
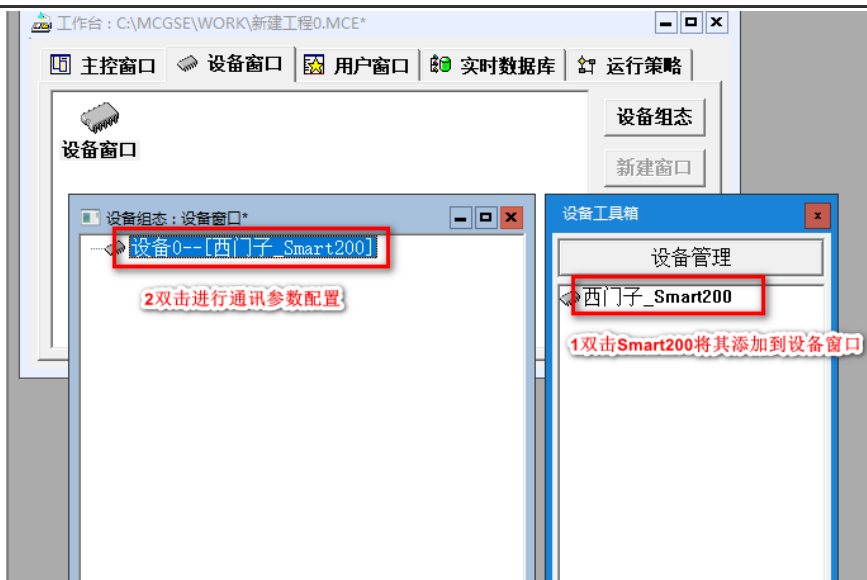
本例以昆仑通态触摸屏为主站，CTH2 系列 高性能升级版型 PLC 做从站，进行 S7 协议的通信。

备注：昆仑通态上位机也可以作为主站与 CTH2 系列 高性能升级版型 PLC 进行 S7 协议的通信。

主站配置

参考如下步骤对昆仑通态触摸屏进行 S7 协议的主站配置：





从站配置

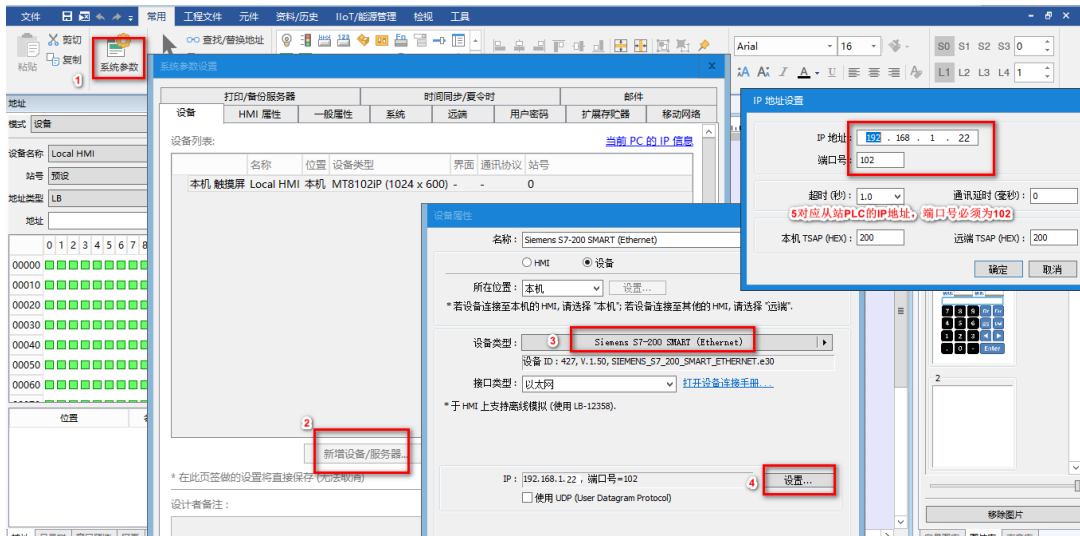
在 CTH2 系列 高性能升级版型 PLC 的系统块中配置好 IP 地址即可，操作示意如下：



示例 3：威纶通触摸屏与 CTH2 系列 高性能升级版型 PLC 的 S7 协议通信

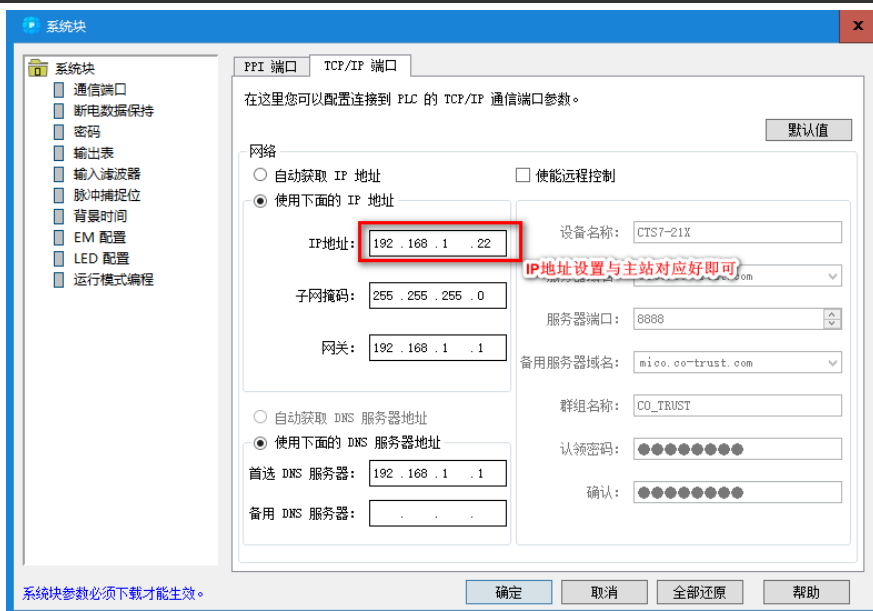
本例以威纶通触摸屏为主站，CTH2 系列 高性能升级版型 PLC 做从站，进行 S7 协议的通信。

主站配置



从站配置

在 CTH2 系列 高性能升级版型 PLC 的系统块中配置好 IP 地址即可，操作示意如下：



6.5 模拟量输入输出扩展板应用

CTH200 系列 CPU 均可以搭载模拟量输入输出扩展板使用，在应用系统中，扩展板通过总线连接接口，嵌入到 CTH200 系列 PLC 的主控 CPU 上。该扩展板把采集到的模拟量信号、处理结果和诊断信息通过总线发送给 CTH200 系列 CPU。CPU 主控系统接收到数据后，再根据用户程序以及相关信息进行一系列的处理，同时 CPU 主控系统将数字量数据发送给扩展模块，由模块控制输出模拟量信号的大小。

模拟量扩展板时有两种访问方式：调用专门的扩展指令库或直接访问 SM 区。下面以 CPU H224X 为例介绍其具体用法。

6.5.1 安装注意事项

打开 CPU 后盖上的扩展板顶盖(如下图灰框所示)，然后对准针脚安装上去固定住，再盖好扩展板顶盖就安装完成了。注意动作适度，防止压坏盖板。

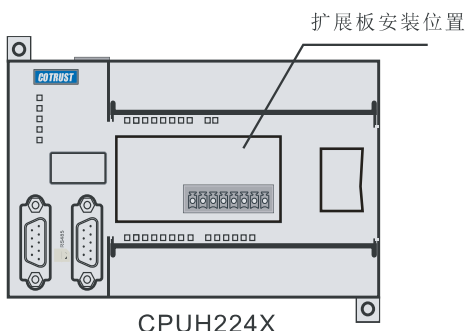


图 6-22 扩展板安装位置

有关模拟量输入输出扩展板与 CPU 之间的端子连接信息，请参考章节 4.10.1 模拟量扩展板规范。

6.5.2 扩展板访问

模拟量输入输出扩展板与 CAN-01 扩展板在使用方面略有不同，后者插接后即可直接使用，但该扩展板进行通信和编程时需要调用专用指令或使用专用 SM 存储区。

● 直接访问 SM 区

CPU 中的 SMW116~SMW126 为扩展板模拟量映射区。先映射 AI，从 SMW116 开始分配 4 个字，后映射 AQ，从 SMW124 开始分配 2 个字。

2AI/1AQ: SMW116 对应扩展板的 AIW0; SMW118 对应 AIW2; SMW124 对应 AQW0

功能	映射访问地址	功能	映射访问地址
A+输入	SMW116	模块类型	SMB114
B+输入	SMW118	模块状态	SMB115
VO 输出	SMW124		

4AI/2AQ: SMW116 对应扩展板的 AIW0; SMW118 对应 AIW2; SMW120 对应 AIW4; SMW122 对应 AIW6; SMW124 对应 AQW0; SMW126 对应 AQW2

功能	映射访问地址	功能	映射访问地址
A+输入	SMW116	VO/IO 输出	SMW124
B+输入	SMW118	V1/I1 输出	SMW126
C+输入	SMW120	模块类型	SMB114
D+输入	SMW122	模块状态	SMB115

SMB115 模块状态定义如下表所示:

名称	功能	数值
模块类型	有模块	0x1E: 4AI/2AQ 0x19: 2AI/1AQ 0x20: CAN 扩展板
	无模块	0x00
模块状态	运行正常	0x00
	通信错误	0x01
	校准失败	0x02
	访问错误	0Xff

6.6 配方和数据记录应用

6.6.1 使用配方

MagicWorks PLC 软件和 CTH200 系列中的 CPU H224X/ H226XL/ H226XM/ H226IM/ H226IL/ H226IH 支持配方功能。MagicWorks PLC 软件中提供了配方向导程序来帮助您组织配方和定义配方。

所有配方存在存储卡中。因此，为了使用配方功能，必须要在 PLC 中插入一块 64K 或者 256K 的存储卡。

所有配方存在存储卡中。但是，当用户程序处理一条配方时，该条配方被读入 PLC 的存储区。例

如：如果您生产饼干的话，您会有很多种饼干的配方，巧克力夹心饼干、甜饼干和麦片饼干。但在同一时间只能生产一种饼干，因而必须选择合适的配方读入 PLC 的存储区。

图 6-4 阐述了一个使用配方来生产多种饼干的处理过程。每一种饼干的配方存在存储卡中。操作员使用 TP10 文本显示器来选择所要生产饼干的种类，用户程序将配方读入 PLC 的存储区。

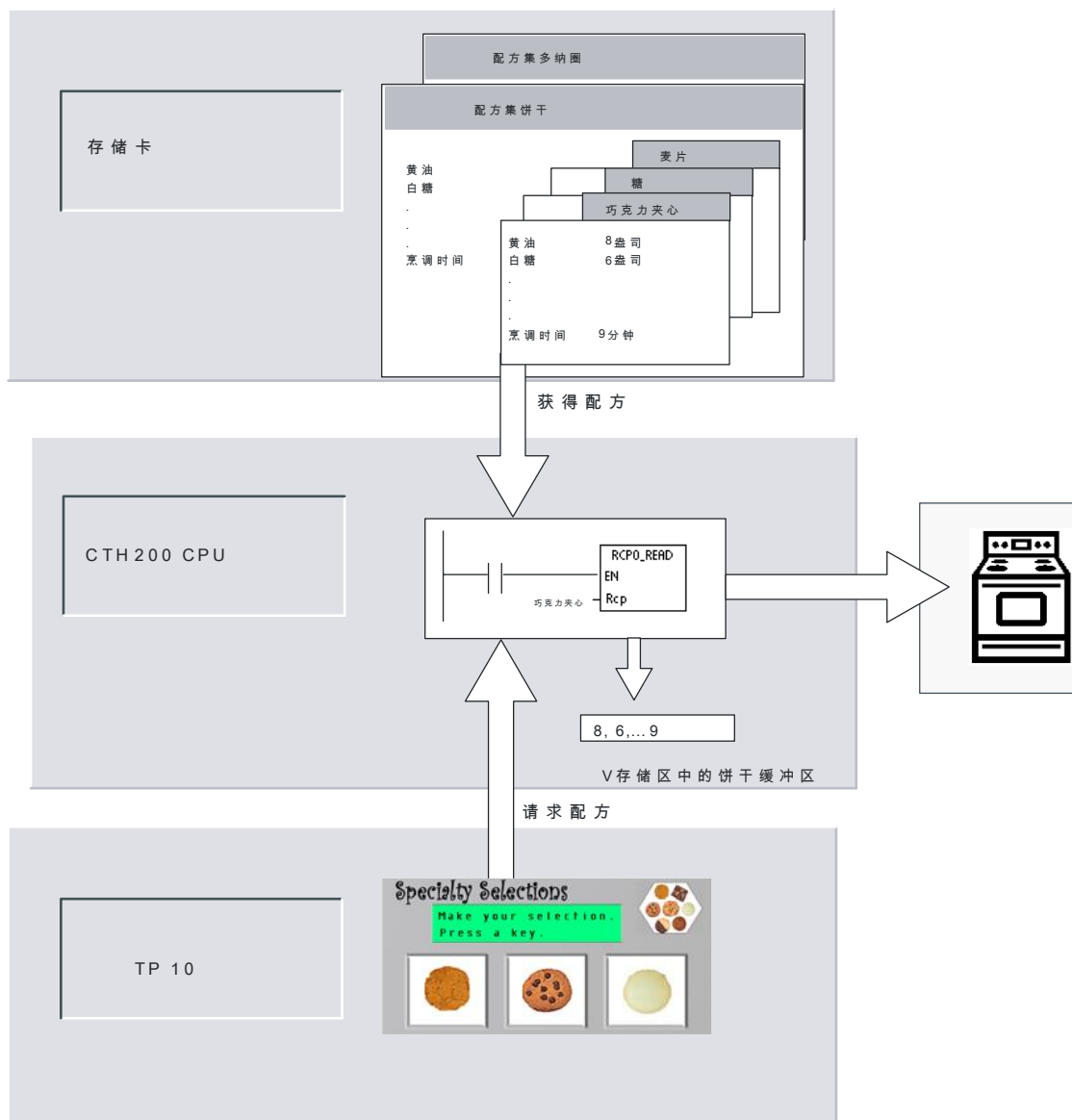


图 6-27 配方应用举例

配方的定义和术语

为了帮助您理解配方向导，首先来解释以下定义和术语。

1. 配方结构是由配方向导生成的一套组件。这些组件包括指令子程序，数据块标签和符号表。
2. 配方集是一个配方集合，它们拥有相同的参数集合。但是依赖于配方，参数的数值各不相同。
3. 一条配方是一组参数值，它提供了生产一种产品和控制一个过程所需要的信息。

例如：生产甜甜圈和饼干的配方就分别属于不同的配方集。而饼干配方集中又包含多种不同的配方，例如：巧克力夹心饼干和甜饼干。下表中列出了例子中的域和值。

配方应用举例—饼干

域名	数据类型	巧克力夹心 (配方 0)	甜饼干 (配方 1)	注释
黄油	字节	8	8	盎司
白糖	字节	6	12	盎司
红糖	字节	6	0	盎司
鸡蛋	字节	2	1	个
香草	字节	1	1	茶匙
面粉	字节	18	32	盎司
碳酸氢钠	实数	1.0	0.5	茶匙
发酵粉	实数	0	1.0	茶匙
盐	实数	1.0	0.5	茶匙
巧克力夹心	实数	16	0.0	盎司
柠檬皮	实数	0.0	1.0	大汤匙
烹调时间	实数	9.0	10.0	分钟

使用配方向导

使用配方向导来创建配方和配方集。配方是存在存储卡中的。使用配方向导可以直接输入配方和配方集。如需修改配方，可以再次运行配方向导，或者在用户程序中调用 RCPx_WRITE 指令子程序。用配方向导来创建配方结构包含以下步骤：

1. 为每个配方集建立一个符号表。每张表中都包含着与配方域名相同的符号名。L 这些符号定义了访问当前载入 PLC 存储区的配方值的 V 存储区地址。每张表还包含一个用于标识每个配方的符号常数。
2. 为每个配方集建立一个数据块标签。这个标签定义了符号表中所描述的 V 存储区地址的初始值。
3. 生成一个 RCPx_READ 指令子程序。该指令用于将指定的配方从存储卡中读取到 V 存储区中。
4. 生成一个 RCPx_WRITE 指令子程序。该指令用于将 V 存储区中的配方值写入存储卡中。

定义配方

要用配方向导创建一个配方，可在命令菜单中选择工具 > 配方向导。这时屏幕上将出现配方向导基本操作的一个简单介绍。点击“下一步”，开始配制您的配方。要创建一个配方集，请执行以下步骤，参见图 6-5 定义配方。

1. 为配方集指定域名。如同预先定义的那样，每一个名字都将成为项目中的一个符号名。
2. 在下拉列表中选择数据类型。
3. 为每个名字输入默认值和注释。在该配方集中的所有新配方将使用这些默认值作为初始值。
4. 点击“下一步”，编辑配方集中的每条配方。



图 6-28 定义配方

在配方表中为每条配方定义所有的数据域。您最多可以有 4 个配方集。而每个配方集中的配方个数，只受存储卡容量的限制。

创建和编辑配方

创建和编辑配方界面允许您创建单条配方并为这些配方分配数值。每一个可编辑的列都表示一个独立的配方。

可以按“增加配方”按钮来创建配方。每个配方会将创建配方集时所指定的默认值作为初始值。

也可以用另外一种方式创建配方。单击鼠标右键使用关联菜单，您可以复制和粘贴已有的配方。新的列将被插入在当前光标位置的左侧。

根据配方集和配方号，每条新的配方将被赋予一个默认名称。该默认名称格式为：DEFx_RCPy。

要创建并编辑配方，执行以下步骤。

参见图 6-6。

1. 点击“下一步”，进入创建和编辑配方窗口。
2. 点击“增加配方”按钮，插入一条配方。
3. 将配方默认名改为一个合适名称。
4. 根据需求改变每个配方数据的值。
5. 点击“下一步”。

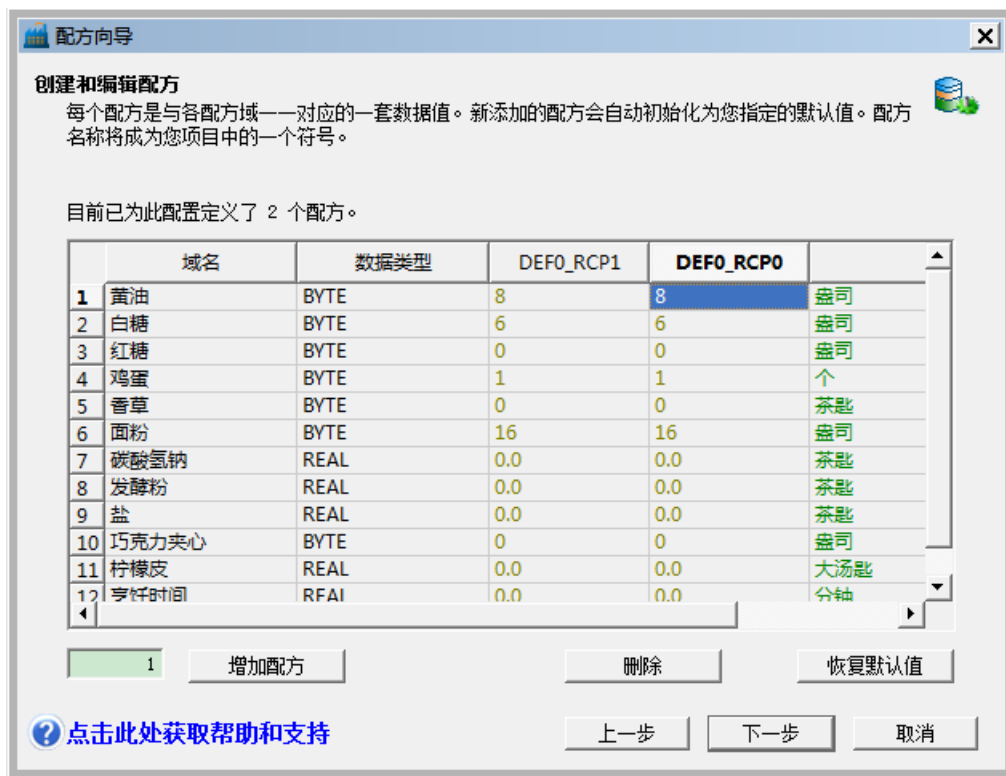


图 6-29 创建和编辑配方

分配存储区

分配存储区画面用于指定 V 存储区的起始地址，从这一起始地址开始存储从存储卡中读取的配方。您可以自己选择 V 存储区地址，也可以使用配方向导建议的地址，配方向导会推荐您使用正确长度的尚未使用的 V 存储区。

要为配方分配存储区，执行以下步骤。

参见图 6-7。

1. 在窗口中输入地址值作为您希望储存配方的 V 存储区起始地址。
2. 您也可以通过点击“建议地址”按钮，让配方向导为您选择一个正确长度的尚未使用的 V 存储区。
3. 点击“下一步”。

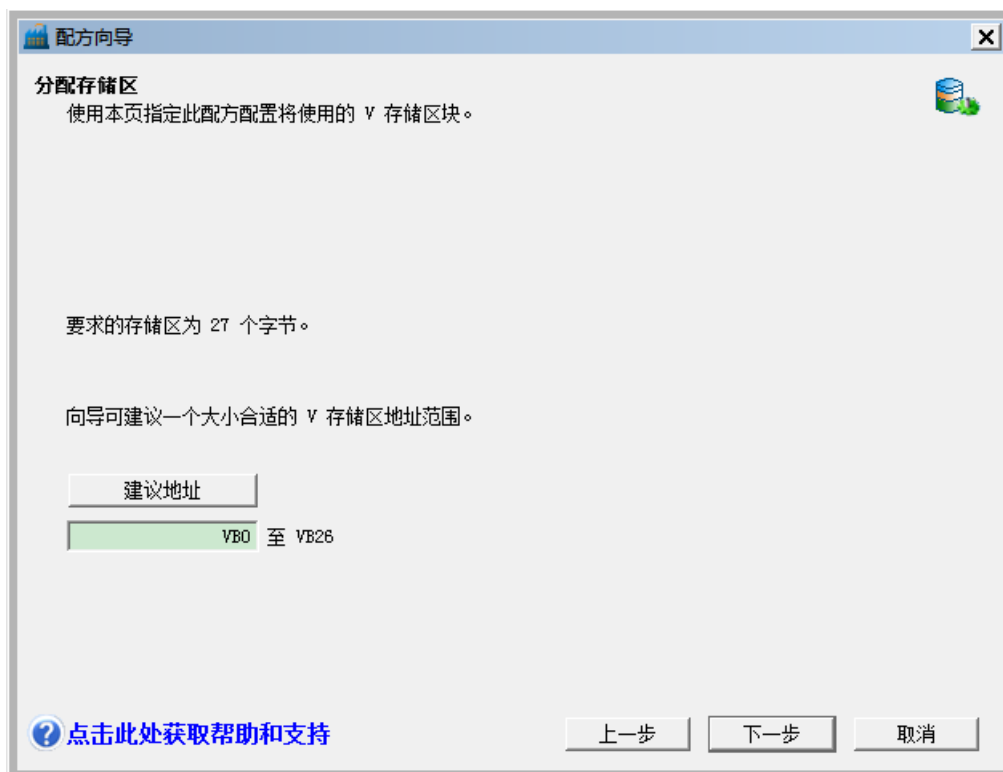


图 6-30 分配存储区

项目组件

项目组件画面列出了将要被添加到您项目中的不同组件。参见图 6-8。点击“完成”来完成配方向导并将这些组件添加到项目中。每个配方结构拥有唯一的名字，这些名字会显示在项目树中，配方集名（RCPx）附加在名字尾部。

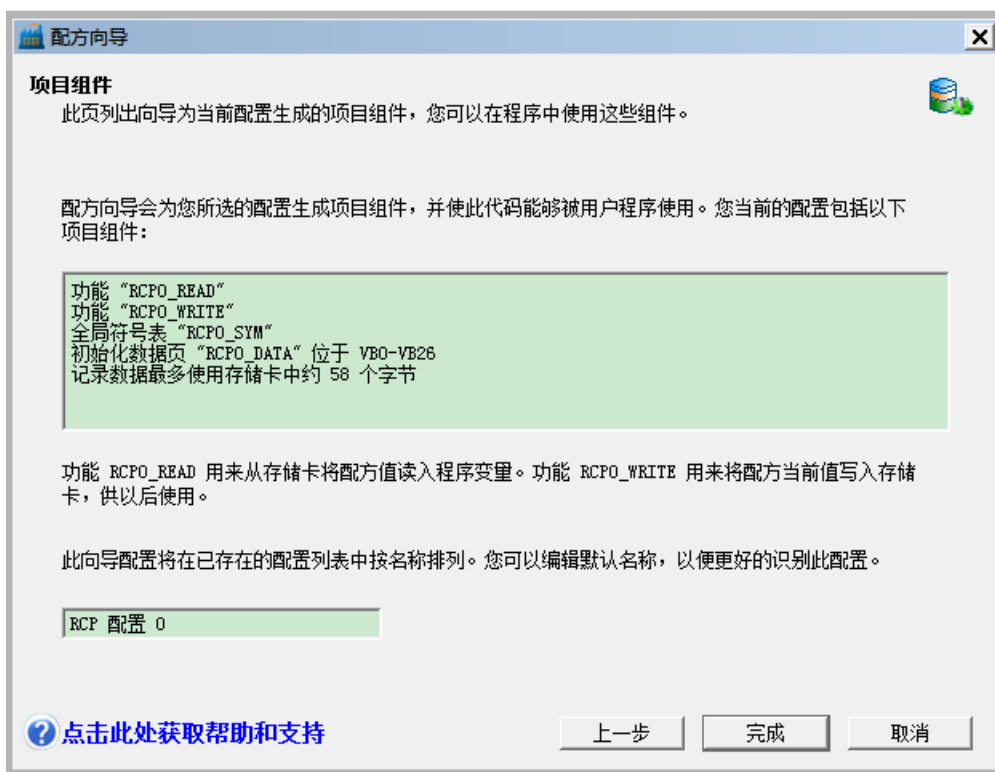


图 6-31 项目组件

使用符号表

为每一个配方集创建一个符号表。每张表定义一些常用数值来表示每条配方。可以在 RCPx_READ 和 RCPx_WRITE 指令中使用这些符号来表示想要的配方。参见图 6-9。每张表中也为配方中的每个域创建符号名。您可以使用这些符号来访问 V 存储区中的配方值。

状态列	符号	地址	
2	DEFO_RCP0	1	
3	黄油	VB0	盎司
4	白糖	VB1	盎司
5	红糖	VB2	盎司
6	鸡蛋	VB3	个
7	香草	VB4	茶匙
8	面粉	VB5	盎司
9	碳酸氢钠	VD6	茶匙
10	发酵粉	VD10	茶匙
11	盐	VD14	茶匙
12	巧克力夹心	VB18	盎司
13	柠檬皮	VD19	大汤匙
14	烹饪时间	VD23	分钟

图 6-32 符号表

下载一个带有配方的项目

要下载一个带有配方的项目，执行以下步骤。参见图 6-10。

1. 选择“文件—>下载”。
2. 在对话框中，确保程序块、数据块和配方均被选中。
3. 点击“下载”按钮。



图 6-33 下载配方

删除已有的配方

要编辑已有的配方，执行以下步骤。参见图 6-11。

1. 打开配方向导，点击下拉列表选择一个已有的配方。
2. 可以通过“删除配置”按钮来删除一个已有的配方。

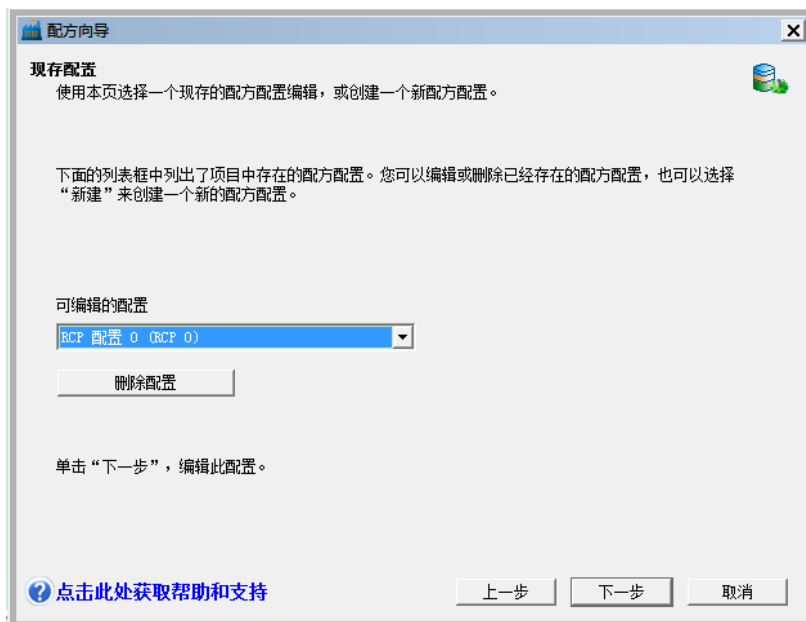


图 6-34 编辑配方

由配方向导创建的指令

RCPx_Read 子程序

子程序 RCPx_READ 是由配方向导创建的，它用于将配方从存储卡中读取到 V 存储区中。

RCPx_READ 指令中的 x 是指包含您想读取的配方的配方集编号。当 EN 输入为高电平时，允许指令执行。Rcp 输入端决定了从存储卡中读取哪条配方。Error 输出端返回该指令的执行结果。表 6-8 中给出了错误代码表。

RCPx_Write 子程序

子程序 RCPx_WRITE 是由配方向导创建的。调用该指令可以使 V 存储区中的配方内容替代存储卡中的配方。RCPx_WRITE 指令中的 x 是指包含您想替代的配方的配方集编号。当 EN 输入为高电平时，允许指令执行。Rcp 输入决定了替代存储卡中的哪条配方。Error 输出端返回该指令的执行结果。表 6-8 中给出了错误代码表。

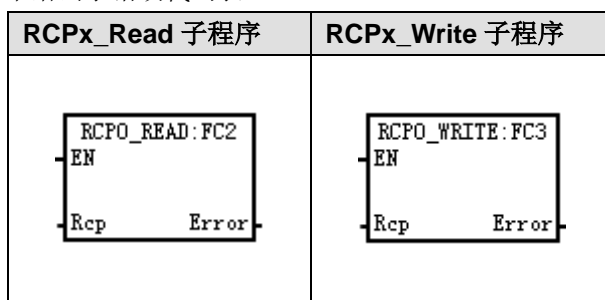


表 6-7 配方子程序的有效操作数

输入/输出	数据类型	操作数
Rcp	字	VW, IW, QW, MW, SW, SMW, LW, AC, *VD, *AC, *LD, 常数
Error	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

表 6-8 配方子程序的错误代码

错误代码	描述
0	无错误
132	存储卡访问失败



提示

EEPROM 存储卡的写操作有次数限制，典型值为一百万次。超出限制，EEPROM 将失效。因而请务必确认不是在每个程序周期中都执行 RCPx_WRITE 指令。否则在很短的时间内，存储卡就会被损坏。

6.6.2 使用数据记录

MagicWorks PLC 提供数据记录向导，将过程测量数据存入存储卡中。将过程数据移入存储卡可以节省 V 存储区的地址空间，否则这些数据将储存在 V 存储区中。

MagicWorks PLC 软件和 CTH200 PLC 已经支持数据记录功能。使用这一功能，您可以在程序控制下永久地保存过程数据记录。这些记录可以包含时间日期标签。您最多可以配置 4 个独立的数据记录。可以用新的数据记录向导来定义数据记录的记录格式。

所有数据记录都存在存储卡中。因此，为了使用数据记录功能，必须要在 PLC 中插入一块 64K 或者 256K 的存储卡。不过，您必须使用 S7—200 资源管理器将数据记录中的内容上载到计算机。

图 6-12 中显示了数据记录应用的一个实例。

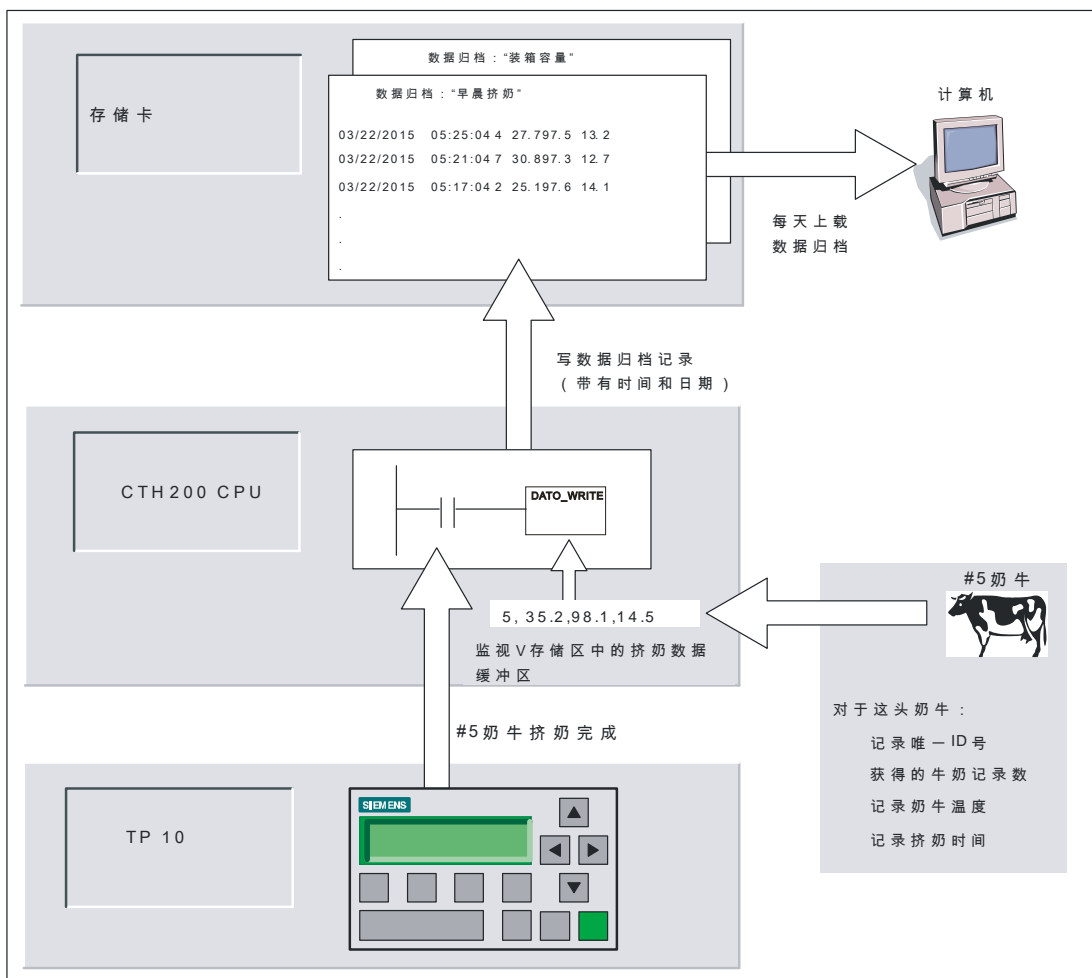


图 6-35 数据记录应用实例

数据记录的定义和术语

为了帮助您理解数据记录向导，首先来解释以下定义和术语。

1. 数据记录是指通常按照日期时间排序的一组记录。每条记录代表着一些过程事件，过程事件中记录了一套过程数据。由数据记录向导来定义数据的组织结构。
2. 一条数据记录记录是指写入数据记录中的单独的数据行。

使用数据记录向导

使用数据记录向导最多可以配置 4 个数据记录。数据记录向导用于：

1. 定义数据记录记录的格式。
2. 选择数据记录的可选项，例如：时间标签、日期标签和有上载时清除数据记录等。
3. 指定数据记录中储存记录的最大数目。
4. 创建用于在数据记录中储存记录的项目代码。

用数据记录向导创建数据记录包含以下步骤：

1. 为每个数据记录创建一个符号表。每张表中都包含着与数据记录域名相同的符号名。这些符号定义了储存当前数据记录所需的 V 存储区的地址。每张表还包含一个用于标识每个数据记录的符号常数。
2. 为每条数据记录记录建立一个数据块标签，从而为每一个数据记录域分配了 V 存储器地址。用户程序使用这些 V 存储区地址来采集当前记录数据。
3. 生成一个 DATx_WRITE 子程序。这条指令将指定的数据记录从 V 存储区复制到存储卡中。每执行一次 DATx_WRITE 指令，将会在存储卡的数据记录中添加一条新的数据记录。

数据记录选项

您可以为您的数据记录配置如下可选项。参见图 6-36。

时间标签

您可以在每条数据记录记录中包含时间标签。如果您选中该项，当用户程序写入一条数据记录记录时，CPU 将自动在记录中加入时间标签。

日期标签

您可以在每条数据记录记录中包含日期标签。如果您选中该项，当用户程序写入一条数据记录记录时，CPU 将自动在记录中加入日期标签。

清除记录数据

当数据被上载后，您可以清除数据记录中的所有记录。如果您选中该项，每次上载之后，数据记录将被清除。

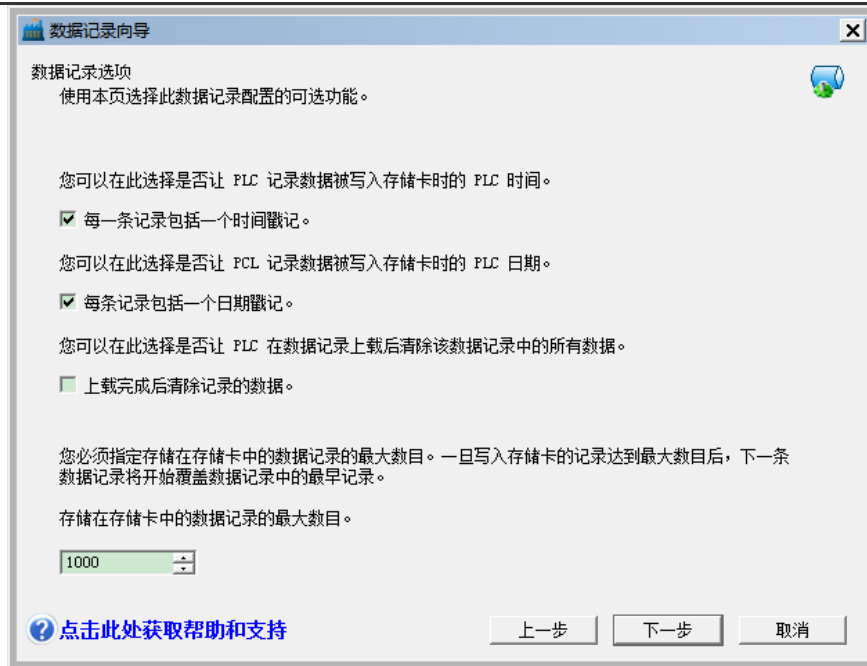


图 6-36 数据记录选项

数据记录是一个环形队列（当记录满时，一条新的记录将代替最旧的那条记录）。您必须指定数据记录中储存记录的最大数目。最大数目的最大值为 65534，默认值为 1000。

定义数据记录

您为数据记录指定域，每个域都成为项目中的一个符号。您必须为每个域指定数据类型。一条数据记录记录可以包含 4 到 203 个字节的数据。要在数据记录中定义数据域，执行以下步骤。参见图 6-37。

1. 点击“域名”单元格来输入域名。名称变为用户程序引用的符号。
2. 点击“数据类型”单元格，从下拉列表中选择数据类型。
3. 点击“注释”单元格来输入注释。
4. 根据需要使用多行来定义一条记录。
5. 点击“下一步”。



图 6-37 定义数据记录

分配存储区

数据记录向导在 PLC 的 V 存储区中创建一个块。该块是一个存储区地址，一条数据记录记录在被写入存储卡之前，存储在这个存储区地址中。您可以为您要放置配置的 V 存储区指定起始地址。您可以自己选择 V 存储区地址，也可以使用数据记录向导建议的地址。数据记录向导会推荐您使用正确长度的尚未使用的 V 存储区。块的长度根据您在数据记录向导中的不同选择而有所不同。参见图 6-38。

要为数据记录分配存储区，执行以下步骤：

1. 在窗口中输入地址值作为您创建数据记录记录的 V 存储区地址。
2. 您也可以点击“建议地址”按钮，让数据记录向导为您选择一个正确长度的尚未使用的 V 存储区。
3. 点击“下一步”。

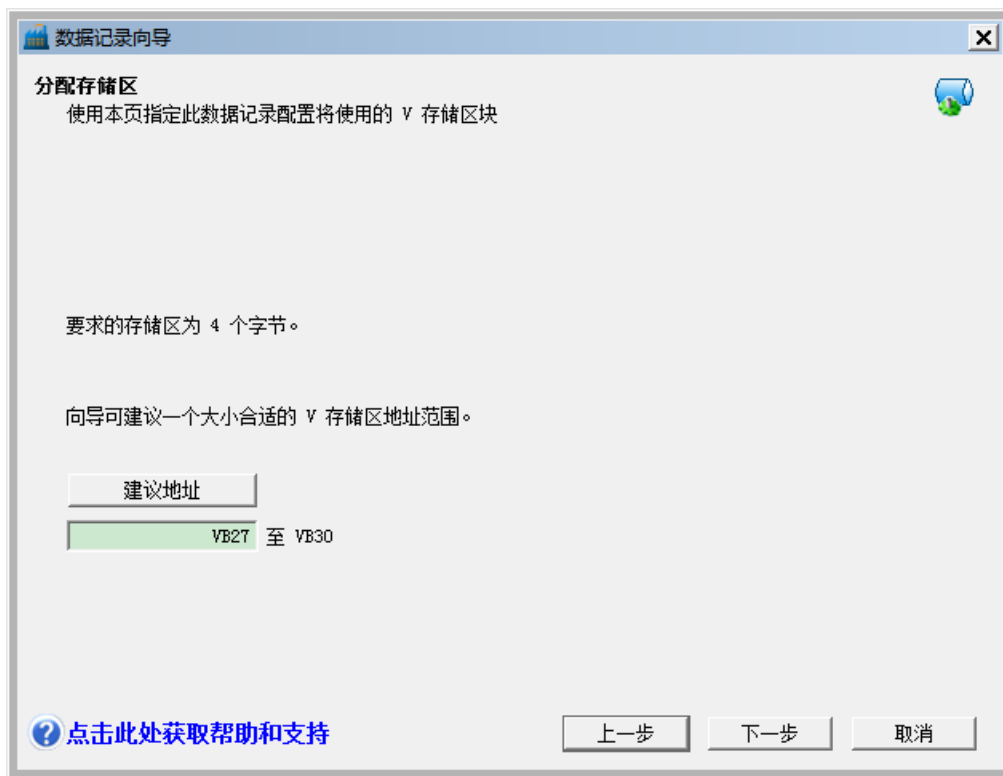


图 6-38 分配存储区

项目组件

项目组件画面列出了将要被添加到您项目中的不同组件。参见图 6-39。点击“完成”来完成数据记录向导，并将这些组件添加到项目中。每个数据记录结构拥有唯一的名字。这些名字会显示在项目树中。数据记录集名 (DATx) 被附加在名字尾部。



图 6-39 项目组件

使用符号表

为每一个数据记录创建一个符号表。每张表定义一些常数来表示每个数据记录。可以在 DATx_WRITE 指令中使用这些符号。每张表中也为数据记录中的每个域创建符号名。您可以使用这些符号来访问 V 存储区中的数据记录数值。

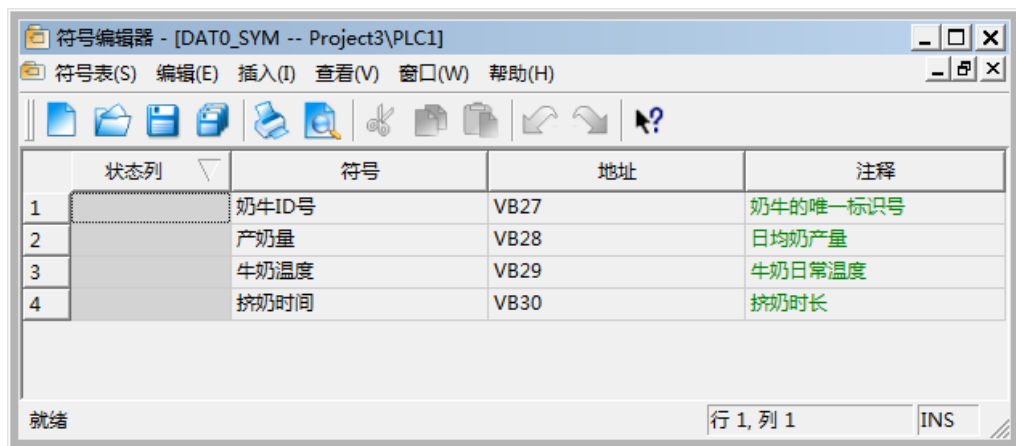


图 6-40 符号表

下载一个带有数据记录的项目

在使用数据记录之前，您必须将带有数据记录的项目下载到 CPU 中。如果一个项目中带有数据记录，那么在下载窗口中，默认选中“数据日志”选项。



提示

当您下载一个带有数据记录的项目时，当前存在存储卡中的所有数据记录记录将丢失。

要下载一个带有数据记录的项目，执行以下步骤。参见图 6-41。

1. 选择“文件—> 下载”。
2. 在对话框中，确保选中“数据记录”选项。
3. 点击“下载”按钮



图 6-41 下载含数据日志的项目

使用 S7—200 资源管理器

S7—200 资源管理器用于从存储卡中读取数据记录，并将数据记录存储在 CSV 文件中。每次读取数据记录，都创建一个新文件。这个文件存在数据记录目录中。文件名格式为：PLC 地址，数据记录名，日期和时间。您可以选择当数据记录被成功读取时，是否自动启动与 CSV 扩展名相关联的应用程序。您可以用鼠标右键单击数据记录文件，在关联菜单中作出选择。数据记录目录将在安装过程中被指定。如果没有安装 STET 7，默认的安装目录为 c:\programfiles\siemens\Microsystems。如果安装了 STEP 7，默认的安装目录为 c:\siemens\Microsystems。

要读取数据记录，执行以下步骤：

1. 打开 Windows 资源管理器，将看到 My S7—200 Network 文件夹。
2. 选择 My S7—200 Network 文件夹。
3. 选择正确的 S7—200 PLC 文件夹。
4. 选择存储卡文件夹。
5. 找到正确的数据记录文件。这些文件的名称为 DAT Configuration x (DATx)。
6. 用鼠标右键调出上下文关联菜单，选择 Upload。

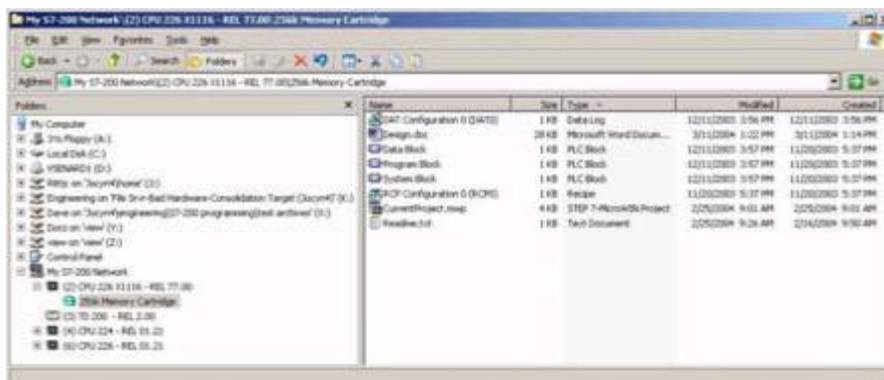


图 6-42 使用 S7—200 资源管理器

编辑已有的数据记录

要编辑已有的数据记录，执行以下步骤：

1. 如图 6-43 所示，点击下拉列表，选择一个已有的数据记录。
2. 可以通过“删除配置”按钮来删除一个已有的数据记录。您可以拥有最多 4 个不同的数据记录。

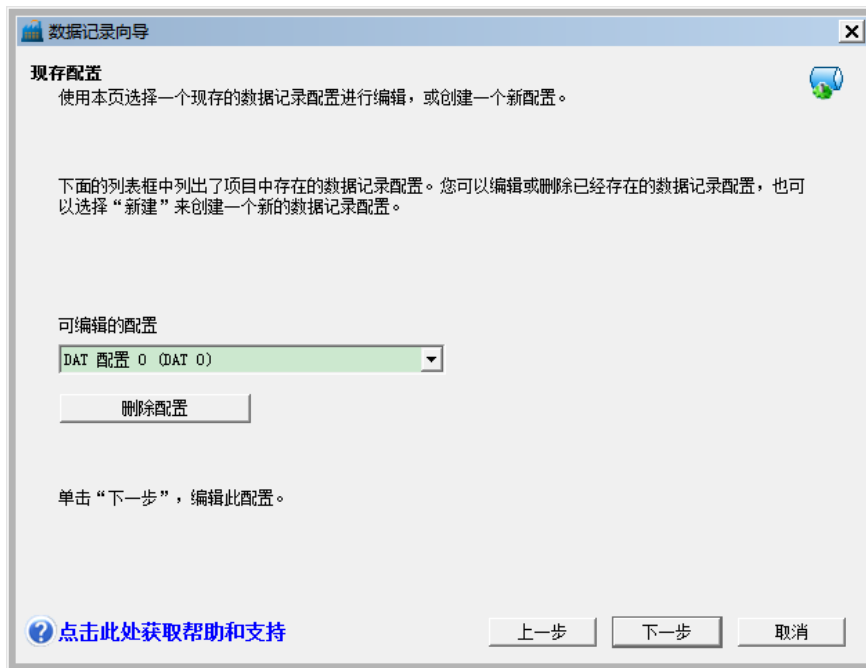


图 6-43 编辑已有的数据记录配置

由数据记录向导创建的指令

数据记录向导会在您的项目中添加一条指令子程序。

DATx_WRITE 子程序

子程序 DATx_WRITE 用于将数据记录域的当前值写入存储卡。DATxWRITE 将记录数据中的一条记录写入存储卡。子程序的调用如图所示。



当指令访问存储卡失败时，会返回错误代码 132。

表 6-9 DATx_WRITE 子程序的参数

输入/输出	数据类型	描述
Error	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD



提示

EEPROM 存储卡的写操作有次数限制，典型值为一百万次。一旦超出限制，EEPROM 将失效。因而请务必确认不是在每个程序周期中都执行 DATx_WRITE 指令。否则在很短的时间内，存储卡就会被损坏。

7 电源预算

CTH200 系列 CPU 本单元有一个内部电源，它可以为本单元和扩展单元供电。同时 CTH200 系列 CPU 提供 5VDC 逻辑电源，可用于系统中任何扩展单元供电。需密切注意系统配置，以确保 CPU 能提供所选扩展模块所需 5V 供电。如果配置需要的电源大于 CPU 提供的电源，必须删除某些扩展模块或选择可以提供更大电源的 CPU。参考章节 [7.2 5VDC 电源](#) 了解关于 CTH200CPU 所提供的 5VDC 逻辑预算与扩充模块 5VDC 电源要求信息。

CTH200 系列 CPU 还提供 24VDC 传感器电源，可为输入点、扩充模块上的继电器线圈或其它需要电源提供 24VDC。如果所需电源超出传感器电源预算，就必须在系统中添加外部 24VDC 电源。参考章节 [7.3 24VDC 电源](#) 了解特定 CTH200 系列 CPU 的 24VDC 传感器电源预算。如果确实需要外部 24VDC 电源，请确保电源没有并行连接到 CTH200 系列 CPU 的传感器电源。为提高电噪声保护，建议请连接到不同电源的公共端（M）。

7.1 电源要求

CPU 本单元提供 5VDC 和 24VDC 电源：

大多数 CPU 本体单元都有一个 24VDC 的外供电源，它可以为本体输入点和扩展模块的继电器线圈提供 24VDC。

当有扩展单元连接时，CPU 本体单元为扩展单元提供 5VDC 电源。如果扩展单元的 5V 电源要求超出了 CPU 模块的电源额定输出，必须卸下扩展单元，直到需求在电源预定值之内才行。

第4章技术规范已规定提供关于 CPU 电源预算和扩充模块电源要求的信息。



提示

如果 5VDC 电源超出 CPU 电源预算，可能无法连接 CPU 允许的最大数目的模块。



注意

和 CTH200DC 传感器的电源并联连接的外部 24VDC 电源会导致冲突，因为每个电源都寻求建立自己的电位输出。这种冲突会使一个或两个电源使用寿命缩短或立即故障，随后对 PLC 系统进行不确定的操作。这种不可预知的操作会导致人身伤害和设备毁坏。

CTH200 系列 CPU 的 24VDC 传感器的电源和任何外部电源应该分别为不同点供电。而公用电源的单独连接是允许的。

7.2 5VDC 电源

表 7-1 5VDC 电源功耗表

CPU 和 DP 模块	扩展 IO 的 5VDC 电流 (Ma)	扩展模块	5VDC 电流消耗(Ma)
		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 运控模块	90
CPU H224	660		
CPU H224X	660		
CPU H226L	660		
CPU H226M	660		
CPU H226XL	660		
CPU H226XM	660		
CPU H228XL	660		
CPU H226IM	660		
CPU H226IL	660		
CPU H226IH	660		
SM277B	660		
SM277C	660		
SM277PN	1100		

根据上表的功耗，计算 CTH200 系统的最大输入输出配置参看 [7.4 电源预算计算示例](#)。

7.3 24VDC 电源

表 7-2 24VDC 电源功耗表

24VDC 电流供应 (Ma)		扩展模块 24VDC 电流消耗 (Ma)	
		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 运控模块	80
CPU H224	300		
CPU H224X	300		
CPU H226L	300		
CPU H226M	300		
CPU H226XL	300		
CPU H226XM	300		
CPU H228XL	300		
CPU H226IL	300		
CPU H226IM	300		
CPU H226IH	300		
SM277B	400		
SM277C	300		
SM277PN	800		

7.4 电源预算计算示例

利用以下设计方法确定本体单元电源可以提供多少单元的功率配置。

表 7.3 所示是一个 CTH200 系列 PLC 的电源预算计算，系统中包含下列内容：

- CPU H224X
- SM223 扩展模块，其中 SM223-1PH32 有 8 个直流输入/8 个继电器输出
- SM221 扩展模块，其中 SM221-1BF32 有 8 个直流输入
- 该装置共有 62 个输入和 42 个输出。

此例中 CTH200 CPU 为扩展模块提供足够的 5VDC 电流，但没有为所有输入和扩充继电器线圈从传感器提供足够的 24VDC 电流。I/O 需要 536 毫安，而 CTH200CPU 只提供 300 毫安。此装置需要至少 236 毫安的 24VDC 附加电源以操作所有包含的 24VDC 输入和输出。

表 7-3 电源预算表

CPU 本体电源预算	5VDC	24VDC
CPU H224X 本体电源可提供电流	660 Ma	300 Ma
减		
系统要求	5VDC	24VDC
CPU H224X, 14 个输入		14* 4 Ma= 56 Ma
4 个 SM 223-1PH32, 需要 5V 电源电流	4*89 Ma= 356Ma	
2 个 SM 221-1BF32, 需要 5V 电源电流	2*57 Ma= 114 Ma	
4 个 SM 223-1PH32, 每个有 8 个输入		4*8*4 Ma= 128 Ma
4 个 SM 223-1PH32, 每个有 8 个继电器线圈		4*8*9 Ma= 288 Ma
2 个 SM 221-1BF32, 每个有 8 个输入		2*8*4 Ma= 64 Ma
总需求电源消耗	470 Ma	536 Ma
等于		
电流平衡预算	5VDC	24 VDC
总电流平衡	190 Ma	-236 Ma

8 故障诊断

当系统出现故障时，CPU 执行以下操作：

- 1) 进入 STOP（停止）模式
- 2) 点亮 SF/DIAG(红)LED 指示灯和 STOP 指示灯
- 3) 断开输出

请先检查以下条件是否满足：

- 1) CTH200 系列主控模块及扩展模块是否正常供电。
- 2) CTH200 系列主控模块及扩展模块 I/O 端子的螺丝和接插件是否松动。
- 3) 检查通信电缆的连接情况，确保无误。
- 4) 搜索不到 PLC，请检查通信设置，例如改变波特率、连接串口或 IP 等重新搜索。

除以上方法外，还可通过 MagicWorks PLC 读取诊断信息，或者通过 PLC 的 LED 指示灯状态检查 PLC 自身和外部有无异常。

8.1 通过 MagicWorks PLC 进行诊断

故障状态会持续到清除错误之后，诊断信息读取方式：打开 MagicWorks PLC 软件→双击打开项目界面→选择菜单项“PLC”→“信息”即可查看错误信息。

表 8-1 诊断功能

支持的事件类型	具体事件编码及描述	
	编码	事件描述
CPU 非致命错误	0x00	无错误
	0x01	在执行 HDEF 方框之前启用 HSC（保留）
	0x02	输入中断分配冲突，分配至已经分配给 HSC 的点（保留）
	0x03	输入分配冲突，分配至已经分配给输入中断或其他 HSC 的 HSC（保留）
	0x04	尝试在中断例行程序中执行 ENI、DISI 或 HDEF 指令
	0x05	尝试在完成第一个 HSC/PLS 之前执行第二个带有相同号码的 HSC/PLS，与主程序中的 HSC/PLS 发生中断例行程序冲突（保留）
	0x06	间接编址错误
	0x07	TODW（日写入时间）或 TODR（日读取时间）数据错误
	0x08	超出最大用户子程序嵌套层数
	0x09	在端口0中同时执行 XMT/RCV 指令
	0x0A	尝试通过执行另一条用于相同 HSC 的 HDEF 指令重新定义 HSC（保留）
	0x0B	在端口1中同时执行 XMT/RCV 指令
	0x0C	保留
	0x0D	保留

	0x0E	保留
	0x0F	在比较触点指令中遇到非法数字数值
	0x10	保留
	0x11	保留
	0x12	保留
	0x13	非法 PID 回路表
	0x80	程序过大，CPU 无法生成可执行代码；请缩小程序大小
	0x81	保留
	0x82	非法指令；检查指令助记符0
	0x83	缺少 MEND，或主程序中不允许存在指令；增加 MEND 指令或删除不正确的指令
	0x85	缺少 FOR；增加 FOR 指令或删除 NEXT 指令
	0x86	缺少 NEXT；增加 NEXT 指令或删除 FOR 指令
	0x87	缺少标签（LBL、INT、子程序）；增加适当的标签
	0x88	缺少 RET 或子程序中不允许存在指令；在子程序结尾处增加 RET 或删除不正确的指令
	0x89	缺少 RETI 或中断例行程序中不允许存在指令；在中断例行程序结尾处增加 RETI 或删除不正确的指令
	0x8B	至 SCR 段非法 JMP 或从 SCR 段非法 JMP
	0x8C	重复标签（LBL、INT、SBR）；为其中一个标签重新命名
	0x8D	非法标签（LBL、INT、SBR）；核实未超出允许使用的标签数
	0x90	非法参数；核实指令允许使用的参数
	0x91	范围错误（包括地址信息）；检查操作数范围
	0x92	指令计数域错误（包括计数信息）；核实最大计数
	0x93	超过 FOR/NEXT 嵌套层数
	0x94	用地址信息向非易失性内存写入范围错误
	0x95	缺少 LSCR 指令（载入 SCR）
	0x96	缺少 SCRE 指令（SCR 结束）或在 SCRE 前出现不允许使用的指令
	0x97	用户程序包含不带号码及带号码的 EU/ED 指令
	0x98	尝试在配备不带号码 EU/ED 指令的程序中执行运行时间编辑
	0x99	过多隐含程序段
	0x9A	在用户中断中尝试转入自由端口模式
	0x9B	非法索引（字符串操作，已指定该操作中的一个起始位置数值0）
CPU 致命错误	0x00	不存在严重错误
	0x01	保留
	0x02	保留
	0x03	扫描看门狗超时错误
	0x04	保留
	0x05	保留
	0x06	保留
	0x07	保留
	0x08	保留
	0x09	保留
	0x0A	保留






	0x0B	保留
	0x0C	保留
	0x0D	保留
	0x0E	保留
	0x0F	保留
	0x10	内部软件错误
	0x11	比较触点间接编址错误
	0x12	比较节点非法浮点数值错误
	0x13	保留
	0x14	比较节点范围错误
	0x15	电源波动报警
每个扫描周期监视记录一次		
模块诊断事件 参考表8-2	0x00	模块无故障
	0x01	模块忙
	0x02	模块超时未响应
	0x03	模块类型不匹配
	0x04	模块版本不匹配
	0x05	软件错误
	0x06	模块等待标志超时
	0x07	总线应答错误
	0x08	总线 CRC 校验错误
	0x10	内存偏移量超范围
	0x11	模块没有准备好
	0x12	模块组态错误
	0x13	模块不支持本条指令
	0x15	模块内部诊断
	0x16	模块没有电源

表 8-2 特殊存储区诊断功能

SMB	错误描述
SMB8	模块 1 标识寄存器
SMB9	模块 1 错误寄存器
SMB10	模块 2 标识寄存器
SMB11	模块 2 错误寄存器
SMB12	模块 3 标识寄存器
SMB13	模块 3 错误寄存器
SMB14	模块 4 标识寄存器
SMB15	模块 4 错误寄存器
SMB16	模块 5 标识寄存器
SMB17	模块 5 错误寄存器
SMB18	模块 6 标识寄存器
SMB19	模块 6 错误寄存器
SMB20	模块 7 标识寄存器
SMB21	模块 7 错误寄存器
SMB200~SMB549	智能模块状态

8.2 通过 CTH200 系列主控模块进行诊断

表 8-4 CTH200 的 LED 状态指示灯状态描述

指示灯	颜色	描述
SF/DIAG 指示灯	红色 	亮起：系统故障，熄灭：无错
RMC 指示灯	绿色 	亮起：CPU 与远程服务器通讯成功（Ethernet 通信口参数已正确配置） 熄灭：与远程服务器通讯失败或禁止与远程服务器通讯（通过拨码开关控制是否允许和禁止和服务器通讯）
RUN 指示灯	绿色 	亮起：系统运行，熄灭：系统停止
STOP 指示灯	橙色 	亮起：系统停止，熄灭：系统运行
LINK/ACT 指示灯	绿色 	亮起：连接，闪烁：传输，熄灭：断开


<注>：在设备开始掉电到完全掉电的电源波动阶段，STOP 指示灯和 SF 指示灯会同时亮起，系统会记录一个事件。

9 Trace 追踪功能

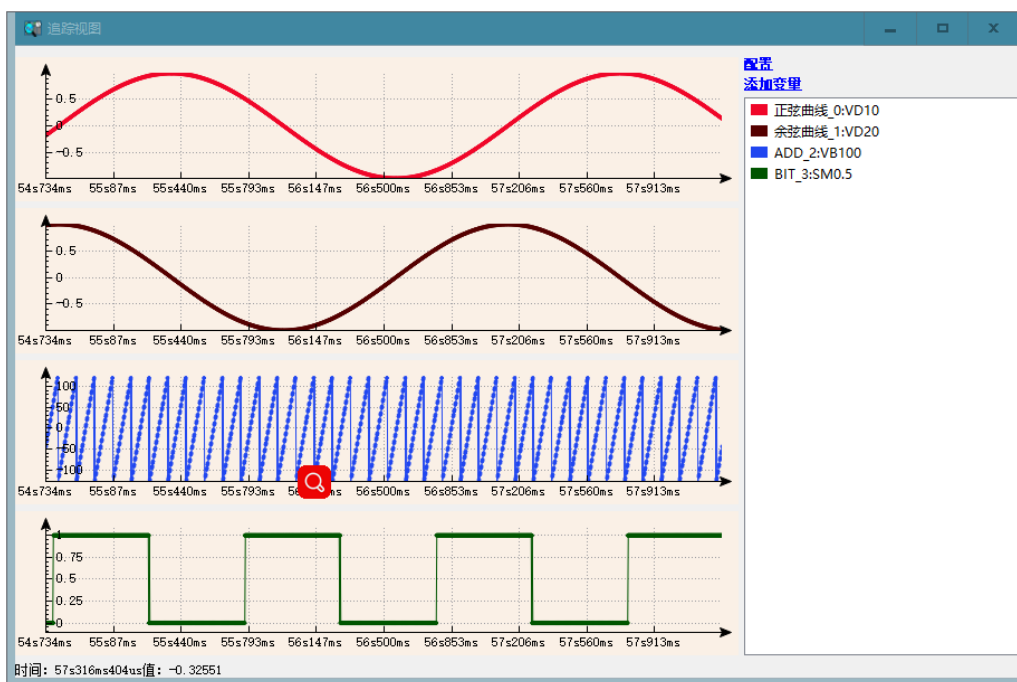
Trace 追踪功能是 MagicWorks PLC 软件的状态表控件中新增的功能，用于在一个或多个图表中配置和显示特定于应用程序的跟踪数据。

<备注> 高性能升级版 CPU 固件版本为 2.64 或以上支持 Trace 追踪功能。

在应用程序运行时，可以在 MagicWorks PLC 的“追踪视图”中查看跟踪变量的值曲线记录。要求是设置跟踪配置，将跟踪配置传输到 PLC，并开始跟踪记录。记录的数据将传输到上位机，并根据配置以图表形式显示。跟踪时可以浏览数据。

启动“追踪”功能，用户可在状态表 ->菜单 ->视图 ->追踪 或在状态表工具栏中点击图标  启动追踪视图。

如下图所示：左边是数据显示图表，右边是变量列表，目前可添加最多 8 个变量和 8 个图表。



配置对话框

对话框包括了对追踪工程的数据记录的追踪配置，点击“追踪视图”右上方的“配置”或鼠标右键是弹出菜单中的“配置”可以进入到配置对话框。

添加变量：添加一个要监控的变量，目前规格：最多可添加 8 个变量。

- 可在“追踪视图”右上角点击添加变量添加
- 可在“追踪记录”视图中右键菜单中添加
- 在图表视图中选中其中一个图表的“显示变量”结点在右键弹出菜单中添加

删除变量：在追踪视图中选中一个变量右键删除变量，弹出确认对话框，确认后删除变量，注意：变量删除后不能撤消

分配变量：在追踪视图中选中一个变量在右键弹出菜单中选中“分配到图表”，可以把变量分配到指定的图表中。在添加变量的时候会默认分配到第一个图表中。目前可最多添加 8 个图表。

添加图表：图表视图中可通过右键菜单“添加图表”可在视图中插入一个新的图表。

删除图表：图表视图中可选择要删除的图表通过右键菜单“删除图表”，并在对话框中确认后删除图表。注意：图表删除后不能撤消。

记录设置：点击追踪记录中的“追踪”可在右边进行“记录设置”

启用触发器：通过使能一个变量来记录一段时间内的数据，点击右边复选框可以使能/不使能，当使能触发器时，还需要配置触发变量，触发沿和触发后采样时间。

触发变量：当使能触发器的时候有效，被用作触发器的信号，有效的触发信号是一个变量，PLC 中的一个位地址。例如 M0.0。

触发沿：当使能触发器的时候有效，定义触发边沿检测：

上升沿：对于 BOOL 触发变量，触发在值从 FALSE 变成 TRUE 的时候发生。

下降沿：对于 BOOL 触发变量，触发发生在当值从 TRUE 变成 FALSE 的时候。

上升沿和下降沿：对于 BOOL 触发变量，触发发生在值改变的时候。

触发后采样时间：触发器在触发后继续采样指定的时间后停止采样。

采样周期：定义的 PLC 采样的时间周期，可在下拉框中选择 PLC 周期、EtherCAT 周期、脉冲周期和 1ms 中断。

变量设置：对添加的变量或已经存在的变量进行配置

变量地址：输入要监控的 PLC 中的有效地址，当地址无效时会报错，必须配置。

变量名称：当前变量的别名，可供用户更好地识别变量，可以不配置。

显示格式：共有 4 种显示格式供用户选择：有符号、无符号、浮点型和位。

曲线颜色：添加变量时会随机生成一种颜色，用户也可自行修改。

线类型：两点之间连线的类型是，可选择直线、阶梯和没有连线。

点类型：有圆点、交叉和无三种样式选择。

时间轴配置：点击图表视图中的时间轴结点可配置对时间轴进行设置。

显示方式：监控数据的过程中时间轴（X 轴）的变化方式：

自动：时间轴自动缩放

固定长度：显示一段常数长度。

固定：固定显示一段数值从最小到最大。

最小值：段的初始值。显示模式为“固定”时有效。

最大值：段的结束值。显示模式为“固定”时有效。

长度：常数段长度。

网格：时间轴方向图的网格线。

Y 轴配置

点击图表视图中的 Y 轴结点可配置对 Y 轴进行设置。

显示方式：监控数据的过程中 Y 轴的变化方式：

自动：时间轴自动缩放。

固定：固定显示一段数值从最小到最大

最小值：段的初始值。显示模式为“固定”时有效。

最大值：段的结束值。显示模式为“固定”时有效。

网格：Y 轴方向图的网格线。

描述： Y 轴被标记为“描述”。

跟踪图的预览图： 可以预览表配置好的图表，也可进行图表的颜色。


追踪视图右键菜单

配置： 可打开配置对框，如果当前处于监控状态，将会自动暂停

下载配置： 将配置好的变量下载到 PLC，如果数据有误将会报错，否则会自动开始读取数据。如果修改过变量就必须重新下载数据之后才能读：如添加、删除过变量，修改过变量的地址，显示格式等。

注意： 下载前要确保软件已边上 PLC 并选用的是“CTH300/200 Local(TCP/IP)”通信接口，否则无法边接到 PLC。

开始/暂时读数： 下载数据后，可以开始/暂停读数。

光标：  光标是一个跟踪光标，平行于 y 轴中垂直的黑线。可以用于标识光标位置的时间和对应的 y 值，如果光标所在的时间点没有对应的 Y 值，将显示离光标最近的一个点的值。

在没有跟踪光标可用时，添加一个光标到跟踪图。

当一个跟踪光标可用时，再添加一个光标到跟踪图。

当两个跟踪光标可用时，删除显示的跟踪光标。

无任何痕迹光标跟踪图： 在这种模式下，你可以通过鼠标指针轨迹图运行。集中于光标的 X 值，会以普通风格显示在状态栏中（例如，时间: 1m23s456ms Value: 1）。

一个跟踪光标的跟踪图： 在状态栏和 y 值，输出结果主要标志是跟踪光标的时间。例子：

Time:1m23s456ms。

两个跟踪光标的跟踪图： 在状态栏中，输出两次，由该两个跟踪光标（例如标记的时间间隔，Time: 1m23s456ms — Time: 1m24s456ms (Δ 1s)）。

跟踪图的用户输入： 如果一个或两个跟踪光标可用，那么你可以沿着 X 轴移动。

鼠标： 拖动跟踪光标的三角形到其它位置。在状态栏中同步更新 y 值。

键盘： 方向键左右可移动黑色跟踪光标。

显示/隐藏变量： 如果不想查看某个变量，可以在右边变量列表中选中变量右键弹出菜单中“隐藏变量”，变量就不会在图表中显示。如果某个变量已经被隐藏了又想重新查看变量的曲线，可以在右边变量列表中选中变量右键弹出菜单中“显示变量”，该变量就会重新显示到图表上。

重置视图： 根据时间轴和 Y 轴配置的显示模式,使视图恢复默认刷新状态。

多通道： 追踪视图中以第一个图表为模板为每一个变量分配一个图表并重新初始化图表。

注意： 使用的“多通道”功能后旧的图表配置数据将被删除。

单通道： 追踪视图把所有变量都放到第一个图表中并重新初始化图表。

注意： 使用的“单通道”功能后旧的图表配置数据将被删除。

保存轨迹： 将读取的数据保存到一个*.cttrace 的文件中。

装载轨迹： 从*.cttrace 文件中读装载保存的数据。

附录

A CT-MODBUS 主站和从站库的使用

A.1 CT_MODBUS 库介绍

一共有 4 个库，分别是 PORT0 口的主站、从站库，PORT1 口的主站和从站库。

CT_MODBUS 功能块主要针对西门子 MODBUS 功能块占用 CPU 大量程序空间和数据空间而提供的一组内嵌的简单易用的 MODBUS 协议库。CT_MODBUS 功能块是集成在 CPU 内部，不占用用户数据空间，作为一组库函数提供给用户使用。



提示

如需该库请前往合信网站免费下载，网址：<http://www.co-trust.com>

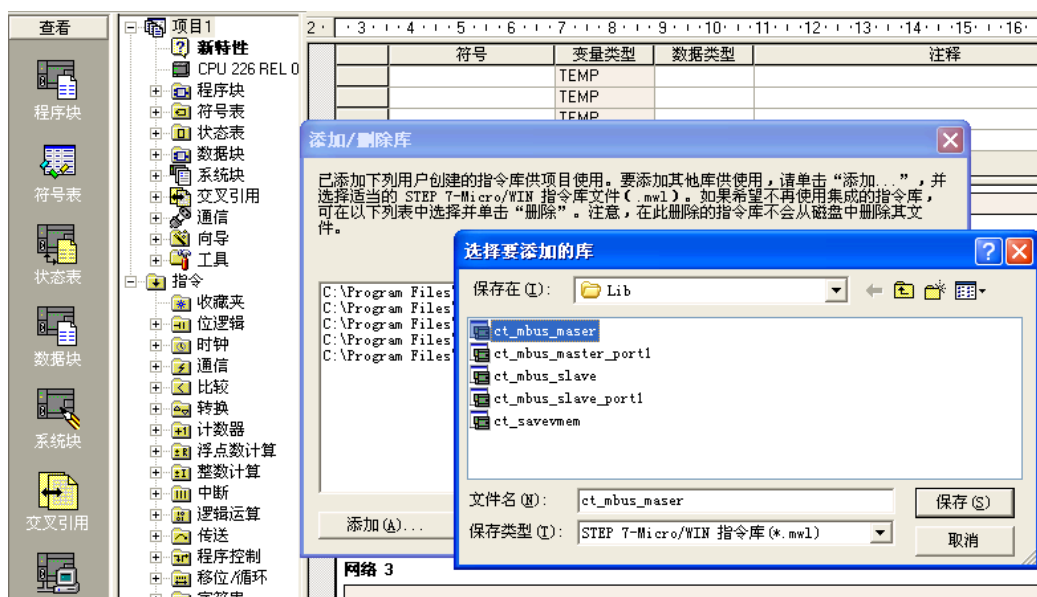
A.2 安装说明

【添加库文件】

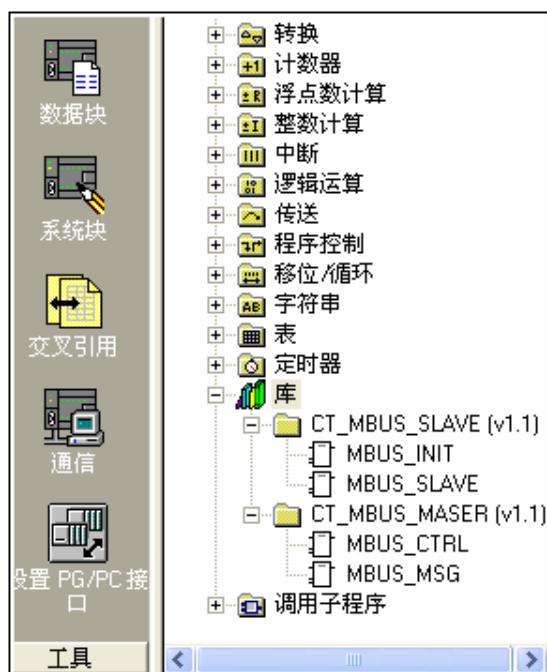
在“文件”—“添加/删除库”，找到库文件“ct_mbus_master.mwl”和“ct_mbus_slave.mwl”，如下图所示：



在存放“ct_mbus_master.mwl”和“ct_mbus_slave.mwl”文件的位置找到文件，如下图所示，单击“添加”按钮。

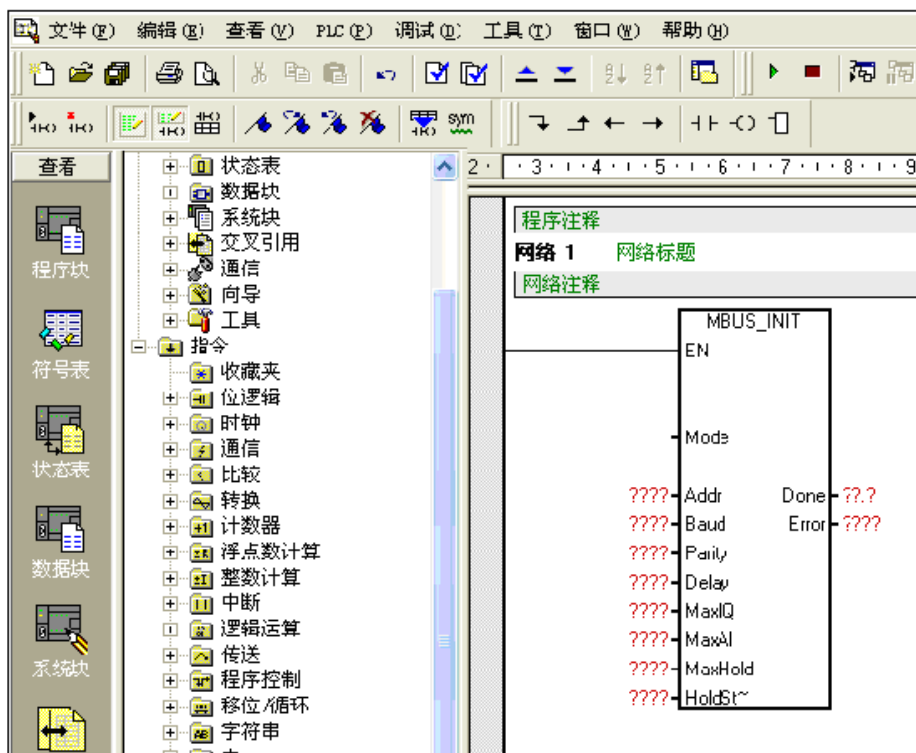


安装成功后，在目录树的“库”下可看到新增加的 CT_MBUS_MASTER 和 CT_MBUS_SLAVE 库：



【调用 CT_MODBUS 库】

单击需要添加功能块的“网络”，双击“库”下面的“MBUS_INIT”，“MBUS_SLAVE”，“MBUS_CTRL”，“MBUS_MASTER”就会在“网络”里出现相应的功能块。结果如下图所示：



A.3 CT_MODBUS 库功能说明

【Modbus 地址】

Modbus 地址通常是包含数据类型和偏移量的 5 个或 6 个字符值。第一个或前两个字符决定数据类型，最后的四个字符是符合数据类型的一个适当的值。Modbus 主站则将这个地址对应到正确的功能上。

Modbus 从站指令支持以下地址：00001 至 00128 是实际输出，对应于 Q0.0—Q15.7；10001 至 10128 是实际输入，对应于 I0.0—I15.7；30001 至 30032 是模拟输入寄存器，对应于 AIW0 至 AIW62；40001 至 4XXXX 是保持寄存器，对应于 V 内存区；

所有 Modbus 地址都是从 1 开始编号的。下表所示为 Modbus 地址与 CTH200 地址的对应关系。Modbus 从站协议允许您对 Modbus 主站可访问的输入、输出、模拟输入和保持寄存器（V 区）的数量进行限定。

Modbus地址	CTH200地址
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)

【使用 Modbus 从站协议指令】

※ CT_MODBUS 从站协议指令占用 CTH200 CPU 的资源

1) 根据使用不同的 Modbus 协议库占用自由口 0 或者 1 作为 Modbus 从站协议通讯。当 Port 0 或者 Port 1 作为 Modbus 协议通讯时，它不能再作为其它任何目的使用，包括与 MagicWorks PLC 或 SETP7-Micro/WIN 通讯，自由口通讯。MBUS_INIT 指令控制 Port 的设定是 Modbus 协议还是 PPI。

2) 与选用 Port 自由口通讯相关的所有的 SM。

3) 占用 92 个字节程序空间。

※ 在 CTH200 程序中使用 Modbus 从站协议指令遵循的步骤

1) 在您的程序中插入 MBUS_INIT 指令并且只在一个循环周期中执行该指令，MBUS_INIT 指令可用于对 Modbus 通讯参数的初始化或修改。当您插入 MBUS_INIT 指令时，几个隐藏的子程序和中断服务程序会自动地添加到您的程序中。

2) 在您的程序中只使用一个 MBUS_SLAVE 指令。该指令在每个循环周期中执行，为接收到的所有请求提供服务。

3) 用通讯电缆将 CTH200 CPU 通讯口与 Modbus 主站连接起来。

※ Modbus 从站协议指令所支持的功能

Modbus 从站协议指令支持 Modbus RTU 协议。这些指令使用 S7-200 的自由口功能，支持大部分常用 Modbus 功能。以下是所支持的 Modbus 功能：

功能	描述
1	读单个/多个线圈（实际输出）状态。功能1返回任意数量输出点的接通/断开状态（Q）
2	读单个/多个接触器（实际输入）状态。功能2返回任意数量的输入点的接通/断开状态（I）
3	读单个/多个保持寄存器。功能3返回V存储器的内容，保持寄存器在Modbus下是字类型，在一个请求中最多可读120个字
4	读单个/多个输入寄存器。功能4返回模拟输入值
5	写单个线圈（实际输出）。功能5将实际输出点设置为指定值。该输出点不是被强制，用户程序可以重写由Modbus的请求而写入的值

6	写单个保持寄存器。功能6写一个单个保持寄存器的值到S7-200的V存储区
15	写多个线圈（实际输出）。功能15写多个实际输出值到S7-200的Q映像区。起始输出点必须是一个字节的开始（如，Q0.0或Q2.0），并且要写的输出的数量是8的倍数。这是Modbus从站协议指令的限定。这些点不是被强制，用户程序可以重写由Modbus的请求而写入的值
16	写多个保持寄存器。功能16写多个保存寄存器到S7-200的V区。在一个请求中最多可写120字

※ MBUS_INIT 指令

MBUS_INIT 指令用于使能和初始化或禁止 Modbus 通讯。MBUS_INIT 指令必须无错误的执行，然后才能够使用 MBUS_SLAVE 指令。在继续执行下一条指令前，MBUS_INIT 指令必须执行完并且 Done 位被立即置位。MBUS_INIT 指令应该在每次通讯状态改变时只执行一次。因此，EN 输入端应使用边沿检测元素以脉冲触发，或只在第一个循环周期内执行一次。

参数地址	说明	类型	数值范围	备注
Mode	选择通讯协议：输入1值将Port定义为Modbus协议并使能该协议，输入值为0将Port定义为PPI并禁止Modbus协议	位		
Addr	设置本站地址	字节	1~247	
Baud	设置波特率(bps)	双字	1200、2400、4800、9600、19200、38400、57600、115200	
Parity	设置校验	字节	0—无校验 1—奇校验 2—偶校验	所有设置使用一个停止位。
Delay	通过为标准Modbus信息超时增加指定数量的毫秒，扩展标准Modbus信息结束超时条件	整型	0~32767	单位：毫秒
MaxIQ	设置可使用的I点和Q点数	整型	0~128。 数值为0则禁止对输入和输出的读写。	建议MaxIQ取值128，即允许访问所有I/Q点。
MaxAI	设置可使用的字输入寄存器（AI）的个数	整型	0~32。 为0时则禁止读模拟输入。	MaxAI的建议值： H224/H224X/H226L/H226M/H226XL为32 H226XM/H228XL 为194
MaxHold	设置可以使用的V存储区字保持寄存器的个数	整型	0~32767	单位：字
HoldStart	设置可以使用的V存储区的保持寄存器的起始地址	双字	指向V存储区的指针。	
Done	当MBUS_INIT指令完成时，Done输出接通	位		
Error	Error输出字节包含该指令的执	字节		

	行结果			
--	-----	--	--	--

※ MBUS_SLAVE 指令

MBUS_SLAVE 指令用于服务来自 Modbus 主站的请求，必须在每个循环周期都执行，以便检查和响应 Modbus 请求。当 EN 输入接通时，该指令在每一循环周期内执行。MBUS_SLAVE 指令无输入参数。

参数说明：

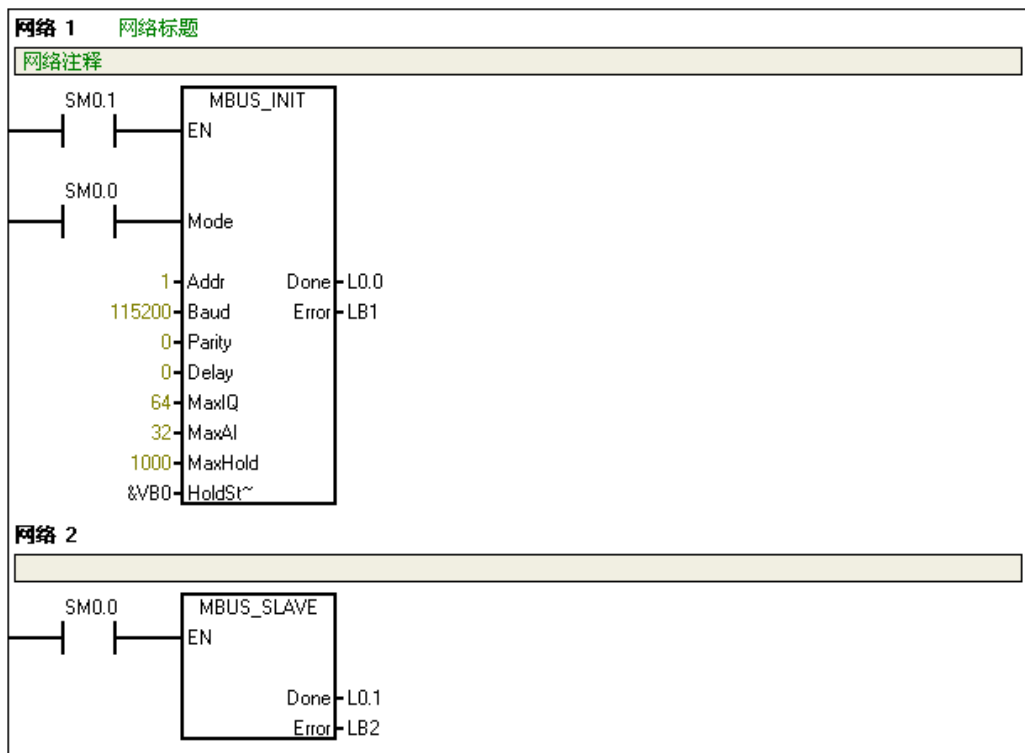
参数地址	说明	类型	数值范围	备注
Done	当 MBUS_SLAVE 指令响应 Modbus 请求时 Done 输出接通。如果没有服务的请求，Done 输出会断开	位		
Error	输出包含该指令的执行结果	字节	参见错误代码表	该输出只有 Done 接通时才有效。如果 Done 断开，错误代码不会改变。

错误代码表：

0	无错误
1	存储区范围错误
2	非法波特率或校验
3	非法从站地址
4	Modbus 参数的非法值
5	保持寄存器与 Modbus 从站符号地址重叠
6	接收校验错误
7	接收 CRC 错误
8	非法功能请求/不支持的功能
9	请求中有非法存储区地址
10	从站功能未使能

※ Modbus 从站协议指令使用实例

下面的梯形图程序就是建立了一个从站地址为 1，波特率为 115200，无校验的 MODBUS 从站：



※ MBUS_INIT 参数配置说明

Addr	设置从站地址为 1
Boud	设置通信波特率为115200
Party	设置奇偶校验为无校验
Delay	设置延迟时间为0毫秒
MaxIQ	设置最大可使用64个I点和64个Q点（即000001-0000064和010001-010064）
MaxAI	设置最大可读32路模拟量输入（即030001-030032）
MaxHold	设置最大可使用的V区字保存寄存器的个数（以字为单位）
StartHold	Modbus主站可以访问S7-200 V区保存寄存器的起始地址（如&VB0）

【使用 Modbus 主站协议指令】

※ Modbus 主站协议指令占用 CTH200 的 CPU 资源

1) 根据使用不同的 Modbus 协议库占用自由口 Port 0 或者 Port 1 作为 Modbus 从站协议通讯。当 Port 0 或者 Port 1 作为 Modbus 协议通讯时，它不能再作为其它任何目的的使用，包括与 MagicWorks PLC 或 SETP7-Micro/WIN 通讯，自由口通讯。MBUS_INIT 指令控制 Port 的设定是 Modbus 协议还是 PPI。

2) 与选用 Port 自由口通讯相关的所有的 SM。

3) 占用 119 个字节程序空间。

※ MBUS_CTRL 指令

使用 SM0.0 调用 MBUS_CTRL 指令完成主站的初始化，并启动其功能控制。

参数说明:

参数地址	说明	类型	数值范围	备注
Mode	设置通讯模式:为 1 时, 使能 Modbus 协议功能; 为 0 时恢复系统为 PPI 协议	位		
Baud	设置波特率(bps)	双字	11200, 2400, 4800, 9600, 19200, 38400, 57600, 115200	
Parity	设置校验	字节	0—无校验 1—奇校验 2—偶校验	所有设置使用一个停止位。
Timeout	主站等待从站响应的 时间, 以毫秒为单位	整型	1~32767	典型的设置值为 1000 毫秒 (1 秒)。
Done	完成位, 初始化完成, 此位会自动置 1。	位		
Error	初始化错误代码	字节	0—无错误 1—校验选择非法 2—波特率选择非法 3—模式选择非法	只有在 Done 位 为 1 时有效。

※ MBUS_MSG 指令

使用 SM0.0 调用 Modbus RTU 主站读写子程序 MBUS_MSG 指令, First 接通发送一个 Modbus 请求。同一时刻只能有一个读写功能 (即 MBUS_MSG) 使能。

参数说明:

参数地址	说明	类型	数值范围	备注
First	读写请求位	位		每一个新的读写请求必须 使用脉冲触发。
Slave	设置从站地址	字节	1~247	
RW	操作命令	字节	0~读 1~写	
Addr	选择读写的数据类 型	双字	00000 至 0xxxx—开关量 输出 10000 至 1xxxx—开关量 输入 30000 至 3xxxx—模拟量 输入 40000 至 4xxxx—保持寄 存器	
Count	通讯的数据个数 (位 或字的个数)	整型		Modbus 主站每一 个 MBUS_MSG 指令可读/写 的最大数据量为 120 个字。
DataPtr	数据指针, 如果是读	双字		

	指令，读回的数据放到这个数据区中；如果是写指令，要写出的数据放到这个数据区中			
Done	完成位，读写功能完成位	位		
Error	错误代码		参见错误代码表	只有在 Done 位为 1 时，错误代码才有效。

错误代码表:

0	无错误
1	响应校验错误
2	未用
3	接收超时（从站无响应）
4	请求参数错误
5	Modbus/自由口未使能
6	Modbus正在忙于其它请求
7	响应错误（响应不是请求的操作）
8	响应CRC校验和错误
101	从站不支持请求的功能
102	从站不支持数据地址
103	从站不支持此种数据类型
104	从站设备故障
105	从站接收了信息，但是响应被延迟
106	从站忙，拒绝了该信息
107	从站拒绝了信息
108	从站存储器奇偶错误

B CTH200 的 CPU 扩展程序空间使用

B.1 功能介绍

动态库功能块是 CTH200 系列 CPU 为扩大用户编程程序空间和增加程序保密性给用户提供的特殊功能。CTH200 系列 CPU 动态库是提前下载到 PLC 中，应用程序下载时再编译到程序中的功能独立的程序块。CTH200 系列 CPU 可加载两个大小各个 24K 的动态库("ct_lib1"和"ct_lib2")。

B.2 使用说明

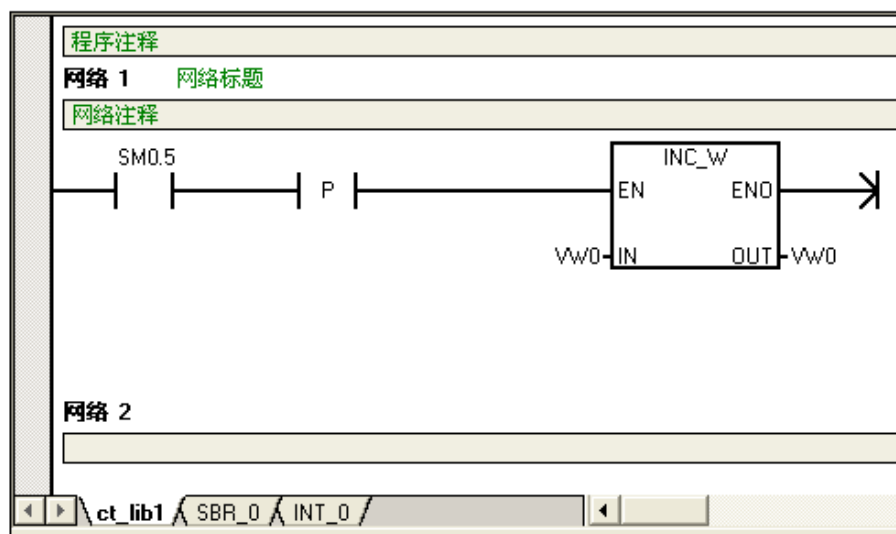
【动态库的使用范围】

CTH200 系列 CPU 最多可加载两个动态库("ct_lib1"和"ct_lib2")，不同类型的 CPU 所支持的动态库及动态库的大小如下表所示：

CPU型号	ct_lib1	ct_lib2
H226L/H226M /H224/H224X	允许4K	不支持
H224X	允许24K	不支持
H226XL/H226XM/H228XL	允许24K	
H226IM/H226IL/H226IH	允许64K	

【创建动态库】

在工程中，创建所有要作为动态库的程序块，将主程序块的名称命名为 `ct_lib1` 或 `ct_lib2`，下载到 PLC 中，在 PLC 中就生成了库函数包括工程中的所有子程序块的动态连接库。



【下载动态库】

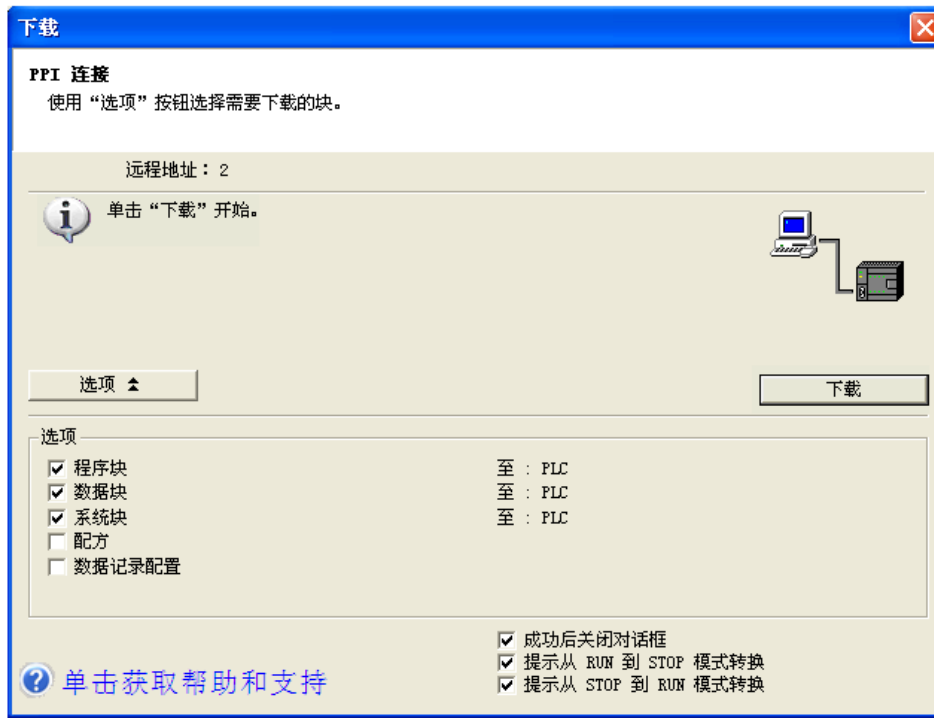
将编辑好的动态库程序下到 PLC 中（如下图所示），在 PLC 中就生成了库函数包括工程中的所有子程序块的动态连接库。

每次成功下载了新的动态库之后，PLC 中原来的库和程序块完全被清除。PLC 中生成了名为 `ct_lib1` 的动态库。



提示

在下载动态库时确保只下载程序块。



同样的流程适用于下载另一个名为 `ct_lib2` 的动态库到 PLC 中。

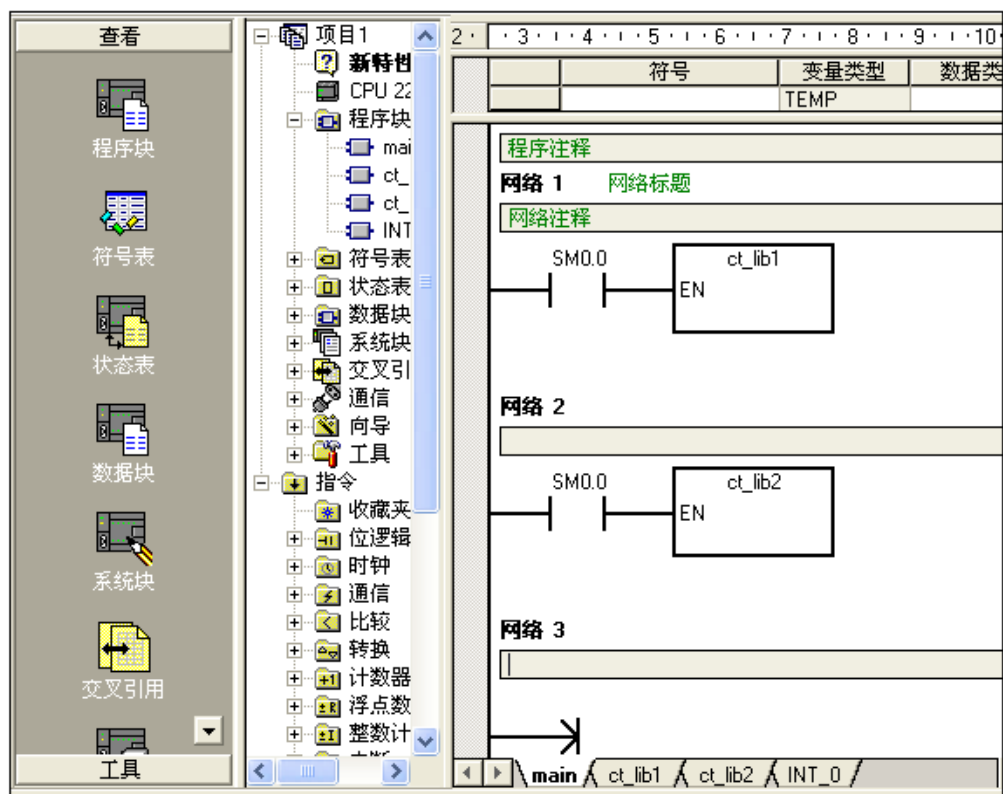
【使用动态库】

在工程中，先创建已经下载到 PLC 中的动态库同名的空的子程序块，如 `ct_lib1` 或 `ct_lib2`，在工程中再调用这些空的子程序块，程序下载到 PLC 后，此工程中的空程序块在 CPU 内部会被替换成原下载的与之同名的库函数，运行 CPU 时执行的就是相应动态库里的程序。



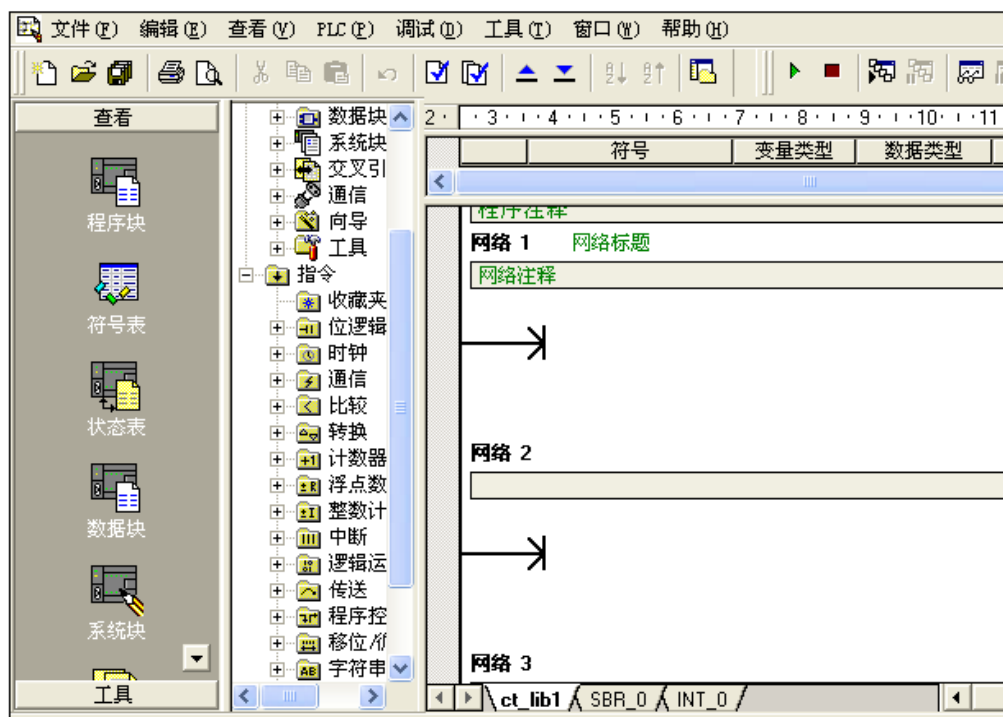
提示

使用时最好先加载动态库，然后再加载使用动态库的程序。



【清除动态库】

下载新的动态库时，PLC 中原有的动态库被清除。下载一个主程序名称为 ct_lib1 或 ct_lib2 的空程序块工程到 PLC 中，PLC 中的相应的动态库就会被完全清除。



C CTH200 系列 PLC 扩展 100K 数据块库的使用

C.1 功能介绍

扩展数据空间是 CTH200 系列 CPU 在现有存储区域外扩展的 100K 数据存储空间，可供用户访问，这块存储空间的数据保持特性等同于 V 内存数据空间，在 CPU 断电情况下通过超级电容来保持数据，数据最长保持时间可达 100 个小时，数据存储量为 100K 字节。我司专门为这块数据的访问提供了库指令，用户可以将 COTRUST 提供的库添加到 MicroWin 中，通过库中提供的读写指令访问这块数据空间，实现扩展数据空间和其他数据空间的数据交换。



提示

如需该库请前往合信官网免费下载，网址：<http://www.co-trust.com>

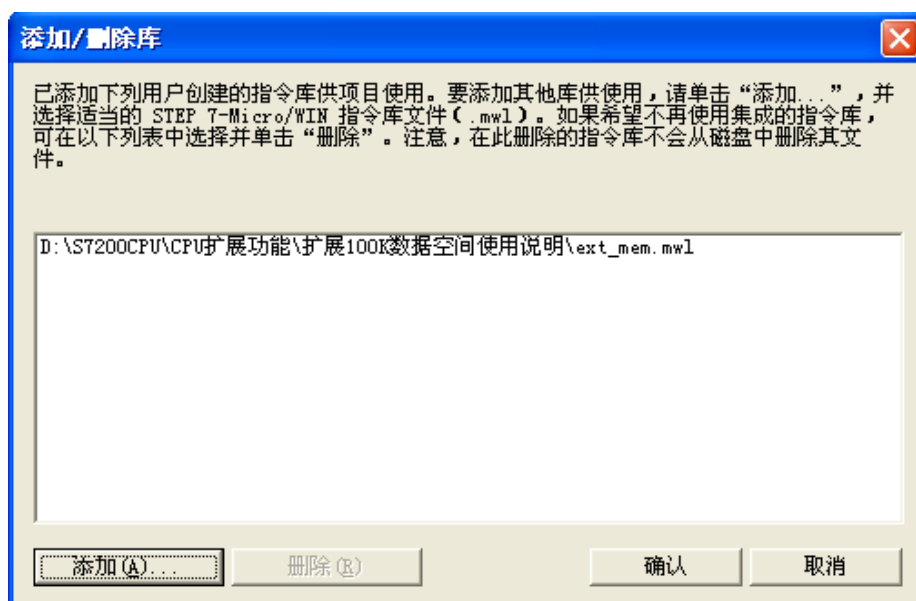
C.2 安装说明

【添加库文件】

在“文件”----“添加/删除库”，找到库文件“pid_t.mwl”，如下图所示。

文件 (F)	编辑 (E)	查看 (V)	PLC (P)
新建 (N)		Ctrl+N	
打开 (O)...		Ctrl+O	
关闭 (C)			
保存 (S)		Ctrl+S	
另存为 (A)...			
设置密码 (M)...			
导入 (I)...			
导出 (E)...			
上载 (U)...		Ctrl+U	
下载 (D)...		Ctrl+D	
新建库 (L)...			
添加/删除库 (R)...			
库存储区 (M)...			
页面设置 (T)...			
打印预览 (V)			
打印 (P)...		Ctrl+P	
退出 (X)			

在存放 ext_mem.mwl 文件的位置找到此文件，如下图所示，点“添加”按钮。

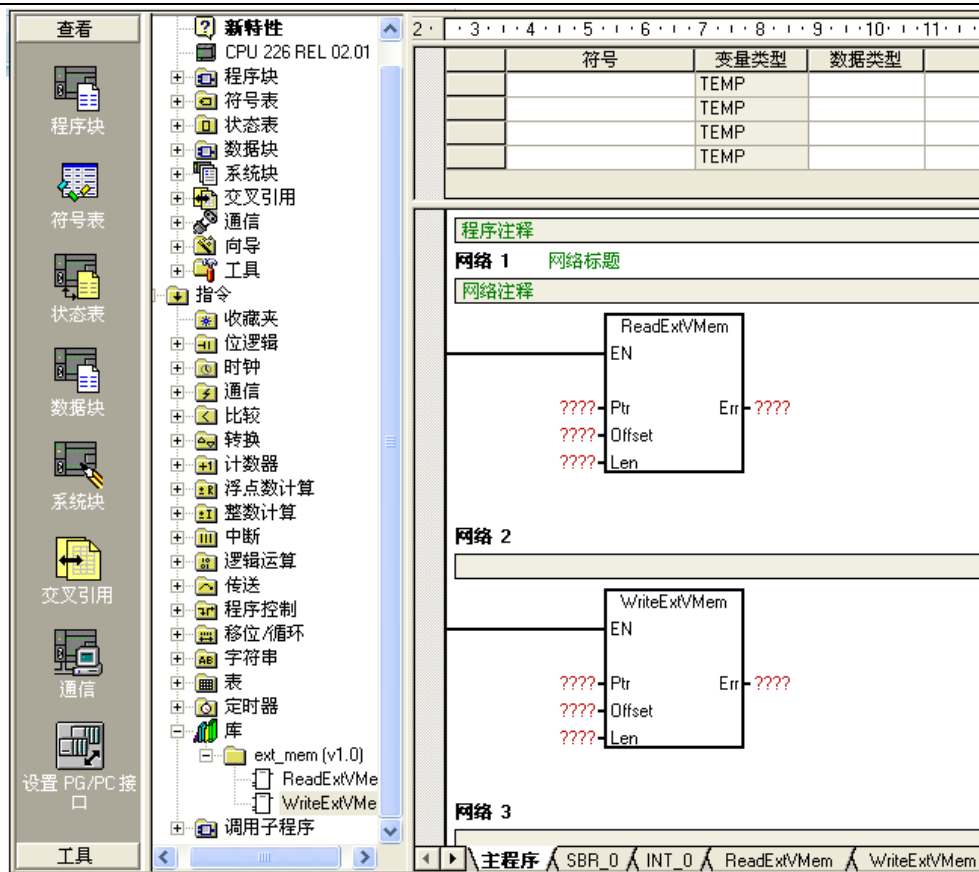


安装成功后，在目录树的“库”下可以看到新增加的 ext_mem 的库：



【调用 ext_mem 库】

单击要添加功能块的“网络”，双击“库”下面的“ReadExtVMem”和“WriteExtVMem”，就会在“网络”里出现相应的功能块。结果如下图所示：



C.3 Ext_mem 库功能说明

【使用读函数 ReadExtVMem 从扩展数据空间读取数据】

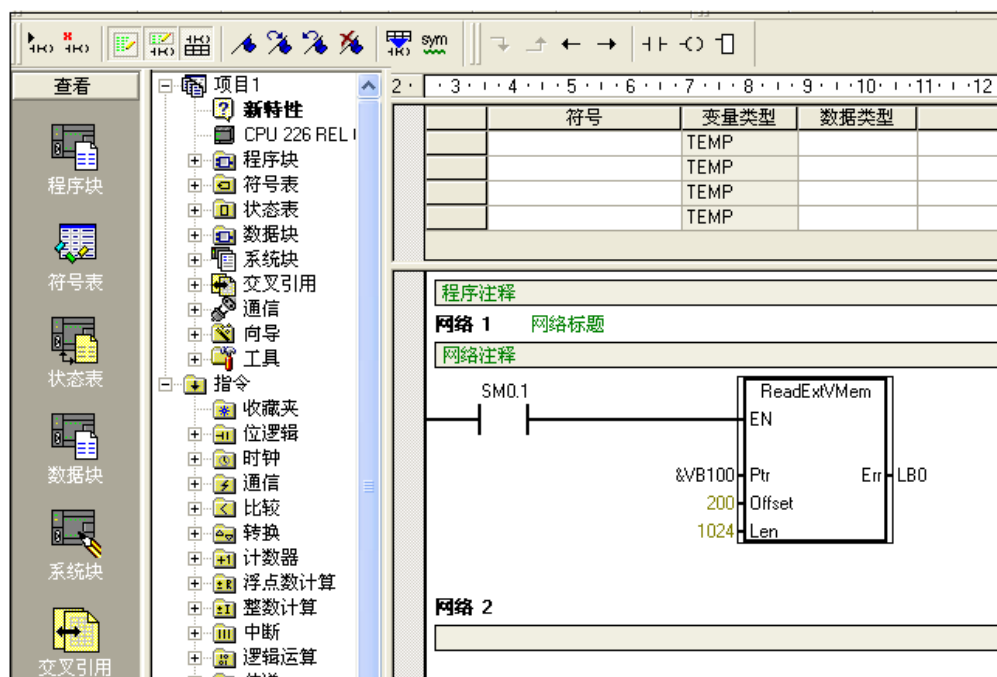
参数地址	说明	类型	数值范围	备注
Ptr	表示要读到的内存地址指针	DWORD		如: &VB0, &IB0
Offset	表示读取扩展内存中的起始偏移地址	DWORD	0-102399	
Len	表示读取内存长度(字节数)	DWORD		
Err	返回值表示读取是否成功	BYTE		0 表示读取成功, 其它表示失败。

【使用写函数 WriteExtVMem 向扩展数据空间写入数据】

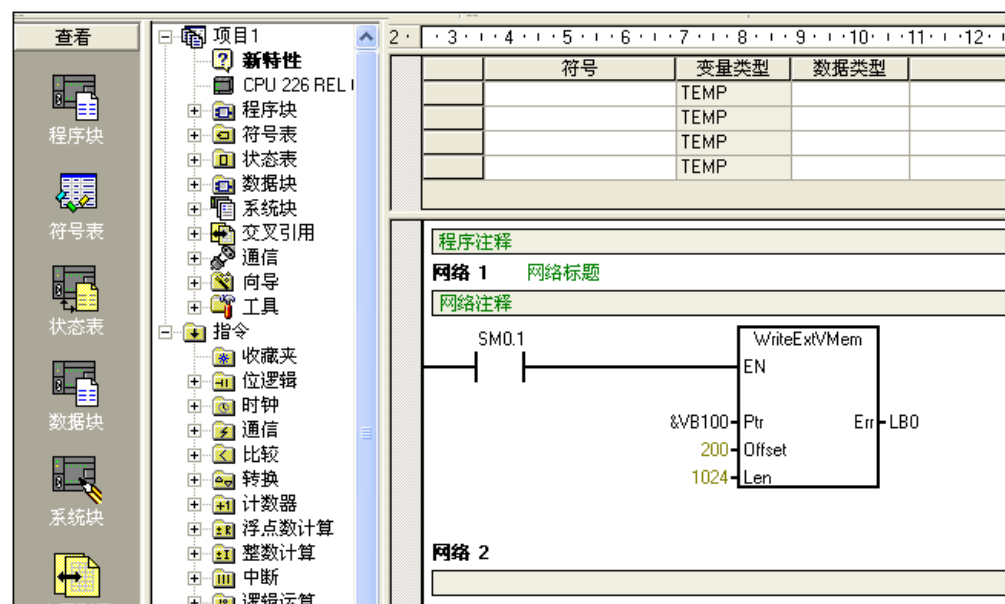
参数地址	说明	类型	数值范围	备注
Ptr	表示要写到扩展数据空间的源数据内存地址指针	DWORD		如: &VB0, &IB0
Offset	表示写到扩展内存中的起始偏移地址	DWORD	0-102399	
Len	表示写入内存长度(字节数)	DWORD		
Err	返回值表示写入是否成功	BYTE		0 表示写入成功, 其它表示失败。

C.4 应用例子

1、把扩展内存中从偏移量 200 开始的 1024 个字节读到 VB100 开始的内存中



2、把 VB100 开始的 1024 个字节写到扩展内存中从偏移量 200 开始的内存中



D CTH200 系列 PLC PID 控制库的使用

D.1 CPU 嵌入式 PID_T 库的使用

【功能介绍】

PID_T 功能块集成在 CPU 内部，不占用用户数据空间，作为一个库函数提供给用户使用。PID_T 主要针对温度控制的智能 PID 功能，带有自整定、自适应功能，用户无需复杂编程，只需调用和设置一些简单的参数就可以使用，温度控制准确。



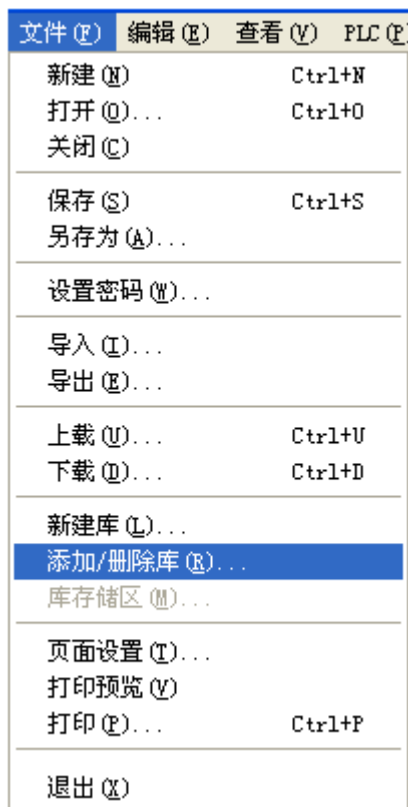
提示

如需该库请前往合信官网免费下载，网址：<http://www.co-trust.com>

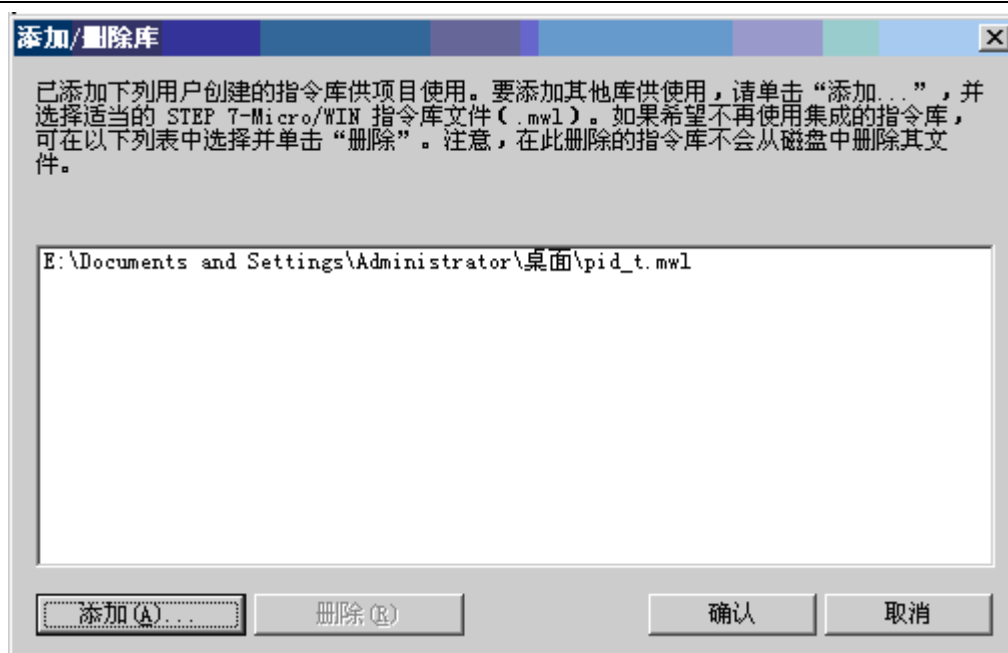
【安装说明】

※ 添加库文件

在“文件”----“添加/删除库”，找到库文件“pid_t.mwl”，如下图所示。



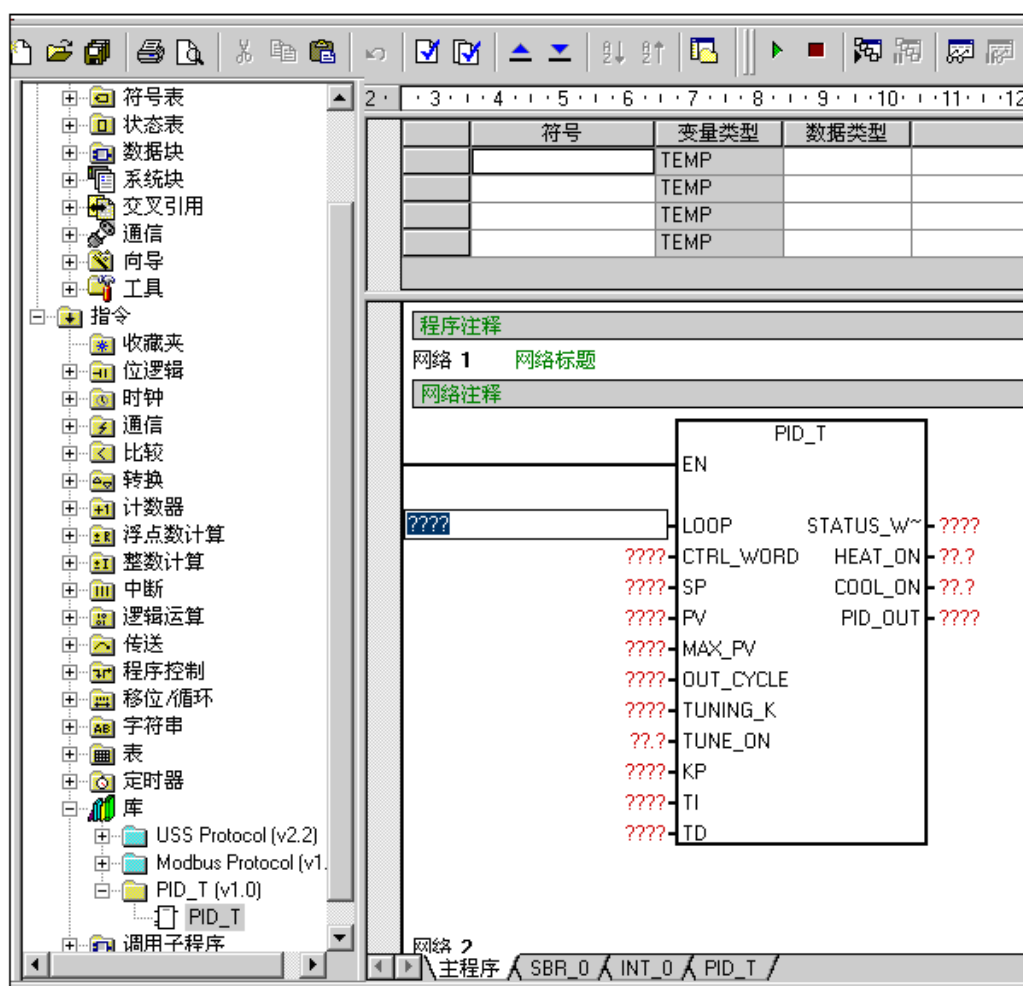
在存放 pid_t.mwl 文件的位置找到此文件，如下图所示，点击“添加”按钮。



安装成功后，在目录树的“库”下可以看到新增加的 PID_T 的库：

※ 调用 PID_T 库

点击要添加功能块的“网络”，双击“库”下面的“PID_T”，就会在“网络”里出现相应的功能块。结果如下图所示：



【PID_T 指令说明】

地址参数说明

参数地址	说明	类型	数值范围	备注
LOOP	属于第几个回路PID, 不能重复, 从0开始。	字, 常数或变量	0-63	表示该控制回路ID。
CTRL_WORD	控制字(控制PID运行)	字, 常数或变量		常用控制字: 1) 16#03(只有加热输出, 带自适应功能) 2) 16#07(加热冷却输出, 带自适应功能)
SP	设定值	字, 常数或变量	-32768-32767	单位: 0.1°C
PV	测量值(反馈值)	字, 变量	-32768-32767	单位: 0.1°C
MAX_PV	测量值的最大值	字, 常数或变量	-32768-32767	单位: 0.1°C
OUT_CYCLE	脉冲输出周期	字, 常数或变量	1-255	单位: 秒
TUNING_K	自整定系数	双字, 浮点数	0.5-2.0	0.5:要求系统控制超调量小。 1.0:正常响应。 2.0:要求系统控制响应快, 超调量大。
TUNING_ON	启动自整定	位, 变量		自整定结束后自动复位。
Kp	比例系数	字变量		如果字变量赋值为常数, 则不能执行自整定功能。
Ti	积分时间	字变量	1-3600	单位: 秒 如果字变量赋值为常数, 则不能执行自整定功能。
Td	微分时间	字变量	0-3600	单位: 秒 如果字变量赋值为常数, 则不能执行自整定功能
STATUS_WORD	状态字	字, 变量		运行状态及报警状态。
HEAT_ON	加热输出	位		
COOL_ON	冷却输出	位		
PID_OUT	PID模拟输出	字, 整数, 变量		定义为仅加热输出时, 输出范围为: 0-32000 带冷却输出时: -32000-32000

【控制字、状态字位地址】

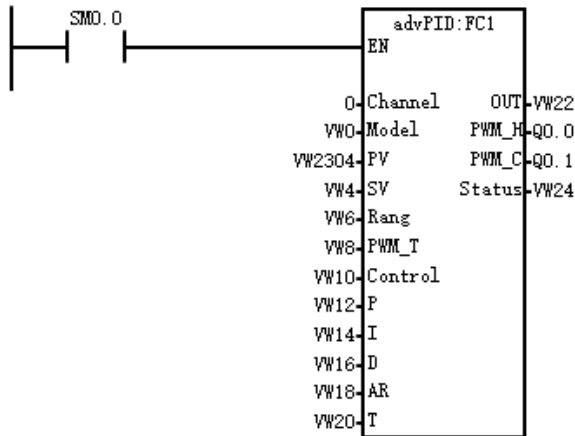
控制字的位地址意义如下:

控制字位	设置	备注
0	0	PID 停止
	1	PID 运行
1	0	积分一直起作用，比例系数 K_p 不自动调整
	1	积分分离及比例系数自动调整
2	0	PID 单极输出
	1	PID 双极输出
3	0	保留
	1	保留
4	0	积分起作用
	1	积分不起作用
5	0	微分起作用
	1	微分不起作用
6		保留
7		保留

状态字位地址的意义：

状态字位	值	备注
0	0	无断线故障
	1	断线故障
1	0	自整定未进行
	1	正在自整定
2	0	无自整定故障
	1	自整定故障
3	0	不加热
	1	正在加热
4	0	不冷却
	1	正在冷却
5	0	PID 停止状态
	1	PID 运行状态
6		保留
7		保留

【advPID 指令说明】



参数	输入输出属性	数据类型	描述	默认值
Channel	IN	WORD	表示 PID 通道编号，每个通道的计算相互独立，在控制多个对象时，调用多个 PID 块，应使用不同的通道编号来控制。 取值范围：0~7	0
Model	IN	WORD	表示 PID 算法模式选择，可根据被控对象选择相应的模式，PID 块会给出相应的 PID 参数。 0：默认模式，适用于绝大多数控制对象。 1：开启 PID 在线自校正的模式，在该模式下，PID 算法会根据调节的效果对 P，I 参数进行实时校正，控制对象和设定值（SV）的条件会不断变化，该控制系统可自动计算 PID 并进行控制。虽然不要求高控制性能，但由于控制对象的条件频繁变化，当自动调整无法运行时该控制系统将非常有效。 注意：在此模式下不要手动调节 PID 参数，否则可能引起无法预知的影响。 10：控制对象为 TEC（半导体制冷器）时专用的算法模式。 取值范围：0~32000	0
PV	IN	WORD	PV（Process Variable）表示被控对象的值。比如被控对象的温度，压力，流量，等。最后一位为小数，例 4000 表示 400.0。 注意：值范围不能超过 Rang 表示的范围 取值范围：-20000~20000	0
SV	IN	WORD	表示对被控对象的期望值，设定值。最后一位为小数，例 4000 表示 400.0。 注意：值范围不能超过 Rang 表示的范围 取值范围：-20000~20000	0
Rang	IN	WORD	表示被控对象的动作范围，比如温度可以取传感器的量程换算，如-200-400℃换算为 6000，电压电流也可以先比例换算为合适的范围，如 0-10V 换算为 10000。	4000

			取值范围：1~20000																					
PWM_T	IN	WORD	<p>设定 PWM 输出的周期，单位：S。</p> <p>注意：为了减少外围输出元件寿命损耗，建议对于 SSR（固态继电器）输出取值一般不小于 2，对于机械继电器取值一般不小于 30。</p> <p>取值范围：0~32000</p>	0																				
Control	IN_OUT	WORD	<p>PID 算法的控制设定，以字的 16 个位进行功能选择</p> <p>注：如果没有选择控制极性，默认为单极反向控制。</p> <p>DEC(十进制)操作加上表格中相应数字即相应位置 1</p> <table border="1"> <tr> <td>Bit 0 DEC:+1</td> <td>1: 启动 PID 计算 0: 关闭 PID, PID 将停止运行, 输出为 0</td> </tr> <tr> <td>Bit 1 DEC:+2</td> <td>1: 启动 PID 自整定算法, 整定结束后会自动复位 0: 关闭 PID 自整定算法</td> </tr> <tr> <td>Bit 2 DEC:+4</td> <td>1: 输出为单极反向控制, PWM 仅热端输出, 即 SV 大于 PV 时, 输出增加, 用于加热控制, 输出范围为 0 - 32000 0: 未选择</td> </tr> <tr> <td>Bit 3 DEC:+8</td> <td>1: 输出为单极正向控制, PWM 仅热端输出, 即 SV 大于 PV 时, 输出减少, 用于制冷控制, 输出范围为 0 - 32000 0: 未选择</td> </tr> <tr> <td>Bit 4 DEC:+16</td> <td>1: 输出为对偶控制, PWM 热端反向, PWM 冷端正向, 即 SV 大于 PV 时, 输出增加, PWM 端极性相反, 输出范围为 -32000~32000 0: 未选择</td> </tr> <tr> <td>Bit 5 DEC:+32</td> <td>1: 输出为对偶控制, PWM 热端反向, PWM 冷端反向, 即 SV 大于 PV 时, 输出增加, PWM 端极性相同, 输出范围为 -32000~32000 0: 未选择 (暂未实现)</td> </tr> <tr> <td>Bit 6 DEC:+64</td> <td>1: 输出为对偶控制, PWM 热端正向, PWM 冷端正向, 即 SV 大于 PV 时, 输出增加, PWM 端极性相同, 输出范围为 -32000~32000 0: 未选择 (暂未实现)</td> </tr> <tr> <td>Bit 7 DEC: +128</td> <td>1: 输出为对偶控制, PWM 热端正向, PWM 冷端反向, 即 SV 大于 PV 时, 输出增加, PWM 端极性相反, 输出范围为 -32000~32000 0: 未选择 (暂未实现)</td> </tr> <tr> <td>Bit 15 DEC: +32768</td> <td>1: PID 参数复位 0: 无</td> </tr> <tr> <td>Bit 0 DEC:+1</td> <td>1: 启动 PID 计算 0: 关闭 PID, PID 将停止运行, 输出为 0</td> </tr> </table>	Bit 0 DEC:+1	1: 启动 PID 计算 0: 关闭 PID, PID 将停止运行, 输出为 0	Bit 1 DEC:+2	1: 启动 PID 自整定算法, 整定结束后会自动复位 0: 关闭 PID 自整定算法	Bit 2 DEC:+4	1: 输出为单极反向控制, PWM 仅热端输出, 即 SV 大于 PV 时, 输出增加, 用于加热控制, 输出范围为 0 - 32000 0: 未选择	Bit 3 DEC:+8	1: 输出为单极正向控制, PWM 仅热端输出, 即 SV 大于 PV 时, 输出减少, 用于制冷控制, 输出范围为 0 - 32000 0: 未选择	Bit 4 DEC:+16	1: 输出为对偶控制, PWM 热端反向, PWM 冷端正向, 即 SV 大于 PV 时, 输出增加, PWM 端极性相反, 输出范围为 -32000~32000 0: 未选择	Bit 5 DEC:+32	1: 输出为对偶控制, PWM 热端反向, PWM 冷端反向, 即 SV 大于 PV 时, 输出增加, PWM 端极性相同, 输出范围为 -32000~32000 0: 未选择 (暂未实现)	Bit 6 DEC:+64	1: 输出为对偶控制, PWM 热端正向, PWM 冷端正向, 即 SV 大于 PV 时, 输出增加, PWM 端极性相同, 输出范围为 -32000~32000 0: 未选择 (暂未实现)	Bit 7 DEC: +128	1: 输出为对偶控制, PWM 热端正向, PWM 冷端反向, 即 SV 大于 PV 时, 输出增加, PWM 端极性相反, 输出范围为 -32000~32000 0: 未选择 (暂未实现)	Bit 15 DEC: +32768	1: PID 参数复位 0: 无	Bit 0 DEC:+1	1: 启动 PID 计算 0: 关闭 PID, PID 将停止运行, 输出为 0	0
			Bit 0 DEC:+1	1: 启动 PID 计算 0: 关闭 PID, PID 将停止运行, 输出为 0																				
			Bit 1 DEC:+2	1: 启动 PID 自整定算法, 整定结束后会自动复位 0: 关闭 PID 自整定算法																				
			Bit 2 DEC:+4	1: 输出为单极反向控制, PWM 仅热端输出, 即 SV 大于 PV 时, 输出增加, 用于加热控制, 输出范围为 0 - 32000 0: 未选择																				
			Bit 3 DEC:+8	1: 输出为单极正向控制, PWM 仅热端输出, 即 SV 大于 PV 时, 输出减少, 用于制冷控制, 输出范围为 0 - 32000 0: 未选择																				
			Bit 4 DEC:+16	1: 输出为对偶控制, PWM 热端反向, PWM 冷端正向, 即 SV 大于 PV 时, 输出增加, PWM 端极性相反, 输出范围为 -32000~32000 0: 未选择																				
			Bit 5 DEC:+32	1: 输出为对偶控制, PWM 热端反向, PWM 冷端反向, 即 SV 大于 PV 时, 输出增加, PWM 端极性相同, 输出范围为 -32000~32000 0: 未选择 (暂未实现)																				
			Bit 6 DEC:+64	1: 输出为对偶控制, PWM 热端正向, PWM 冷端正向, 即 SV 大于 PV 时, 输出增加, PWM 端极性相同, 输出范围为 -32000~32000 0: 未选择 (暂未实现)																				
			Bit 7 DEC: +128	1: 输出为对偶控制, PWM 热端正向, PWM 冷端反向, 即 SV 大于 PV 时, 输出增加, PWM 端极性相反, 输出范围为 -32000~32000 0: 未选择 (暂未实现)																				
			Bit 15 DEC: +32768	1: PID 参数复位 0: 无																				
Bit 0 DEC:+1	1: 启动 PID 计算 0: 关闭 PID, PID 将停止运行, 输出为 0																							
P	IN_OUT	WORD	<p>比例带, 设定 PID 参数的比例范围, 有效范围为 0~999%, 最后一位为小数</p> <p>---取值范围: 0~9999</p>	0																				
I	IN_OUT	WORD	<p>积分时间, 有效范围为 0 - 3200 秒, 最后一位为小数</p> <p>注: 当值为 0 时, 取消积分作用</p> <p>---取值范围: 0~32000</p>	0																				
D	IN_OUT	WORD	<p>微分时间, 有效范围为 0~3200 秒, 最后一位为小数</p> <p>注: 当值为 0 时, 取消微分作用</p>	0																				

			---取值范围: 0~32000		
AR	IN_OUT	WORD	抗积分饱和, 设定进行积分工作的范围, 有效范围为 0 - 100%。 最后一位为小数 ---取值范围: 0~1000	1000	
T		WORD	PID 计算周期时间, 有效范围为 1~100 秒, 最后一位为小数。 取值范围: 1~1000	1	
OUT	OUT	WORD	PID 的输出 范围: 单极性 0~32000, 双极性 -32000~32000	0	
PWM_H	OUT	BOOL	PWM 热端输出, 1 为 ON, 0 为 OFF	0	
PWM_C	OUT	BOOL	PWM 冷端输出, 1 为 ON, 0 为 OFF		
Status	OUT	WORD	当前 PID 的运行状态字, 以相应位来表示状态	0	
			状态位		描述
			Bit 0		1: PID 正在运行 0: PID 停止运行
			Bit 1		1: PID 自整定正在运行 0: PID 自整定未运行
			Bit 2		1: PID 自适应正在运行 0: PID 自适应未运行
			Bit 3		1: PWM 热端输出 ON 0: PWM 热端输出 OFF
			Bit 4		1: PWM 冷端输出 ON 0: PWM 冷端输出 OFF
			Bit 14		1: PID 计算故障 0: 正常状态
Bit 15	1: PID 自整定失败 0: 正常状态				

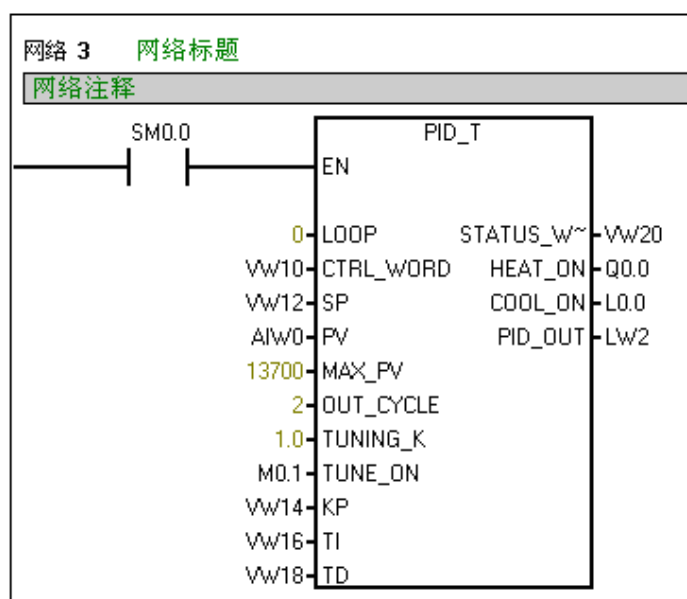
【应用举例】

系统需求

在这个例子里, 我们以一个温区为例, 系统及 I/O 配置表如下:

系统配置	CPU H226XL + 231-7PD32	温度采集采用四路热电偶模块
控制要求	1、只有加热输出, 没有冷却输出 2、要求自整定参数 3、K型热电偶	
I/O点分布		
Q0.0	加热输出	
AIW0	温度输入	K型热电偶
M0.0	PID运行/停止位	
M1.0	自整定启动位	

应用程序



程序说明

PID_T 的参数说明如下：

参数	地址或数值	说明	备注
LOOP	0	此为第一个回路，所以是0	
CTRL_WORD	VW10		
SP	VW12		
PV	AIW0		
MAX_PV	13700	因为K型热电偶的最大输入为13700	
OUT_CYCLE	2	2秒，这是脉冲输出周期	
TUNING_K	1.0		
TUNING_ON	M0.1	置1时开始整定，整定结束后复位	
Kp	VW14	比例系数，整定结束后，整定值自动写到此变量，用户还可以自行调整	
Ti	VW16	积分时间，整定结束后，整定值自动写到此变量，用户还可以自行调整	
Td	VW18	微分时间，整定结束后，整定值自动写到此变量，用户还可以自行调整	
STATUS_WORD	VW20	状态字	
HEAT_ON	Q0.0	加热输出点	
COOL_ON	L0.0	因为没有用到，所以用了一个局部变量	
PID_OUT	LW2	因为没有用到，所以用了一个局部变量	

D.2 CTH200 系列 PLC 热电偶型温度 PID 模块控制库使用

【功能介绍】

“PID_setting”功能库是专门为热电偶型的 PID 扩展模块（如：SM231-7TD、SM231-7TF）提供参数设定的。热电偶 PID 模块内部集成 PID 算法，用户无需复杂编程，只需调用“PID_setting”库给模块设置一些简单的参数就可以使用，温度控制准确。



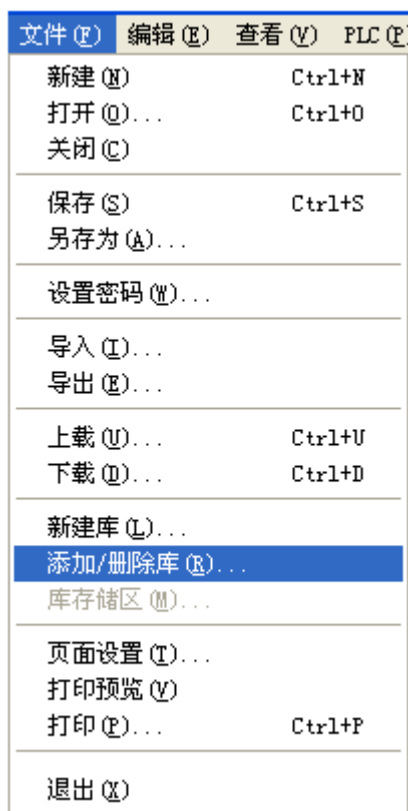
提示

1. 如需该库，请前往 <http://www.co-trust.com> 网站免费下载。
2. 此库适用 CTH2 231-7TD32、CTH2 231-7TF32 模块；
3. SM231-7TD32 和 SM231-7TF32 使用时会占用一部分 V 存储区，请编程人员编程时不要使用这些 V 存储区。
4. 西门子 CPU222 因数据空间限制不能使用此库。

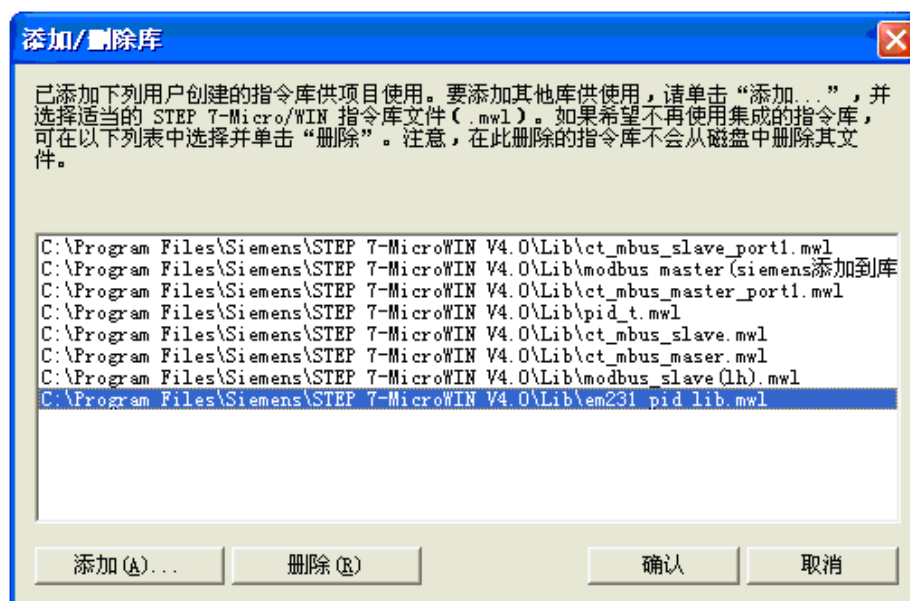
【安装说明】

※ 添加库文件

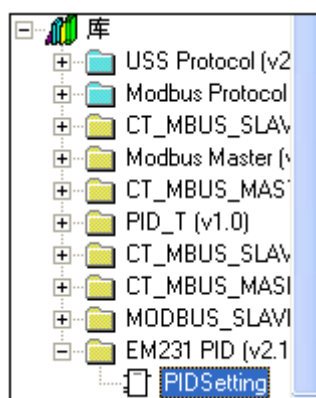
在“文件”----“添加/删除库”，找到库文件“sm231 pid lib.mwl”，如下图所示。



在存放“sm231 pid lib.mwl”文件的位置找到此文件，如下图所示，点“添加”按钮。

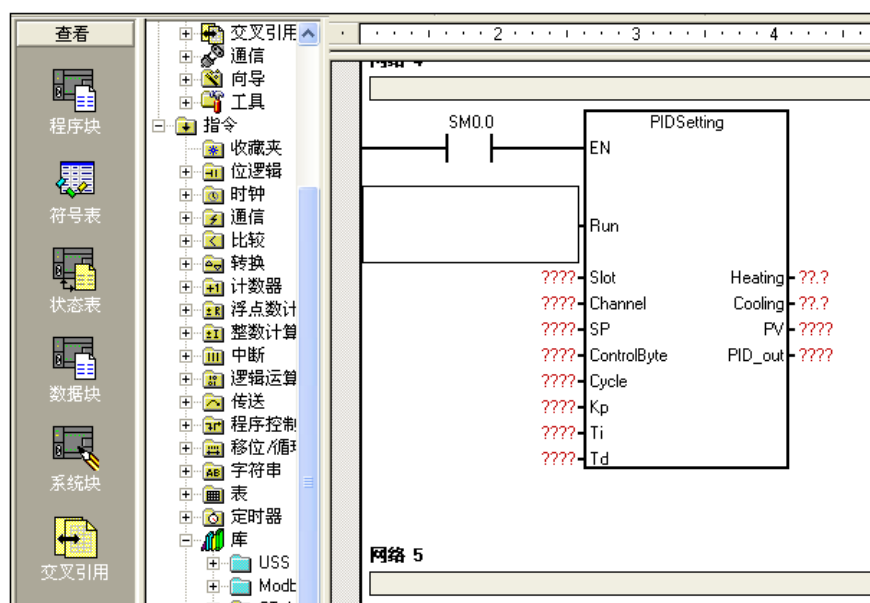


安装成功后，在目录树的“库”下可以看到新增加的 PID_setting 的库：



※ 调用 PID_Setting 库

单击要添加功能块的“网络”，双击“库”下面的“PID_Setting”，就会在“网络”里出现相应的功能块。结果如下图所示：



※ PID_setting 库功能说明

地址参数说明

参数地址	说明	类型	数值范围	备注
Run	运行	位	0 或 1	
Slot	槽号从 0 开始	字, 常数或变量	0~6	
Channel	通道号	字, 常数或变量	0~7	
SP	设定值	字, 常数或变量	-2000~32767	单位: 0.1°C
CTRLByte	控制字节, 控制 PID 运行	常数或变量		常用控制字节: 1.16#03(只有加热输出, 带自适应功能) 2.16#07(加热冷却输出, 带自适应功能)
Cycle	脉冲输出周期	字, 常数或变量	1~255	单位: 秒
Kp	比例系数	字, 整数, 常数或变量		
Ti	积分时间	字, 整数, 常数或变量	1~3600	单位: 秒
Td	微分时间	字, 整数, 常数或变量	0~3600	单位: 秒
Heating	加热输出	位		
Cooling	冷却输出	位		
PV	测量值 (反馈值)	字, 变量	-2000~32767	单位: 0.1°C
PID_out	PID 模拟输出	字, 整数, 变量	定义为仅加热输出时, 输出范围: 0~32000, 带冷却输出时: -32000~32000	

※ 控制位地址

控制字的位地址意义如下:

控制字位	设置	备注
0	0	PID 停止
	1	PID 运行
1	0	积分一直起作用, 比例系数 Kp 不自动调整
	1	积分分离及比例系数自动调整
2	0	PID 单极输出
	1	PID 双极输出
3	0	保留
	1	保留
4	0	积分起作用
	1	积分不起作用
5	0	微分起作用
	1	微分不起作用
6		保留
7		保留

※ PID 地址与参数配置

• PID 地址计算公式

地址名称	计算公式	备注
PID 参数地址	$A=(2048+S*256)+16*C$	S 为模块所在的槽号（范围：0~6） C 为通道号
PID 正向脉冲输出地址	$X=(2048+S*256)+12$	
PID 负向脉冲输出地址	$Y=(2048+S*256)+13$	

• PID 参数输出部分（模块到 CPU）

内容	地址	数值设置范围	实际对应数值
实际温度	VW A	0~13000	0~1300 度
状态字	VW A+2		
PID 模拟量输出	VW A+4	-32000~32000	

• PID 参数输入部分(CPU 到模块)

内容	地址	数值设置范围	实际对应数值
设定温度	VW A+128	0~13000	0~1300 度
控制字节		VB A+130 位等于 0 时	VB A+130 位等于 1 时
	V(A+130).0	PID 不运行, 没输出	PID 运行
	V(A+130).1	积分一直起作用, 比例系数 Kp 不自动调整	积分分离及比例系数自动调整
	V(A+130).2	PID 单极输出, 0~32000	PID 双极输出, -32000-32000, 具有加热和冷却功能
	V(A+130).3	未使用	
	V(A+130).4	积分起作用	积分不起作用
	V(A+130).5	微分起作用	微分不起作用
V(A+130).6	实际温度值滤波, 抗干扰能力更强	实际温度值不滤波	
PID 脉冲输出周期设定	VW A+132	1~255	1~255 秒
Kp(比例系数)	VW A+134	0~9999	0~999.9
Ti(积分时间)	VW A+136	0~3600	0~3600 秒
Td(微分时间)	VW A+138	0~3600	0~3600 秒

• 加热脉冲输出地址

0 通道脉冲输出	V X.0
1 通道脉冲输出	V X.1
2 通道脉冲输出	V X.2
3 通道脉冲输出	V X.3
4 通道脉冲输出	V X.4
5 通道脉冲输出	V X.5
6 通道脉冲输出	V X.6
7 通道脉冲输出	V X.7

• 冷却脉冲输出地址

0 通道脉冲输出	V Y.0
1 通道脉冲输出	V Y.1
2 通道脉冲输出	V Y.2
3 通道脉冲输出	V Y.3
4 通道脉冲输出	V Y.4
5 通道脉冲输出	V Y.5
6 通道脉冲输出	V Y.6
7 通道脉冲输出	V Y.7

【应用举例】

系统说明

本例程设置 1 个 SM231-7TD 扩展模块（插槽号为 0）第 1 个 PID 回路（通道号为 0）的参数。调用 PIDSetting 来设定该回路的参数，无需计算 PID 参数地址，只需输入回路所在的插槽号和通道号，再使能 Run 来运行该回路。

Q0.0 为正向脉冲输出；Q0.1 为负向脉冲输出；

VW0 为实际温度；VW2 为 PID 模拟量输出；

修改 PID 设定参数使用其他地址；

设定温度：VW120；

控制字：VB122；

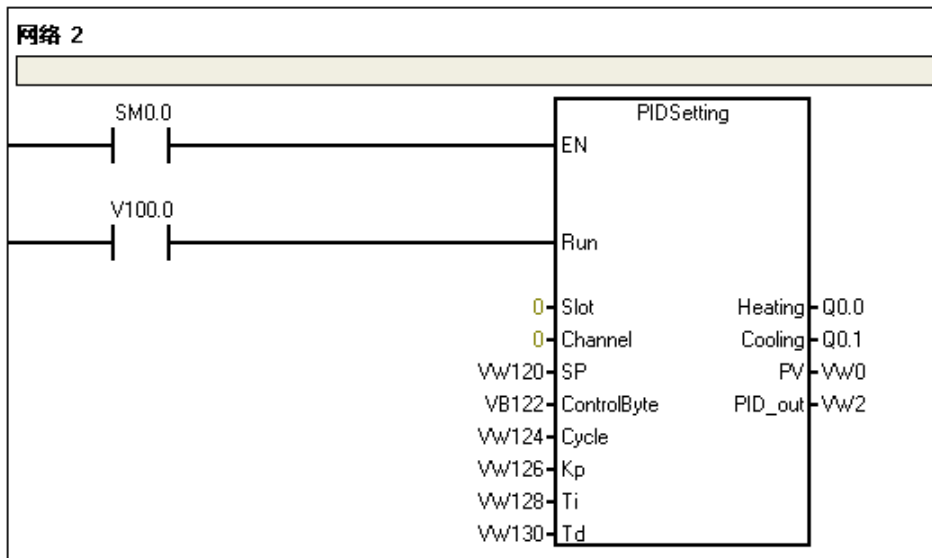
脉冲输出周期：VW124；

比例系数：VW126；

积分时间：VW128；

微分时间：VW130；

应用程序





提示

为了保证 PID 模块能正常使用，编写其它程序时请务必不要使用如下的 PID 模块占用的 V 存储区。

模块在第 0 号插槽所占用的地址为：VW2048 到 VW2298

模块在第 1 号插槽所占用的地址为：VW2304 到 VW2554

模块在第 2 号插槽所占用的地址为：VW2560 到 VW2810

模块在第 3 号插槽所占用的地址为：VW2816 到 VW3066

模块在第 4 号插槽所占用的地址为：VW3072 到 VW3322

模块在第 5 号插槽所占用的地址为：VW3328 到 VW3578

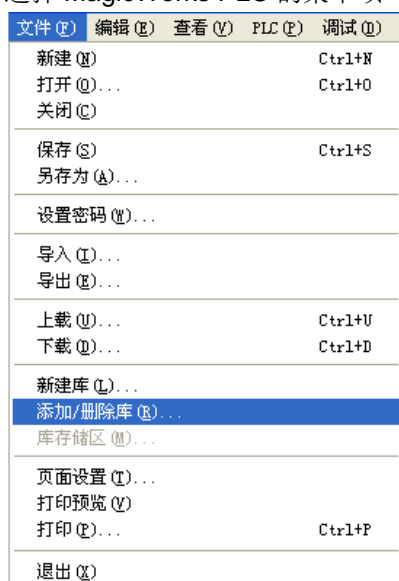
模块在第 6 号插槽所占用的地址为：VW3584 到 VW3834

E motion_ctrl_lib(v1.5)库文件介绍

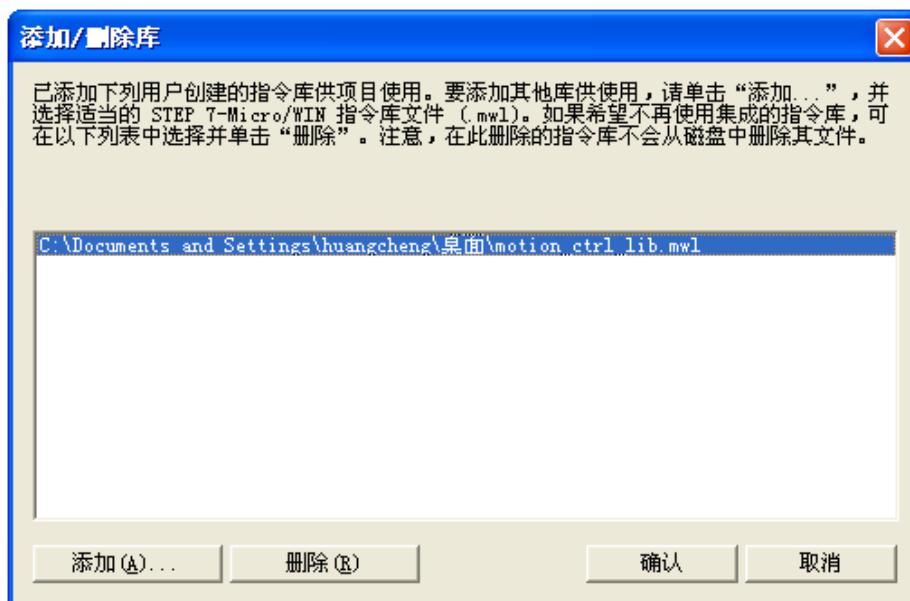
motion_ctrl_lib (v1.5) 无需复杂编程，只需要调用和设置一些简单的参数就可以使用该控制库。支持该库的 CPU 具有 4 轴独立控制功能，可进行任意两轴的直线插补或圆弧插补，同时支持线性加减速控制。

E.1 添加库文件

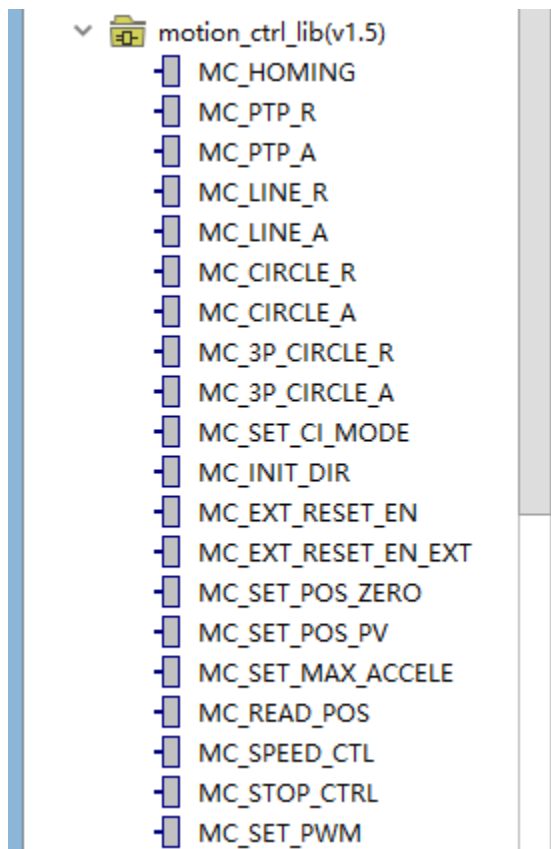
选择 MagicWorks PLC 的菜单项“文件”→“添加/删除库”，如下图所示：



点击“添加...” ，浏览库文件的保存目录。选择 motion_ctrl_lib.mwl 文件，并点击“确认”。



库添加成功之后，查看指令树中的库节点，即可看到新库 motion_ctrl_lib.mwl 已经被添加到指令树中。



【库指令中运动轴与 CPU 的 I/O 对应】

常规输出	Q0.0	Q0.1	Q0.2	Q0.3	Q0.4	Q0.5
H224/ H226L/H226M	Pulse_0	Pulse_1	Pulse_2	无	无	无
H224X/ H226XL/ H228XL	Pulse_0	Pulse_1	Dir_0	Dir_1	无	无
H226XM/H226IM/ H226IL	Pulse_0	Pulse_1	Pulse_2	Dir_0	Dir_1	Dir_2

常规输出	Q0.0	Q0.1	Q0.2	Q0.3	Q0.4	Q0.5	Q0.6	Q0.7	Q1.0	Q1.1
H226IH	Pulse_0	Dir_0	Pulse_1	Dir_1	Pulse_2	Dir_2	Pulse_3	Dir_3	Q1.0	Q1.1



提示

Pulse_x -----x 轴脉冲输出； Dir_x -----x 轴方向输出；

E.2 motion_ctrl_lib (v1.5) 库指令一览表

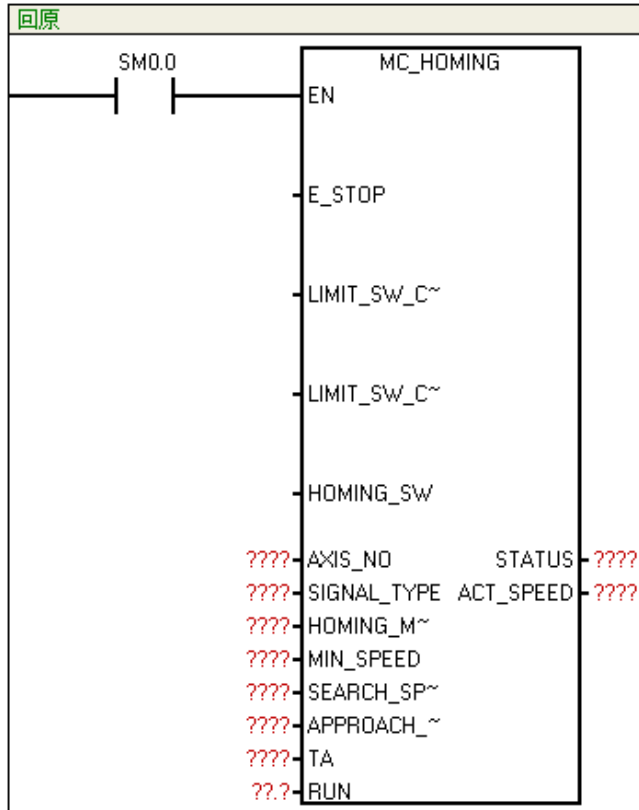
库指令一览表

函数名	指令名	H226IH	H224/H226L/ H226M	H224X/H226XL/ H226XM/H228XL/ H226IM /H226IL
MC_HOMING	回原指令	支持	不支持	支持
MC_INIT_DIR	配置电机方向指令	支持	不支持	支持
MC_EXT_RESET_EN	外部复位坐标使能指令	支持	不支持	支持
MC_EXT_RESET_EN_EXT	外部复位坐标使能指令 II	支持	不支持	支持
MC_SET_POS_ZERO	软件清零指令	支持	不支持	支持
MC_SET_POS_PV	设置目标位置指令	支持	不支持	不支持
MC_SET_MAX_ACCELE	设置最大加速度指令	支持	不支持	支持
MC_READ_POS	读位置指令	支持	不支持	支持
MC_SPEED_CTL	速度控制指令	支持	支持	支持
MC_PTP_R	单轴相对运动指令	支持	支持	支持
MC_PTP_A	单轴绝对运动指令	支持	不支持	支持
MC_LINE_R	两轴直线插补相对运动指令	支持	不支持	不支持
MC_LINE_A	两轴直线插补绝对运动指令	支持	不支持	不支持
MC_CIRCLE_R	两轴圆弧插补相对运动指令	支持	不支持	不支持
MC_CIRCLE_A	两轴圆弧插补绝对运动指令	支持	不支持	不支持
MC_3P_CIRCLE_R	两轴三点画弧插补相对运动指令	支持	不支持	不支持
MC_3P_CIRCLE_A	两轴三点画弧插补绝对运动指令	支持	不支持	不支持
MC_SET_CI_MODE	设置连续插补功能指令	支持	不支持	不支持
MC_STOP_CTL	控制停止指令	支持	支持	支持
MC_SET_PWM	占空比设置指令	不支持	不支持	不支持

<备注> MC_SET_PWM 指令仅 226EH 支持。

MC_HOMING (回原指令)

① 函数名: MC_HOMING



② 功能: 通过设置回原模式等参数, 可寻找设备原点。

轴号与外部复位 IO 信号 (如回原 Z pulse) 的对应关系:

- 轴 0 —— I0.2 (HSC0, SM37.0) 轴 1 —— I1.0 (HSC1, SM47.0)
- 轴 2 —— I1.4 (HSC2, SM57.0) 轴 3 —— I0.5 (HSC4, SM147.0)

若回原模式以原点开关为参考时 (回原模式 3 或 4), 必须将原点开关信号 (指令的 **HOMING_SW** 参数) 接至上述对应点, 否则无法找到原点。

③ 参数

参数名	输入输出属性	参数描述	类型	数值范围	备注								
E_STOP	IN	紧急停止位。 1: 有效, 0: 无效	BOOL	0~1	1、只有 RUN =1 与 E_STOP =0 时才能运行。 2、当 E_STOP 为 1 时, RUN 内部复位。								
LIMIT_SW_CCW	IN	CCW 逆时针行程限位输入	BOOL	0~1									
LIMIT_SW-CW	IN	CW 顺时针行程限位输入	BOOL	0~1									
HOMING_SW	IN	原点信号输入	BOOL	0~1									
AXIS_NO	IN	轴号	BYTE	0~3	该参数在运行过程中不可修改。								
SIGNAL_TYPE	IN	<table border="1" style="margin-left: 20px;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> </table> 信号类型 Bit0: 逆时针行程限位输入信号类型 0—高电平 1—低电平 Bit1: 顺时针行程限位输入信号类型 0—高电平 1—低电平 Bit2: 原点开关信号类型 0—高电平 1—低电平	7	6	5	4	3	2	1	0	BYTE	0~255	
7	6	5	4	3	2	1	0						
HOMING_MODE	IN	回原模式	BYTE	1~14	模式说明详见章节“4 回原功能”。								
MIN_SPEED	IN	最小速度。 单位: Hz	DWORD	0~200000	1、当速度小于 5Hz 时, 脉冲输出关闭, 即无输出。								
SEARCH_SPEED	IN	原点搜索速度。 单位: Hz	DWORD	0~200000	2、该参数在运行过程中可以修改。								
APPROACH_SPEED	IN	原点接近速度。 单位: Hz	DWORD	0~200000	3、搜索速度不应太大, 接近速度应尽量小。								
TA	IN	加减速时间。单位: ms	DWORD	0~10000	1、该参数在运行过程中可以修改。 2、具体指令的加速度详见附录 E3 提示 ③								

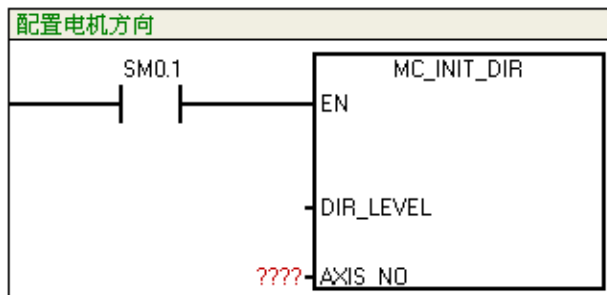
RUN	IN	运行使能位 1: 有效	BOOL	0~1	1、只有 RUN =1 与 E_STOP =0 时才能运行。 2、当运行完成后, RUN 内部复位。 3、当 E_STOP 为 1 时, RUN 内部复位。								
STATUS	OUT	<table border="1" style="margin-left: 20px;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> </table> 输出状态字节: Bit0: 参数配置错误标志 1—参数配置错误 0—参数配置正常 Bit1: 运行标志 1—正在运行 0—不运行 Bit2: 完成标志 1—完成, 指令执行完毕 0—未完成, 指令未执行 Bit3: 忙标志 1—忙标志有效, 该轴正在被其它指令占用。 0—忙标志无效, 指令正在执行或此执行已完成。 Bit4: 预留。 Bit5: 是否找到原点 1—找到原点 0—没找到原点	7	6	5	4	3	2	1	0	BYTE	0~255	Bit0: 1、只对轴参数配置错误和回原模式超范围进行判断。 2、其它参数不作报错, 会自动设置成一个最接近的合理值。 3、若 TA=0 若没有设置最大加速度, 则报参数故障; TD 亦然。
7	6	5	4	3	2	1	0						
ACT_SPEED	OUT	当前速度	DWORD	0~200000									

④ 说明

程序对各输入的检测以扫描方式实现 (Z 相信号不受此影响), 故当开关量变化时处理不及时, 可能有些延迟。若回原速度 (包括搜索速度和接近速度) 太大时, 这个处理延迟被放大, 导致回原不准。

MC_INIT_DIR (配置电机方向指令)

① 函数名: MC_INIT_DIR



② 功能: 配置电机的方向。

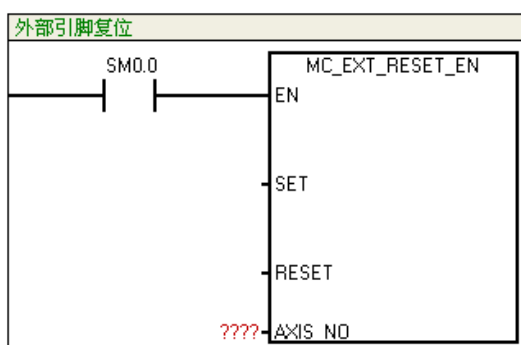
建议此指令只在 CPU 上电第一个扫描周期执行一次。

③ 参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
DIR_LEVEL	IN	配置方向信号为正向时的有效电平。 DIR_LEVEL 为 1 时,设置对应方向轴输出“1”时为电机正转。 DIR_LEVEL 为 0 时,设置对应方向轴输出“0”时为电机正转。	BOOL	0~1	默认值: 1,即默认方向轴输出为“1”时为电机正转。
AXIS_NO	IN	设置轴号	BYTE	0~3	

MC_EXT_RESET_EN (外部复位坐标使能指令)

① 函数名: MC_EXT_RESET_EN



② 功能: 当调用该指令, 设置是否使能外部 IO 复位绝对坐标值。

轴号与外部复位信号的对应关系:

轴 0 — I0.2 (HSC0, SM37.0) 轴 2 — I1.4 (HSC2, SM57.0)

轴 1 — I1.0 (HSC1, SM47.0) 轴 3 — I0.5 (HSC4, SM147.0)

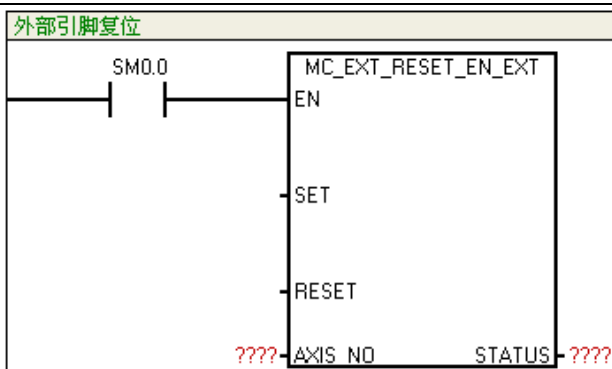
各轴的复位信号输入引脚与对应高速计数器的复位信号引脚一致;各轴的复位信号有效电平由对应高速计数器的复位有效电平控制位决定,若没有配置对应高速计数器,则默认复位有效电平为高电平。

③ 参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
SET	IN	SET 上升沿, 设置外部复位使能, 每次调用时, SET 应先复位, 然后再置 1。	BOOL	0~1	
RESET	IN	RESET 上升沿, 禁止外部复位使能, 每次调用时, RESET 应先复位, 然后再置 1。	BOOL	0~1	
AXIS_NO	IN	设置轴号	BYTE	0~3	

MC_EXT_RESET_EN_EXT (外部复位坐标使能指令 II)

① 函数名: MC_EXT_RESET_EN_EXT



② 功能：当调用该指令，设置是否使能外部 IO 复位绝对坐标值，并查看指令执行状态。

轴号与外部复位信号的对应关系：

轴 0 — I0.2 (HSC0, SM37.0) 轴 2 — I1.4 (HSC2, SM57.0)

轴 1 — I1.0 (HSC1, SM47.0) 轴 3 — I0.5 (HSC4, SM147.0)

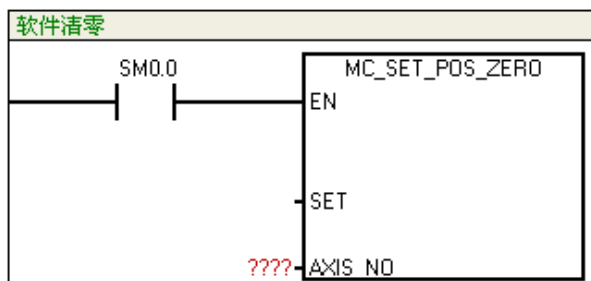
各轴的复位信号输入引脚与对应高速计数器的复位信号引脚一致；各轴的复位信号有效电平由对应高速计数器的复位有效电平控制位决定，若没有配置对应高速计数器，则默认复位有效电平为高电平。

③ 参数

参数名	输入输出属性	参数描述	类性	数值范围	备注
SET	IN	SET上升沿，设置外部复位使能，每次调用时，SET应先复位，然后再置1。	BOOL	0~1	
RESET	IN	RESET上升沿，禁止外部复位使能，每次调用时，RESET 应先复位，然后再置1。	BOOL	0~1	
AXIS_NO	IN	设置轴号	BYTE	0~3	
STATUS	OUT	状态位：  Bit0: 复位状态标志位 1—复位完成 0—复位未完成 Bit1~Bit7: 预留	BYTE	0~1	

MC_SET_POS_ZERO (软件清零指令)

① 函数名：MC_SET_POS_ZERO



② 功能：将绝对坐标复位。

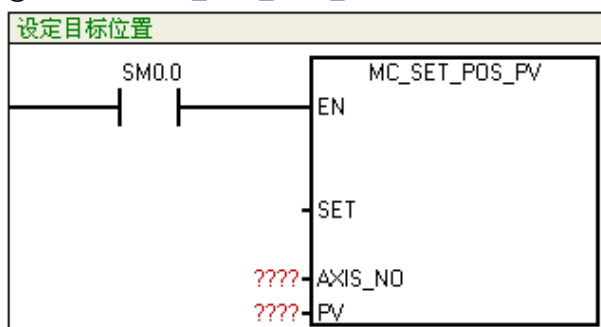
当机器运动到某一位置时，调用该指令，相当于把该轴的原点设定在该位置。那么以后每次调用“读绝对坐标”命令，就能得到相对于该点的坐标值。

③ 参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
SET	IN	清零功能使能位。 在SET上升沿把绝对坐标清0，每次调用时，SET应先置0，然后再置1。	BOOL	0~1	
AXIS_NO	IN	设置轴号。	BYTE	0~3	

MC_SET_POS_PV（设置目标位置指令）

① 函数名：MC_SET_POS_PV



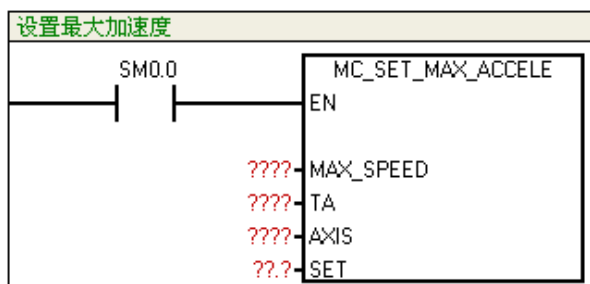
② 功能：此指令用于将机器所处的绝对位置写入到模块。比如，机器运行到某一位置时断电，可将其此时所处的位置保存下来，等下次上电时，将此位置写回到模块，则模块绝对位置计数起点与机器实际起点位置一致，而机器不需回到原点；若此位置刚好为原点，则此指令与MC_SET_POS_ZERO效果相同。

③ 参数

参数名	输入输出属性	参数描述	属性	数值范围	备注
AXIS_NO	IN	设置轴号。	BYTE	0~3	
SET	IN	SET上升沿，指令使能，每次调用时，SET应先复位，然后再置1。	BOOL	0~1	
PV	IN	设定的目标位置，分正负。输出正脉冲表示沿X轴的正方向，负脉冲数表示沿着X轴的负方向。	DINT	-2147483648 ~ +2147483647	

MC_SET_MAX_ACCELE（设置最大加速度指令）

① 函数名：MC_SET_MAX_ACCELE



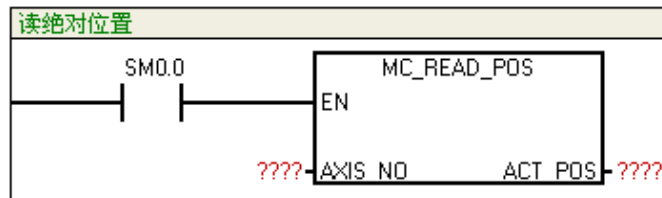
② 功能：设置最大加速度（= MAX_SPEED/TA）（TA≠0）（若没有调用此指令，则认为没有设置最大加速度）

③ 参数

参数名	输入输出属性	参数描述	属性	数值范围	备注
MAX_SPEED	IN	长轴最大速度，即运行中的最大速度。 单位：Hz	DWORD	0~200000	运行过程中可以修改。
TA	IN	加速/减速时间， 单位：ms	DWORD	0~10000	运行过程中可以修改；若 TA=0，则认为没有设置最大加速度。
AXIS	IN	设置轴号	BYTE	0~3	此指令无错误状态输出，轴号必须设置正确。
SET	IN	在以上参数确定后，给 SET 一个上升沿以使设置生效。	BOOL	0~1	

MC_READ_POS (读位置指令)

① 函数名：MC_READ_POS



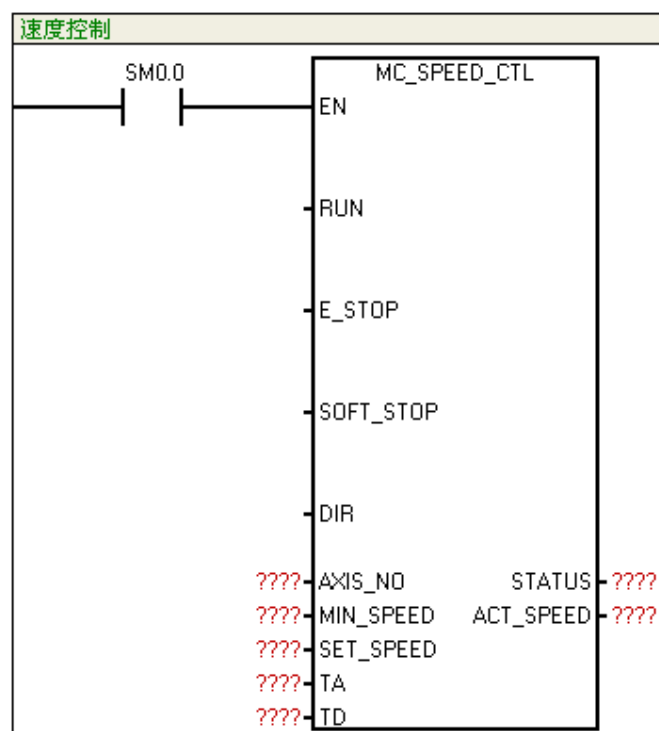
② 功能：读取每轴的绝对坐标值。一旦设定原点坐标后，那么该值会根据输出的脉冲和方向的关系进行代数计算：正转输出一个脉冲：+1，反转输出一个脉冲：-1。最后得到的是一个以设定点为原点的绝对坐标。

③ 参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
AXIS_NO	IN	设置轴号	BYTE	0~3	
ACT_POS	OUT	当前轴的绝对坐标（1个脉冲代表1个单位坐标）	DINT	-2147483648 ~ +2147483647	此指令无错误状态输出，轴号必须设置正确。

MC_SPEED_CTL (速度控制指令)

① 函数名: MC_SPEED_CTL



② 功能: 控制单轴输出脉冲的频率, 可任意时候改变输出脉冲的频率(速度)。当接收到软停止命令时, 会自动减速停止。当收到紧急停止命令时, 会马上停止脉冲输出, 不经过减速。

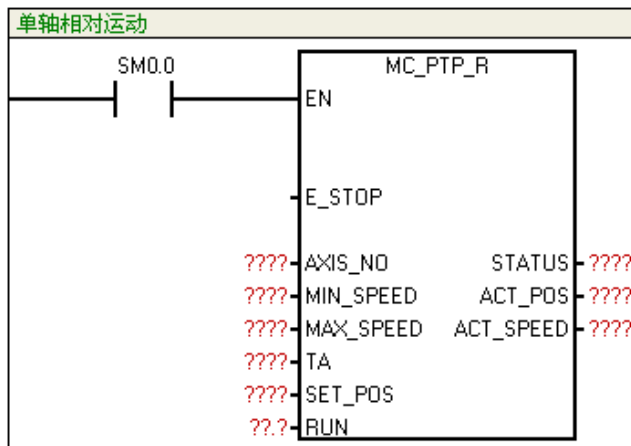
③ 参数

参数名	输入输出属性	参数描述	数据类型	数值范围	备注
RUN	IN	运行使能位。 1: 有效, 0: 无效。	BOOL	0/1	只有RUN=1与E_Stop=0且SOFT_STOP=0时才能运行。
E_STOP	IN	紧急停止位。1: 有效, 0: 无效。 当收到有效紧急停止命令后, 输出脉冲会马上停止, 不经过减速。	BOOL	0/1	只有RUN=1与E_Stop=0且SOFT_STOP=0时才能运行。
SOFT_STOP	IN	软停止位。1: 有效, 0: 无效。 当收到有效软停止命令时, 输出脉冲会减速停止。	BOOL	0/1	只有RUN=1与E_Stop=0且SOFT_STOP=0时才能运行。
DIR	IN	脉冲的方向位	BOOL	0/1	该参数在运行过程中能修改。
AXIS_NO	IN	设置轴号	BYTE	0~3	该参数在运行过程中不能修改。
MIN_SPEED	IN	最小速度, 即启动时或停止时的速度。单位: Hz	DWORD	0~200000	第5章由于处理器的局限, 当速度小于5Hz时, 脉冲输出关闭, 即无输出;
SET_SPEED	IN	设定速度, 在收到停止命令前, 输出脉冲会加速或减速到此速度。单位: Hz	DWORD	0~200000	2、该参数在运行过程中可以修改。 3、将设定速度写0, 可实现软停功能。

TA	IN	加速时间，从最小速度到设定速度的加速时间。单位：ms	DWORD	0~10000	1、该参数在运行过程中可以修改； 2、加速度只在启动时和TA/TD变化时计算，计算方法详见附录E3提示②								
TD	IN	减速时间，从设定速度到最小速度的减速时间。单位：ms	DWORD	0~10000									
STATUS	OUT	输出状态字节： <table border="1" style="margin-left: 20px;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> </table> Bit0: 参数配置错误标志 1—参数配置错误 0—参数配置正常 Bit1: 运行标志 1—正在运行，该指令正在输出脉冲，并且未执行完。 0—不运行，该因公共资源被其他指令占用，所以指令还没得以运行；或者指令已经运行完毕 Bit2: 完成标志 1—完成，指令执行完毕。 0—未完成，执行没执行或者指令正在执行中但没完成 Bit3: 忙标志 1—忙标志有效，该轴正在被其它指令占用 0—忙标志无效，指令正在执行或此执行完成 Bit4~Bit7: 预留	7	6	5	4	3	2	1	0	BYTE	0~255	Bit0: 1、只对轴参数配置错误进行判断； 2、MIN_SPEED/SET_SPEED等参数不作报错，会自动设置成一个最接近的合理值。 3、若TA=0若没有设置最大加速度，则报参数故障；TD亦然。
7	6	5	4	3	2	1	0						
ACT_S PEED	OUT	当前速度（频率）输出	DWORD	0~200000									

MC_PTP_R（单轴相对运动指令）

① 函数名：MC_PTP_R



② 功能：用作单轴点对点控制（单轴定长驱动）。调用一次可输出固定脉冲，通过最大、最小速度和加减速时间的设定，输出的脉冲在启动时会逐渐的加速到最大的速度，当脉冲数快要跑完时，脉冲的频率会自动减下来，以防止在启动或停止时的机器的惯性太大而引起振动或卡死。

③ 参数

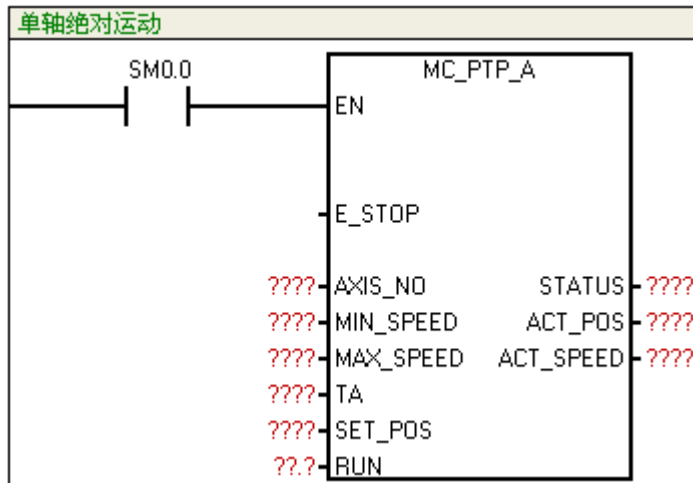
参数名	输入输出属性	参数描述	类型	数值范围	备注
E_STOP	IN	紧急停止位。 1: 有效 0: 无效	BOOL	0/1	1、只有Run=1与E_Stop=0时才能运行。 2、E_STOP 为1时，RUN 内部复位。

AXIS_NO	IN	设置轴号	BYTE	0~3	该参数在运行过程中不能修改。								
MIN_SPEED	IN	最小速度，即启动时或停止时的速度。 单位：Hz	DWORD	0~200000	1、由于处理器的局限，当速度小于5Hz时，脉冲输出关闭，即无输出； 2、该参数在运行过程中可以修改。 3、建议 MIN_SPEED 不要小于 500，否则在脉冲输出将完成减速结束时最小速度受限（通常限制为 500）。 4、将 MAX_SPEED 写 0，可实现软停功能（即输出脉冲减速停止），此时使能位 RUN 不复位；若将速度写回大值，可继续输出脉冲直至输出完成。								
MAX_SPEED	IN	最大速度，即运行中的最大速度。 单位：Hz	DWORD	0~200000									
TA	IN	加速 / 减速时间。 单位：ms	DWORD	0~10000	1、该参数在运行过程中可以修改。 2、加速度只在启动时和TA/TD变化时计算，计算方法详见附录E3提示⑤								
SET_POS	IN	输出的脉冲数，分正负。正脉冲数表示沿 X 轴的正方向，负脉冲数表示沿着 X 轴的负方向（此为相对坐标）	DINT	-2147483648 ~ +2147483647	该参数在运行过程中可以修改，当新设定值大于已输出的脉冲数，那么最后输出的脉冲会以新设定值为准。当新设定值小于已输出脉冲数，则会马上停止脉冲输出。								
RUN	IN/OUT	运行使能位。 1：有效 0：无效	BOOL	0/1	1、只有 RUN=1 与 E_STOP=0 时才能运行。 2、当运行完成后，RUN 内部复位。 3、当 E_STOP 为 1 时，RUN 内部复位。								
STATUS	OUT	输出状态字节： <table border="1" style="margin-left: 20px;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> </table> Bit0：参数配置错误标志 1—参数配置错误 0—参数配置正常 Bit1：运行标志 1—正在运行，该指令正在输出脉冲，且指令未执行完。 0—不运行，因公共资源被其他指令占用，所以指令还没得以运行；或者指令已经运行完毕	7	6	5	4	3	2	1	0	BYTE	0~255	Bit0 注： 1、只对轴参数进行判断； 2、MIN_SPEED/MAX_SPEED 等参数不作报错，会自动设置成一个最接近的合理值。 3、若 TA=0 若没有设置最大加速度，则报参数故障；TD 亦然。
7	6	5	4	3	2	1	0						

		<p>Bit2: 完成标志 1—完成, 指令执行完毕 0—未完成, 指令未执行或指令正在执行中但未完成</p> <p>Bit3: 忙标志 1—忙标志有效, 该轴正在被其它指令占用 0—忙标志无效, 指令正在执行或此执行已完成</p> <p>Bit4~Bit7: 预留</p>			
ACT_POS	OUT	当前的相对坐标或本指令已输出的脉冲数	DINT	-2147483648 ~ +2147483647	
ACT_SPEED	OUT	当前实际运行速度	DWORD	0~200000	

MC_PTP_A (单轴绝对运动指令)

① 函数名: MC_PTP_A



② 功能: 用作单轴点对点控制 (非定长, 而是定点)。调用一次可在原脉冲数基础上输出脉冲至指定脉冲数, 通过最大、最小速度和加减速时间的设定, 输出的脉冲在启动时会逐渐的加速到最大的速度, 当脉冲数快要跑完时, 脉冲的频率会自动减下来, 以防止在启动或停止时的机器的惯性太大而引起振动或卡死。

③ 参数

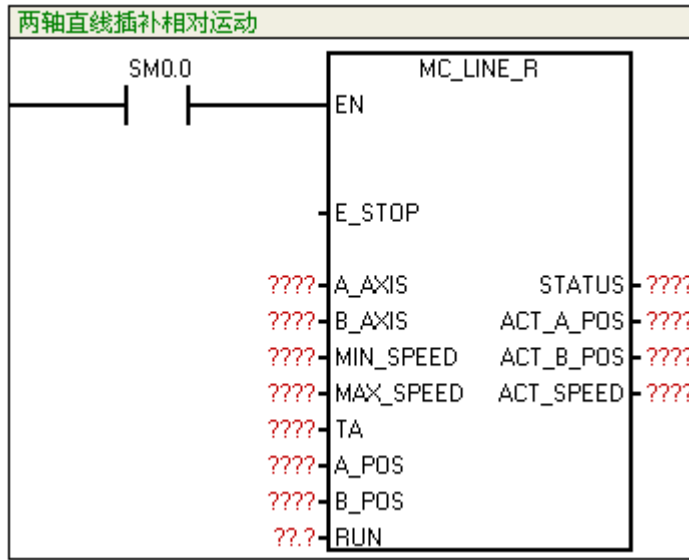
参数名	输入输出属性	参数描述	类型	数值范围	备注
E_STOP	IN	紧急停止位。 1: 有效 0: 无效	BOOL	0/1	1、只有 Run =1 与 E_Stop=0 时才能运行。 2、当 E_STOP 为 1 时, RUN 内部复位。
AXIS_NO	IN	设置轴号	BYTE	0~3	该参数在运行过程中不能修改。

MIN_SPEED	IN	最小速度，即启动时或停止时的速度。单位：Hz。	DWORD	0~200000	1、由于处理器的局限，当速度小于5Hz时，脉冲输出关闭，即无输出。 2、该参数在运行过程中可以修改。 3、建议MIN_SPEED不要小于500，否则在脉冲输出将完成减速结束时最小速度受限（通常限制为500）。 4、将MAX_SPEED写0，可实现软停功能（即输出脉冲减速停止），此时使能位RUN不复位；若将速度写回大值，可继续输出脉冲直至输出完成。								
MAX_SPEED	IN	最大速度，即运行中的最大速度。单位：Hz	DWORD	0~200000	1、该参数在运行过程中可以修改。 2、加速度只在启动时和TA/TD变化时计算，计算方法详见附录E3提示⑥								
TA	IN	加速 /减速时间，单位ms	DWORD	0~10000	该参数在运行过程中可以修改，当新设定值大于已输出的脉冲数，那么最后输出的脉冲会以新设定值为准。当新设定值小于已输出脉冲数，则会马上停止脉冲输出。								
SET_POS	IN	输出的脉冲数，分正负。正脉冲数表示沿 X 轴的正方向，负脉冲数表示沿着 X 轴的负方向（此为绝对坐标）	DINT	-2147483648 ~ +2147483647	1) 只有 RUN=1 与 E_STOP=0 时才能运行。 2、当运行完成后，RUN 内部复位。 3、当E_STOP 为1时，RUN 内部复位。								
RUN	IN/OUT	运行使能位。 1: 有效 0: 无效	BOOL	0/1	输出状态字节： <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> </table> Bit0: 参数配置错误标志 1—参数配置错误。 0—参数配置正常。 Bit1: 运行标志 1—正在运行，该指令正在输出脉冲，且指令未执行完。 0—不运行，因公共资源被其他指令占用，所以指令还没得以运行；或者指令已经运行完毕。 Bit2: 完成标志 1—完成，指令执行完毕。 0—未完成，指令未执行或指令正在执行中但未完成。 Bit3: 忙标志	7	6	5	4	3	2	1	0
7	6	5	4	3	2	1	0						
STATUS	OUT	输出状态字节： <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> </table> Bit0: 参数配置错误标志 1—参数配置错误。 0—参数配置正常。 Bit1: 运行标志 1—正在运行，该指令正在输出脉冲，且指令未执行完。 0—不运行，因公共资源被其他指令占用，所以指令还没得以运行；或者指令已经运行完毕。 Bit2: 完成标志 1—完成，指令执行完毕。 0—未完成，指令未执行或指令正在执行中但未完成。 Bit3: 忙标志	7	6	5	4	3	2	1	0	BYTE	0~255	Bit0: 1、只对轴参数进行判断。 2) MIN_SPEED/MAX_SPEED等参数不作报错，会自动设置成一个最接近的合理值。 3、若TA=0若没有设置最大加速度，则报参数故障；TD亦然。
7	6	5	4	3	2	1	0						

		1—忙标志有效，该轴正在被其它指令占用。 0—忙标志无效，指令正在执行或此执行已完成。 Bit4~Bit7: 预留			
ACT_POS	OUT	当前的绝对坐标	DINT	-2147483648 ~ +2147483647	
ACT_SPEED	OUT	当前实际运行速度	DWORD	0~200000	

MC_LINE_R (两轴直线插补相对运动指令)

① 函数名: MC_LINE_R



② 功能: 可在任意两轴之间、平面上任意区域内进行直线插补功能(设置点为相对坐标)。

③ 参数

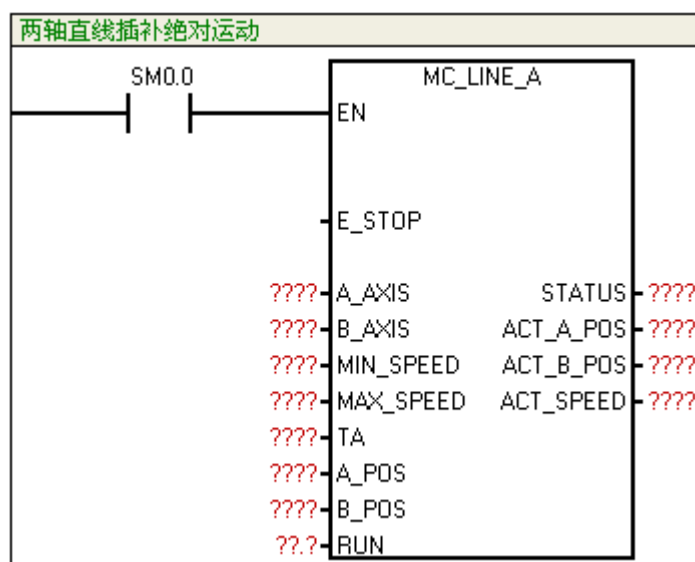
参数名	输入输出属性	参数描述	类型	数值范围	备注
E_STOP	IN	紧急停止位。 1: 有效 0: 无效	BOOL	0/1	1、只有 RUN =1 与 E_STOP =0 时才能运行。 2、当 E_STOP为 1 时，RUN 内部复位。
A_AXIS	IN	插补 A 轴的轴号。 插补需要两个轴，即虚拟的 A轴和B轴。我们需要映射到实际输出的 0、1、2、3 轴上。该参数即可设定A轴映射到相应轴上。	BYTE	0~3	该参数在运行过程中不可修改。
B_AXIS	IN	插补 B 轴的轴号。 插补需要两个轴，即虚拟的 A轴和B轴。我们需要映射到实际输出的 0、1、2、3 轴上。该参数即可设定B轴映射到相应的轴上。	BYTE	0~3	

MIN_SPEED	IN	长轴最小速度，即启动时或停止时的速度。单位：Hz	DWORD	0~200000	1、由于处理器的局限，当速度小于5Hz时，脉冲输出关闭，即无输出。 2、该参数在运行过程中可以修改。 3、建议MIN_SPEED不要小于500，否则在脉冲输出将完成减速结束时最小速度受限（通常限制为500）。 4、将MAX_SPEED写0，可实现软停功能（即输出脉冲减速停止），此时使能位RUN不复位；若将速度写回大值，可继续输出脉冲直至输出完成。
MAX_SPEED	IN	长轴最大速度，即运行中的最大速度。单位：Hz	DWORD	0~200000	该参数在运行过程中可以修改。 加速度只在启动时和TA/TD变化时计算，计算方法详见附录E3提示③
TA	IN	加速/减速时间。单位：ms	DWORD	0~10000	
A_POS	IN	虚拟 A 轴的终点(相对)坐标	DINT	-2147483648 ~ +2147483647	1、该参数在运行过程中不能修改； 2、单位：脉冲，与实际位移转换计算方法如下： POS（插补指令中需填入的脉冲个数），L（坐标上实际点的绝对值，单位 mm）， S（运动轴上丝杆导程，单位 mm）， M（步进驱动器的细分或伺服驱动的分辨率（即电机转动一圈需要的脉冲个数））； 计算公式：POS = L*M÷S（单位：脉冲数）
B_POS	IN	虚拟 B 轴的终点(相对)坐标	DINT	-2147483648 ~ +2147483647	
RUN	IN/OUT	运行使能位。 1：有效 0：无效	BOOL	0/1	1、只有 RUN =1 与 E_STOP =0 时才能运行。 2、当运行完成后，RUN 内部复位。

					3、当 E_STOP为1 时，RUN内部复位。								
STATUS	OUT	<p>输出状态字节：</p> <table border="1" style="margin-left: 20px;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> </table> <p>Bit0: 参数配置错误标志 1—参数配置错误。 0—参数配置正常。</p> <p>Bit1: 运行标志 1—正在运行，该指令正在输出脉冲，并且还没执行完。 0—不运行，该因公共资源被其他指令占用，所以指令还没得以运行；或者指令已经运行完毕。</p> <p>Bit2: 完成标志 1—完成，指令执行完毕。 0—未完成，执行没执行或者指令正在执行中但没完成。</p> <p>Bit3: 忙标志 1: 忙标志有效，直线插补模块或相应的轴被其指令占用。 0: 忙标志无效，指令正在执行或此执行完成。</p> <p>Bit4~Bit7: 预留</p>	7	6	5	4	3	2	1	0	BYTE	0~255	<p>Bit0:</p> <p>1、只对轴参数配置错误进行判断。</p> <p>2、MIN_SPEED /MAX_SPEED 等参数不作报错，会自动设置成一个最接近的合理值。</p> <p>3、若TA=0若没有设置最大加速度，则报参数故障；TD亦然。</p>
7	6	5	4	3	2	1	0						
ACT_A_POS	OUT	A 轴的当前位置(相对坐标，本次调用实际输出脉冲数)，如果 A 轴配给0轴，那么该值就表示0轴的相对坐标。	DINT	-2147483648 ~ +2147483647									
ACT_B_POS	OUT	B 轴的当前位置(相对坐标，本次调用实际输出脉冲数)，如果 B 轴配给1轴，那么该值就表示1轴的相对坐标。	DINT	-2147483648 ~ +2147483647									
ACT_SPEED	OUT	当前的实际速度	DWORD	0~200000									

MC_LINE_A (两轴直线插补绝对运动指令)

① 函数名: MC_LINE_A



② 功能：可在任意两轴之间、平面上任意区域内进行直线插补功能（设置点为绝对坐标）。

③ 参数

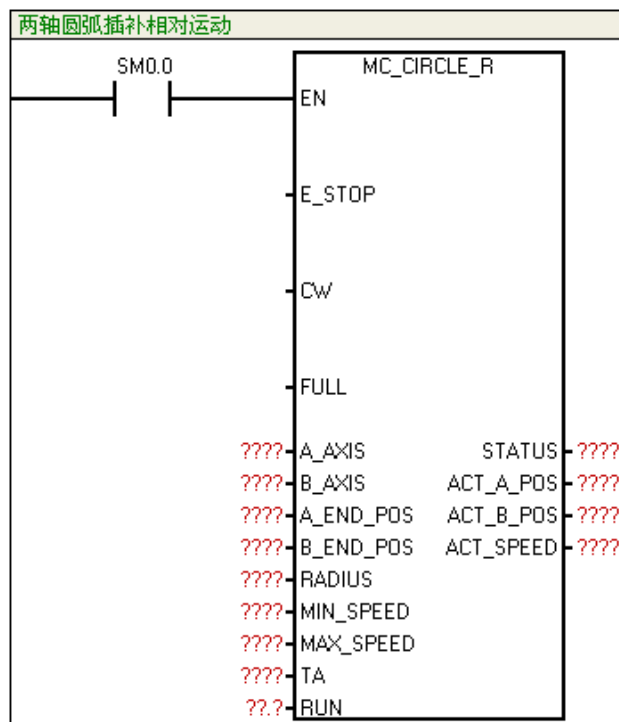
参数名	输入输出属性	参数描述	类型	数值范围	备注
E_STOP	IN	紧急停止位。 1: 有效 0: 无效	BOOL	0/1	1、只有 RUN =1 与 E_STOP =0 时才能运行。 2、当 E_STOP 为 1 时，RUN 内部复位。
A_AXIS	IN	插补 A 轴的轴号。 插补需要两个轴，即虚拟的A轴和B轴。我们需要映射到实际输出的 0、1、2、3 轴上。该参数即可设定A轴映射到相应轴上。	BYTE	0~3	该参数在运行过程中不可修改。
B_AXIS	IN	插补 B 轴的轴号。 插补需要两个轴，即虚拟的A轴和B轴。我们需要映射到实际输出的 0、1、2、3 轴上。该参数即可设定B轴映射到相应的轴上。	BYTE	0~3	
MIN_SPEED	IN	长轴最小速度，即启动时或停止时的速度。单位：Hz	DWORD	0~200000	1、当速度小于 5Hz 时，脉冲输出关闭，即无输出。 2、该参数在运行过程中可以修改。 3、建议 MIN_SPEED 不要小于 500，否则在脉冲输出将完成减速结束时最小速度受限（通常限制为 500）。
MAX_SPEED	IN	长轴最大速度，即运行中的最大速度。单位：Hz	DWORD	0~200000	4、将 MAX_SPEED 写 0，可实现软停功能（即输出脉冲

					减速停止)，此时使能位 RUN 不复位；若将速度写回大值，可继续输出脉冲直至输出完成。								
TA	IN	加速/减速时间。单位：ms	DWORD	0~10000	1、该参数在运行过程中可以修改。 2、加速度只在启动时和 TA/TD 变化时计算，计算方法详见附录 E3 提示 ③								
A_POS	IN	虚拟 A 轴的终点（绝对）坐标	DINT	-2147483648 ~ +2147483647	1、该参数在运行过程中不能修改。 2、单位：脉冲，与实际位移转换计算方法：								
B_POS	IN	虚拟 B 轴的终点（绝对）坐标	DINT	-2147483648 ~ +2147483647	（绝对指令必须以脉冲增量来计算）。								
RUN	IN/OUT	运行使能位。 1：有效 0：无效	BOOL	0/1	1、只有 RUN =1 与 E_STOP =0 时才能运行。 2、当运行完成后，RUN 内部复位。 3、当 E_STOP 为 1 时，RUN 内部复位。								
STATUS	OUT	输出状态字节： <table border="1" style="margin-left: 20px;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> </table> Bit0：参数配置错误标志 1—参数配置错误 0—参数配置正常 Bit1：运行标志 1—正在运行，该指令正在输出脉冲，并且还没执行完。 0—不运行，该因公共资源被其他指令占用，所以指令还没得以运行；或者指令已经运行完毕 Bit2：完成标志 1—完成，指令执行完毕。 0—未完成，执行没执行或者指令正在执行中但没完成 Bit3：忙标志 1：忙标志有效，直线插补模块或相应的轴被其指令占用 0：忙标志无效，指令正在执行或此执行完成 Bit4~Bit7：预留	7	6	5	4	3	2	1	0	BYTE	0~255	Bit0： 1、只对轴参数配置错误进行判断。 5. MIN_SPEED / MAX_SPEED 等参数不作报错，会自动设置成一个最接近的合理值。 3、若 TA=0 若没有设置最大加速度，则报参数故障；TD 亦然。
7	6	5	4	3	2	1	0						
ACT_A_POS	OUT	A 轴的当前位置（绝对坐标），如果 A 轴配给 0 轴，那么该值就表示 0 轴的绝对坐标	DINT	-2147483648 ~ +2147483647									
ACT_B_POS	OUT	B 轴的当前位置（绝对坐标），如果 B 轴配给 1 轴，那么该值	DINT	-2147483648 ~ +2147483647									

		就表示1轴的绝对坐标		+2147483647	
ACT_SPEED	OUT	当前的实际速度	DWORD	0~200000	

MC_CIRCLE_R (两轴圆弧插补相对运动指令)

① 函数名: MC_CIRCLE_R



② 功能: 可在任意两轴之间进行圆弧插补(设置点为相对坐标)。

③ 参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
E_STOP	IN	紧急停止位。 1: 有效 0: 无效	BOOL	0/1	1、只有 RUN =1与 E_STOP =0时才能运行。 2、当 E_STOP为 1 时, RUN内部复位。
CW	IN	顺时针或逆时针插补标志。 1: 顺时针 0: 逆时针	BOOL	0/1	该参数在运行过程中不能修改。
FULL	IN	全圆标志。 1: 全圆 0: 圆弧	BOOL	0/1	该参数在运行过程中不能修改。
A_AXIS	IN	插补 A 轴的轴号。 插补需要两个轴, 即虚拟的 A 轴和 B 轴。我们需要映射到实际输出的 0、1、2、3 轴上。该参数即可设定A 轴映射到那一轴上。	BYTE	0~3	该参数在运行过程中不能修改。

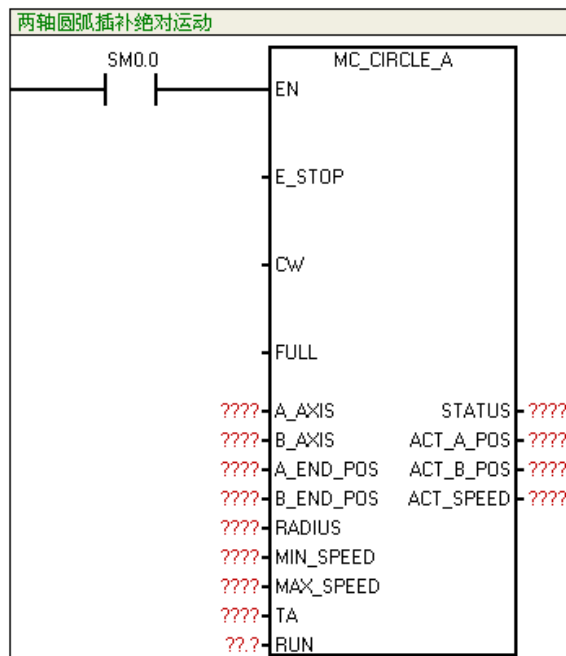
B_AXIS	IN	插补 B 轴的轴号。 插补需要两个轴，即虚拟的 A 轴和 B 轴。我们需要映射到实际输出的 0、1、2、3 轴上。该参数即可设定 B 轴映射到那一轴上。	BYTE	0~3	该参数在运行过程中不能修改。
A_END_POS	IN	如果 FULL 为 0，表示虚拟 A 轴的终点（相对起点）坐标； 如果 FULL 为 1，那么该点只表示圆弧上不同于起点的另一点的（相对起点）坐标，并非终点坐标。	DINT	-2147483648 ~ +2147483647	1、该参数在运行过程中不能修改。
B_END_POS	IN	如果 FULL 为 0，表示虚拟 B 轴的终点（相对起点）坐标； 如果 FULL 为 1，那么该点只表示圆弧上不同于起点的另一点的（相对起点）坐标，并非终点坐标。	DINT	-2147483648 ~ +2147483647	2、单位：脉冲，与实际位移转换计算方法见附录 E3 提示 ②。
RADIUS	IN	圆弧的半径。 1、分正、负。正数：表示走弧度小于 180 的圆弧轨迹。负数：表示走弧度大于 180 度的圆弧轨迹。 2、RADIUS 的绝对值表示圆弧的半径大小。	DINT	$ R < 3 \times 10^6$	1、该参数在运行过程中不能修改。 2、单位：脉冲，与实际位移转换计算方法见附录 E3 提示 ②。
MIN_SPEED	IN	长轴最小速度，即启动时或停止时的速度。单位：Hz	DWORD	0~200000	1、由于处理器的局限，当速度小于 5Hz 时，脉冲输出关闭，即无输出。 2、该参数在运行过程中可以修改。 3、建议 MIN_SPEED 不要小于 500，否则在脉冲输出将完成减速结束时最小速度受限（通常限制为 500）。
MAX_SPEED	IN	长轴最大速度，即运行中的最大速度。单位：Hz	DWORD	0~200000	4、将 MAX_SPEED 写 0，可实现软停功能（即输出脉冲减速停止），此时使能位 RUN 不复位；若将速度写回大值，可继续输出脉冲直至输

					出完成。								
TA		加速/减速时间。 单位: ms	DWORD	0~10000	<p>1、该参数在运行过程中可以修改。</p> <p>2、加速度只在启动时和 TA/TD 变化时计算, 计算方法: 当 TA≠0, 加速度= (MAX_SPEED-MIN_SPEED)/TA (若设置有最大加速度, 则受限于最大加速度); 若 TA=0, 则采用指令 MC_SET_MAX_ACCELE 设置的最大加速度, 若没有设置最大加速度, 则报参数故障; TD 亦然。对于双轴指令, 若两轴均设置了最大加速度。则采用其中的较小值作为系统加速度。</p>								
RUN	IN/OUT	运行使能位。 1: 有效 0: 无效	BOOL	0/1	<p>1、只有 RUN =1与 E_STOP=0时才能运行。</p> <p>2、当运行完成后, RUN 内部复位。</p> <p>3、当 E_STOP 为 1 时, RUN 内部复位。</p>								
STATUS	OUT	<p>输出状态字节:</p> <table border="1" style="margin-left: 20px;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> </table> <p>Bit0: 参数配置错误标志 1—参数配置错误 0—参数配置正常</p> <p>Bit1: 运行标志 1—正在运行, 该指令正在输出脉冲, 并且还没执行完。 0—不运行, 因公共资源被其他指令占用, 所以指令还未运行; 或者指令已经运行完毕。</p> <p>Bit2: 完成标志</p>	7	6	5	4	3	2	1	0	BYTE	0~255	<p>Bit0:</p> <p>1、只对轴参数配置错误进行判断。</p> <ul style="list-style-type: none"> ● MIN_SPEED/MAX_SPEED 等参数不作报错, 会自动设置成一个最接近的合理值。 ● 若TA=0若没有设置最大加速度, 则报参数故障; TD 亦然。
7	6	5	4	3	2	1	0						

		<p>1—完成，指令执行完毕。</p> <p>0—未完成，执行未执行或者指令正在执行但没完成。</p> <p>Bit3: 忙标志</p> <p>1—忙标志有效，圆弧插补模块或相应的轴被其他指令占用。</p> <p>0—忙标志无效，指令正在执行或此指令执行完成。</p> <p>Bit4~Bit7: 预留</p>			
ACT_A_POS	OUT	A 轴的当前位置（相对坐标，本次调用实际输出脉冲数），如果 A轴配给 0 轴，那么该值就表示 0轴的相对坐标。	DINT	-2147483648 ~ +2147483647	
ACT_B_POS	OUT	B 轴的当前位置（相对坐标，本次调用实际输出脉冲数），如果B轴配给 1 轴，那么该值就表示 1轴的相对坐标。	DINT	-2147483648 ~ +2147483647	
ACT_SPEED	OUT	当前的实际速度	DWORD	0~200000	

MC_CIRCLE_A（两轴圆弧插补绝对运动指令）

① 函数名: MC_CIRCLE_A



② 功能：可在任意两轴之间进行圆弧插补（设置点为绝对坐标）。

③ 参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
-----	--------	------	----	------	----

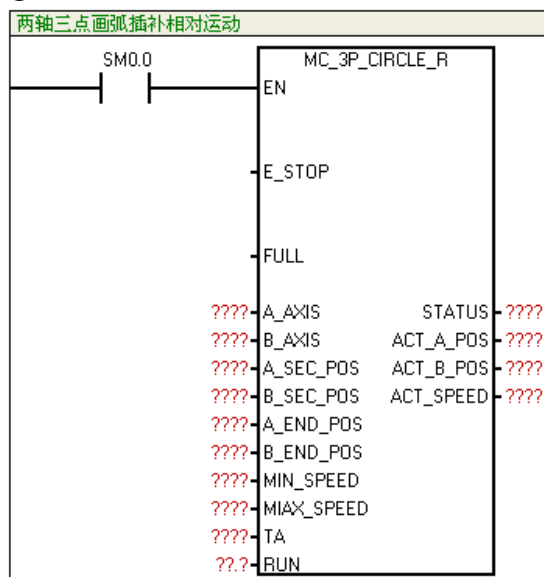
E_STOP	IN	<p>紧急停止位。</p> <p>1: 有效</p> <p>0: 无效</p>	BOOL	0/1	<p>1、只有 RUN =1 与 E_STOP =0 时才能运行。</p> <p>2、当 E_STOP为 1 时，RUN 内部复位。</p>
CW	IN	<p>顺时针或逆时针插补标志。</p> <p>1: 顺时针</p> <p>0: 逆时针</p>	BOOL	0/1	该参数在运行过程中不能修改。
FULL	IN	<p>全圆标志。</p> <p>1: 全圆</p> <p>0: 圆弧</p>	BOOL	0/1	该参数在运行过程中不能修改。
A_AXIS	IN	<p>插补 A 轴的轴号。</p> <p>插补需要两个轴，即虚拟的 A 轴和 B 轴。我们需要映射到实际输出的 0、1、2、3 轴上。该参数即可设定A轴映射到那一轴上。</p>	BYTE	0~3	该参数在运行过程中不能修改。
B_AXIS	IN	<p>插补 B 轴的轴号。</p> <p>插补需要两个轴，即虚拟的 A 轴和 B 轴。我们需要映射到实际输出的 0、1、2、3 轴上。该参数即可设定B轴映射到那一轴上。</p>	BYTE	0~3	
A_END_POS	IN	<p>如果FULL 为 0，表示虚拟 A 轴的终点（绝对）坐标；</p> <p>如果FULL为1，那么该点只表示圆弧上不同于起点的另一点的（绝对）坐标，并非终点坐标。</p>	DINT	-2147483648 ~ +2147483647	<p>1、该参数在运行过程中不能修改。</p> <p>2、单位：脉冲，与实际位移转换计算方法见附录 E3 提示 ●。（绝对指令必须以脉冲增量来计算）</p>
B_END_POS	IN	<p>如果 FULL 为 0，表示虚拟 B 轴的终点（绝对）坐标；</p> <p>如果FULL为1，那么该点只表示圆弧上不同于起点的另一点的（绝对）坐标，并非终点坐标。</p>	DINT	-2147483648 ~ +2147483647	
RADIUS	IN	<p>圆弧的半径。</p> <p>1、分正、负。正数：表示走弧度小于 180 的圆弧轨迹。负数：表示走弧度大于 180 度的圆弧轨迹。</p> <p>2、RADIUS 的绝对值表示圆弧的半径大小。</p>	DINT	$ R < 3 \times 10^6$	<p>1、该参数在运行过程中不能修改。</p> <p>2、单位：脉冲，与实际位移转换计算方法见附录 E3提示●。</p>

MIN_SPEED	IN	长轴最小速度，即启动时或停止时的速度。单位：Hz	DWORD	0~200000	1、当速度小于5Hz时，脉冲输出关闭，即无输出。 2、该参数在运行过程中可以修改。 3、建议 MIN_SPEED 不要小于 500，否则在脉冲输出将完成减速结束时最小速度受限（通常限制为 500）。 4、将 MAX_SPEED 写 0，可实现软停功能（即输出脉冲减速停止），此时使能位 RUN 不复位；若将速度写回大值，可继续输出脉冲直至输出完成。								
MAX_SPEED	IN	长轴最大速度，即运行中的最大速度。单位：Hz	DWORD	0~200000									
TA		加速/减速时间。单位：ms	DWORD	0~10000	1、该参数在运行过程中可以修改。 2、加速度只在启动时和 TA/TD 变化时计算，计算方法详见附录E3提示③								
RUN	IN/OUT	运行使能位。 1: 有效 0: 无效	BOOL	0/1	1、只有 RUN =1 与 E_STOP=0 时才能运行。 2、当运行完成后，RUN 内部复位。 3、当 E_STOP 为 1 时，RUN 内部复位。								
STATUS	OUT	输出状态字节： <table border="1" style="margin-left: 20px;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> </table> Bit0: 参数配置错误标志 1—参数配置错误 0—参数配置正常 Bit1: 运行标志 1—正在运行，该指令正在输出脉冲，并且还没执行完。 0—不运行，因公共资源被其他	7	6	5	4	3	2	1	0	BYTE	0~255	Bit0: 1、只对轴参数配置错误进行判断。 2、MIN_SPEED / MAX_SPEED 等参数不作报错，会自动设置成一个最接近的合理值。 3、若TA=0若没有设置最大加速度，
7	6	5	4	3	2	1	0						

		指令占用，所以指令还未运行； 或者指令已经运行完毕。 Bit2: 完成标志 1—完成，指令执行完毕。 0—未完成，执行未执行或者指令正在执行但没完成。 Bit3: 忙标志 1—忙标志有效，圆弧插补模块或相应的轴被其他指令占用。 0—忙标志无效，指令正在执行或此指令执行完成。 Bit4~Bit7: 预留			则报参数故障； TD亦然。
ACT_A_POS	OUT	A 轴的当前位置（绝对坐标）， 如果 A轴配给 0 轴，那么该值 就表示 0轴的绝对坐标。	DINT	-2147483648 ~ +2147483647	
ACT_B_POS	OUT	B 轴的当前位置（绝对坐标）， 如果 B轴配给 1 轴，那么该值 就表示 1轴的绝对坐标。	DINT	-2147483648 ~ +2147483647	
ACT_SPEED	OUT	当前的实际速度	DWORD	0~200000	

MC_3P_CIRCLE_R（两轴三点画弧插补相对运动指令）

① 函数名：MC_3P_CIRCLE_R



② 功能：可在任意两轴之间进行圆弧插补（设置点为相对坐标）。

③ 参数

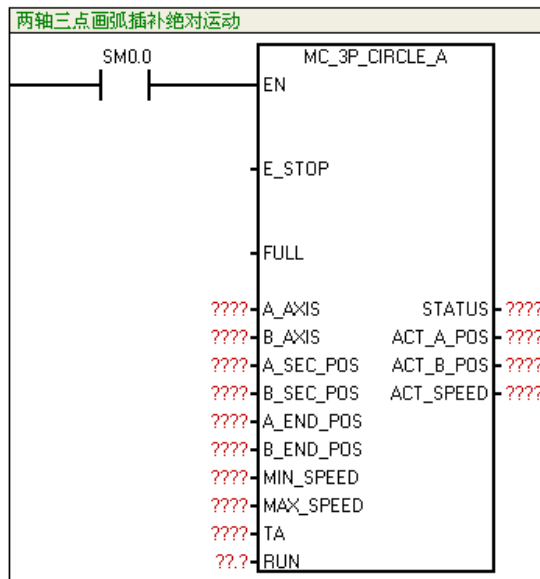
参数名	输入输出属性	参数描述	类型	数值范围	备注
E_STOP	IN	紧急停止位。 1: 有效 0: 无效	BOOL	0/1	1、只有 RUN =1与 E_STOP =0时才能运行。 2、当 E_STOP为 1 时, RUN内部复位。
FULL	IN	全圆标志。 1: 全圆 0: 圆弧	BOOL	0/1	该参数在运行过程中不能修改。
A_AXIS	IN	插补 A 轴的轴号。 插补需要两个轴, 即虚拟的 A 轴和 B 轴。我们需要映射到实际输出的 0、1、2、3 轴上。该参数即可设定A轴映射到那一轴上。	BYTE	0~3	该参数在运行过程中不能修改。
B_AXIS	IN	插补 B 轴的轴号。 插补需要两个轴, 即虚拟的 A 轴和 B 轴。我们需要映射到实际输出的 0、1、2、3 轴上。该参数即可设定B轴映射到那一轴上。	BYTE	0~3	该参数在运行过程中不能修改。
A_SEC_POS	IN	决定圆弧的第二点的A轴坐标(此指令用于三点确定一圆弧或圆时, 此参数为运行方向上的第2点) (此为相对坐标)	DINT	-2147483648 ~ +2147483647	1、插补的起点坐标即为该圆弧的第1点。(相对指令起点坐标为(0, 0))。 2、该参数在运行过程中不能修改。 ● 单位: 脉冲, 与实际位移转换计算方法见附录 E3 提示 ②。 ● 插补的起点坐标默认为(0,0)。
B_SEC_POS	IN	决定圆弧的第二点的B轴坐标(此指令用于三点确定一圆弧或圆时, 此参数为运行方向上的第2点) (此为相对坐标)	DINT	-2147483648 ~ +2147483647	
A_END_POS	IN	决定圆弧的第三点的A轴坐标, 若FULL为0, 即也是终点坐标(此指令用于三点确定一圆弧或圆时, 此参数为运行方向上的第3点) (此为相对坐标)	DINT	-2147483648 ~ +2147483647	
B_END_POS	IN	决定圆弧的第三点的B轴坐标, 若FULL为0, 即也是终点坐标(此指令用于三点确定一圆弧或圆时, 此参数为运行方向上的第3点) (此为相对坐标)	DINT	-2147483648 ~ +2147483647	

MIN_SPEED	IN	长轴最小速度，即启动时或停止时的速度。单位：Hz	DWORD	0~200000	1、当速度小于5Hz时，脉冲输出关闭，即无输出。 2、该参数在运行过程中可以修改。 3、建议 MIN_SPEED 不要小于 500，否则在脉冲输出将完成减速结束时最小速度受限（通常限制为 500）。								
MAX_SPEED	IN	长轴最大速度，即运行中的最大速度。单位：Hz	DWORD	0~200000	4、将 MAX_SPEED 写 0，可实现软停功能（即输出脉冲减速停止），此时使能位 RUN 不复位；若将速度写回大值，可继续输出脉冲直至输出完成。								
TA		加速/减速时间。单位：ms	DWORD	0~10000	1、该参数在运行过程中可以修改。 2、加速度只在启动时和 TA/TD 变化时计算，计算方法详见附录 E3 提示 ⑤								
RUN	IN/OUT	运行使能位。 1: 有效 0: 无效	BOOL	0/1	1、只有 RUN =1 与 E_STOP=0 时才能运行。 2、当运行完成后，RUN 内部复位。 3、当 E_STOP 为 1 时，RUN 内部复位。								
STATUS	OUT	输出状态字节： <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> </table> Bit0: 参数配置错误标志 1—参数配置错误 0—参数配置正常 Bit1: 运行标志 1—正在运行，该指令正在输出脉冲，并且还没执行完。 0—不运行，因公共资源被其他指令占用，所以指令还未运行；或指令已经运行完毕。 Bit2: 完成标志 1—完成，指令执行完毕。	7	6	5	4	3	2	1	0	BYTE	0~255	Bit0: 1、只对轴参数配置错误进行判断。 2、MIN_SPEED/ MAX_SPEED 等参数不作报错，会自动设置成一个最接近的合理值。 3、若TA=0若没有设置最大加速度，则报参数故障；TD亦然。
7	6	5	4	3	2	1	0						

		<p>0—未完成, 执行未执行或者指令正在执行但没完成。</p> <p>Bit3: 忙标志</p> <p>1—忙标志有效, 圆弧插补模块或相应轴被其他指令占用。</p> <p>0—忙标志无效, 指令正在执行或此指令执行完成。</p> <p>Bit4~Bit7: 预留</p>			
ACT_A_POS	OUT	A 轴的当前位置(相对坐标, 本次调用实际输出脉冲数), 如果 A轴配给 0 轴, 那么该值就表示0轴的相对坐标。	DINT	-2147483648 ~ +2147483647	
ACT_B_POS	OUT	B 轴的当前位置(相对坐标, 本次调用实际输出脉冲数), 如果 B轴配给 1 轴, 那么该值就表示1轴的相对坐标。	DINT	-2147483648 ~ +2147483647	
ACT_SPEED	OUT	当前的实际速度	DWORD	0~200000	

MC_3P_CIRCLE_A (两轴三点画弧插补绝对运动指令)

① 函数名: MC_3P_CIRCLE_A



② 功能: 可在任意两轴之间进行圆弧插补 (设置点为绝对坐标)。

③ 参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
E_STOP	IN	<p>紧急停止位。</p> <p>1: 有效</p> <p>0: 无效</p>	BOOL	0/1	<p>1、只有 RUN =1 与 E_STOP =0 时才能运行。</p> <p>2、当 E_STOP</p>

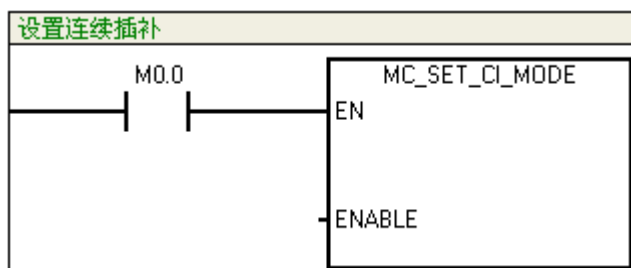
					为 1 时, RUN 内部复位。
FULL	IN	全圆标志。 1: 全圆 0: 圆弧	BOOL	0/1	该参数在运行过程中不能修改。
A_AXIS	IN	插补 A 轴的轴号。 插补需要两个轴, 即虚拟的 A 轴和 B 轴。我们需要映射到实际输出的 0、1、2、3 轴上。该参数即可设定 A 轴映射到那一轴上。	BYTE	0~3	该参数在运行过程中不能修改。
B_AXIS	IN	插补 B 轴的轴号。 插补需要两个轴, 即虚拟的 A 轴和 B 轴。我们需要映射到实际输出的 0、1、2、3 轴上。该参数即可设定 B 轴映射到那一轴上。	BYTE	0~3	该参数在运行过程中不能修改。
A_SEC_POS	IN	决定圆弧的第二点的 A 轴坐标 (此指令用于三点确定一圆弧或圆时, 此参数为运行方向上的第 2 点) (此为绝对坐标)	DINT	-2147483648 ~ +2147483647	1、插补的起点坐标即为该圆弧的第 1 点 (绝对指令起点坐标即为当前点的绝对坐标)。 2、该参数在运行过程中不能修改。 3、单位: 脉冲, 与实际位移转换计算方法见附录 E3 提示 ②。(绝对指令必须以脉冲增量来计算)。
B_SEC_POS	IN	决定圆弧的第二点的 B 轴坐标 (此指令用于三点确定一圆弧或圆时, 此参数为运行方向上的第 2 点) (此为绝对坐标)	DINT	-2147483648 ~ +2147483647	
A_END_POS	IN	决定圆弧的第三点的 A 轴坐标, 若 FULL 为 0, 即也是终点坐标 (此指令用于三点确定一圆弧或圆时, 此参数为运行方向上的第 3 点) (此为绝对坐标)	DINT	-2147483648 ~ +2147483647	
B_END_POS	IN	决定圆弧的第三点的 B 轴坐标, 若 FULL 为 0, 即也是终点坐标 (此指令用于三点确定一圆弧或圆时, 此参数为运行方向上的第 3 点) (此为绝对坐标)	DINT	-2147483648 ~ +2147483647	

MIN_SPEED	IN	长轴最小速度，即启动时或停止时的速度。单位：Hz	DWORD	0~200000	<p>1、当速度小于5Hz时，脉冲输出关闭，即无输出。</p> <p>2、该参数在运行过程中可以修改。</p> <p>3、建议 MIN_SPEED 不要小于 500，否则在脉冲输出将完成减速结束时最小速度受限（通常限制为 500）。</p>								
MAX_SPEED	IN	长轴最大速度，即运行中的最大速度。单位：Hz	DWORD	0~200000	<p>4、将 MAX_SPEED 写 0，可实现软停功能（即输出脉冲减速停止），此时使能位 RUN 不复位；若将速度写回大值，可继续输出脉冲直至输出完成。</p>								
TA		加速/减速时间。单位：ms	DWORD	0~10000	<p>1、该参数在运行过程中可修改。</p> <p>2、加速度只在启动时和 TA/TD 变化时计算，计算方法详见附录E3提示④。</p>								
RUN	IN/OUT	运行使能位。 1: 有效 0: 无效	BOOL	0/1	<p>1、只有 RUN =1 与 E_STOP=0 时才能运行。</p> <p>2、运行完成后，RUN内部复位。</p> <p>3、当 E_STOP 为 1 时，RUN 内部复位。</p>								
STATUS	OUT	<p>输出状态字节：</p> <table border="1" style="margin-left: 20px;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> </table> <p>Bit0: 参数配置错误标志 1—参数配置错误 0—参数配置正常</p> <p>Bit1: 运行标志 1—正在运行，该指令正在输出脉冲，并且还没执行完。</p>	7	6	5	4	3	2	1	0	BYTE	0~255	<p>Bit0:</p> <p>1、只对轴参数配置错误进行判断。</p> <p>2、MIN_SPEED /MAX_SPEED 等参数不作报</p>
7	6	5	4	3	2	1	0						

		<p>0—不运行，因公共资源被其他指令占用，所以指令还未运行；或者指令已经运行完毕。</p> <p>Bit2: 完成标志</p> <p>1—完成，指令执行完毕。</p> <p>0—未完成，执行未执行或者指令正在执行但没完成。</p> <p>Bit3: 忙标志</p> <p>1—忙标志有效，圆弧插补模块或相应的轴被其他指令占用。</p> <p>0—忙标志无效，指令正在执行或此指令执行完成。</p> <p>Bit4~Bit7: 预留</p>			<p>错，会自动设置成一个最接近的合理值。</p> <p>3、若TA=0若没有设置最大加速度，则报参数故障；TD亦然。</p>
ACT_A_POS	OUT	A 轴的当前位置(绝对坐标)，如果 A轴配给 0 轴，那么该值就表示 0轴的绝对坐标。	DINT	-2147483648 ~ +2147483647	
ACT_B_POS	OUT	B 轴的当前位置(绝对坐标)，如果 B轴配给 1轴，那么该值就表示 1轴的绝对坐标。	DINT	-2147483648 ~ +2147483647	
ACT_SPEED	OUT	当前的实际速度	DWORD	0~200000	

MC_SET_CI_MODE (设置连续插补指令)

① 函数名: MC_SET_CI_MODE



② 功能: 设置是否使能连续插补功能。

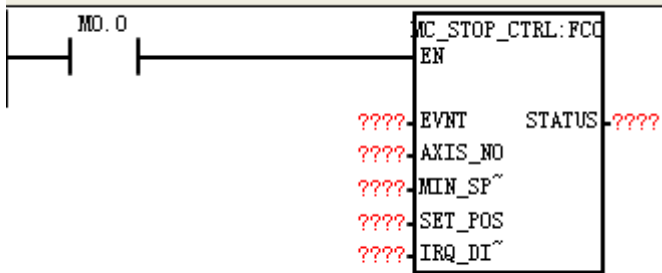
③ 参数

参数名	输入输出属性	参数描述	属性	数值范围	备注
ENABLE	IN	0: 关闭连续插补功能 1: 使能连续插补功能	BOOL	0~1	建议连续插补运行过程中不要修改。

MC_STOP_CTRL (控制停止指令)

① 函数名: MC_STOP_CTRL

网络注释



② 功能：通过一个中断事件，触发一个轴停止，使得正在运行的轴按设定参数执行停止过程。支持位置指令和速度指令，任何插补指令均不支持这种停止方式。仅支持 IO 中断号，不支持通信类中断号。

③ 参数

参数名	输入输出属性	参数描述	属性	数值范围	备注
EVNT	IN	中断事件号	BOOL	0~33	除掉中间的通信中断、定时中断，详情参见 CTH200 支持的中断事件
AXIS_NO	IN	设置轴号	BYTE	0~3	
MIN_SPEE	IN	最小速度，启动或停止时的速度	BYTE	0~200000	单位：HZ，建议最小速度不要小于 500，否则在脉冲输出将完成减速结束时最小速度受限（通常限制为 500）
SET_POS	IN	设定输出脉冲数（相对坐标）	DWORD	0~+2147483647	
IRQ_DIRECT	IN	该参数仅仅对中断号为 27, 14, 17, 30起作用，当中断号设置为以上四个值中的一个时可以给 IRQ_DIRECT配置为0，表示高速计数器从增计数变为减计数，然后开始执行电机缓停，配置为1表示为电机从减计数变为增计数，然后执行电机缓停，如果设置的非1则默认使用下降沿中断。	DWORD		
STATUS	OUT	脉冲输出状态	BYTE		具体见下表

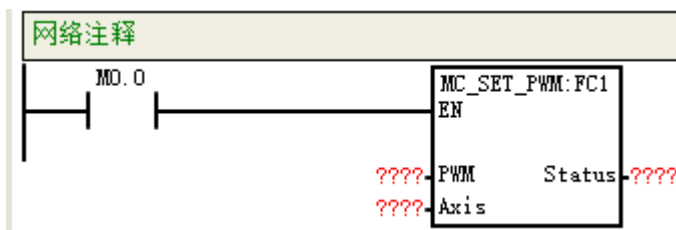
顺序：bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0

Bit7	保留
Bit6	保留
Bit5	保留
Bit4	保留
Bit3	1: 缓停指令个数超限； 0: 缓停指令个数正常。

Bit2	1: 执行完缓停; 0: 没有执行缓停或者正在缓停中。
Bit1	1: 电机正在缓停; 0: 执行完缓停或者是没有启动缓停;
Bit0	轴号状态位。1: 轴号设置错误; 0: 轴号设置正确

MC_SET_PWM (占空比设置指令)

① 函数名: MC_SET_PWM



② 功能: 该指令用于调节运控输出脉冲的占空比, 只有 V5 平台 PLC 硬件支持该功能, 固件版本 V2.73 以上支持该功能。该指令只用于 PLC 226EH。

该指令执行立即生效, 如果此时脉冲正在输出, 则占空比在调用指令之后立即修改。

③ 参数

参数名	输入输出属性	参数描述	属性	数值范围	备注
PWM	IN	输出占空比	REAL	0.0~1.0	范围0.0至1.0, 对应0%至100%占空比。
Axis	IN	轴号, 需要调节占空比的轴号范围跟PLC型号最大支持的轴数相同。	BYTE		
Status	OUT	状态字, 0-初始状态, 1-完成, 2-占空比超范围, 3-轴号错误	BYTE		

提示

提示 ① 当某轴调用运动控制指令后, 那么该轴只有在重新上电复位或 CPU 停机时, 才可以恢复普通 IO 的功能。

提示 ② 直线插补指令 (A_POS、B_POS), 圆弧插补指令中 (A_END_POS、B_END_POS、RADIUS) 计算方法:

POS → 插补指令中需填入的脉冲个数;

L → 坐标上实际点的绝对值, 单位 mm;

S → 运动轴上丝杆导程, 单位 mm;

M → 步进驱动器的细分或伺服驱动的分辨率 (即电机转动一圈需要的脉冲个数);

计算公式: $POS = L * M \div S$ (单位: 脉冲数)

提示 ③ 当 $TA \neq 0$, 加速度 = $(MAX_SPEED - MIN_SPEED) / TA$ (若设置有最大加速度, 则受限于最大加速度); 若 $TA = 0$, 则采用指令 MC_SET_MAX_ACCELE 设置的最大加速度, 若没有设置最大加速度, 则报参数故障; TD 亦然。对于双轴指令, 若两轴均设置了最大加速度, 则采用其中的较小值作为系统加速度。

F 回原功能

通常运动系统中都要用一个位置传感器设置一个位置参考点，即原点位置，以便于进行精确的位置控制。运动控制系统动作时通常会执行寻找该原点的动作叫系统坐标位置归零，即回原。在进行精确的运动控制之前，需要设定运动坐标系的原点。

F.1 回原模式

CTH200 系列控制器可使用以下 14 种回原模式，用户可根据对精度的要求及实际应用需求选择。

回原模式	说明
1	参考负向原点开关和 Z 相信号的原点模式
2	参考正向原点开关和 Z 相信号的原点模式
3	只参考负向原点开关的原点模式
4	只参考正向原点开关的原点模式
5	只参考 Z 相信号的原点模式（负向回原）
6	只参考 Z 相信号的原点模式（正向回原）
7	参考原点开关、Z 相信号和正限位的原点模式（采正向原点开关左边沿以左的 Z 相信号）
8	参考原点开关、Z 相信号和正限位的原点模式（采正向原点开关左边沿以右的 Z 相信号）
9	参考原点开关、Z 相信号和正限位的原点模式（采正向原点开关右边沿以左的 Z 相信号）
10	参考原点开关、Z 相信号和正限位的原点模式（采正向原点开关右边沿以右的 Z 相信号）
11	参考原点开关、Z 相信号和负限位的原点模式（采正向原点开关右边沿以右的 Z 相信号）
12	参考原点开关、Z 相信号和负限位的原点模式（采正向原点开关右边沿以左的 Z 相信号）
13	参考原点开关、Z 相信号和负限位的原点模式（采正向原点开关左边沿以右的 Z 相信号）
14	参考原点开关、Z 相信号和负限位的原点模式（采正向原点开关左边沿以左的 Z 相信号）

F.2 回原动作

无论机械初始处于什么位置，当设备（原点开关、正向行程限位开关、负向行程限位开关）安装完好，伺服所寻找的设备原点总是唯一的。以下各模式示意图中的竖线“|”代表机械初始位置，圆圈“⊗”代表原点位置。

提示

轴号与外部复位 IO 信号（如回原 Z pulse）的对应关系：

轴 0 —— I0.2（HSC0，SM37.0） 轴 1 —— I1.0（HSC1，SM47.0）

轴 2 ——I1.4 (HSC2, SM57.0) 轴 3 ——I0.5 (HSC4, SM147.0)

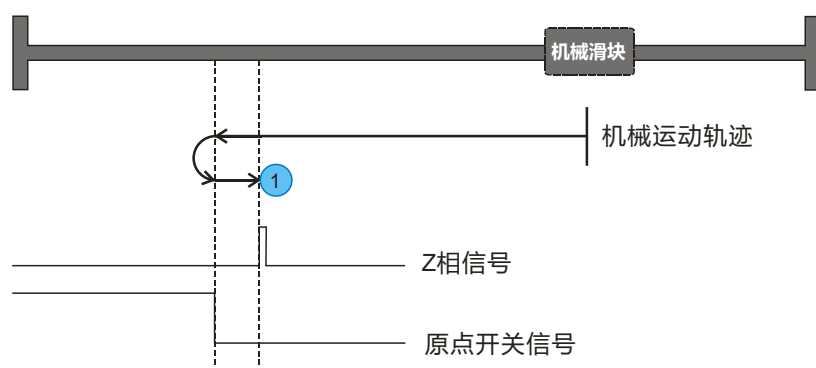
若回原模式以原点开关为参考时（回原模式 3 或 4），必须将原点开关信号（指令的 **HOMING_SW** 参数）接至 CPU 226H 的上述对应点，否则无法找到原点。

无论机械初始处于什么位置，当设备（原点开关、正向行程限位开关、负向行程限位开关）安装完好，伺服所寻找的设备原点总是唯一的。以下各模式示意图中的竖线“|”代表机械初始位置，圆圈“⊗”代表原点位置。

回原模式 1：DI 触发回原（仅 H224X/ H226XL/ H228XL/H226XM/H226IM/H226IL/H226IH 支持该模式）

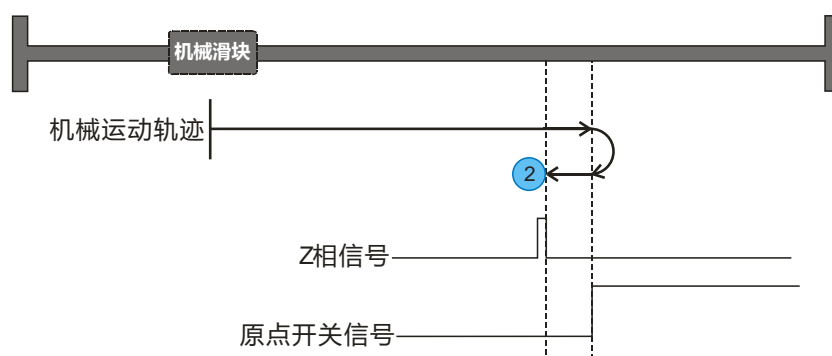
将外部 DI 引脚配置成“原点开关信号”，触发此信号（上升沿），则复位 P216（用户位置坐标）为原点 0（此模式为特殊回原模式，不输出回原完成信号）。

回原模式 1：参考负向原点开关和 Z 相信号的原点模式



原点开关位于机械负方向。机械往原点开关方向运动，在检测到原点开关后减速停止，再反转退出原点开关，找电机的下一个 Z 相信号并将该位置记录为原点，电机立即停止。

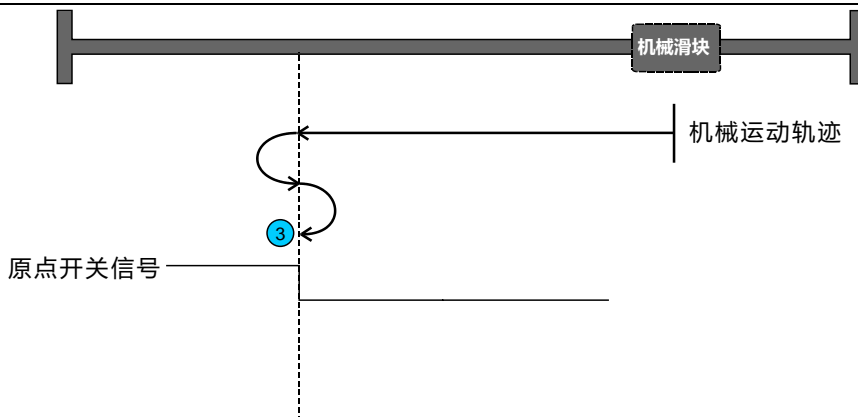
回原模式 2：参考正向原点开关和 Z 相信号的原点模式



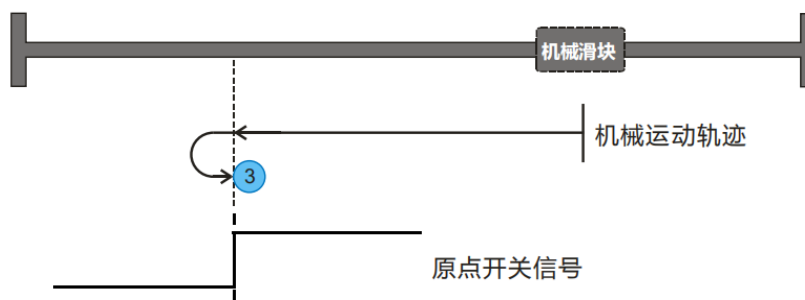
原点开关位于机械正方向。机械往原点开关方向运动，在检测到原点开关后减速停止，再反转退出原点开关，找电机的下一个 Z 相信号并将该位置记录为原点，电机立即停止。

回原模式 3：参考负向原点开关的原点模式（请将原点开关信号接至轴对应的 Z 相信号点）

1) 当原点开关信号类型设定为高电平有效时。原点位于机械负方向，机械往原点开关方向运动，在检测到原点开关时，先减速至停止，再反转退出原点开关，在检测到原点开关下降沿时，再次减速停止并换向，从而再次接近原点开关，在开关产生上升沿时停止即找到原点，如下图：

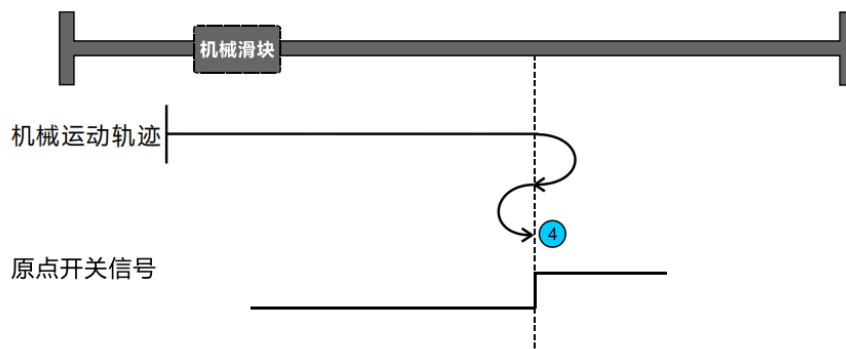


2) 当原点开关信号类型设定为低电平有效时。原点开关位于机械负方向，机械往原点开关方向运动，在检测到原点开关时先减速至停止，反转退出原点开关，在退出原点开关产生上升沿时停止即找到原点，如下图：

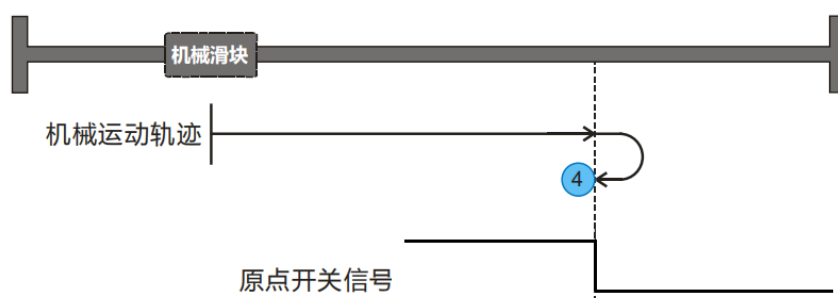


回原模式 4: 参考正向原点开关的原点模式（请将原点开关信号接至轴对应的 Z 相信号点）

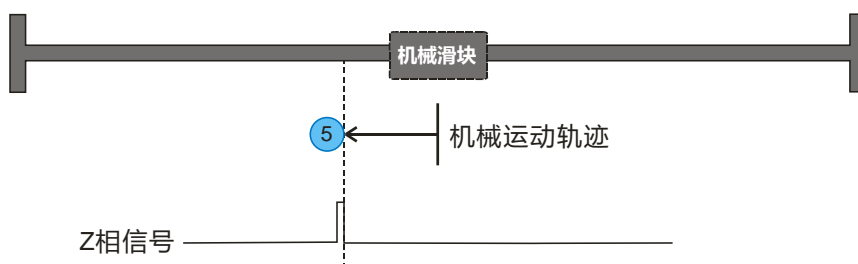
1) 当原点开关信号类型设定为高电平有效时。原点开关位于机械正方向，机械往原点开关方向运动，在检测到原点开关时先减速停止，再反转退出原点开关，在出现原点开关下降沿时，再次减速停止并换向，从而再次接近原点开关，在开关产生上升沿时停止即找到原点，如下图：



2) 当原点开关信号类型设定为低电平有效时。原点开关位于机械正方向，机械往原点开关方向运动，在检测到原点开关时先减速停止，再反转退出原点开关，在退出原点开关产生上升沿时停止即找到原点，如下图：

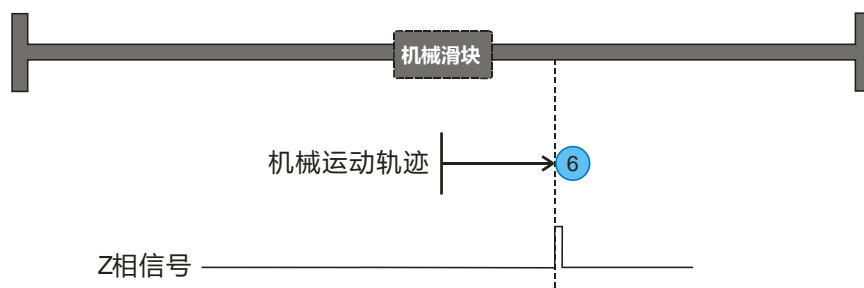


回原模式 5: 参考 Z 相信号的原点模式 (负向回原)



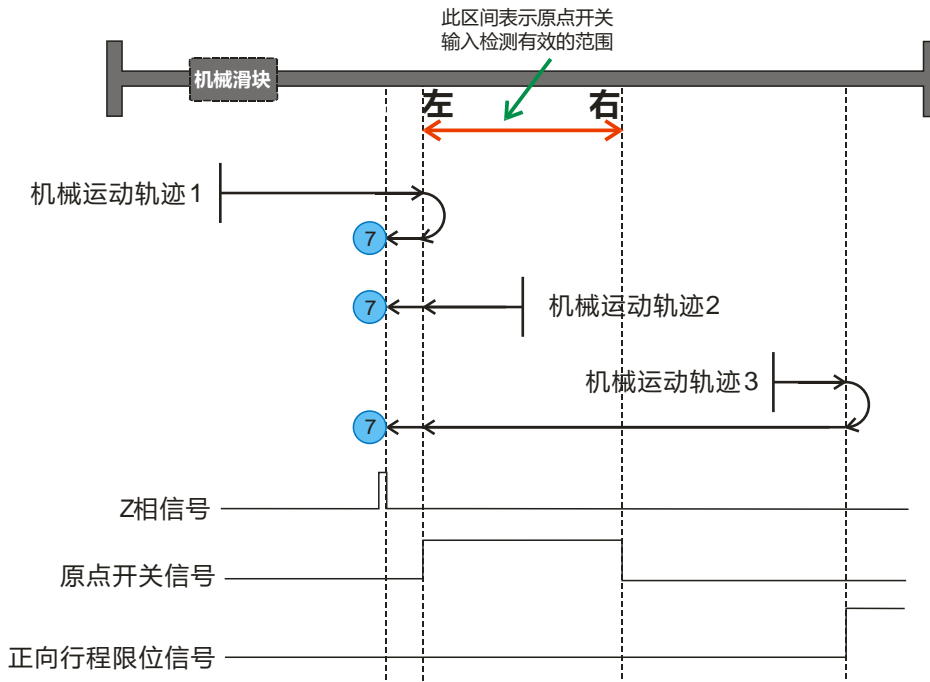
电机从当前位置往负方向运动，找到下一个 Z 相信号时将该位置记录为原点。

回原模式 6: 参考 Z 相信号的原点模式 (正向回原)



电机从当前位置往正方向运动，找到下一个 Z 相信号时将该位置记录为原点。

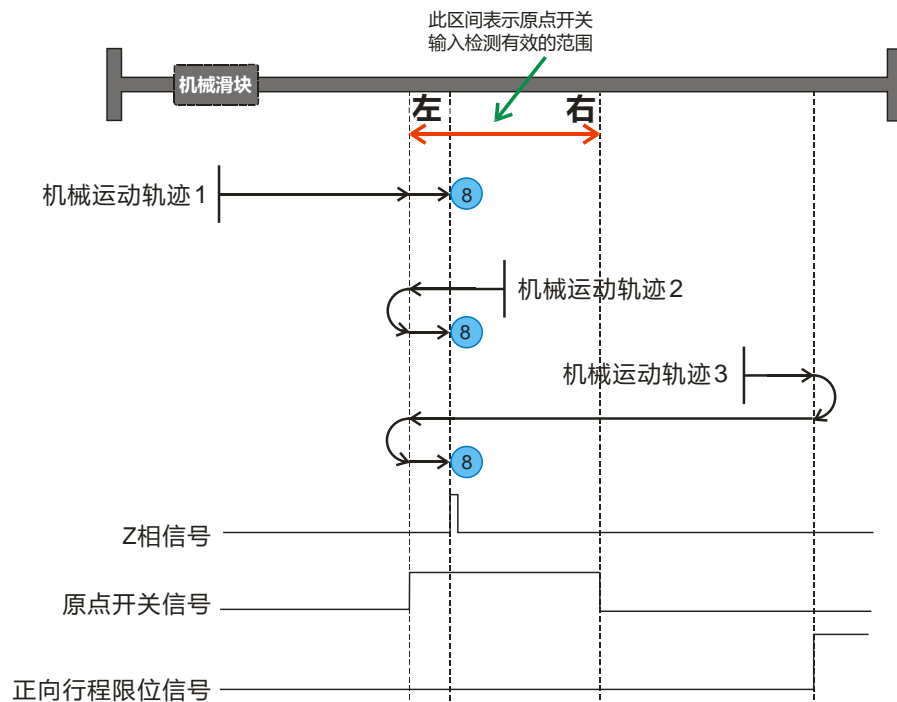
回原模式 7: 参考原点开关、Z 相信号和正限位的原点模式 (采正向原点开关左边沿以左的 Z 相信号)



如上图所示，机械滑块往正限位方向（正方向）滑行，Z 相信号处于原点开关信号左边沿以左的位置，即原点开关信号有效范围外。

当机械处于原点开关范围内（机械运动轨迹 2），则直接往负方向运行即可寻原点；当机械处于原点开关范围外（机械运动轨迹 1 和机械运动轨迹 3），机械往限位开关方向恒定运行（正方向），根据检测到原点开关与限位开关的先后可知运动轨迹，从而可寻原点。

回原模式 8：参考原点开关、Z 相信号和正限位的原点模式（采正向原点开关左边沿以右的 Z 相信号）

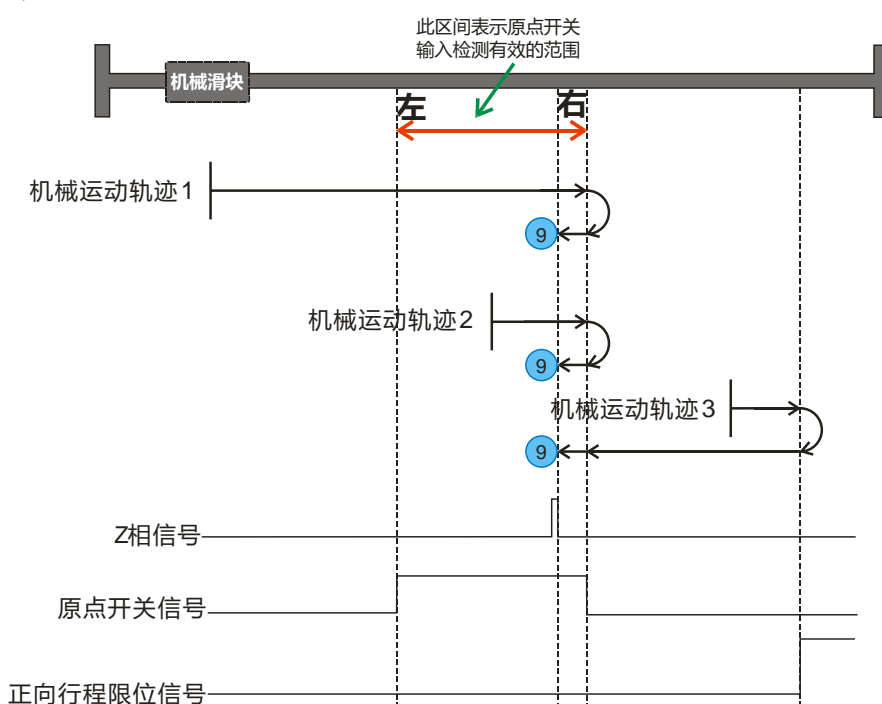


如上图所示，机械滑块往正限位方向（正方向）滑行，Z 相信号处于原点开关信号左边沿以右的位置，即原点开关信号有效范围内。

当机械处于原点开关范围内（机械运动轨迹 2），则直接往负方向运行即可寻原点；当机械处于原

点开关范围外（机械运动轨迹 1 和机械运动轨迹 3），机械往限位开关方向恒定运行（正方向），根据检测到原点开关与限位开关的先后可知运动轨迹，从而可寻原点。

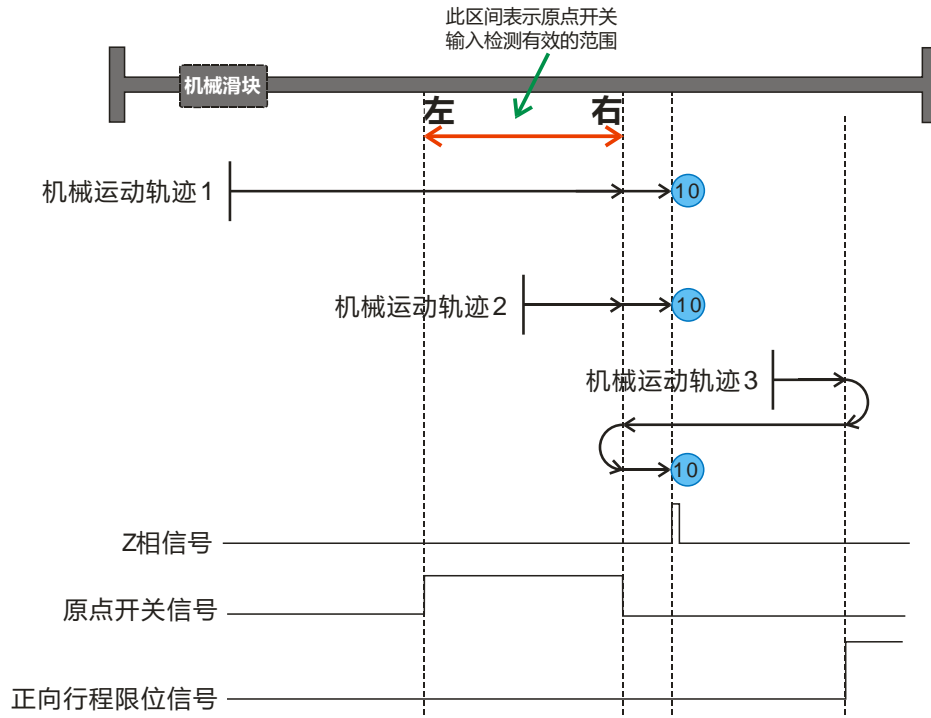
回原模式 9：参考原点开关、Z 相信号和正限位的原点模式（采正向原点开关右边沿以左的 Z 相信号）



如上图所示，机械滑块往正限位方向（正方向）滑行，Z 相信号处于原点开关信号右边沿以左的位置，即原点开关信号有效范围内。

当机械处于原点开关范围内（机械运动轨迹 2），则直接往正方向运行即可寻原点；当机械处于原点开关范围外（机械运动轨迹 1 和机械运动轨迹 3），机械往限位开关方向恒定运行（正方向），根据检测到原点开关与限位开关的先后可知运动轨迹，从而可寻原点。

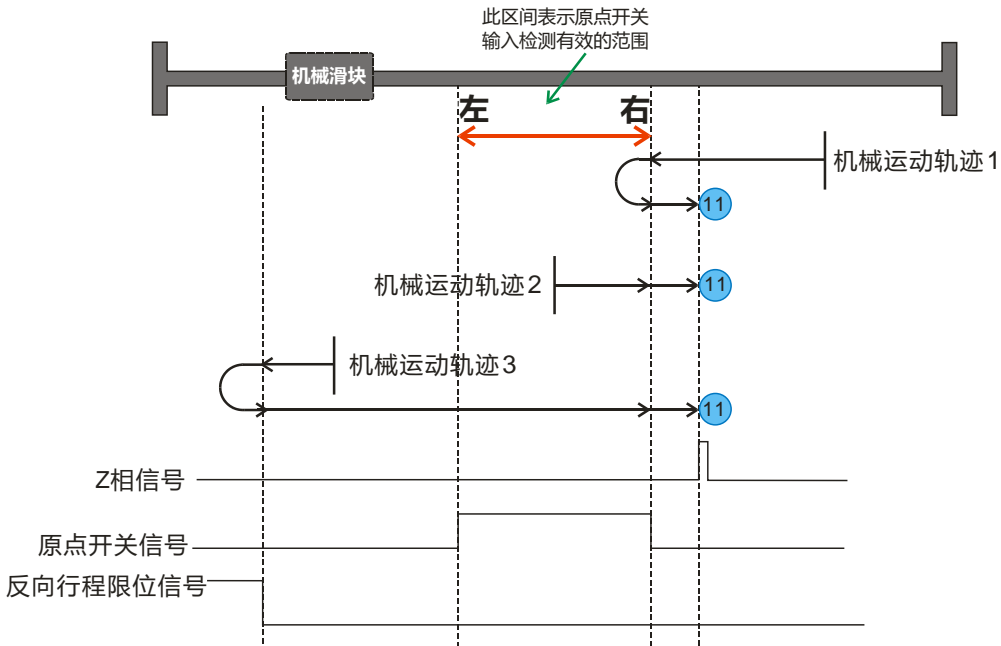
回原模式 10：参考原点开关、Z 相信号和正限位的原点模式（采正向原点开关右边沿以右的 Z 相信号）



如上图所示，机械滑块往正限位方向（正方向）滑行，Z 相信号处于原点开关信号右边沿以右的位置，即原点开关信号有效范围外。

当机械处于原点开关范围内（机械运动轨迹 2），则直接往正方向运行即可寻原点；当机械处于原点开关范围外（机械运动轨迹 1 和机械运动轨迹 3），机械往限位开关方向恒定运行（正方向），根据检测到原点开关与限位开关的先后可知运动轨迹，从而可寻原点。

回原模式 11：参考原点开关、Z 相信号和负限位的原点模式（采正向原点开关右边沿以右的 Z 相信号）

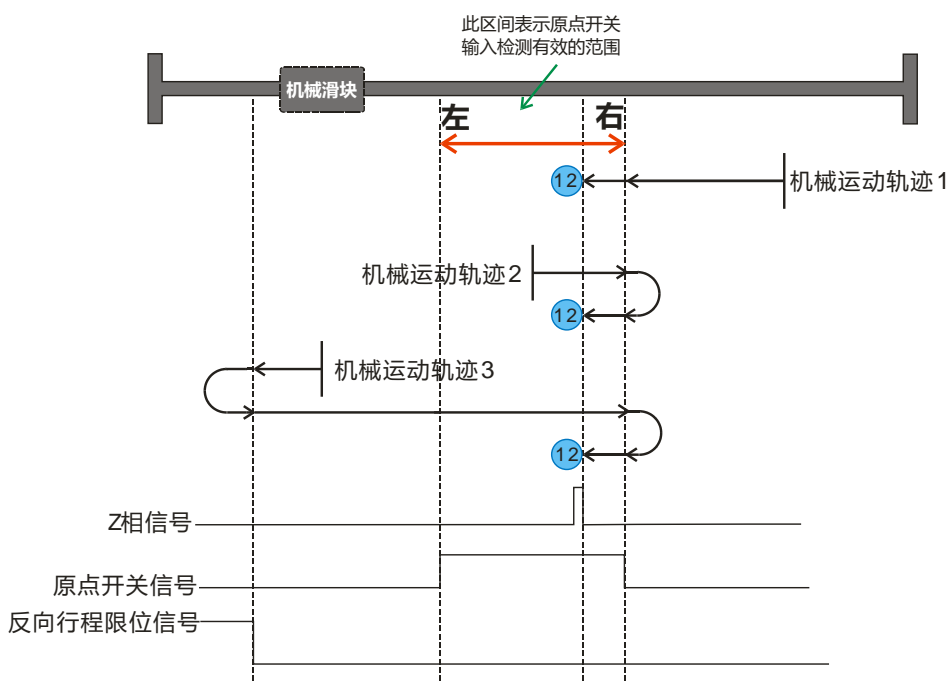


如上图所示，机械滑块往负限位方向（负方向）滑行，Z 相信号处于原点开关信号右边沿以右的位置，即原点开关信号有效范围外。

当机械处于原点开关范围内（机械运动轨迹 2），则直接往正方向运行即可寻原点；当机械处于原

点开关范围外（机械运动轨迹 1 和机械运动轨迹 3），机械往限位开关方向恒定运行（负方向），根据检测到原点开关与限位开关的先后可知运动轨迹，从而可寻原点。

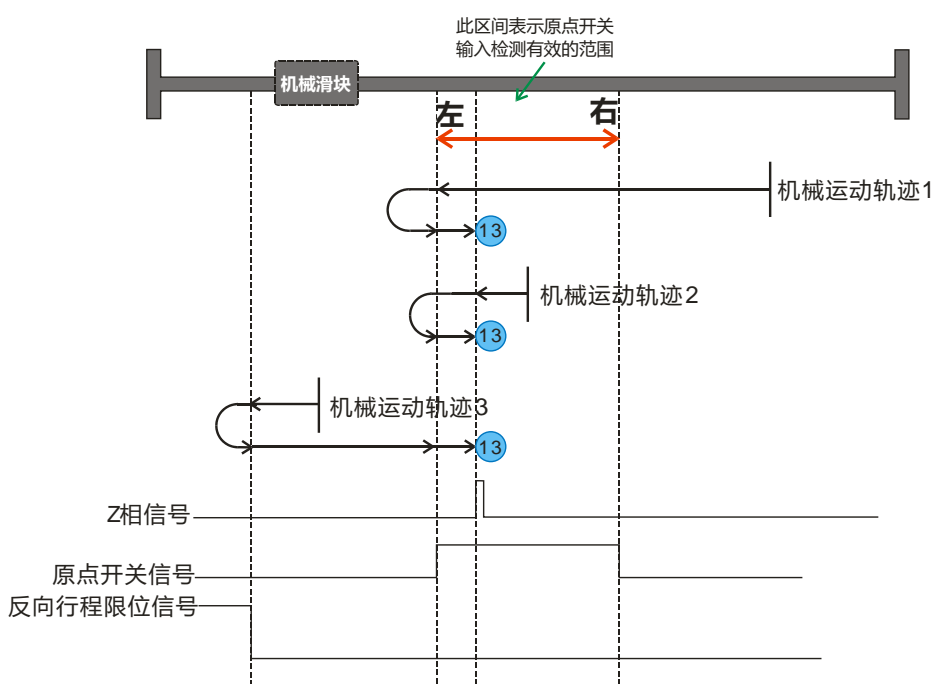
回原模式 12: 参考原点开关、Z 相信号和负限位的原点模式（采正向原点开关右边沿以左的 Z 相信号）



如上图所示，机械滑块往负限位方向（负方向）滑行，Z 相信号处于原点开关信号右边沿以左的位置，即原点开关信号有效范围内。

当机械处于原点开关范围内（机械运动轨迹 2），则直接往正方向运行即可寻原点；当机械处于原点开关范围外（机械运动轨迹 1 和机械运动轨迹 3），机械往限位开关方向恒定运行（负方向），根据检测到原点开关与限位开关的先后可知运动轨迹，从而可寻原点。

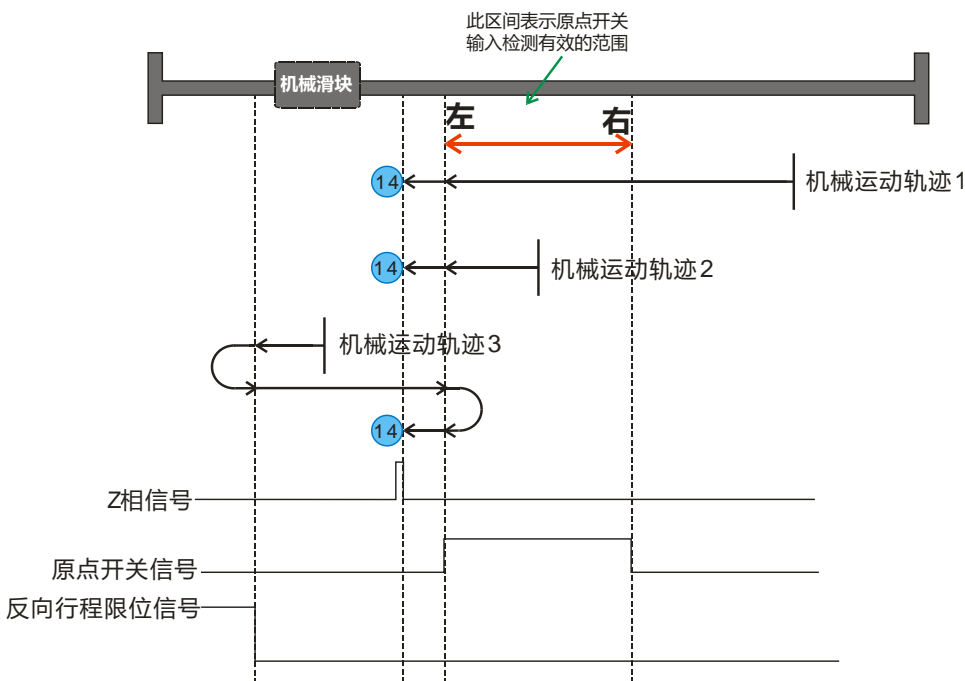
回原模式 13: 参考原点开关、Z 相信号和负限位的原点模式（采正向原点开关左边沿以右的 Z 相信号）



如上图所示，机械滑块往负限位方向（负方向）滑行，Z 相信号处于原点开关信号左边沿以右的位置，即原点开关信号有效范围内。

当机械处于原点开关范围内（机械运动轨迹 2），则直接往负方向运行即可寻原点；当机械处于原点开关范围外（机械运动轨迹 1 和机械运动轨迹 3），机械往限位开关方向恒定运行（负方向），根据检测到原点开关与限位开关的先后可知运动轨迹，从而可寻原点。

回原模式 14: 参考原点开关、Z 相信号和负限位的原点模式（采正向原点开关左边沿以左的 Z 相信号）



如上图所示，机械滑块往负限位方向（负方向）滑行，Z 相信号处于原点开关信号左边沿以左的位置，即原点开关信号有效范围外。

当机械处于原点开关范围内（机械运动轨迹 2），则直接往负方向运行即可寻原点；当机械处于原点开关范围外（机械运动轨迹 1 和机械运动轨迹 3），机械往限位开关方向恒定运行（负方向），根据检测到原点开关与限位开关的先后可知运动轨迹，从而可寻原点。

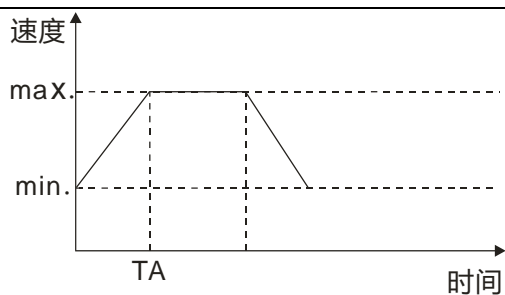
G 位置控制

位置控制是指控制某个物体完成从当前位置运动到另一个位置的指定位移，一般称为点对点运动或定长运动。设置最大速度、最小速度、目标位置等参数后执行位置控制指令，当前位移等于设定位移时，CPU 将减速停止。

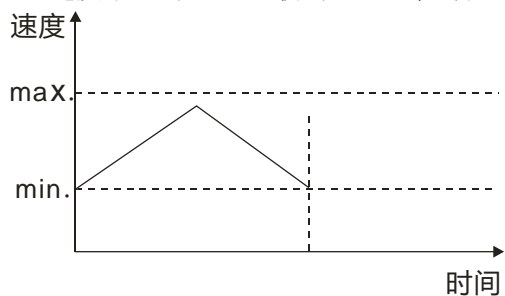
G.1 位置控制类型

位置控制中以位置为目标，达到位置即停止，表示为 Point to Point。以下简称为“PTP 控制”。

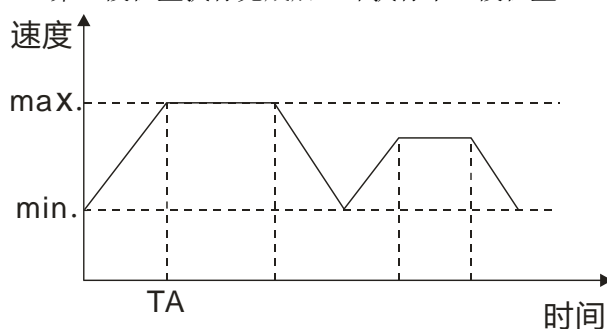
1) 速度达到 max. 后按照该速度匀速运行，收到 STOP 信号后，减速至 min. 停止，即完成目标位置。



2) 速度未达到 max., 收到 STOP 信号后, 减速至 min. 停止, 即完成目标位置。



3) 第一段位置执行完成后立即执行下一段位置。



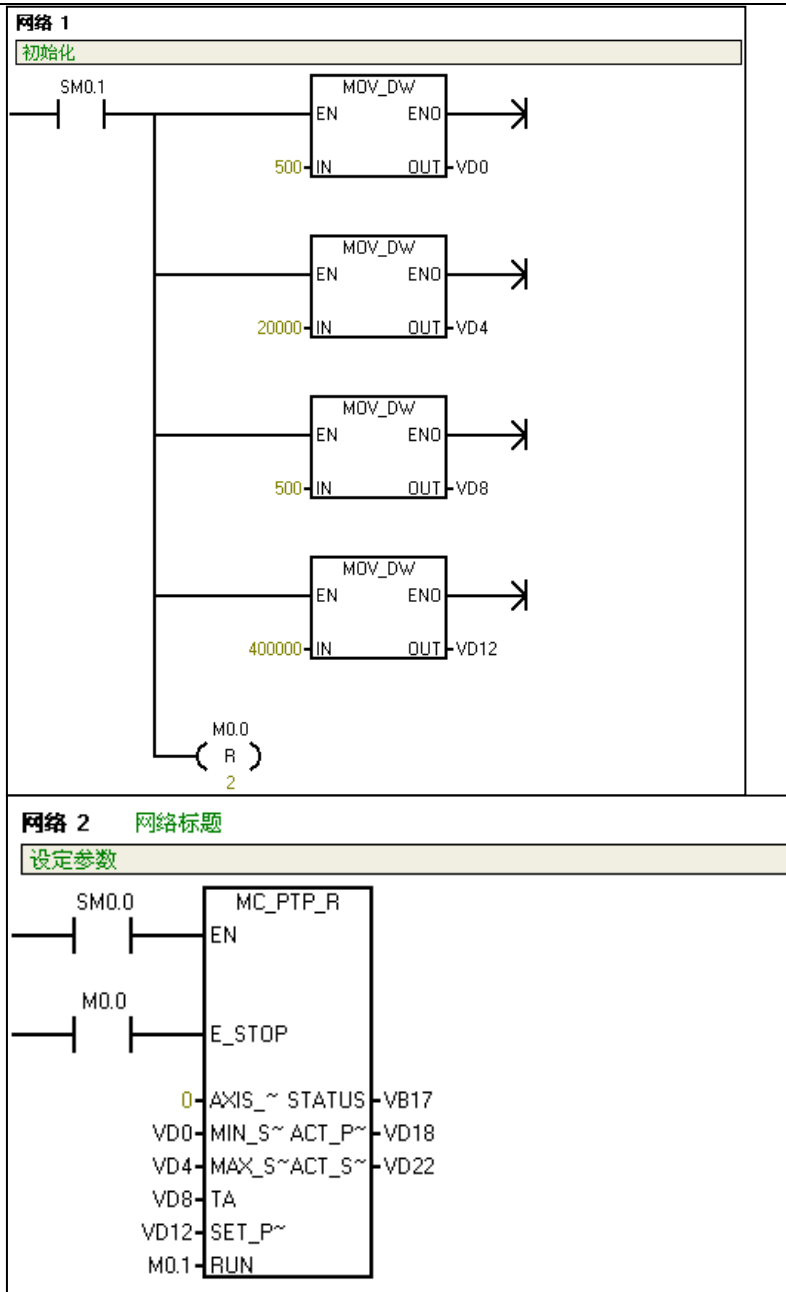
G.2 单轴相对运动指令使用

程序注释

功能：用作单轴点对点控制（单轴定长驱动）。

调用一次可输出固定脉冲，通过最大、最小速度和加减速时间的设定，输出的脉冲在启动时会逐渐的加速到最大的速度，当脉冲数快要跑完时，脉冲的频率会自动减下来，以防止在启动或停止时的机器的惯性太大而引起振动或卡死。

M0.0 ----- 紧急停止位；
 轴号为0-----Q0.0脉冲输出、Q0.1方向输出；
 VD0 ----- 启动/停止速度
 VD4 ----- 加速完成后的正常速度；
 VD8 ----- 加速时间（ms）；
 VD12----- 要输出的脉冲数；
 VB17----- 输出状态字节；
 VD18----- 输出脉冲个数；
 VD22----- 当前输出脉冲速度（频率）。



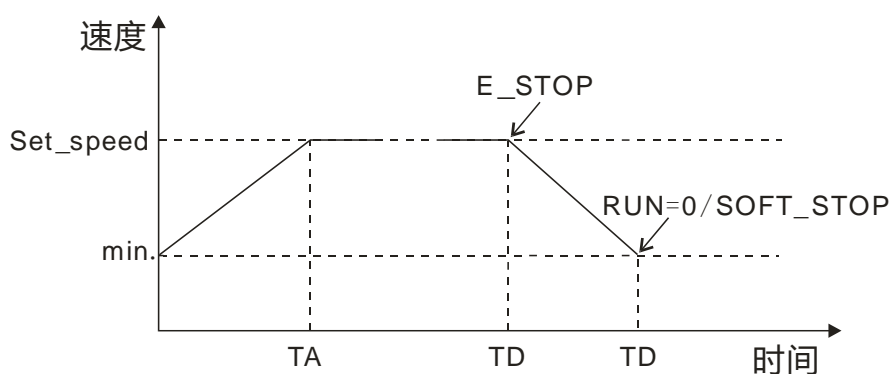
H 速度控制

速度控制是指电机从起始速度开始运行，加速至指定速度连续运动。只有当接收到停止命令后，才减速直至停止（也可设为立即停止）。该模式下不控制运动距离，它的主要用途是：寻找机械原点、速度控制。

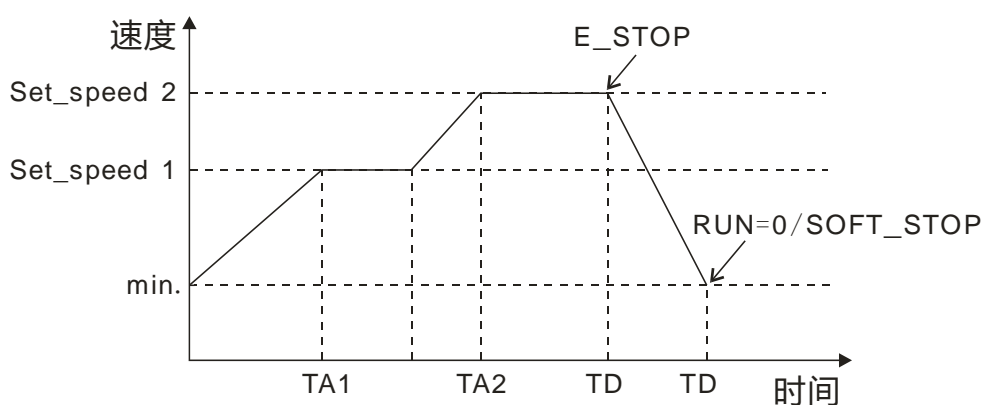
H.1 速度控制类型

速度控制中以速度为目标，达到该速度后匀速运行，表示为 **Set Speed Point**。以下简称为“SPD 控制”。

- 1) 速度达到目标速度 `set_speed` 后按照该速度匀速运行，若收到 `SOFT_STOP` 信号/`RUN=0`，减速至 `min.` 停止；若收到 `E_STOP` 命令后立即停止。



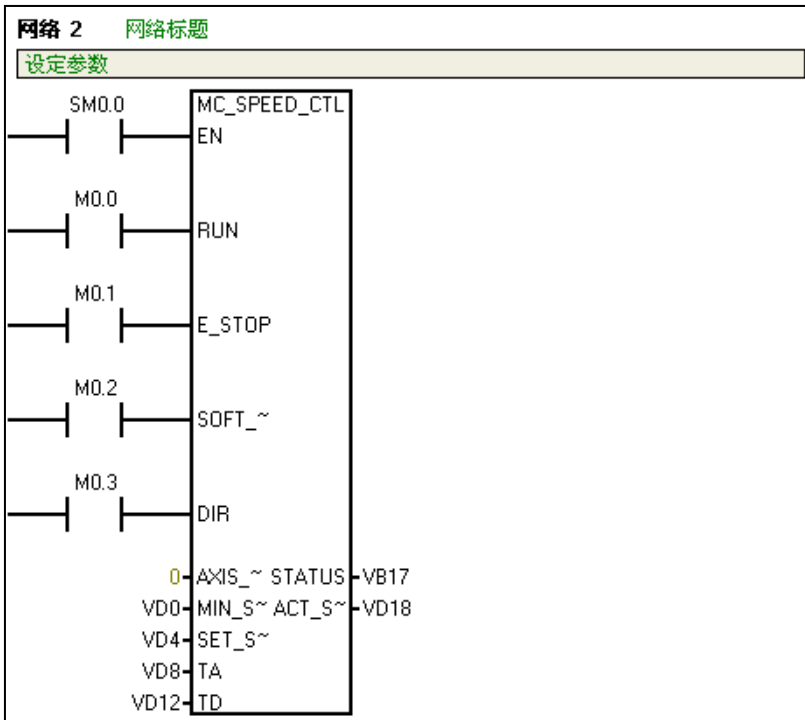
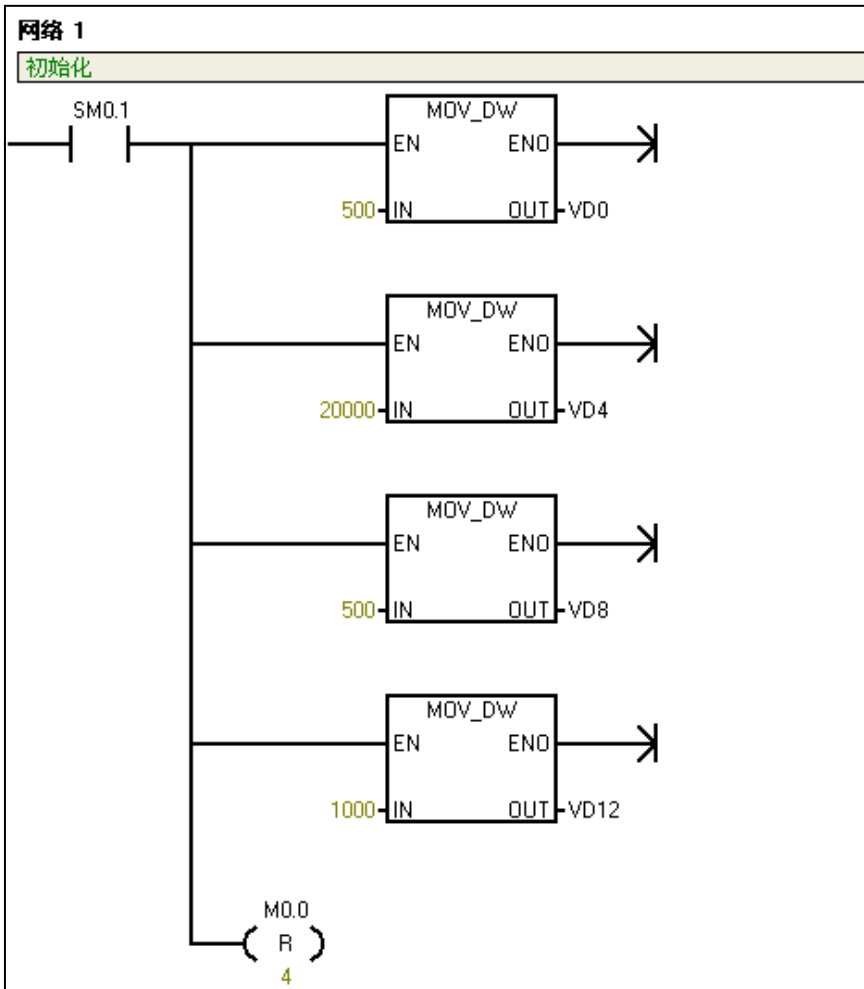
2) 速度达到目标速度 `set_speed 1` 后按照该速度匀速运行，运行过程中目标速度 `set_speed 1` 被修改为 `set_speed 2`，由当前速度加速运行到新的目标速度 `set_speed 2` 后，若收到 `SOFT_STOP` 信号/`RUN=0`，减速至 `min.` 停止；若收到 `E_STOP` 命令后立即停止。



H.2 单轴速度运动指令使用

程序注释
 功能：控制单轴输出脉冲的频率，可任意时候改变输出脉冲的频率（速度）。
 当接收到软停止命令时，会自动减速停止。当收到紧急停止命令时，会马上停止脉冲输出，不经过减速。

M0.0 ----- 运行使能位；
 M0.1 ----- 紧急停止位；
 M0.2 ----- 软停止位；
 M0.3 ----- 脉冲方向位（0为反方向，1为正方向）；
 轴号为0-----Q0.0脉冲输出、Q0.1方向输出；
 VD0 ----- 启动/停止速度
 VD4 ----- 加速完成后的正常速度；
 VD8 ----- 加速时间（ms）；
 VD12 ----- 减速时间（ms）；
 VB17 ----- 输出状态字节；
 VD18 ----- 当前输出脉冲速度（频率）。



I 直线插补

位置控制中，以直线为轨迹对实现群组化的 X 轴、Y 轴这 2 轴、或者包括 Z 轴在内的 3 轴的电机动作进行控制的插补控制。指定方法包括合成速度指定、长轴速度指定两种。

I.1 两轴直线插补运动指令使用

程序注释

功能：可在任意两轴之间、平面上任意区域内进行直线插补功能。

M0.0 ----- 紧急停止位；

轴号为1-----Q0.2脉冲输出、Q0.3方向输出；

轴号为3-----Q0.6脉冲输出、Q0.7方向输出；

VD0 ----- 长轴启动/停止速度

VD4 ----- 长轴加速完成后的正常速度；

VD8 ----- 加减速时间；

VD12 ----- 虚拟A轴的终点(相对坐标)；

VD16 ----- 虚拟B轴的终点(相对坐标)；

M0.1 ----- 运行使能位；

VB21 ----- 输出状态字节；

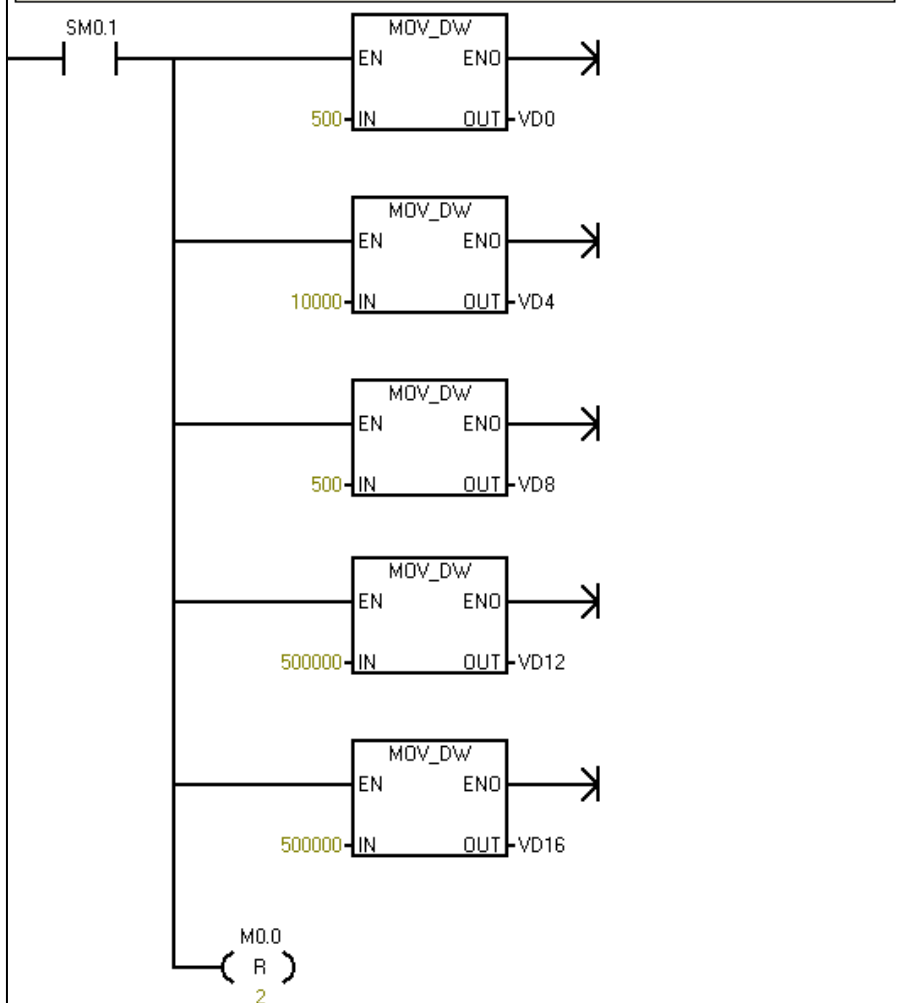
VD22 ----- A轴的当前位置（相对坐标，本次调用实际输出脉冲数）；

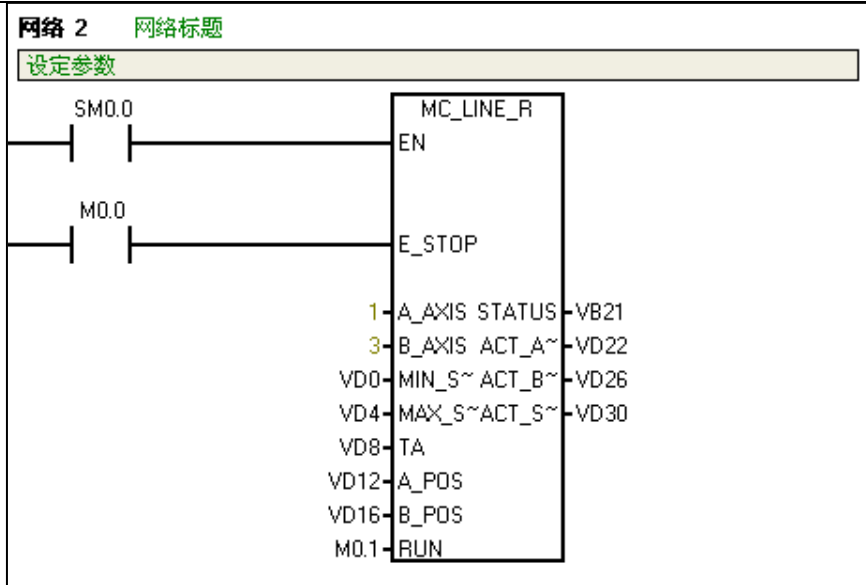
VD26 ----- B轴的当前位置（相对坐标，本次调用实际输出脉冲数）；

VD30 ----- 当前的实际速度（频率）。

网络 1

初始化





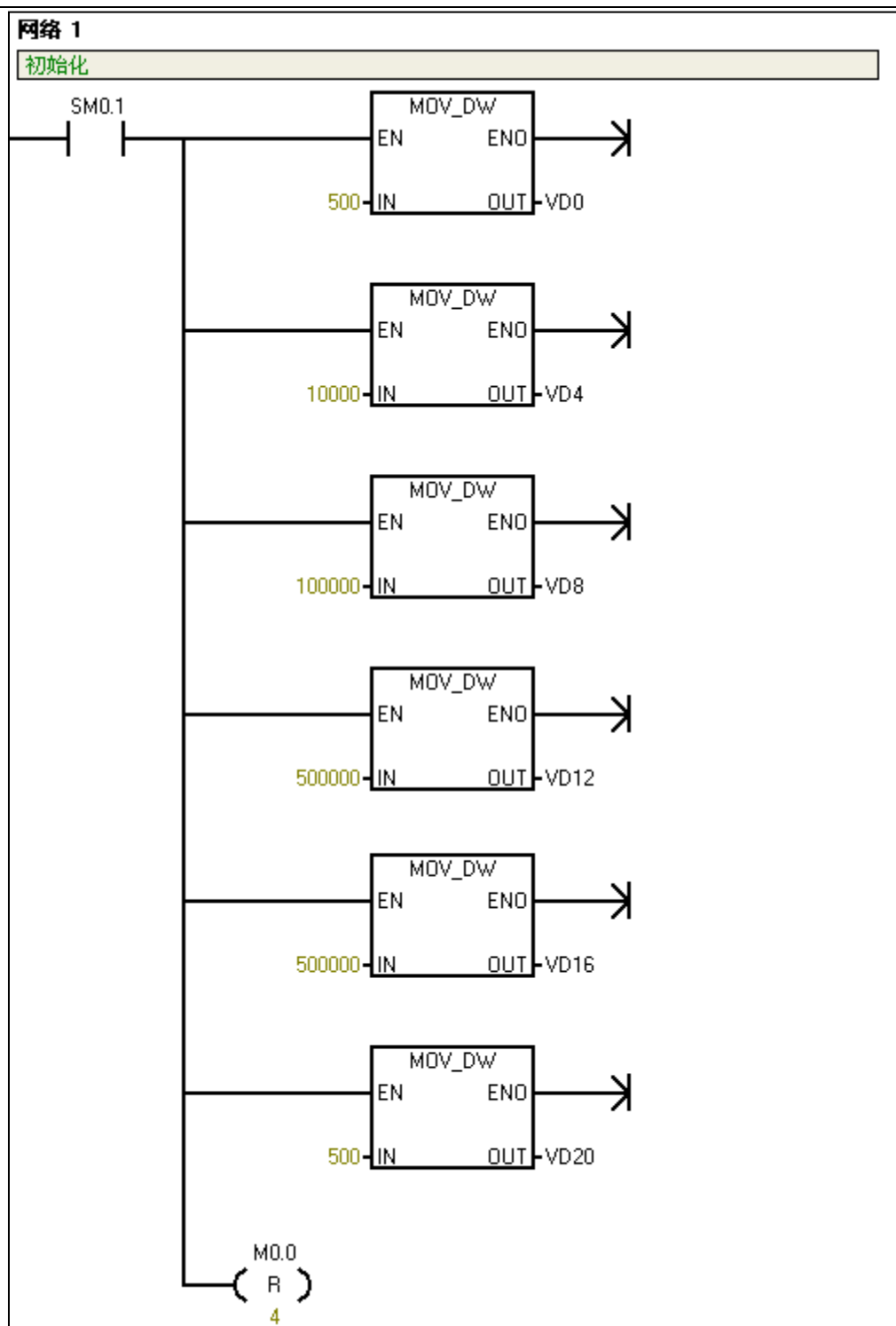
J 圆弧插补

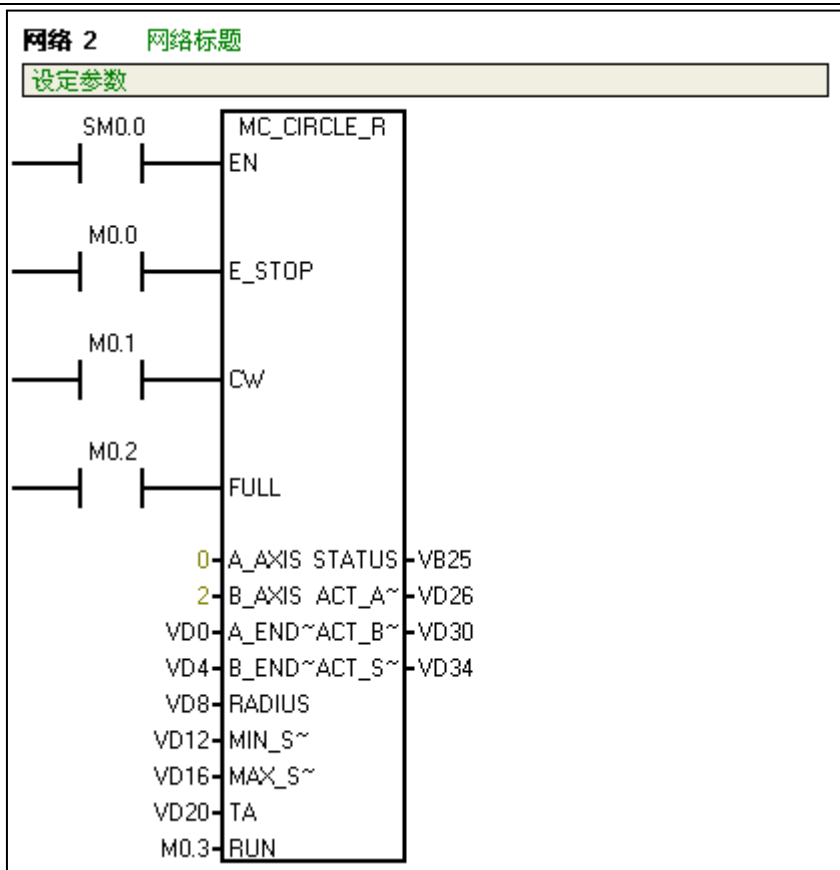
圆弧插补从当前位置开始，根据所指定的圆心、终点位置、圆弧半径（正数或负数）及插补的方向（按顺时针或逆时针）来进行。

位置控制中，以圆弧为轨迹对实现群组化的 X 轴、Y 轴这 2 轴的电机动作进行控制的插补控制。圆弧的指定方法包括中心点指定、通过点指定两种。

J.1 两轴圆弧插补运动指令使用

程序注释	
功能：可在任意两轴之间、平面上任意区域内进行圆弧插补功能。	
M0.0	紧急停止位；
M0.1	顺时针或逆时针插补标志位（1、顺时针、0、逆时针）；
M0.2	全圆标志位（1、全圆、0-圆弧）
轴号为0	Q0.0脉冲输出、Q0.1方向输出；
轴号为2	Q0.4脉冲输出、Q0.5方向输出；
VD0	长轴启动/停止速度
VD4	长轴加速完成后的正常速度；
VD8	圆弧的半径；
VD12	虚拟A轴的终点(相对坐标)；
VD16	虚拟B轴的终点(相对坐标)；
VD20	加减速时间；
M0.3	运行使能位；
VB25	输出状态字节；
VD26	A轴的当前位置（相对坐标，本次调用实际输出脉冲数）；
VD30	B轴的当前位置（相对坐标，本次调用实际输出脉冲数）；
VD34	当前的实际速度（频率）。





K SM231 称重模块及对应库函数的使用

称重模块把称重传感器的电压信号转换成数字量 AIW，SM231 称重模块库的作用就是把 AIW 变成实际重量。

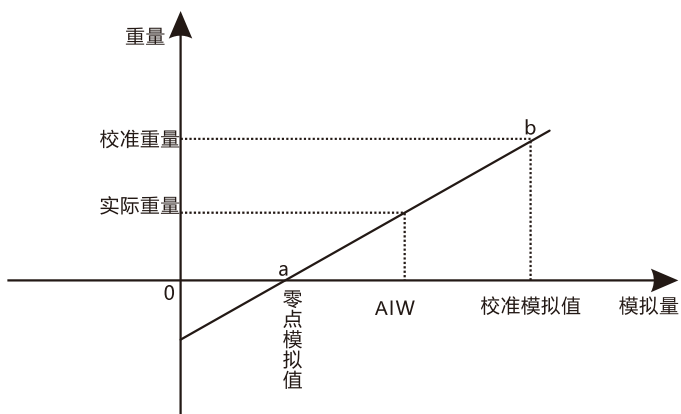


图 F-1 测量重量原理

如上图，通过 a、b 两点得出重量与模拟值的直线关系式，然后就可以得出 AIW 对应的实际重量。为了得到 a、b 两点的坐标值，就需要调零和校准。

调零和校准方法

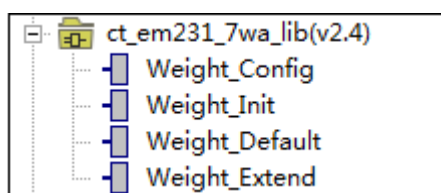
首先假定称重传感器已水平固定，托盘已水平固定在称重传感器上面。

调零时，托盘上不放任何重物，稳定后得到零点模拟值。

校准时，托盘上放有校准用的砝码，稳定后得到校准模拟值。

K.1 称重库函数说明

SM231 称重模块对应称重库 EM231_7WA_LIB(V2.40)，相关库文件和使用说明可从合信官网免费下载，网址：<http://www.co-trust.com> 该库所含的指令参数介绍如下：



称重配置库

- ① 库名称：Weight_Config
- ② 功能：设置通道数目和各通道参数表的起始地址；

库函数	参数名称	输入输出属性	类型	数值范围	出厂值	说明
	EN	IN	BYTE	--	--	使能端
	ParaListBase	IN	DWORD	--	--	设置参数表起始地址指针，从该地址开始存放参数表，参数具体定义请参见“参数表说明” 输入操作数前必须有一个“&”符号
	ChannelNum	IN	BYTE	0~255	--	设置连接的 SM231 通道总数目

注意：必须调用一次，且只在第一个循环周期执行一次。如果有 N 个通道，则所有通道参数表总共需要占用 ChannelParaNum * N 个字节。用户必须自己确保通道参数表占用的 V 内存没有超出允许范围，并且没有覆盖其它程序占用的空间。

参数表说明

调用称重库时，每个通道占用 72 个字节内存，该存储区保存的参数定义如下表（以通道 0 起始地址 VB0 为例）：

参数名	地址	说明	备注
Mode	VB0	模式	基本/扩展模式
Sensitivity	VB1	传感器灵敏度	
LimitFreq	VB2	低通滤波频率	
FilterDepth	VB3	滤波深度	0-255, 0 或 1 时表示不进行均值滤波
WeightRange	VW4	最大秤量范围	
FirCalWeight	VW6	一次校准重量	
SecCalWeight	VW8	二次校准重量	
TareInput	VW10	皮重输入	
MinWeight	VW12	最小秤量范围	一般为 20d, d 为数字阶跃
Step	VB14	数字阶跃	范围：1、2、5、10、20
StandstillTime	VW15	停顿时间	单位：ms
StandstillRange	VW17	停顿范围	
ZeroValue	VW19	零点采样值	
FirCalValue	VW21	一次校准采样值	

SecCalValue	VW23	二次校准采样值	
GWProcessValue	VW25	毛重过程值	
NWProcessValue	VW27	净重过程值	
TWProcessValue	VW29	皮重过程值	
AnalogValueInit	VW31	滤波前采样值	
AnalogValue	VW33	滤波后采样值	
GrossWeight	VW35	毛重	
NetWeight	VW37	净重	
TareWeight	VW39	皮重	
Status_I	VB41	"Weight_Init"状态字节	
Status_D	VW42	"Weight_Default"状态字	
Status_E	VB44	"Weight_Extend"状态字节	
InternalVariable1	VB45	内部变量 1	
AQWx	VW46	L 区备份	
AQWx2	VW48	L 区备份	
LB21_D	VB50	Weight_Default L 区备份	
LB57_D	VB51	Weight_Default L 区备份	
LB58_D	VB52	Weight_Default L 区备份	
LB59_D	VB53	Weight_Default L 区备份	
LB59_E	VB54	Weight_Extend L 区备份	
LW36_E	VW55	Weight_Extend L 区备份	
LD40_E	VD57	Weight_Extend L 区备份	
Reserved	VB61	保留	
ZeroTraceTime	VD62	零点跟踪计时	
StandBeginTime	VD66	停顿状态判断的起始时间	
ZeroTraceValue	VW70	零点跟踪调整值	

称重初始化库

① 库名称: Weight_Init

② 功能: 配置称重模块所接传感器灵敏度、低通滤波截止频率、平均滤波深度;

库函数	参数名称	输入输出属性	类型	数值范围	出厂值	说明
<div style="border: 1px solid black; padding: 5px; width: fit-content;"> Weight_Init:FC3 EN Channel AQWx Sensit~ AQWx2 LimitF~ Status Filter~ </div>	Channel	IN	BYTE	0~255		通道号
	Sensitivity	IN/OUT	BYTE	1: 1mV/V 2: 2mV/V 4: 4mV/V 不允许有其它定义	2	传感器灵敏度 (默认值为 2)
	LimitFreq	IN/OUT	BYTE	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 不允许有其它定义	4	低通滤波截止频率 (默认值为 4)
	FilterDepth	IN/OUT	BYTE	0~255, 0 或 1 时表示不进行平均滤波	2000	平均滤波深度 (默认值为 15)
	AQWx	OUT	WORD	格式: "0x53" + "Sensitivity"		对应称重通道第一个模拟量输出值
	AQWx2	OUT	WORD	格式: "LimitFreq" +		对应称重通道第二

				"FilterDepth"		个模拟量输出值
	Status	OUT	BYTE	Bit0: 特征值设置错误 Bit1: 低通滤波频率设置错误 Bit2: 非法通道号 Bit3: 加载出厂设置完成		状态字节

注意：该指令通过 SM0.0 调用

关于传感器灵敏度、低通滤波截止频率和平均滤波深度的参数说明，请参见附录 L.2 称重库说明。

称重标准库

①库名称：Weight_Default

②功能：实现校秤（调零，校准）和测量（去皮）等功能。

③参数

库函数	参数名称	输入输出属性	类型	数值范围	出厂值	说明
Weight_Default EN Mode LoadFa~ ZeroSet FirCal~ SeekTa~ Delete~ Analog~GNWeig~ Analog~TareWe~ Channel Status Weight~ FirCal~ ZeroVa~ FirCal~	Mode	IN	BOOL	--		模式选择 0: 基本模式, 1: 扩展模式
	LoadFacSetting	IN	BOOL	--		装载出厂设置, 上升沿有效
	ZeroSet	IN	BOOL	--	5461	设置零点, 上升沿有效
	FirCalibrate	IN	BOOL	--		一次校准, 上升沿有效
	SeekTare	IN	BOOL	--		求皮重, 上升沿有效
	DeleteTare	IN	BOOL	--		删除皮重, 上升沿有效
	AnalogValueInit	IN	WORD			滤波前模拟值, 对应该称重通道的第一个模拟量输入
	AnalogValue	IN	WORD			滤波后模拟值, 对应该称重通道的第二个模拟量输入
	Channel	IN	BYTE			通道号 0-6
	WeightRange	IN/OUT	INT		0~32767	最大称量范围 (默认值为 2000)
	FirCalWeight	IN/OUT	INT	大于量程的 5%	2000	一次校准所使用的砝码重量 (默认值为 2000)
	ZeroValue	IN/OUT	WORD			零点采样值 (默认值为 5461)
	FirCalValue	IN/OUT	WORD		60074	一次校准采样值 (默认值为 60074)
	GNWeight	OUT	INT			毛重/净重
	TareWeight	OUT	INT			皮重
Status	OUT	WORD			状态	

注意：关于模式选择、一次校准、皮重等参数说明，请参见附录 L.2 称重库说明。

Status 注释:

位	状态名称	适用范围	描述
Bit0	掉电报警	基本/扩展模式	0: 模块电源正常, 1: 模块无电源 Bit0 为 1 时, 采样值为 0xFFFF
Bit1	断线报警	基本/扩展模式	0: 传感器连接正常, 1: 传感器断线 Bit1 为 1 时, 采样值为 0xFFFE
Bit2	超量程报警	基本模式	1: 毛重值 \geq 额定重量, 采样值为 0xFFFD
	Max+9e	扩展模式	1: 毛重值 \geq 额定重量+9e, 此处 e 为数字阶跃
Bit3	已定皮重	基本/扩展模式	1: 皮重存储器被占用 (皮重过程值 \neq 0)
Bit4	已预设皮重	扩展模式	1: 预设皮重
Bit5	1/4d	扩展模式	1: 毛重小于 $\pm 0.25d$, 此处 d 为数字阶跃
Bit6	停顿	扩展模式	1: 停顿状态确定
Bit7	已找到零点	基本/扩展模式	1: 已找到零点
Bit8	已校准	基本模式	1: 一次校准完成
		扩展模式	1: 一次校准完成(二次校准重量为 0)或者一次校准和二次校准完成
Bit9	低重量	扩展模式	1: 当前重量小于最小称量范围
Bit10	命令只能在停顿状态执行	扩展模式	1: 扩展模式时, 找零点及求皮重必须处于停顿状态
Bit11	命令只允许在已调零状态	基本/扩展模式	1: 一次校准或者二次校准前必须已调零
Bit12	命令只允许在已校准状态	基本/扩展模式	1: 求皮重或预设皮重时必须处于已校准状态
Bit13	校准砝码重量太小	基本/扩展模式	1: 一次校准砝码与零点; 二次校准砝码与一次校准砝码之间重量差值不能小于 5%FS
Bit14	非法皮重值	基本/扩展模式	1: 皮重值不能小于 0 或者大于最大称量范围
Bit15	非法通道号	基本/扩展模式	--

称重扩展库

① Weight_Extend

② 功能: 实现二次校准, 设置最小重量, 设置数字阶跃, 停顿状态检测、预设皮重和零点跟踪等功能。

③ 参数

库函数	参数名称	输入输出属性	类型	数值范围	出厂值	说明
<pre> Weight_ExtEn:FCSE EN - SecCal~ - TarePr~ - ZeroTr~ - Channel Status - SecCal~ - MinWei~ - Step - Stands~ - Stands~ - TareIn~ - SecCal~ - ZeroTr~ - ZeroTr~ </pre>	SecCalibrate	IN	BOOL	--	--	二次校准，上升沿有效
	TarePreset	IN	BOOL	--	--	预设皮重
	Channel	IN	BYTE	0-6	--	通道号
	SecCalWeight	IN/OUT	INT	0	0	二次校准时的砝码重量(默认为0，表示不进行二次校准)
	MinWeight	IN/OUT	INT		20	最小重量，默认为20d，此重量值只能用于具有最小重量以上的规定数字阶跃的校准记录。最小重量取决于所用的传感器类型与型号。
	Step	IN/OUT	BYTE	1/2/5/10/20，不允许其他定义	1	数字阶跃（默认为1）
	StandstillTime	IN/OUT	INT		1000	停顿时间(默认为1000)单位：ms
	StandstillRange	IN/OUT	INT		10	停顿范围（默认为10）
	TareInput	IN/OUT	INT		0	皮重输入，即预设皮重值(默认为0)
	SecCalValue	IN/OUT	WORD		0	二次校准模拟值(默认为0)
	ZeroTraceEn	IN	BOOL		--	使能零点跟踪
	ZeroTraceRange	IN/OUT	INT	1~3	--	零点跟踪范围(默认为1)
	ZeroTraceTime	IN/OUT	INT	1000	--	零点跟踪时间，单位ms，默认为1000ms
	Status	OUT	BYTE	--	--	状态字节

Status 注释:

位	状态名称	解释
Bit0	二次校准时的砝码重量太小	二次校准时的砝码重量与一次校准时的砝码重量之差小于量程的5%
Bit1	数字阶跃不正确	数字阶跃只能为1、2、5、10、20
Bit2	停顿时间不正确	停顿时间必须大于0
Bit3	停顿范围不正确	停顿范围必须大于0
Bit4	预设皮重值超范围	预设皮重值不能为负或超过额定称量范围
Bit5	预留	--
Bit6	停顿	停顿状态确定
Bit7	非法通道号	--

K.2 称重库说明

本节说明称重库各个指令的重要参数以及使用时的注意事项。

1. 称重配置库 Weight_Config

该库使用 SM0.1 调用，其中，ParaListBase 参数设定用户指定参数表起始地址指针和所用内存范围，每个通道占用 72 个字节的内存；ChannelNum 为用户连接的 SM231 通道数目，该参数只能大于或等于实际使用的通道数目。

2. 称重初始化库 Weight_Init

该库使用 SM0.0 调用，用户可以根据应用需求修改传感器灵敏度 Sensitivity、低通滤波截止频率 LimitFreq、平均滤波深度 Filterdepth，这三个参数在装载出厂设置时会给相应的值。（传感器灵敏度必须在校准前修改，低通滤波截止频率和平均滤波深度任意时刻修改都能生效。）

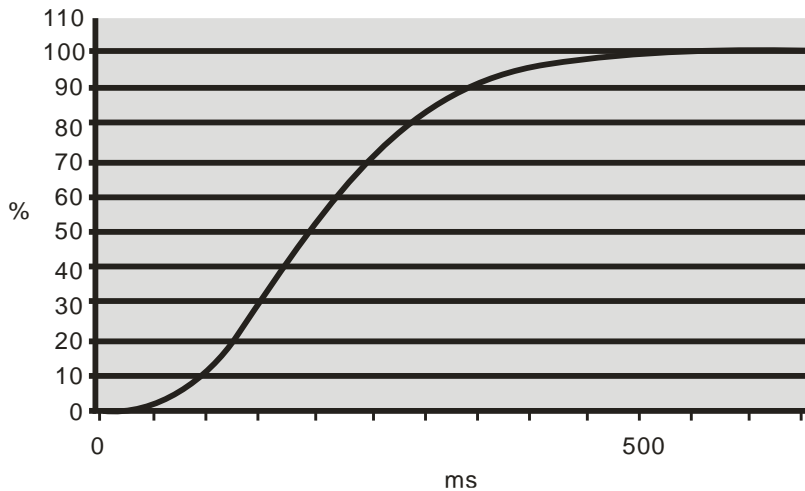
● 传感器灵敏度

根据所连接称重传感器的特征值，可选择 1mV/V，2mV/V 或 4mV/V。

例如：如果所连称重传感器的特征值为 2.85mV/V，那么下一个较高特征值通常须设为 4mV/V。

● 低通滤波器截止频率

模块中配备了一个临界条件下导电的低通滤波器，其目的是抑制干扰。下图显示了在 $f_g=2\text{Hz}$ 时数字低通滤波器的阶跃响应：



为了抑制干扰，必须定义合适的截止频率。称重模块对测量值变化的反应“速度”是通过定义极限频率而设定的。

例如：截止频率为 5Hz 时，称重模块对重量变化的反应会非常快，而截止频率为 0.5Hz 时，相应反应则会迟缓许多。

● 平均滤波深度

滤波器可平滑稳定称量值，防止干扰。重量值根据 n 个重量平均值测定；传感器采样频率为 50Hz，因此每过 20 毫秒计算一次重量。例如，如果滤波深度 $n=10$ ，则表示有 10 个测量值被用来计算平均值。每过 20 毫秒便轮换一次计算值。

如果干扰频率对应于关系“ $1/(\text{滤波器深度} * 20 \text{ ms})$ ”的整数倍，那么平均值滤波器也能实现周期性干扰的非常好的阻尼。

3. 称重标准库 Weight_Default

该库使用 SM0.0 调用，使用扩展指令进行第二次校准时，Mode 置 1。第一次使用 231-7WA 模块时需装载出厂设置，相应参数会自动赋值，不需要手动赋值。具体哪些参数需装载出厂设置请参见前文的库指令说明。

● 模式说明

称重模块 SM231-7WA 提供两种模式的库使用方法。通过配置"Weight_Default"中的参数"Mode"可以选择使用基本模式或者扩展模式的称重库指令。

1) 基本模式 (Mode=0)

(1) 只需调用"Weight_Config"、"Weight_Init"和"Weight_Default"3 个库；

(2) 支持的功能有：

- 设置传感器灵敏度、低通截止频率、滤波深度；
- 装载出厂参数；
- 设置最大称量范围；
- 设定零点；
- 一次校准；
- 求皮重；
- 删除皮重；
- 读取滤波前后采样值；
- 掉电报警、断线报警、超量程报警、已调零、已校准、已设皮重等状态指示；

2) 扩展模式 (Mode=1)

(1) 调用"Weight_Config"、"Weight_Init"、"Weight_Default"和"Weight_Extend"4 个库；

(2) 支持的功能有：

- 支持基本模式下的所有功能；
- 设置最小称量范围；
- 二次校准；
- 预设皮重；
- 数字阶跃可定义为 1/2/5/10/20；
- 停顿状态确定（设置零点和求皮重时必须处于停顿状态）；
- Max+9e(GB/T 7724-2008)、已预设皮重、1/4d（GB/T 23111-2008）、停顿、低重量等；
- 零点跟踪；

● 皮重

皮重指商品外包装材料的重量（即运输包装的重量）。

比如，汽车上磅前 GNweight 参数值为汽车重量，用上升沿指令导通 SeekTare 后，GNweight 变为 0，Tareweight 为之前的 GNweight 值。

汽车装满货物后再次上磅时，GNweight 参数值为货物净重。需要指出的是，无皮重时，GNWeight 为毛重；有皮重时，GNWeight 为净重。

● 一次校准

模块接收的是传感器的电信号，未校准时重量和传感器的电信号无准确的对应关系。

比如在称重台上放置一个 1000g 的砝码，程序中 GNweight 具体的显示值由第一次校准时设定，如果第一次校准时放置 1000g 砝码并将 FirCalweight 设为 2000（200g），则 GNweight 随后会将每 1000g 物体转换为 200g。因此第一次校准非常重要。

4. 称重扩展库 Weight_Extend

该库在扩展模式（Mode=1）下才能使用，需使用 SM0.0 来调用，可用于进行二次校准、预设皮重、设定数字阶跃以及启用零点跟踪功能。

停顿时间

停顿监视用于识别称重台何时能处于一种稳定的平衡状态。如果重量值在一段规定时间（停顿时间）内的变化小于规定的偏差范围（停顿值），则确定称重台停顿。停顿监视用在称重台的静态操作中（主要用于命令：零点采样，求皮重）。

数字阶跃

数字阶跃可以相应地定义为 1，2，5，10 或 20。

零点跟踪

用于消除秤长时间使用后出现的零点漂移，导通则有效，不导通无效。

启用零点跟踪功能时，当毛重/净重的绝对值小于阈并持续一段时间，将当前滤波后模拟量写入零点时模拟量，并将原零点时模拟量保存，毛重/净重输出为 0。

取消零点跟踪功能后，将原零点时模拟值写回。零点跟踪的最大范围为 10d(d 是数字阶跃)，超过最大范围后零点跟踪无效。

预设皮重

TareInput 参数为输入皮重参数，写入一个皮重值后用上升沿导通 TarePreset，Tareweight 参数值则为 TareInput 值，上升沿指令导通 DeleteTare，皮重值变为零。例如：

称重台上不放置任何重物，TareInput 值设为 500，导通预设皮重指令 TareInput 后，TareWeight 为 500，上升沿导通删除皮重 DeleteTare 指令后，TareWeight 值变为 0，GNweight 值变为-500。

Weight_Extend 指令中的预设皮重与 Weight_Default 指令中求皮重的不同点与相同点：

不同点：

TareInput 可以任意设置，与是否放置实物无关，设置值即为皮重显示值；

Weight_Default 指令中必须在放置实物后，GNweight 中有值，才能求皮重，皮重值即为 GNweight 值。

相同点：

均通过上升沿导通 DeleteTare 指令来删除皮重，从而回到去皮重之前的状态；

求皮重后，GNweight 值为求皮重前的 GNweight 值减皮重值。

二次校准

Secselweight 参数为第二次放置砝码值（1000g 砝码设为 10000），注意：二次校准时的砝码重量与第一次校准时的砝码重量之差大于量程的 5%。

一次校准后，待 Weight_Extend 指令中状态字为 64（停顿状态确定）后，在称重台上放置二次校准的砝码。

如需了解有关称重模块的详细使用说明，请访问合信官网获取《CTH200 静态称重模块 SM231-7WA 用户手册》，地址：<http://www.co-trust.com>

L SM277A 模块的使用

SM277A 模块具有以下主要特点:

- 光电隔离技术, 抗干扰能力强, 可靠性高。
- 集成终端电阻, 总线采用接线方式, 无需专用网络接头。
- 供电电源有反接保护和浪涌吸收功能, 适用于恶劣的工业环境。

使用规范

- 信号线应采用屏蔽双绞线, 且屏蔽线要双端接地。
- 在系统有良好接地的情况下模块的接地端应接到地线上, 否则不接地。
- 如果作为最后一个站, 终端电阻应打在 ON 的位置。

PROFIBUS-DP 网络

【通信】

PROFIBUS-DP 网络经过其 DP 通信端口连接到 SM277A DP 从站模块, 而 SM277A DP 从站模块经过串行 I/O 总线连接到 CTH200 系列 CPU。SM277A 采用接线端子连接 PROFIBUS 网络, 不使用标准连接器。

【功能】

SM277A 是用于 PROFIBUS DP 网络的模块化从站设备, 最多可以连接 6 个用于数字或模拟输入和输出的扩展模块。

作为 DP 从站, SM277A 模块可以在 9.6K 至 12M 波特率之间的任何 PROFIBUS 波特率上运行。SM277A 模块接收从主站传来的多种不同的 I/O 配置和不同数量的数据, 并向主站发送数据, SM277A 还能读写 CTH200 系列 CPU 中已定义的变量数据块。这样用户能与主站交换任何类型的数据。

MPI 网络

【通信】

SM277A 模块作为一个 MPI 从站与同一网络上如 PG/PC 编程站、Copanel HMI 或 S7-300/S7-400 CPU 等其它主站进行通信。该模块利用 S7-300/400 等主站模块的 XGET/XPUT 指令功能, 提供 MPI 主站与 CTH200 的网络通信。当 SM277A 模块进行 MPI 通信时, MPI 主站必须使用 SM277A 模块的站址向 CTH200 CPU 发送信息。发送给 SM277A 模块的 MPI 信息将通过 SM277A 传送给 CTH200 系列 CPU。

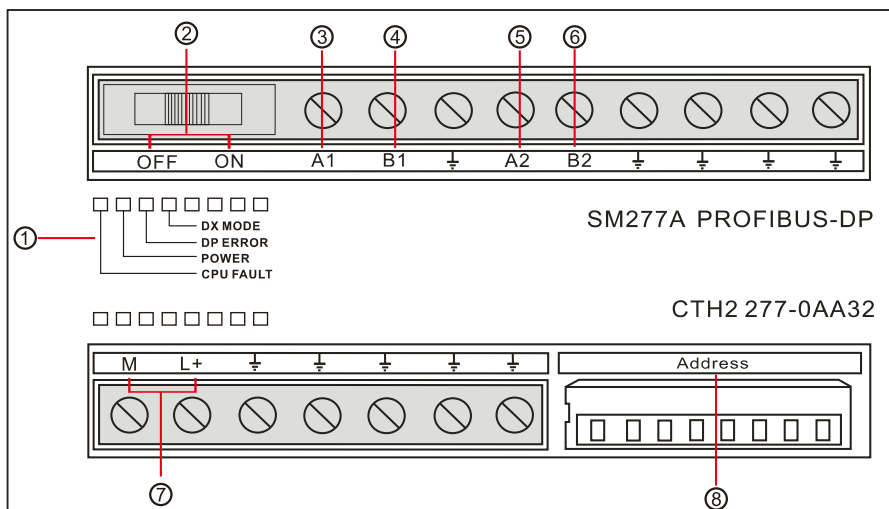
【功能】

SM277A 是用于 MPI 网络的从站设备时, 最多可以连接 6 个用于数字或模拟输入和输出的扩展模块。

作为 MPI 从站, SM277A 模块可以在 9.6K 至 12M 波特率之间的任何 MPI 波特率上运行。SM277A 模块接收从主站传来的多种不同的 I/O 配置和不同数量的数据, 并向主站发送数据, 而且 SM277A 还能读写 CTH200 系列 CPU 中已定义的变量数据块。这样, 使得用户能与主站交换任何类型的数据。

SM277A 组件

状态 LED 位于模块的正面，地址开关、终端电阻开关、DP 从站的接口端子以及用户电源位于接线端子侧如下图所示：



- ① 状态 LED。
- ② 终端电阻选择开关：打到 ON 表示带终端电阻，打到 OFF 表示不带终端电阻。
- ③ 隔离的信号 A1。
- ④ 隔离的信号 B1。
- ⑤ 隔离的信号 A2。
- ⑥ 隔离的信号 B2。
- ⑦ 用户电源。
- ⑧ 地址开关：用 8 位的拨码开关设定，以二进制数表示，有效选择范围为 0~126。



提示

A1/B1 与 A2/B2 均为 RS485 电气接口，内部电气特性一致，用户可任选一组连接，也可两组都接。

S7-300 与 CTH200 系列 SM277A 之间的 PROFIBUS DP 通讯

S7-300 与 CTH200 通过 SM277A 进行 PROFIBUS DP 通讯，需要在 STEP7 中进行 S7-300 站组态，在 CTH200 系统中不需要对通讯进行组态和编程，只需要将要进行通讯的数据整理存放在 V 存储区与 S7-300 的组态 SM277A 从站时的硬件 I/O 地址相对应就可以了。

在 OB1 中编写调用 FC1 (DP_SEND) 和 FC2 (DP_RECV) 的程序，DP 主站可以读写从站数据，从而实现 S7-300 与 CTH200 CPU 之间的通讯。执行 DP_SEND 指令将 CTH200 CPU 的内存数据输出到 SM277A 扩展模块；通过执行 DP_RECV 将 SM277A 扩展模块的数据输入到 CTH200 的内存。

图 G-1 为包含 CTH200 CPU 和 SM277A DP 从站模块的 PROFIBUS 网络。

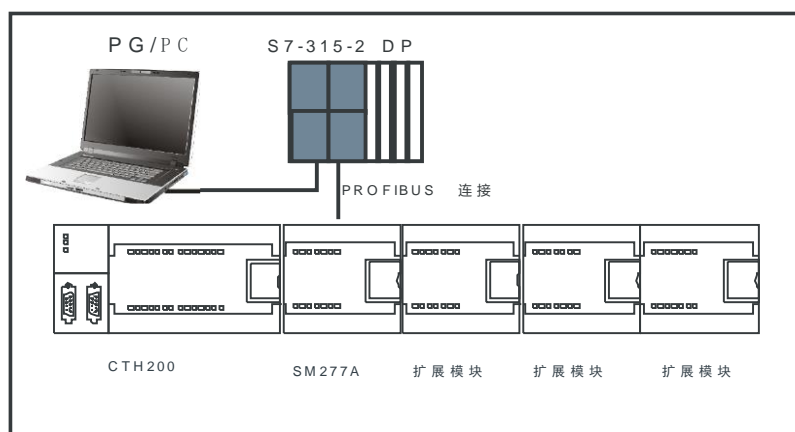


图 G-1 PROFIBUS 网络示例

注：要将 SM277A 作为一个 DP 从站使用，用户必须设定与主站组中的地址相匹配的 DP 端口地址。从站地址是使用 SM277A 模块上的拨码开关设定的。

S7-300 与 CTH200 的 SM277A 之间的 MPI 通讯

CTH200 CPU 与 S7-300 之间采用 MPI 通讯方式时，CTH200 PLC 中不需要编写任何与通讯有关的程序，只需要将要交换的数据整理到一个连续的 V 存储区当中即可，而 S7-300 中需要在 OB1（或是定时中断组织块 OB35）当中调用系统功能 X_GET（SFC67）和 X_PUT(SFC68），实现 S7-300 与 CTH200 CPU 之间的通讯，调用 SFC67 和 SFC68 时 VAR_ADDR 参数填写 CTH200 的数据地址区，这里需填写 P#DB1.xxx BYTE n 对应的就是 CTH200 CPU V 存储区当中 VBxx 到 VB（xx+n）的数据区。

图 G-2 所示为包含 CTH200 和 SM277A DP 从站模块的 MPI 网络。

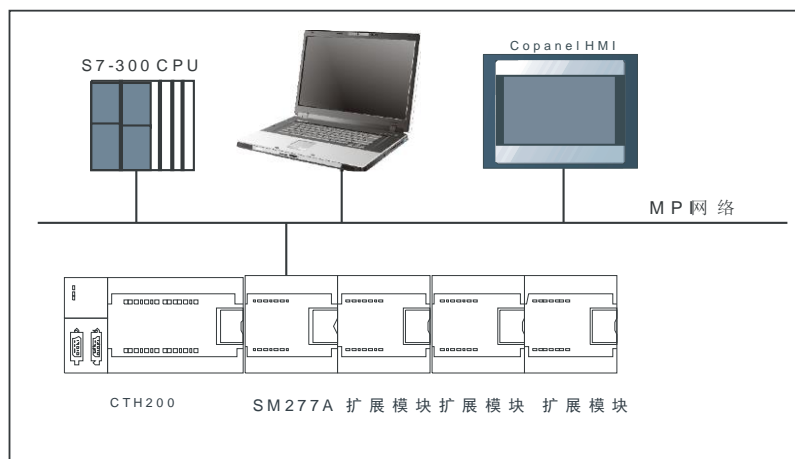


图 G-2 MPI 网络示例

注：从站地址(NEST_ID)的设定需要与 SM277A 模块上的拨码地址一致。

- CPU 315-2 作为 DP 主站，并用带有 STEP 7 编程软件的编程设备进行了组态。
- CTH200 CPU 是 CPU 315-2 DP 主站的一个 DP 从站。
- CPU 315-2 DP 主站借助用户程序中的 DP_SEND 和 DP_RECV 指令从 CTH200 中读写数据。

- S7-300 主站模块、PG\PC 编程站、Copanel HMI 作为 MPI 主站
- CTH200 是一个 MPI 从站。
- 利用 S7-300/400 等主站模块的 XGET / XPUT 指令从 CTH200 中读写数据。

M SM277B 模块的使用

SM277B 是用于 PROFIBUS DP 网络的模块化从站设备，可与 DP 主站进行通信。SM277B 最多可以连接 6 个用于数字或模拟输入和输出的扩展模块。SM277B 采用接线端子连接 PROFIBUS 网络，而不使用标准连接器，波特率自动调整为与主站一致。

【主要特点】

- 光电隔离技术，抗干扰能力强，可靠性高。
- 供电电源有反接保护和浪涌吸收功能，适用于恶劣工业环境。

【使用规范】

- 信号线应采用屏蔽双绞线，且屏蔽线要双端接地。
- 在系统有良好接地的情况下模块的接地端应接到地线上，否则不接地。
- 如果作为最后一个站，终端电阻应打在 ON 的位置。

M.1 PROFIBUS-DP 网络结构

分布式 I/O 系统包含主动(主站)和被动(从站)节点，它们通过 PROFIBUS-DP 互相连接。

下图显示的是 SM277B 位于一个典型的 PROFIBUS-DP 网络组态中的情况：

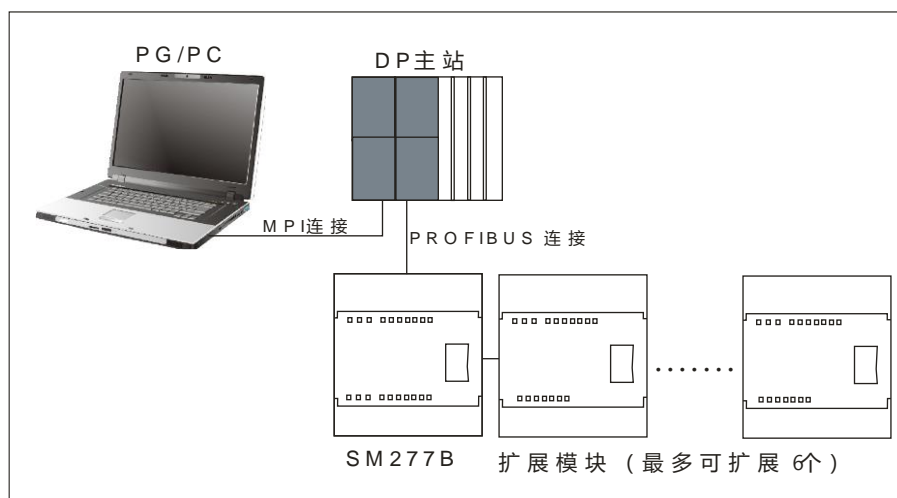


图 H-1 PROFIBUS-DP 网络的结构

M.2 SM277B 的组件

下图为 SM277B 结构：

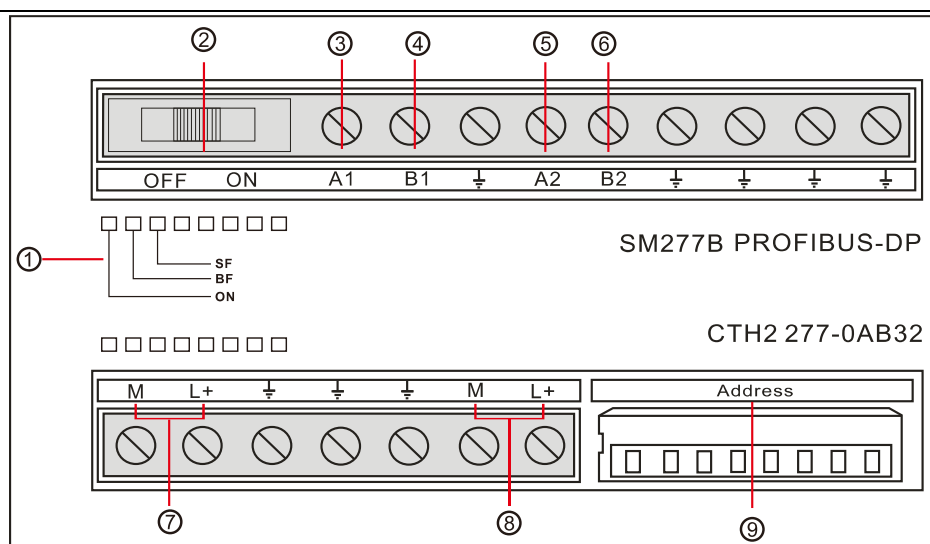


图 H-2 SM277B 组件结构

- ① 状态 LED。
- ② 终端电阻选择开关：打到 ON 表示带终端电阻，打到 OFF 表示不带终端电阻。
- ③ 隔离的信号 A（网络输入）。
- ④ 隔离的信号 B（网络输入）。
- ⑤ 隔离的信号 A（网络级联输出）。
- ⑥ 隔离的信号 B（网络级联输出）。
- ⑦ 用户电源。
- ⑧ 传感器电源。
- ⑨ 地址开关：用 8 位拨码开关设定，以二进制数表示，有效选择范围为 1-125。

M.3 使用指南

本节介绍 SM277B 的使用步骤，结合“CPU312-1AE13 与 SM277B 通信”实例逐步介绍 SM277B 硬件组态、用户程序编写和调试以及系统诊断等过程。

实例所用的网络结构如下：

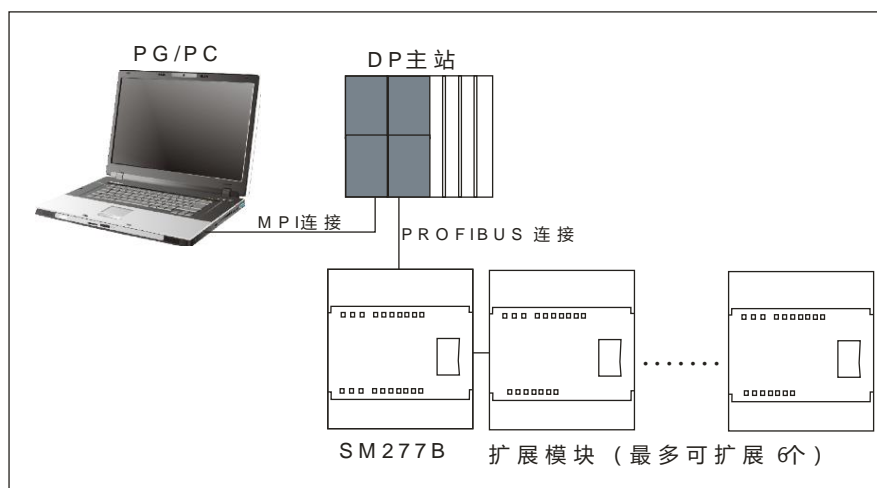


图 H-3 SM277B 网络结构

示例所用的组件

组件	说明
一台装有STEP7的PG/PC	STEP 7的版本必须支持组态DP主站
一条MPI编程电缆	编程用（下载网络硬件配合和工程程序，监控数据）
一个DP主站系统	一个CPU312-1AE13，一个CP 342-5（作为主站）
SM277B	DP从站设备
一根PROFIBUS通信电缆	PROFIBUS电缆必须具有一个标准的PROFIBUS连接器用于连接DP主站。
一个可以连接到SM277B上的CTH200扩展模块(SM)	比如CTH200的SM223-1BL32模块

M.3.1 硬件组态

硬件组态步骤如下：

1) 启动 SIMATIC Manager

选择“文件”->“新建”，在出现的对话框中输入项目名称并选择一个目录用来保存项目。

2) 创建一个 SIMATIC 300 站点

选择要使用的站(例如：SIMATIC 300 站点)。

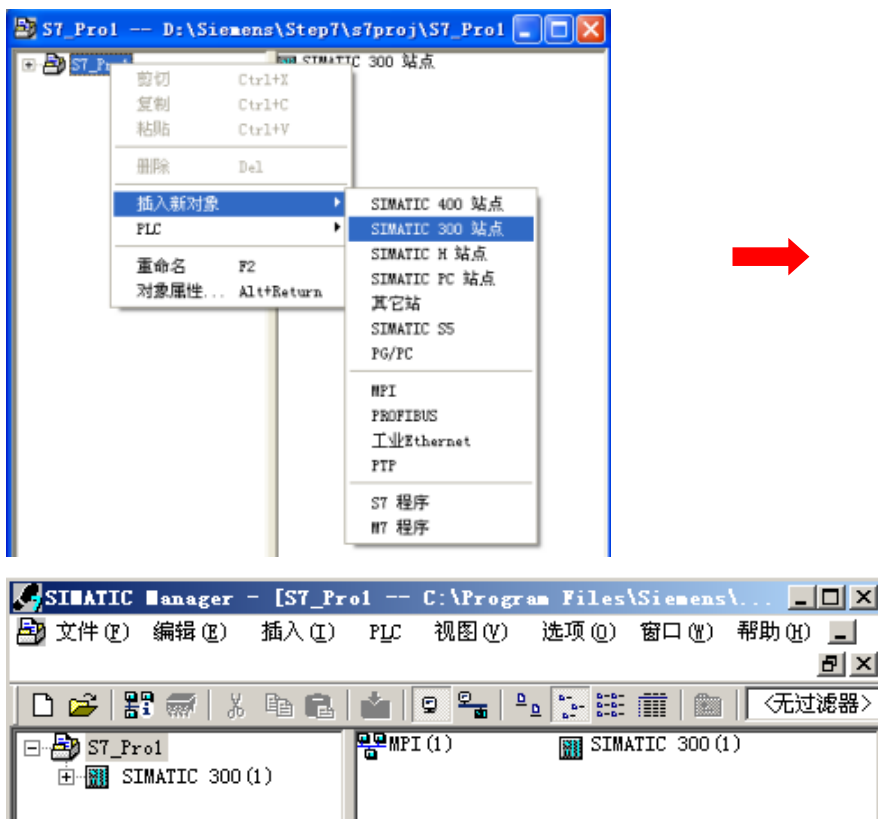


图 H-4 插入 SIMATIC 300 站点

3) 打开 HW-Config，进行硬件组态

双击步骤 2 中插入的 300 站点“SIMATIC 300(1)”，如下显示：

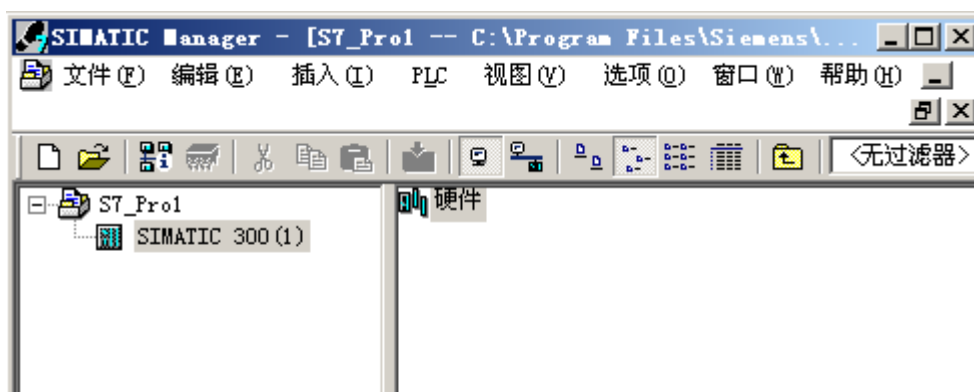


图 H-5 打开 HW-Config 界面

双击上图中的“硬件”打开 HW-config 界面来设置分布式 I/O(DP)机架、模块和 PROFIBUS 连接等。

4) 添加一个导轨

必须先为所使用的站添加导轨（机架，“Rail”），该导轨用于安装 DP 主站。

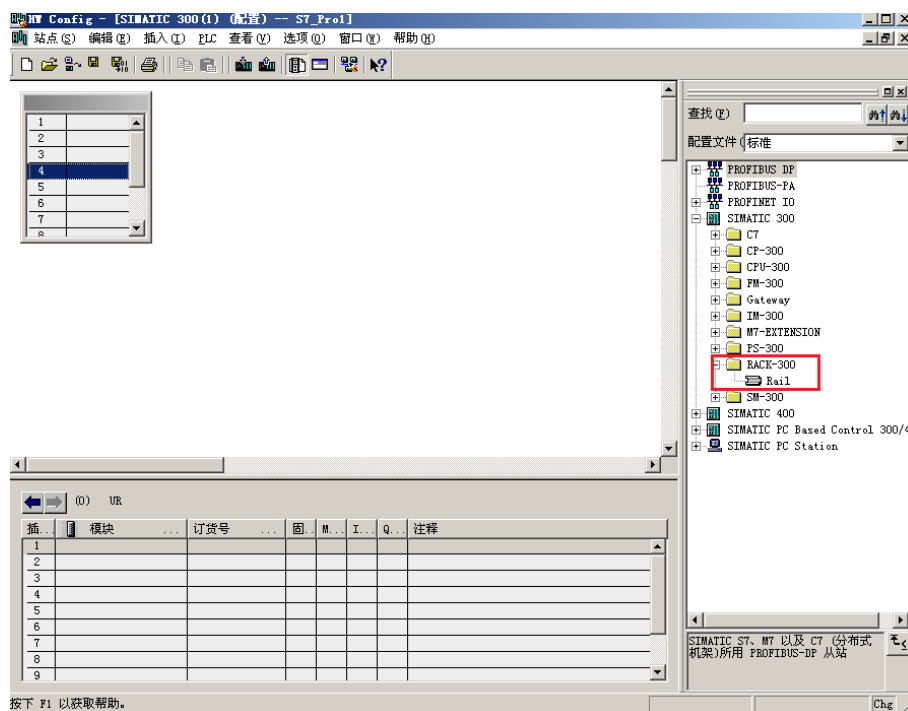


图 H-6 添加导轨

5) 添加 Power Supply (可选)

向导轨添加所用的电源。只能将电源置于机架的第一个插槽中。

6) 添加 DP 主站的 CPU 和 PROFIBUS 连接 (CP 342-5)

将硬件目录中实际使用的 DP 主站的 CPU 拖至导轨第二个插槽中，然后将硬件目录中实际使用的 CP 342-5 拖至导轨第四个插槽中，如下图所示：

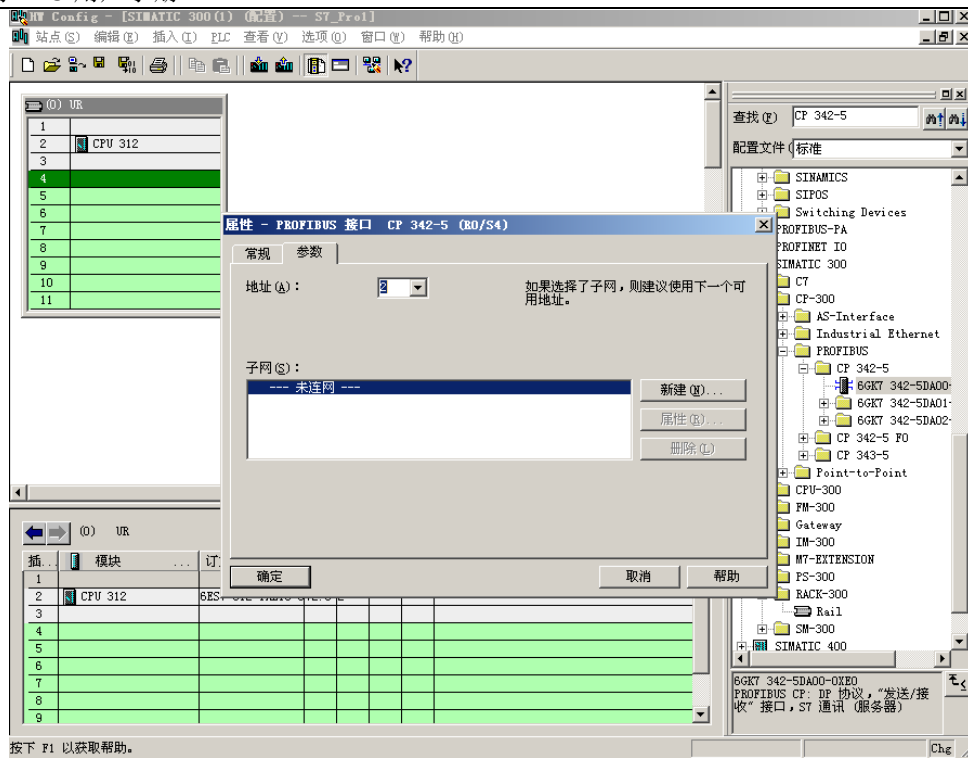


图 H-7 添加 DP 主站的 CPU

在上图对话框中，设置 PROFIBUS CP 342-5 的接口地址（为 2），并单击“新建”按钮为 DP 主站添加一个 PROFIBUS 连接，并在“属性”对话框中设置需要的网络参数，包括传输率（按默认 1.5M 设置）和配置文件（选“DP”），如下图所示：

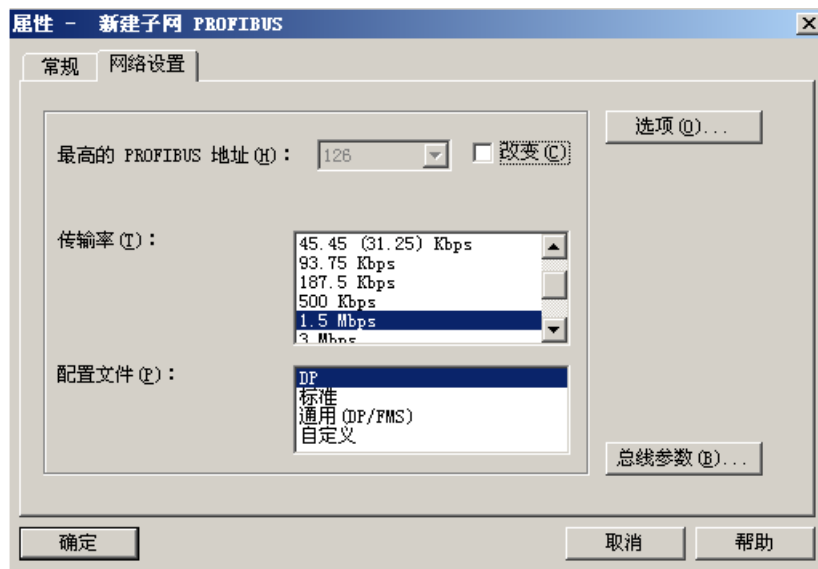


图 H-8 主站网络设置

单击“确认”按钮完成 DP 主站的 CPU 和 PROFIBUS 连接的添加。

确定后，设置 CP 342-5D 的数据输入输出地址，工作模式等。双击导轨上的 CP342-5，在弹出的属性对话框，如下图所示：

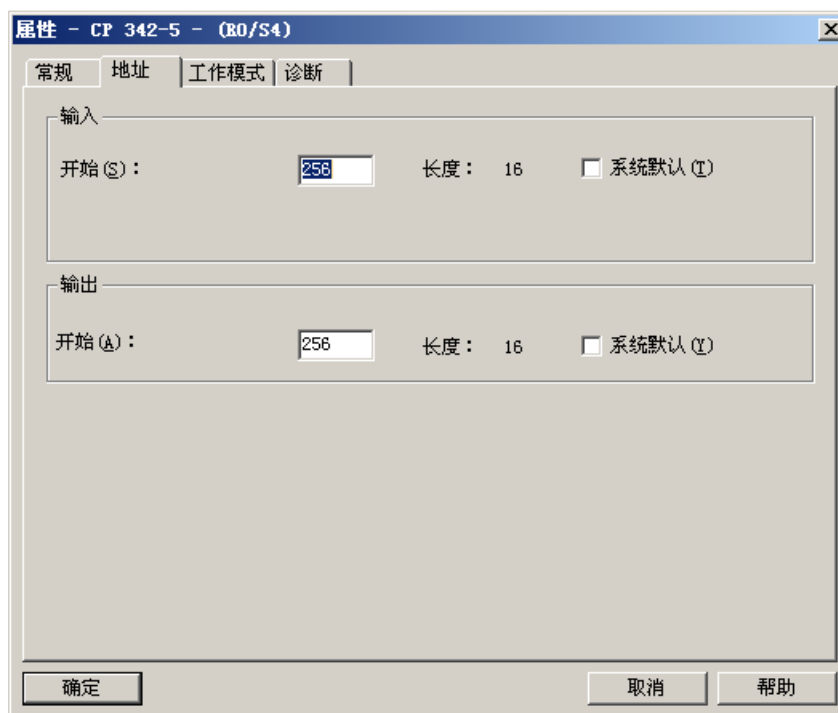


图 H-9 CP 342-5D 输入输出地址设置

设置 CP 342-5 的输入输出地址

在属性对话框中的【地址】项中设置，“输入”开始和“输出”开始地址按默认设置(默认为 256(16#100，子程序的 CPLADDR 参数要使用此值))

设置 CP 342-5 的工作模式

在属性对话框中的[工作模式]项中将 CP 342-5 设置为“DP 主站”，如下图所示：

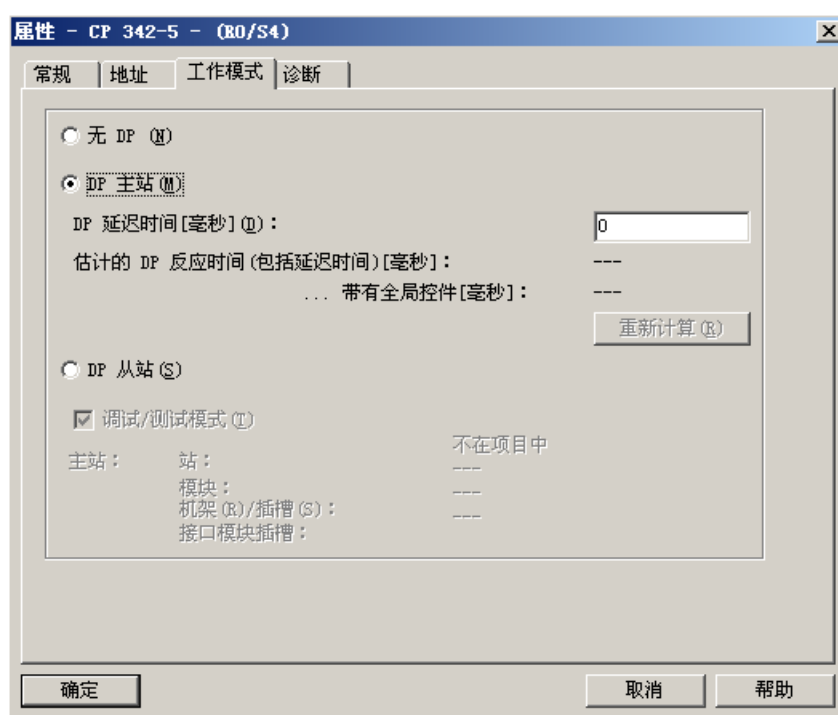


图 H-10 CP 342-5D 工作模式设置

设置好 CP342-5 属性后，会自动在右边的空白处添加一 PROFIBUS DP 主站系统，如下图所示：

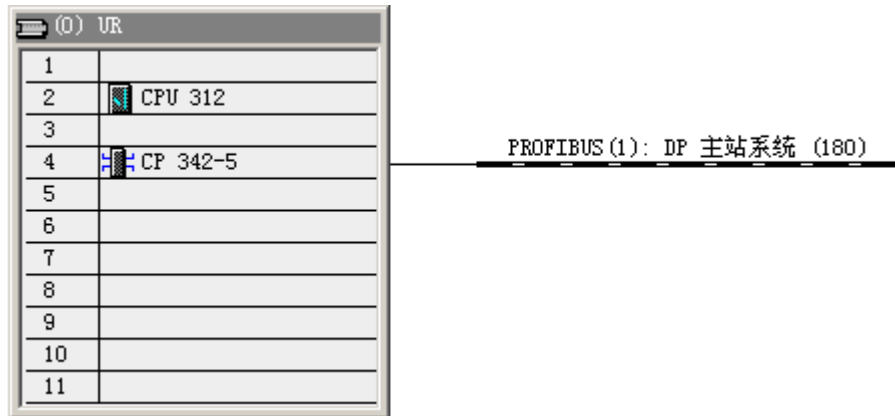


图 H-11 CP 342-5D 属性设置

1) 导入 SM277B 的 GSD 文件

首次使用 SM277B，必须在 STEP 7 上安装包含设备功能信息的 GSD 文件，否则跳过该步。

安装方法：

在“HW-config”界面中，选择菜单[选项]->[安装 GSD 文件]，在弹出的“安装 GSD 文件”对话框单击“浏览”按钮打开 SM277B GSD 文件所在的目录，然后单击“确认”按钮，出现如下第 2 图所示，然后选中“SM277B.gsd”，再单击“安装”按钮进行安装过程，安装结束后单击“关闭”按钮完成 SM277B GSD 文件的安装。

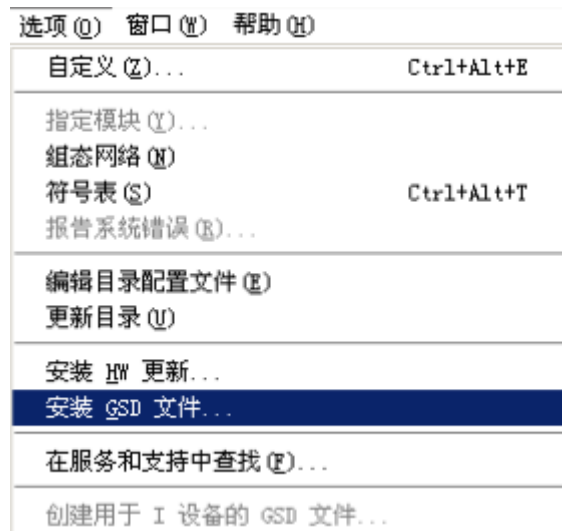


图 H-12 导入 GSD 文件

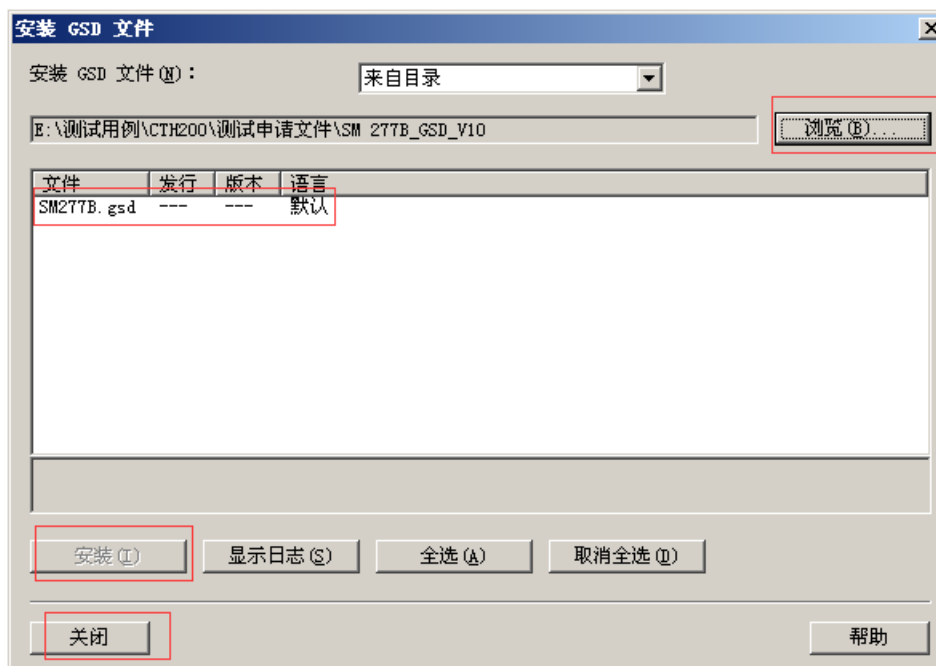


图 H-13 安装 GSD 文件

**提示**

注：请前往 <http://www.co-trust.com> 网站下载 SM277B 的 GSD 文件(SM277B.GSD)。

安装完成后，HW-Config 硬件目录中可以看到 SM277B 及其可扩展的模块信息，如下图所示：

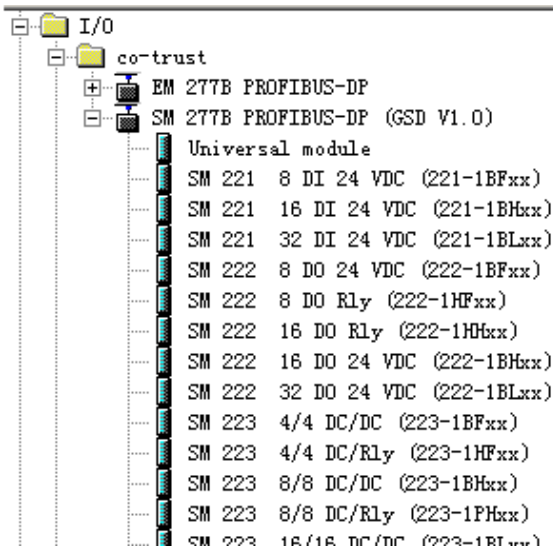


图 H-14 GSD 文件安装完成

2) 添加 SM277B DP 从站

将 SM277 PROFIBUS-DP 从硬件目录中拖至“PROFIBUS(1): DP 主站系统(180)”网上。此时会弹出一个属性框，如下图所示在弹出的属性对话框中设置 SM277B 从站和网络设置。

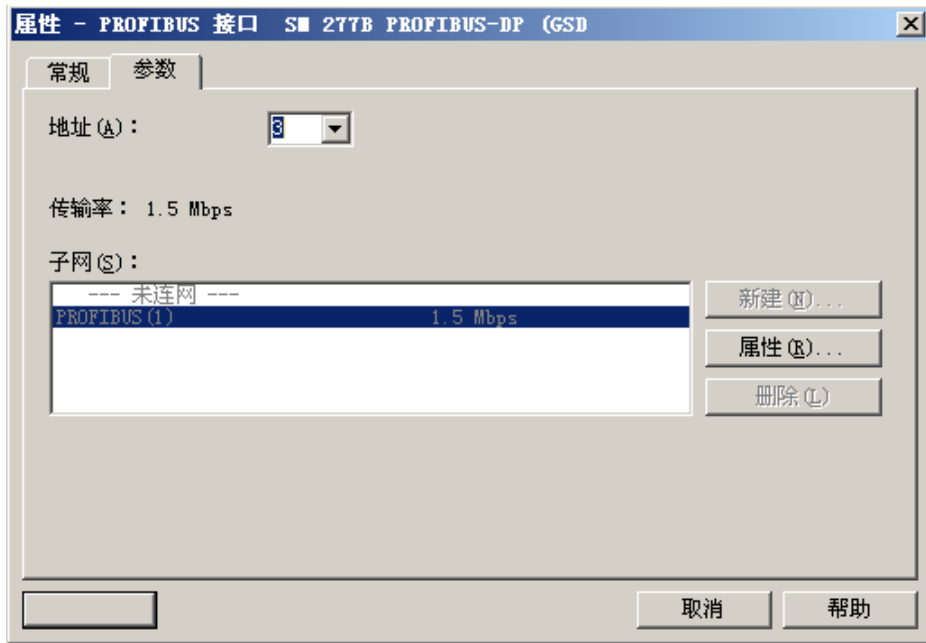


图 H-15 添加 SM277B DP 从站

设置从站地址

在属性框的[参数]项下的“地址”中设置 SM277B 从站地址。



提示

有效地址为 1 至 125 地址, 在 PROFIBUS DP 上具有唯一行, 且地址一定要设为 SM277B 模块的拨码开关所拨的地址。

设置从站网络设置

单击属性框的[参数]项下的“属性”按钮, 此时会弹出一个子对话框, 如下图所示:

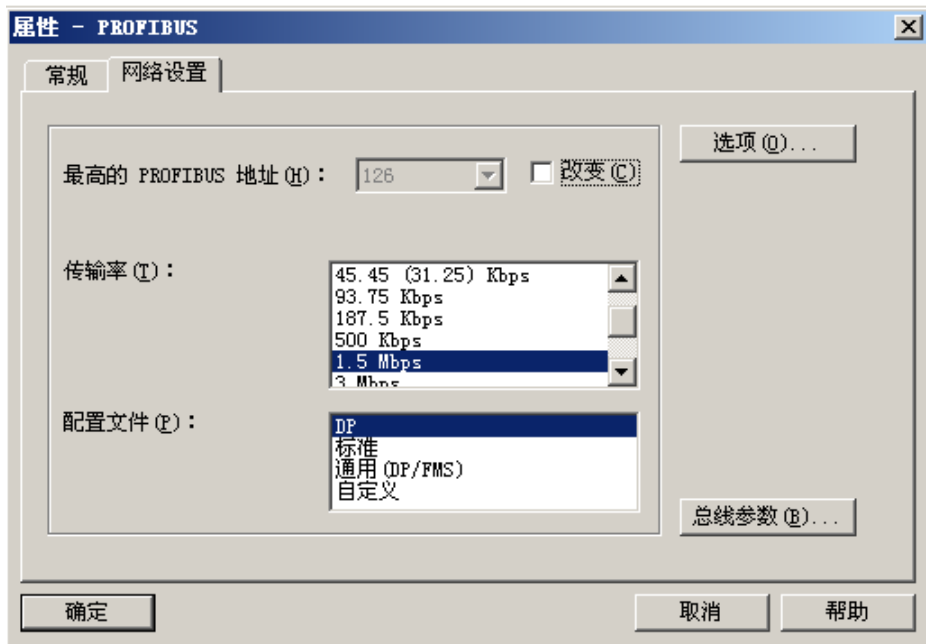


图 H-16 从站属性设置

单击该子对话框的【网络设置】项，设置“传输率”（传输率按默认 1.5M 设置）和“配置文件”（设置为“DP”），此时 SM277B 从站的添加和配置就完成了，如下图所示：

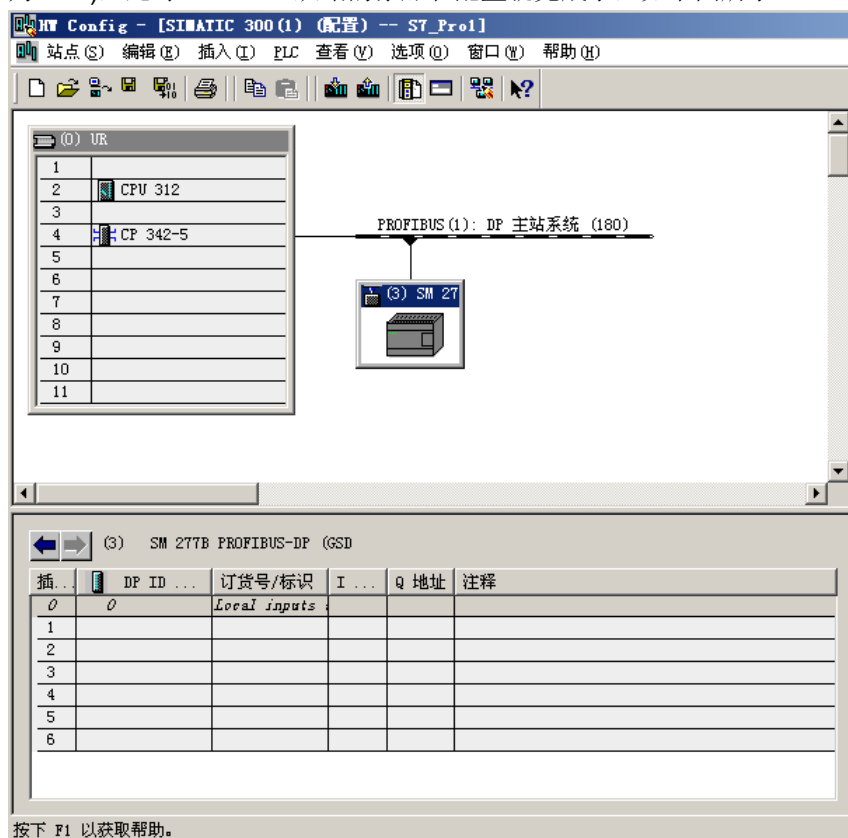


图 H-17 从站属性配置

注：确保 SM277B 从站属性的[参数赋值]项的“十六进制参数赋值”按默认设置，如下图所示值（00,00,00）（双击 SM277B 图标可以看到），否则可能会通信异常。

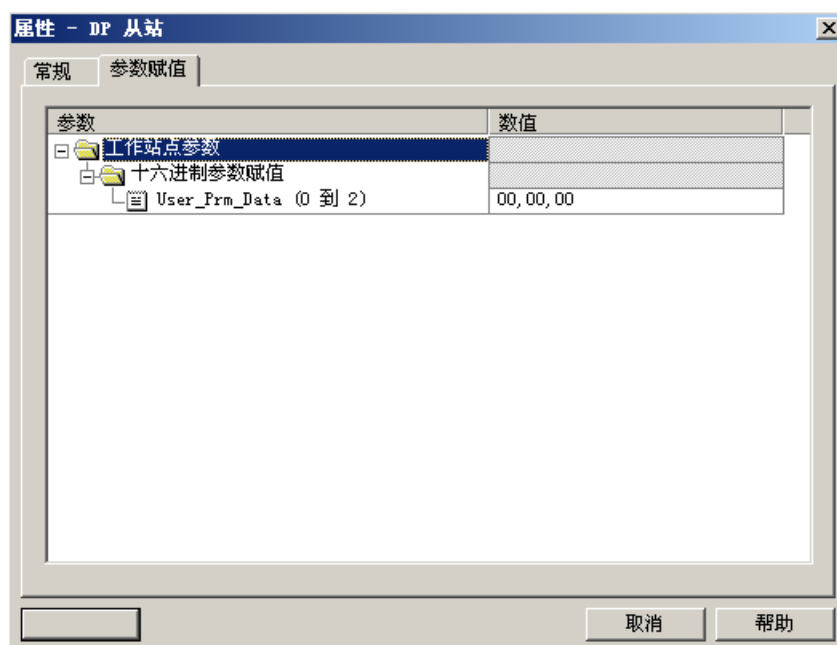


图 H-18 DP 从站参数赋值

※ 为 SM277B 从站添加扩展模块

从硬件目录“SM277B PROFIBUS-DP”下将需要的扩展模块按您需要的顺序依次拖至 SM277B 的

空白槽号中，完成 DP 从站扩展模块的添加，如下图所示：

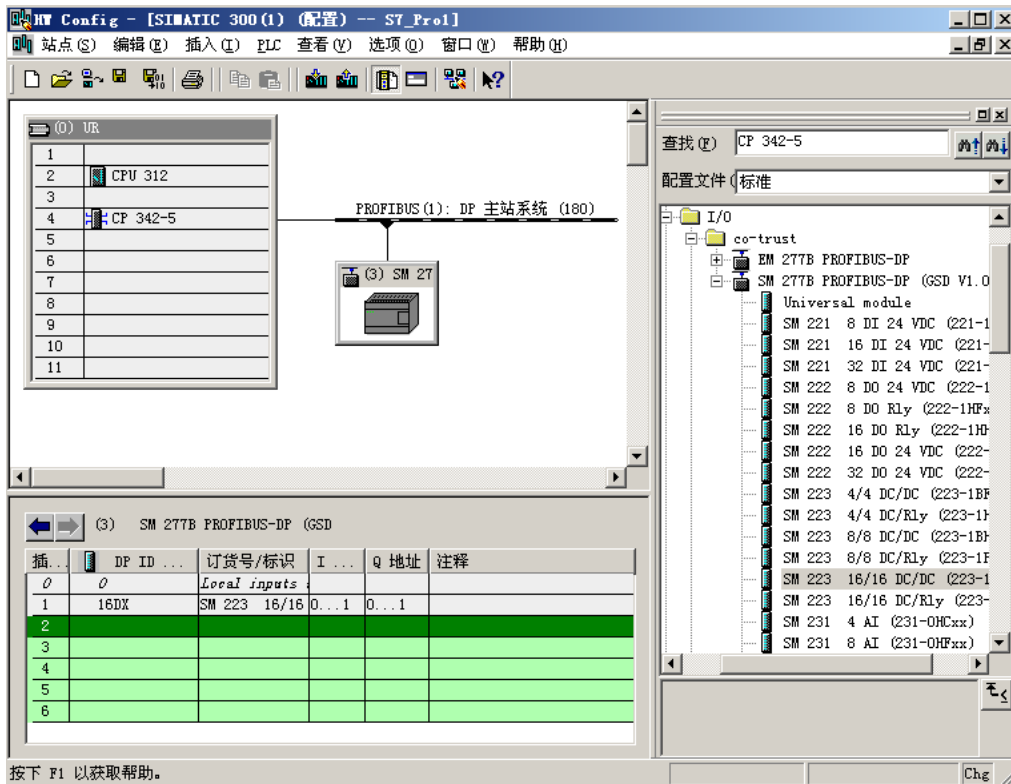


图 H-19 为 SM277B 添加扩展模块

※ 将硬件组态下载到 DP 主站的 CPU

执行 HW-config 界面的菜单“站点”->“保存并编译”硬件组态。

到此已经在 STEP7 中完成了硬件配置，在传送组态之前，请确认已经在 DP 主站和 PC/PG(MPI 连接)间建立了连接，再执行同界面的菜单“PLC”->“下载”，才能将硬件组态下载到 DP 主站 CPU 中去。

M.3.2 编程与调试

SM277B 读取扩展模块的输入，并使得这些输入可被 DP 主站使用。DP 主站向 SM277B 提供输出信息，用于向扩展模块写入信息。为此，通过 LAD/STL/FBD 编辑器在 OB 1 中创建用户程序，并将程序下载到主站系统的 CPU 中。

【编程】

步骤：

1) 在 OB1 中编写调用 FC1 (DP_SEND) 和 FC2 (DP_RECV) 的程序，实现 DP 主站读写从站 16 个字节数据，如下图所示：

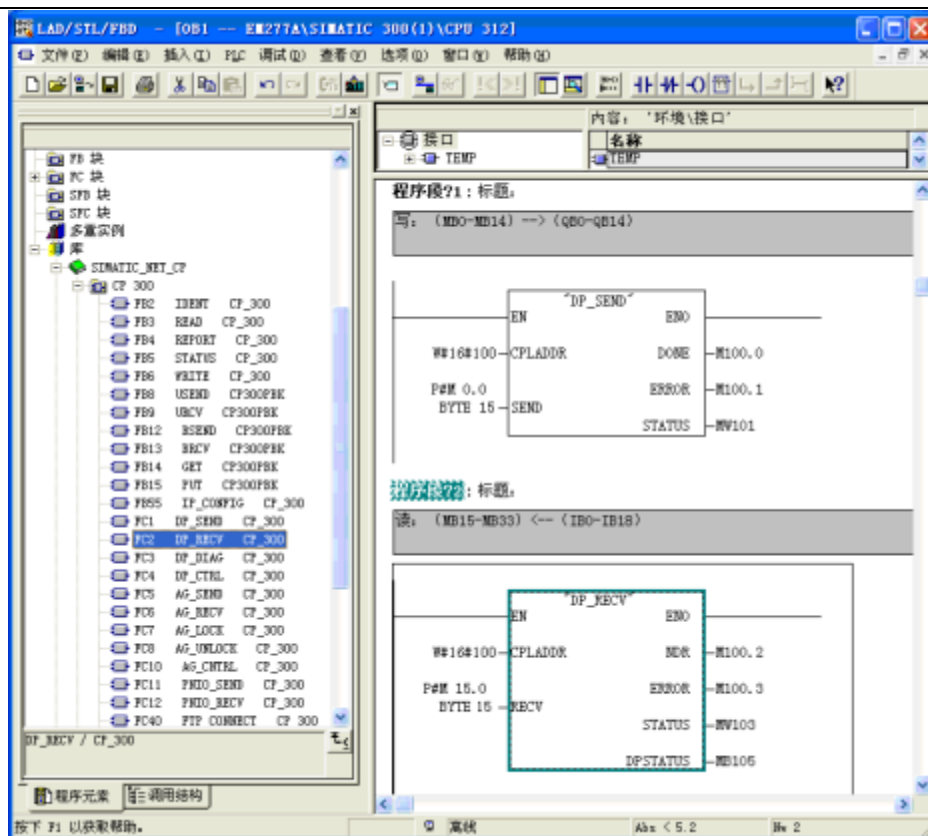


图 H-20 编程界面

程序功能说明：

通过执行 FC1 (DP_SEND) 将 CPU312 的内存 MB0-MB14 的数据输出到 SM277B 扩展模块的 QB0-QB14；通过执行 FC2 (DP_RECV) 将 SM277B 扩展模块 IB0-IB14 数据输入到 CPU312 的 MB15-MB29 内存。

CPLADDR：即 CP342-5 的地址 256 (16#100)。



提示

若选用的是带 DP 通信口的 CPU (如：CPU313C-2DP)，则无需调用 FC1 和 FC2，直接访问组态 SM277B 模块所分配的地址 (如 IB0-IB14、QB0-QB14)。

2) 保存用户程序。

【调试】

按以下步骤进行调试：

- 1) 连接好 CP 342-5 与 SM277B 的通信电缆，设置好 SM277B 的地址和中端电阻，接通系统各设备的电源。
 - 2) STEP7 接通主站，并将程序下载到主站系统的 CPU 中。
 - 3) 将主站置为 RUN。
 - 4) 在 STEP7 的变量表中设置和监控要传送的输出输入数据及程序执行的状态结果，观察 SM277B 扩展模块的输出状态，观察 SM277B 模块的状态指示灯 (LED)。
- 调试过程中可能会出现一些故障，故障的诊断请参考 N.3.3 诊断。

M.3.3 诊断

可以通过 LED 的状态和 STEP 7 的故障信息诊断以及特定的编程诊断来诊断 PROFIBUS DP 主从站网络。

通过 SM277B 诊断

通过诊断 SM277B 的 LED 状态灯可提供与主站连接方面的诊断信息以及 SM277B 固件自诊断的信息。

接通 SM277B 的电源之后，标示为“ON”(绿)的 LED 指示灯将亮起。

如果 BF 和 SF LED 保持熄灭状态，则 SM277B 正常运行。

如果 BF 或 SF LED 亮起，则说明扩展模块、地址设定超范围、接线或者主程序组态中有错误。

SM277B 的 LED 灯的功能如下表所示：

LED	颜色	功能说明
ON	绿色	SM277B 通电之后电源指示灯亮起。该LED由SM277B的硬件控制。
SF(系统故障)	红色	如果SM277B发现扩展模块故障或地址设定超范围，则 System Fault(系统故障)LED亮起。
BF(总线故障)	红色	没有与主站进入DP交换时，该LED将闪烁。

如果主系统的运行无故障，则错误的原因可能为：

- STEP 7 中的硬件组态没有与网络中的设备完全匹配。这时需要更改硬件组态，并将更改后的组态下载到 DP 主站 CPU 中。
- STEP 7 组态了错误参数。
- STEP7 中组态的地址和 SM277B 的实际 PROFIBUS 地址不相同，或者 SM277B 的实际 PROFIBUS 地址设置为 126 或 127 这类非有效地址。如果 SM277B 的实际地址无误，则需要更改 STEP 7 中从站设备的 PROFIBUS 地址，并重新将组态下载到 DP 主站 CPU 中。如果 SM277B 的实际地址有误，则需更改 SM277B 的实际地址，更改后需要将 SM277B 断电并重新上电。
- 未正确设置 PROFIBUS 网络的终端电阻。如果处于网络最后一个节点需拨到 ON，否则拨到 OFF。
- PROFIBUS 接线不正确或 PROFIBUS 电缆已损坏。检查接线和电缆并确保无误。

通过 STEP7 诊断

【在 HW-Config 中读取诊断信息】

对于所有 SIMATIC S7/M7 系列模块，均可调用 S7 诊断。对于基本单元和 I/O 模块，S7 诊断的结构是相同的。为了在 STEP 7 中进行诊断，可以读主站设备的诊断缓冲区或者从 SM277B 读诊断信息。

【确定故障】

在 HW-Config 中，可按照以下步骤确定故障所在：

- 通过菜单 Station >Open ONLINE 打开项目的在线窗口。
- 查找正在显示诊断设备的状态和故障情况的符号。按 F1 键可以打开诊断符号的帮助页面，查看诊断的注释。
- 选择菜单 PLC >Faulty Modules 显示故障模块列表。如果有故障，则打开在线视图后，视图

将不会自动更新。选择 **view>update** 显示当前的状态。

【读取主站和从站的诊断信息】

在 HW-Config 中，可以打开模块信息获取详细的诊断信息。双击显示故障的模块。

- 对于 DP 主站：选择 **Diagnostic Buffer** 标签，察看模块的诊断信息。
- 对于 SM277B 从站设备：选择 **General** 标签察看模块状态。选择 **DP Slave diagnostics** 标签，并点击 **Hex** 格式按钮可显示 SM277B 的诊断字节。

M.3.4 诊断模块电源预算

SM277B 的电源预算是计算所提供的电源功率是否足够提供给它所连接的扩展模块。为了进行电源预算，需要对 5 VDC 和 24 VDC 的电源分别进行计算。

SM277B电源	5VDC电源	24VDC电源
电源供应	SM277B为系统中所有扩展模块提供一个5VDC逻辑电源。	SM277B提供一个24VDC传感器电源，向扩展模块现场输入点或其它需求的地方提供24VDC电源。
最大可用直流电源	SM277B扩展的所有模块附加电流和要求少于660mA	请确认功率需求不超过400mA的预算量。



警告

- 1、禁止同时使用 SM277B 直流传感器电源和任何外部电源为同一设备供电。
- 2、一台设备请勿使用两个电源。
- 3、为 SM277B 24VDC 传感器电源连接一个外部 24VDC 电源会导致两个电源互相冲突，这种冲突的结果可能导致电源寿命缩短或者两个电源均出现故障，随后可能导致 PLC 系统运行出现故障。意外操作可能造成严重的人身伤害甚至死亡，同时/或者造成设备损坏。

N 永久保存 V 内存功能库“CT_savevmem”的使用

功能介绍

CT_SAVEVMEM 作为一个库函数提供给用户使用。其功能是将用户需要保存的一段 V 内存的数据保存到永久性内存中，使得这些数据在很长一段时间内不丢失。



提示

- 适用于参数保存，不要过于频繁保存。
请勿在系统块中为需要使用的永久保存 V 内存地址设置断电保持。

目前所有 CTH200 CPU 均支持永久保存 V 内存库，具体数据存储大小见下表：

CPU 名称	数据存储空间
H224/ H226L/H226M	基本 8KB，不可扩展
H224X	基本 8KB，可扩展至 108KB
H226XL/H226XM/ H228XL	基本 10KB，可扩展至 110KB

安装说明

- 1) 添加库文件

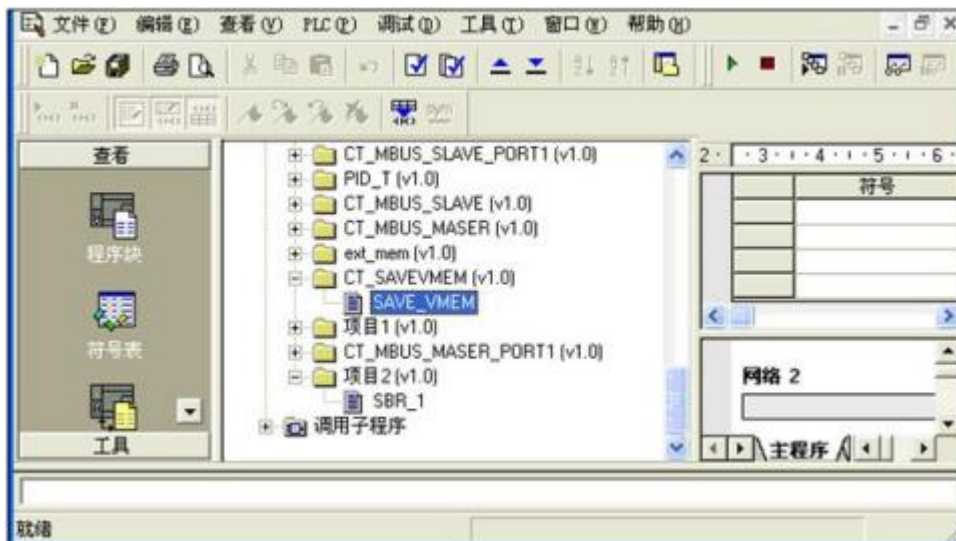
在“文件”--“添加/删除库”，找到库文件“ct_savevmem”，如下图所示。



在弹出的对话框中点“添加”，找到存放的“ct_savevmem.mwl”文件的位置，找到此文件，选中此文件后点“保存”，再在“添加/删除库”中选中刚添加的“ct_savevmem.mwl”文件，单击“确认”。

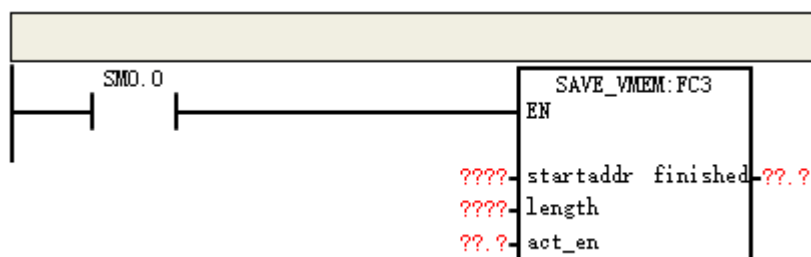


安装成功后，在目录树的“库”下可以看到新增加的“ct_savevmem”库：



2) 调用 CT_SAVEVMEM 库

单击要添加功能块的“网络”，双击“库”下面的“SAVE_VMEM”，就会在“网络”里出现相应的功能块，结果如下图所示：



提示

确保在写完成前 EN 一直处于接通状态，也就是说，最好使用 sm0.0 或 act_en 调用。

3) CT_SAVEVMEM 库功能说明

参数地址	说明	类型	备注
str_addr	V 内存的起始地址	WORD	可以是常量或变量。 例如起始地址 VW500 的 str_addr 是 500。
length	长度（以字为单位）	WORD	要永久保存连续V内存的长度。 例如 VB500-VB4499 的长度是 2000，或 VW500-VW4498 的长度是2000。
act_en	写允许位	BOOL	此位为 1 时开始写永久内存，写结束后此位自动复位，写入期间此位要保持为 1。
finished	写结束标志位	BOOL	写入开始时此位自动复位，写结束时自动置位为 1。

注：写的总长度为字的整数倍。

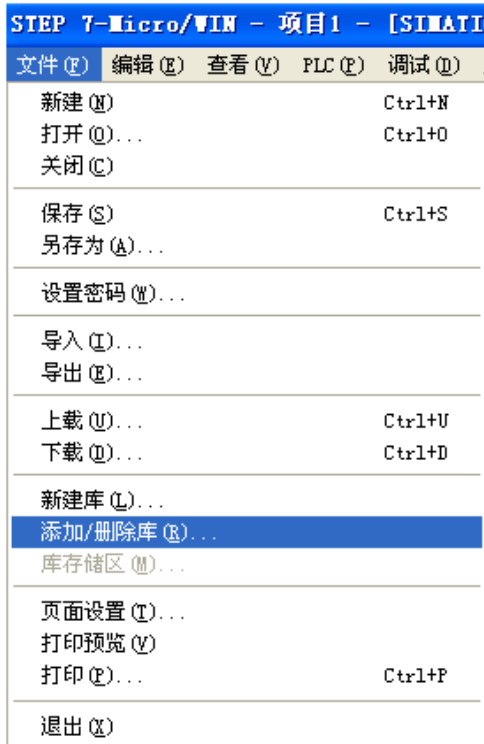
0 SM253 运动控制模块库的使用

0.1 功能介绍

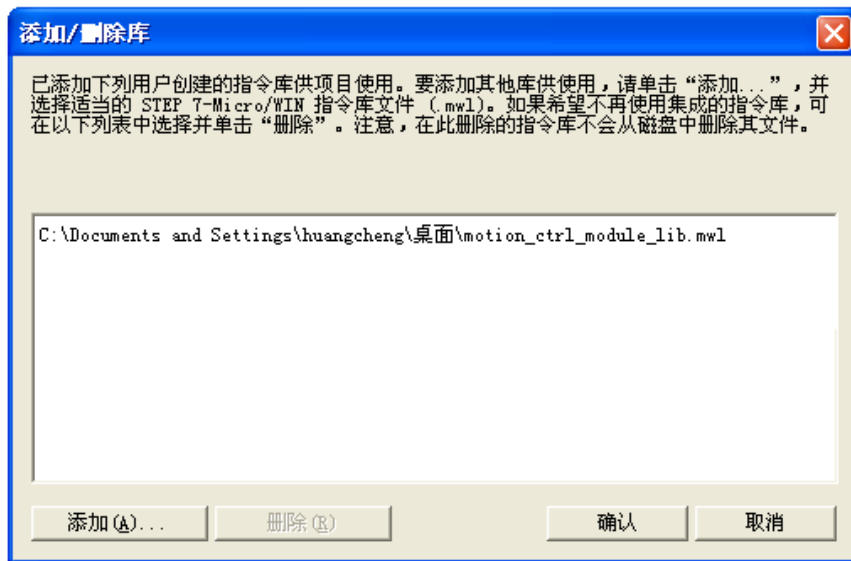
SM253 运动控制库是 CTH200 系列扩展模块 SM253 运动控制模块的专用库，可实现多轴独立控制功能

0.2 安装说明

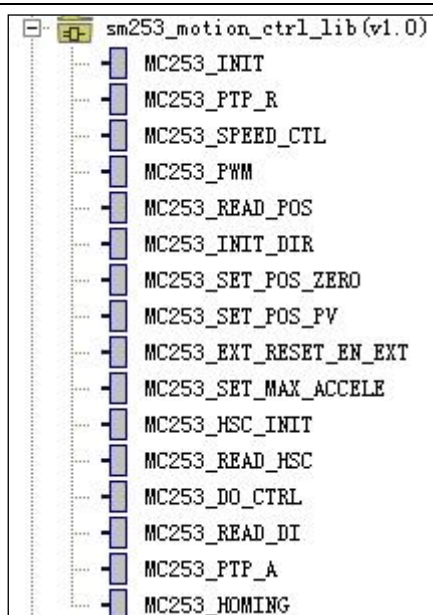
选择菜单项“文件”>“添加/删除库”，找到库文件“sm253_motion_ctrl_lib.mwl”，如下图所示：



浏览至您保存库文件的目录。选择库文件，并点击“添加”：



添加完成后单击“确认”完成添加，查看一下指令树中的库节点。可以看到新库“sm253_motion_ctrl_lib”已经被添加到指令树中



O.3 sm253_motion_ctrl_lib 库指令一览表

表 J-1 库指令一览表

函数名	指令名称、功能	是否支持
MC253_EXT_RESET_EN	外部复位坐标使能指令	SM253 模块不支持
MC253_INIT_DIR	配置电机方向指令	支持
MC253_READ_POS	读位置指令	支持
MC253_PTP_R	单轴相对运动指令	支持
MC253_CIRCLE_R	两轴圆弧插补运动指令	SM253 模块不支持
MC253_SPEED_CTL	速度控制指令	支持
MC253_SET_POS_ZERO	软件回零指令	支持
MC253_SET_POS_PV	设置目标位置指令	支持
MC253_LINE_R	两轴直线插补运动指令	SM253 模块不支持
MC253_EXT_RESET_EN_EXT	外部复位坐标使能指令II	支持
MC253_SET_MAX_ACCELE	设置最大加速度指令	支持
MC253_SET_CI_MODE	设置连续插补功能指令	SM253 模块不支持
MC253_PWM	脉冲宽度调制指令	支持
MC253_INIT	运控模块初始化指令	支持且必须在程序初始调用
MC253_DO_CTRL	控制模块输出指令	支持
MC253_READ_DI	读取模块输入状态指令	支持
MC253_HSC_INIT	设置模块高速计数器指令	支持
MC253_READ_HSC	读取模块高速计数器状态指令	支持
MC253_PTP_A	单轴绝对运动指令	支持
MC253_HOMING	回原指令	支持

O.4 sm253_motion_ctrl_lib 库指令详解

对 Micro/Win 编程而言，运控模块的指令和 CPU 的指令格式是一样的，而区别在于所使用的指令

库不同,CPU 指令前缀为 MC,而运控模块 SM253 的指令前缀为 MC253。CPU 指令的轴号为 0~3,而运控模块指令从 0 开始编号,可以往下延续,系统自动分配到每个运控模块上,如:运控模块指令轴号 0~1,分别对应第一个运控模块的 0 轴和 1 轴;运控模块指令轴号 2~3,分别对应第二个运控模块的 0 轴和 1 轴;类推,若超出运控模块范围,则认为参数错误,如:只有 1 个运控模块(只有 2 轴),而运控模块指令指定轴 2,则轴 2 找不到对应,从而无法使用。

同样地,模块高速计数器与 CPU 高速计数器不是统一编号的,使用模块的高速计数器必须调用模块指令,模块高速计数器从 0 开始编号,可以往下延续,系统自动分配到每个运控模块上,直至模块不足分配。如:模块指令高速计数器标号 0~1,分别对应第一个模块的 HSC0 和 HSC1;模块指令高速计数器 2~3,分别对应第二个模块的 HSC0 和 HSC1。

※ 特别注意

1、初始化和库存储区

使用 SM253 运动控制模块,要求在系统程序初始调用指令 MC253_INIT 以初始化系统控制变量(需且仅需一次,即只能用 SM0.1 调用)。注意,SM253 运控模块使用了 V 内存空间,用户可通过“库存储区分配”自行指定库存储区地址,在用户程序中用户不得使用库所占用的地址空间。(若 CPU 后只挂了 1 个 SM253 模块,则库占用了 218 个字节;若挂了 2 个,则库占用了 236 个字节;若挂了 3 个,则库占用了 254 个字节;若挂了 4 个,则库占用了 272 个字节;若挂了 5 个,则库占用了 290 个字节。)

2、模拟量滤波

当使用 SM253 运动控制模块时,必须在上位机软件系统块->输入滤波器->模拟量中将 SM253 运动控制模块对应的通道清除滤波功能,并将系统块下载到 PLC;否则将造成 SM253 运控模块的库指令不可使用。(查看 PLC 信息,其中 4 入 4 出的模拟量模块对应的通道即为 SM253 运控模块对应的通道。)

3、通信状态位

SM253 运控模块除了 MC253_INIT 之外的其它所有指令新增了通信状态位,若此状态位报警则表明通信有故障,通信信息未必可靠。此时要查看总线连接是否正常或者运控通道的模拟量滤波是否清除等。

4、脉冲输出指令的执行顺序

同轴脉冲输出指令(MC253_PTP_R、MC253_SPEED_CTL、MC253_PWM)最多只有 3 份数据缓冲区,即同一时刻只处理 3 条该类指令。当同一轴(比如均为 0 轴)上有多条该类指令时,若缓冲区未满(已使能的输出指令不足 3 条),则新使能的指令可得到及时处理,那么指令的执行顺序与指令使能的时间顺序一致,而与指令位置顺序没有必然关系;若缓冲区已满(已使能的输出指令不小于 3 条),则新使能的指令不能得到及时处理,而只能待之前使能的指令结束并释放缓冲区后由程序扫描按位置先后顺序获得处理,即是执行顺序与其位置顺序有关,而不一定是按指令使能的时间顺序。可简单理解为“未满 3 按时间,满 3 按位置”(如例 1 所示)。

故,要实现一定的运动轨迹,若这些运动指令的位置顺序与期望实现的轨迹一致,则可一次性将这些指令全部使能。若这些运动指令的位置顺序与期望实现的轨迹不一致,则必须在使能的指令未满 3 条的情形下去使能期望轨迹所对应的指令(如例 2 所示)。

例 1) 一次性使能所有指令

缓冲区未填满 3 份时使能的指令(指令 A/B/C)按使能时间顺序执行。如下表:将五条指令一次性全部使能,使能的顺序依次为:指令 A->指令 C->指令 B->指令 E->指令 D,如表列 1 所示;则

先使能的三条指令按照使能先后依次执行，即执行顺序为：指令 A-->指令 C-->指令 B，如表列 3 所示：

指令使能顺序	指令位置顺序	指令执行顺序
1	0—AXIS 1—RUN 指令 A	1
3	0—AXIS 1—RUN 指令 B	3
2	0—AXIS 1—RUN 指令 C	2
5	0—AXIS 1—RUN 指令 D	--
4	0—AXIS 1—RUN 指令 E	--

缓冲区已填满 3 份时剩余已使能的指令（指令 D/E）按其位置顺序依次执行。当指令 A 执行结束后释放缓冲区，由于程序自上而下扫描，指令 D 先获得该缓冲区；待指令 C 执行结束并释放缓冲区后，指令 E 获得该缓冲区。

指令使能的顺序依次为：指令 A-->指令 C-->指令 B-->指令 E-->指令 D，如表列 1 所示；指令执行的顺序为：指令 A-->指令 C-->指令 B-->指令 D-->指令 E，如表列 3 所示：

指令使能顺序	指令位置顺序	指令执行顺序
1	0—AXIS 0—RUN 指令 A	1
3	0—AXIS 0—RUN 指令 B	3
2	0—AXIS 1—RUN 指令 C	2
5	0—AXIS 1—RUN 指令 D	4
4	0—AXIS 1—RUN 指令 E	5

例 2) 先使能三条指令，等有指令执行完成释放缓冲区后，再接着使能剩余的指令。

缓冲区未填满 3 份时使能的指令（指令 A/B/C）按使能时间顺序执行，如下表：先使能 3 条指令，使能顺序依次为：指令 A-->指令 C-->指令 B，如表列 1 所示，则此三条指令按照使能先后依次执

行，即执行顺序为：指令 A-->指令 C-->指令 B，如表列 3 所示：

指令使能顺序	指令位置顺序	指令执行顺序
1	0—AXIS 1—RUN 指令 A	1
3	0—AXIS 1—RUN 指令 B	3
2	0—AXIS 1—RUN 指令 C	2
--	0—AXIS 0—RUN 指令 D	--
--	0—AXIS 0—RUN 指令 E	--

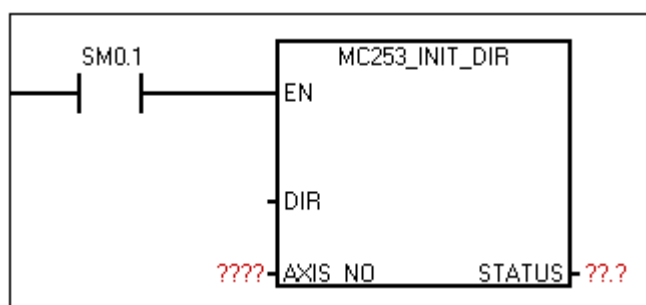
待指令 A 执行完成并释放缓冲区后，先使能指令 E，则指令 E 获得该缓冲区；待指令 C 执行完成并释放缓冲区后，再使能指令 D，则指令 D 获得该缓冲区。

指令使能的顺序依次为：指令 A-->指令 C-->指令 B-->指令 E->指令 D，如表列 1 所示；指令执行的顺序为：指令 A-->指令 C-->指令 B-->指令 E->指令 D，如表列 3 所示：

指令使能顺序	指令位置顺序	指令执行顺序
1	0—AXIS 0—RUN 指令 A	1
3	0—AXIS 0—RUN 指令 B	3
2	0—AXIS 1—RUN 指令 C	2
5	0—AXIS 1—RUN 指令 D	5
4	0—AXIS 1—RUN 指令 E	4

配置电机方向指令

- ① 函数名：MC253_INIT_DIR



② 功能：配置电机的方向



提示

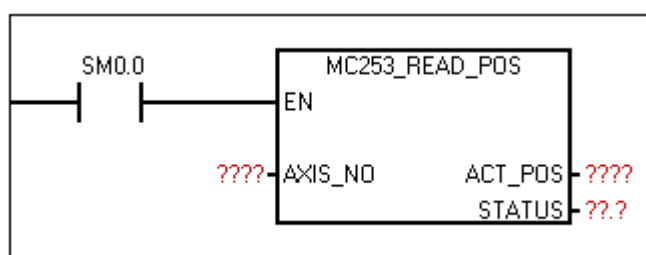
执行此指令只在 CPU 上电第一个扫描周期执行一次。

③ 参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
DIR	IN	配置方向信号为正向时的有效电平。 DIR为1时，设置对应方向轴输出“1”时为电机正转。 DIR为0时，设置对应方向轴输出“0”时为电机反转。	Bool	0~1	默认值：1，即默认方向轴输出为“1”，为电机正转。
AXIS_NO	IN	设置轴号（每个SM253模块有2轴，轴号范围由运控模块数目决定）	Byte	0~255	
STATUS	OUT	通信状态标志位 1：通信超时	Bool	0~1	

读位置指令

① 函数名：MC253_READ_POS



② 功能：读取每轴的绝对坐标值。一旦设定原点坐标后，那么该值会根据输出的脉冲和方向的关系进行代数计算：正转输出一个脉冲：+1，反转输出一个脉冲：-1。最后得到的是一个以设定点为原点的绝对坐标。

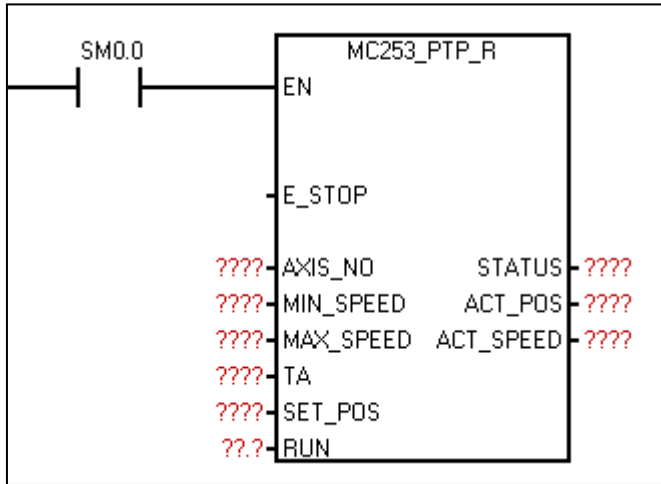
③ 参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
AXIS_NO	IN	设置轴号（每个SM253模块	Byte	0~255	

		有2轴，轴号范围由运控模块数目决定)			
ACT_POS	OUT	当前轴的绝对坐标（1个脉冲代表1个单位坐标）	Dint	-2147483648 ~ +2147483647	此指令无错误状态输出，轴号必须设置正确。
STATUS	OUT	通信状态标志位 1: 通信超时	Bool	0~1	

单轴相对运动指令

① 函数名: MC253_PTP_R



② 功能：用作单轴点对点控制（单轴定长驱动）。调用一次可输出固定脉冲，通过最大、最小速度和加减速时间的设定，输出的脉冲在启动时会逐渐的加速到最大的速度，当脉冲数快要跑完时，脉冲的频率会自动减下来，以防止在启动或停止时的机器的惯性太大而引起振动或卡死。

③ 参数

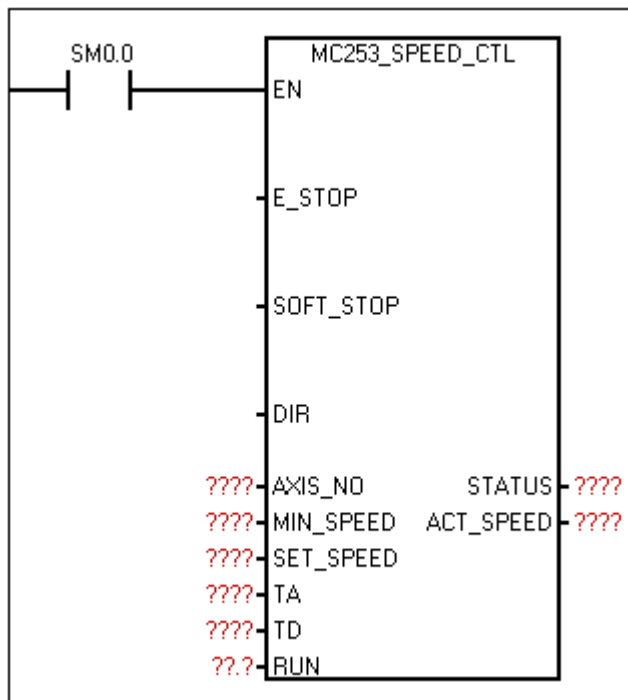
参数名	输入输出属性	参数描述	类型	数值范围	备注
E_STOP	IN	紧急停止位。 1: 有效 0: 无效	Bool	0/1	1、只有 Run ==1 与E_Stop ==0 时才能运行。 2、当E_STOP 为1 时,RUN 内部复位。
AXIS_NO	IN	设置轴号（每个SM253 模块有2轴，轴号范围由运控模块数决定）	Byte	0~255	该参数在运行过程中不能修改。
MIN_SPEED	IN	最小速度，即启动时或停止时的速度。单位：HZ	Dword	500~200000	➤ 最小速度的设定要小于最大速度。 ➤ 此参数在运行过程中可以修改。
MAX_SPEED	IN	最大速度，即运行中的最大速度。单位：HZ	Dword	500~200000	
TA	IN	加速/减速时间。 单位：ms	Dword	0~10000 (见附注	该参数在运行过程中可以修改（加速时间设置见错误!未找到引用源。附注 2)

SET_POS	IN	输出的脉冲数，分正负。 正脉冲数表示沿X轴的正方向，负脉冲数表示沿着 X 轴的负方向。	Dint	-2147483648 ~ +2147483647	该参数在运行过程中可以修改。 当新设定值大于已输出的脉冲数，那么最后输出的脉冲会以新设定值为准。 当新设定值小于已输出脉冲数，那么会马上停止脉冲输出。								
RUN	IN/OUT	运行使能位。 1: 有效 0: 无效	Bool	0/1	1、只有 RUN ==1 与 E_STOP ==0 时才能运行。 2、当运行完成后，RUN 内部复位。 3、当 E_STOP 为 1时，RUN 内部复位。								
STATUS	OUT	输出状态字节： <table border="1" style="margin-left: 20px;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> </table> Bit0: 参数配置错误标志 1—参数配置错误 0—参数配置正常 Bit1: 运行标志 1—正在运行，该指令正在输出脉冲，且指令未执行完。 0—不运行，因公共资源被其他指令占用，所以指令还没得以运行；或者指令已经运行完毕。 Bit2: 完成标志 1—完成，指令执行完毕。 0—未完成，指令未执行或指令正在执行中但未完成。 Bit3: 忙标志 1—忙标志有效，该轴正在被其它指令占用。 0—忙标志无效，指令正在执行或此执行已完成。 Bit4: 模块急停标志（见附注3） 1—模块急停标志有效，该轴被外部条件禁止运行。 0—模块急停标志无效。	7	6	5	4	3	2	1	0	Byte	0~255	Bit0: 1、只对轴参数和TA配置错误进行判断； 2、MIN_SPEED/MAX_SPEED等参数不作报错，会自动设置成一个最接近的合理值。 3、253模块未被PLC识别或者是没有连接该模块时会提示参数配置错误。
7	6	5	4	3	2	1	0						

		Bit5~Bit6: 预留 Bit7: 指令通信状态标志 1—通信超时 0—无超时			
ACT_POS	OUT	当前的相对坐标或本指令已输出的脉冲数。	Dint	-2147483648 ~ +2147483647	
ACT_SPE ED	OUT	当前实际运行速度。	Dword	500~200000	

速度控制指令

① 函数名: MC253_SPEED_CTL



② 功能: 控制单轴输出脉冲的频率, 可任意时候改变输出脉冲的频率(速度)。当接收到软停止命令时, 会自动减速停止。当收到紧急停止命令时, 会马上停止脉冲输出, 不经过减速。

③ 参数

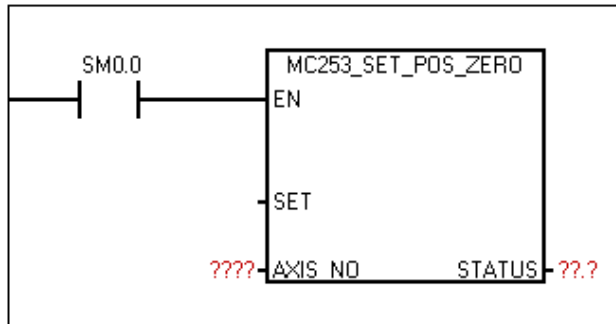
参数名	输入输出属性	参数描述	类型	数值范围	备注
RUN	IN	运行使能位。 1: 有效, 0: 无效。	Bool	0/1	1、只有 RUN ==1 与 E_Stop == 0 与 SOFT_STOP==0时才能运行。 2、当运行完成后, RUN内部复位。 3、当E_STOP 为1时, RUN内部复位。
E_STOP	IN	紧急停止位。1: 有效,	Bool	0/1	1、只有 RUN ==1 与

		0: 无效。 当收到有效紧急停止命令后, 输出脉冲会马上停止, 不经过减速。			E_Stop ==0 与 Soft_Stop==0时才能运行。 2、当E_STOP 为 1 时, RUN 内部复位。								
SOFT_STOP	IN	软停止位。1: 有效, 0: 无效。 当收到有效软停止命令时, 输出脉冲会减速停止。	Bool	0/1	只有RUN ==1 与 E_Stop==0 与 SOFT_STOP==0时才能运行。								
DIR	IN	脉冲的方向位	Bool	0/1	该参数在运行过程中能修改。								
AXIS_NO	IN	设置轴号 (每个SM253 模块有2轴, 轴号范围由运控模块数目决定)	Byte	0~255	该参数在运行过程中不能修改。								
MIN_SPEED	IN	最小速度, 即启动时或停止时的速度。 单位: HZ	Dword	0~200000	1、设定速度为0时, 没有脉冲输出; 设定最小速度非0且小于92时, 模块默认为92; 若设定速度小于最小速度, 模块默认将最小速度的值修改为设定速度的值。 2、该参数在运行过程中可以修改。								
SET_SPEED	IN	设定速度, 在收到停止命令前, 输出脉冲会加速或减速到此速度。	Dword	0~200000									
TA	IN	加速时间, 从最小速度到设定速度的加速时间。 单位: 毫秒	Dword	0~10000 (见附注1)	注: 该参数在运行过程中可以修改。(加速时间设置见附注)。								
TD	IN	减速时间, 从设定速度到最小速度的减速时间。 单位: 毫秒	Dword	0~10000 (见附注1)									
STATUS	OUT	输出状态字节: <table border="1" style="margin-left: 20px;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> </table> Bit0: 参数配置错误标志 1—参数配置错误 0—参数配置正常 Bit1: 运行标志 1—正在运行, 该指令正在输出脉冲, 且指令未执行完。 0—不运行, 因公共资源被其他指令占用, 所以指令还没得以运行; 或者指令已经运行完毕。 Bit2: 完成标志 1—完成, 指令执行完毕。	7	6	5	4	3	2	1	0	Byte	0~255	Bit0: 1、只对轴参数和 TA/TD配置错误进行判断; 2、MIN_SPEED/SET_SPEED等参数不作报错, 会自动设置成一个最接近的合理值。 3、253模块未被PLC 识别或者是没有连接该模块时会提示参数配置错误。
7	6	5	4	3	2	1	0						

		<p>0—未完成，指令未执行或指令正在执行中但未完成。</p> <p>Bit3: 忙标志 1—忙标志有效，该轴正在被其它指令占用。 0—忙标志无效，指令正在执行或此执行已完成。</p> <p>Bit4: 模块急停标志（见附注3） 1—模块急停标志有效，该轴被外部条件禁止运行。 0—模块急停标志无效。</p> <p>Bit5~Bit6: 预留</p> <p>Bit7: 指令通信状态标志 1—通信超时 0—无超时</p>			
ACT_SPEED	OUT	当前速度（频率）输出	Dword	500~2000 00	该值可能跟实际值会有一点偏差，最大不超过5K，跟加速时间和设定的速度有关。

软件回零指令

① 函数名: MC253_SET_POS_ZERO



② 功能: 复位绝对坐标。



提示

当机器运动到某一位置时，调用该指令，相当于把该轴的原点设定在该位置。那么以后每次调用“读绝对坐标”命令，就能得到相对于该点的坐标值。

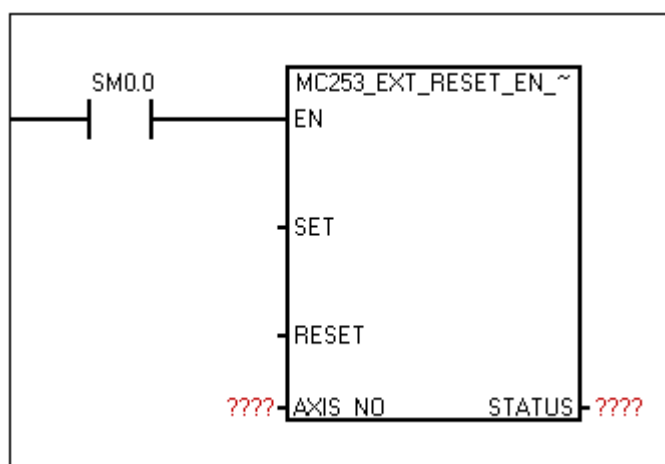
③ 参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
SET	IN	清零功能使能位。 在SET上升沿把绝对坐标清0，每次调用时，SET应先置0，然后再置1。	Bool	0~1	

AXIS_NO	IN	设置轴号（每个SM253模块有2轴，轴号范围由运控模块数目决定）	Byte	0~255	
STATUS	OUT	通信状态标志位 1：通信超时	Bool	0~1	

外部复位坐标使能指令 II

① 函数名：MC253_EXT_RESET_EN_EXT



② 功能：当调用该指令，设置是否使能外部 IO 复位绝对坐标值。



提示

轴号与外部复位信号的对应关系：

轴 0 ——I0.2（MC253_HSC0）

轴 1 ——I0.6（MC253_HSC1）

③ 参数

参数名	输入输出属性	参数描述	类型	数值范围	备注								
SET	IN	SET上升沿，设置外部复位使能，每次调用时，SET应先复位，然后再置1。	Bool	0~1									
RESET	IN	RESET上升沿，禁止外部复位使能，每次调用时，RESET应先复位，然后再置1。	Bool	0~1									
AXIS_NO	IN	设置轴号（每个SM253模块有2轴，轴号范围由运控模块数目决定）	Byte	0~255									
STATUS	OUT	状态位： <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> </table> Bit0：复位状态标志位 1—复位完成	7	6	5	4	3	2	1	0	Byte	0~255	
7	6	5	4	3	2	1	0						

		0—复位未完成 Bit1~Bit6: 预留 Bit7: 通信状态标志位 1—通信超时 0—无超时			
--	--	--	--	--	--

④ 使用说明

设 0 轴调用此指令。在 SET 上升沿使能外部复位功能之后，若 I0.2 检测到“有效复位信号”，则系统复位 0 轴绝对坐标，同时 STATUS 置位指示复位完成。在 RESET 上升沿禁止外部复位功能之后，即使 I0.2 检测到“有效复位信号”，系统亦不复位 0 轴绝对坐标，同时 STATUS 清零指示非复位状态。

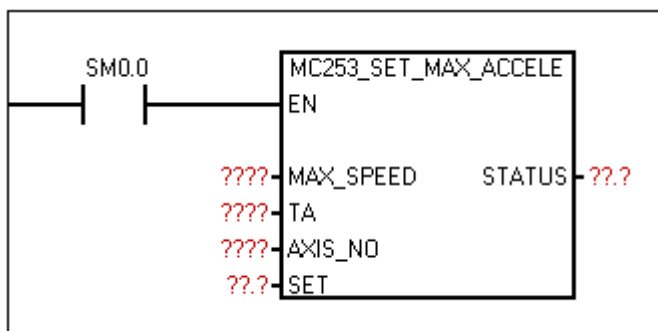


提示

所谓“有效复位信号”，每轴的复位信号都有一个外部 IO 与其对应，同时由相应的寄存器设置其有效电平。如，0 轴对应 I0.2，由 MC253_HSC0 指令 MC253_HSC_INIT 参数 CONTROL 的第 0 位可以设置 0 轴的有效复位电平，当设置为 0 时，则 0 轴的有效复位信号为 I0.2 的高电平状态；当设置为 1 时，则 0 轴的有效复位信号为 I0.2 的低电平状态；此设置当且仅当相应高速计数器（0 轴对应为 MC253_HSC0）得到使能时才有效，否则（即没有使能高速计数器）系统默认高电平为有效复位信号，如 0 轴，则 I0.2 高电平为有效复位信号。其它轴同理，各轴相关控制对应关系见此节②。

设置最大加速度指令

① 函数名：MC253_SET_MAX_ACCELE



② 功能：设置最大加速度 (= MAX_SPEED/TA) (TA≠0) (若没有调用此指令，则认为没有设置最大加速度)

③ 参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
MAX_SPEED	IN	长轴最大速度，即运行中的最大速度。 单位：HZ	Dword	0~200000	运行过程中可以修改。
TA	IN	加速/减速时间。 单位：ms	Dword	0~10000 (见附注1)	运行过程中可以修改； 若TA=0，则认为没有设置最大加速度。
AXIS_NO	IN	设置轴号（每个SM253模块	Byte	0~255	

		有2轴，轴号范围由运控模块数目决定)			
SET	IN	在以上参数确定后，给SET一个上升沿以使设置生效。	Bool	0~1	
STATUS	OUT	通信状态标志位 1: 通信超时	Bool	0~1	

④ 使用说明

若设置 X 轴参数 TA=0，或者 X 轴没有调用此指令，则认为 X 轴没有设置最大加速度；否则认为 X 轴设置有最大加速度，其值 MAX_ACCELE=MAX_SPEED/TA。此指令的意义在于：

- 可以设置一个合适的加速度限制某轴上各指令的加速度

如 PTP 指令，设置 AXIS_NO=0，MIN_SPEED=1000，MAX_SPEED=11000，TA=500，则理论上此 PTP 运动的加速度为 20HZ/ms ($= (MAX_SPEED - MIN_SPEED) / TA$)；若 0 轴调用了 MC253_SET_MAX_ACCELE 指令设置最大加速度为 15HZ/ms，则 PTP 实际可行的加速度为 15HZ/ms。（MC253_LINE_R 和 MC253_CIRCLE_R 亦然）

- 某轴上某指令要获取最大的加速度运行

如 PTP 指令要以最大加速度运行，可以先在同轴上调用 MC253_SET_MAX_ACCELE 指令设置最大加速度（即 MC253_SET_MAX_ACCELE 指令的参数 TA 不能为 0，否则无法得到最大加速度），同时将 PTP 指令的参数 TA 设为 0。若没有设置最大加速度而 PTP 指令的 TA=0，则在 PTP 指令上警报参数故障。（MC253_LINE_R 和 MC253_CIRCLE_R 亦然）

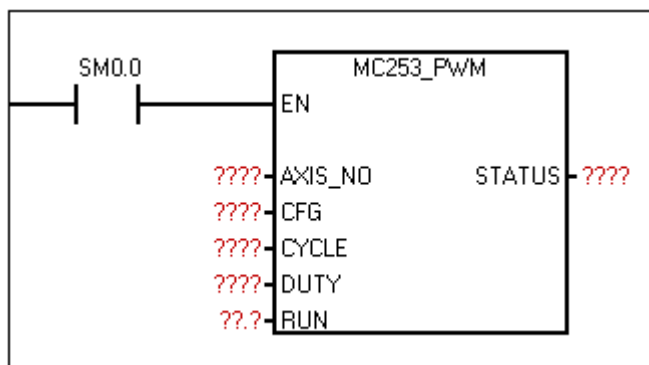


提示

1. 对于双轴指令 MC253_LINE_R 和 MC253_CIRCLE_R，若两轴均设置了最大加速度，则取两者中的较小值作为此双轴系统的最大加速度；若只有一轴设置了最大加速度，则取其为此双轴系统的最大加速度；若双轴均没有设置最大加速度，则此双轴系统对加速度没有限制。
2. 在确定该指令的参数 MAX_SPEED、TA、AXIS_NO 后，要给 SET 一个上升沿，这些参数才更新生效。

脉冲宽度调制指令

① 函数名：MC253_PWM



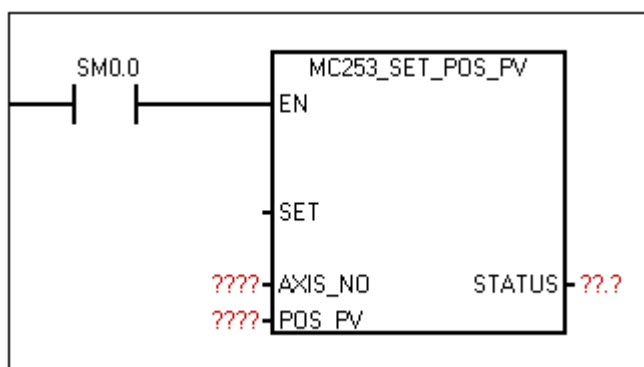
- ② 功能：通过设置周期和占空比参数，可以输出不同周期和占空比的脉冲。

③ 参数

参数名	输入输出属性	参数描述	类型	数值范围	备注								
AXIS_NO	IN	设置轴号（每个SM253模块有2轴，轴号范围由运控模块数目决定）	Byte	0~255	运行过程中不能修改。								
CFG	IN	基准时间单元 0: 1us, 1: 0.5ms	Byte	0~1	运行过程中不能修改。								
CYCLE	IN	脉冲的周期	Word	2~65535	运行过程中不能修改。								
DUTY	IN	脉冲的占空比	Word	0~65535	运行过程中不能修改。								
RUN	IN	运行使能	Bool	0~1									
STATUS	OUT	输出状态字节： <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> </table> Bit0: 参数配置错误标志 1—参数配置错误 0—参数配置正常 Bit1: 运行标志 1—正在运行，该指令正在输出脉冲，且指令未执行完。 0—不运行，因公共资源被其他指令占用，所以指令还没得以运行；或者指令已经运行完毕。 Bit2: 完成标志 1—完成，指令执行完毕。 0—未完成，指令未执行或指令正在执行中但未完成。 Bit3: 忙标志 1—忙标志有效，该轴正在被其它指令占用。 0—忙标志无效，指令正在执行或此执行已完成。 Bit4: 模块急停标志（见附注3） 1—模块急停标志有效，该轴被外部条件禁止运行。 0—模块急停标志无效。 Bit5~Bit6: 预留 Bit7: 指令通信状态标志 1—通信超时 0—无超时	7	6	5	4	3	2	1	0	Byte	0~255	Bit0: ● 只对轴参数配置错误进行判断； ● CYCLE/DUTY等参数不作报错，会自动设置成一个最接近的合理值。 ● 253模块未被PLC识别或者是没有连接该模块时会提示参数配置错误。
7	6	5	4	3	2	1	0						

设置目标位置指令

① 函数名: MC253_SET_POS_PV



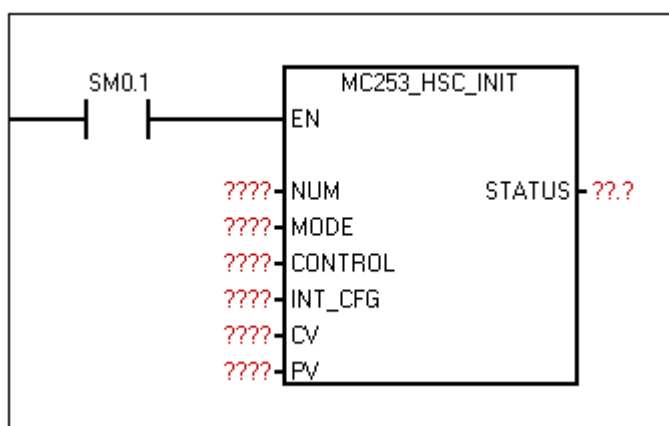
② 功能: 此指令用于将机器所处的绝对位置写入到模块。比如, 机器运行到某一位置时断电, 可将其此时所处的位置保存下来, 等下次上电时, 将此位置写回到模块, 则模块绝对位置计数起点与机器实际起点位置一致, 而机器不需回到原点; 若此位置刚好为原点, 则此指令与 MC253_SET_POS_ZERO 相同效果。

③ 参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
AXIS_NO	IN	设置轴号 (每个 SM253 模块有 2 轴, 轴号范围由运控模块数目决定)	Byte	0~255	
SET	IN	SET 上升沿, 指令使能, 每次调用时, SET 应先复位, 然后再置 1。	Bool	0~1	
POS_PV	IN	设定的目标位置, 分正负。输出正脉冲表示沿 X 轴的正方向, 负脉冲数表示沿着 X 轴的负方向。	Dint	-2147483648 ~ +2147483647	
STATUS	OUT	通信状态标志位 1: 通信超时	Bool	0~1	

设置模块高速计数器指令

① 函数名: MC253_HSC_INIT



②功能：配置高速计数器。



提示

执行此指令只在 CPU 上电第一个扫描周期执行一次。

③参数

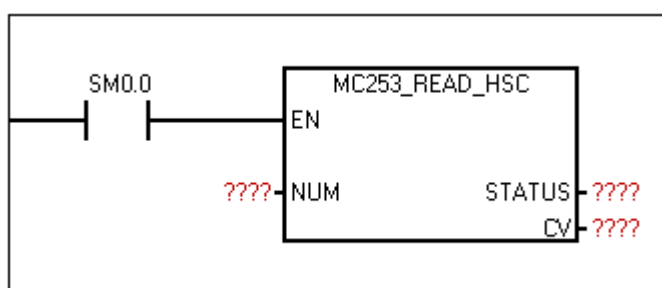
参数名	输入输出属性	参数描述	类型	数值范围	备注								
NUM	IN	高速计数器标号(每个 SM253 模块有 2 个高速计数器, 标号范围由运控模块数目决定)	Byte	0~255									
MODE	IN	计数模式	Byte	0~12	并不支持所有, 详见附表。								
CONTROL	IN	控制字: <table border="1" style="margin-left: 20px;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> </table> Bit0: 复位有效电平控制位 1—低电平 0—高电平 Bit1: 预留 Bit2: 正交计数器速率选择 1—1x 速率 0—4x 速率 Bit3: 方向控制位 1—增计数 0—减计数 Bit4: 更新方向 1—更新方向 0—不更新 Bit5: 更新预设值 1—写新的预设值 0—不更新 Bit6: 更新当前值 1—写新的当前值 0—不更新 Bit7: 有效位 1—有效 0—无效	7	6	5	4	3	2	1	0	Byte	0~255	
7	6	5	4	3	2	1	0						
INT_CFG	IN	中断配置 (预留)	Byte	预留	预留, 暂不支持中断功能。								
CV	IN	新的当前值	Dword	-2147483648 ~ +2147483647									
PV	IN	新的预设值	Dword	-2147483648 ~ +2147483647									
STATUS	OUT	通信状态标志位 1: 通信超时	Bool	0~1									

附表

模式	描述	输入点		
	MC253_HSC0	I0.0	I0.1	I0.2
	MC253_HSC1	I0.4	I0.5	I0.6
0	带内部方向控制的单相计数器	时钟		
1		时钟		复位
3	带外部方向控制的单相计数器	时钟	方向	
4		时钟	方向	复位
9	A/B相正交计数器	时钟 A	时钟 B	
10		时钟 A	时钟 B	复位

读取模块高速计数器状态指令

①函数名：MC253_READ_HSC



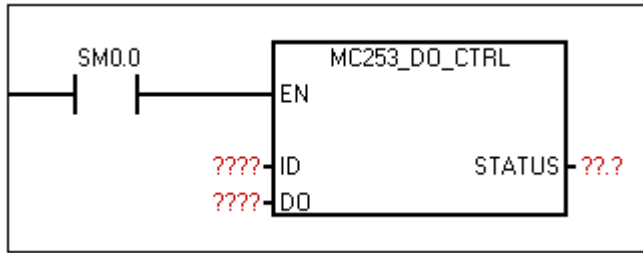
②功能：读取模块高速计数器的计数状态和计数当前值。

③参数

参数名	输入输出属性	参数描述	类型	数值范围	备注								
NUM	IN	高速计数器标号（每个 SM253 模块有 2 个高速计数器，标号范围由运控模块数目决定）	Byte	0~255									
STATUS	OUT	状态字： <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> </table> Bit0: 通信状态位 1—通信超时 Bit1~Bit4: 预留 Bit5: 当前计数方向位 1—增计数 Bit6: 当前值等于预设值位 1—等于 Bit7: 当前值大于预设值位 1—大于	7	6	5	4	3	2	1	0	Byte	0~255	
7	6	5	4	3	2	1	0						
CV	OUT	新的当前值	Dword	-2147483648 ~ +2147483647									

控制模块输出指令

①函数名：MC253_DO_CTRL



②功能：控制运控模块的输出值。

③参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
ID	IN	运控模块 ID 号	Byte	0~255	只适用于运控模块
DO	IN	模块输出端口值	Byte	0~255	DO 对应 8 路 Q 点输出，字节低位对应 Q 点低位，即 DO 的 bit0 对应 Q0.0，类推。
STATUS	OUT	通信状态标志位 1: 通信超时	Bool	0~1	

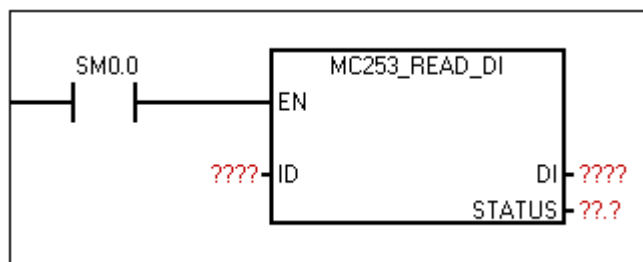


提示

- 运控模块的编号只是内部统一，即若 PLC 带 3 个模块，其中模块 0 为运控模块，模块 1 为非运控模块，模块 2 为运控模块，则模块 0 为 ID0，模块 1 不是运控模块不可控，模块 2 为 ID1。
- 若 CPU 后面挂单个运控模块，则由于传输延时，输出指令从使能到执行延时约为 780us；若 CPU 后面挂最多 5 个运控模块，则输出指令从使能到执行延时约为 930us。

读取模块输入状态指令

①函数名：MC253_READ_DI



②功能：读取控制运控模块的输入值。

③参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
ID	IN	运控模块 ID 号	Byte	0~255	只适用于运控模块
DI	OUT	模块输入端口值	Byte	0~255	DI: 对应 8 路 I 点输入，字节低位对应 I 点低位，即 DI 的 bit0 对应 I0.0，类推。
STATUS	OUT	通信状态标志位 1: 通信超时	Bool	0~1	

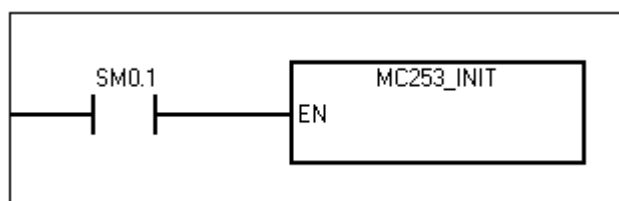


提示

- 运控模块的编号只是内部统一的，即若 PLC 带 3 个模块，其中模块 0 为运控模块，模块 1 为非运控模块，模块 2 为运控模块，则模块 0 为 ID0，模块 1 不是运控模块不可控，模块 2 为 ID1。
- 当某轴调用运动控制指令，若指令符合运行条件，则运行过程中该轴端口按指令输出脉冲或方向（对应端口不受指令 MC253_DO_CTRL 控制），运行结束后端口可恢复普通 IO 功能（即可接受指令 MC253_DO_CTRL 控制）；用户使用指令 MC253_DO_CTRL 当慎重，以免影响运动控制的使用需要。

运控模块初始化指令

① 函数名：MC253_INIT

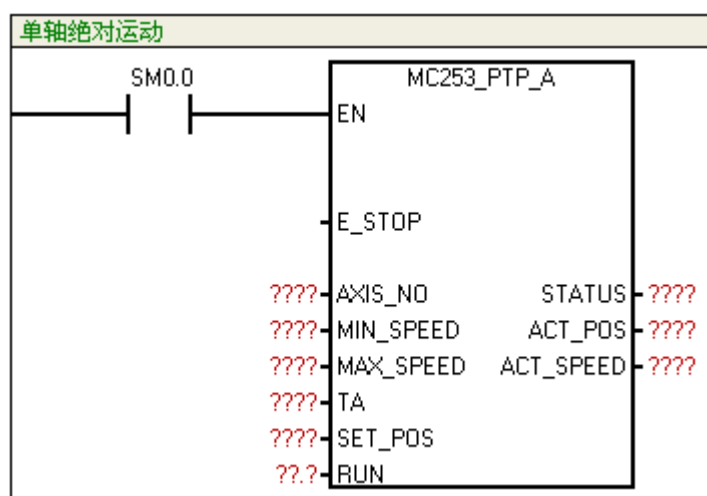


② 功能：初始化运控模块的系统控制变量（参考本章节**特别注意 1**）。

③ 参数：无

单轴绝对运动指令

① 函数名：MC253_PTP_A



② 功能：用作单轴点对点控制（非定长，而是定点）。调用一次可在原脉冲数基础上输出脉冲至指定脉冲数，通过最大、最小速度和加减速时间的设定，输出的脉冲在启动时会逐渐的加速到最大的速度，当脉冲数快要跑完时，脉冲的频率会自动减下来，以防止在启动或停止时的机器的惯性太大而引起振动或卡死。

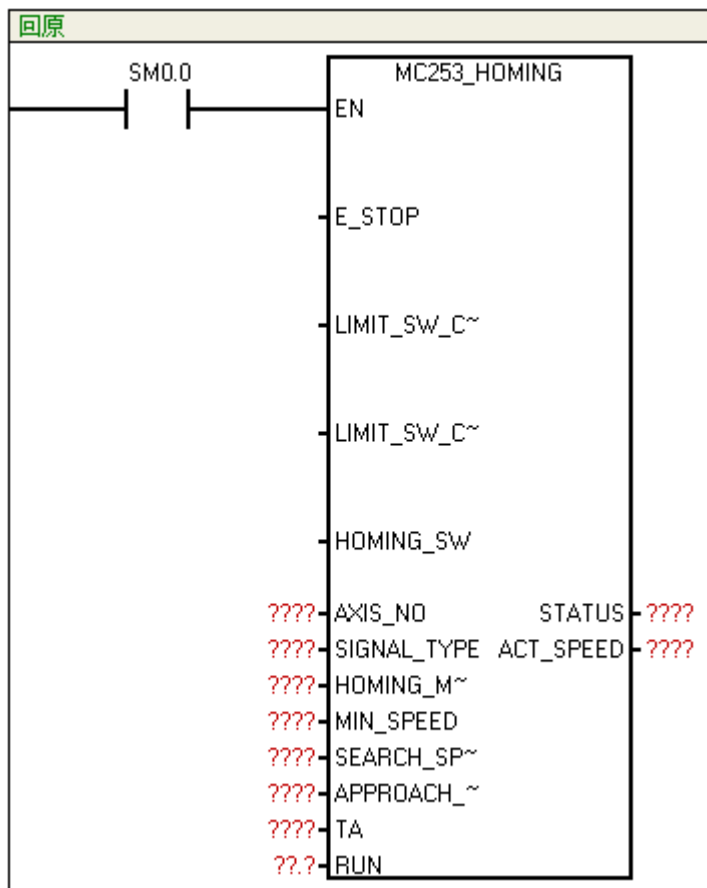
③ 参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
E_STOP	IN	紧急停止位。 1: 有效 0: 无效	BOOL	0/1	1、只有 Run =1 与 E_Stop=0 时才能运行。 2、当 E_STOP 为 1 时, RUN 内部复位。
AXIS_NO	IN	设置轴号	BYTE	0~255	该参数在运行过程中不能修改。
MIN_SPEED	IN	最小速度, 即启动时或停止时的速度。单位: HZ。	DWORD	500~200000	1、最小速度的设定要小于最大速度。 2、该参数在运行过程中可以修改。
MAX_SPEED	IN	最大速度, 即运行中的最大速度。单位: HZ	DWORD	500~200000	
TA	IN	加速 / 减速时间, 单位 ms	DWORD	0~10000 (见附注1)	该参数在运行过程中可以修改 (加速时间设置见附注2)
SET_POS	IN	输出的脉冲数, 分正负。正脉冲数表示沿 X 轴的正方向, 负脉冲数表示沿着 X 轴的负方向(此为绝对坐标)	DINT	-2147483648 ~ +2147483647	该参数在运行过程中可以修改, 当新设定值大于已输出的脉冲数, 那么最后输出的脉冲会以新设定值为准。当新设定值小于已输出脉冲数, 则会马上停止脉冲输出。
RUN	IN/OUT	运行使能位。 1: 有效 0: 无效	BOOL	0/1	3) 只有 RUN=1 与 E_STOP=0 时才能运行。 2、当运行完成后, RUN 内部复位。 3、当 E_STOP 为 1 时, RUN 内部复位。

STATUS	OUT	输出状态字节： <table border="1" style="margin-left: 20px;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> </table> <p>Bit0: 参数配置错误标志 1—参数配置错误 0—参数配置正常</p> <p>Bit1: 运行标志 1—正在运行, 该指令正在输出脉冲, 且指令未执行完。 0—不运行, 因公共资源被其他指令占用, 所以指令还没得以运行; 或者指令已经运行完毕</p> <p>Bit2: 完成标志 1—完成, 指令执行完毕 0—未完成, 指令未执行或指令正在执行中但未完成</p> <p>Bit3: 忙标志 1—忙标志有效, 该轴正在被其它指令占用 0—忙标志无效, 指令正在执行或此执行已完成</p> <p>Bit4: 模块急停标志 (见附注3) 1—模块急停标志有效, 该轴被外部条件禁止运行。 0—模块急停标志无效。</p> <p>Bit5~Bit6: 预留</p> <p>Bit7: 指令通信状态标志 1—通信超时 0—无超时</p>	7	6	5	4	3	2	1	0	BYTE	0~255	Bit0 : 1、只对轴参数进行判断; 2、MIN_SPEED/ MAX_SPEED/TA 等参数不作报错, 会自动设置成一个最接近的合理值。 3、253模块未被PLC识别或者是没有连接该模块时会提示参数配置错误。
		7	6	5	4	3	2	1	0				
		ACT_POS	OUT	当前的绝对坐标	DINT	-2147483648 ~ +2147483647							
ACT_SPEED	OUT	当前实际运行速度	DWORD	500~200 000	该值可能跟实际值会有一点偏差, 最大不超过 5K, 跟加速时间和设定的速度有关。								

回原指令

① 函数名: MC253_HOMING



② 功能：通过设置回原模式等参数，可寻找设备原点。

轴号与外部复位 IO 信号（如回原 Z 相信号）的对应关系：

轴 0 → IO.2（MC253_HSC0）

轴 1 → IO.6（MC253_HSC1）

若回原模式以原点开关为参考时（模式 3 或模式 4），必须将原点开关信号（指令的 HOMING_SW 参数）接至上述对应点，否则无法找到原点。

③ 参数

参数名	输入输出属性	参数描述	类型	数值范围	备注								
E_STOP	IN	紧急停止位。 1: 有效, 0: 无效	BOOL	0~1	1、只有 RUN =1 与 E_STOP =0 时才能运行。 2、当 E_STOP 为 1 时，RUN 内部复位。								
LIMIT_SW_CCW	IN	CCW 限位开关	BOOL	0~1									
LIMIT_SW-CW	IN	CW 限位开关	BOOL	0~1									
HOMING_SW	IN	原点开关	BOOL	0~1									
AXIS_NO	IN	轴号	BYTE	0~255	该参数在运行过程中不能修改。								
SIGNAL_TYPE	IN	<table border="1" style="margin-left: 20px;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> </table> 信号类型 Bit0: 逆时针限位开关信号类型 0—高电平 1—低电平	7	6	5	4	3	2	1	0	BYTE	0~255	
7	6	5	4	3	2	1	0						

		Bit1: 顺时针限位开关信号类型 0—高电平 1—低电平 Bit2: 原点开关信号类型 0—高电平 1—低电平			
HOMING_MODE	IN	回原模式	BYTE	1~14	参见附注 4
MIN_SPEED	IN	最小速度。 单位: HZ	DWORD	0~200000	1、最小速度的设定要小于最大速度; 2、该参数在运行过程中可以修改。 3、搜索速度不应太大, 接近速度应尽量小。
SEARCH_SPEED	IN	原点搜索速度。 单位: HZ	DWORD	0~200000	
APPROACH_SPEED	IN	原点接近速度。 单位: HZ	DWORD	0~200000	
TA	IN	加减速时间。单位: ms	DWORD	0~10000 (见附注 1)	该参数在运行过程中可以修改(加速时间设置见附注 2)
RUN	IN	运行使能位 1: 有效	BOOL	0~1	1、只有RUN =1与E_STOP =0时才能运行。 2、当运行完成后, RUN 内部复位。 3、当 E_STOP为1时, RUN 内部复位。

④ 说明

程序对各开关的检测以扫描方式实现, 故当开关量变化时处理不及时, 可能有些延迟。若回原速度(包括搜索速度和接近速度)太大时, 这个处理延迟被放大, 导致回原不准。

附注

1、当 $TA \neq 0$, 加速度 = $(MAX_SPEED - MIN_SPEED) / TA$ (若设置有最大加速度, 则受限于最大加速度);

若 $TA = 0$, 则采用指令 MC253_SET_MAX_ACCELE 设置的最大加速度, 若没有设置最大加速度, 则报参数故障。TD 亦然。对于双轴指令, 若两轴均设置了最大加速度, 则采用其中的较小值作为系统加速度。

2、理论上, 指令加速度 = $[(MAX_SPEED - MIN_SPEED) / TA]$, 如果由此所求的加速度过于小(小于 1), 则指令内部默认加速度为 1。用户可按预期加速度, 合理设置 TA/TD。

3、在模块输入端口引入了“模块急停信号”: 当模块检测到此信号时, 禁止脉冲输出, 并在脉冲输出指令(如 MC253_PTP_R、MC253_SPEED_CTL 和 MC253_PWM 指令)状态位处报警。轴号与急停信号的对应关系:

轴 0 → I0.3
轴 1 → I0.7

4、有关回原模式的详细说明请参见本文档附录 E.4 回原模式详解。

O.5 调试示例

混合运动控制指令使用

控制步进电机从 A 点到 B 点往返运动, 步进电机细分 1000, 丝杆导程 5mm, A 到 B 的位移 L 为 2000mm。



【系统说明】

本例中，在 CPU H226XL 后挂 1 个 SM253 模块，设置 SM253 模块第 0 轴做点到点运动的参数。主要调用 MC253_PTP_R 来设定控制参数。

I0.2 为 A 点硬件归零复位点（此点为行程开关量输入，设此点为机械原点）；

I1.0 为系统急停输入；

Q0.0 为脉冲输出，Q0.1 方向输出。

【程序块】

程序注释
应用CPU H226XL控制步进电机从A点到B点往返运动，步进电机细分1000，丝杆导程5mm，A到B的位移L为2000mm。

本例设置SM253-1BH32第0轴做点到点运动的参数，主要调用MC253_PTP_R来设定控制参数。
I0.2为A点硬件归零复位点（此点为行程开关输入量，设此点为机械原点）；
I1.0为系统急停输入；
Q0.0为脉冲输出，Q0.1为方向输出；

编程思路：
第一步，启动运行前，调用速度控制指令让机械先回零；
第二步，回零完成后，调用单轴相对运动指令做点到点运动；
第三步，通过改变MC_PTP_R指令中的SET_POS的值实现从A到B往返运动；

请参考章节 J.4 特别注意 1

网络 1 网络标题
使用MC253模块必须使用SM0.1调用MC253_INIT以初始化系统

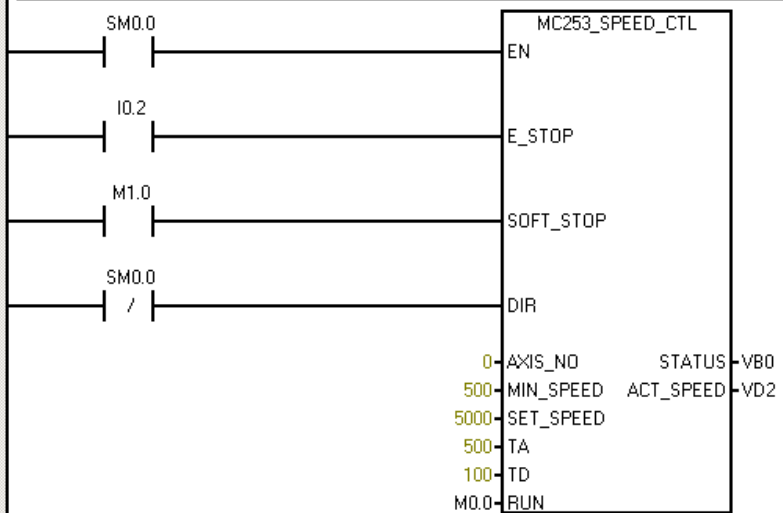
符号	地址	注释
MC253_INIT	FC4	

网络 2
初始化相应的位

网络 3 网络标题
使能外部复位坐标指令
功能：实现机械回零功能
M0.0 ----- 回零运行位（通过HMI等设置此位），上升沿设置外部复位使能；
M0.7 ----- 上升沿禁止外部复位使能；

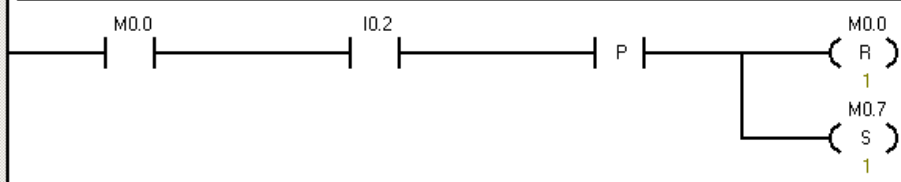
网络 4

速度控制指令
 功能：先控制0轴反向旋转，回到机械原点（即A点为运动的起始点）
 M0.0 ----- 运行使能位；
 I0.2 ----- 行程限位开关，作指令的急停开关，即回到零点时关闭速度控制指令；
 M1.0 ----- 软停止位；
 脉冲方向为反转，实现回零；
 轴号为0；
 启动/停止速度为500HZ；
 设定速度为5000HZ（加速时间到后的正常速度）；
 加速时间500ms；
 减速时间100ms；
 输出状态字放在VB0中；
 当前速度（频率）放在VD2中



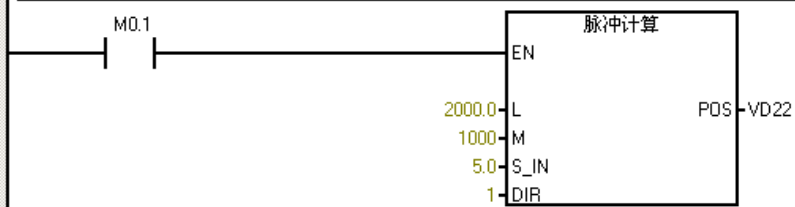
网络 5

归零完成后复位相应的位；



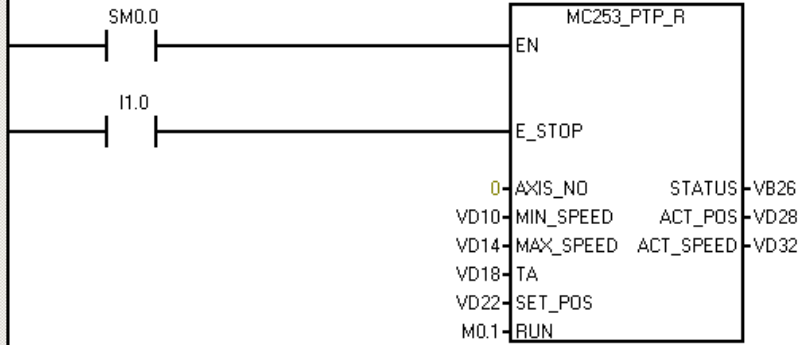
网络 6

M0.1----- A点到B点运动使能位
 A到B点位移2000.0mm；
 步进电机的细分1000；
 丝杆导程5.0mm；
 DIR 为1（1--正转、0--反转）
 脉冲数放在 VD22中



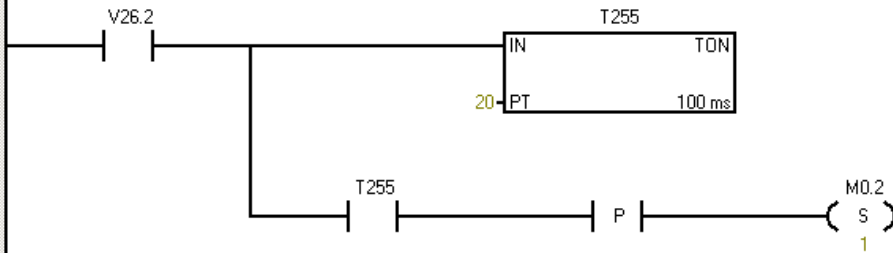
网络 7

单轴相对运动控制指令
 功能：实现A点到B点运动控制
 I1.0-----紧急停止位；
 轴号为0
 VD10-----启动/停止速度；
 VD14-----加速完成后正常运行速度；
 VD18-----加减速时间；
 VD22-----输出的脉冲数；
 M0.1-----运行使能位；
 输出状态字节-----VB26；
 已输出的脉冲数-----VD28；
 当前实际运行速度-----VD32；



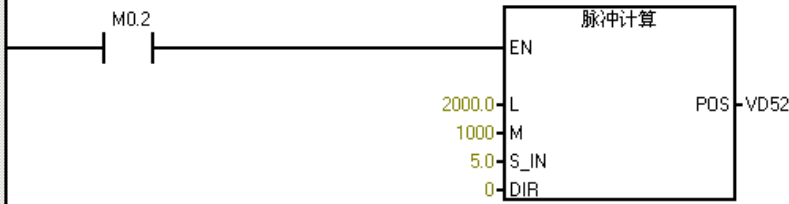
网络 8

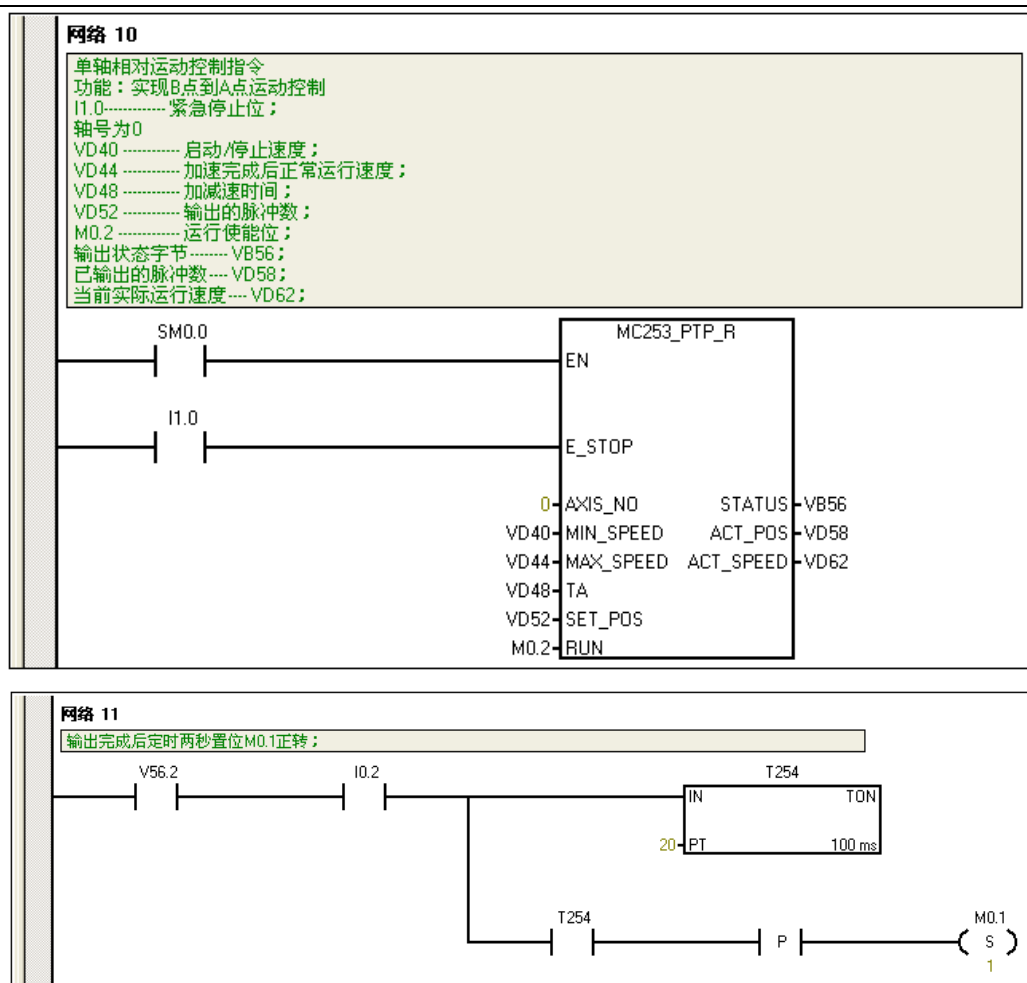
输出完成后定时两秒置位M0.2反转；



网络 9

M0.2-----B点到A点运动使能位
 B到A点位移2000.0mm；
 步进电机的细分1000；
 丝杆导程5.0mm；
 DIR 为0（1---正转、0---反转）
 脉冲数放在 VD52中





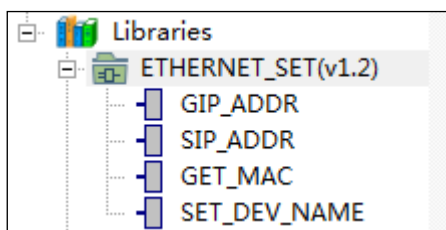
P 以太网设置指令库

P.1 功能介绍

以太网设置指令库 `ETHERNET_SET(V1.2)` 用于设置 PLC 以太网口本地通信参数，无需停止 CPU 即可设置、获取远程以太网 PLC 的 IP 地址、MAC 地址和设备名称。

P.2 指令详解

以太网设置指令库包含以下四条指令，相关库文件可从合信官网免费下载，网址：<http://www.co-trust.com> 该库所含的指令参数介绍如下：



获取 IP 地址库

- ① 库名称：GIP_ADDR

② 功能：获取 IP 地址

库函数	参数名称	输入输出属性	类型	说明
	EN	IN	BOOL	使能端，允许 SM0.0 调用
	STATUS	OUT	BYTE	状态字 bit0=1 表示获取成功 bit1=1 表示获取失败 其他位未使用
	IP_ADDR	OUT	DWORD	IP 地址，共四个字节，每个字节由低到高分别表示 IP 地址的对应四个数值。
	MASK	OUT	DWORD	子网掩码，共四个字节，每个字节由低到高分别表示子网掩码的对应四个数值。
	GATE	OUT	DWORD	网关，共四个字节，每个字节由低到高分别表示网关的对应四个数值。

设置 IP 地址库

① 库名称：SIP_ADDR

② 功能：设置 IP 地址；

库函数	参数名称	输入输出属性	类型	说明
	EN	IN	BOOL	使能端，用沿触发，设置指令不能循环调用，因为设置 IP 和 EPPROM 一样有次数限制，且不停写以太网将无法通讯
	IP_ADDR	IN	DWORD	P 地址，共四个字节，每个字节由低到高分别表示 IP 地址的对应四个数值。
	MASK	IN	DWORD	子网掩码，共四个字节，每个字节由低到高分别表示子网掩码的对应四个数值。
	GATE	IN	DWORD	网关，共四个字节，每个字节由低到高分别表示网关的对应四个数值。
	STATUS	OUT	BYTE	状态字 bit0=1 表示设置成功 bit1=1 表示非法 IP 地址 bit2=1 表示 IP 与掩码不匹配 Bit3=1 表示 IP 与网关不匹配 其他位未使用

获取 MAC 地址库

① 库名称：GET_MAC

② 功能：获取 MAC 地址；

库函数	参数名称	输入输出属性	类型	说明
	EN	IN	BOOL	使能端，允许 SM0.0 调用
	STATUS	OUT	BYTE	状态字 bit0=1 表示获取成功 其他位未使用

	MAC0~ MAC5	OUT	BYTE	MAC 地址 xx:xx:xx:xx:xx:xx MAC5-----MAC0
--	---------------	-----	------	--

设置设备名库

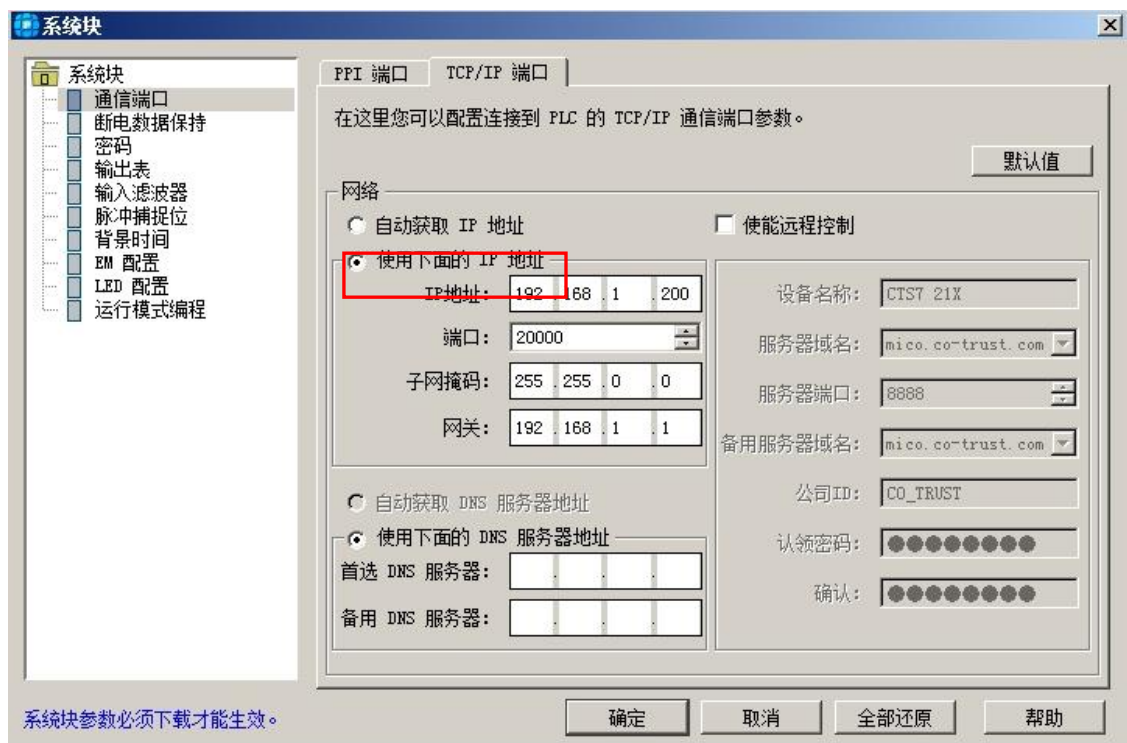
① 库名称: SET_DEV_NAME

② 功能: 设置设备名称;

库函数	参数名称	输入输出属性	类型	说明
	EN	IN	BOOL	使能端, 用沿触发, 设置指令不能循环调用, 因为设置 IP 和 EPPROM 一样有次数限制, 且不停写以太网将无法通讯
	NAME	IN	DWORD	设备名指针, 指向区域的第一个字节表示长度。整个字符串包含长度字节最大 32bytes。
	STATUS	OUT	BYTE	状态字 bit0 为 1 表示设置成功; bit1 为 1 表示长度错误; bit2 为 1 表示保存失败; bit3 表示非法字符

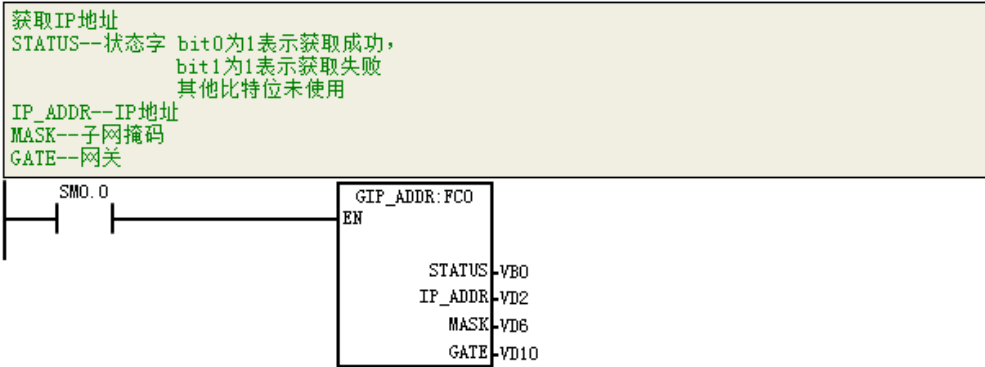
注意事项:

除了库, 也可以用 TD4S 设置 IP。若采用 TD4S 文本屏设置 IP, 请在系统块 TCP/IP 端口设置中选择使用静态 IP, 不能选择使用自动获取 IP 地址模式, 否则设置无法生效。而使用库设置的话就不需要此操作。

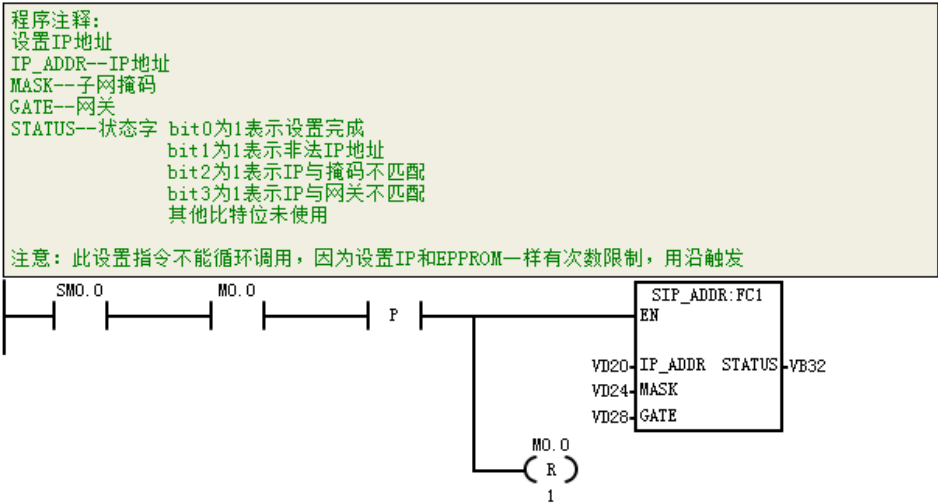


以太网指令设置库的应用示例程序如下所示：

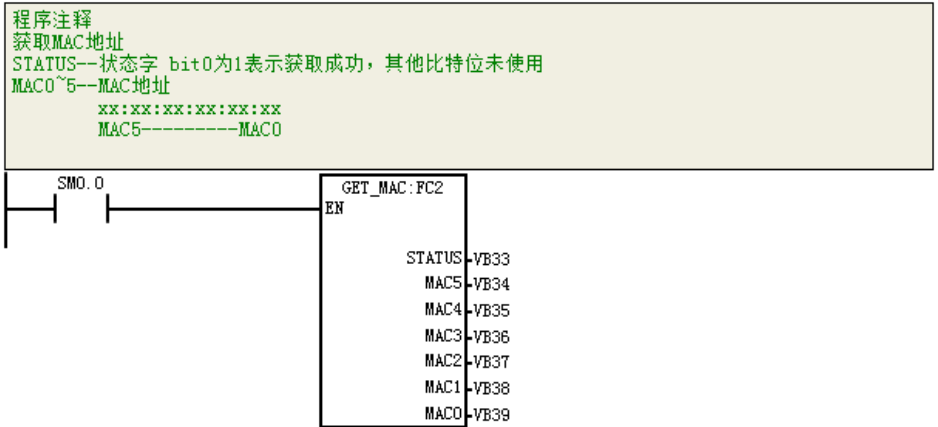
网络 1 网络标题



网络 2

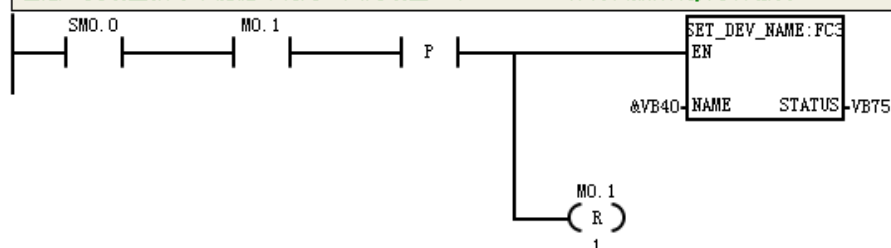


网络 3



网络 4

程序注释
 NAME 设备名指针，指向区域的第一个字节表示长度。整个字符串包含长度字节最大32bytes。
 STATUS bit0为1表示设置成功；bit1为1表示长度错误；bit2为1表示保存失败；bit3表示非法字符。
 注意：此设置指令不能循环调用，因为设置IP和EPPROM一样有次数限制，用沿触发



Q 以太网“ct_socket”通信库的使用

Q.1 功能介绍

H224, H226M 和 H226L 支持 ct_socket (v1.1) 库指令，高性能升级版 H224X, H226XM 和 H226XL 支持 ct_socket_lib_v1_2 (v1.2) 库指令。

备注：高性能升级版 H224X/H226XM/H226XL 的固件版本从 V2.29 或以上支持 SOCKET 功能，标准型 H224/H226M/H226L 的固件版本从 V1.29 或以上支持 SOCKET 功能。

目前 CPU 支持 2 个 UDP 连接和 2 个 TCP 客户端连接，提供的指令 SOCK_Open、SOCK_Send、SOCK_Recv、SOCK_Close，共 4 条。

约定项	范围	说明
连接号(SockID)	0~255	255 为无效连接号
连接类型 (SockType)	0, 1	0为UDP连接，1为TCP客户端连接
端口号 (Lport/Rport)	1~65535	本地端口号要选用不被其他连接占用的端口号，如系统块里默认端口号20000为PLC监控连接使用。TCP客户端的本地端口号由底层分配，Lport可设置为任何值（UDP模式下不能设为65535），Rport需与服务器端口号一致。

套接字错误码约定：

错误码	说明
0	无错误
1	连接已关闭
2	端口号错误。常用，本地端口号被占用，或端口号超出范围报错
3	设备断线
4	UDP 建立连接失败。常用，创建连接失败，返回连接号 255
5	TCP 客户端连接远方服务端失败
6	TCP 服务端监听失败
7	数据长度错误，最大数据长度为 512 字节
8	数据区为空错误
9	连接号错误
10	接收数据错误（没接收到数据）
11	连接类型错误

12	IP 信息错误
13	超时错误
14	TCP 客户端需要重连标志。常用，TCP 重连

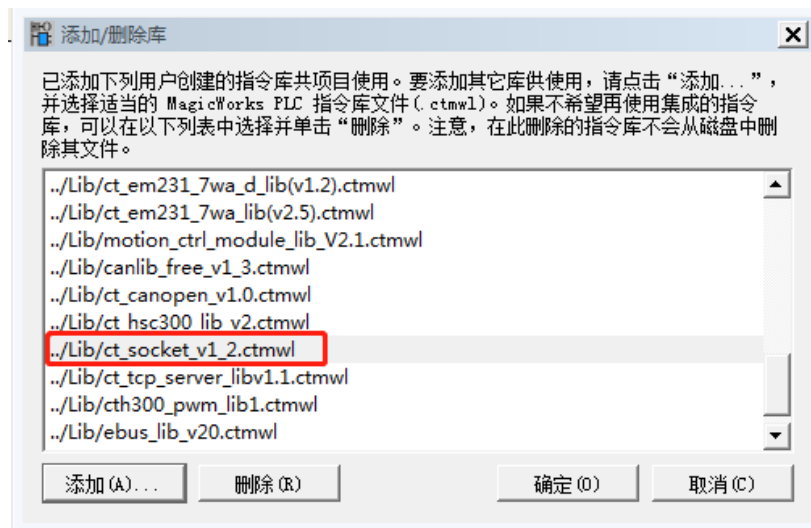
Q.2 使用说明

【添加库文件】

在项目管理器界面或编程界面选择“文件”--“添加/删除库”，如下图所示：



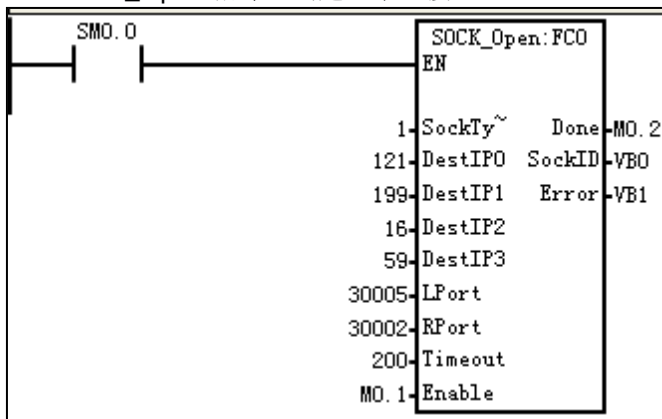
找到“ct_socket_v1_2.ctmwl”如下图所示，点击“确定”按钮。



安装成功后，在程序块界面目录树的“库”下可以看到新增加的“ct_socket (v1.2)”：

【指令功能说明】

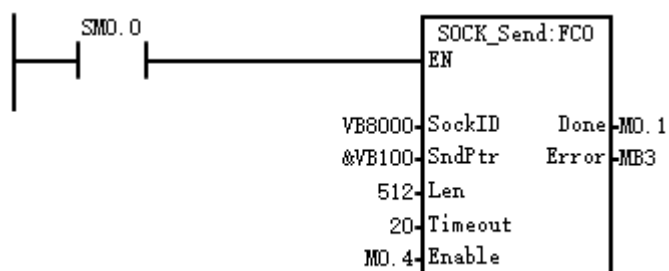
1) SOCK_Open 指令：创建一个连接



参数地址	说明	变量类型	数据类型	备注
------	----	------	------	----

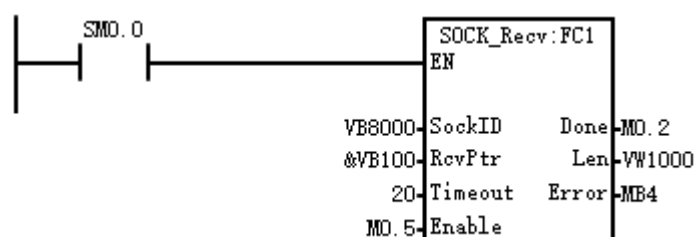
SockTy	链接类型, UDP(0)和 TCP-CLIENT(1)	IN	BYTE	不支持 TCP 服务器连接。
DestIP0	目标IP地址的第1个字节。	IN	BYTE	例如目标地址为192.168.1.202, 第一个字节为192。
DestIP1	目标IP地址的第2个字节。	IN	BYTE	例如目标地址为192.168.1.202, 第一个字节为168。
DestIP2	目标IP地址的第3个字节。	IN	BYTE	例如目标地址为192.168.1.202, 第一个字节为1。
DestIP3	目标IP地址的第4个字节。	IN	BYTE	例如目标地址为192.168.1.202, 第一个字节为202。
Lport	本地端口, UDP使用	IN	WORD	TCP-CLIENT模式, LPort由底层随机分配
Rport	目标端口	IN	WORD	与远程服务器端口号一致。
Timeout	超时时间	IN	BYTE	单位: 100ms。
Enable	使能位	I/O	BOOL	
Done	读写功能完成位	OUT	BOOL	使能后自动清零。
SockID	输出连接号	OUT	BYTE	底层分配, 应用要提供一个全局内存保存连接号。
Error	错误代码	OUT	BYTE	错误码。0: 无错

2) SOCK_Send 指令: 数据发送



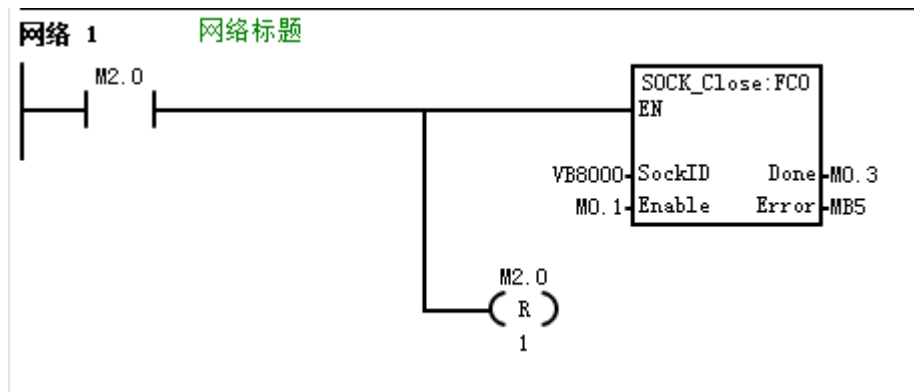
参数地址	说明	变量类型	数据类型	备注
SockID	输出连接号	OUT	BYTE	底层分配, 应用要提供一个全局内存保存连接号。
SndPtr	数据缓冲区	IN	DWORD	指向待发送数据的指针, 可以指向I、Q、M或V存储器的指针(如 &VB100)
Len	待发送字节数	IN	WORD	范围为1~512字节。
Timeout	超时时间	IN	BYTE	单位: 100ms。
Enable	使能位	I/O	BOOL	使能
Done	读写功能完成位	OUT	BOOL	使能后自动清零。
Error	错误代码	OUT	BYTE	错误码。0: 无错

3) SOCK_Recv 指令: 接受数据



参数地址	说明	变量类型	数据类型	备注
SockID	输出连接号	OUT	BYTE	底层分配，应用要提供一个全局内存保存连接号。
RcvPtr	数据缓冲区	IN	DWORD	指向接收数据存储位置的指针，可以指向I、Q、M或V存储器的指针（如&VB100）
Timeout	超时时间	IN	BYTE	单位：100ms。
Enable	使能位	I/O	BOOL	使能
Done	读写功能完成位	OUT	BOOL	使能后自动清零。
Len	实际接收字节数	IN	WORD	范围1~512字节。
Rport	目标端口	IN	WORD	与远程服务器端口号一致。
Error	错误代码	OUT	BYTE	错误码。0：无错

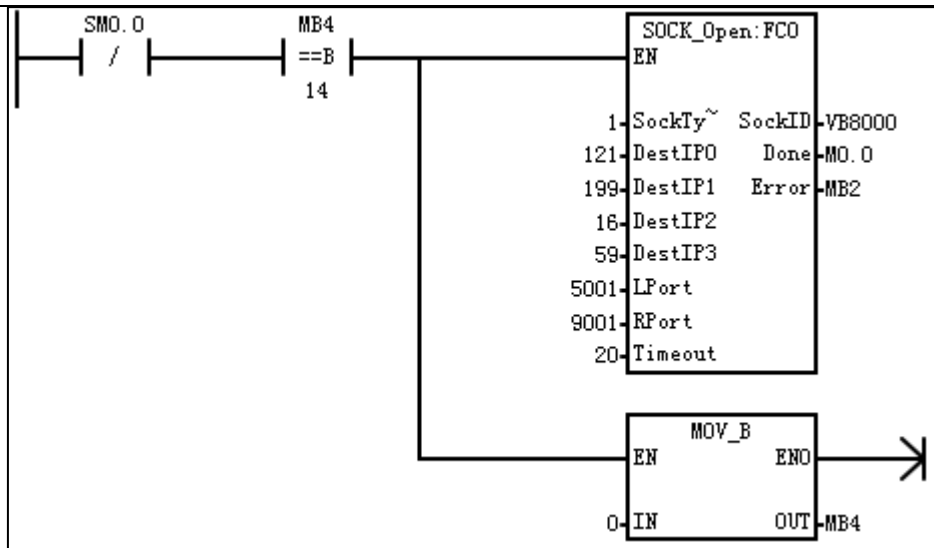
4) SOCK_Close 指令：关闭连接



参数地址	说明	变量类型	数据类型	备注
SockID	输出连接号	OUT	BYTE	底层分配，应用要提供一个全局内存保存连接号。
Done	读写功能完成位	OUT	BOOL	使能后自动清零。
Error	错误代码	OUT	BYTE	错误码。0：无错
Enable	使能位	I/O	BOOL	使能

【应用实例】

TCP 客户端掉线重连，通过发送出错标志来确定是否重新打开连接：重连标志 14



R CANfree 通信指令库介绍

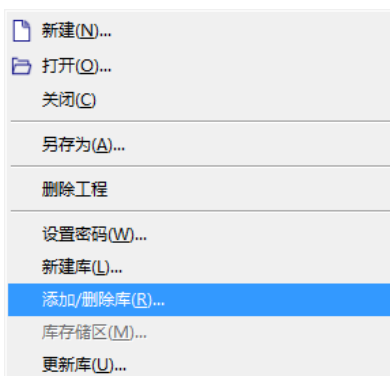
R.1 功能介绍

CTH200 高性能系列 CPU 可通过调用 CANfree 通信指令库来进行 CANfree 通信，完成 CANfree 数据收发和滤波，CANfree 是在 CAN 通信基础协议上完成了一次封装，方便用户更加简单直接地使用，本节主要介绍该指令库的安装和应用。

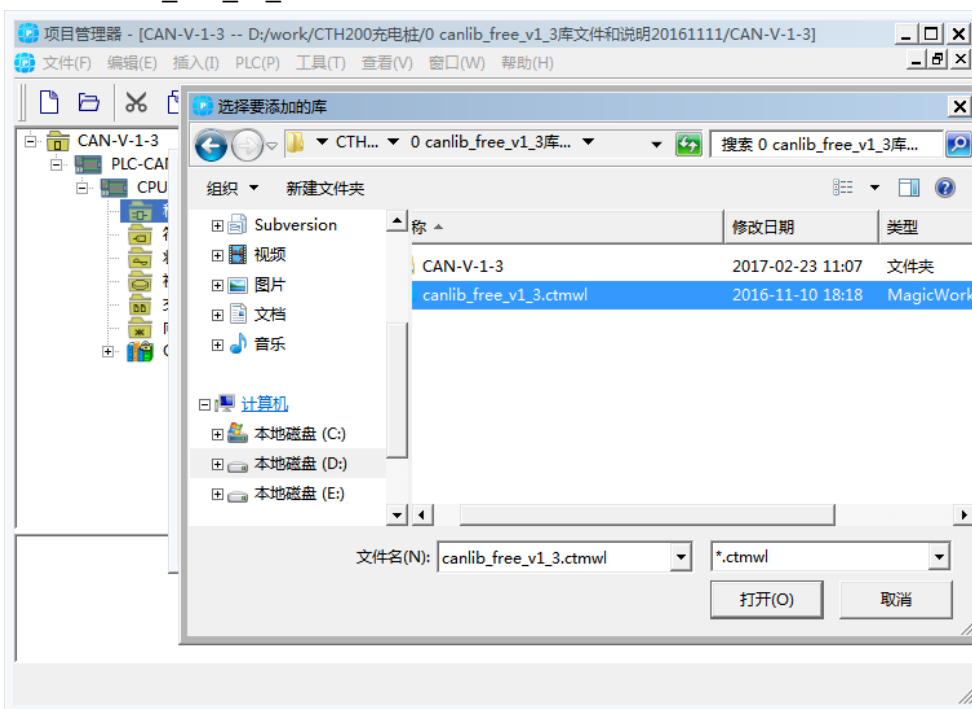
R.2 添加库文件

MagicWorks PLC 中添加库

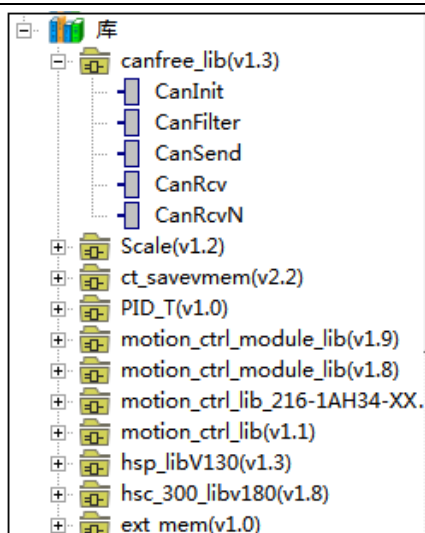
打开 MagicWorks PLC (V2.10 及更高版本)，选择“文件”--“添加/删除库”，在弹出的窗口单击“添加”：



在存放“canlib_free_v1_3.ctmwl”文件的位置选中文件，如下图所示，单击“打开”并确认添加。



安装成功后，在目录树的“库”下可以看到新增加的 canfree_lib(v1.3)。

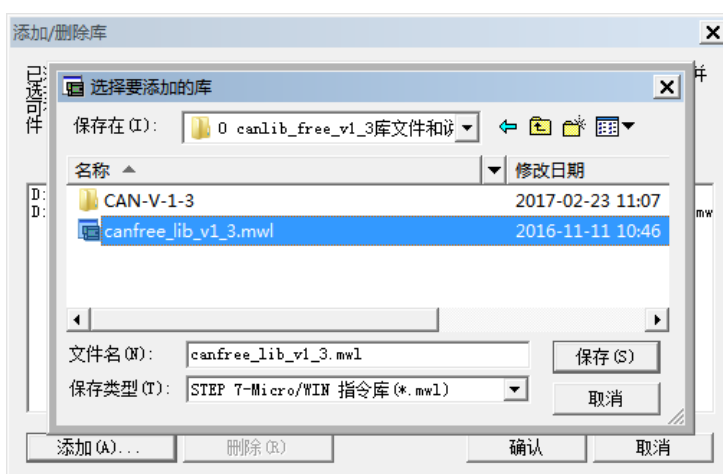


MicroWIN 中添加库

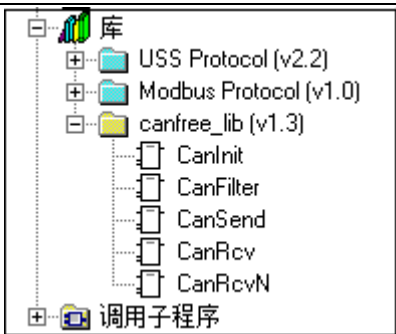
打开 STEP 7-MicroWIN，在“文件”--“添加/删除库”，找到库文件“canlib_free_v1_3.mwl”，如下图所示：



在存放“canlib_free_v1_3.mwl”文件的位置找到文件，如下图所示，单击“添加”按钮。



安装成功后，在目录树的“库”下可以看到新增加的 canfree_lib(v1.3)。



R.3 库指令说明

CAN 通信指令库共有五条指令，分别是：

- CANInt（CAN 初始化指令）
- CANSend（CAN 发送数据指令）
- CANRcv（CAN 接收数据指令）
- CANFilter（CAN 数据过滤指令）
- CanRcvN（接收多帧数据）

CAN 初始化

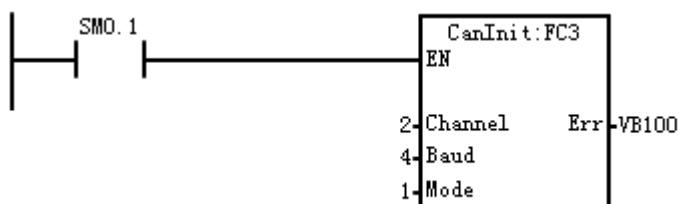
- ① 指令名称：CANInt
- ② 功能：初始化 CAN 自由口；

表 S-1 CAN 初始化指令

库函数	参数名称	输入输出属性	类型	数值范围	说明
	EN	IN	BOOL	--	--
	Channel	IN	BYTE	0-2	CAN 通道选择 0: 本地 CAN0 通道 1: 本地 CAN1 通道 2: CAN 扩展板通道
	Baud	IN	BYTE	0-7	波特率 0(1MHz); 1 (800kHz); 2 (500kHz); 3 (250kHz); 4 (125kHz); 5 (50kHz); 6 (20kHz); 7 (10kHz)
	Mode	IN	BYTE	0-1	帧格式 0: CAN_A 标准帧 1: CAN_B 扩展帧
	Err	OUT	BYTE	0-4	错误码，参见表 S-4。0-无错，4-参数错误

③ 示例

初始化 CAN 口参数为：CAN 小板通道，125KHz 波特率，帧格式为扩展帧格式



CAN 发送数据

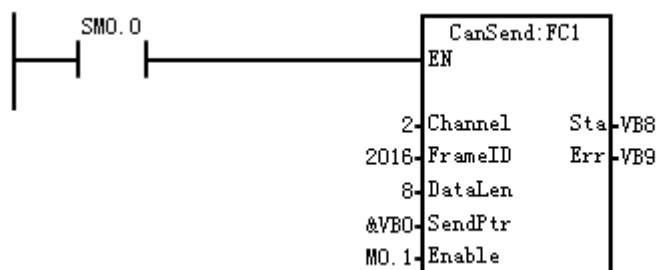
- ① 指令名称：CANSend;
- ② 功能：CAN 发送数据；启用该指令，Enable 置 1 则发送一次

表 S-2 CAN 发送数据指令

库函数	参数名称	输入输出属性	类型	数值范围	说明
	EN	IN	BOOL	--	--
	Channel	IN	BYTE	0-2	CAN 通道选择 0: 本地 CAN0 通道 1: 本地 CAN1 通道 2: CAN 扩展板通道
	FrameID	IN	Dword	--	发送数据的帧 ID
	DataLen	IN	BYTE	--	发送数据的帧长度
	SendPtr	IN	Dword	--	发送数据缓冲区指针
	Enable	IN	BOOL	--	使能位
	Sta	OUT	BYTE	0-1	发送状态（0-未完成，1-完成） 发送完成置 1，自动清 0
Err	OUT	BYTE	0-4	错误码（参见表 S-4） 本地 CAN0/CAN1 通道：0-无错； 1-发送超时；4-参数错误 CAN 扩展板通道：0-无错，2- 发送缓冲区已满	

③ 示例

通过 CAN 扩展板通道发送 8 字节数据，帧 ID 为 2016（通过 SM0.0 调用，通过使能位 Enable 触发）



CAN 接收数据

- ① 指令名称：CANRcv;
- ② 功能：CAN 接收数据；收到数据后会把数据填到接收缓冲区，接收指令是查询接收缓冲区有没有数据。Enable 置 1 一次，就查一下接收缓冲区。

表 S-3 CAN 接收数据指令

库函数	参数名称	输入输出属性	类型	数值范围	说明
<div style="border: 1px solid black; padding: 5px; width: fit-content;"> CanRcv:FC2 EN Channel DataLen Enable FrameID RcvPtr Sta Err </div>	EN	IN	BOOL	--	--
	Channel	IN	BYTE	0-2	CAN 通道选择 0: 本地 CAN0 通道 1: 本地 CAN1 通道 2: CAN 扩展板通道
	Enable	IN	BOOL		使能位
	RcvPtr	IN	Dword	--	接收数据缓冲区指针
	FrameID	OUT	Dword	--	接收到数据的帧 ID
	DataLen	OUT	BYTE	--	接收到数据的帧长度
	Sta	OUT	BYTE	0-1	接收状态: 0-接收未完成 1-接收完成
Err	OUT	BYTE	0-4	错误码 (参见表 S-4) 0-无错; 3-接收缓冲区已空	

③ 示例

通过 CAN 扩展板通道接收指针所指存储器中的数据, 存放帧 ID 为 2016 (通过 SM0.0 调用, 通过使能位 Enable 触发)

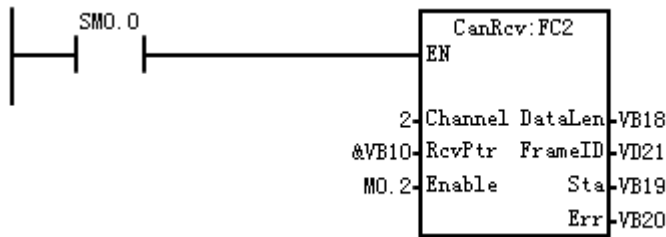


表 S-4 Err 错误码定义

错误编码	定义
0	无错误
1	发送超时
2	缓冲区已满 (发送)
3	缓冲区已空 (接收)
4	参数错误

CAN 接收数据报文过滤

① 指令名称: CANFilter;

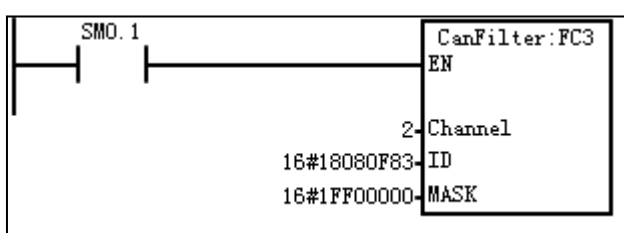
② 功能: CAN 接收数据报文过滤; 如果只接收特定帧 ID 的 CAN 报文, 可以调用过滤指令。如果程序不调用过滤指令, 默认过滤 ID 和掩码都是 0, 即接收任何 CAN 报文。

表 S-5 CAN 数据过滤指令

库函数	参数名称	输入输出属性	类型	数值范围	说明
<pre> CanFilter:FC3 -EN -Channel -ID -MASK </pre>	EN	IN	BOOL	--	--
	Channel	IN	BYTE	0-2	CAN 通道选择 0: 本地 CAN0 通道 1: 本地 CAN1 通道 2: CAN 扩展板通道
	ID	IN	BOOL		要接收的数据报文 ID
	MASK	IN	DWORD	--	对 ID 进行掩码控制

③ 示例

通过 CAN 扩展板接收帧 ID 为 0x180xxxxx 的报文



接收多帧数据

① 指令名称: CanRcvN;

② 功能: 读取若干帧 CAN 数据, 输出实际数量帧 CAN 数据, 顺序存放到缓冲区内存, 13 字节偏移一帧。

表 S-6 CAN 接收多帧数据

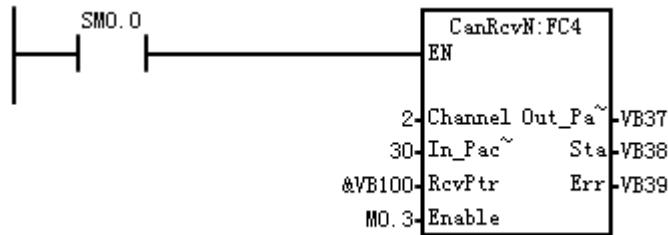
库函数	参数名称	输入输出属性	类型	数值范围	说明
<pre> CanRcvN:FC3 -EN -Channel Out_Pa~ -In_Pac~ Sta -RcvPtr Err -Enable </pre>	EN	IN	BOOL	--	--
	Channel	IN	BYTE	0-2	CAN 通道选择 0: 本地 CAN0 通道 1: 本地 CAN1 通道 2: CAN 扩展板通道
	In_packets	IN	BYTE		
	RcvPtr	IN	DWORD		接收数据内存地址
	Enable	IN_OUT	BOOL	--	使能位
	Out_packets	OUT	BYTE		实际接收到的帧数
	Sta	OUT	BYTE	0-1	接收状态: 0 - 未完成; 1 - 完成
Err	OUT	BYTE	0-4	错误码(0 无错误, 3 缓冲区已空, 4 参数错误)	

CAN 数据结构 (共 13 字节)

帧 ID	帧长 len<=8	帧数据 Data
4bytes	1byte	8bytes

③ 示例

通过 CAN0 口一次接收 30 帧数据存储在 VB100 中，将实际的输出帧数存放到缓冲区地址 VB37 中。



S 指令集

S.1 指令速查

表 T-1 CTH200 指令速查

位逻辑指令			
LD	常开触点载入指令 LD SM0.0	ALD	与载入指令 ALD
A	常开触点与指令 A SM0.0	OLD	或载入指令 OLD
O	常开触点或指令 O SM0.0	LPS	逻辑进栈指令 LPS
LDN	常闭触点载入指令 LDN SM0.0	LDS	载入堆栈指令 LDS 1
AN	常闭触点与指令 AN SM0.0	LRD	逻辑读取指令 LRD
ON	常闭触点或指令 ON SM0.0	LPP	逻辑出栈指令 LPP
LDI	常开触点立即载入指令 LDI I0.0	=	输出指令 = Q0.0
AI	常开触点与立即指令 AI I0.0	=I	立即输出指令 =I Q0.0
OI	常开触点或立即指令 OI I0.0	S	置位指令 S Q0.0 1
LDNI	常闭触点立即载入指令 LDNI I0.0	SI	立即置位指令 SI Q0.0 1
ANI	常闭触点与立即指令 ANI I0.0	R	重设指令 R Q0.0 1
ONI	常闭触点或立即指令 ONI I0.0	RI	立即重设指令 RI Q0.0 1
NOT	逻辑栈顶取反指令 NOT	AENO	功率流与指令 AENO
EU	上升沿检测指令	NOP	空操作指令

	EU		NOP 1
ED	下降沿检测指令 ED	--	--
比较指令			
LDBx	字节比较 (=,<,>,<=,>=,<>) 载入指令 LDB= 1, VB0	OBx	字节操作数比较 (=,<,>,<=,>=,<>) 或指令 OB= 1, VB0
LDWx	整数比较 (=,<,>,<=,>=,<>) 载入指令 LDW= 10000, VW0	OWx	整数比较 (=,<,>,<=,>=,<>) 或指令 OW= 10000, VW0
LDDx	长整数比较 (=,<,>,<=,>=,<>) 载入指令 LDD= 100000, VD0	ODx	双整数比较 (=,<,>,<=,>=,<>) 或指令 OD= 100000, VD0
LDRx	浮点数比较 (=,<,>,<=,>=,<>) 载入指令 LDR= 1.0, VD0	ORx	浮点数比较 (=,<,>,<=,>=,<>) 或指令 OR= 1.0, VD0
ABx	字节操作数比较 (=,<,>,<=,>=,<>) 与指令 AB= 1, VB0	LDSx	字符串比较 (=,<>) 载入指令 LDS="1234567890",VB0
AWx	整数比较 (=,<,>,<=,>=,<>) 与指令 AW= 10000, VW0	ASx	字符串比较 (=,<>) 与指令 AS="1234567890", VB0
ADx	双整数比较 (=,<,>,<=,>=,<>) 与指令 AD= 100000, VD0	OSx	字符串比较 (=,<>) 或指令 OS="1234567890", VB0
ARx	浮点数比较 (=,<,>,<=,>=,<>) 与指令 AR= 1.0, VD0	--	--
传送指令			
MOVB	字节移动指令 MOVB 1 VB0	SWAP	高低字节交换指令 SWAP VW0
MOVW	字移动指令 MOVW 1000, VD0	BIR	移动字节立即读指令 BIR IB0, VB0
MOVD	双字移动指令 MOVD 100000, VD0	BIW	移动字节立即写指令 BIW VB0, QB0
MOVR	浮点数移动指令 MOVR 1.0, VD0	SRCP	存储配方到存储卡
BMB	块移动字节指令 BMB VB0, VB100, 1	LRCP	从存储卡装载配方
BMW	块移动字指令 BMW VW0, VW100, 1	SDL	存储数据记录到存储卡
BMD	块移动双字指令 BMD VD0, VD100, 1	--	--
整数计算指令			
+I	整数加法指令	MUL	整数与双整数相乘

	+I 10000, VW0		MUL 10000, VD0
-I	整数减法指令 -I 10000, VW0	DIV	整数与双整数相除 DIV VW0, VD0
*I	整数乘法指令 *I 10000, VW0	INCB	字节递增指令 INCB VB0
/I	整数除法指令 /I 10000, VW0	DECB	字节递减指令 DECB VB0
+D	双整数加法指令 +D 100000, VD0	INCW	整数递增指令 INCW VW0
-D	双整数减法指令 -D 100000, VD0	DECW	整数递减指令 DECW VW0
*D	双整数乘法指令 *D 100000, VD0	INCD	长整数递增指令 INCD VD0
/D	双整数除法指令 /D 100000, VD0	DECD	长整数递减指令 DECD VD0
浮点数运算指令			
+R	实数加法指令 +R 1.0, VD0	COS	余弦运算指令 COS 1.0, VD0
-R	实数加法指令 -R 1.0, VD0	TAN	正切运算指令 TAN 1.0, VD0
*R	实数乘法指令 *R 1.0, VD0	LN	自然对数运算指令 LN 1.0, VD0
/R	实数除法指令 /R 1.0, VD0	EXP	自然指数运算指令 EXP 1.0, VD0
SQRT	平方根指令	PID	PID 环路运算指令 PID VB0, 1
SIN	正弦运算指令 SIN 1.0, VD0	--	--
转换指令			
BTI	字节至整数转换 BTI 1, VW0	DTR	双整数至实数转换 DTR 100000, VD0
ITB	整数至字节转换 ITB 10000, VB0	DTS	双整数至字符串 DTS 100000, VB0, 10
ITD	整数至双整数转换 ITD 10000, VD0	ROUND	进位取整指令 ROUND 1.0, VD0
ITS	整数转换为字符串 ITS 10000, VB0, 10	TRUNC	截位取整指令 TRUNC 1.0, VD0
DTI	双整数至整数转换 DTI 100000, VW0	RTS	实数至字符串转换 RTS 1.0, VB0, 10
BCDI	BCD 至整数转换 BCDI VW0	STI	字符串至整数转换 STI"1234567890", 5, VW0
IBCD	整数至 BCD 转换 IBCD VW0	STD	字符串至双整数转换 STD"1234567890", 5, VD0
ITA	整数至 ASCII 转换 ITA 10000, VB0, 10	STR	字符串至实数转换 STR"1234567890", 5, VD0

DTA	双整数至 ASCII 转换 DTA 100000, VB0, 10	DECO	解码指令 DECO 1, VW0
RTA	实数至 ASCII 码转换指令	ENCO	编码指令 ENCO 10000, VB0
ATH	ASCII 至 16 进制转换 ATH VB0, VB100, 10	SEG	段指令 SEG 1, VB0
HTA	16 进制至 ASCII 转换 HTA VB0, VB100, 10	--	--
实时时钟			
TODR	读取实时时钟 TODR VB0	TODRX	读取扩展的实时时钟 TODRX VB0
TODW	设置实时时钟 TODW VB0	TODWX	设置扩展的实时时钟 TODWX VB0
移位循环指令			
SLB	向左移位字节 SLB VB0, 4	RLW	向左旋转字 RLW VW0, 8
SLW	向左移位字 SLW VW0, 8	RLD	向左旋转双字 RLD VD0, 16
SLD	向左移位双字 SLD VD0, 16	RRB	向右旋转字节 RRB VB0, 4
SRB	向右移位字节 SRB VB0, 4	RRW	向右旋转字 RRW VW0, 8
SRW	向右移位字	RRD	向右旋转双字 RRD VD0, 16
SRD	向右移位双字 SRD VD0, 16	SHRB	移位寄存器位指令 SHRB I0.0, V0.0, 8
RLB	向左旋转字节 RLB VB0, 4	--	--
逻辑计算指令			
INVB	反转字节指令 INVB VB0	ORB	或字节指令 ORB 1, VB0
INVW	反转字指令 INVW VW0	ORW	或字指令 ORW 10000, VW0
INVD	反转双字指令 INVD VD0	ORD	或双字指令 OD 100000, VD0
ANDB	与字节指令 ANDB 1, VB0	XORB	异或字节指令 XORB 1, VB0
ANDW	与字指令 ANDW 10000, VW0	XORW	异或字指令 XORW 10000, VW0
ANDD	与双字指令 ANDD 100000, VD0	XORD	异或双字指令 XORD 100000, VD0
计数器指令			
CTU	向上计数指令 CTU C1, 10000	HDEF	高速计数器定义 HDEF 0, 0
CTD	向下计数指令	HSC	高速计数器指令

	CTD C1, 10000		
CTUD	向上/向下计数指令 CTUD C1, 10000	PLS	脉冲输出指令
定时器指令			
TON	打开延时计时器 TON T37, 10000	BITIM	读取开始时间计时器 BITIM VD0
TONR	保留性打开延时计时器 TONR T31, 10000	R_UTIM	读取当前微秒值指令
TOF	关闭延时计时器 TOF T37, 10000	CITIM	计算间隔时间计时器 CITIM VD0, VD100
字符串指令			
SLEN	取字符串长度指令 SLEN"1234567890", VB0	SCAT	并置字符串指令 SCAT"1234567890", VB0
SCPY	复制字符串指令 SCPY"1234567890", VB0	SFND	在字符串中查找字符串指令 SFND"12345678890", "321", VB0
SSCPY	从字符串复制子字符串指令 SSCPY"1234567890", 1, 10, VB0	CFND	在字符串中查找一个字符指令 CFND"12345678890", "a", VB0
表指令			
FILL	内存填充指令 FILL 10000, VW0, 10	FIFO	先入先出指令 FIFO VW200, VW400
ATT	增加至表格指令 ATT 10000, VW0	LIFO	后入先出指令 LIFO VW200, VW400
FNDx	表格查找 (=,<,>,<=,>=,<>) 指令 FND=VW0, 9999, VW1000	--	--
中断指令			
CRETI	从中断程序有条件返回指令	ATCH	附加中断指令
ENI	启用中断指令 ENI	DTCH	分离中断指令
DISI	禁用中断指令 DISI	CEVNT	清除中断事件指令 CEVNT 10
程序控制指令			
FOR	FOR-NEXT 循环开始 FOR-NEXT FOR VW0, 1, 10 NEXT	LSCR	载入顺序控制中继
NEXT	FOR-NEXT 循环结束	SCRE	顺序控制中结束
JMP	跳转至标签指令	SCRT	顺序控制中继转换
LBL	标签指令	CSCRE	有条件顺序控制中结束
WDR	监视器重设指令	CRET	从子程序有条件返回指令
DLED	诊断 LED 指令 DLED 1	END	有条件结束
CALL	调用子例行程序指令	STOP	停机指令

通信指令			
XMT	自由口发送指令	SPA	设置端口指令
RCV	自由口接收指令	GPA	获得端口地址指令
NETR	网络读取指令	EBUSR	读取模块信息指令
NETW	网络写入指令	EBUSW	写入模块信息指令
UDPNETR	以太网网络读功能	EBUSGETDIA	获取模块内部诊断信息指令
UDPNETW	以太网网络写入功能	EBUSSNDCMD	发送模块操作命令指令
MBSNDMSG	发送 Modbus 消息	--	--

S.2 指令详解

位逻辑指令

标准触点指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表:		
LD	bit	
A	bit	
O	bit	
LDN	bit	
AN	bit	
ON	bit	
梯形图:		
输入输出参数	数据类型	适用操作数
位	BOOL	I, Q, M, SM, T, C, V, S, L

操作数说明: 如果参数 bit 数据类型为 I 或 Q, 标准触点指令从内存或过程映象寄存器读取引用值。

功能说明:

当参数位变量 bit 为 1 时, 常开触点 (LD、A、O) 接通; 当参数位变量 bit 为 0 时, 常闭触点 (LDN、AN、ON) 接通。

在 STL 中, 常开触点用载入 (LD)、与 (A)、或 (O) 指令表示, 常开指令将地址位数值置于栈顶。

常闭触点由负装载 (LDN)、与非 (AN)、或非 (ON) 指令表示, 常闭指令将地址位数值的逻辑非置于栈顶。

在 LAD 中, 常开和常闭指令用触点表示。

示例程序:

STL:

ORGANIZATION_BLOCK 主程序: OB1

Network 1

```

LD      I0.0
A       I0.1
O       I0.2
=       Q0.0 //如果 I0.0 与 I0.1 同时为 1，或者 I0.2 为 1，则 Q0.0 点亮
    
```

Network 2

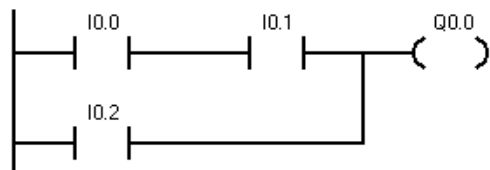
```

LDN     I1.0
AN      I1.1
ON      I1.2
=       Q1.0 //如果 I1.0 与 I1.1 同时为 0，或者 I1.2 为 0，则 Q1.0 点亮
    
```

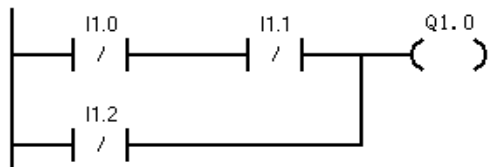
梯形图:

ORGANIZATION_BLOCK 主程序: OB1

Network 1



Network 2



立即触点指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表:		
LDI	bit	
AI	bit	
OI	bit	
LDNI	bit	
ANI	bit	
ONI	bit	
梯形图:		
输入输出参数	数据类型	适用操作数
位	BOOL	I

功能说明:

当实际输入点为 1 时，立即常开触点（LDI、AI、OI）接通。

当实际输入点为 0 时，立即常闭触点（LDNI、ANI、ONI）接通。


立即指令执行时获取实际输入值，不更新 CPU 进程映像寄存器。立即触点的参数值会立即更新，而不依赖于 CPU 的扫描周期。

在 STL 中，立即常开触点用立即载入（LDI）、立即与（AI）和立即或（OI）指令表示，立即常开指令立即将实际输入值装载、与、或至栈顶。

立即常闭触点用立即载入非（LDNI）、立即与非（ANI）和立即或非（ONI）指令表示，立即常闭

指令立即将实际输入点数值的逻辑非装载、与、或至栈顶。

在 LAD 中，立即常开和立即常闭指令用触点表示。

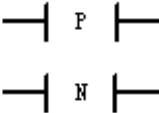
取反指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表: NOT		
梯形图: 		
输入输出参数	数据类型	适用操作数
无	无	无

功能说明:

取反 (NOT) 触点将输入的使能位状态取反。当使能位为 1，取反 (NOT) 触点断开，当使能位为 0，取反 (NOT) 触点接通。

在 STL 中，取反 (NOT) 指令将堆栈顶部的数值从 0 改变为 1，或从 1 改变为 0。

在 LAD 中，取反 (NOT) 指令用触点表示。

边沿检测指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表: EU ED		
梯形图: 		
输入输出参数	数据类型	适用操作数
无	无	无

功能说明:

上升沿检测 (EU) 指令比较本次扫描与上次扫描的输入能流的变化，如能流有上升沿变化时 (OFF→ON)，将能流置为 ON 状态一个扫描间隔；其他情况下将能流置为 OFF 状态。

下降沿检测 (ED) 指令比较本次扫描与上次扫描的输入能流的变化，如能流有下降沿变化时 (ON→OFF)，将能流置为 ON 状态一个扫描间隔；其他情况下将能流置为 OFF 状态。

在 STL 中，上升沿检测指令一旦在堆栈顶部数值中检测到 0 至 1 转换时，则将堆栈顶值设为 1；否则，将其设为 0。下降沿检测指令一旦在堆栈顶部数值中检测到 1 至 0 转换时，则将堆栈顶值设为 1；否则，将其设为 0。对于运行时编辑操作，您必须为边沿检测指令输入一个参数。有关详情，请参阅在运行模式中进行程序编辑。在 LAD 中，边沿检测指令用触点表示。



提示

因为边沿检测指令要求执行“打开至关闭”或“关闭至打开”转换，所以无法在首次扫描时检测上升沿或下降沿。在首次扫描中，CTH200 设置由这些指令指定的位状态。在其后的扫描中，这些指令才能检测到指定位的转换。

逻辑堆栈指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表: ALD OLD LPS LDS n LRD LPP		
梯形图: 无		
输入输出参数	数据类型	适用操作数
LDS 指令操作数 n	无	常数 (0~8)

功能说明:

与载入指令 (ALD) 采用逻辑与操作将堆栈第一级和第二级中的数值组合, 并将结果载入堆栈顶部。执行 ALD 后, 堆栈深度减 1。

或载入指令 (OLD) 采用逻辑或操作将堆栈第一级和第二级中的数值组合, 并将结果载入堆栈顶部。执行 OLD 后, 堆栈深度减 1。

逻辑进栈指令 (LPS) 复制堆栈中的顶值并使该数值进栈。堆栈底值被推出栈并丢失。

逻辑出栈指令 (LPP) 将堆栈中的一个数值出栈。第二个堆栈数值成为堆栈新顶值。

逻辑读取指令 (LRD) 将第二个堆栈数值复制至堆栈顶部。不执行进栈或出栈, 但旧堆栈顶值被复制破坏。

载入堆栈指令 (LDS) 复制堆栈中的堆栈位 n, 并将该数值置于堆栈顶部。堆栈底值被推出并丢失。

输出指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表: = bit		
梯形图: 		
输入输出参数	数据类型	适用操作数
输出位 bit	BOOL	I, Q, M, SM, T, C, V, S, L
使能位 (LAD)	BOOL	使能位


功能说明:

输出指令 (=) 将输出位的新数值写入过程映像寄存器。

在 STL, 位于堆栈顶端的数值被复制至指定的位。

在 LAD 中, 当输出指令被执行时, CPU 将过程映像寄存器中的输出位打开或关闭, 指定的位被设为等于使能位。

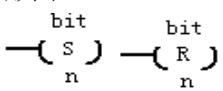
立即输出指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表: =I bit		
梯形图:		

bit 		
输入输出参数	数据类型	适用操作数
输出位 bit	BOOL	Q
使能位 (LAD)	BOOL	使能位

功能说明:

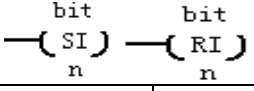
执行指令时，立即输出指令 (=I) 将新值写入实际输出和对应的过程映像寄存器位置。执行“立即输出”指令时，实际输出点的值被立即设为等于使能位。“I”表示立即输出；执行指令时，新值被写入实际输出和对应的过程映像寄存器位置。这与非立即输出不同，非立即输出仅将新值写入过程映像寄存器而不马上更新实际输出。

对于 STL，立即输出指令将位于堆栈顶端的数值复制至指定的实际输出位 bit。

置位复位指令	设置 ENO = 0 的错误条件	0006 (间接地址)、0091 (操作数超出范围)
	影响的特殊内存位	无
指令列表: S bit, n R bit, n		
梯形图: 		
输入输出参数	数据类型	适用操作数
位 bit	BOOL	I, Q, M, SM, T, C, V, S, L
位数 n	字节	VB, IB, QB, MB, SMB, SB, LB, AC, 常数, *VD, *AC, *LD

功能说明:

当能流有效时，置位打开或复原从指定地址 bit 开始的 n 个点，最多可以置位和复位 255 个点。如果“复位”指令指定一个定时器位 (T) 或计数器位 (C)，指令将复位定时器或计数器位，并清除定时器或计数器的当前值。

立即置位复位指令	设置 ENO = 0 的错误条件	0006 (间接地址)、0091 (操作数超出范围)
	影响的特殊内存位	无
指令列表: SI bit, n RI bit, n		
梯形图: 		
输入输出参数	数据类型	适用操作数
位 bit	BOOL	Q
位数 n	字节	VB, IB, QB, MB, SMB, SB, LB, AC, 常数, *VD, *AC, *LD

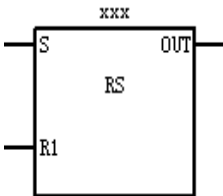
功能说明: 能流有效时，立即置位指令 (SI) 和立即复位指令 (RI) 将立即置位打开或复位清零从指定地址 bit 开始的 n 个点，可以置位或复位 1 至 128 个点。执行立即置位复位指令时，新值被写入实际输出点和相应的过程映像寄存器位置。这与非立即指令不同，非立即指令只将新值写入过程映像寄存器而不会在扫描周期中立即更新物理输出点。

功率流与指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表： AENO		
梯形图： 梯形图无此指令		
输入输出参数	数据类型	适用操作数
无	无	无

功能说明： AENO 是 ENO 位的逻辑与运算，运算的结果放入栈顶。ENO 是 LAD 中指令盒的布尔输出。如果指令盒在 EN 输入位置被使能而且执行时无错，则 ENO 输出向下一个指令盒传递能流。ENO 位用于堆栈顶部，影响其后指令执行的使能位。STL 指令没有 EN 输入；堆栈顶值必须为逻辑 1，有条件指令才能执行，STL 中也没有 ENO 输出，但采用 AENO 指令可存取 ENO 输出位。AENO 可用于生成与方框 ENO 位相同的效果。

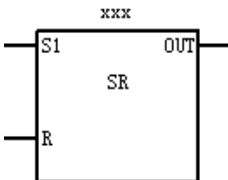
空操作指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表： NOP n		
梯形图： 		
输入输出参数	数据类型	适用操作数
n	字节	常数 (0~255)

功能说明： 空操作指令 (NOP) 对用户程序执行空操作，可以用在用户程序需要短暂延时的地方。

复位主双稳态触发器	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表： STL 中不存在此指令		
梯形图： 		
输入输出参数	数据类型	适用操作数
S	BOOL	Power Flow
R1	BOOL	Power Flow
xxx	BOOL	I, Q, M, V, S
OUT	BOOL	Power Flow

功能说明： 复原主双稳态触发器 (RS) 是一种复原主要位的锁存器。如果设置 (S) 和复原 (R) 信号均为真实，则输出 (OUT) 为虚假。“位”参数指定被设置或复原的布尔参数。供选用输出反映位参数的信号状态。

设置主双稳态触发器	设置 ENO = 0 的错误条件	无
-----------	------------------	---

	影响的特殊内存位	无
指令列表： STL 中不存在此指令		
梯形图： 		
输入输出参数	数据类型	适用操作数
S1	BOOL	Power Flow
R	BOOL	Power Flow
xxx	BOOL	I, Q, M, V, S
OUT	BOOL	Power Flow

功能说明：设置主双稳态触发器（SR）是一种设置主要位的锁存器。如果设置（S1）和复原（R）信号均为真实，则输出（OUT）为真实。“位”参数指定被设置或复原的布尔参数。供选用输出反映位参数的信号状态。

比较指令

字节比较指令	设置 ENO = 0 的错误条件	非法间接地址
	影响的特殊内存位	无
指令列表： LDB= IN1, IN2 AB= IN1, IN2 OB= IN1, IN2 LDB<> IN1, IN2 AB<> IN1, IN2 OB<> IN1, IN2 LDB>= IN1, IN2 AB>= IN1, IN2 OB>= IN1, IN2 LDB<= IN1, IN2 AB<= IN1, IN2 OB<= IN1, IN2 LDB> IN1, IN2 AB> IN1, IN2 OB> IN1, IN2 LDB< IN1, IN2 AB< IN1, IN2 OB< IN1, IN2		
梯形图：		

输入输出参数	数据类型	适用操作数
IN1	字节	IB, QB, MB, SMB, VB, SB, LB, AC, 常数, *VD, *LD, *AC
IN2	字节	IB, QB, MB, SMB, VB, SB, LB, AC, 常数, *VD, *LD, *AC

功能说明:

无论使能位状态如何, "比较"指令均会执行。字节比较指令用于比较两个字节型的值, 字节比较不带符号。

在 STL 中, 比较结果为真时, 栈顶为 1。

在 LAD 中, 比较结果为真时, 触点打开。

<注意> 执行比较指令时如出现以下错误为致命错误, 会使 CTH200 立即停止执行程序:

- 非法间接地址
- 比较实数指令中存在非法实数 (例如, NAN)

为防止出现以上错误, 请确保在使用比较指令前指针已正确初始化并且参数中不存在非法实数。

双整数比较指令	设置 ENO = 0 的错误条件	非法间接地址
	影响的特殊内存位	无
<p>指令列表:</p> <p>LDD= IN1, IN2</p> <p>AD= IN1, IN2</p> <p>OD= IN1, IN2</p> <p>LDD<> IN1, IN2</p> <p>AD<> IN1, IN2</p> <p>OD<> IN1, IN2</p> <p>LDD>= IN1, IN2</p> <p>AD>= IN1, IN2</p> <p>OD>= IN1, IN2</p> <p>LDD<= IN1, IN2</p> <p>AD<= IN1, IN2</p> <p>OD<= IN1, IN2</p> <p>LDD> IN1, IN2</p> <p>AD> IN1, IN2</p> <p>OD> IN1, IN2</p> <p>LDD< IN1, IN2</p> <p>AD< IN1, IN2</p> <p>OD< IN1, IN2</p>		
<p>梯形图:</p>		

输入输出参数	数据类型	适用操作数
IN1	双整数	ID, QD, MD, SD, SMD, VD, LD, HC, AC, 常数, *VD, *LD,
IN2	双整数	*AC

功能说明:

无论使能位状态如何，“比较”指令均会执行。双整数比较指令用于比较两个双整型的值，双整数比较带符号（如 `16#7FFFFFFF > 16#80000000`，因为 `16#80000000` 为负）。

在 STL 中，比较结果为真时，栈顶为 1。

在 LAD 中，比较结果为真时，触点打开。

<注意> 执行比较指令时如出现以下错误为致命错误，会使 CTH200 立即停止执行程序：

- 非法间接地址
- 比较实数指令中存在非法实数（例如，NAN）

为防止出现以上错误，请确保在使用比较指令前指针已正确初始化并且参数中不存在非法实数。

字符串比较指令	设置 ENO = 0 的错误条件	非法间接地址；遇到长度超过 254 个字符的字符串；字符串的起始地址和长度无法放入一个指定的内存区
	影响的特殊内存位	无
指令列表： LDS= IN1, IN2 AS= IN1, IN2 OS= IN1, IN2 LDS<> IN1, IN2 AS<> IN1, IN2 OS<> IN1, IN2		
梯形图： 		
输入输出参数	数据类型	适用操作数
IN1	字符串	VB, 常数字符串, LB, *VD, *LD, *AC
IN2	字符串	VB, LB, *VD, *LD, *AC

ASCII 常数字符串数据类型的格式:

字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束（“字符串常数”）。

字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的格式（一个方格代表一个字节）：

字符串长度	字节 1	字节 2	字节 3	字节 n	字节 254
-------	------	------	------	------	--------

功能说明:

无论使能位状态如何，“比较”指令均会执行。字符串比较指令用于比较两个字符串是否相同。

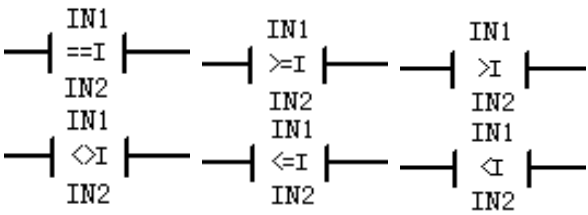
在 STL 中，比较结果为真时，栈顶值为 1。

在 LAD 中，比较结果为真时，触点打开。

<注意> 执行比较指令时如出现以下错误为致命错误，会使 CTH200 立即停止执行程序：

- 非法间接地址
- 长度超过 255 个字符的字符串
- 字符串的起始地址和长度无法放入一个指定的内存区

为了防止出现以上错误，请确保在使用比较指令前指针和存放 ASCII 字符串的内存位置已正确初始化并且为 ASCII 字符串保留的缓冲区可完全放置在指定的内存区中。

整数比较指令	设置 ENO = 0 的错误条件	非法间接地址
	影响的特殊内存位	无
指令列表： LDW= IN1, IN2 AW= IN1, IN2 OW= IN1, IN2 LDW<> IN1, IN2 AW<> IN1, IN2 OW<> IN1, IN2 LDW>= IN1, IN2 AW>= IN1, IN2 OW>= IN1, IN2 LDW<= IN1, IN2 AW<= IN1, IN2 OW<= IN1, IN2 LDW> IN1, IN2 AW> IN1, IN2 OW> IN1, IN2 LDW< IN1, IN2 AW< IN1, IN2 OW< IN1, IN2		
梯形图： 		
输入输出参数	数据类型	适用操作数
IN1	整数	IW, QW, MW, SW, SMW, T, C, VW, LW, AIW, AC,
IN2	整数	常数, *VD, *LD, *AC

功能说明:

无论使能位状态如何, "比较"指令均会执行。整数比较指令用于比较两个整数型的值, 整数比较带符号(如 $16\#7FFF > 16\#8000$, 因为 $16\#8000$ 为负)。

在 STL 中, 比较结果为真时, 栈顶为 1。

在 LAD 中, 比较结果为真时, 触点打开。

<注意> 执行比较指令时如出现以下错误为致命错误, 会使 CTH200 立即停止执行程序:

- 非法间接地址
- 比较实数指令中存在非法实数(例如, NAN)

为防止出现以上错误, 请确保在使用比较指令前指针已正确初始化并且参数中不存在非法实数。

实数比较指令	设置 ENO = 0 的错误条件	非法间接地址 非法实数
	影响的特殊内存位	无
指令列表:		
LDR= IN1, IN2		
AR= IN1, IN2		
OR= IN1, IN2		
LDR<> IN1, IN2		
AR<> IN1, IN2		
OR<> IN1, IN2		
LDR>= IN1, IN2		
AR>= IN1, IN2		
OR>= IN1, IN2		
LDR<= IN1, IN2		
AR<= IN1, IN2		
OR<= IN1, IN2		
LDR> IN1, IN2		
AR> IN1, IN2		
OR> IN1, IN2		
LDR< IN1, IN2		
AR< IN1, IN2		
OR< IN1, IN2		
梯形图:		
输入输出参数	数据类型	适用操作数
IN1	实数	ID, QD, MD, SD, SMD, VD, LD, AC, 常数, *VD, *LD,
IN2	实数	*AC

功能说明:

无论使能位状态如何, "比较"指令均会执行。实数比较指令用于比较两个实数的值, 实数比较带符

号。

在 STL 中，比较结果为真时，栈顶值为 1。

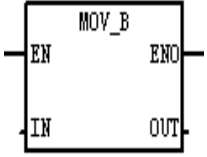
在 LAD 中，比较结果为真时，触点打开。

<注意> 执行比较指令时如出现以下错误为致命错误，会使 CTH200 立即停止执行程序：

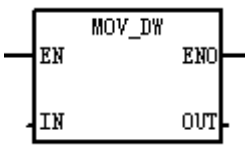
- 非法间接地址
- 比较实数指令中存在非法实数（例如，NAN）

为防止出现以上错误，请确保在使用比较指令前指针已正确初始化并且参数中不存在非法实数。

传送指令

字节赋值指令	设置 ENO = 0 的错误条件	0006 间接地址
	影响的特殊内存位	无
指令列表： MOV_B IN, OUT		
梯形图： 		
输入输出参数	数据类型	适用操作数
IN	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *LD, *AC
OUT	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC

功能说明： 字节赋值指令（MOV_B）将输入字节 IN 的值拷贝赋给输出字节型变量 OUT。

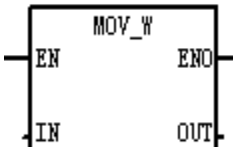
双字赋值指令	设置 ENO = 0 的错误条件	0006 间接地址
	影响的特殊内存位	无
指令列表： MOV_DW IN OUT		
梯形图： 		
输入输出参数	数据类型	适用操作数
IN	双字	VD, ID, QD, MD, SD, SMD, LD, HC, &VB, &IB, &QB, &MB, &SB, &T, &C, &SMB, &AIW, &AQW AC, 常数, *VD, *LD, *AC
OUT	双字	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC

功能说明： 双字赋值指令（MOV_DW）将输入双字 IN 的值拷贝赋给输出双字型变量 OUT。


浮点数赋值指令	设置 ENO = 0 的错误条件	0006 间接地址
	影响的特殊内存位	无
指令列表： MOV_R IN OUT		
梯形图：		

		
输入输出参数	数据类型	适用操作数
IN	浮点数	VD, ID, QD, MD, SD, SMD, LD, AC, 常数, *VD, *LD, *AC
OUT	浮点数	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC

功能说明：浮点数赋值指令（MOV_R）将输入实数 IN 的值拷贝赋给输出浮点型变量 OUT。

字赋值指令	设置 ENO = 0 的错误条件	0006 间接地址
	影响的特殊内存位	无
指令列表： MOV _W IN, OUT		
梯形图：		
		
输入输出参数	数据类型	适用操作数
IN	字	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, 常数, AC, *VD, *AC, *LD
OUT	字	VW, T, C, IW, QW, SW, MW, SMW, LW, AC, AQW, *VD, *AC, *LD

功能说明：字赋值指令（MOV_W）将输入字 IN 的值拷贝赋给输出字型变量 OUT。

存储数据记录到存储卡	设置 ENO = 0 的错误条件	0006 间接地址
	影响的特殊内存位	无
指令列表： SDL LOG, BUF		
梯形图：		
		
输入输出参数	数据类型	适用操作数
LOG	字节	常数 0~3
BUF	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC

功能说明：将当前项目中的数据记录 LOG 拷贝到存储卡中。

存储配方到存储卡	设置 ENO = 0 的错误条件	0006 间接地址
	影响的特殊内存位	无
指令列表： SRC _P RCP, DATA, BUF		
梯形图：		

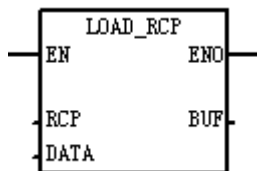
输入输出参数	数据类型	适用操作数
RCP	字节	常数 0~3
DATA	字	VW, T, C, IW, QW, SW, MW, SMW, LW, AC, AQW, *VD, *AC, *LD
BUF	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC

功能说明：将当前项目中的配方 RCP 拷贝到存储卡中。

从存储卡装载配方	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无

指令列表： LRCP RCP, DATA, BUF

梯形图：



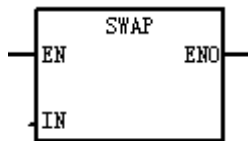
输入输出参数	数据类型	适用操作数
RCP	字节	常数 0~3
DATA	字	VW, T, C, IW, QW, SW, MW, SMW, LW, AC, AQW, *VD, *AC, *LD
BUF	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC

功能说明：将存储卡上的配方 RCP 拷贝到以 BUF 指定的地址为起始地址的内存中。

字节交换指令	设置 ENO = 0 的错误条件	0006 间接地址
	影响的特殊内存位	无

指令列表： SWAP IN

梯形图：



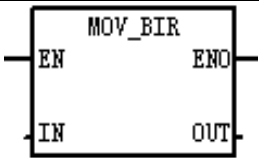
输入输出参数	数据类型	适用操作数
IN	字	VW, IW, QW, MW, SW, SMW, T, C, LW, AC, *VD, *AC, *LD

功能说明：字节交换指令（SWAP）交换输入字 IN 的高位字节和低位字节。

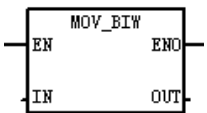
赋值字节立即读取指令	设置 ENO = 0 的错误条件	0006 间接地址 无法存取扩充模块（仅限立即指令）
	影响的特殊内存位	无

指令列表： BIR IN, OUT

梯形图：

		
输入输出参数	数据类型	适用操作数
IN	字节	IB, *VD, *LD, *AC
OUT	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

功能说明: 赋值字节立即读取指令(BIR)读取物理端口输入字节 IN 的值,并将结果拷贝赋给 OUT 指定的地址中,不更新进程映像寄存器。

赋值字节立即写入指令	设置 ENO = 0 的错误条件	0006 间接地址 无法存取扩充模块 (仅限立即指令)
	影响的特殊内存位	无
指令列表: BIW IN, OUT		
梯形图:		
		
输入输出参数	数据类型	适用操作数
IN	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *AC, *LD
OUT	字节	QB, *VD, *LD, *AC

功能说明: 赋值字节立即写入指令(BIW)从读取输入字节 IN 的值,并将值立即写入指定的物理端口 OUT 和对应的进程映像寄存器。

成块赋值字节指令	设置 ENO = 0 的错误条件	0006 间接地址 0091 操作数超出范围
	影响的特殊内存位	无
指令列表: BMB IN, OUT, N		
梯形图:		
		
输入输出参数	数据类型	适用操作数
IN	字节	VB, IB, QB, MB, SB, SMB, LB, *VD, *AC, *LD
OUT	字节	VB, IB, QB, MB, SB, SMB, LB, *VD, *AC, *LD
N	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *AC, *LD

功能说明: 成块赋值字节指令(BMB)将从起始地址 IN 开始的 N 个字节的值拷贝赋给从输出地址 OUT 开始的 N 个字节, N 的范围为 1 至 255。

成块赋值双字指令	设置 ENO = 0 的错误条件	0006 间接地址 0091 操作数超出范围
	影响的特殊内存位	无
指令列表: BMD IN, OUT, N		
梯形图:		

输入输出参数	数据类型	适用操作数
IN	双字	VD, ID, QD, MD, SD, SMD, LD, *VD, *AC, *LD
OUT	双字	VD, ID, QD, MD, SD, SMD, LD, *VD, *AC, *LD
N	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *AC, *LD

功能说明：块赋值双字指令（BMD）将从起始地址 IN 开始的 N 个双字的值拷贝赋给从输出地址 OUT 开始的 N 个双字，N 的范围为 1 至 255。

成块赋值字指令	设置 ENO = 0 的错误条件	0006 间接地址 0091 操作数超出范围
	影响的特殊内存位	无
	指令列表： BMW IN, OUT, N	
梯形图：		
输入输出参数	数据类型	适用操作数
IN	字	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, *VD, *LD, *AC
OUT	字	VW, IW, QW, MW, SW, SMW, LW, T, C, AQW, *VD, *LD, *AC
N	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *AC, *LD

功能说明：成块赋值字指令（BMW）将从起始地址 IN 开始的 N 个字的值拷贝赋给从输出地址 OUT 开始的 N 个字，N 的范围为 1 至 255。

整数计算指令

字节自加自减指令	设置 ENO = 0 的错误条件	0006 间接地址 SM1.1 溢出
	影响的特殊内存位	SM1.0 零结果 SM1.1 溢出
指令列表： INCB OUT DECB OUT		
梯形图：		

输入输出参数	数据类型	适用操作数
IN	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *LD, *AC
OUT	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC

功能说明:

字节自加指令 (INCB) 在输入字节 IN 上加 1, 并将加 1 的结果放入 OUT 指定的地址中。

字节自减指令 (DECB) 在输入字节 IN 上减 1, 并将减 1 的结果放入 OUT 指定的地址中。

字节自加和自减运算不带符号。

在 LAD 中: $IN + 1 = OUT$

$IN - 1 = OUT$

在 STL 中: $OUT + 1 = OUT$

$OUT - 1 = OUT$

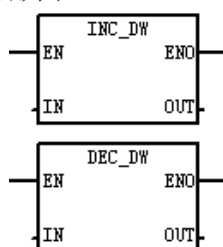
双整数自加自减指令	设置 ENO = 0 的错误条件	0006 间接地址 SM1.1 溢出
	影响的特殊内存位	SM1.0 零结果 SM1.1 溢出 SM1.2 负结果

指令列表:

INCD OUT

DECD OUT

梯形图:



输入输出参数	数据类型	适用操作数
IN	双整数	VD, ID, QD, MD, SD, SMD, LD, AC, HC, 常数, *VD, *LD, *AC
OUT	双整数	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC

功能说明:

双整数自加指令 (INCD) 在输入双字 IN 上加 1, 并将加 1 的结果放入 OUT 指定的地址中。

双整数自减指令 (DECD) 在输入双字 IN 上减 1, 并将减 1 的结果放入 OUT 指定的地址中。

双整数自加和自减运算带符号 (16#7FFFFFFF > 16#80000000)。

在 LAD 中: $IN + 1 = OUT$

$IN - 1 = OUT$

在 STL 中: $OUT + 1 = OUT$

$OUT - 1 = OUT$

整数自加自减指令	设置 ENO = 0 的错误条件	0006 间接地址 SM1.1 溢出
	影响的特殊内存位	SM1.0 零结果 SM1.1 溢出 SM1.2 负结果

指令列表:

INCW OUT

DECW OUT

梯形图:

输入输出参数	数据类型	适用操作数
IN	整数	VW, IW, QW, MW, SW, SMW, AC, AIW, LW, T, C, 常数, *VD, *LD, *AC
OUT	整数	VW, IW, QW, MW, SW, SMW, LW, AC, T, C, *VD, *LD, *AC 整数

功能说明:

整数自加指令 (INCW) 在输入字 IN 上加 1, 并将加 1 的结果放入 OUT 指定的地址中。

整数自减指令 (DECW) 在输入字 IN 上减 1, 并将减 1 的结果放入 OUT 指定的地址中。

整数自加和自减运算带符号 (16#7FFF > 16#8000)。

在 LAD 中: $IN + 1 = OUT$

$IN - 1 = OUT$

在 STL 中: $OUT + 1 = OUT$

$OUT - 1 = OUT$

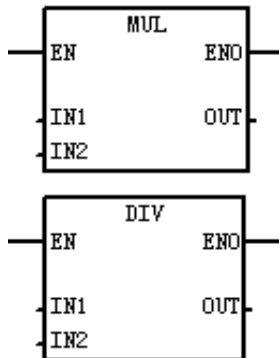
整数与双整数的乘除指令	设置 ENO = 0 的错误条件	0006 间接地址 SM1.1 溢出 SM1.3 除数为0
	影响的特殊内存位	SM1.0 零结果 SM1.1 溢出 SM1.2 负结果 SM1.3 除数为0

指令列表:

MUL IN1 OUT

DIV IN1 OUT

梯形图:



输入输出参数	数据类型	适用操作数
IN1	整数	VW, IW, QW, MW, SW, SMW, T, C, LW, AC, AIW, 常数, *VD, *LD, *AC
IN2	整数	VW, IW, QW, MW, SW, SMW, T, C, LW, AC, AIW, 常数, *VD, *LD, *AC
OUT	双整数	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

功能说明:

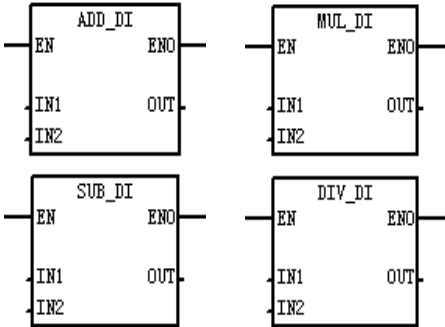
在 STL 乘法指令中, 32 位 OUT 的低位字 (16 位) 被用作乘数之一。在 STL 除法指令中, 32 位 OUT 的低位字 (16 位) 被用作除数。特殊内存 (SM) 位指示错误和非法数值。如果在除法运算过程中设置 SM1.3 (除以零), 则其他数学状态位被保留不变。否则, 在完成数学运算时, 所有受支持的数学状态位均包含有效状态。

在 LAD 中: $IN1 * IN2 = OUT$

$IN1 / IN2 = OUT$

在 STL 中: $IN1 * OUT = OUT$

$OUT / IN1 = OUT$

双整数加减乘除指令	设置 ENO = 0 的错误条件	0006 间接地址 SM1.1 溢出 SM1.3 除数为 0(/I)
	影响的特殊内存位	SM1.0 零结果 SM1.1 溢出 SM1.2 负结果 SM1.3 除数为 0(/I)
指令列表: +D IN1 OUT -D IN1 OUT *D IN1 OUT /D IN1 OUT		
梯形图: 		
输入输出参数	数据类型	适用操作数
IN1	双整数	VD, ID, QD, MD, SMD, SD, LD, AC, HC, 常数, *VD,
IN2	双整数	*LD, *AC
OUT	双整数	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

功能说明:

双整数加法运算指令 (+D) 将两个 32 位整数相加, 并将计算结果 (32 位整数) 放入 OUT 指定的地址中。如果结果超出 32 位长度表示范围, 则设置溢出位。

双整数减法运算指令 (-D) 将两个 32 位整数相减, 并将计算结果 (32 位整数) 放入 OUT 指定的地址中。如果结果超出 32 位长度表示范围, 则设置溢出位。

双整数乘法运算指令 (*D) 将两个 32 位整数相乘, 并将乘积结果 (32 位整数) 放入 OUT 指定的地址中。如果结果超出 32 位长度表示范围, 则设置溢出位。

双整数除法运算指令 (/D) 将两个 32 位整数相除, 并将商 (32 位整数) 放入 OUT 指定的地址中, 不保留余数。如果结果超出 32 位长度表示范围, 则设置溢出位。

在 LAD 中: $IN1 + IN2 = OUT$

IN1 - IN2 = OUT

IN1 * IN2 = OUT

IN1 / IN2 = OUT

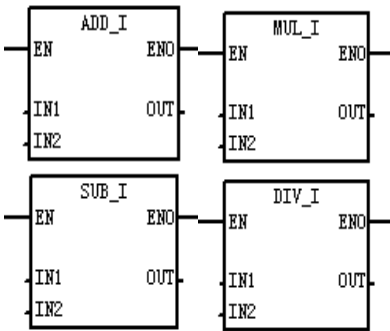
在 STL 中: IN1 + OUT = OUT

OUT - IN1 = OUT

IN1 * OUT = OUT

OUT / IN1 = OUT

SM1.1 指示溢出错误和非法数值。如果设置 SM1.1, 则 SM1.0 和 SM1.2 状态无效, 且原来的输入操作数不改动。如果未设置 SM1.3, 则数学操作完成, 并产生有效的结果, 且 SM1.0 和 SM1.2 包含有效状态。如果在除法运算过程中设置 SM1.3, 则其他数学状态位将保持不变。

整数加减乘除指令	设置 ENO = 0 的错误条件	0006 间接地址 SM1.1 溢出 SM1.3 除数为0(/I)
	影响的特殊内存位	SM1.0 零结果 SM1.1 溢出 SM1.2 负结果 SM1.3 除数为0(/I)
指令列表: +I IN1 OUT -I IN1 OUT *I IN1 OUT /I IN1 OUT		
梯形图: 		
输入输出参数	数据类型	适用操作数
IN1	整数	VW, IW, QW, MW, SW, SMW, T, C, AC, LW, AIW, 常数, *VD, *LD, *AC
IN2	整数	
OUT	整数	VW, IW, QW, MW, SW, SMW, T, C, LW, AC, *VD, *LD, *AC

功能说明:

整数加法运算指令 (+I) 将两个 16 位整数相加, 并将计算结果 (16 位整数) 放入 OUT 指定的地址中。如果结果超出 16 位长度表示范围, 则设置溢出位。

整数减法运算指令 (-I) 将两个 16 位整数相减, 并将计算结果 (16 位整数) 放入 OUT 指定的地址中。如果结果超出 16 位长度表示范围, 则设置溢出位。

整数乘法运算指令 (*I) 将两个 16 位整数相乘, 并将乘积结果 (16 位整数) 放入 OUT 指定的地址中。如果结果超出 16 位长度表示范围, 则设置溢出位。

整数除法运算指令 (/I) 将两个 16 位整数相除, 并将商 (16 位整数) 放入 OUT 指定的地址中, 不保留余数。如果结果超出 16 位长度表示范围, 则设置溢出位。

在 LAD 中: IN1 + IN2 = OUT

$$IN1 - IN2 = OUT$$

$$IN1 * IN2 = OUT$$

$$IN1 / IN2 = OUT$$

在 STL 中: $IN1 + OUT = OUT$

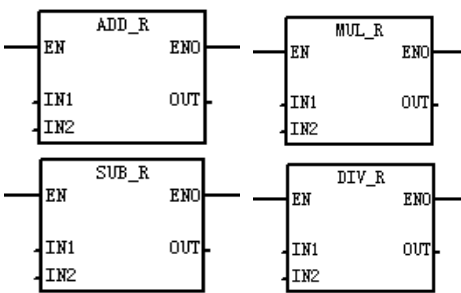
$$OUT - IN1 = OUT$$

$$IN1 * OUT = OUT$$

$$OUT / IN1 = OUT$$

SM1.1 指示溢出错误和非法数值。如果设置 SM1.1, 则 SM1.0 和 SM1.2 状态无效, 且原来的输入操作数不改动。如果未设置 SM1.3, 则数学操作完成, 并产生有效的结果, 且 SM1.0 和 SM1.2 包含有效状态。如果在除法运算过程中设置 SM1.3, 则其他数学状态位将保持不变。

浮点数运算

浮点数加减乘除指令	设置 ENO = 0 的错误条件	0006 间接地址 SM1.1 溢出 SM1.3 除数为0(/R)
	影响的特殊内存位	SM1.0 零结果 SM1.1 溢出 SM1.2 负结果 SM1.3 除数为0(/R)
指令列表: +R IN1 OUT -R IN1 OUT *R IN1 OUT /R IN1 OUT		
梯形图: 		
输入输出参数	数据类型	适用操作数
IN1	浮点数	VD, ID, QD, MD, SMD, SD, LD, AC, 常数, *VD, *LD,
IN2	浮点数	*AC
OUT	浮点数	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

功能说明:

浮点数加法指令 (+R) 将两个 32 位实数相加, 并将计算结果 (32 位实数) 放入 OUT 指定地址中。

浮点数减法指令 (-R) 将两个 32 位实数相减, 并将计算结果 (32 位实数) 放入 OUT 指定地址中。

浮点数乘法指令 (*R) 将两个 32 位实数相乘, 并将乘积 (32 位实数) 放入 OUT 指定的地址中。

浮点数除法指令 (/R) 将两个 32 位实数相除, 并将商 (32 位实数) 放入 OUT 指定的地址中。

在 LAD 中: $IN1 + IN2 = OUT$

$$IN1 - IN2 = OUT$$

$$IN1 * IN2 = OUT$$

IN1 / IN2 = OUT

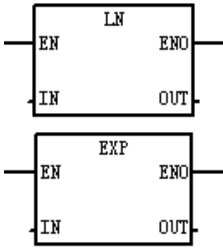
在 STL 中: IN1 + OUT = OUT

OUT - IN1 = OUT

IN1 * OUT = OUT

OUT / IN1 = OUT

SM1.1 指示溢出错误和非法数值。如果在除法运算中设置 SM1.3, 则其他数学状态位不变, 而且原始输入操作数也不变。SM1.1 用于表示溢出错误和非法数值。如果设置 SM1.1, 则 SM1.0 和 SM1.2 状态无效, 原始输入操作数不变。如果在除法运算过程中未设置 SM1.1 和 SM1.3, 则说明数学运算已完成, 得出有效结果, 而且 SM1.0 和 SM1.2 包含有效状态。

自然对数自然指数指令	设置 ENO = 0 的错误条件	0006 间接地址 SM1.1 溢出
	影响的特殊内存位	SM1.0 零结果 SM1.1 溢出 SM1.2 负结果
指令列表: LN IN, OUT EXP IN, OUT		
梯形图: 		
输入输出参数	数据类型	适用操作数
IN	浮点数	VD, ID, QD, MD, SMD, SD, LD, AC, 常数, *VD, *LD, *AC
OUT	浮点数	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

功能说明:

自然对数指令 (LN) 对 IN 中的数值进行自然对数计算, 并将结果置于 OUT 中。欲从自然对数获得以 10 为底数的对数, 用自然对数除以 2.302585 (约等于 10 的自然对数)。欲将任何实数提升为另一个实数的乘幂, 包括分数指数: 将"自然指数"指令与"自然对数"指令相结合。例如, 欲将 X 提升为 Y 乘幂, 输入以下指令: EXP (Y * LN (X))。

自然指数指令 (EXP) 进行 e 的 IN 次方指数计算, 并将结果置于 OUT 中。欲将任何实数提升为另一个实数的乘幂, 包括分数指数: 将"自然指数"指令与"自然对数"指令相结合。例如, 欲将 X 提升为 Y 乘幂, 输入以下指令 EXP (Y * LN (X))。

举例:

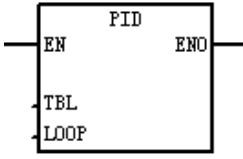
5 的立方 = 5³=EXP(3*LN(5))=125

125 的立方根 = 125^(1/3)=EXP(1/3*LN(125))=5

5 的立方的平方根 = 5^(3/2)=EXP(3/2*LN(5))=11.18034

SM1.1 用于指示溢出错误和非法数值。如果设置 SM1.1, 则 SM1.0 和 SM1.2 状态无效, 且原来的输入操作数不改动。如果未设置 SM1.3, 则数学操作完成, 并产生有效的结果, 且 SM1.0 和 SM1.2 包含有效状态。

PID 控制指令	设置 ENO = 0 的错误条件	0006 间接地址 SM1.1 溢出
----------	------------------	-----------------------

	影响的特殊内存位	SM1.1 溢出
指令列表: PID TBL LOOP		
梯形图:		
		
输入输出参数	数据类型	适用操作数
TBL	字节	VB
LOOP	字节	常数(0 to 7)

功能说明:

本指令有两个操作数: 表示回路表起始地址的 TBL 地址和 0 至 7 常数的"回路"号码。程序中可使用八条 PID 指令。如果两条或多条 PID 指令使用相同的回路号码(即使它们的表格地址不同), PID 计算会互相干扰, 结果难以预料。回路表存储用于控制和监控回路运算的参数, 包括程序变量、设置点、输出、增益、样本时间、整数时间(重设)、导出时间(速率)以及整数和(偏差)的当前值及先前值。

如果回路表起始地址或指令中指定的 PID 回路号码操作数超出范围, CPU 编译器将生成一则错误(范围错误), 编译将会失败。

PID 指令不对某些回路表输入值进行范围检查。您必须保证程序变量和设置点(以及作为输入的偏差和先前程序变量)是 0.0 和 1.0 之间的实数。

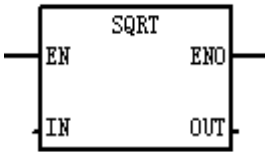
如果进行 PID 计算的数学运算时遇到错误, 将设置 SM1.1(溢出或非法数值)并终止 PID 指令的执行。(对回路表中的输出数值的更新可能不完整, 因此您应当忽略这些数值, 并在执行下一个回路 PID 指令之前纠正引起数学错误的输入值。)

取样速率:

欲按要求的样品速率进行 PID 计算, 必须按计时器的控制速率从定时中断程序或从主程序执行 PID 指令。样品时间必须通过回路表作为 PID 指令输入提供。

使用 PID 向导:

MagicWorks PLC 提供 PID 向导, 指导您为闭环控制程序定义 PID 算法。选择菜单命令工具 > 指令向导, 并从指令向导窗口选择 PID。

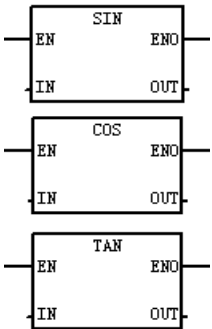
平方根指令	设置 ENO = 0 的错误条件	0006 间接地址 SM1.1 溢出
	影响的特殊内存位	SM1.0 零结果 SM1.1 溢出 SM1.2 负结果
指令列表: SQRT IN OUT		
梯形图:		
		
输入输出参数	数据类型	适用操作数
IN	浮点数	VD, ID, QD, MD, SMD, SD, LD, AC, 常数, *VD, *LD, *AC
OUT	浮点数	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

功能说明:

平方根指令 (SQRT) 对 32 位实数 (IN) 取平方根, 产生一个 32 位实数结果并将结果放入 OUT 中, 如以下等式所示:

$$\sqrt{IN} = OUT$$

SM1.1 用于表示溢出错误和非法数值。如果设置 SM1.1, 则 SM1.0 和 SM1.2 状态无效, 原始输入操作数不变。如果未设置 SM1.1, 则说明数学运算已完成, 得出有效结果, 而且 SM1.0 和 SM1.2 包含有效状态。

正弦余弦正切指令	设置 ENO = 0 的错误条件	0006 间接地址 SM1.1 溢出
	影响的特殊内存位	SM1.0 零结果 SM1.1 溢出 SM1.2 负结果
指令列表: SIN IN OUT COS IN OUT TAN IN OUT		
梯形图: 		
输入输出参数	数据类型	适用操作数
IN	浮点数	VD, ID, QD, MD, SMD, SD, LD, AC, 常数, *VD, *LD, *AC
OUT	浮点数	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC


功能说明:

正弦指令 (SIN) 对角度值 IN 进行三角运算, 并将结果放置在 OUT 中。输入角以弧度为单位。欲将输入角从角度转换成弧度, 用角度乘以 1.745329E-2 (约等于 pi/180)。

余弦指令 (COS) 对角度值 IN 进行三角运算, 并将结果放置在 OUT 中。输入角以弧度为单位。欲将输入角从角度转换成弧度, 用角度乘以 1.745329E-2 (约等于 pi/180)。

正切指令 (TAN) 对角度值 IN 进行三角运算, 并将结果放置在 OUT 中。输入角以弧度为单位。欲将输入角从角度转换成弧度, 用角度乘以 1.745329E-2 (约等于 pi/180)。

SM1.1 用于指示溢出错误和非法数值。如果设置 SM1.1, 则 SM1.0 和 SM1.2 状态无效, 且原来的输入操作数不改动。如果未设置 SM1.3, 则数学操作完成, 并产生有效的结果, 且 SM1.0 和 SM1.2 包含有效状态。

转换指令 

字符串至整数转换指令	设置 ENO = 0 的错误条件	0006 间接地址 0091 操作数范围 009B 非法指数(指定起始位置值0的字符串操作) SM1.1 溢出或非法值
------------	------------------	--

		影响的特殊内存位	SM1.1(溢出或非数值)
指令列表: STI IN, INDEX, OUT			
梯形图:			
输入输出参数	数据类型	适用操作数	
IN	字符串	VB, 常数字符串, LB, *VD, *LD, *AC	
OUT	整数	VW, IW, QW, MW, SW, SMW, LW, T, C, AQW, AC, *VD, *LD, *AC	
INDEX	字节	VB, IB, QB, MB, SB, SMB, LB, 常数, AC, *VD, *LD, *AC	

ASCII 常数字符串数据类型的格式:

字符串是一系列字符和对应的内存地址, 每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度(即字符数)的整数。如果常数字符串被直接输入程序编辑器或数据块, 那么该字符串必须用双引号字符起始和结束("字符串常数")。

字符串的最大长度是 255 个字节(254 个字符加上长度字节)。下面的内存图显示了字符串数据类型的格式(一个方格代表一个字节):

字符串长度	字节 1	字节 2	字节 3	字节 n	字节 254
-------	------	------	------	------	--------

STI 指令用以下形式转换字符串:

[spaces] [+ or -] [digits 0-9]

INDEX 值通常设为 1, 从字符串的第一个字符开始转换。可将该值设为其他值, 在字符串中的不同点开始转换。当输入字符串包含不属于需要转换数字一部分的文本时, 可采用此种方法。例如, 如果输入字符串是"value:16000", 您可以将 INDEX 设为值 7, 跳过字符串开始的字"value:"。

当达到字符串结束时或遇到第一个无效字符时, 转换终止。无效字符是数字(0-9)以外的任何字符。

每当转换产生一个对于输出值过大的整数值时, 则设置溢出错误(SM1.1)。例如, 如果输入字符串产生一个大于 32767 或小于-32768 的值时, STI 指令设置溢出错误。

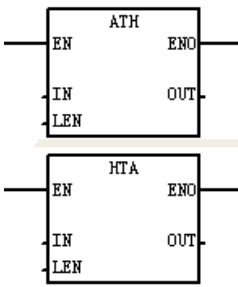
如果当输入字符串未包含有效值而无法执行转换时, 也会设置溢出错误(SM1.1)。例如, 如果输入字符串包含"A123", 转换指令会设置 SM1.1(溢出), 输出值保持不变。

下表是有效和无效字符串转换为整数/双整数/实数时的举例(INDEX = 1):

输入字符串	有效性	整数转换结果	双整数转换结果	实数转换结果
"65535"	有效	65535	65535	65535.0
"-0032768"	有效	-32768	-32768	-32768.0
"+32767.999 "	有效	32767	32767	32767.0
"00065535FFFF"	有效	65535	65535	65535.0
"65536"	有效	65536 (SM1.1=1, 溢出)	65536	65536.0
"-002147483648"	有效	-2147483648 (SM1.1=1, 溢出)	-2147483648	-2147483648.0
"+002147483648"	有效	+2147483648 (SM1.1=1, 溢出)	+2147483648 (SM1.1=1, 溢出)	+2147483648.0
"F255"	无效(SM1.1=1,			

	无效)			
"++255"	无效(SM1.1=1, 无效)			
" "	无效(SM1.1=1, 无效)			

功能说明：字符串至整数转换指令将字符串数值 IN 转换为整数，并将转换结果存储在 OUT 指定的地址中，从字符串的第 INDEX 个字符位置开始转换。

ASCII 码与十六进制数字的转换指令	设置 ENO = 0 的错误条件	0006 间接地址 0091 操作数范围 SM1.7 非法 ASCII 值(ATH)
	影响的特殊内存位	SM1.7 (非法 ASCII)
指令列表： ATH IN, OUT, LEN HTA IN, OUT, LEN		
梯形图： 		
输入输出参数	数据类型	适用操作数
IN	字节	VB, IB, QB, MB, SB, SMB, LB, *VD, *AC, *LD
OUT	字节	VB, IB, QB, MB, SB, SMB, LB, *VD, *AC, *LD
LEN	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *LD, *AC

操作数说明：

有效 ASCII 输入字符为：

字母数字字符 0~9 和 A~F
 对应十六进制代码值 30~39 和 41~46

功能说明：

ASCII 至 HEX 转换指令将从内存位置 IN 开始的长度为 LEN 的 ASCII 字符号码转换十六进制数字，并放入 OUT 指定的地址。ASCII 字符串的最大长度为 255 字符。
 HEX 至 ASCII 转换指令将从输入字节 IN 开始的十六进制数字转换成从 OUT 开始的 ASCII 字符。欲转换的十六进制数字位数由长度 LEN 指定，可转换的最大十六进制数字位数为 255。

整数与 BCD 码转换指令	设置 ENO = 0 的错误条件	0006 间接地址 SM1.6 无效 BCD 数值
	影响的特殊内存位	SM1.6 (无效 BCD)
指令列表： BCDI OUT IBCD OUT		
梯形图：		

输入输出参数	数据类型	适用操作数
IN	字	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, 常数, *VD, *AC, *LD
OUT	字	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *LD, *AC

操作数说明:

BCD 格式的有效范围:

数据类型	最小值			最大值		
	十进制	十六进制	BCD 码	十进制	十六进制	BCD 码
字						
字	0	0000	0000 0000 0000 0000	9999	9999	1001 1001 1001 1001

功能说明:

BCD 至整数指令将二进制编码的十进制值 IN 转换成整数值，并将结果载入 OUT 指定的变量中。IN 的有效范围是 0 至 9999 BCD。

整数至 BCD 指令将输入整数值 IN 转换成二进制编码的十进制数，并将结果载入 OUT 指定的变量中。IN 的有效范围是 0 至 9999 BCD。例如，您可以将双整值转换为实数。您还可以在整数和 BCD 格式之间转换。对于 STL，IN 和 OUT 参数使用相同的地址。

字节至整数转换指令	设置 ENO = 0 的错误条件	0006 间接地址
	影响的特殊内存位	无
指令列表: BTI IN, OUT		
梯形图:		
输入输出参数	数据类型	适用操作数
IN	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *AC, *VD, *LD
OUT	整数	VW, IW, QW, MW, SW, SMW, LW, AQW, T, C, AC, *VD, *LD, *AC

功能说明: 字节至整数转换指令将字节数值 IN 转换成整数值，并将结果写入 OUT 指定的变量中。因为字节不带符号，所以无符号扩展。

解码指令	设置 ENO = 0 的错误条件	0006 间接地址
	影响的特殊内存位	无
指令列表: DECO IN, OUT		
梯形图:		

输入输出参数	数据类型	适用操作数
IN	字节	VB, IB, QB, MB, SMB, LB, SB, AC, 常数, *VD, *LD, *AC
OUT	字	VW, IW, QW, MW, SMW, LW, SW, AQW, T, C, AC, *VD, *AC, *LD

功能说明: 解码指令设置输出字 OUT 中与输入字节 IN 低半字节 (低 4 位) 表示的位数相对应的位为 1, 设置输出字 OUT 的所有其它位均为 0。

示例程序:

LD SM0.0

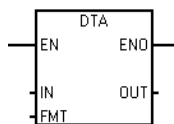
DECO 2#10101111, VW0 //设置 VW0 的第 15(2#1111)位为 1, 其它位为 0

运行结果: VW0 = 2#1000 0000 0000 0000

双整数至 ASCII 码转换指令	设置 ENO = 0 的错误条件	0006 间接地址 FMT 位 > 0 (用于 FMT 值最高的4个位) nnn > 5
	影响的特殊内存位	无

指令列表: DTA IN, OUT, FMT

梯形图:



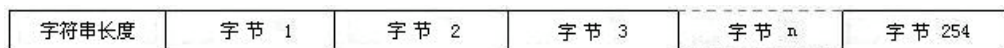
输入输出参数	数据类型	适用操作数
IN	双整数	VD, ID, QD, MD, SD, SMD, LD, HC, 常数, AC, *VD, *AC, *LD
OUT	字节	VB, IB, QB, MB, SB, SMB, LB, *VD, *LD, *AC
FMT	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *LD, *AC

操作数说明:

ASCII 常数字符串数据类型的格式:

字符串是一系列字符和对应的内存地址, 每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度 (即字符数) 的整数。如果常数字符串被直接输入程序编辑器或数据块, 那么该字符串必须用双引号字符起始和结束 ("字符串常数")。

字符串的最大长度是 255 个字节 (254 个字符加上长度字节)。下面的内存图显示了字符串数据类型的格式(一个方格代表一个字节):



操作数 FMT 定义:

位号	7	6	5	4	3	2	1	0
值	0	0	0	0	c	n	n	n

c: 整数与小数部分的分隔符, c = 1:使用逗号作为整数与小数部分的分隔符; c = 0:使用小数点作为整数与小数部分的分隔符

nnn: 小数部分的位数, 有效范围为 0~5, 输出字符串的长度始终为 8 个字符。nnn 等于 0 会使转换结果显示为不带小数点, 当 nnn 值大于 5 时, 输出显示为 12 个 ASCII 空格字符的字符串。

整数至 ASCII 码的转换遵循下列规则:

- 1) 正值写入输出缓冲区, 不带符号。
- 2) 负值写入输出缓冲区, 带起始负号(-)。

3) 小数点左侧的起首零（与小数点相邻的数字除外）被压缩。

4) 输出缓冲区中的数值右对齐。

下表显示几个使用小数点（ $c = 0$ ）和小数点右面四位小数（ $nnn = 100$ ）格式的值范例。

	OUT	字节 OUT										
		+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11
IN=123							0	.	0	1	2	3
IN=-12345						-	1	.	2	3	4	5
IN=1234567					1	2	3	.	4	5	6	7

功能说明：整数至 ASCII 转换指令将整数值 IN 转换成 ASCII 字符数组。格式 FMT 指定转换的精度，以及是将小数点显示为逗号还是点号。转换结果放入从 OUT 开始的 12 个连续字节中。ASCII 字符数组总是 12 个字符。

双整数至整数转换指令	设置 ENO = 0 的错误条件	0006 间接地址
	影响的特殊内存位	SM1.1 (溢出)
	指令列表: DTI IN, OUT	
梯形图:		
输入输出参数	数据类型	适用操作数
IN	双整数	VD, ID, QD, MD, SD, SMD, LD, HC, AC, 常数, *VD, *LD, *AC
OUT	整数	VW, IW, QW, MW, SW, SMW, LW, AQW, T, C, AC, *VD, *LD, *AC

功能说明：双整数至整数转换指令将输入的双整数值 IN 转换成整数值，并将结果放入 OUT 指定的变量中。如果转换的值过大，则无法在输出中表示，会导致溢出错误，输出不受影响。

双整数至实数转换指令	设置 ENO = 0 的错误条件	0006 间接地址
	影响的特殊内存位	无
指令列表: DTR IN, OUT		
梯形图:		
输入输出参数	数据类型	适用操作数
IN	双整数	VD, ID, QD, MD, SD, SMD, LD, HC, AC, 常数, *VD, *AC, *LD
OUT	实数	VD, ID, QD, MD, SD, SMD, LD, HC, AC, *VD, *AC, *LD

功能说明：双整数至实数转换指令将 32 位带符号的整数 IN 转换成 32 位实数，并将结果放入 OUT 指定的变量中。

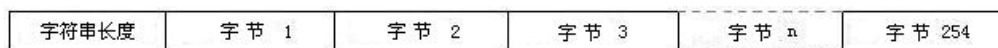
双整数至字符串	设置 ENO = 0 的错误	0006 间接地址
---------	----------------	-----------

转换指令	条件	0091 操作数范围 非法格式 (nnn > 5)
	影响的特殊内存位	无
指令列表: DTS IN, OUT, FMT		
梯形图:		
输入输出参数	数据类型	适用操作数
IN	双整数	VD, ID, QD, MD, SD, SMD, LD, HC, 常数, AC, *VD, *AC, *LD
OUT	字符串	VB, *VD, LB, *AC, *LD
FMT	字节	VB, IB, QB, MB, SB, SMB, LB, 常数, AC, *VD, *LD, *AC

ASCII 常数字符串数据类型的格式:

字符串是一系列字符和对应的内存地址, 每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度(即字符数)的整数。如果常数字符串被直接输入程序编辑器或数据块, 那么该字符串必须用双引号字符起始和结束 ("字符串常数")。

字符串的最大长度是 255 个字节 (254 个字符加上长度字节)。下面的内存图显示了字符串数据类型的格式(一个方格代表一个字节):



操作数 FMT 定义:

位号	7	6	5	4	3	2	1	0
值	0	0	0	0	c	n	n	n

c: 整数与小数部分的分隔符, c = 1:使用逗号作为整数与小数部分的分隔符; c = 0:使用小数点作为整数与小数部分的分隔符。

nnn: 小数部分的位数, 有效范围为 0~5, 输出字符串的长度始终为 12 个字符。nnn 等于 0 会使转换结果显示格式为不带小数点, 当 nnn 值大于 5 时, 输出显示为 12 个 ASCII 空格字符的字符串。

下表显示几个使用小数点 (c = 0) 和小数点右面四位小数 (nnn = 100) 格式的值范例。位于 OUT 位置的值是字符串长度。

	OUT	字节 OUT											
		+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11	+12
IN=123	12							0	.	0	1	2	3
IN=12345	12							1	.	2	3	4	5
IN=1234567	12					1	2	3	.	4	5	6	7

功能说明: 双整数至字符串转换指令将输入的双整数 IN 转换为长度为 12 个字符的 ASCII 字符串。格式 (FMT) 指定小数点右面的转换精度, 小数点是显示为逗号还是句点, 转换结果字符串写入从 OUT 开始的 13 个连续字节中。

提示: 根据以下规则制定格式的输出字符串:

- 整数不带符号写入输出缓冲区。
- 负数带起始减号 (-) 写入输出缓冲区。
- 小数点左面的起始零 (除紧靠小数点的数字外) 被压缩。
- 小数点右面的数值被取整, 使之符合指定的小数点右面的位数。

- 输出字符串的大小最小必须比小数点右面的位数大三个字节。
- 输出字符串中的数值必须右对齐。

编码指令	设置 ENO = 0 的错误条件	0006 间接地址
	影响的特殊内存位	无
指令列表: ENCO IN, OUT		
梯形图:		
		
输入输出参数	数据类型	适用操作数
IN	字	VW, IW, QW, MW, SMW, LW, SW, AIW, T, C, AC, 常数, *VD, *AC, *LD
OUT	字节	VB, IB, QB, MB, SMB, LB, SB, AC, *VD, *LD, *AC

功能说明: 编码指令将输入字 IN 位值为 1 的最低位的位数作为输出字节 OUT 的低半字节, 写入输出字节的最低 4 位中。

示例程序:

```
LD      SM0.1
MOVB   0, VB0
LD      SM0.0
ENCO   2#1100000000, VB0      //2#1100000000 最低不为 0 的位是第 8 位, 因此 VB0
                                的低半字节为 2#1000
```

运行结果: VB0 = 2#0000 1000

整数至 ASCII 码转换指令	设置 ENO = 0 的错误条件	0006 间接地址 FMT 位 > 0 (用于 FMT 值最高的 4 个位) nnn > 5
	影响的特殊内存位	无
指令列表: ITA IN, OUT, FMT		
梯形图:		
		
输入输出参数	数据类型	适用操作数
IN	整数	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, AC, 常数, *VD, *LD, *AC
OUT	字节	VB, IB, QB, MB, SB, SMB, LB, *VD, *LD, *AC
FMT	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *LD, *AC

ASCII 常数字符串数据类型的格式:

字符串是一系列字符和对应的内存地址, 每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度(即字符数)的整数。如果常数字符串被直接输入程序编辑器或数据块, 那么该字符串必须用双引号字符起始和结束("字符串常数")。

字符串的最大长度是 255 个字节(254 个字符加上长度字节)。下面的内存图显示了字符串数据类型的格式(一个方格代表一个字节):

字符串长度	字节 1	字节 2	字节 3	字节 n	字节 254
-------	------	------	------	------	--------

操作数 FMT 定义:

位号	7	6	5	4	3	2	1	0
值	0	0	0	0	c	n	n	n

c: 整数与小数部分的分隔符, c = 1:使用逗号作为整数与小数部分的分隔符; c = 0:使用小数点作为整数与小数部分的分隔符。

nnn: 小数部分的位数, 有效范围为 0~5, 输出字符串的长度始终为 8 个字符。nnn 等于 0 会使转换结果显示为不带小数点, 当 nnn 值大于 5 时, 输出显示为 8 个 ASCII 空格字符的字符串。

整数至 ASCII 码的转换遵循下列规则:

- 1) 正值写入输出缓冲区, 不带符号。
- 2) 负值写入输出缓冲区, 带起始负号(-)。
- 3) 小数点左侧的起首零(与小数点相邻的数字除外)被压缩。
- 4) 输出缓冲区中的数值右对齐。

下表显示几个使用小数点 (c = 0) 和小数点右面三位小数 (nnn = 011) 格式的值范例。

	OUT	字节 OUT						
		+1	+2	+3	+4	+5	+6	+7
IN=12				0	.	0	1	2
IN=-123			-	0	.	1	2	3
IN=1234				1	.	2	3	4
IN=-12345		-	1	2	.	3	4	5

功能说明: 整数至 ASCII 转换指令将整数值 IN 转换成 ASCII 字符数组。格式 FMT 指定转换的精度, 以及是将小数点显示为逗号还是点号。转换结果放入从 OUT 开始的 8 个连续字节中。ASCII 字符数组总是 8 个字符。

整数至字节转换指令	设置 ENO = 0 的错误条件	0006 间接地址 溢出或非数值
	影响的特殊内存位	SM1.1 (溢出)
	指令列表: ITB IN, OUT	
梯形图:		
输入输出参数	数据类型	适用操作数
IN	整数	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, 常数, *VD, *LD, *AC
OUT	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

功能说明: 整数至字节转换指令将输入的整数 IN 转换成字节值, 并将结果放入 OUT 指定的变量中。数值 0 至 255 被转换。超过此范围的输入值将导致溢出, 输出不受影响。

整数至双整数转换指令	设置 ENO = 0 的错误条件	0006 间接地址
	影响的特殊内存位	无

指令列表: ITD IN, OUT		
梯形图:		
输入输出参数	数据类型	适用操作数
IN	整数	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, 常数, *VD, *LD, *AC
OUT	双整数	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC

功能说明: 整数至双整数转换指令将输入的整数值 IN 转换成双整数值, 并将结果放入 OUT 指定的变量中。符号被扩展。

整数至字符串转换指令	设置 ENO = 0 的错误条件	0006 间接地址 0091 操作数范围 非法格式 (nnn > 5)
	影响的特殊内存位	无
指令列表: ITS IN, OUT, FMT		
梯形图:		
输入输出参数	数据类型	适用操作数
IN	整数	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, 常数, AC, *VD, *LD, *AC
OUT	字符串	VB, *VD, LB, *AC, *LD
FMT	字节	VB, IB, QB, MB, SB, SMB, LB, 常数, AC, *VD, *LD, *AC

ASCII 常数字符串数据类型的格式:

字符串是一系列字符和对应的内存地址, 每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度(即字符数)的整数。如果常数字符串被直接输入程序编辑器或数据块, 那么该字符串必须用双引号字符起始和结束("字符串常数")。

字符串的最大长度是 255 个字节(254 个字符加上长度字节)。下面的内存图显示了字符串数据类型的格式(一个方格代表一个字节):

字符串长度	字节 1	字节 2	字节 3	字节 n	字节 254
-------	------	------	------	------	--------

操作数 FMT 定义:

位号	7	6	5	4	3	2	1	0
值	0	0	0	0	c	n	n	n

c: 整数与小数部分的分隔符, **c = 1**: 使用逗号作为整数与小数部分的分隔符; **c = 0**:使用小数点作为整数与小数部分的分隔符。

nnn: 小数部分的位数, 有效范围为 0~5, 输出字符串的长度始终为 8 个字符。**nnn** 等于 0 会使转换结果显示格式为不带小数点, 当 **nnn** 值大于 5 时, 输出显示为 8 个 ASCII 空格字符的字符串。

下表显示几个使用小数点 (**c = 0**) 和小数点右面三位小数 (**nnn = 011**) 格式的值范例。位于 **OUT** 位置的值是字符串长度。

	OUT	字节 OUT							
		+1	+2	+3	+4	+5	+6	+7	+8
IN=12	8				0	.	0	1	2
IN=123	8				0	.	1	2	3
IN=1234	8				1	.	2	3	4
IN=12345	8			1	2	.	3	4	5

功能说明: 整数至字符串转换指令将输入的整数 **IN** 转换为长度为 8 个字符的 ASCII 字符串。格式 (**FMT**) 指定小数点右面的转换精度, 小数点是显示为逗号还是句点, 转换结果字符串写入从 **OUT** 开始的 9 个连续字节中。

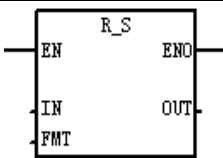
提示: 根据以下规则制定格式的输出字符串:

- 整数不带符号写入输出缓冲区。
- 负数带起始减号 (-) 写入输出缓冲区。
- 小数点左面的起始零 (除紧靠小数点的数字外) 被压缩。
- 小数点右面的数值被取整, 使之符合指定的小数点右面的位数。
- 输出字符串的大小最小必须比小数点右面的位数大三个字节。
- 输出字符串中的数值必须右对齐。

取整指令	设置 ENO = 0 的错误条件	0006 间接地址 SM1.1 溢出或非法值
	影响的特殊内存位	SM1.1 (溢出)
指令列表: ROUND IN, OUT		
梯形图:		
输入输出参数	数据类型	适用操作数
IN	实数	VD, ID, QD, MD, SD, SMD, LD, AC, 常数, *VD, *LD, *AC
OUT	双整数	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC

功能说明: 取整指令将输入的实数 **IN** 转换成双整数值, 并将结果存入 **OUT** 指定的变量中。小数部分采取四舍五入的方法取整。

实数至字符串转换指令	设置 ENO = 0 的错误条件	0006 间接地址 0091 操作数范围 非法格式: nnn > 5 ssss < 3 ssss < 要求的字符数
	影响的特殊内存位	无
指令列表: RTS IN, OUT, FMT		
梯形图:		

		
输入输出参数	数据类型	适用操作数
IN	实数	VD, ID, QD, MD, SD, SMD, LD, 常数, AC, *VD, *LD, *AC
OUT	字符串	VB, LB, *VD, *AC, *LD
FMT	字节	VB, IB, QB, MB, SB, SMB, LB, 常数, AC, *VD, *LD, *AC

ASCII 常数字符串数据类型的格式:

字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束（"字符串常数"）。

字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的格式(一个方格代表一个字节):

字符串长度	字节 1	字节 2	字节 3	字节 n	字节 254
-------	------	------	------	------	--------

操作数 FMT 定义:

位号	7	6	5	4	3	2	1	0
值	s	s	s	s	c	n	n	n

ssss: 输出字符串长度，小于 3 无效。

c: 整数与小数部分的分隔符，c = 1:使用逗号作为整数与小数部分的分隔符；c = 0:使用小数点作为整数与小数部分的分隔符。

nnn: 小数部分的位数，有效范围为 0~5，nnn 等于 0 会使转换结果显示格式为不带小数点，当 nnn 值大于 5 时或当指定的输出字符串长度太小无法存储转换的值时，输出字符串用 ASCII 空格字符填充。

下表显示几个使用小数点（c = 0）和小数点右面有一位数（nnn = 001）以及输出字符串长度为 7 个字符（ssss = 0111）格式的值范例。位于 OUT 位置的值是字符串长度。

	OUT	字节 OUT						
		+1	+2	+3	+4	+5	+6	+7
IN=12345.6	7	1	2	3	4	5	.	6
IN=0.0005	7					0	.	0
IN=-3.6544444	7				-	3	.	7

功能说明: 实数至字符串转换指令将实数值 IN 转换为 ASCII 字符串。格式（FMT）指定输出字符串的长度、转换精度以及小数点是显示为逗号还是句点。转换结果放入以 OUT 开始的字符串中。结果字符串长度在格式中指定，可以是 3 至 15 个字符。CTH200 使用的实数格式最多可支持 7 个高位数字。设置 7 个以上高位数字会产生取整错误。

提示: 根据以下规则制定格式的输出字符串:

- 整数不带符号写入输出缓冲区。
- 负数带起始减号 (-) 写入输出缓冲区。
- 小数点左面的起始零（除紧靠小数点的数字外）被压缩。
- 小数点右面的数值被取整，使之符合指定的小数点右面的位数。
- 输出字符串的大小最小必须比小数点右面的位数大三个字节。
- 输出字符串中的数值必须右对齐。

段指令	设置 ENO = 0 的错误条件	0006 间接地址
	影响的特殊内存位	无
指令列表: SEG IN, OUT		
梯形图:		
输入输出参数	数据类型	适用操作数
IN	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *AC, *LD
OUT	字节	VB, IB, QB, MB, SMB, LB, AC, *VD, *AC, SB, *LD

操作数说明:

下图显示了段指令中用到的七段显示器段编码。

(进) LSD	段显示	(出) -gfe dcba		(进) LSD	段显示	(出) -gfe dcba
0	0	0011 1111			8	B
1	1	0000 0110		9	9	0110 0111
2	2	0101 1011		A	A	0111 0111
3	3	0100 1111		B	b	0111 1100
4	4	0110 0110		C	c	0011 1001
5	5	0110 1101		D	d	0101 1110
6	6	0111 1101		E	E	0111 1001
7	7	0000 0111		F	F	0111 0001

功能说明: 段 (SEG) 指令允许您生成照明七段显示段的位格式。

字符串至双整数转换指令	设置 ENO = 0 的错误条件	0006 间接地址 0091 操作数范围 009B 非法指数(指定起始位置值0的字符串操作) SM1.1 溢出或非法值
	影响的特殊内存位	SM1.1(溢出或非法值)
指令列表: STD IN, INDEX, OUT		
梯形图:		
输入输出参数	数据类型	适用操作数
IN	字符串	VB, 常数字符串, LB, *VD, *LD, *AC
OUT	双整数	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC
INDEX	字节	VB, IB, QB, MB, SB, SMB, LB, 常数, AC, *VD, *LD, *AC

ASCII 常数字符串数据类型的格式:

字符串是一系列字符和对应的内存地址, 每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度(即字符数)的整数。如果常数字符串被直接输入程序编辑器或数据块, 那么该字符串必须用双引号字符起始和结束("字符串常数")。

字符串的最大长度是 255 个字节(254 个字符加上长度字节)。下面的内存图显示了字符串数据

类型的格式（一个方格代表一个字节）：

字符串长度	字节 1	字节 2	字节 3	字节 n	字节 254
-------	------	------	------	------	--------

STD 指令用以下形式转换字符串：

[spaces] [+ or -] [digits 0-9]

INDEX 值通常设为 1，从字符串的第一个字符开始转换。可将该值设为其他值，在字符串中的不同点开始转换。当输入字符串包含不属于需要转换数字一部分的文本时，可采用此种方法。例如，如果输入字符串是"value:16000"，您可以将 INDEX 设为值 7，跳过字符串开始的字"value:"。

当达到字符串结束或遇到第一个无效字符时，转换终止。无效字符是数字（0-9）以外的任何字符。每当转换产生一个对于输出值过大的整数值时，则设置溢出错误（SM1.1）。例如，如果输入字符串产生一个大于+2147483647 或小于-2147483648 的值时，STD 指令设置溢出错误。

如果当输入字符串未包含有效值而无法执行转换时，也会设置溢出错误（SM1.1）。例如，如果输入字符串包含"A123"，转换指令会设置 SM1.1（溢出），输出值保持不变。

下表是有效和无效字符串转换为整数/双整数/实数时的举例（INDEX = 1）：

输入字符串	有效性	转换为整数	转换为双整数	转换为实数
"65535"	有效	65535	65535	65535.0
"-0032768"	有效	-32768	-32768	-32768.0
"+32767.999 "	有效	32767	32767	32767.0
"00065535FFFF"	有效	65535	65535	65535.0
"65536"	有效	65536 (SM1.1=1, 溢出)	65536	65536.0
"-002147483648"	有效	-2147483648 (SM1.1=1, 溢出)	-2147483648	-2147483648.0
"+002147483648"	有效	+2147483648 (SM1.1=1, 溢出)	+2147483648 (SM1.1=1, 溢出)	+2147483648.0
"F255"	无效(SM1.1=1, 无效)			
"++255"	无效(SM1.1=1, 无效)			
" "	无效(SM1.1=1, 无效)			

功能说明：字符串至双整数转换指令将字符串数值 IN 转换为双整数值，并将转换结果存储在 OUT 指定的地址中，从字符串的第 INDEX 个字符位置开始转换。

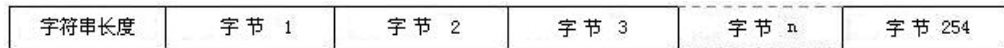
字符串至实数转换指令	设置 ENO = 0 的错误条件	0006 间接地址 0091 操作数范围 009B 非法指数(指定起始位置值0的字符串操作) SM1.1 溢出或非法值
	影响的特殊内存位	SM1.1(溢出或非法值)
指令列表： STR IN, INDEX, OUT		
梯形图：		

输入输出参数	数据类型	适用操作数
IN	字符串	VB, 常数字符串, LB, *VD, *LD, *AC
OUT	实数	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC
INDEX	字节	VB, IB, QB, MB, SB, SMB, LB, 常数, AC, *VD, *LD, *AC

ASCII 常数字符串数据类型的格式:

字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束（"字符串常数"）。

字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的格式(一个方格代表一个字节):



STI 指令用以下形式转换字符串:

[spaces] [+ or -] [digits 0-9]

INDEX 值通常设为 1，从字符串的第一个字符开始转换。可将该值设为其他值，在字符串中的不同点开始转换。当输入字符串包含不属于需要转换数字一部分的文本时，可采用此种方法。例如，如果输入字符串是"value:16000"，您可以将 INDEX 设为值 7，跳过字符串开始的字"value:"。

字符串至实数转换指令不会使用实数的科学计数法或指数形式转换字符串。该指令不会产生溢出错误（SM1.1），但只会将字符串转换为直至指数的实数，然后终止转换。例如，字符串"1.234E6"将转换为实数值 1.234，而不产生错误讯息。

当达到字符串结束或遇到第一个无效字符时，转换终止。无效字符是数字（0-9）以外的任何字符。每当转换产生一个对于输出值过大的整数值时，则设置溢出错误（SM1.1）。

如果当输入字符串未包含有效值而无法执行转换时，也会设置溢出错误（SM1.1）。例如，如果输入字符串包含"A123"，转换指令会设置 SM1.1（溢出），输出值保持不变。

下表是有效和无效字符串转换为整数/双整数/实数时的举例（INDEX = 1）:

输入字符串	有效性	转换为整数	转换为双整数	转换为实数
"65535"	有效	65535	65535	65535.0
"-0032768"	有效	-32768	-32768	-32768.0
"+32767.999 "	有效	32767	32767	32767.0
"00065535FFFF"	有效	65535	65535	65535.0
"65536"	有效	65536 (SM1.1=1, 溢出)	65536	65536.0
"-002147483648"	有效	-2147483648 (SM1.1=1, 溢出)	-2147483648	-2147483648.0
"+002147483648"	有效	+2147483648 (SM1.1=1, 溢出)	+2147483648 (SM1.1=1, 溢出)	+2147483648.0
"F255"	无效(SM1.1=1, 无效)			
"++255"	无效(SM1.1=1, 无效)			
" "	无效(SM1.1=1,			

	无效)			
--	-----	--	--	--

功能说明：字符串至实数转换指令将字符串数值 IN 转换为实数值，并将转换结果存储在 OUT 指定的地址中，从字符串的第 INDEX 个字符位置开始转换。

截断指令	设置 ENO = 0 的错误条件	0006 间接地址 SM1.1 溢出或非法值
	影响的特殊内存位	SM1.1 (溢出)
指令列表: TRUNC IN, OUT		
梯形图:		
输入输出参数	数据类型	适用操作数
IN	实数	VD, ID, QD, MD, SD, SMD, LD, AC, 常数, *VD, *LD, *AC
OUT	双整数	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD

功能说明：截断指令将 32 位实数 IN 转换成 32 位双整数，并将结果放入 OUT 指定的变量中。只有实数的整数部分被转换，小数部分被丢弃。如果输入的实数值过大或无效，将会导致溢出，输出不受影响。

实数至 ASCII 码转换指令	设置 ENO = 0 的错误条件	0006 间接地址 nnn > 5 ssss < 3 ssss < OUT 中的字符数
	影响的特殊内存位	无
指令列表: RTA IN, OUT, FMT		
梯形图:		
输入输出参数	数据类型	适用操作数
IN	实数	VD, ID, QD, MD, SD, SMD, LD, 常数, AC, *VD, *LD, *AC
OUT	字符串	VB, LB, *VD, *AC, *LD
FMT	字节	VB, IB, QB, MB, SB, SMB, LB, 常数, AC, *VD, *LD, *AC

ASCII 常数字符串数据类型的格式:

字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束 ("字符串常数")。

字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的格式(一个方格代表一个字节):

字符串长度	字节 1	字节 2	字节 3	字节 n	字节 254
-------	------	------	------	------	--------

操作数 FMT 定义:

位号	7	6	5	4	3	2	1	0
值	s	s	s	s	c	n	n	n

ssss: 输出字符串长度, 小于 3 无效。

c: 整数与小数部分的分隔符, c = 1:使用逗号作为整数与小数部分的分隔符; c = 0:使用小数点作为整数与小数部分的分隔符。

nnn: 小数部分的位数, 有效范围为 0~5, nnn 等于 0 会使转换结果显示格式为不带小数点, 当 nnn 值大于 5 时或当指定的输出字符串长度太小无法存储转换的值时, 输出字符串用 ASCII 空格字符填充。

下表显示几个使用小数点 (c = 0) 和小数点右面有一位数 (nnn = 001) 以及输出字符串长度为 7 个字符 (ssss = 0111) 格式的值范例。位于 OUT 位置的值是字符串长度。

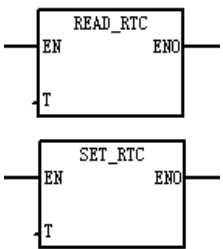
	字节 OUT	字节 OUT					
		+1	+2	+3	+4	+5	+6
IN=12345.6	1	2	3	4	5	.	6
IN=0.0005					0	.	0
IN=-3.6544444				-	3	.	7

功能说明: 实数至 ASCII 码转换指令将实数值 IN 转换为 ASCII 字符。格式 (FMT) 指定输出字符串的长度、转换精度以及小数点是显示为逗号还是句点。转换结果放入以 OUT 开始的字符串中。结果字符串长度在格式中指定, 可以是 3 至 15 个字符。

提示: 根据以下规则制定格式的字符串:

- 整数不带符号写入输出缓冲区。
- 负数带起始减号 (-) 写入输出缓冲区。
- 小数点左面的起始零 (除紧靠小数点的数字外) 被压缩。
- 小数点右面的数值被取整, 使之符合指定的小数点右面的位数。
- 输出字符串的大小最小必须比小数点右面的位数大三个字节。
- 输出字符串中的数值必须右对齐。

实时时钟

读取和设置实时时钟指令	设置 ENO = 0 的错误条件	0006 间接地址 0007 TOD 数据错误(TODW) 000C 不存在时钟
	影响的特殊内存位	SM4.3 (对此时钟有两个同时访问, 非重要错误0007)
指令列表: TODR T TODW T		
梯形图: 		
输入输出参数	数据类型	适用操作数
时钟 T	字节	VB, IB, QB, MB, SMB, SB, LB, *VD, *AC, *LD

操作数说明: 所有日期和时间值必须采用 BCD 格式编码 (例如, 16#97 代表 2002 年)。如下表:

8 个字节时间缓冲区格式 (T)

T 字节	说明	字节数据
------	----	------

0	年 (0-99)	当前年份后两位 (BCD 码)
1	月 (1-12)	当前月份 (BCD 码)
2	日期 (1-31)	当前日期 (BCD 码)
3	小时 (0-23)	当前小时 (BCD 码)
4	分钟 (0-59)	当前分钟 (BCD 码)
5	秒 (0-59)	当前秒 (BCD 码)
6	00	保留字节, 始终设置为00
7	星期几 (1-7)	当前是星期几, 1=星期日 (BCD 码)

长时间掉电或内存丢失后, 实时时钟会被初始化为以下日期和时间:

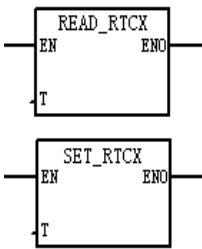
日期: 90 年 1 月 1 日

时间: 00:00:00

星期: 星期日

提示: CTH200 CPU 无法根据日期和星期是否正确, 所以在输入时应当确保日期正确。请勿在主程序和中断程序中使用 TODR/TODW 指令。当另一条 TODR/TODW 指令正在执行时, 若尝试执行中断程序中的 TODR/TODW 指令, 则该指令不会被执行, 同时 SM4.3 被置位。

功能说明: 读取实时时钟指令 (TODR) 从硬件时钟读取当前时间和日期, 并将其写入以地址 T 起始的 8 个字节的缓冲区。设置实时时钟指令 (TODW) 设置当前硬件时钟为以 T 起始的 8 个字节的缓冲区的内容。

读取和设置扩展 实时时钟指令	设置 ENO = 0 的错误条件	0006 间接地址 0007 TOD 数据错误(TODW) 000C 不存在时钟 0091 范围错误
	影响的特殊内存位	SM4.3 (对此时钟有两个同时访问, 非重要错误0007)
指令列表: TODRX T TODWX T		
梯形图: 		
输入输出参数	数据类型	适用操作数
时钟 T	字节	VB, IB, QB, MB, SMB, SB, LB, *VD, *AC, *LD

操作数说明: 所有日期和时间值必须采用 BCD 格式编码 (例如, 16#97 代表 2002 年)。如下表:

8 个字节时间缓冲区格式 (T)

T 字节	说明	字节数据
0	年 (0-99)	当前年份后两位 (BCD 码)
1	月 (1-12)	当前月份 (BCD 码)
2	日期 (1-31)	当前日期 (BCD 码)
3	小时 (0-23)	当前小时 (BCD 码)
4	分钟 (0-59)	当前分钟 (BCD 码)
5	秒 (0-59)	当前秒 (BCD 码)
6	00	保留字节, 始终设置为00
7	星期几 (1-7)	当前是星期几, 1=星期日 (BCD 码)
8	模式(00H-03H, 08H,	修正模式:

	10H-13H, FFH)	00H = 修正已禁用 01H = 欧盟 (相对于 UTC 的时区调整 = 0小时) 02H = 欧盟 (相对于 UTC 的时区调整 = +1小时) 03H = 欧盟 (相对于 UTC 的时区调整 = +2小时) 04H-07H = 保留 08H = 欧盟 (相对于 UTC 的时区调整 = -1小时) 09H-0FH = 保留 10H = 美国 11H = 澳大利亚 12H = 澳大利亚 (塔斯马尼亚) 13H = 新西兰 14H-FDH = 保留 FEH = 保留
9	修正小时数 (0-23)	修正数量, 小时 (BCD 码)
10	修正分钟数 (0-59)	修正数量, 分钟 (BCD 码)
11	开始月份 (1-12)	夏时制的开始月份 (BCD 码)
12	开始日期 (1-31)	夏时制的开始日期 (BCD 码)
13	开始小时 (0-23)	夏时制的开始小时 (BCD 码)
14	开始分钟 (0-59)	夏时制的开始分钟 (BCD 码)
15	结束月份 (1-12)	夏时制的结束月份 (BCD 码)
16	结束日期 (1-31)	夏时制的结束日期 (BCD 码)
17	结束小时 (0-23)	夏时制的结束小时 (BCD 码)
18	结束分钟 (0-59)	夏时制的结束分钟 (BCD 码)

欧盟常规: 在三月最后一个星期日的 UTC 时间凌晨一点将时间向前调一小时。在十月最后一个星期日的 UTC 时间凌晨两点将时间往回调一小时。(当做出修正时, 当地时间因相对于 UTC 的时区调整而不同)

美国常规: 在四月第一个星期日的当地时间凌晨两点将时间向前调一小时。在十月最后一个星期日的当地时间凌晨两点将时间往回调一小时。

澳大利亚常规: 在十月最后一个星期日的当地时间凌晨两点将时间向前调一小时。在三月最后一个星期日的当地时间凌晨三点将时间往回调一小时。

澳大利亚(塔斯马尼亚)常规: 在十月第一个星期日的当地时间凌晨两点将时间向前调一小时。在三月最后一个星期日的当地时间凌晨三点将时间往回调一小时。

新西兰常规: 在十月第一个星期日的当地时间凌晨两点将时间向前调一小时。在三月十五日或之后的第一个星期日的当地时间凌晨三点将时间往回调一小时。

长时间掉电或内存丢失后, 实时时钟会被初始化为以下日期和时间:

日期: 90 年 1 月 1 日

时间: 00:00:00

星期: 星期日

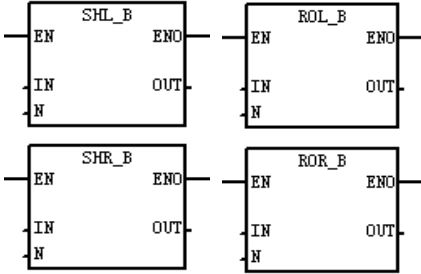
提示: CTH200 CPU 无法根据日期和星期是否正确, 所以在输入时应当确保日期正确。请勿在主程序和中断程序中使用 TODRX/TODWX 指令。当另一条 TODRX/TODWX 指令正在执行时, 若尝试执行中断程序中的 TODRX/TODWX 指令, 则该指令不会被执行, 同时 SM4.3 被置位。

功能说明:

读取扩展实时时钟指令 (TODRX) 从硬件时钟读取当前时间、日期及夏时制配置, 并将其写入以地址 T 起始的 19 个字节的缓冲区。

设置扩展实时时钟指令 (TODWX) 设置当前硬件时钟为以 T 起始的 19 个字节的缓冲区的内容。

移位/循环指令 

字节移位/循环指令	设置 ENO = 0 的错误条件	0006 间接地址
	影响的特殊内存位	SM1.0 如果移位结果是零，则设置零位 SM1.1 为移出的最后一个位设置溢出位
指令列表： SLB OUT, N SRB OUT, N RLB OUT, N RRB OUT, N		
梯形图： 		
输入输出参数	数据类型	适用操作数
IN	字节	VB, IB, QB, MB, SMB, SB, LB, AC, 常数, *VD, *LD,
N	字节	*AC
OUT	字节	VB, IB, QB, MB, SMB, SB, LB, AC, *VD, *LD, *AC

功能说明：

右移字节（SRB）和左移字节（SLB）指令将输入数值（IN）根据移位计数（N）向右或向左移动，并将结果载入输出字节（OUT）。移位指令对每个移出位补 0。如果移位数目（N）大于或等于 8，则数值最多被移位 8 次。如果移位数目大于 0，溢出内存位（SM1.1）采用最后一次移出位的数值。如果移位操作结果为 0，设置 0 内存位（SM1.0）。右移和向左移字节操作不带符号。

循环右移字节（RRB）和循环左移字节（RLB）指令将输入字节数值（IN）向右或向左旋转 N 位，并将结果载入输出字节（OUT）。旋转具有循环性。如果移位数目（N）大于或等于 8，执行旋转之前先对位数（N）进行模数 8 操作，从而使位数在 0 至 7 之间。如果移动位数为 0，则不执行旋转操作。如果执行旋转操作，旋转的最后一位数值被复制至溢出位（SM1.1）。如果移动位数不是 8 的整倍数，旋转出的最后一位数值被复制至溢出内存位（SM1.1）。如果旋转数值为 0，设置 0 内存位（SM1.0）。循环右移和循环左移字节操作不带符号。

双字移位/循环指令	设置 ENO = 0 的错误条件	0006 间接地址
	影响的特殊内存位	SM1.0 如果移位结果是零，则设置零位 SM1.1 为移出的最后一个位设置溢出位
指令列表： SLD OUT, N SRD OUT, N RLD OUT, N RRD OUT, N		
梯形图：		

输入输出参数	数据类型	适用操作数
IN	双字	VD, ID, QD, MD, SD, SMD, LD, AC, HC, 常数, *VD, *LD, *AC
N	字节	VB, IB, QB, MB, SMB, SB, LB, AC, 常数, *VD, *LD, *AC
OUT	双字	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC

功能说明:

右移双字 (SRD) 和左移双字 (SLD) 指令将输入双字数值 (IN) 向右或向左移动 N 位, 并将结果载入输出双字 (OUT)。移位指令对每个移出位补 0。如果移位数目 (N) 大于或等于 32, 则数值最多被移位 32 次。如果移位数目大于 0, 溢出内存位 (SM1.1) 采用最后一次移出位数值。如果移位操作结果为 0, 设置 0 内存位 (SM1.0)。请注意当使用带符号数据类型时, 符号位被移位。

循环右移双字 (RRD) 和循环左移双字 (RLD) 指令将输入双字数值 (IN) 向右或向左旋转 N 位, 并将结果载入输出双字 (OUT)。旋转具有循环性。如果移位数目 (N) 大于或等于 32, 执行旋转之前在移动位数 (N) 上执行模数 32 操作。从而使位数在 0 至 31 之间。如果移动位数为 0, 则不执行旋转操作。如果执行旋转操作, 旋转的最后一位数值被复制至溢出位 (SM1.1)。如果移动位数不是 32 的整倍数, 旋转出的最后一位数值被复制至溢出内存位 (SM1.1)。如果旋转数值为 0, 设置 0 内存位 (SM1.0)。循环右移和循环左移双字操作不带符号。

字移位/循环指令	设置 ENO = 0 的错误条件	0006 间接地址
	影响的特殊内存位	SM1.0 如果移位结果是零, 则设置零位 SM1.1 为移出的最后一个位设置溢出位
指令列表: SLW OUT, N SRW OUT, N RLW OUT, N RRW OUT, N		
梯形图: 		
输入输出参数	数据类型	适用操作数
IN	字	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, 常数, *VD, *LD, *AC
N	字节	VB, IB, QB, MB, SMB, SB, LB, AC, 常数, *VD, *LD, *AC
OUT	字	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *LD, *AC

功能说明:

右移字 (SRW) 和左移位字 (SLW) 指令将输入字 (IN) 数值向右或向左移动 N 位, 并将结果载入输出字 (OUT)。移位指令对每个移出位补 0。如果移位数目 (N) 大于或等于 16, 则数值最多被移位 16 次。如果移位数目大于 0, 溢出内存位 (SM1.1) 采用最后一次移出位数值。如果移位操作结果为 0, 设置 0 内存位 (SM1.0)。请注意当您使用带符号的数据类型时, 符号位被移位。

循环右移字 (RRW) 和循环左移字 (RLW) 指令将输入字数值 (IN) 向右或向左旋转 N 位, 并将结果载入输出字 (OUT)。旋转具有循环性。如果移动位数 (N) 大于或等于 16, 在旋转执行之前的移动位数 (N) 上执行模数 16 操作。从而使移动位数在 0 至 15 之间。如果移动位数为 0, 则不执行旋转操作。如果执行旋转操作, 旋转的最后一位数值被复制至溢出位 (SM1.1)。如果移动位数不是 16 的整倍数, 旋转出的最后一位数值被复制至溢出内存位 (SM1.1)。如果旋转数值为 0, 设置 0 内存位 (SM1.0)。循环右移和循环左移字操作不带符号。

移位寄存器指令	设置 ENO = 0 的错误条件	0006 间接地址 0091 操作数超出范围 0092 计数域错误
	影响的特殊内存位	SM1.1 为移出的最后一个位设置溢出位
指令列表: SHRB DATA, S_BIT, N		
梯形图:		
		
输入输出参数	数据类型	适用操作数
DATA	布尔	I, Q, M, SM, T, C, V, S, L
S_BIT	布尔	I, Q, M, SM, T, C, V, S, L
N	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *LD, *AC

功能说明:

移位寄存器位 (SHRB) 指令将 DATA 数值移入移位寄存器。S_BIT 指定移位寄存器的最低位。N 指定移位寄存器的长度和移位方向 (移位加 = N, 移位减 = -N)。SHRB 指令移出的每个位被放置在溢出内存位 (SM1.1) 中。该指令由最低位 (S_BIT) 和由长度 (N) 指定的位数定义。

使用以下等式计算"移位寄存器"最高位地址 (MSB.b) :

$$MSB.b = [(S_BIT \text{ 字节}) + ([N] - 1 + (S_BIT \text{ 位})) / 8] . [\text{被 } 8 \text{ 除的余数}]$$

例如: 如果 S_BIT 是 V33.4 和 N is 14, 以下计算显示 MSB.b 是 V35.1。

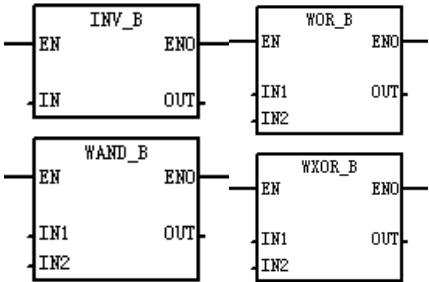
$$\begin{aligned} MSB.b &= V33 + ([14] - 1 + 4) / 8 \\ &= V33 + 17 / 8 \\ &= V33 + 2, \text{ 余数为 } 1 \\ &= V35.1 \end{aligned}$$

在"移位减" (用长度 (N) 的负值表示) 中, 输入数据移入移位寄存器的最高位中, 并移出最低位 (S_BIT)。移出的数据被放置在溢出内存位 (SM1.1) 中。

在"移位加" (用长度 (N) 的正值表示) 中, 输入数据 (DATA) 移入移位寄存器的最高位中 (由 S_BIT 指定), 并移出移位寄存器的最高位。移出的数据被放置在溢出内存位 (SM1.1) 中。

移位寄存器的最大长度为 64 位 (无论正负)。

逻辑计算指令

字节取反、与、或、异或运算指令	设置 ENO = 0 的错误条件	0006 间接地址
	影响的特殊内存位	SM1.0 操作结果等于零
指令列表： INVB OUT ANDB IN1 OUT ORB IN1 OUT XORB IN1 OUT		
梯形图： 		
输入输出参数	数据类型	适用操作数
IN1	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *AC, *LD
IN2	字节	
OUT	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

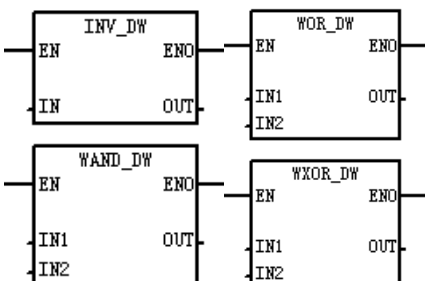
功能说明：

字节取反指令（INVB）对输入字节 IN 执行求补操作，并将结果放入 OUT 指定的地址中。

字节与运算指令（ANDB）对两个输入字节（IN1 和 IN2）的对应位执行与运算操作，并将结果放入 OUT 指定的地址中。

字节或运算指令（ORB）对两个输入字节（IN1 和 IN2）的对应位执行或运算操作，并将结果放入 OUT 指定的地址中。

字节异或运算指令（XORB）对两个输入字节（IN1 和 IN2）的对应位执行异或运算操作，并将结果放入 OUT 指定的地址中。

双字取反、与、或、异或运算指令	设置 ENO = 0 的错误条件	0006 间接地址
	影响的特殊内存位	SM1.0 操作结果等于零
指令列表： INVD OUT ANDD IN1, OUT ORD IN1, OUT XORD IN1, OUT		
梯形图： 		
输入输出参数	数据类型	适用操作数
IN1	双字	VD, ID, QD, MD, SD, SMD, LD, HC, AC, 常数, *VD, *AC, *LD
IN2	双字	
OUT	双字	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD

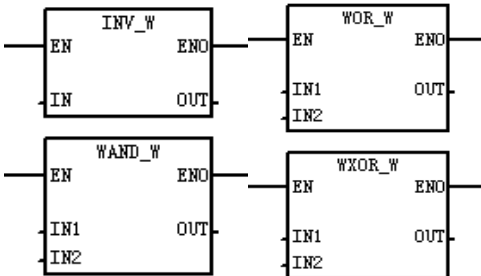
功能说明：

双字取反指令（INVD）对输入双字 IN 执行求补操作，并将结果放入 OUT 指定的地址中。

双字与运算指令（ANDD）对两个输入双字（IN1 和 IN2）的对应位执行与运算操作，并将结果放入 OUT 指定的地址中。

双字或运算指令（ORD）对两个输入双字（IN1 和 IN2）的对应位执行或运算操作，并将结果放入 OUT 指定的地址中。

双字异或运算指令（XORD）对两个输入双字（IN1 和 IN2）的对应位执行异或运算操作，并将结果放入 OUT 指定的地址中。

整数取反、与、或、异或运算指令	设置 ENO = 0 的错误条件	0006 间接地址
	影响的特殊内存位	SM1.0 操作结果等于零
指令列表： INVD OUT ANDW IN1, OUT ORW IN1, OUT XORW IN1, OUT		
梯形图： 		
输入输出参数	数据类型	适用操作数
IN1	字	VW, IW, QW, MW, SW, SMW, T, C, AIW, LW, AC, 常数, *VD, *AC, *LD
IN2	字	
OUT	字	VW, IW, QW, MW, SW, SMW, T, C, LW, AC, *VD, *AC, *LD

功能说明：

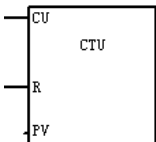
字取反指令（INVD）对输入字 IN 执行求补操作，并将结果放入 OUT 指定的地址中。

字与运算指令（ANDW）对两个输入字（IN1 和 IN2）的对应位执行与运算操作，并将结果放入 OUT 指定的地址中。

字或运算指令（ORW）对两个输入字（IN1 和 IN2）的对应位执行或运算操作，并将结果放入 OUT 指定的地址中。

字异或运算指令（XORW）对两个输入字（IN1 和 IN2）的对应位执行异或运算操作，并将结果放入 OUT 指定的地址中。

计数器指令

加计数器指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表： CTU Cxxx, PV		
梯形图： 		
输入输出参数	数据类型	适用操作数

Cxxx	字	常数 (C0~C255)
CU	布尔	使能位
R	布尔	使能位
PV	整数	VW, IW, QW, MW, SMW, LW, AIW, AC, T, C, 常数, *VD, *AC, *LD, SW

功能说明: 每次加计数输入端 CU 从 OFF 变为 ON 时,加计数器指令 CTU 从当前值计数加 1。当前值 (Cxxx) 大于或等于预设值 (PV) 时,计数器位 (Cxxx) 打开。复原 (R) 输入打开或执行"复原"指令时,计数器被复原。达到最大值(32767)时,计数器停止计数。计数器范围: Cxxx=C0 至 C255 在 STL 中, CTU 复原输入是堆栈顶值, 加计数输入是装载在第二个堆栈位置的值。

<注释> 因为每个计数器有一个当前值, 请勿将相同的计数器号码设置给一个以上计数器。(号码相同的加计数器、加 / 减计数器和减计数器存取相同的当前值)

减计数器指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表: CTD Cxxx, PV		
梯形图:		
输入输出参数	数据类型	适用操作数
Cxxx	字	常数 (C0~C255)
CD	布尔	使能位
R	布尔	使能位
PV	整数	VW, IW, QW, MW, SMW, LW, AIW, AC, T, C, 常数, *VD, *AC, *LD, SW

功能说明: 每次减计数输入端 CD 从 OFF 变为 ON 时,减计数器指令 CTD 从当前值计数减 1。当前值 Cxxx 等于 0 时,计数器位 (Cxxx) 打开。载入输入 (LD) 打开时,计数器复原计数器位 (Cxxx) 并用预设值 (PV) 载入当前值。当前值达到零时,减计数器停止计数,计数器位 Cxxx 打开。计数器范围: Cxxx=C0 至 C255 在 STL 中, CTD 载入输入是堆栈顶值, 而减计数输入是装载在第二个堆栈位置的数值。

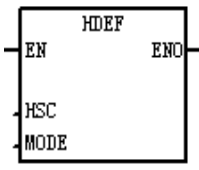
<注释> 因为每个计数器有一个当前值, 请勿将相同的计数器号码设置给一个以上计数器。(号码相同的加计数器、加 / 减计数器和减计数器存取相同的当前值)

加减双向计数器指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表: CTUD Cxxx, PV		
梯形图:		
输入输出参数	数据类型	适用操作数
Cxxx	字	常数(C0~C255)

CU	布尔	使能位
CD	布尔	使能位
R	布尔	使能位
PV	整数	VW, IW, QW, MW, SMW, LW, AIW, AC, T, C, 常数, *VD, *AC, *LD, SW

功能说明：每次加计数输入端 CU 从 OFF 变为 ON 时，加减双向计数器指令 CTUD 从当前值计数加 1。每次减计数输入端 CD 从 OFF 变为 ON 时，加减双向计数器指令 CTUD 从当前值计数减 1。计数器的当前值 Cxxx 保持当前计数。每次执行计数器指令时，预设值 PV 与当前值进行比较。达到最大值（32767），位于加计数输入位置的下一个上升沿使当前值返转为最小值（-32,768）。在达到最小值（-32,768）时，位于减计数输入位置的下一个上升沿使当前计数返转为最大值（32,767）。当当前值 Cxxx 大于或等于预设值 PV 时，计数器位 Cxxx 打开。否则，计数器位关闭。当"复原"（R）输入打开或执行"复原"指令时，计数器被复原。达到 PV 时，CTUD 计数器停止计数。计数器范围：Cxxx=C0 至 C255 在 STL 中，CTUD 复原输入是堆栈顶值，减计数输入是装载在第二个堆栈位置的值，加计数输入是装载在第三个堆栈位置的值。

<注释> 因为每个计数器有一个当前值，请勿将相同的计数器号码设置给一个以上计数器。（号码相同的加计数器、加 / 减计数器和减计数器存取相同的当前值。）

高速计数器定义指令	设置 ENO = 0 的错误条件	0003 输入点冲突 0004 中断中的非法指令 000A HSC 重新定义
	影响的特殊内存位	无
指令列表： HDEF HSC, MODE		
梯形图： 		
输入输出参数	数据类型	适用操作数
HSC	字节	常数（0~4或5）
MODE	字节	常数（0~10或11）

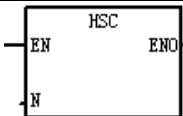
操作数说明：

HSC：高速计数器的 ID，目前 CPU H224/H226L/H226M 支持 4 个高速计数器，不支持 HSC4 和 HCS5，CPU H224X/H226XL/H226XM/H228XL 支持 6 个高速计数器，因此 HSC 的范围为常数 0~5。

MODE：高速计数器的模式，高速计数器有 12 种模式，MODE 的范围为常数 0~11。CPU H224/H226L 仅支持模式 0 和 9，HSC0。

功能说明：高速计数器定义指令 HDEF 选择特定的高速计数器（HSCx）的操作模式。模式选择定义高速计数器的时钟、方向、起始和复原功能。您可以为每台高速计数器使用一条"高速计数器定义"指令。

高速计数器指令	设置 ENO = 0 的错误条件	0001 HSC 在 HDEF 之前 0005 HSC/PLS 同步
	影响的特殊内存位	无
指令列表： HSC N		
梯形图：		

		
输入输出参数	数据类型	适用操作数
N	字	常数 (0~4或5)

操作数说明:

N: 高速计数器的 ID, 目前 CPU H224/H226L/H226M 支持 4 个高速计数器, 不支持 HSC4 和 HCS5, CPU H224X/H226XL/H226XM/H228XL 支持 6 个高速计数器, 因此 N 的范围为常数 0~5.

功能说明: 高速计数器指令 HSC 根据 HSC 特殊内存位的状态配置和控制高速计数器。参数 N 指定高速计数器的号码。高速计数器最多可配置为十二种不同的操作模式。每台计数器在功能受支持的位置有专用时钟、方向控制、复原和起始输入。对于双相计数器, 两个时钟均可按最高速度运行。在正交模式中, 您可以选择一倍 (1x) 或四倍 (4x) 的最高计数速率。所有的计数器按最高速率运行, 而不会相互干扰。

一般而言, 高速计数器被用作驱动式计时器, 改设备有一个安装了增量轴式编码器的轴, 以恒定速度转动。轴式编码器每圈提供一个确定的技术值和一个复位脉冲。来自轴式编码器的时钟和复位脉冲作为高速计数器的输入。

高速计数器装入一组预置值中的第一个值, 当前技术值小于当前预置值时, 希望的输出有效。计数器设置成在当前值等于预置值并且有复位时产生中断。

随着每次当前计数值等于预置值的中断事件发生, 一个新的预置值装入, 然后重新设置下一个输出状态。当出现复位中断事件后, 设置第一个预置值和第一个输出状态, 循环又一次从头开始。

由于中断事件产生的塑料远低于高速计数器的计数速率, 用高速计数器可实现精确控制, 这与 PLC 整个扫描周期关系不大。采用中断的方法允许在简单的状态控制中用独立的中断程序装入新的预置值, 同样, 也可以在一个中断服务程序中处理所有的中断事件。

对于操作模式相同的计数器, 其计数功能的相同的。计数器共有四类:

- 1) 支持内部方向控制的单相计数器
- 2) 支持外部方向控制的单相计数器
- 3) 支持两个时钟输入的双相计数器
- 4) A/B 相正交计数器

<注意>: 并不是所有计数器都能够使用每一种模式, 您可以使用以下类型: 无复位或启动输入, 有复位无启动输入或者复位启动输入。

- 当激活复位输入端时, 计数器清除当前值并一直保持到复位端失效。
- 当激活启动输入端时, 允许计数器计数。启动端失效后, 计数器的当前值保持为常数, 并且忽略时钟事件。
- 如果在启动输入端失效的同时激活了复位信号, 则忽略复位信号, 当前值保持不变。如果在激活复位信号的同时激活了启动输入端, 则当前值会被清除。

在使用高速计数器之前, 应该用 HDEF (高速计数器定义) 指令为计数器选择一种计数模式。是初次扫描存储器位 SM0.1 (仅在第一次扫描周期接通, 之后断开) 来调用一个包含 HDEF 指令的子程序。

定义计数器的模式和输入

使用高速计数器定义指令来定义计数器的模式和输入。

下表给出了与高速计数器相关的时钟、方向控制、复位和启动输入点。同一个输入点不能用于两个

不同的功能，但是任何一个没有被高速计数器当前模式使用的输入点都可以用于其他用途。利润，如果 HSC0 正被用于模式 1，它占用 I0.0 和 I0.2，则 I0.1 可以被边缘中断或 HSC3 占用。



提示

HSC0 的所有模式（模式 12 除外）总是使用 I0.0，HSC4 的所有模式总是使用 I0.3，因此在使用这些计数器时，相应的输入点不能用于其它功能。

高速计数器输入点

模式	中断描述	输入点			
	HSC0	I0.0	I0.1	I0.2	
	HSC1	I0.6	I0.7	I1.0	I1.1
	HSC2	I1.2	I1.3	I1.4	I1.5
	HSC3	I0.1			
	HSC4	I0.3	I0.4	I0.5	
	HSC5	I0.4			
0	支持内部方向控制的单相计数器	时钟			
1		时钟		复位	
2		时钟		复位	启动
3	支持外部方向控制的单相计数器	时钟	方向		
4		时钟	方向	复位	
5		时钟	方向	复位	启动
6	带有增减计数时钟的双相计数器	增时钟	减时钟		
7		增时钟	减时钟	复位	
8		增时钟	减时钟	复位	启动
9	A/B 相正交计数器	时钟 A	时钟 B		
10		时钟 A	时钟 B	复位	
11		时钟 A	时钟 B	复位	启动
12	只有 HSC0 和 HSC3 支持模式 12 HSC0 计数 Q0.0 输出脉冲数 HSC3 计数 Q0.1 输出的脉冲数				

HSC 模式举例

图 H-1 到 H-5 给出了每种模式下计数器功能的时序图。

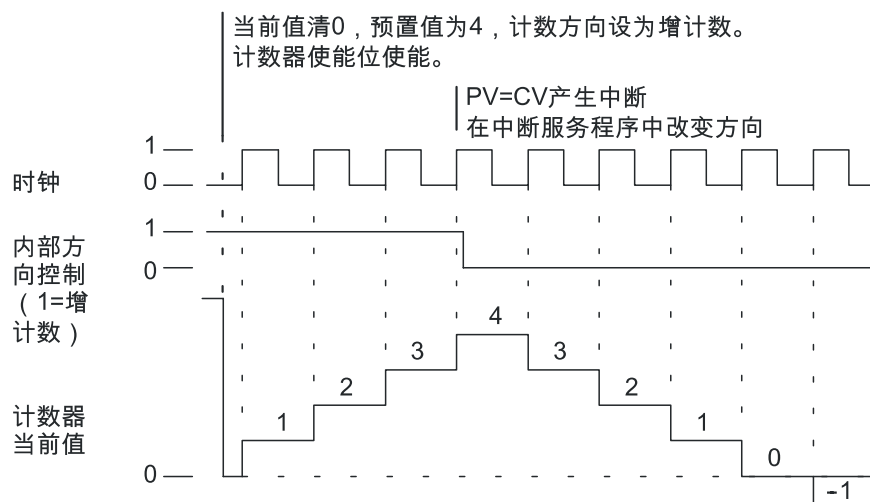


图 H-1 模式 0/1/2

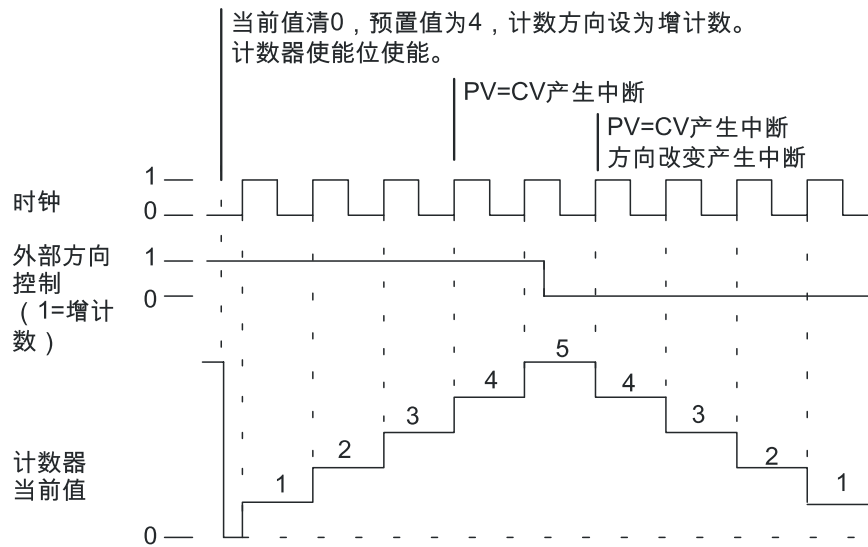


图 H-2 模式 3/4/5

当您使用模式 6/7/8 时，如果增时钟输入的上升沿与减时钟输入的上升沿之间时间间隔小于 0.3 微秒，高速计数器会把这些事件当作同时发生，如果出现这种情况，当前值不变，计数方向指示不变。只要增时钟的上升沿与减时钟输入的上升沿之间时间间隔大于 0.3 微秒，高速计数器分别捕捉每个事件。在以上两种情况下都不会产生错误，计数器保持正确的当前值。

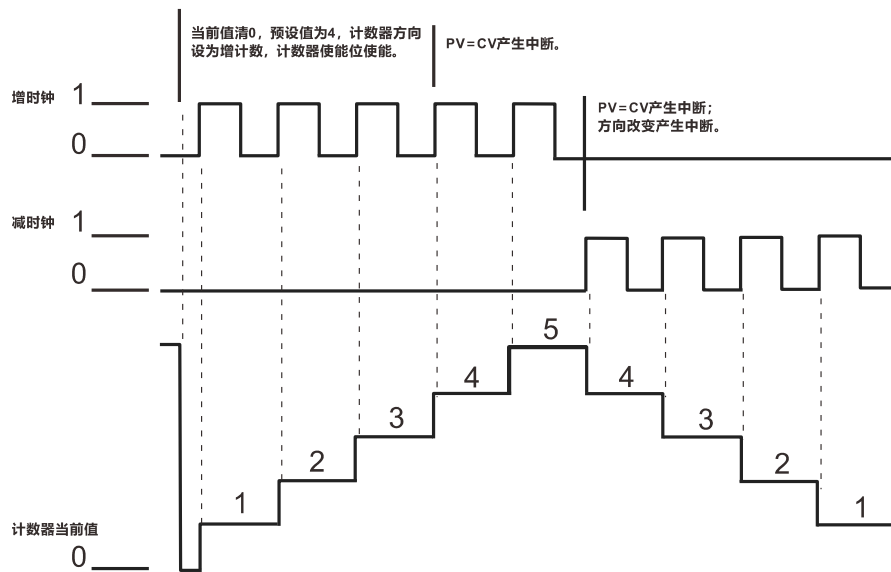


图 H-3 模式 6/7/8

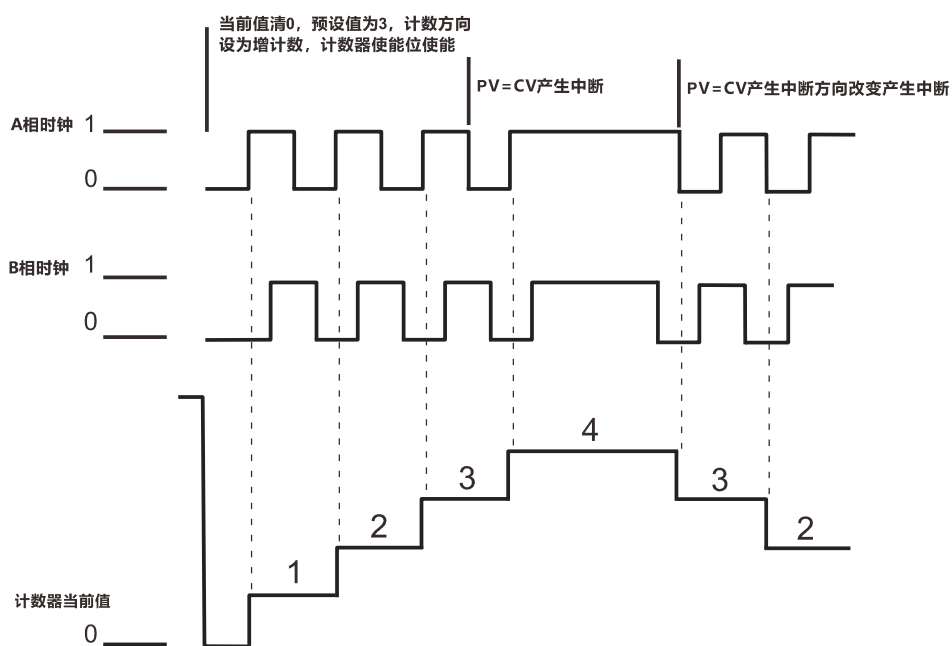


图 H-4 模式 9/10/11（一倍速正交模式）

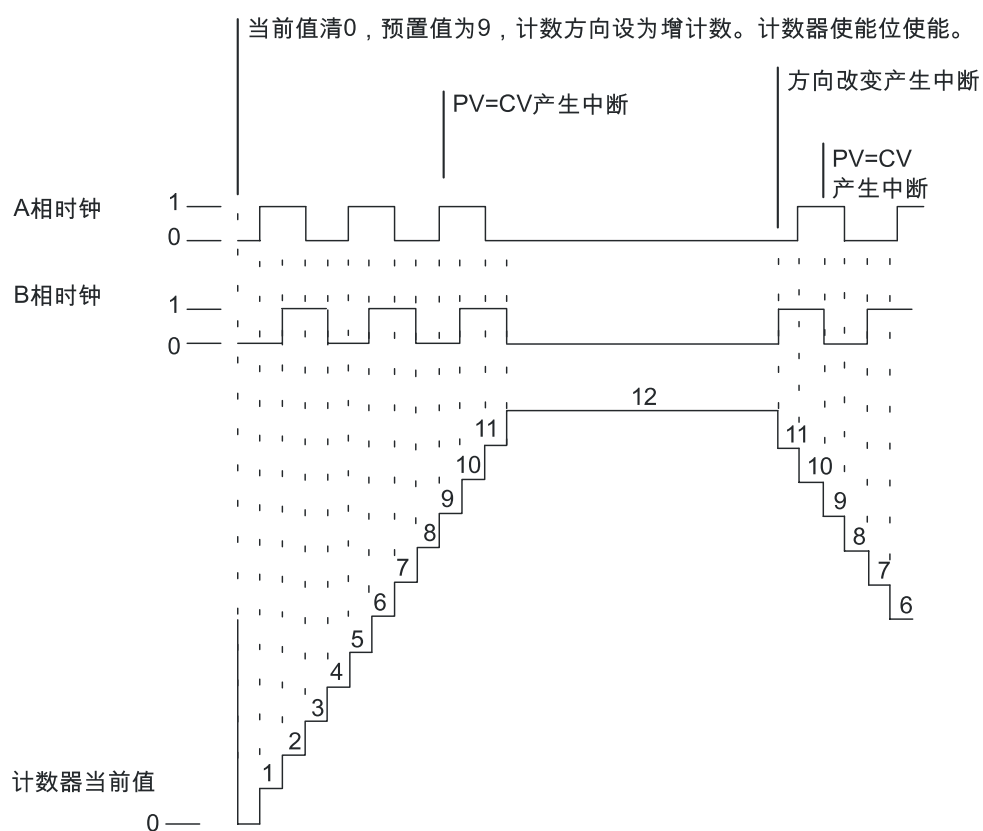


图 H-5 模式 9/10/11 操作实例（四倍速正交模式）

复位和启动操作

如图 H-6 所示，复位和启动操作都是一样使用复位和启动输入的所有模式。在复位和启动输入图中，复位输入和启动输入都被编程设定为高电平有效。

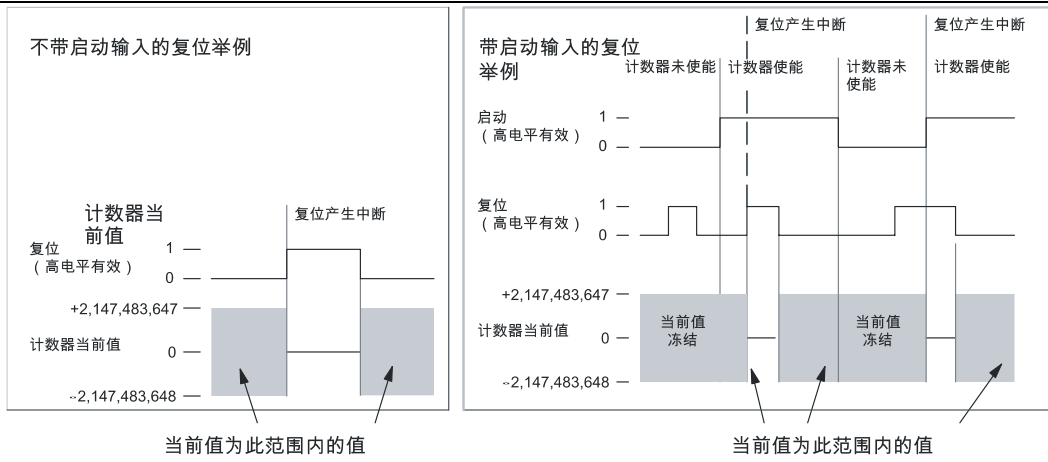


图 H-6 带有或不带启动输入的复位操作举例

对于高速计数器,有三个控制位用于配置复位和启动信号的有效状态以及选择一倍速或四倍速计数模式(仅用于正交计数器)。这些位位于各个计数器的控制字节中,并且只有在 HDEF 指令执行时使用。下表中给出了这些位的具体定义。



提示

在执行 HDEF 指令前,必须把这些控制位设定到希望的状态。否则,计数器对计数模式的选择取默认设置。

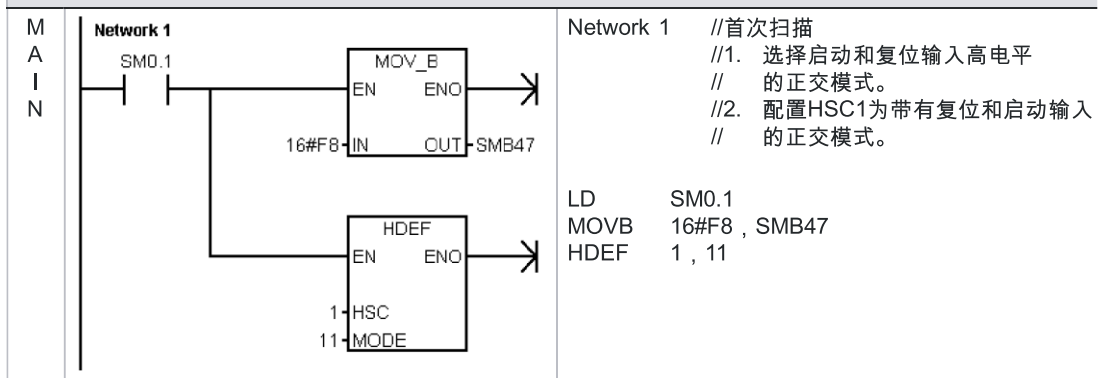
一旦 HDEF 指令执行,您就不能再更改计数器设置,除非先进入 STOP 模式。

复位和启动输入的有效电平以及 1x/4x 控制位

HSC0	HSC1	HSC2	HSC3	描述(仅当 HDEF 执行时使用)
SM37.0	SM47.0	SM57.0	SM147.0	复位有效电平控制位* 0 = 复位高电平有效; 1 = 复位低电平有效
—	SM47.1	SM57.1	—	启动有效电平控制位* 0 = 启动高电平有效; 1 = 启动低电平有效
	SM47.2	SM57.2	SM147.2	正交计数器计数速率选择: 0 = 4x 计数率; 1 = 1x 计数率

*表示默认设置为: 复位输入和启动输入高电平有效, 正交计数率为四倍速(四倍输入时钟频率)。

示例: 高速计数器定义指令



设置控制字节

只有定义了计数器和计数器模式,才能对计数器的动态参数进行编程。每个高速计数器都有一个控制字节,包括使能或禁止计数器、控制计数方向(只对模式 0/1)、装载初始值和装载预置值。

在执行 HSC 指令时,要检验控制字节和相关初始值与预置值。下表详细说明了这些控制位。

HSC0-HSC5 的控制位

HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	中断描述
SM37.3	SM47.3	SM57.3	SM137.3	SM147.3	SM157.3	计数方向控制位： 0 = 减计数；1 = 增计数
SM37.4	SM47.4	SM57.4	SM137.4	SM147.4	SM157.4	向 HSC 中写入计数方向： 0 = 不更新；1 = 更新（计数方向）
SM37.5	SM47.5	SM57.5	SM137.5	SM147.5	SM157.5	向 HSC 中写入预置值： 0 = 不更新；1 = 更新预置值
SM37.6	SM47.6	SM57.6	SM137.6	SM147.6	SM157.6	向 HSC 中写入新的初始值： 0 = 不更新；1 = 更新初始值
SM37.7	SM47.7	SM57.7	SM137.7	SM147.7	SM157.7	HSC 计数使能： 0 = 禁止 HSC；1 = 允许 HSC

设置初始值和预置值

每个高速计数器都有一个 32 位初始值和一个 32 位预置值。初始值和预置值都是符号整数，为了向高速计数器装入新的初始值和预置值，必须先设置控制字节，并且把初始值和预置值存入特殊存储器，然后执行 HSC 指令，从而将新值传送到高速计数器。下表对保存新初始值和预置值的特殊存储器作了说明。

除去控制字节和新的初始值与预置值保存字节外，每个高速计数器的当前值只能使用数据类型 HC（高速计数器当前值）后接下表列出的计数器号(0-5)的格式进行读取。可用读操作直接获取的当前值，但是写操作只能用 HSC 指令来实现。

HSC0-HSC5 的新当前值和新预设值

要装入的值	HSC0	HSC1	HSC2	HSC3	HSC4	HSC5
新当前值	SMD38	SMD48	SMD58	SMD138	SMD148	SMD158
新预设值	SMD42	SMD52	SMD62	SMD142	SMD152	SMD162

HSC0-HSC5 的当前值

值	HSC0	HSC1	HSC2	HSC3	HSC4	HSC5
当前值	HC0	HC1	HC2	HC3	HC4	HC5

高速计数器寻址(HC)

如果要指定高速计数器的地址，访问高速计数器的计数值，要使用存储器类型 HC 和计数器号（比如 HC0）。高速计数器的当前值是只读值，只能以双字（32 位）分配地址，如下图所示：



图 H-7 访问高速计数器的当前值

指定中断

所有计数器模式都支持在 HSC 的当前值等于预设值时产生一个中断事件。使用外部复位端的计数模式支持外部复位中断。除去模式 0、1 和 2 之外，所有计数器模式都支持计数方向改变中断。每种中断条件都可以分别使能或禁止。



注意

使用外部复位中断时，请勿写入初始值或是在改中断服务程序中禁止再次启用高速计数

器，否则将产生致命错误。

状态字节

每个高速计数器都有一个状态字节，其中的状态存储位指出了当前计数方向，当前值是否大于或等于预置值。表 6-12 给出了每个高速计数器状态位的定义。



提示

只有在执行中断服务程序时，状态位才有效。监视高速计数器状态的目的是使其它事件能够产生中断，完成更重要的操作。

HSC0-HSC5 的状态位

HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	中断描述
SM36.0	SM46.0	SM56.0	SM136.0	SM146.0	SM156.0	N/A
SM36.1	SM46.1	SM56.1	SM136.1	SM146.1	SM156.1	N/A
SM36.2	SM46.2	SM56.2	SM136.2	SM146.2	SM156.2	N/A
SM36.3	SM46.3	SM56.3	SM136.3	SM146.3	SM156.3	N/A
SM36.4	SM46.4	SM56.4	SM136.4	SM146.4	SM156.4	N/A
SM36.5	SM46.5	SM56.5	SM136.5	SM146.5	SM156.5	当前计数方向状态位： 0 = 减计数；1 = 增计数
SM36.6	SM46.6	SM56.6	SM136.6	SM146.6	SM156.6	当前值等于预置值状态位： 0 = 不等；1 = 相等
SM36.7	SM46.7	SM56.7	SM136.7	SM146.7	SM156.7	当前值大于预置值状态位： 0 = 小于等于；1 = 大于

高速计数器的初始化示例

下面以 HSC1 为例对高速计数器初始化操作步骤进行描述。在初始化描述中，假定 PLC 已置于 RUN 模式，因此首次扫描标志位设为 true。切记，如果不是这种情况，在 PLC 进入 RUN 模式后，对每一个高速计数器的 HDEF 指令只能执行一次，如果对某个高速计数器第二次执行 HDEF 指令则会引起运行错误，而且不能改变第一次执行 HDEF 指令时对计数器的设置。



提示

下列步骤分别介绍了如何改变计数方向、初始值和预置值，不过您可以在相同操作步骤中对全部或任意参数组合进行设置，只要设置正确的 SMB47 后执行 HDEF 指令即可。

初始化模式 0、1 或 2

HSC1 为内部方向控制的单相增/减计数器（模式 0、1 或 2），初始化步骤如下：

1. 用初次扫描存储器位(SM0.1=1)调用执行初始化操作的子程序。由于采用这样的子程序调用，后续扫描不会再调用这个子程序，从而减少了扫描时间，也提供了一个结构优化的程序。
2. 初始化子程序中，根据所需控制操作对 SMB47 置位，例如：
SMB47=16#F8 产生如下结果：
 允许计数
 写入新的初始值
 写入新的预置值
 将计数方向设为增
 将启动和复位输入设为高电平有效
3. 执行 HDEF 指令时，HSC 输入置 1，MODE 输入置 0（无外部复位或启动）或置 1（有外部

复位但无启动)或置 2 (有外部复位和启动)。

4. 向 SMD48 (双字) 写入所需的初始值 (若写入 0, 则清除)。
5. 向 SMD52 (双字) 写入所需的预置值。
6. 为捕获当前值(CV)等于预置值(PV)中断事件, 编写中断子程序并指定 CV=PV 中断事件 (事件编号 13) 调用该中断子程序。
7. 为捕获外部复位事件, 编写中断子程序, 并指定外部复位中断事件 (事件编号 15) 调用该中断子程序。
8. 执行全局中断使能指令(ENI)来使能 HSC1 中断。
9. 执行 HSC 指令, 使 PLC 对 HSC1 进行编程。
10. 退出子程序。

初始化模式 3、4 或 5

HSC1 为外部方向控制的单相增/减计数器 (模式 3、4 或 5), 初始化步骤如下:

1. 用初次扫描存储器位(SM0.1=1)调用执行初始化操作的子程序。由于采用这样的子程序调用, 后续扫描不会再调用这个子程序, 从而减少了扫描时间, 也提供了一个结构优化的程序。
2. 初始化子程序中, 根据所需控制操作对 SMB47 置位, 例如:

SMB47=16#F8 产生如下结果:

- 允许计数
- 写入新的初始值
- 写入新的预置值
- 将初始计数方向设为增
- 将启动和复位输入设为高电平有效

3. 执行 HDEF 指令时, HSC 输入置 1, MODE 输入置 3 (无外部复位或启动)或置 4 (有外部复位但无启动)或置 5 (有外部复位和启动)。
4. 向 SMD48 (双字) 写入所需的初始值 (若写入 0, 则清除)。
5. 向 SMD52 (双字) 写入所需的预置值。
6. 为捕获当前值(CV)等于预置值(PV)中断事件, 编写中断子程序并指定 CV=PV 中断事件 (事件编号 13) 调用该中断子程序。
7. 为捕获计数方向改变中断事件, 编写中断子程序, 并指定计数方向改变中断事件 (事件编号 14) 调用该中断子程序。
8. 为捕获外部复位事件, 编写中断子程序, 并指定外部复位中断事件 (事件编号 15) 调用该中断子程序。
9. 执行全局中断使能指令(ENI)来使能 HSC1 中断。
10. 执行 HSC 指令, 使 PLC 对 HSC1 进行编程。
11. 退出子程序。

初始化模式 6、7 或 8

HSC1 为具有增/减两种时钟的双相增/减计数器 (模式 6、7 或 8), 初始化步骤如下:

1. 用初次扫描存储器位(SM0.1=1)调用执行初始化操作的子程序。由于采用这样的子程序调用, 后续扫描不会再调用这个子程序, 从而减少了扫描时间, 也提供了一个结构优化的程序。

2. 初始化子程序中，根据所需控制操作对 **SMB47** 置位，例如：

SMB47=16#F8 产生如下结果：
 允许计数
 写入新的初始值
 写入新的预置值
 将初始计数方向设为增
 将启动和复位输入设为高电平有效

3. 执行 **HDEF** 指令时，**HSC** 输入置 **1**，**MODE** 输入置 **6**（无外部复位或启动）或置 **7**（有外部复位但无启动）或置 **8**（有外部复位和启动）。

4. 向 **SMD48**（双字）写入所需的初始值（若写入 **0**，则清除）。

5. 向 **SMD52**（双字）写入所需的预置值。

6. 为捕获当前值(**CV**)等于预置值(**PV**)中断事件，编写中断子程序并指定 **CV=PV** 中断事件（事件编号 **13**）调用该中断子程序。

7. 为捕获计数方向改变中断事件，编写中断子程序，并指定计数方向改变中断事件（事件编号 **14**）调用该中断子程序。

8. 为捕获外部复位事件，编写中断子程序，并指定外部复位中断事件（事件编号 **15**）调用该中断子程序。

9. 执行全局中断使能指令(**ENI**)来使能 **HSC1** 中断。

10. 执行 **HSC** 指令，使 **PLC** 对 **HSC1** 进行编程。

11. 退出子程序。

初始化模式 **9、10 或 11**

HSC1 为 **A/B** 相正交计数器（模式 **9、10 或 11**），初始化步骤如下：

用初次扫描存储器位(**SM0.1=1**)调用执行初始化操作的子程序。由于采用这样的子程序调用，后续扫描不会再调用这个子程序，从而减少了扫描时间，也提供了一个结构优化的程序。

初始化子程序中，根据所需控制操作对 **SMB47** 置位，例如：

1. **1X** 计数方式：

SMB47=16#FC 产生如下结果：
 允许计数
 写入新的初始值
 写入新的预置值
 将初始计数方向设为增
 将启动和复位输入设为高电平有效

2. **4X** 计数方式：

SMB47=16#F8 产生如下结果：
 允许计数
 写入新的初始值
 写入新的预置值
 将 **HSC** 初始计数方向设为增
 将启动和复位输入设为高电平有效

3. 执行 **HDEF** 指令时，**HSC** 输入置 **1**，**MODE** 输入置 **9**（无外部复位或启动）或置 **10**（有外部复位和无启动）或置 **11**（有外部复位和启动）。

4. 向 SMD48（双字）写入所希望的初始值（若写入 0，则清除）。
5. 向 SMD52（双字）写入所希望的预置值。
6. 为了捕获当前值（CV）等于预置值（PV）中断事件，编写中断子程序，并指定 CV=PV 中断事件（事件号 13）调用该中断子程序。参见本章中断一节，以了解中断处理的细节。
7. 为了捕获计数方向改变中断事件，编写中断子程序，并指定计数方向改变中断事件（事件号 14）调用该中断子程序。
8. 为了捕获外部复位事件，编写中断子程序，并指定外部复位中断事件（事件号 15）调用该中断子程序。
9. 执行全局中断允许指令（ENI）来允许 HSC1 中断。
10. 执行 HSC 指令，使 PLC 对 HSC1 编程。
11. 退出子程序。

初始化模式 12

HSC0 为 PTO0 产生的脉冲计数（模式 12），初始化步骤如下：

1. 用初次扫描存储器位（SM0.1=1）调用执行初始化操作的子程序。由于采用了这样的子程序调用，后续扫描不会再调用这个子程序，从而减少了扫描时间，也提供了一个结构优化的程序。
2. 初始化子程序中，根据所希望的控制操作对 SMB37 置数。例如：
SMB47=16#F8 产生如下结果：
 允许计数
 写入新的初始值
 写入新的预置值
 将初始计数方向设为增
 将启动和复位输入设为高电平有效
3. 执行 HDEF 指令时，HSC 输入置 0，MODE 输入置 12。
4. 向 SMD38（双字）写入所希望的初始值（若写入 0，则清除）。
5. 向 SMD42（双字）写入所希望的预置值。
6. 为了捕获当前值（CV）等于预置值（PV）中断事件，编写中断子程序，并指定 CV=PV 中断事件（事件号 13）调用该中断子程序。参看本章中断一节，以了解中断处理的细节。
7. 执行全局中断允许指令（ENI）来允许 HSC1 中断。
8. 执行 HSC 指令，使 PLC 对 HSC0 编程。
9. 退出子程序。

改变模式 0、1、2 或 12 的计数方向

对具有内部方向（控制模式 0、1、2 或 12）的单相计数器 HSC1，改变其计数方向的步骤如下：

1. 向 SMB47 写入所需的计数方向：
SMB47=16#90 允许计数
 置 HSC 计数方向为减。
SMB47=16#98 允许计数
 置 HSC 计数方向为增
2. 执行 HSC 指令，使 PLC 对 HSC1 编程。
SMB47=16#F8 产生如下结果：

允许计数
 写入新的初始值
 写入新的预置值
 将初始计数方向设为增
 将启动和复位输入设为高电平有效

3. 执行 HDEF 指令时, HSC 输入置 1, MODE 输入置 9 (无外部复位或启动) 或置 10 (有外部复位但无启动) 或置 11 (有外部复位和启动)。
4. 向 SMD48 (双字) 写入所需的初始值 (若写入 0, 则清除)。
5. 向 SMD52 (双字) 写入所需的预置值。
6. 为捕获当前值(CV)等于预置值(PV)中断事件, 编写中断子程序并指定 CV=PV 中断事件 (事件编号 13) 调用该中断子程序。
7. 为捕获计数方向改变中断事件, 编写中断子程序, 并指定计数方向改变中断事件 (事件编号 14) 调用该中断子程序。
8. 为捕获外部复位事件, 编写中断子程序, 并指定外部复位中断事件 (事件编号 15) 调用该中断子程序。
9. 执行全局中断使能指令(ENI)来使能 HSC1 中断。
10. 执行 HSC 指令, 使 PLC 对 HSC1 进行编程。
11. 退出子程序。

初始化模式 12

HSC1 为 PTO0 产生的脉冲计数 (模式 12), 初始化步骤如下:

1. 用初次扫描存储器位(SM0.1=1)调用执行初始化操作的子程序。由于采用这样的子程序调用, 后续扫描不会再调用这个子程序, 从而减少了扫描时间, 也提供了一个结构优化的程序。
2. 初始化子程序中, 根据所需控制操作对 SMB37 置位, 例如:
 SMB37=16#F8 产生如下结果:
 允许计数
 写入新的初始值
 写入新的预置值
 将计数方向设为增
 将启动和复位输入设为高电平有效
3. 执行 HDEF 指令时, HSC 输入置 0, MODE 输入置 12。
4. 向 SMD38 (双字) 写入所需的初始值 (若写入 0, 则清除)。
5. 向 SMD42 (双字) 写入所需的预置值。
6. 为捕获当前值(CV)等于预置值(PV)中断事件, 编写中断子程序并指定 CV=PV 中断事件 (事件编号 13) 调用该中断子程序。
7. 执行全局中断使能指令(ENI)来使能 HSC1 中断。
8. 执行 HSC 指令, 使 PLC 对 HSC1 进行编程。
9. 退出子程序。

改变模式 0、1、2 或 12 的计数方向

对具有内部方向 (控制模式 0、1、2 或 12) 的单相计数器 HSC1, 改变其计数方向的步骤如下:

1. 向 SMB47 写入所需的计数方向：

SMB47=16#90 允许计数，将计数方向设为减

SMB47=16#98 允许计数，将计数方向设为增

2. 执行 HSC 指令，使 PLC 对 HSC1 编程

写入新的初始值（任何模式下）

在改变初始值时，迫使计数器处于非工作状态，当计数器被禁止时，它既不计数也不产生中断。

以下步骤描述了如何改变 HSC1 的初始值（任何模式下）：

1. 向 SMB47 写入新的初始值的控制位：

SMB47=16#C0 允许计数，写入新的初始值

2. 向 SMD48（双字）写入所希望的初始值（若写入 0，则清除）。

3. 执行 HSC 指令，使 PLC 对 HSC1 编程。

写入新的预置值（任何模式下）

以下步骤描述了如何改变 HSC1 的预设值（任何模式）：

1. 向 SMB47 写入允许写入新的预置值的控制位：

SMB47=16#A0 允许计数，写入新的预置值

2. 向 SMD52（双字）写入所希望的预置值。

3. 执行 HSC 指令，使 PLC 对 HSC1 编程。

禁止 HSC（任何模式下）

以下步骤描述了如何禁止 HSC1 高速计数器（任何模式）：

1. 写入 SMB47 以禁止计数：

SMB47=16#00 禁止计数

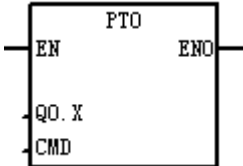
2. 执行 HSC 指令，以禁止计数。

示例：高速计数器指令

<p>M A I N</p>	<p>Network 1 //在首次扫描，调用SBR0</p> <pre>LD SM0.1 CALL SBR0</pre>
<p>S B R 0</p>	<p>Network 1 //在首次扫描，配置HSC1：</p> <ul style="list-style-type: none"> //1. 使能计数器。 // -写初始值。 // -写预置值。 // -设初始方向为增计数。 // -选择启动和复位输入高电平有效。 // -选择4倍速模式。 //2. 配置HSC1为带启动和复位输入的正交模式。 //3. 清除HSC1的初始值。 //4. 置HSC1的预置值为50。 //5. 当HSC1的当前值=预置值时，连接中断服务程序INT_0到事件13。 //6. 全局中断允许。 //7. 执行HSC1 <pre>LD SM0.1 MOVB 16#F8, SMB47 HDEF 1, 11 MOVD +0, SMD48 MOVD +50, SMD52 ATCH INT_0, 13 ENI HSC 1</pre>
<p>I N T 0</p>	<p>Network1 //对HSC1编程</p> <ul style="list-style-type: none"> //1. 清除HSC1的初始值。 //2. 选择写入新的初始值和HSC1使能。 <pre>LD SM0.0 MOVD +0, SMD48 MOVB 16#C0, SMB47 HSC 1</pre>

脉冲输出指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表： PLS N		
梯形图：		
输入输出参数	数据类型	适用操作数
N	字	常数0（Q0.0输出脉冲）或1（Q0.1输出脉冲）

功能说明：脉冲输出（PLS）指令被用于控制在高速输入（Q0.0 和 Q0.1）中提供的"脉冲串输出"（PTO）和"脉宽调制"（PWM）功能。PTO 提供方波（50%占空比）输出，配备周期和脉冲数用户控制功能。PWM 提供连续性变量占空比输出，配备周期和脉宽用户控制功能。

脉冲输出指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表： PTO N, CMD		
梯形图：		
		
输入输出参数	数据类型	适用操作数
N	字节	常数0（Q0.0输出脉冲）或1（Q0.1输出脉冲）
CMD	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC

功能说明：脉冲输出（PLS）指令被用于控制在高速输入（Q0.0 和 Q0.1）中提供的"脉冲串输出"（PTO）和"脉宽调制"（PWM）功能。PTO 提供方波（50%占空比）输出，配备周期和脉冲数用户控制功能。PWM 提供连续性变量占空比输出，配备周期和脉宽用户控制功能。

定时器指令

接通延时定时器指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表： TON Txxx, PT		
梯形图：		
		
输入输出参数	数据类型	适用操作数
IN	布尔	使能位
Txxx	字	常数(0~255)
PT	整数	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, AC, 常数, *VD, *LD, *AC

操作数说明：

定时器类型	分辨率	最大值	定时器 ID
TON/TOF	1ms	32767*1ms	T32, T96
	10ms	32767*10ms	T33-T36, T97-T100, T256~T511
	100ms	32767*100ms	T37-T63, T101-T255
TONR	1ms	32767*1ms	T0, T64
	10ms	32767*10ms	T1-T4, T65-T68
	100ms	32767*100ms	T5-T31, T69-T95

功能说明：接通延时定时器（TON）指令在启用输入为“打开”时，开始计时。当前值（Txxx）大于或等于预设时间（PT）时，定时器位为“打开”。启用输入为“关闭”时，接通延时定时器当前值被清

除。达到预设值后，定时器仍继续计时，达到最大值 32767 时，停止计时。TON、TONR 和 TOF 定时器有三种分辨率。分辨率由定时器 ID 决定，详细情况见操作数说明里的表格。

LAD 和 FBD 定时器选择

- 1) 点击定时器号码域，然后键入定时器号码。
- 2) 倘若您键入的定时器号码无效，则时间基准值继续为"???"。
- 3) 将光标放在定时器框内片刻，即可看到定时器工具提示。请查看此类定时器的有效号码列表。
- 4) 一旦键入有效定时器号码，时间基准值就会在定时器框内显示，例如"10 ms"。

<注释>

- TON 及 TONR 不能共享相同的定时器号码。例如，不能有 TON T32 和 TOF T32。
- 您可以将 TON 用于单间隔计时。
- 可用"复原" (R) 指令复原任何定时器。"复原"指令执行下列操作：
 - 定时器位 = 关闭，定时器当前值 = 0

示例程序：

STL:

ORGANIZATION_BLOCK 主程序：OB1

Network 1

LD I0.0

TON T33, +50 // 如果 I0.0 为打开状态，在 50*10ms 后定时器 T33 超时，定时器的位为 1
// I0.0 打开 T33 启动开始计时，I0.0 关闭 T33 复位并且停止计时

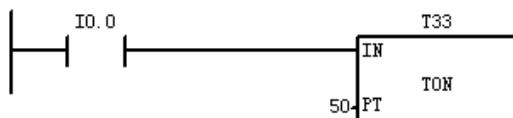
Network 2

LD T33 // 如果 T33 位打开，则点亮 Q0.0，反之关闭 Q0.0
= Q0.0

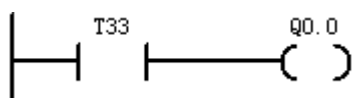
梯形图：

ORGANIZATION_BLOCK 主程序：OB1

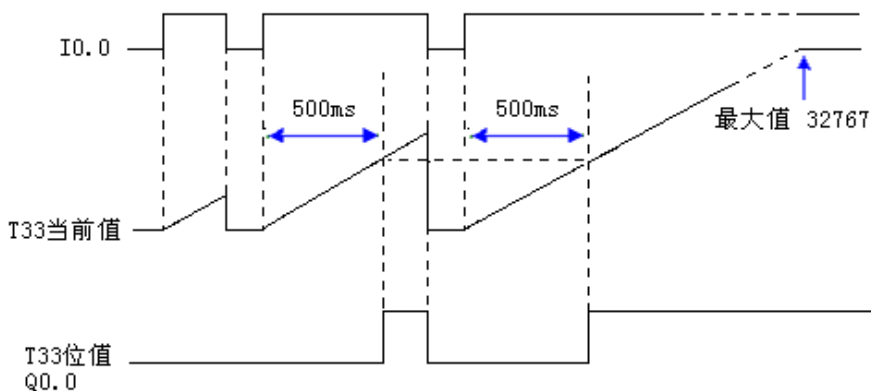
Network 1



Network 2



时序图：



掉电保持接通延时定时器指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表: TONR Txxx, PT		
梯形图:		
		
输入输出参数	数据类型	适用操作数
IN	布尔	使能位
Txxx	字	常数(0~255)
PT	整数	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, AC, 常数, *VD, *LD, *AC

操作数说明:

定时器类型	分辨率	最大值	定时器 ID
TON/TOF	1ms	32767*1ms	T32, T96
	10ms	32767*10ms	T33-T36, T97-T100
	100ms	32767*100ms	T37-T63, T101-T255
TONR	1ms	32767*1ms	T0, T64
	10ms	32767*10ms	T1-T4, T65-T68
	100ms	32767*100ms	T5-T31, T69-T95

功能说明: 掉电保护性接通延时定时器 (TONR) 指令在启用输入为"打开"时, 开始计时。当前值 (Txxx) 大于或等于预设时间 (PT) 时, 计时位为"打开"。当输入为"关闭"时, 保持保留性延迟定时器当前值。您可使用保留性接通延时定时器为多个输入"打开"阶段累计时间。使用"复原"指令 (R) 清除保留性延迟定时器的当前值。达到预设值后, 定时器继续计时, 达到最大值 32767 时, 停止计时。TON、TONR 和 TOF 定时器有三种分辨率。分辨率由定时器 ID 决定, 详细情况见操作数说明里的表格。

LAD 和 FBD 定时器选择

- 1) 点击定时器号码域, 然后键入定时器号码。
- 2) 倘若您键入的定时器号码无效, 则时间基准值继续为"???"。
- 3) 将光标放在定时器框内片刻, 即可看到定时器工具提示。请查看此类定时器的有效号码列表。
- 4) 一旦键入有效定时器号码, 时间基准值就会在定时器框内显示, 例如"10 ms"。

<注释>

您可以将 TONR 用于累积多个计时间隔。

可用"复原" (R) 指令复原任何定时器。"复原"指令执行下列操作:

定时器位 = 关闭, 定时器当前值 = 0

只能用"复原"指令复原 TONR 定时器。

示例程序:**STL:**

ORGANIZATION_BLOCK 主程序: OB1

Network 1

LD I0.0

TONR T0, +500 // 如果 I0.0 为打开状态, 在 500*1ms 后定时器 T0 超时, 定时器的位为 1
// I0.0 打开 T0 启动开始计时, I0.0 关闭定时器当前值保持, 下次 I0.0 打开累加

Network 2

```
LD    T0      // 如果 T0 位打开，则点亮 Q0.0，反之关闭 Q0.0
=     Q0.0
```

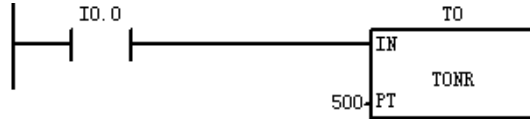
Network 3

```
LD    I1.0    // I1.0 打开时 T0 复位
R     T0, 1
```

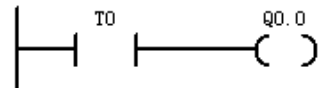
梯形图:

ORGANIZATION_BLOCK 主程序: OB1

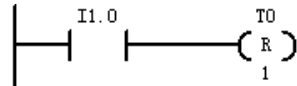
Network 1



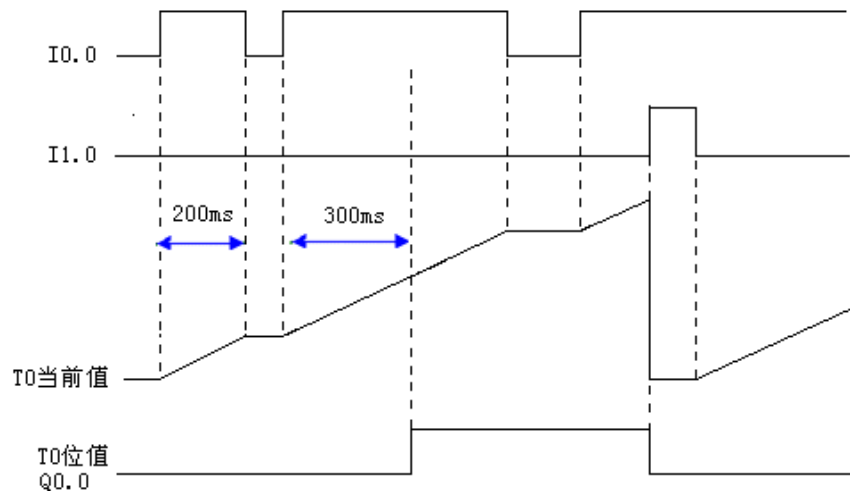
Network 2



Network 3



时序图:



断开延时定时器指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表: TOF Txxx, PT		
梯形图:		
输入输出参数	数据类型	适用操作数
IN	布尔	使能位
Txxx	字	常数(T0~T255)
PT	整数	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, AC, 常数, *VD, *LD, *AC

操作数说明:

定时器类型	分辨率	最大值	定时器 ID
TON/TOF	1ms	32767*1ms	T32, T96
	10ms	32767*10ms	T33-T36, T97-T100
	100ms	32767*100ms	T37-T63, T101-T255
TONR	1ms	32767*1ms	T0, T64
	10ms	32767*10ms	T1-T4, T65-T68
	100ms	32767*100ms	T5-T31, T69-T95

功能说明: 断开延时定时器 (TOF) 用于在输入关闭后, 延迟固定的一段时间再关闭输出。启用输入打开时, 定时器位立即打开, 当前值被设为 0。输入关闭时, 定时器继续计时, 直到消逝的时间达到预设时间。达到预设值后, 定时器位关闭, 当前值停止计时。如果输入关闭的时间短于预设数值, 则定时器位仍保持在打开状态。TOF 指令必须遇到从"打开"至"关闭"的转换才开始计时。如果 TOF 定时器位于 SCR 区域内部, 而且 SCR 区域处于非现用状态, 则当前值被设为 0, 计时器位被关闭, 而且当前值不计。TON、TONR 和 TOF 定时器有三种分辨率。分辨率由定时器 ID 决定, 详细情况见操作数说明里的表格。

LAD 和 FBD 定时器选择

- 1) 点击定时器号码域, 然后键入定时器号码。
- 2) 倘若您键入的定时器号码无效, 则时间基准值继续为"???"。
- 3) 将光标放在定时器框内片刻, 即可看到定时器工具提示。请查看此类定时器的有效号码列表。
- 4) 一旦键入有效定时器号码, 时间基准值就会在定时器框内显示, 例如"10 ms"。

<注释>

TOF 及 TON 不能共享相同的定时器号码。例如, 不能有 TON T32 和 TOF T32。

您可以将 TOF 用于延长时间以超过关闭 (或假) 条件, 例如在电机关闭后使电机冷却。

可用"复原" (R) 指令复原任何定时器。"复原"指令执行下列操作:

定时器位 = 关闭, 定时器当前值 = 0

复原后, TOF 定时器要求启用输入从"打开"转换为"关闭", 以便重新启动。

示例程序:**STL:**

ORGANIZATION_BLOCK 主程序: OB1

Network 1

LD I0.0

TOF T32, +500 // 如果 I0.0 为打开状态, 在 500*1ms 后定时器 T32 超时, 定时器的位为 0
// I0.0 下降沿 T32 启动开始计时, I0.0 上升沿 T32 复位并且停止计时

Network 2

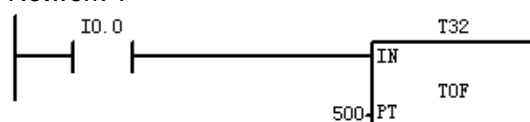
LD T32 // 如果 T32 位打开, 则点亮 Q0.0, 反之关闭 Q0.0

= Q0.0

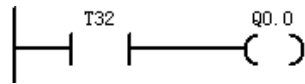
梯形图:

ORGANIZATION_BLOCK 主程序: OB1

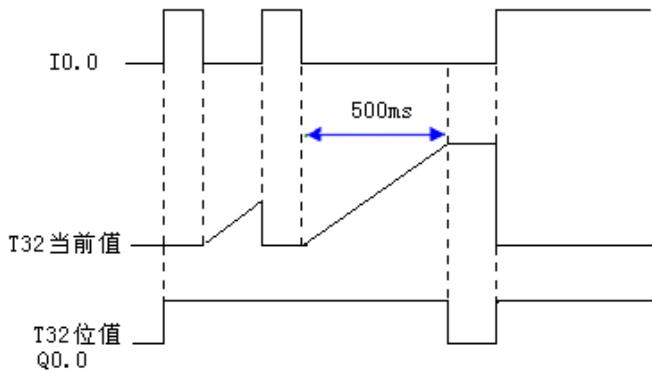
Network 1



Network 2



时序图:



开始间隔时间指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表: BITIM OUT		
梯形图:		
输入输出参数	数据类型	适用操作数
OUT	双字	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

功能说明: 读取内置 1 毫秒计数器的当前值, 并将该值存储于 OUT。双字毫秒值的最大计时间隔为 2 的 32 次方, 即 49.7 日。

取当前微秒值指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表: R_UTIM OUT		
梯形图:		
输入输出参数	数据类型	适用操作数
OUT	双字	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

功能说明: 读取内置 1 毫秒计数器的当前值, 并将该值存储于 OUT, 单位为毫秒, 双字毫秒值的最大计时间隔为 2 的 32 次方, 即 49.7 日。

使用示例: R_UTIM VD0

计算间隔时间指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表: CITIM IN, OUT		
梯形图:		
输入输出参数	数据类型	适用操作数
IN	双字	VD, ID, QD, MD, SMD, SD, LD, HC, AC, *VD, *LD, *AC
OUT	双字	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

功能说明: 计算当前时间与 IN 所提供时间的时差, 将该时差存储于 OUT。双字毫秒值的最大计时间隔为 2 的 32 次方, 即 49.7 日。取决于 BGN_ITIME 指令的执行时间, CAL_ITIME 指令将自动处理发生在最大间隔内的一毫秒定时器翻转。

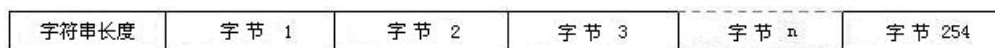
字符串指令

字符串长度指令	设置 ENO = 0 的错误条件	0006 间接地址 0091 操作数范围
	影响的特殊内存位	无
指令列表: SLEN IN, OUT		
梯形图:		
输入输出参数	数据类型	适用操作数
IN	字符串	VB, 常数字符串, LB, *VD, *LD, *AC
OUT	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC

ASCII 常数字符串数据类型的格式:

字符串是一系列字符和对应的内存地址, 每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度(即字符数)的整数。如果常数字符串被直接输入程序编辑器或数据块, 那么该字符串必须用双引号字符起始和结束("字符串常数")。

字符串的最大长度是 255 个字节(254 个字符加上长度字节)。下面的内存图显示了字符串数据类型的格式(一个方格代表一个字节):



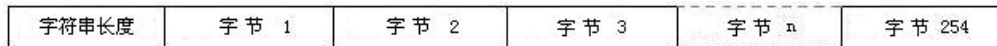
功能说明: 字符串长度指令返回 IN 指定的字符串长度。常数字符串参数最长为 255 个字节。

复制字符串指令	设置 ENO = 0 的错误条件	0006 间接地址 0091 操作数范围
	影响的特殊内存位	无
指令列表: SCPY IN, OUT		
梯形图:		
输入输出参数	数据类型	适用操作数
IN	字符串	VB, 常数字符串, LB, *VD, *LD, *AC
OUT	字符串	VB, *VD, LB, *LD, *AC

ASCII 常数字符串数据类型的格式:

字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束 ("字符串常数")。

字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的格式(一个方格代表一个字节):



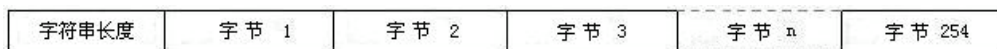
功能说明: 复制字符串指令将 IN 指定的字符串复制至 OUT 指定的字符串。常数字符串参数最长为 255 个字节。

从字符串复制子字符串指令	设置 ENO = 0 的错误条件	0006 间接地址 0091 操作数范围 009B 非法索引（指定起始位置0的字符串操作）
	影响的特殊内存位	无
指令列表: SSCPY IN, INDEX, N, OUT		
梯形图:		
输入输出参数	数据类型	适用操作数
IN	字符串	VB, 常数字符串, LB, *VD, *LD, *AC
INDEX	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *LD, *AC
N	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *LD, *AC
OUT	字符串	VB, *VD, LB, *LD, *AC

ASCII 常数字符串数据类型的格式:

字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束 ("字符串常数")。

字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的格式(一个方格代表一个字节):



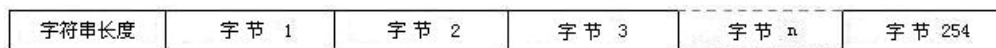
功能说明: 从字符串复制子字符串指令将（从索引（INDEX）开始）IN 指定的具体字符串数目复制至 OUT 指定的字符串。

字符串连接指令	设置 ENO = 0 的错误条件	0006 间接地址 0091 操作数范围
	影响的特殊内存位	无
指令列表: SCAT IN, OUT		
梯形图:		
输入输出参数	数据类型	适用操作数
IN	字符串	VB, 常数字符串, LB, *VD, *LD, *AC
OUT	字符串	VB, *VD, LB, *LD, *AC

ASCII 常数字符串数据类型的格式:

字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束（"字符串常数"）。

字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的格式（一个方格代表一个字节）：



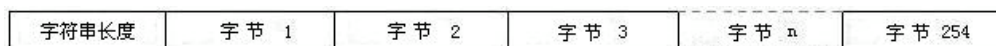
功能说明：字符串连接指令将 IN 指定的字符串附加至 OUT 指定的字符串之后。常数字符串参数最长为 255 个字节。

查找字符串指令	设置 ENO = 0 的错误条件	0006 间接地址 0091 操作数范围 009B 非法索引（指定起始位置0的字符串操作）
	影响的特殊内存位	无
指令列表: SFND IN1, IN2, OUT		
梯形图:		
输入输出参数	数据类型	适用操作数
IN1	字符串	VB, 常数字符串, LB, *VD, *LD, *AC
IN2	字符串	VB, 常数字符串, LB, *VD, *LD, *AC
OUT	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC

ASCII 常数字符串数据类型的格式:

字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束（"字符串常数"）。

字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的格式（一个方格代表一个字节）：



功能说明：在字符串内查找字符串指令在字符串 IN1 中搜索首次出现的字符串 IN2。搜索从 OUT 起始位置开始。如果找到一个与字符串 IN2 完全符合的字符系列，该系列的第一个字符位置被写入 OUT。如果在字符串 IN1 中未找到字符串 IN2，OUT 被设为 0。单个常数字符串最长为 126 个字节，两个常数字符串综合最长为 240 个字节。

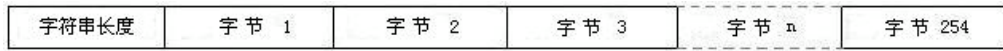
在字符串内查找字符串指令	设置 ENO = 0 的错误条件	0006 间接地址 0091 操作数范围 009B 非法索引（指定起始位置0的字符串操作）
	影响的特殊内存位	无
指令列表: CFND IN1, IN2, OUT		
梯形图:		
输入输出参数	数据类型	适用操作数
IN1	字符串	VB, 常数字符串, LB, *VD, *LD, *AC

IN2	字符串	VB, 常数字符串, LB, *VD, *LD, *AC
OUT	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC


ASCII 常数字符串数据类型的格式:

字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束（"字符串常数"）。

字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的格式（一个方格代表一个字节）：



功能说明: 在字符串中查找字符指令在字符串 IN1 中搜索字符串 IN2 中描述的字符集中的任何字符首次出现的位置。搜索从起始位置 OUT 开始。如果找到一个相符的字符，该字符位置被写入 OUT。如果未找到相符的字符，OUT 被设为 0。单个常数字符串最长为 126 个字节，两个常数字符串综合最长为 240 个字节。

表指令 

内存填充指令	设置 ENO = 0 的错误条件	0006 间接地址 0091 操作数超出范围
	影响的特殊内存位	无
指令列表: FILL IN, OUT, N		
梯形图:		
		
输入输出参数	数据类型	适用操作数
IN	整数	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, 常数, *VD, *LD, *AC
N	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *LD, *AC
OUT	整数	VW, IW, QW, MW, SW, SMW, LW, T, C, AQW, *VD, *LD, *AC

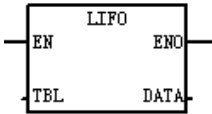
功能说明: 内存填充 (FILL) 指令用包含在地址 IN 中的字值写入 N 个连续字，从地址 OUT 开始。N 的范围是 1 至 255。

先进先出指令	设置 ENO = 0 的错误条件	0006 间接地址 0091 操作数超出范围 SM1.5 空表
	影响的特殊内存位	SM1.5 空表
指令列表: FIFO TBL, DATA		
梯形图:		
		
输入输出参数	数据类型	适用操作数
TBL	字	VW, IW, QW, MW, SW, SMW, LW, T, C, *VD, *LD, *AC

DATA	整数	VW, IW, QW, MW, SW, SMW, LW, AC, T, C, AQW, *VD, *LD, *AC
------	----	---

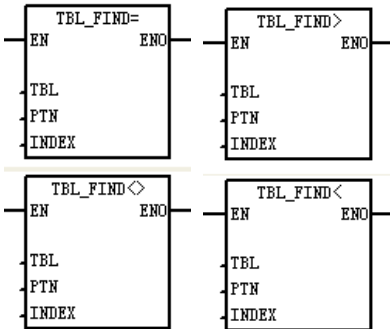
功能说明：先入先出（FIFO）指令通过移除表格（TBL）中的第一个条目，并将数值移至 DATA 指定位置的方法，移动表格中的最早（或第一个）条目。表格中的所有其他条目均向上移动一个位置。每次执行指令时，表格中的条目数减 1。

提示：欲建立表格，首先为最大表条目数建立一个条目。如果您没有这样做，则无法在表格中建立任何条目。此外，所有的表格读取和表格写入指令必须用边缘触发器指令激活。

后进先出指令	设置 ENO = 0 的错误条件	0006 间接地址 0091 操作数超出范围 SM1.5 空表
	影响的特殊内存位	SM1.5 空表
指令列表: LIFO TBL, DATA		
梯形图:		
		
输入输出参数	数据类型	适用操作数
TBL	字	VW, IW, QW, MW, SW, SMW, LW, T, C, *VD, *LD, *AC
DATA	整数	VW, IW, QW, MW, SW, SMW, LW, AC, T, C, AQW, *VD, *LD, *AC

功能说明：后入先出（LIFO）指令将表格中的最新（或最后）一个条目移至输出内存地址，方法是移除表格（TBL）中的最后一个条目，并将数值移至 DATA 指定的位置。每次执行指令时，表格中的条目数减 1。

提示：欲建立表格，首先为最大表条目数建立一个条目。如果您没有这样做，则无法在表格中建立任何条目。此外，所有的表格读取和表格写入指令必须用边缘触发器指令激活。

表格查找指令	设置 ENO = 0 的错误条件	0006 间接地址 0091 操作数超出范围
	影响的特殊内存位	无
指令列表:		
FND= TBL, PIN, INDEX FND<> TBL, PIN, INDEX FND< TBL, PIN, INDEX FND> TBL, PIN, INDEX		
梯形图:		
		
输入输出参数	数据类型	适用操作数
TBL	字	VW, IW, QW, MW, SW, SMW, LW, T, C, *VD, *LD, *AC
PIN	整数	VW, IW, QW, MW, SW, SMW, AIW, LW, T, C, AC, 常

		数, *VD, *LD, *AC
INDEX	字	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *LD, *AC

功能说明: 表格查找 (TBL) 指令在表格 (TBL) 中搜索与某些标准相符的数据。"表格查找"指令搜索表, 从 INDEX 指定的表格条目开始, 寻找与 CMD 定义的搜索标准相匹配的数据数值 (PTN)。命令参数 (CMD) 被指定一个 1 至 4 的数值, 分别代表 =、<>、<, and >。如果找到匹配条目, 则 INDEX 指向表格中的匹配条目。欲查找下一个匹配条目, 再次激活"表格查找"指令之前必须在 INDEX 上加 1。如果未找到匹配条目, INDEX 的数值等于条目计数。一个表格最多可有 100 个条目, 数据项目 (搜索区域) 从 0 排号至最大值 99。

<注释> 当您在用 ATT、LIFO 和 FIFO 指令生成的表格中使用"表格查找"指令时, 条目计数与数据条目直接对应。"表格查找"指令并不要求 ATT、LIFO 和 FIFO 所要求的最大条目数。因此, 如下所示, 您应当将"查找"指令的 SRC 操作数设为一个高于对应的"增加至表格"、"后入先出"或"先入先出"指令 TBL 操作数的一个字地址 (两个字节)。

提示: 欲建立表格, 首先为最大表条目数建立一个条目。如果您没有这样做, 则无法在表格中建立任何条目。此外, 所有的表格读取和表格写入指令必须用边缘触发器指令激活。


追加至表格指令	设置 ENO = 0 的错误条件	0006 间接地址 0091 操作数超出范围 SM1.4 表溢出
	影响的特殊内存位	SM1.4 表溢出
指令列表: ATT DATA, TABLE		
梯形图:		
输入输出参数	数据类型	适用操作数
DATA	整数	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, 常数, *VD, *LD, *AC
TBL	字	VW, IW, QW, MW, SW, SMW, LW, T, C, *VD, , *LD *AC

功能说明: 增加至表格 (ATT) 指令向表格 (TBL) 中加入字值 (DATA)。表格中的第一个数值是表格的最大长度 (TL)。第二个数值是条目计数 (EC), 指定表格中的条目数。新数据被增加至表格中的最后一个条目之后。每次向表格中增加新数据后, 条目计数加 1。表格最多可包含 100 个条目, 不包括指定最大条目数和实际条目数的参数。

提示: 欲建立表格, 首先为最大表条目数建立一个条目。如果您没有这样做, 则无法在表格中建立任何条目。此外, 所有的表格读取和表格写入指令必须用边缘触发器指令激活。

中断指令

从中断程序有条件返回指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表: CRETI		
梯形图:		

		
输入输出参数	数据类型	适用操作数
无	无	无

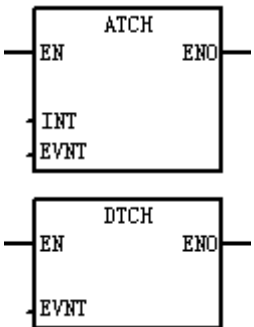
功能说明：从中断指令有条件返回（CRETI）指令可根据先前逻辑条件用于从中断返回。欲增加中断程序，在 Project Manager 主界面点菜单编辑（Edit）>插入（Insert）>Organization Block，。

<注释> MagicWorks PLC 自动为每个中断程序增加一个无条件返回。

您应当优化中断程序，执行具体任务，然后使控制返回主例行程序。保持中断程序简明扼要，可加快执行速度，并且不会长时间延迟其他过程。否则，无法预测的条件会引起主程序控制的装置操作异常。限制在中断程序中不能使用 DISI、ENI、HDEF、LSCR、END 指令。

中断使能和禁止指令	设置 ENO = 0 的错误条件	0004 尝试在中断程序中执行 ENI、DISI 或 HDEF 指令
	影响的特殊内存位	无
指令列表： ENI DISI		
梯形图： 		
输入输出参数	数据类型	适用操作数
无	无	无

功能说明：中断允许（ENI）指令全局性启用所有附加中断事件进程。中断禁止（DISI）指令全局性禁止所有中断事件进程。转换至 RUN（运行）模式时，中断开始时被禁止。一旦进入 RUN（运行）模式，您可以通过执行全局中断允许指令，启用所有中断进程。执行中断禁止指令会禁止处理中断；但是现用中断事件将继续入队等候。

中断连接与分离指令	设置 ENO = 0 的错误条件	0002 配置对 HSC 的输入赋值
	影响的特殊内存位	无
指令列表： ATCH INT, EVENT DTCH EVENT		
梯形图： 		
输入输出参数	数据类型	适用操作数
INT	字节	OBx(x 为常数 2-127)
EVENT	字节	常数 0-33

功能说明:

中断连接 (ATCH) 指令将中断事件 (EVNT) 与中断程序号码 (INT) 相联系, 并启用中断事件。中断分离 (DTCH) 指令取消中断事件 (EVNT) 与所有中断程序之间的关联, 并禁用中断事件。在激活中断程序之前, 必须在中断事件和您希望在事件发生时执行的程序段之间建立关联。使用“中断连接”指令将中断事件 (由中断事件号码指定) 与程序段 (由中断程序号码指定) 联系在一起。您可以将多个中断事件附加在一个中断程序上, 但一个事件不能同时附加在多个中断程序上。当您将一个中断事件附加在一个中断程序上是, 会自动启用中断。如果您用全局禁用中断指令禁用所有的中断, 则每次出现的中断事件均入队等候, 直至使用全局启用中断指令或中断队列溢出重新启用中断。您可以使用“中断分离”指令断开中断事件与中断程序之间的关联, 从而禁用单个中断事件。“中断分离”指令使中断返回至非现用或忽略状态。

清除中断事件指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表: CEVNT EVENT		
梯形图:		
输入输出参数	数据类型	适用操作数
EVENT	字节	常数 0-33

功能说明: 清除中断事件指令会删除中断队列中所有类型为 EVNT 的中断事件。此指令用于清除不必要的中断, 后者可能由假传感器输出暂态造成。

注: 请参考“D 中断事件优先级列表”。

程序控制指令

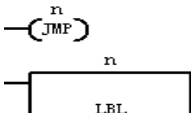
循环指令	设置 ENO = 0 的错误条件	0006 间接地址
	影响的特殊内存位	无
指令列表: FOR INDEX, INIT, FINAL NEXT		
梯形图:		
输入输出参数	数据类型	适用操作数
INDEX	整数	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *LD, *AC
INIT	整数	VW, IW, QW, MW, SW, SMW, T, C, AC, LW, AIW, 常

		数, *VD, *LD, *AC
FINAL	整数	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, AIW, 常数, *VD, *LD, *AC

功能说明: FOR (FOR) 指令执行 FOR 和 NEXT 之间的指令。须指定索引值或当前循环计数 (INDEX)、起始值 (INIT) 和结束值 (FINAL)。NEXT (NEXT) 指令标记 FOR 循环结束, 并将堆栈顶值设为 1。使用 FOR/NEXT 指令描述为指定计数重复循环。每条 FOR 指令要求一个 NEXT 指令。您可以复原 FOR/NEXT 循环 (在 FOR/NEXT 循环中放置一个 FOR/NEXT 循环), 深度可达八。例如, 假定 INIT 值等于 1, FINAL 值等于 10, FOR 与 NEXT 之间的指令被执行 10 次, INDEX 值递增: 1、2、3、...10。如果起始值大于结束值, 则不执行循环。每次执行 FOR 和 NEXT 之间的指令后, INDEX 值递增, 并将结果与结束值比较。如果 INDEX 大于结束值, 循环则终止。

<注释>

- 如果您启用 FOR/NEXT 循环, 则将循环程序, 直至结束反复操作, 除非您从循环内部改变结束值。您可以在 FOR/NEXT 处于循环过程时改变数值。
- 再次启用循环时, 它将初始值复制至索引值 (当前循环次数)。下次被启用时, FOR/NEXT 指令复原。

跳转指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表: JMP n LBL n		
梯形图: 		
输入输出参数	数据类型	适用操作数
n	字节	常数 0~255

功能说明:

跳转至标签 (JMP) 指令对程序中的指定标签 (n) 执行分支操作。跳转接收时, 堆栈顶值始终为逻辑 1。标签 (LBL) 指令标记跳转目的地 (n) 的位置。

您可以在主程序、子程序或中断程序中使用"跳转"指令。"跳转"及其对应的"标签"指令必须始终位于相同的代码段中 (主程序、子程序或中断程序)。您不能从主程序跳转至子程序或中断程序中的标签, 与此相似, 您也不能从子程序或中断程序跳转至该子程序或中断程序之外的标签。您可以在 SCR 段中使用"跳转"指令, 但对应的"标签"指令必须位于相同的 SCR 段内。

顺序控制继电器指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表: LSCR n SCRT n SCRE CSCRE		
梯形图:		

输入输出参数	数据类型	适用操作数
n	布尔	S

功能说明：SCR 指令为您提供一种可自然纳入 LAD、FBD 或 STL 程序的简单、强有力的状态控制编程技术。每当应用程序包含一系列必须重复执行的操作时，SCR 可用于为程序安排结构，以便使之直接与应用程序相对应。因而您能够更快速、更方便地编程和调试应用程序。载入顺序控制继电器（LSCR）指令用指令（N）引用的 S 位数值载入 SCR 和逻辑堆栈。SCR 段被 SCR 堆栈的结果数值激励或取消激励。SCR 堆栈数值被复制至逻辑堆栈的顶端，以便方框或输出线圈可直接与左电源杆连接，无须插入触点。顺序控制继电器转换（SCRT）指令识别要启用的 SCR 位（下一个要设置的 n 位）。当使能位进入线圈或 FBD 方框时，打开引用 n 位，并关闭 LSCR 指令（启用该 SCR 段）的 n 位顺序控制继电器结束（SCRE）指令标记 SCR 段的结束。一旦将电源应用于输入，有条件顺序控制继电器结束（CSCRE）指令即标记 SCR 段结束。CSCRE 只有在 STL 编辑器中才能使用。

<注释> CSCRE 只有在第二代（22x）CPU（从 1.20 版开始）才能使用。

下列说明适用于顺序控制继电器指令。

- "载入 SCR"指令（LSCR）标记 SCR 段的开始，"SCR 结束"指令（SCRE）标记 SCR 段的结束。"载入 SCR"和"SCR 结束"指令之间的所有逻辑执行取决于 S 堆栈数值。"SCR 结束"和下一条"载入 SCR"指令之间的逻辑不取决于 S 堆栈数值。
- "SCR 转换"指令（SCRT）提供一种从现用 SCR 段向另一个 SCR 段转换控制的方法。当"SCR 转换"指令有使能位时，该指令会复原当前现用段的 S 位，并设置被引用段的 S 位。在"SCR 转换"指令执行时，复原现用段的 S 位不会影响 S 堆栈。因此，SCR 段在退出前保持激励状态。
- "有条件 SCR 结束"指令（CSCRE）提供一种无须执行"有条件 SCR 结束"和"SCR 结束"指令之间的指令即可退出现用 SCR 段的方法。"有条件 SCR 结束"指令不会影响任何 S 位，亦不会影响 S 堆栈。

使用 SCR 的限制


- 您不能在一个以上例行程序中使用相同的 S 位。例如，如果您在主程序中使用 S0.1，则不能在子程序中再使用。
- 您不能在 SCR 段中使用 JMP 和 LBL 指令。这表示不允许跳转入或跳转出 SCR 段，亦不允许在 SCR 段内跳转。您可以使用跳转和标签指令在 SCR 段周围跳转。
- 您不能在 SCR 段中使用"结束"指令。

用 SCR 段控制程序流

CTH200 PLC 中的主程序包含每次 PLC 扫描即按顺序执行一次的指令。对于很多应用程序，可能需要以逻辑方式将主程序分为一系列操作步骤，以便反映控制进程中的步骤（例如，一系列机器操作）。

以逻辑方式将程序分为多个步骤的一种方法是使用 SCR 段。通过使用 SCR 段，可以将程序分为一个连续步骤流，或分为可以同时现用的多个步骤流。可以将一个步骤流有条件地分为多个步骤流，或将多个步骤流有条件地重新组合为一个步骤流。

看门狗复位指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无

指令列表: WDR		
梯形图: 		
输入输出参数	数据类型	适用操作数
无	无	无

功能说明:

看门狗复原 (WDR) 指令重新触发 CTH200 CPU 的系统监视程序定时器, 扩展扫描允许使用的时间, 而不会出现看门狗错误。使用"看门狗复原"指令时应当小心。如果使用循环指令阻止扫描完成或严重延迟扫描完成, 下列程序只有在扫描周期完成后才能执行:

- 通讯 (自由端口模式除外)
- I/O 更新 (立即 I/O 除外)
- 强迫更新
- SM 位更新 (不更新 SM0、SM5 至 SM29)
- 运行时间诊断程序
- 10 毫秒和 100 毫秒定时器对于超过 25 秒的扫描不能正确地累计时间
- 用于中断程序时的 STOP (停止) 指令
- 配备离散输出的扩充模块还包括看门狗定时器, 如果模块未被 CTH200 写入, 监视程序定时器会关闭输出。对每个配备离散输出的扩充模块使用立即写入, 在扩展扫描时间期间使正确的输出保持打开。

<注释> 如果您预计扫描时间将超过 500 毫秒, 或者您预计会发生大量中断活动, 可能阻止返回主扫描超过 500 毫秒, 您应当使用 WDR 指令, 重新触发看门狗定时器。每次使用"看门狗复原"指令时, 您还应当使用对每个离散扩充模块中的一个输出字节 (QB) 使用立即写入, 复原每个扩充模块看门狗。如果您使用"看门狗复原"指令允许执行要求很长扫描时间的程序, 将模式开关改变为 STOP (停止) 位置会使 CTH200 在 1.4 秒内转换为 STOP (停止) 模式。

调用指令	设置 ENO = 0 的错误条件	0008 功能嵌套数超过最大
	影响的特殊内存位	无
指令列表: CALL Function Name, parameter 1, ... parameter n (最多允许16个参数)		
梯形图: 		
输入输出参数	数据类型	适用操作数
FCx	功能	功能(x = 0 - 127)
使能位	布尔	使能位
输入参数	布尔, 字节, 字, 整数, 双字, 双整数, 字符串	V, I, Q, M, SM, S, T, C, L, VB, IB, QB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, VW, T, C, IW, QW, MW, SMW, SW, LW, AC, AIW, *VD, *LD, *AC, VD, ID, QD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, &VB, &IB, &QB, &MB, &T, &C, &SB, &AI, &AQ, &SMB, *VD, *LD, *AC, 常数
输入/输出参数	布尔, 字节, 字, 整数, 双字, 双整数	V, I, Q, M, SM, S, T, C, L, VB, IB, QB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, VW, T, C, IW, QW, MW, SMW, SW, LW, AC, AIW, *VD, *LD, *AC, VD, ID, QD, MD, SMD, SD, LD, AC,

		HC, *VD, *LD, *AC
输出参数	布尔, 字节, 字, 整数, 双字, 双整数	V, I, Q, M, SM, S, T, C, L, VB, IB, QB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, VW, T, C, IW, QW, MW, SMW, SW, LW, AC, AIW, *VD, *LD, *AC, VD, ID, QD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC

功能说明: 调用功能 (CALL) 指令将控制转换为功能 (FCx)。您可以使用带参数或不带参数的 "调用功能" 指令。在功能完成执行后, 控制返回至 "调用功能" 之后的指令。每个功能调用的输入 / 输出参数最大限制为 16。如果您尝试下载的程序超过此一限制, 会返回一则错误信息。如果您为功能指定一个符号名, 例如 USR_NAME, 该符号名会出现在指令树的 "功能" 文件夹中。

将参数值指定给功能中的局部内存时应遵守下列规则:

- 1) 参数值指定给局部内存的顺序由 CALL 指定, 参数从 L.0 开始。
- 2) 一至八个连续位参数值被指定给从 Lx.0 开始持续至 Lx.7 的单字节。
- 3) 字节、字和双字数值被指定给局部内存, 位于字节边界 (LBx、LWx 或 LDx) 位置。在带参数的 "调用功能" 指令中, 参数个数、类型与顺序必须与功能局部变量表中定义的变量完全匹配。参数顺序必须以输入参数开始, 其次是输入 / 输出参数, 然后是输出参数。

<注释> 位于指令树中的功能名称的工具提示显示每个参数的名称。

诊断 LED 指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表: DLED IN		
梯形图:		
输入输出参数	数据类型	适用操作数
IN	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *LD, *AC

功能说明:

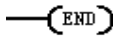
如果输入参数 IN 的数值为零, 则诊断 LED 会被设置为不发光。如果输入参数 IN 的数值大于零, 则诊断 LED 会被设置为发光 (黄色)。标记为 SF/ DIAG 的 CPU 发光二极管 (LED) 能够配置为: 当在系统块内指定的条件为真或当 DIAG_LED 指令以非零 IN 参数得到执行时发黄光。系统块 (配置 LED) 复选框选项: 1) SF/ DIAG LED 会在 CPU 中的某个项被强制时发黄光; 2) SF/ DIAG LED 会在某模块有 I/O 错误时发黄光取消对两个 "配置 LED" 选项的复选, 就会让 DIAG_LED 指令独自控制 SF/ DIAG 黄光。CPU 系统故障 (SF) 用红光表示。

从子程序有条件返回指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表: CRET		
梯形图:		

输入输出参数	数据类型	适用操作数
--------	------	-------

功能说明:

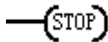
从子程序有条件返回 (CRET) 指令根据前一个逻辑终止子程序。从子程序有条件返回指令是供选用指令。MagicWorks PLC 会自动增加要求使用的从子程序无条件返回指令, 且不在程序编辑器的“子程序 POU”标记显示的程序逻辑中显示。

有条件结束指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表: END		
梯形图: 		
输入输出参数	数据类型	适用操作数

功能说明:

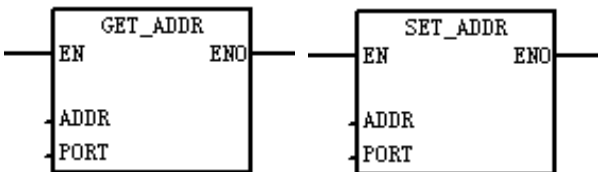
有条件结束 (END) 指令根据前一个逻辑条件终止主用户程序。

<注释> 您可以在主程序中使用“有条件结束”指令, 但不能在子程序或中断程序中使用。MagicWorks PLC 自动在主用户程序中增加无条件结束。

有条件停机指令	设置 ENO = 0 的错误条件	无
	影响的特殊内存位	无
指令列表: STOP		
梯形图: 		
输入输出参数	数据类型	适用操作数

功能说明: 在条件满将 CPU 切换至 STOP 状态。

通信指令

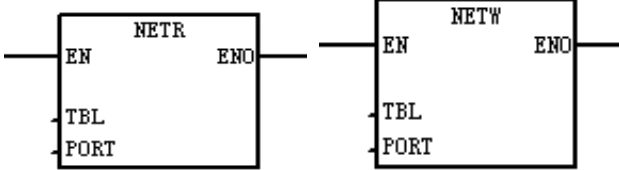
读取设置端口地址指令	设置 ENO = 0 的错误条件	0006 间接地址 0004 尝试在中断程序中执行 SPA(SPA)
	影响的特殊内存位	无
指令列表: GPA ADDR, PORT SPA ADDR, PORT		
梯形图: 		
输入输出参数	数据类型	适用操作数
ADDR(GPA)	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC

ADDR(SPA)	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *LD, *AC
PORT	字节	常数 (0用于 CPU224; 0或1用于 CPU 226/226M/226L)

功能说明:

获得端口地址指令 (GPA) 读取端口号为 PORT 的通信端口站地址, 并将数值置于 ADDR 中指定的地址内。

设置端口地址指令 (SPA) 将端口号为 PORT 的通信端口的站地址设为 ADDR 中指定的数值, 设定的新地址不会被永久性保存。电源重新上电后, 设置的端口会返回至最后的地址 (系统块里设置并下载的地址)。

网络读写指令	设置 ENO = 0 的错误条件	0006 间接地址 表状态字节 E 位为1
	影响的特殊内存位	无
指令列表: NETR TABLE, PORT NETW TABLE, PORT		
梯形图: 		
输入输出参数	数据类型	适用操作数
TBL	字节	VB, MB, *VD, *LD, *AC
PORT	字节	常数 (0用于 CPU224; 0或1用于 CPU 226/226M/226L)

功能说明:

网络读取指令 (NETR) 开始一项通讯操作, 通过指定的端口 PORT 根据表格 TBL 定义从远程设备收集数据。

网络写入指令 (NETW) 开始一项通讯操作, 通过指定的端口 PORT 根据表格 TBL 定义向远程设备写入数据。

网络读写指令最多可从远程站读取或往远程站写入 16 字节信息, 在程序中允许有任意数目的 NETR/NETW 指令, 但在任何时间最多只能有 8 条被激活, 即激活的 NETR 条数加 NETW 条数之和最多为 8 条。

用户还可以使用“网络读取 / 网络写入指令向导”配置通信程序。选择菜单命令“工具”->“NETR/NETW 指令向导”即可启动此向导。

TBL 参数定义

n7	n6	n5	n4	n3	n2	n1	n0
远程站地址							
指针指向							
数据区域							
远程站							
(I/Q/M/V)							
数据长度							

数据字节0
数据字节1
...
数据字节15

n7: 完成 (功能完成) 0 = 未完成 1 = 完成

n6: 现用 (功能入队) 0 = 非现用 1 = 现用

n5: 错误 0 = 无错 1 = 错误

n4: 保留, 始终为 0

远程站址: 存取数据的 PLC 地址

数据指针: 指向 PLC 中数据的间接指针

数据长度: 存取的数据字节数目 (1-16)

接收或传输数据区域: 如下所示, 为数据保留的 1-16 个字节 (由数据长度指定)。

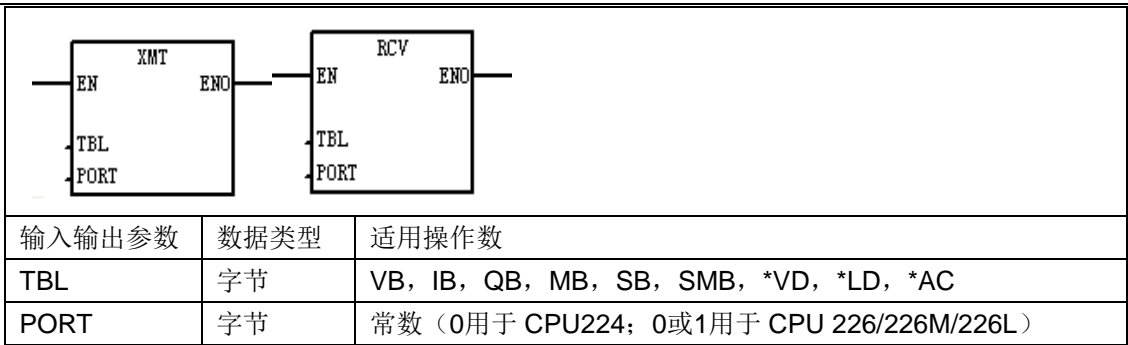
对于 NETR, 该数据区域指执行 NETR 之后从远程站读取的数值存储的区域。

对于 NETW, 该数据区域指执行 NETW 之前发送至远程站的数值存储的区域。

n3~n0: 错误代码, 含义如下:

错误代码	说明
0	无错
1	超时错误; 远程站不应答
2	接收错误; 应答中存在校验、帧或校验和错误
3	脱机错误; 重复站址或故障硬件引起的冲突
4	队列溢出错误; 8 个以上 NETR/NETW 方框被激活
5	违反协议; 未启用 SMB30 中的 PPI+即尝试执行 NETR/NETW
6	非法参数; NETR/NETW 表格包含一个非法或无效数值
7	无资源; 远程站繁忙 (正在上载或下载序列)
8	第 7 层错误; 违反应用程序协议
9	信息错误; 数据地址错误或数据长度不正确

传送接收指令	设置 ENO = 0 的错误条件	0006 间接地址 0009 端口 0 中同时执行 XMT/RCV 000B 端口 1 中同时执行 XMT/RCV 接收状态字节显示接收有错误
	影响的特殊内存位	SMB86 和 SMB186 (接收用户的禁止命令终止接收信息、输入参数错误或无起始或结束条件、收到结束字符、超时、超出最大字符数或奇偶校验错误) SMB3 (自由口接收优先级错误) SMB4 (中断队列溢出)
指令列表: XMT TABLE, PORT RCV TABLE, PORT		
梯形图:		



功能说明:

传送指令 (XMT) 在自由端口模式中使用, 通过通讯端口传送数据。

接收指令 (RCV) 开始或终止“接收信息”服务。您必须指定一个开始条件和一个结束条件, “接收”方框才能操作。通过指定端口 (PORT) 接收的信息存储在数据缓冲区 (TBL) 中。数据缓冲区中的第一个条目指定接收的字节数目。

示例程序:

STL:

ORGANIZATION_BLOCK 主程序: OB1

// 用串口工具写入以 S 开头 E 结束的字符串正确返回, Q0.0/Q1.0 亮

Network 1

```

LD      SM0.1
MOVB   0, VB500
MOVB   0, VB1000

MOVB   16#09, SMB30 //初始化 PORT0 为自由口, 波特率 9600, 8 位数据位, 无校验
MOVB   16#E0, SMB87 //允许接收, 起始结束检测
MOVB   'S', SMB88 //起始字符'S'
MOVB   'E', SMB89 //结束字符'E'
MOVB   200, SMB94 //接收缓冲区为 200 个字节

ATCH   INT0, 23 //启用中断, 接收完成
ATCH   INT1, 9 //发送完成
ENI

RCV    VB100, 0
INTERRUPT_BLOCK INT_0: INT0
    
```

Network 1

```

LD      SM0.0
XMT    VB100, 0 //将收到的字符发送给 Port 0
AENO
=      Q0.0
INCB   VB1000
    
```

Network 2

```

LD      SM0.0
CRETI
INTERRUPT_BLOCK INT_1: INT1
    
```

```
Network 1
LD      SM0.0
RCV     VB100, 0      //发送完成后接收新的字符
AENO
=       Q0.1
INCB    VB500
```

```
Network 2
```

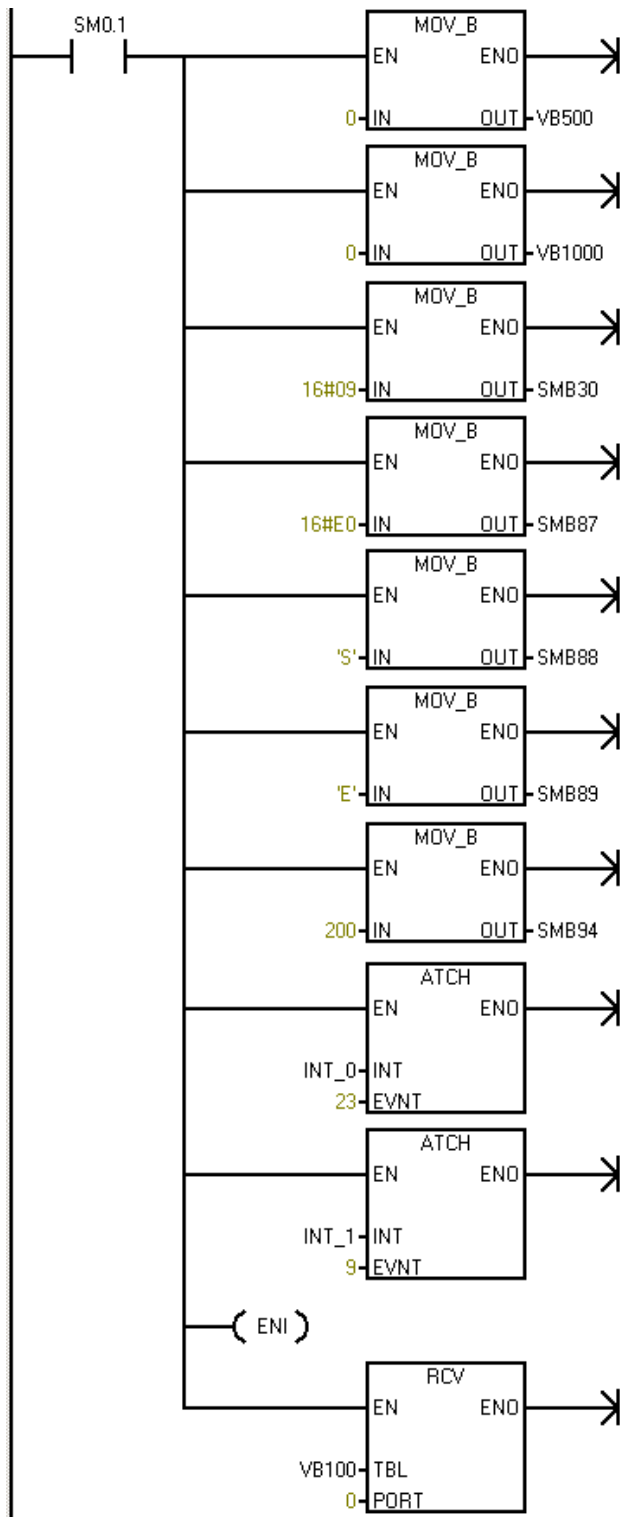
```
LD      SM0.0
CRETI
```

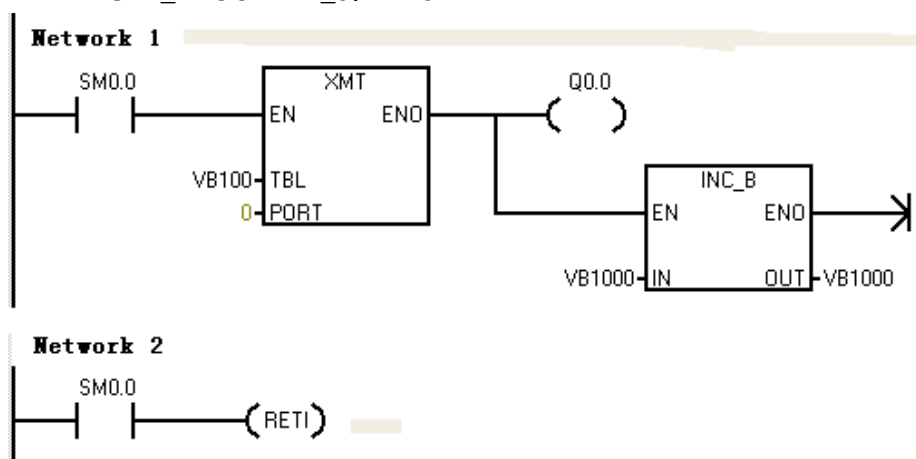
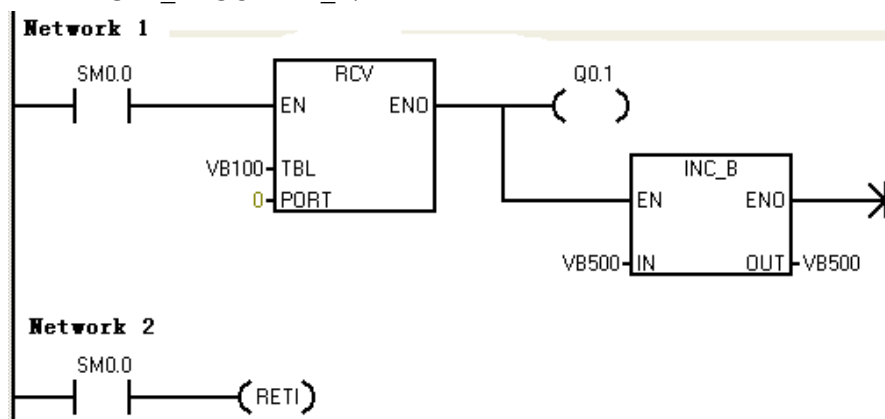
梯形图:

ORGANIZATION_BLOCK 主程序: OB1

// 用串口工具写入以 S 开头 E 结束的字符串正确返回, Q0.0/Q1.0 亮

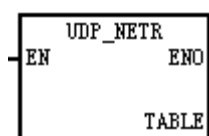
Network 1



INTERRUPT_BLOCK INT_0: INT0**INTERRUPT_BLOCK INT_1: INT1****以太网 PPI 网络读指令 UDP_NETR**

指令列表: UDPNETR EN, TABLE, ENO

梯形图:

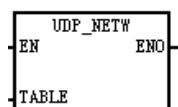


参数	类型	意义
EN	BOOL	UDP_NETR 命令使能位, 上升沿触发
TABLE		包含状态位、IP 地址、指针、数据长度、数据字节, 具体描述参考表“UDP_NETR/UDP_NETW 中参数 TABLE 的描述”。

以太网 PPI 网络写指令 UDP_NETW

指令列表: UDPNETW EN, TABLE, ENO

梯形图:



参数	类型	意义
EN	BOOL	UDP_NETW 命令使能位, 上升沿触发
TABLE		包含状态位、IP 地址、指针、数据长度、数据字节, 具体描述参考 UDP_NETR/UDP_NETW 指令的 TABLE 参数说明。

TABLE 参数使用说明

字节偏移量	7	0			
0	D	A	E	0	错误代码
1	IP 地址第 1 个字节				
2	IP 地址第 1 个字节				
3	IP 地址第 1 个字节				
4	IP 地址第 1 个字节				
5	端口号高字节				
6	端口号低字节				
7	指向远程站指针<I, Q, M, V, DB> (共 4 个字节) 第一个字节				
8	指向远程站指针第 2 个字节				
9	指向远程站指针第 3 个字节				
10	指向远程站指针第 4 个字节				
11	数据长度				
12	数据字节 0				
13	数据字节 1				
...	...				
211	数据字节 199				

D: 命令完成位, 0=未完成、1=完成

A: 现用 (功能入队), 0=非现用、1=现用

E: 错误, 0=无错、1=错误

TABLE 参数中错误代码的描述

错误代码	说明
0	无错误
1	超时错误
2	参数错误
3	未同 Server 建立连接

注释: 支持同时 8 个不同 IP 地址 (即 8 个连接) 的读写, 每个 IP 地址可有 32 条读写数据信息, 每个信息读写的数据长度最大为 200 字节。

以太网 MODBUS 网络读写指令 TCP_MBUS_MSG		
指令列表: MBSNDMSG EN, TABLE, ENO		
梯形图:		
参数	类型	意义
EN		TCP_MBUS_MSG 使能位
TABLE		包含状态位、IP 地址、端口号、读写类型、指针、数据长度、数据字节, 具体描述参考下表 TABLE 参数说明。

TABLE 参数说明

字节偏移量	7	A	E	0	0
0	D	A	E	0	错误代码
1	IP 地址第 1 个字节				
2	IP 地址第 1 个字节				
3	IP 地址第 1 个字节				
4	IP 地址第 1 个字节				
5	端口号高字节				
6	端口号低字节				
7	消息读写类型 0: 读 1: 写				
8	读写地址高字节				
9	读写地址低字节				
10	读写地址高字节				
11	读写地址低字节				
12	数据长度高字节				
13	数据长度低字节				
14	数据字节 0				
15	数据字节 1				
...	...				
253	数据字节 239				

D: 读写功能完成位, 0=未完成、1=完成

A: 现用 (功能入队), 0=非现用、1=现用

E: 错误, 0=无错、1=错误

TABLE 参数中错误代码的描述 (只有在 Done 位为 1 时, 错误代码才有效)

错误代码	说明
0	无错误
1	响应校验错误
2	未用
3	接收超时 (从站无响应)
4	请求参数错误 (slave address, Modbus address, count, RW)
5	Modbus/自由口未使能
6	Modbus 正在忙于其它请求
7	响应错误 (响应不是请求的操作)
8	响应 CRC 校验和错误
101	从站不支持请求的功能
102	从站不支持数据地址
103	从站不支持此种数据类型
104	从站设备故障
105	从站接收了信息, 但是响应被延迟
106	从站忙, 拒绝了该信息
107	从站拒绝了信息
108	从站存储器奇偶错误

注释:

- 同时可有多个从站消息读写，但是每个从站只能同时有一条消息读写。
- TCP_MBUS_MSG 收到应答之后或者出错的时候 Done 才打开。
- TCP MODBUS 最大支持 32 个连接，每条消息一次最大可读写 240 字节。

外部总线读/写指令		
EBUSR TBL		
EBUSW TBL		
梯形图		
输入输出参数	数据类型	适用操作数
TBL	字节	VB, MB, *VD, *LD, *AC

功能说明：从模块读取 nBytes 到 TBL 中指向的地址/将 TBL 中指定的 nBytes 写入到模块。

TABLE 参数表定义：

错误码 ErrCode							
R3	R2	R1	R0	S3	S2	S1	S0
偏移量 Offset0							
偏移量 Offset1							
数据长度 Len							
数据字节 Byte[0]							
数据字节 Byte[1]							
...							
数据字节 Byte[Len-1]							

ErrCode: 0—成功、其它—失败；

R3 R2 R1 R0: 机架号 RackID

S3 S2 S1 S0: 槽号 SlotID

Offset0 Offset1: 模块参数的偏移量，与总线通信应用层协议表中的偏移量对应

Len: 要读取或写入的数据字节数

Byte: 读取或写入的数据

获取外部总线的诊断信息	
EBUSGETDIA TBL	
梯形图	

功能说明：获取模块的诊断信息

TABLE 参数表定义：

错误码 ErrCode							
R3	R2	R1	R0	S3	S2	S1	S0
数据长度 Len							
数据字节 Byte[0]							
数据字节 Byte[1]							
...							
数据字节 Byte[Len-1]							

ErrCode: 0—成功、其它—失败

R3 R2 R1 R0: 机架号 RackID

S3 S2 S1 S0: 槽号 SlotID

Len: 读取的诊断信息的字节数

Byte: 读取的诊断信息内容

外部总线发送命令指令		
EBUSSNDCMD TBL		
梯形图		
输入输出参数	数据类型	适用操作数
TABLE	字节	VB, MB, *VD, *LD, *AC

功能说明: 向指定模块发送一条命令。

TABLE 参数表定义:

D	E	ErrCode5	ErrCode4	ErrCode3	ErrCode2	ErrCode1	ErrCode0
R3	R2	R1	R0	S3	S2	S1	S0
超时时间 (ms) Timeout0							
超时时间 (ms) Timeout1							
输入数据长度 LenIn[0]							
输入数据长度 LenIn[1]							
输入数据字节 ByteIn[0]							
输入数据字节 ByteIn[1]							
...							
输入数据字节 ByteIn[LenIn-1]							
输出数据长度 LenOut[0]							
输出数据长度 LenOut[1]							
输出数据字节 ByteOut[0]							
输出数据字节 ByteOut[1]							
...							
输出数据字节 ByteOut[LenOut-1]							

D: 完成位 (功能完成) 0 = 未完成, 1 = 完成

E: 错误位 0 = 无错, 1 = 错误

ErrCode	错误代码
0x00	没有错误
0x01	发送超时
0x02	模块类型不对
0x03	没有多余的发送控制块
0x04	模块忙
0x05	等待标志超时
0x06	总线标志错误
0x07	软件错误

R3 R2 R1 R0: 机架号 RackID

S3 S2 S1 S0: 槽号 SlotID

Timeout0 Timeout1: 指令执行的超时时间

LenIn: 输入数据的字节数

ByteIn: 输入数据的内容, 格式根据具体模块来定


LenOut: 输出数据的字节数

ByteOut: 输出数据的内容, 格式根据具体模块来定

T 编程电缆

带有 USB-485 接口的合信编程电缆用于 CTH200 CPU 编程、上下载、监控, 使用时 USB 端连接 PC/PG, 485 端连接 CPU 的 PPI 口。

表 U-1 编程电缆的物理特性

	项目	内容
	订货号	CTS7 191-USB30
	支持操作系统	Windows2000/Windows XP/Windows 7
	波特率	300bps ~ 1Mbps 标准波特率自适应
	工作温度	0 ~ +55 °C
	电缆长度	2.5 米
	每台 PC 支持电缆数	1 根



提示

CTH200 系列 CPU 支持我司现有的编程电缆: 191-USB30, 每根编程电缆都有其对应的驱动文件, 请于合信官网下载对您所需的驱动文件: <http://www.co-trust.com>

电缆外形结构:

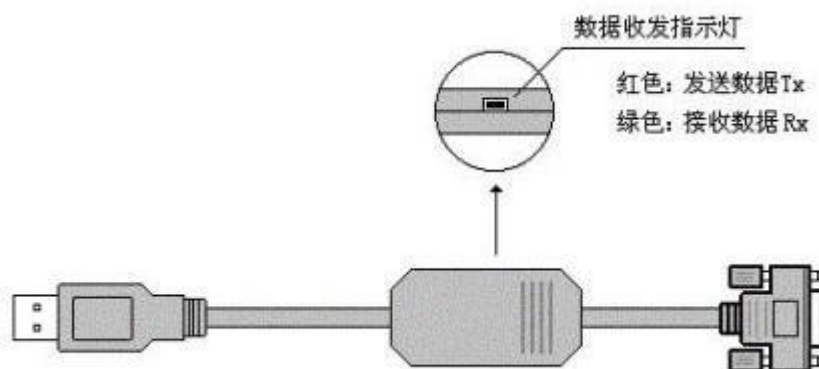
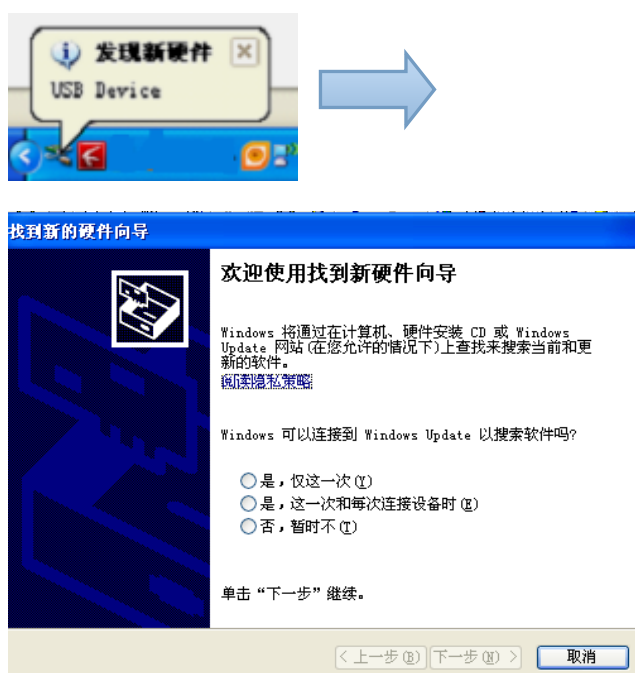


图 U-1 编程电缆外形结构

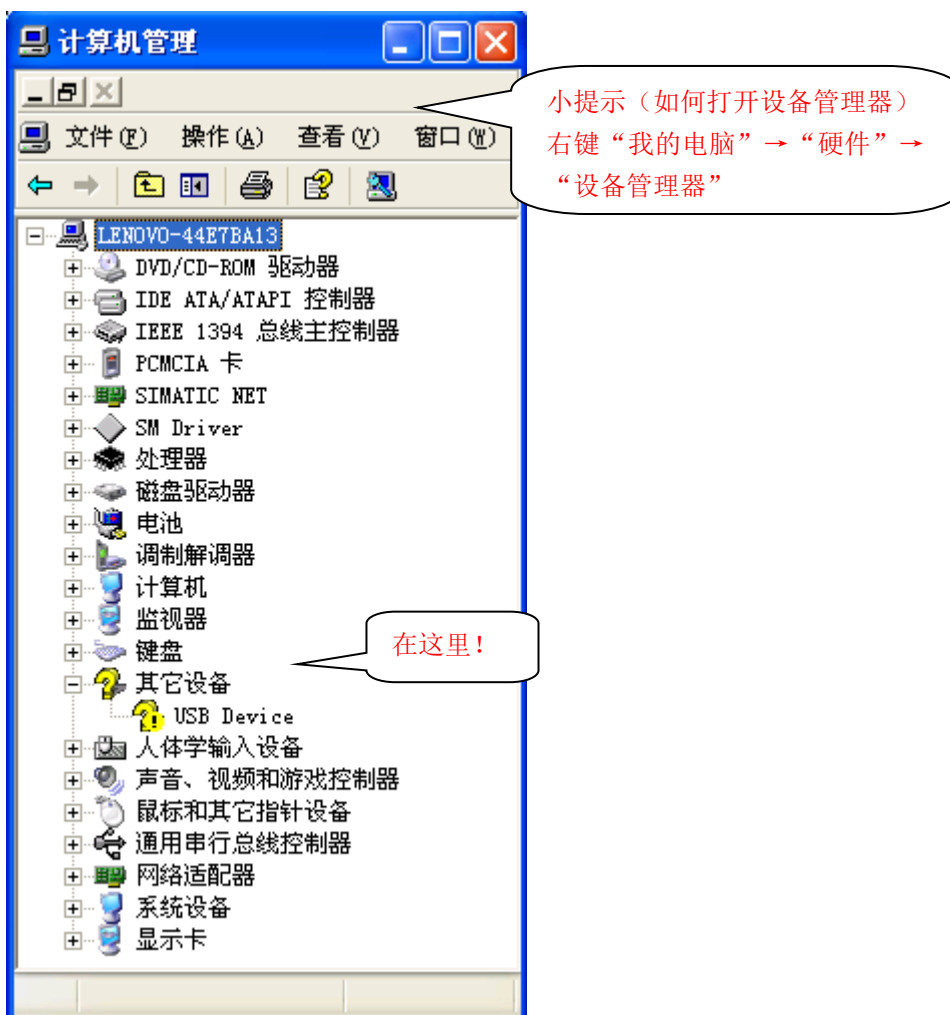
T.1 安装驱动

【第 1 步】 将您的驱动光盘放入电脑或将 COTRUST 网站下载的驱动文件存入您的电脑，然后将您所购买的 CTS7 191-USBX0 电缆插入电脑（台式或笔记本电脑），电脑就会发现新硬件，及弹出安装向导。

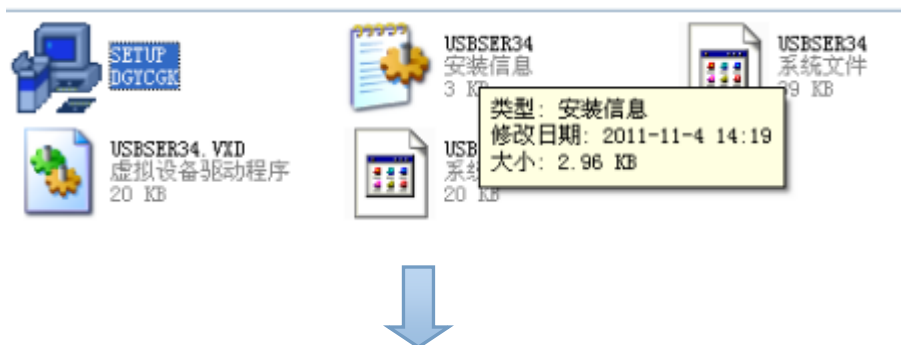
注意：第 2~5 步只是进行“通用串行总线控制器”里的驱动，如果有这个驱动的话就无需安装，最好重新安装一次。

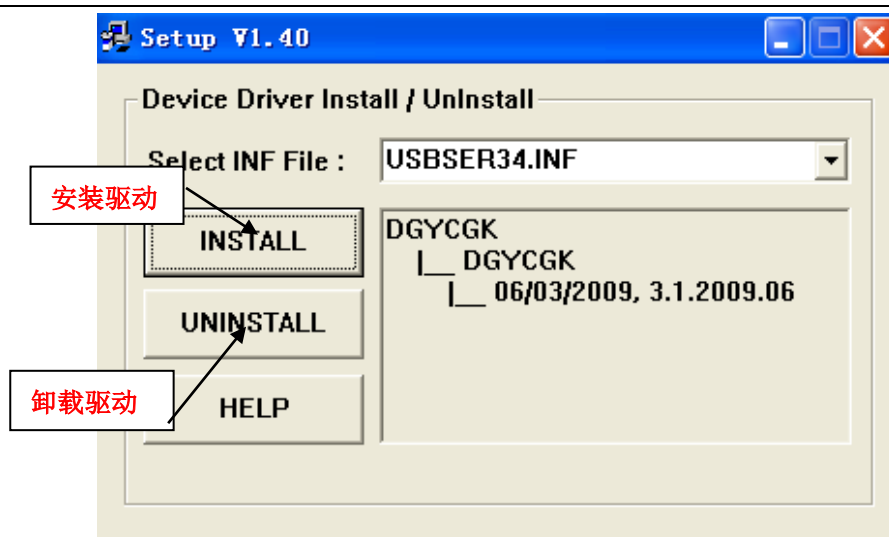



【第 2 步】 如果它不会出现上面两种现象，您可以在您的**设备管理器**里看到端口里黄色的感叹号。



【第 3 步】 以上二种情况均表明您的电缆已被电脑识别，您需要安装驱动，打开光盘或驱动文件中的“RS-232 Driver”文件夹，点击“SETUP.EXE”安装，即会出现“图 2”所示。





【第4步】点击  即可，等到出现以下显示时，您的驱动已安装成功！

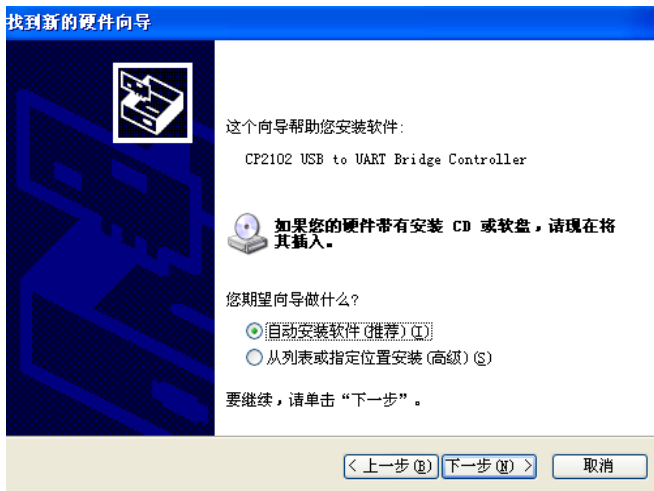
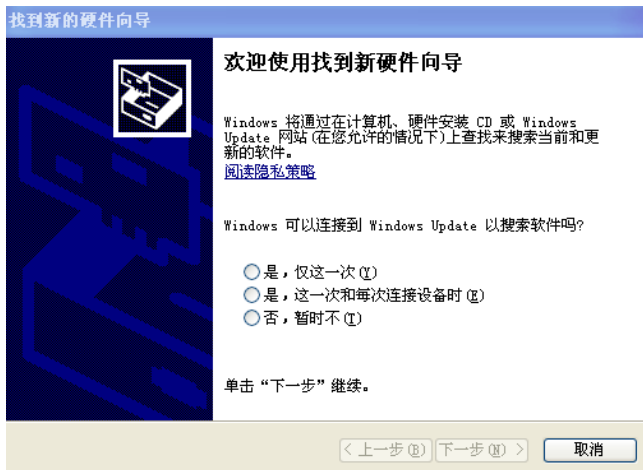


【第五步】点击“确定”即可。安装到这里并没有真正的完成安装，只是进行了“通用串行总线控制器”里的驱动，还需要“端口”里的驱动安装。

【第6步】进入设备管理器，点击右键“更新驱动程序”。



【第7步】这里选择“否，暂时不” → “从列表或指定的位置安装高级”。



【第 8 步】“下一步”，把“搜索光盘”和“搜索指定位置”都打上勾，再点击“浏览”找到光盘相应的"RS485Drivers" 驱动，点击确定即可。





【第 9 步】 在安装过程中会出现以上画面，点“仍然继续”即可。



【第 10 步】 出现以上图片表示第一步安装成功，点击“完成”即可。此时真正完成 USB-PPI 的安装。



T.2 更改 COM 端口号

有些早期的应用软件支持的串口数量是有限的，如某 PLC 编程软件支持的串口号为 COM1～COM4，由于安装 USB 编程电缆之前已安装了其它的 USB 转换设备，安装 USB 编程电缆驱动程序后在设备管理器中显示该电缆地址为 COM5，这样一来编程软件就无法访问该 COM 口。这时候就需改变编程电缆的 COM 口号，方法如下：

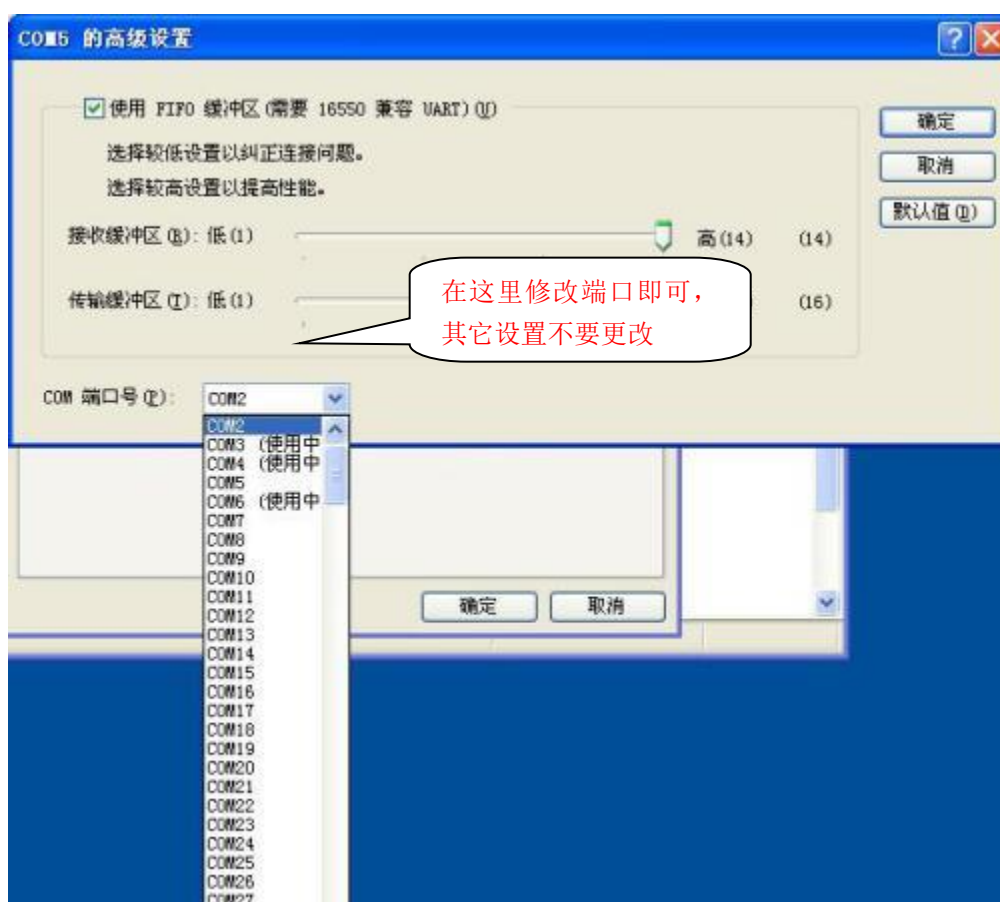
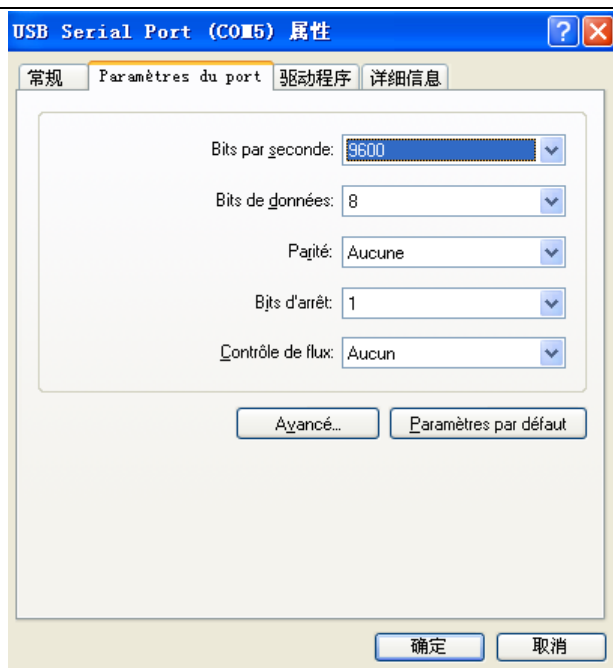
1、如何找到端口位置

在 Windows2000/XP 系统中可直接更改 COM 口号，首先安装好以上驱动后，点击桌面“我的电脑”右键属性 → “硬件” → “设备管理器”→即可看到“端口”。

2、如果修改端口

1 这里以 COM2 为例：找到端口后，点“COM2 口”→右键属性→“端口设置或 Parametres du port” →“高级或 Avance”即可看到，选择对应的空闲 COM 口即可。



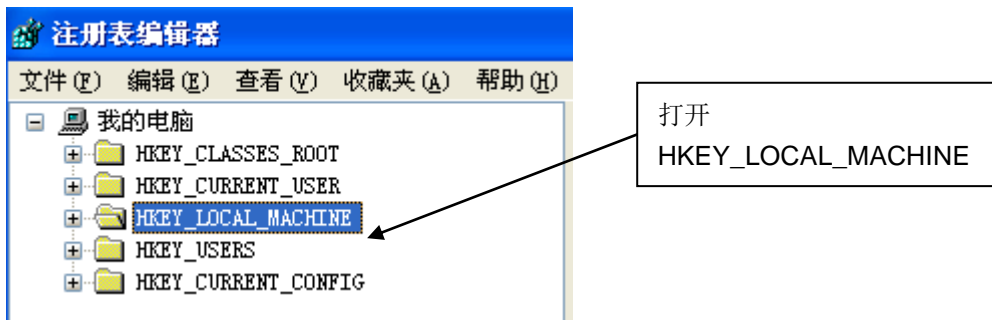


这里需要注意的是：如果您的电脑本身有 COM2 口，您不可以把端口改成重复的 COM 口。

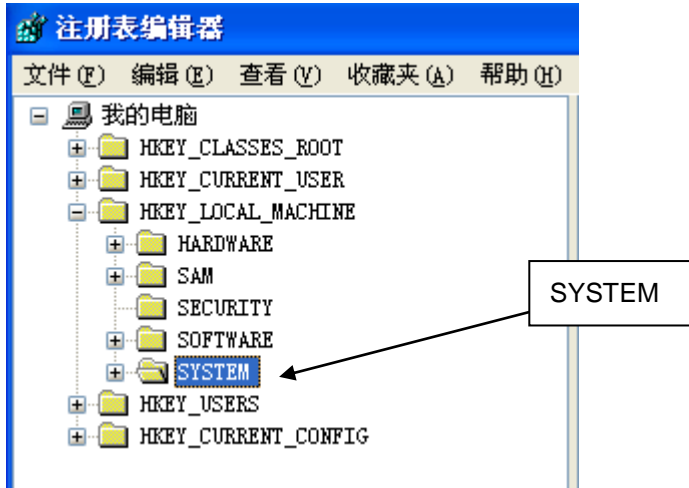
T.3 释放 COM 口

如果 COM1 到 COM8 都提示在使用中（但实际上您没在使用），则可能需要释放在被占用的 COM 口，步骤如下：“开始”-“运行”-输入“regedit”，进入注册表编辑器；

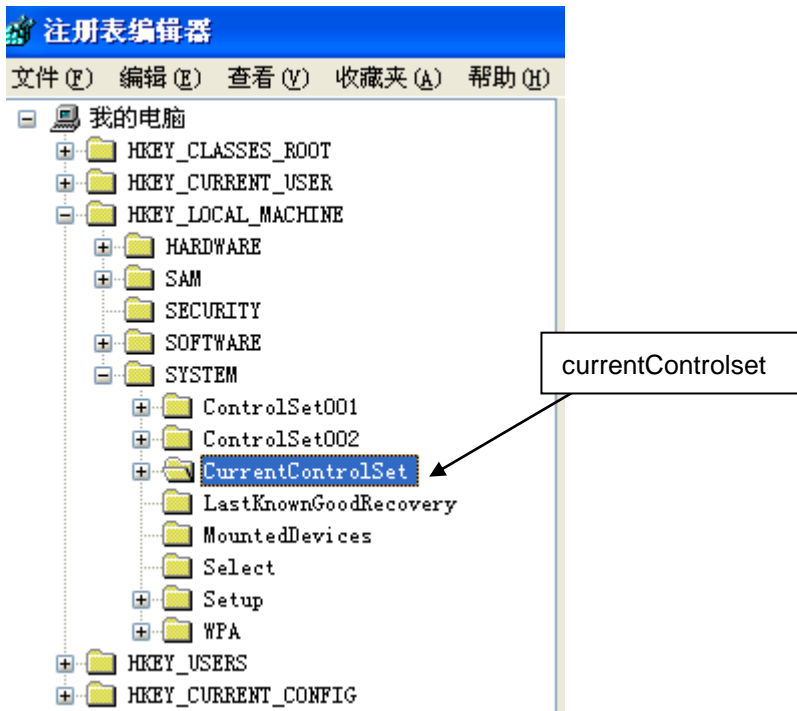
1、打开



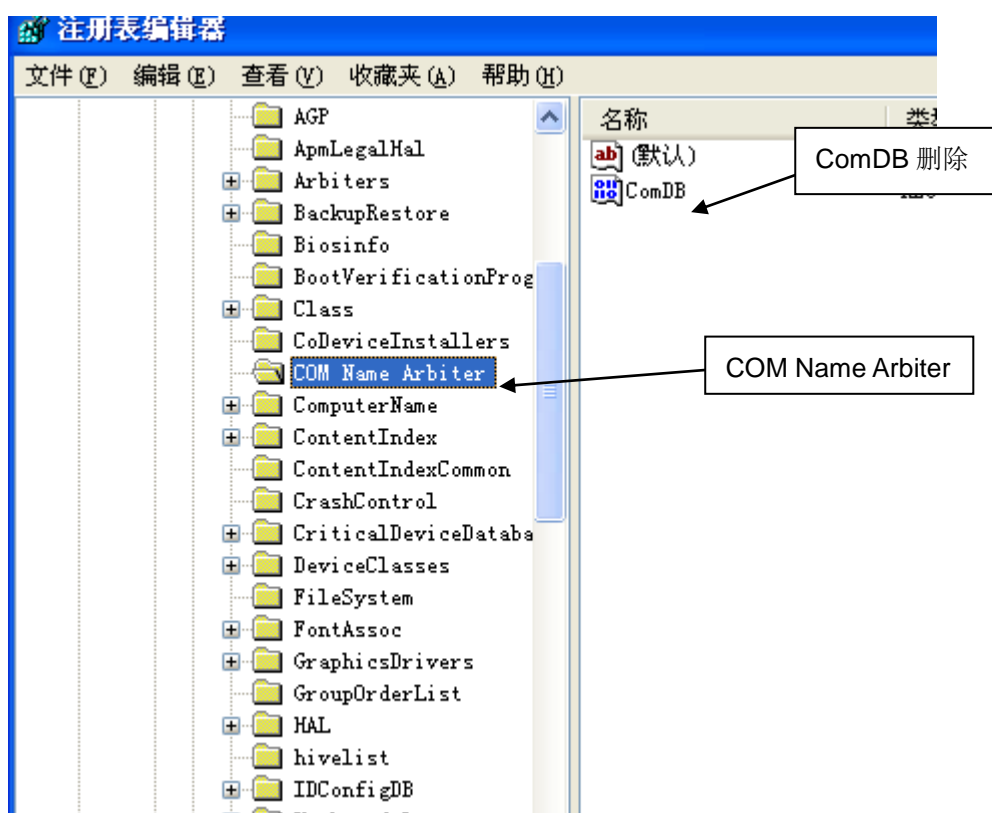
➤ 打开 SYSTEM:



3、打开 currentControlset



4、打开 COM Name Arbiter，删除 ComDB 即可，不行的话重新启动电脑就 OK;



U 编程卡使用说明

表 V-1 产品规格

	规格		指标
	型号		
订货号	旧版	CTS7 291-PC001	
	新版 V2	CTS7 291-PC002	
尺寸 (W×H×D)	63 mm × 33 mm × 17 mm		

注意：

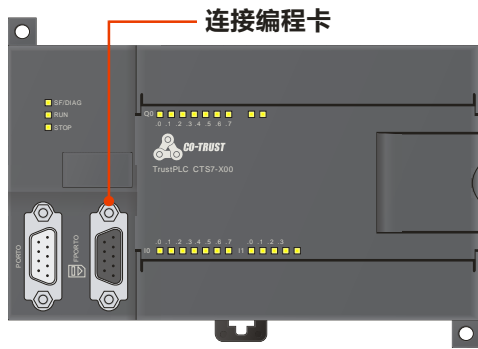
- CTH200 系列 PLC 需使用数据存储卡与便携式编程卡来完成相应数据存储和便携式编程功能。
- CTH200 系列 PLC 不支持第三方存储卡。
- CTH200 系列 H288XL 不支持 CTSC 291-PC001 V1.0 (128K 版本)。
- 新版编程卡 CTS7 291-PC002 具有更大容量 (4M)，支持更多程序下载。

【安全规格】

为了保证编程卡不被恶意破解和修改，编程卡的数据将被加密处理。

【连接接口】

连接 PLC 靠里面的 RS485 口。

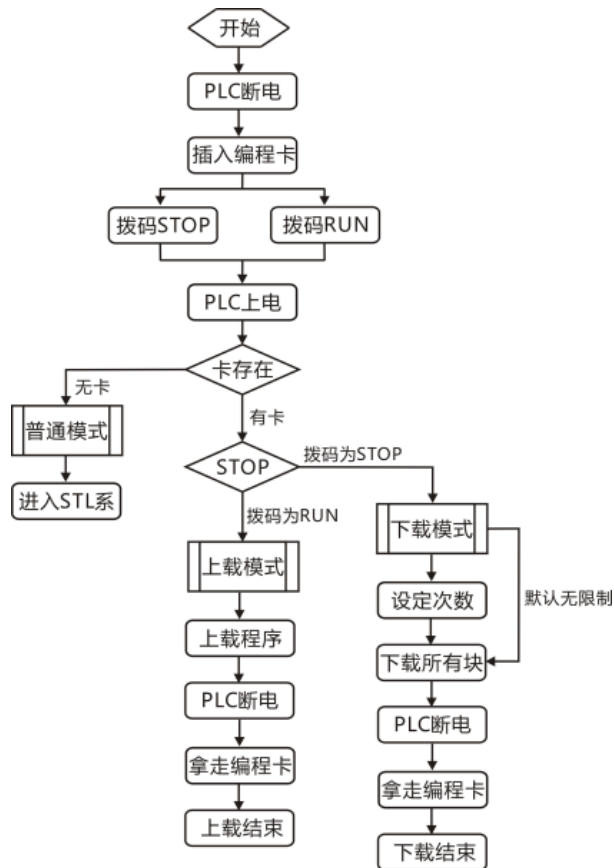


提示

编程卡从 CPU 下载程序时，同时也记录了 CPU 的类型，如果上载的 CPU 类型不符合，不允许上载。

编程卡的使用

【使用流程图】



【编程卡的功能】


※ 编程卡下载功能

让用户把 MicroWIN 工程中的 STL 程序块，STL 程序库，系统块，以及数据块下载到编程卡的存储器去。

使用步骤：

步骤 1：首先把 PLC 断电，然后插入编程卡到指定端口。

步骤 2：把拨码开关置于“STOP”状态，然后给 PLC 加电。

步骤 3: 如果编程卡有效, 系统默认进入“编程卡下载模式”, STOP 橙灯常亮, 直接点击下载。单击工具栏  或打开“文件”->“下载”就可以把程序下载到 PLC 和编程卡中。



注意: 对于没有下载到编程卡里的块, 编程卡中会自动生成一个默认块, 上载时会将默认块上载给 CPU。

步骤 4: 如果编程卡无效, 系统进入普通模式。

步骤 5: 如果用户需要设定编程卡的使用次数, 必须在下载任何块之前通过改变 SMB100 的值来设定, 否则默认是不限制使用 (SMB100 的值为 0)。

步骤 6: 如果用户不希望覆盖目标 PLC 的程序块或者数据块, 可以通过改变 SMB111 的值来设定, 否则默认是全部覆盖。SMB111 的具体定义请参见“编程卡上载功能”部分的表格。

步骤 7: 依次把要编程的块下载到 CPU。

步骤 8: CPU 把下载的所有块发送给编程卡(有发送时, STOP 橙灯闪烁, 所有发送任务完成后, 橙灯常亮)

步骤 9: 当所有块下载完成, 橙灯从闪烁变常亮后超过 10 秒钟, 用户就可以给 PLC 断电, 然后拔走编程卡。(如果下载完成, 此时将 CPU 运行开关拨打到运行, CPU 也不会运行, 只有将 CPU 断电, 拔出编程卡, CPU 才能正常工作。)



提示

- 如果用户使用 187.5K 波特率下载各种块, 在下载完成后, CPU 可能还需要等待较长的时间才能把块传送到编程卡。
- CPU 确认被下载了第一个块时, 会把编程卡的数据全部清除, 并通过 SMB100 设定编程卡的次数限制。
- 如需选择覆盖功能, 请在设定 SMB111 后再下载各个块到 CPU。
- 为安全起见, 用户下载到编程卡的块中建议必须包括系统块, 如不下载系统块, 编程卡会变为新建工程时的默认系统块。下载过程中灯闪时不能断电, 否则可能会导致编程卡内容丢失。

表 V-2 指示灯状态

指示灯	灭	闪烁	亮
SF 灯	正常	下载数据遇到错误	---
STOP 灯-橙色	---	传送数据中	正常
SMB100	设定编程次数限制		
SMB111	覆盖功能控制字节		

闪烁频率：0.5 秒左右（不要求精度）

※ 编程卡上载功能

用户可以把编程卡的所有块上载到 CPU，并更新 CPU 的 EPROM。

使用步骤：

步骤 1：首先把 PLC 断电，然后插入编程卡到指定端口。

步骤 2：把拨码开关置于“RUN”状态，然后给 PLC 加电。

步骤 3：如果编程卡有效，系统进入“编程卡上载模式”，RUN 绿灯常亮。

步骤 4：如果编程卡无效，系统进入普通模式。

步骤 5：CPU 检测到卡存在，将清除原有所有块，并把所有块读入到 EPROM 中。

注意：对于没有下载到编程卡里的块，编程卡中会自动生成一个默认块，上载时会将默认块上载给 CPU。

步骤 6：RUN 绿灯在上载过程中会闪烁，等绿灯从闪烁变常亮后超过 10 秒钟，用户就可以给 PLC 断电，然后拨走编程卡。

步骤 7：重新给 PLC 加电，新的 STL 程序将被执行。

以下情况可能会导致上载失败：

- 1> 编程卡使用次数已到限额（不影响 CPU 的原有程序等块）。
- 2> 编程卡中信息校验失败（CPU 的原有程序有可能被删除）。
- 3> 编程卡中其他块校验失败（校验无误的块会被上载到 CPU，原有块删除）。
- 4> 对有密码保护的 CPU 进行编程卡上载时，如果编程卡未被设置为覆盖程序块(即 Bit0 未设为 0)，那么上载操作会失败，从而保护用户程序安全。

上载失败后，SF 红灯会闪烁。

<注意>

SMB110 至 SMB111 用于编程卡上载控制数据，具体定义如下表所示：

表 V-3 SMB110~111 地址具体定义：

SM 地址	功能描述
SMB110	用于启动强制上载保护功能
SMB111	编程卡覆盖选项： 0 -覆盖程序块和数据块；1 -不覆盖程序块，覆盖数据块； 2 -覆盖程序块，不覆盖数据块；3 -不覆盖程序块和数据块； 4 -与为0时相同

表 V-4 指示灯状态

	灭	闪烁	亮
RUN 灯绿色	---	加载程序中	正常
SF 灯	正常	遇到错误	---
SMB101	剩余使用次数（包括本次），如果无限制，显示 255		

闪烁频率：0.5 秒左右（不要求精度）

※ 编程卡次数限制功能

编程卡次数限制可以允许客户设定编程卡的使用次数，每次上载成功后，编程卡中可使用的次数就减一。

使用步骤：

步骤 1：当 PLC 处于“编程卡下载模式”时，在用户 MicroWIN 下载第一个块前，通过设定 SMB100 来设定次数限制，并且在 SMB101 的位置显示出来。

步骤 2：用户每上载成功一次（包括上载失败的第三种情况），次数限制就会被减少 1。

表 V-5 参数值含义说明

值	说明
0	无限制使用
1-255	可使用的次数
默认值	0

V 电池卡和存储卡使用说明

CTH200 配有专用电池卡和存储卡，其中 CPU H228XL 不支持存储卡。

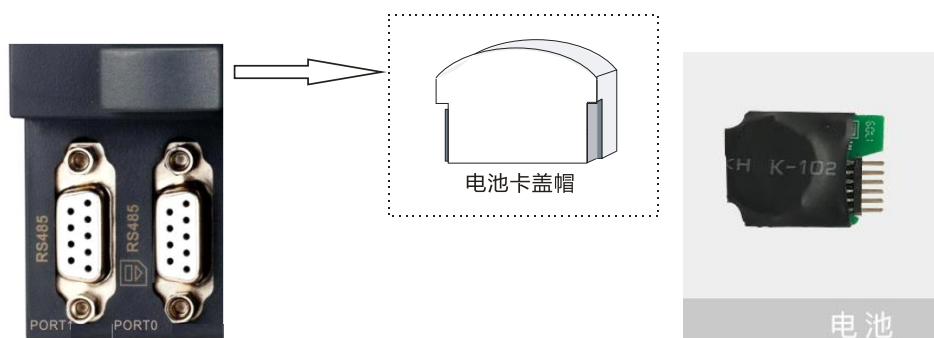
表 W-1 订货数据

类型	描述	订货号
存储卡	存储卡，64K（配方和数据记录）	CTS7 291-MC064
存储卡	存储卡，256K（配方和数据记录）	CTS7 291-MC256
电池卡	电池卡（数据保持时间）：200 天，典型	CTH2 291-8BA33

具有可拆卸特性，如需更换电池卡或存储卡，请按以下操作步骤：

- 1) 在设备断电状态下，拔下电池卡盖帽。
- 2) 取下卡槽中的原有电池卡，操作时请沿竖直方向拔卡，勿损坏卡槽。
- 3) 将装有新电池的电池卡插入卡槽并盖上盖帽。

在更换电池前，先备份好数据再给 PLC 断电，然后直接更换新电池就可以。



注意

- 1、请更换本公司同规格电池卡。如要想更换电池，应注意购买同型号的电池。取、装电池时动作要轻，千万不要碰坏其他元器件。更换电池前请保存好相关数据以免丢失。
- 2、CT 存储卡与西门子存储卡使用方法相同，主要用于存储配方和数据记录等信息，但 CTH200 存储卡在 PLC 上电重启后存储卡内容不会自动复制到 PLC 中。

W 特殊存储器功能说明

SMB0: 系统状态位

SMB0 的 8 个位在系统中有特殊作用，在每次扫描周期结束时 PLC 会更新这 8 个位。用户在程序可以利用这些位的特殊功能来实现一些常用的操作，具体描述见下表：

SMB0位	功能描述（只读）
SM0.0	该位始终打开，如果您希望程序中某段语句每一周期都能执行，可用此位作使能位。
SM0.1	该位只在首次扫描时为1，可用作初始化程序段的使能位。
SM0.2	如果断电保持数据丢失，该位为打开一个扫描周期。该位可用作错误内存位或激活特殊启动顺序的使能位。
SM0.3	从电源打开进入 RUN（运行）模式时，该位为接通一个扫描周期。该位可用于在启动操作之前提供机器预热时间。
SM0.4	该位提供一个周期为1分钟30秒打开30秒关闭的方波时钟脉冲。
SM0.5	该位提供一个周期为1秒钟0.5秒打开0.5秒关闭的方波时钟脉冲。
SM0.6	该位为扫描时钟，本次扫描时置1，下次扫描时置0。可用作扫描计数器的使能位。
SM0.7	该位指示 PLC 开关的位置（0：TERM、1：RUN）。可用该位作自由口通信程序段的使能位，这样当开关在 RUN 位置时，自由口通信方式有效，切换至 TERM 位置时，PPI 通信有效。

SMB1: 指令执行状态位

SMB1 的 8 个位用于指示某些指令执行中的错误或特殊状态，详细描述见下表：

SMB1位	功能描述（只读）
SM1.0	指令执行结果为0时，该位置1。
SM1.1	指令执行结果溢出或查出非法数值时，该位置1。
SM1.2	指令执行结果为负数时，该位置1。
SM1.3	指令中除数为0，该位置1。
SM1.4	执行 ATT 指令时，如果追加后的表长度超出表范围，该位置1。
SM1.5	执行 LIFO 或 FIFO 指令时，如果表为空，该位置1。
SM1.6	当试图把一个非 BCD 数转换为二进制数时，该位置1。
SM1.7	当 ASCII 码不能转换为有效的十六进制数时，该位置1。

SMB2: 自由口接收字符

SMB2 为自由口接收字符缓冲区，在自由口通信方式下，接收到的每个字符都放在这里，便于程序读取。详细描述见下表：

SM 地址	功能描述（只读）
SMB2	该字节包含自由口通讯过程中从端口0或端口1接收的每个字符。

SMB3: 自由口奇偶校验错误

SMB3 用于显示自由口奇偶校验错误。自由口通信模式下，在接收字符中检测到校验错误时，SM3.0 被置为 1。通常情况下，在程序接收和读取存储在 SMB2 中的信息字符数值之前，先使用该位检测自由口信息字符是否有传输错误。详细描述见下表：

SMB3位	功能描述（只读）
SM3.0	该位表示在端口0和端口1中是否存在校验错误。 0：无错，1：错误
SM3.1~SM3.7	保留

SMB4：队列溢出

SMB4 包含中断队列溢出位，中断是否允许标志位，发送空闲位及是否存在强制标志位。队列溢出表明要么是中断发生的频率高于 CPU，要么是中断已经被全局中断禁止指令所禁止。详细描述见下表：

SMB4位	功能描述（只读）
SM4.0	通信中断队列溢出时，该位打开。仅限在中断程序中使用，队列空置且控制返回主程序时，该位被复位。
SM4.1	输入中断队列溢出时，该位打开。仅限在中断程序中使用，队列空置且控制返回主程序时，该位被复位。
SM4.2	定时中断队列溢出时，该位打开。仅限在中断程序中使用，队列空置且控制返回主程序时，该位被复位。
SM4.3	检测到运行时间编程错误时，该位打开。
SM4.4	该位反映全局中断启用状态，启用中断时，该位打开。
SM4.5	端口0变送器闲置时，该位打开。
SM4.6	端口1变送器闲置时，该位打开。
SM4.7	当任何内存位置被强制时该位打开。

SMB5：I/O 状态

SMB5 包含系统里发现的 I/O 错误状态位，这些位提供了所发现的 I/O 错误的信息，详细描述见下表：

SMB5位	功能描述（只读）
SM5.0	当有 I/O 错误时，将该位置1。
SM5.1	当 I/O 总线上连接了过多的数字量 I/O 点时，将该位置1。
SM5.2	当 I/O 总线上连接了过多的模拟量 I/O 点时，将该位置1。
SM5.3	当 I/O 总线上连接了过多的智能 I/O 模块时，将该位置1。
SM5.4~SM5.7	保留

SMB6：CPU 标识寄存器

SMB6 为 CPU 识别（ID）寄存器，用于识别 CPU 的类型，详细描述见下表：

SM 地址	功能描述（只读）
SMB6	CPU 标识寄存器
SM6.0~SM6.3	保留
SM6.4~SM6.7	用于识别 CPU 类型 0000：CPU212，CPU222；0010：CPU214，CPU224； 0110：CPU221；1000：CPU215； 1001：CPU216，CPU226，CPU226XM；1110：H35-00；

SMB6 格式:

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				保留			

SMB8 至 SMB21: I/O 模块标识和错误寄存器

SMB8 到 SMB21 是按照字节对形式（相邻两个字节）为扩展模 0 到 6 而准备的。每对字节的偶数位字节为模块识别寄存器，奇数位字节为模块错误寄存器。前者标记着模块类型、I/O 类型、输入和输出点数；后者为对相应模块所测得的 I/O 错误提示。详细描述见下表：

SM 地址	功能描述（只读）
SMB8	模块0识别寄存器
SMB9	模块0错误寄存器
SMB10	模块1识别寄存器
SMB11	模块1错误寄存器
SMB12	模块2识别寄存器
SMB13	模块2错误寄存器
SMB14	模块3识别寄存器
SMB15	模块3错误寄存器
SMB16	模块4识别寄存器
SMB17	模块4错误寄存器
SMB18	模块5识别寄存器
SMB19	模块5错误寄存器
SMB20	模块6识别寄存器
SMB21	模块6错误寄存器

偶数字节（模块 ID 寄存器）格式:

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: 模块是否存在 0--有模块 1--无模块	tt: 模块类型 00--非智能 I/O 模块 01--智能模块 10--保留 11--保留	a: I/O 类型 0--开关量 1--模拟量	ii: 输入点数 00--无输入 01--2 AI 或 8 DI 10--4 AI 或 16 DI 11--8 AI 或 32 DI	qq: 输出点数 00--无输出 01--2 AQ 或 8 DQ 10--4 AQ 或 16 DQ 11--8 AQ 或 32 DQ			

奇数字节（模块错误寄存器）格式：

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: 配置错误 0--无错误 1--有错误			b: 总线错误 或校验错误 0--无错误 1--有错误	r: 超范围 错误 0--无错误 1--有错误	p: 用户电 源错误 0--无错误 1--有错误	f: 熔断器 错误 0--无错误 1--有错误	t: 端子块 松动错误 0--无错误 1--有错误

SMW22 至 SMW26：扫描时间

SMW22、SMW24 和 SMW26 提供扫描时间信息：以毫秒计的最短扫描时间、最长扫描时间及上次扫描时间。详细描述见下表：

SM 地址	功能描述（只读）
SMW22	上次扫描时间
SMW24	进入 RUN 方式后，所记录的最短扫描时间
SMW26	进入 RUN 方式后，所记录的最长扫描时间

SMB28 至 SMB29：模拟电位器

特殊内存字节 28 和 29 包含与模拟电位器 0 和 1 轴位置对应的数字值。模拟电位器位于 CPU 前方存取门后方。用一把小螺丝刀调整电位计（沿顺时针方向增加，或沿逆时针方向减少）。此类只读数值可被程序用于各种不同的功能，例如，为定时器或计数器更新当前值，输入或改动预设值或设置限制。模拟电位器有一个 0 至 255 的额定范围，以及 ± 2 计数的可靠重复。详细描述见下表：

SM 地址	功能描述（只读）
SMB28	存储模拟调节器0的输入值。在 STOP/RUN 方式下，每次扫描时更新该值。
SMB29	存储模拟调节器1的输入值。在 STOP/RUN 方式下，每次扫描时更新该值。

SMB30 和 SMB130：自由口控制寄存器

SMB30 控制自由口 0 的通讯方式，SMB130 控制自由口 1 的通讯方式。您可以对 SMB30 和 SMB130 进行写和读，利用这两个字节设置自由口通信的操作方式，并提供自由口或者系统所支持的协议之间的选择。详细描述见下表：

SM 地址	功能描述
SMB30	自由口0控制寄存器
SMB130	自由口1控制寄存器
SM30.0~SM30.1	端口0通信协议选择
SM130.0~SM130.1	端口1通信协议选择
SM30.2~SM30.4	自由口0波特率
SM130.2~SM130.4	自由口1波特率
SM30.5	自由口0每个字符的数据位
SM130.5	自由口1每个字符的数据位
SM30.6~SM30.7	自由口0的校验选择
SM130.6~SM130.7	自由口1的校验选择

自由口控制字节格式:

MSB				LSB			
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
p	p	d	b	b	b	m	m
pp: 校验选择 00--无校验 01--偶校验 10--无校验 11--奇校验		d: 每个字符的数据位 0--8 位数据位 1--7 位数据位	bbb: 自由口波特率 000--38400 bps 001--19200 bps 010--9600 bps 011--4800 bps 100--2400 bps 101--1200 bps 110--115200 bps 111--57600 bps			mm: 协议设置 00--点到点接口协议 (PPI/从站模式) 01--自由口协议 10--PPI/主站模式 11--保留 (缺省是 PPI/从站模式)	

<注意> 当选择 mm = 10 (PPI 主站), PLC 将成为网络的一个主站, 可以执行 NETR 和 NETW 指令。PPI 模式下忽略 2 至 7 位。

SMB31 和 SMW32: 永久存储器 (EEPROM) 写控制

在用户程序的控制下, 您可以把 V 存储器中的数据存入永久存储器, 亦称非易失存储器。先把被存数据的地址存入 SMW32 中, 然后把存入命令存入 SMB31 中。一旦您发出存储命令, 则直到 CPU 完成存储操作 SM31.7 被置 0 之前, 您不可以改变 V 存储器的值。

在每次扫描周期末尾, CPU 检查是否有向永久存储器区中存数据的命令。如果有, 则将该数据存入永久存储器中。详细描述见下表:

SM 地址	功能描述
SMB31	永久存储器命令寄存器
SM31.0~SM31.1	要保存的数据大小 00 = 字节; 01 = 字节; 10 = 字; 11 = 双字
SM31.2~SM31.6	保留, 始终为 00000
SM31.7	存入永久存储器, 每次存储操作完成后, S7-200 复位该位。 0 = 无执行存储器操作的请求 1 = 用户程序申请向永久存储器存储数据
SMW32	所存数据的 V 存储器地址, 该值是相对于 V0 的偏移量。当执行存储命令时, 把该数据存到永久存储器中相应的位置。

SMB34 和 SMB35: 定时中断的时间间隔寄存器

SMB34 和 SMB35 分别定义了定时中断 0 和中断 1 的时间间隔。您可以指定从 1 毫秒至 255 毫秒的时间间隔 (以 1 毫秒为增量)。如果相应的定时中断事件被连接到一个中断服务程序, CTH200 CPU 就会获取该时间间隔值。若要改变该时间间隔, 您必须把定时中断事件再分配给同一或另一中断程序, 或通过中断分离来终止定时中断事件。

SM 地址	功能描述
SMB34	定时中断 0 的时间间隔数值
SMB35	定时中断 1 的时间间隔数值

SMB36 至 SMB65: HSC0、HSC1 和 HSC2 寄存器

SMB36 至 SMB65 被用于监控和控制高速计数器 HSC0、HSC1 和 HSC2 操作。详细描述见下表:

SM 地址	功能描述
HSC0	
SMB36	HSC0计数器状态
SM36.0~SM36.4	保留
SM36.5	HSC0当前计数方向位: 1=增计数
SM36.6	HSC0当前值等于预设值位: 1=等于
SM36.7	HSC0当前值大于预设值位: 1=大于
SMB37	HSC0控制字节
SM37.0	复位的有效控制位: 0=高电平复位有效, 1=低电平复位有效
SM37.1	保留
SM37.2	正交计数器的计数速率选择: 0=4x 计数速率; 1=1x 速率
SM37.3	HSC0方向控制位: 1=增计数
SM37.4	HSC0更新方向: 1=更新方向
SM37.5	HSC0更新预设值: 1=向 HSC0写新的预设值
SM37.6	HSC0更新当前值: 1=向 HSC0写新的初始值
SM37.7	HSC0有效位: 1=有效
SMD38	HSC0新的初始值, 把新的当前值写入 SMD38后将 SM37.6设为1, 则当前值被更新
SMD42	HSC0新的预置值, 把新的预设值写入 SMD42后将 SM37.5设为1, 则预设值被更新
HSC1	
SMB46	HSC1计数器状态
SM46.0~SM46.4	保留
SM46.5	HSC1当前计数方向位: 1=增计数
SM46.6	HSC1当前值等于预设值位: 1=等于
SM46.7	HSC1当前值大于预设值位: 1=大于
SMB47	HSC1控制字节
SM47.0	复位的有效控制位: 0=高电平复位有效, 1=低电平复位有效
SM47.1	HSC1启动有效电平控制位: 0=高电平, 1=低电平
SM47.2	正交计数器的计数速率选择: 0=4x 计数速率; 1=1x 速率
SM47.3	HSC1方向控制位: 1=增计数
SM47.4	HSC1更新方向: 1=更新方向
SM47.5	HSC1更新预设值: 1=向 HSC1写新的预设值
SM47.6	HSC1更新当前值: 1=向 HSC1写新的初始值
SM47.7	HSC1有效位: 1=有效
SMD48	HSC1新的初始值, 把新的当前值写入 SMD48后将 SM47.6设为1, 则当前值被更新
SMD52	HSC1新的预设值, 把新的预设值写入 SMD52后将 SM47.5设为1, 则预设值被更新
HSC2	
SMB56	HSC2计数器状态
SM56.0~SM56.4	保留
SM56.5	HSC2当前计数方向位: 1=增计数

SM56.6	HSC2当前值等于预设值位：1=等于
SM56.7	HSC2当前值大于预设值位：1=大于
SMB57	HSC2控制字节
SM57.0	复位的有效控制位：0=高电平复位有效，1=低电平复位有效
SM57.1	HSC1启动有效电平控制位：0=高电平，1=低电平
SM57.2	正交计数器的计数速率选择：0=4x 计数速率；1=1x 速率
SM57.3	HSC2方向控制位：1=增计数
SM57.4	HSC2更新方向：1=更新方向
SM57.5	HSC2更新预设值：1=向 HSC2写新的预设值
SM57.6	HSC2更新当前值：1=向 HSC2写新的初始值
SM57.7	HSC2有效位：1=有效
SMD58	HSC2新的初始值，把新的当前值写入 SMD58后将 SM57.6设为1，则当前值被更新
SMD62	HSC2新的预置值，把新的预设值写入 SMD62后将 SM57.5设为1，则预设值被更新

<注释>

- 1) 计数器状态位仅限在执行由高速计数器事件触发的中断程序里有效。
- 2) 当您使用 HSC 外部复位中断事件时，请勿在程序里重新设置新的当前值，然后在连接于该事件的中断程序里重新使能计数器，此操作会引起严重错误。

SMB66 至 SMB85: PTO/PWM 寄存器

SMB66 至 SMB85 被用于监控和控制 PLC（脉冲）指令的脉冲链输出和脉冲宽度调制功能。详细描述见下表：

SM 地址	功能描述
PTO0	
SMB66	PTO0状态字节
SM66.0~SM66.3	保留
SM66.4	PTO0包络终止：0=无错，1=由于增量计算错误而终止
SM66.5	PTO0包络终止：0=不由用户命令终止；1=由用户命令终止
SM66.6	PTO0管道溢出（当使用外部包络时由系统清除，否则由用户程序清除）：0=无溢出，1=有溢出
SM66.7	PTO0空闲位：0=PTO 忙，1=PTO 空闲
SMB67	Q0.0的控制脉冲串输出和脉冲宽度调制
SM67.0	PTO0/PWM0更新周期：1=写新的周期值
SM67.1	PWM0更新脉冲宽度值：1=写新的脉冲宽度
SM67.2	PTO0更新脉冲量：1=写新的脉冲量
SM67.3	PTO0/PWM0基准时间单元：0=1 μs/格，1=1 ms/格
SM67.4	同步更新 PWM0：0=异步更新，1=同步更新
SM67.5	PTO0操作：0=单段操作（周期和脉冲数存在 SM 存储器中），1=多段操作（包络表存在 V 存储器区）
SM67.6	PTO0/PWM0模式选择：0=PTO，1=PWM
SM67.7	PTO0/PWM0有效位：1=有效
SMW68	PTO0/PWM0周期（2~65，535个时间基准）
SMW70	PWM0脉冲宽度值（0~65，535个时间基准）

SMD72	PTO0脉冲计数值（1~232 --1）
PTO1	
SMB76	PTO1状态字节
SM76.0~SM76.3	保留
SM76.4	PTO1包络终止：0=无错，1=由于增量计算错误而终止
SM76.5	PTO1包络终止：0=不由用户命令终止；1=由用户命令终止
SM76.6	PTO1管道溢出（当使用外部包络时由系统清除，否则由用户程序清除）： 0=无溢出，1=有溢出
SM76.7	PTO1空闲位：0=PTO 忙，1=PTO 空闲
SMB77	Q0.1的控制脉冲串输出和脉冲宽度调制
SM77.0	PTO1/PWM1更新周期：1=写新的周期值
SM77.1	PWM1更新脉冲宽度值：1=写新的脉冲宽度
SM77.2	PTO1更新脉冲量：1=写新的脉冲量
SM77.3	PTO1/PWM1基准时间单元：0=1 μs/格，1=1 ms/格
SM77.4	同步更新 PWM1：0=异步更新，1=同步更新
SM77.5	PTO1操作：0=单段操作（周期和脉冲数存在 SM 存储器中），1=多段操作 （包络表存在 V 存储器区）
SM77.6	PTO1/PWM1模式选择：0=PTO，1=PWM
SM77.7	PTO1/PWM1有效位：1=有效
SMW78	PTO1/PWM1周期（2~65535个时间基准）
SMW80	PWM1脉冲宽度值（0~65535个时间基准）
SMD82	PTO1脉冲计数值（1~232 --1）

SMB86 至 SMB94 和 SMB186 至 SMB194：接收信息控制

SMB86 至 SMB94 以及 SMB186 至 SMB194 被用于控制和读取有关“接收信息”指令的状态。

详细描述见下表：

SM 地址	功能描述
P0接收信息	
SMB86	P0接收信息状态字节
SMB87	P0接收信息控制字节
SMB88	P0信息字符的开始
SMB89	P0信息字符的结束
SMW90	P0空闲行时间间隔用毫秒给出。在空闲行时间结束后接收的第一个字符是新信息的开始。
SMW92	P0字符间/信息间定时器超时值（用毫秒表示）。如果超过时间，就停止接收信息。
SMB94	P0接收字符的最大数（1~255字节）。 <注意> 这个区一定要设为希望的最大缓冲区，即使不使用字符计数信息终止。
P1接收信息	
SMB186	P1接收信息状态字节
SMB187	P1接收信息控制字节
SMB188	P1信息字符的开始
SMB189	P1信息字符的结束
SMW190	P1空闲行时间间隔用毫秒给出。在空闲行时间结束后接收的第一个字符是新信息的开始。

SMW192	P1字符间/信息间定时器超时值（用毫秒表示）。如果超过时间，就停止接收信息。
SMB194	P1接收字符的最大数（1到255字节）。 <注意> 这个区一定要设为希望的最大缓冲区，即使不使用字符计数信息终止。

接收信息状态字节格式：

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: 接收用户的禁止命令终止接收信息
- r = 1: 输入参数错误或无起始或结束条件导致接收信息终止
- e = 1: 收到结束字符
- t = 1: 接收信息终止：超时
- c = 1: 接收信息终止：超出最大字符数
- p = 1: 接收信息终止：奇偶校验错误

接收信息控制字节格式：

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 = 禁止接收信息功能。
1 = 允许接收信息功能。
每次执行 RCV 指令时检查允许/禁止接收信息位。
- sc: 0 = 忽略 SMB88 或 SMB188。
1 = 使用 SMB88 或 SMB188 的值检测起始信息。
- ec: 0 = 忽略 SMB89 或 SMB189。
1 = 使用 SMB89 或 SMB189 的值检测结束信息。
- il: 0 = 忽略 SMW90 或 SMW190。
1 = 使用 SMW90 或 SMW190 的值检测空闲状态。
- c/m: 0 = 定时器是内字符间定时器。
1 = 定时器是信息定时器。
- tmr: 0 = 忽略 SMW92 或 SMW192。
1 = 当 SMW92 或 SMW192 中的定时时间超出时终止接收。
- bk: 0 = 忽略中断条件。
1 = 用中断条件作为信息检测的开始。

SMW98: 扩展 I/O 总线错误

SMW98 给出有关扩展 I/O 总线的错误数的信息。详细描述见下表：

SM 地址	功能描述
SMW98	当扩展总线出现校验错误时，该处每次增加1。当系统得电时或用户程序写入零，可以进行清零。

SMB110 至 SMB112: 编程卡上载控制

SMB110 至 SMB111 用于编程卡上载控制，

SM 地址	功能描述
SMB110	用于启动强制上载保护功能
SMB111	编程卡覆盖选项： 0 -覆盖程序块和数据块；1 -不覆盖程序块，覆盖数据块； 2 -覆盖程序块，不覆盖数据块；3 -不覆盖程序块和数据块； 4 -与为0时相同

SMB112 至 SMB113: 显示当前 CPU 信息

SMB 字节列出当前 CPU 的版本信息。详细描述参见下表：

SM 地址	功能描述
SMB112	始终正确显示当前 CPU 版本信息（SMB112=16#83，即表示 CTH200系列）。
SMB113	始终正确显示当前 CPU 型号 H224/H224X RLY: 16#02; H224/H224X PNP: 16#03 H226X/H226XL/H226XM/H228XL RLY: 16#08; H226X/H226XL/H226XM PNP: 16#09

SMB114 至 SMB115: 扩展板信息

SMB114: 扩展板模块类型

SMB114 = 0x19: 2AI/1AQ (CTH2-AMS-03S1-EB)

0x1E: 4AI/2AQ (CTH2-AMS-06S1-EB/ CTH2-AMS-06S2-EB)

0x8A: RS485 (CTH2-485-01S1-EB)

0x8D: CAN V2 (CTH2-CAN-01S2-EB)

具体位定义如下表所示

	Bti7	Bit6	Bti5	Bit4	Bit3	Bit2	Bti1	Bit0
	E	TT		t	ll		OO	
E 模块是否存在	1: 模块不存在 0: 模块存在							
TT (模块类型)	00: 非智能模块 01: 智能模块 10: 特殊非智能模块 (如 7PF 模块等) 11: 合信智能模块 (如 PID 模块等)							
t (模块子类型)	非智能模块 0: 开关量 1: 模拟量 智能模块							

	0: 普通智能 1: 合信智能模块 (PID)
II (输入点数)	非智能模块 00: 无输入 01: AI/8DI 10: 2AI/16DI 11: 4AI/32DI
	智能模块: 保留
OO (输出点数)	非智能模块 00: 无输出 01: 1AO/8DO 10: 2AO/8DO 11: 4AO/32DO
	智能模块: 保留

SMB115: 扩展板状态

SMB115 = 0x0: 模块没有错误

0x255: 扩展板访问错误

其它: 扩展板内部诊断

SMW116 至 SMW126: 扩展板模拟量映射区

映射区的具体对应关系如下表所示:

扩展板	模拟量映射区
2AI/1AQ	SMW116对应扩展板的 AIW0; SMW118对应 AIW2; SMW124对应 AQW0
4AI/2AQ	SMW116对应扩展板的 AIW0; SMW118对应 AIW2; SMW120对应 AIW4 SMW122对应 AIW6; SMW124对应 AQW0; SMW126对应 AQW2

SMB131 CAN 扩展板访问周期设置

CAN 扩展板访问周期 (毫秒), 设为 0 时默认为 1 毫秒

SMB136 至 SMB165: HSC3、HSC4 和 HSC5 寄存器

SMB136 至 SMB165 用于监视和控制高速计数器 HSC3、HSC4 和 HSC5 操作。详细描述见下表:

SM 地址	功能描述
HSC3	
SMB136	HSC3计数器状态
SM136.0~4	保留
SM136.5	HSC3当前计数方向位: 1=增计数
SM136.6	HSC3当前值等于预设值位: 1=等于
SM136.7	HSC3当前值大于预设值位: 1=大于
SMB137	HSC3控制字节
SM137.0~2	保留
SM137.3	HSC3方向控制位: 1=增计数
SM137.4	HSC3更新方向: 1=更新方向
SM137.5	HSC3更新预设值: 1=向 HSC3写新的预设值

SM137.6	HSC3更新当前值：1=向 HSC3写新的初始值
SM137.7	HSC3有效位：1=有效
SMD138	HSC3新的初始值，把新的当前值写入 SMD138后将 SM137.6设为1，则当前值被更新
SMD142	HSC3新的预置值，把新的预设值写入 SMD142后将 SM137.5设为1，则预设值被更新
HSC4	
SMB146	HSC4计数器状态
SM146.0~4	保留
SM146.5	HSC4当前计数方向位：1=增计数
SM146.6	HSC4当前值等于预设值位：1=等于
SM146.7	HSC4当前值大于预设值位：1=大于
SMB147	HSC4控制字节
SM147.0	复位的有效控制位：0=高电平复位有效，1=低电平复位有效
SM147.1	保留
SM147.2	正交计数器的计数速率选择：0=4x 计数速率；1=1x 速率
SM147.3	HSC4方向控制位：1=增计数
SM147.4	HSC4更新方向：1=更新方向
SM147.5	HSC4更新预设值：1=向 HSC4写新的预设值
SM147.6	HSC4更新当前值：1=向 HSC4写新的初始值
SM147.7	HSC4有效位：1=有效
SMD148	HSC4新的初始值，把新的当前值写入 SMD148后将 SM147.6设为1，则当前值被更新
SMD152	HSC4新的预设值，把新的预设值写入 SMD152后将 SM147.5设为1，则预设值被更新
HSC5	
SMB156	HSC5计数器状态
SM156.0~4	保留
SM156.5	HSC5当前计数方向位：1=增计数
SM156.6	HSC5当前值等于预设值位：1=等于
SM156.7	HSC5当前值大于预设值位：1=大于
SMB157	HSC5控制字节
SM157.0~2	保留
SM157.3	HSC5方向控制位：1=增计数
SM157.4	HSC5更新方向：1=更新方向
SM157.5	HSC5更新预设值：1=向 HSC5写新的预设值
SM157.6	HSC5更新当前值：1=向 HSC5写新的初始值
SM157.7	HSC5有效位：1=有效
SMD158	HSC5新的初始值，把新的当前值写入 SMD158后将 SM157.6设为1，则当前值被更新
SMD162	HSC5新的预置值，把新的预设值写入 SMD162后将 SM157.5设为1，则预设值被更新

<注释>

- 1) 计数器状态位仅限在执行由高速计数器事件触发的中断程序里有效。

2) 当您使用 HSC 外部复位中断事件时, 请勿在程序里重新设置新的当前值, 然后在连接于该事件的中断程序里重新使能计数器, 此操作会引起严重错误。

SMB166 至 SMB185: PTO0, PTO1 包络定义表

SMB166 到 SMB185 用来显示包络步的数量和包络表的地址和 V 存储器区中表的地址 (H224/H226L/H226M 不支持; H224X/H226XL/H226XM/H228XL 支持)。详细描述见下表:

SM 地址	功能描述
PTO0	
SMB166	PTO0的包络步当前计数值
SMB167	保留
SMW168	PTO0的包络表 V 存储器地址 (从 V0开始的偏移量)
SMB170	线性 PTO0状态字节
SMB171	线性 PTO0结果字节
SMD172	指定线性 PTO0发生器工作在手动模式时产生的频率。频率是一个以 Hz 为单位的 双整型值。
PTO1	
SMB176	PTO1的包络步当前计数值
SMB177	保留
SMW178	PTO1的包络表 V 存储器地址 (从 V0开始的偏移量)
SMB180	线性 PTO1状态字节
SMB181	线性 PTO1结果字节
SMD182	指定线性 PTO1发生器工作在手动模式时产生的频率。频率是一个以 Hz 为单位的 双整型值。

SM195: 通信状态

SM195.6	CPU 连接上远程服务器以后 SM195.6=1, 否则设置为0
SM195.7	控制 TCP_Modbus 通信中超时是否发心跳包。SM195.7=1, 通信超时发心跳包; 否则不发; 默认为0。

SMB200 至 SMB549: 智能模块状态

SMB200 到 SMB549 预留存储智能扩展模块的信息。详细描述见下表:

0槽智能 模块	1槽智能 模块	2槽智能 模块	3槽智能 模块	4槽智能 模块	5槽智能 模块	6槽智能 模块	功能描述
SMB200 ~215	SMB250 ~265	SMB300 ~315	SMB350 ~365	SMB400 ~415	SMB450 ~465	SMB500 ~515	模块名称(16个 ASCII 字符)
SMB216 ~219	SMB266 ~269	SMB316 ~319	SMB366 ~369	SMB416 ~419	SMB466 ~469	SMB516 ~519	SW 修订号(4个 ASCII 字符)
SMW220	SMW270	SMW320	SMW370	SMW420	SMW470	SMW520	错误代码
SMB222 ~249	SMB272 ~299	SMB322 ~349	SMB372 ~399	SMB422 ~449	SMB472 ~499	SMB522 ~549	与特定模块类 型相关的信息

X 订货信息

表 Y-1 产品订货信息一览表

产品名称与规格	产品订货号
CPU 模块	
基本型	
CPU H224 12KB 程序空间/8KB 数据空间, 220VAC 电源, 14DI/10DO 继电器输出, 2A, 1 个 PPI, 1 个自由通讯口, 1 个以太网通讯口, 支持块展板, 不支持扩展模块	CTH2 214-1BA33-0X24
CPU H226L 12KB 程序空间/8KB 数据空间, 220VAC 电源, 24DI/16DO 继电器输出, 2A, 2 个 PPI/自由通讯口, 1 个以太网通讯口, 支持块展板, 不支持扩展模块	CTH2 216-1BA33-0X40
标准型	
CPU H224 12KB 程序空间/8KB 数据空间, 24VDC 电源, 14DI/10DO 晶体管源型输出, 0.5A, 1 个 PPI, 1 个自由通讯口, 1 个以太网通讯口, 3 路 50KHz 运控输出	CTH2 214-1AD33-0X24
CPU H224 12KB 程序空间/8KB 数据空间, 220VAC 电源, 14DI/10DO 继电器输出, 2A, 1 个 PPI, 1 个自由通讯口, 1 个以太网通讯口	CTH2 214-1BD33-0X24
CPU H226L 12KB 程序空间/8KB 数据空间, 24VDC 电源, 24DI/16DO 晶体管源型输出, 0.5A, 2 个 PPI/自由通讯口, 1 个以太网通讯口, 3 路 50KHz 运控输出	CTH2 216-2AD33-0X40
CPU H226L 12KB 程序空间/8KB 数据空间, 220VAC 电源, 24DI/16DO 继电器输出, 2A, 2 个 PPI/自由通讯口, 1 个以太网通讯口	CTH2 216-2BD33-0X40
CPU H226M 12KB 程序空间/8KB 数据空间, 24VDC 电源, 14DI/10DO 晶体管源型输出, 0.5A, 2 个 PPI/自由口, 1 个以太网口, 3 路 50KHz 运控输出	CTH 216-1AD33-0X24
CPU H226M 12KB 程序空间/8KB 数据空间, 220VAC 电源, 14DI/10DO 继电器输出, 2A, 2 个 PPI/自由口, 1 个以太网口	CTH 216-1BD33-0X24
高性能型	
CPU H224X 16KB 程序空间/108KB 数据空间, 24VDC 电源, 14DI/10DO 晶体管源型输出, 0.5A, 1 个 PPI, 1 个自由通讯口, 1 个以太网通讯口, 2 路 50KHz 输出 (Pulse/Dir 或 PTO/PWM)	CTH2 214-1AX33-0X24
CPU H224X 16KB 程序空间/108KB 数据空间, 220VAC 电源, 14DI/10DO 继电器输出, 2A, 1 个 PPI, 1 个自由通讯口, 1 个以太网口	CTH2 214-1BX33-0X24
CPU H226XL 72KB 程序空间/110KB 数据空间, 24VDC 电源, 24DI/16DO 晶体管源型输出, 0.5A, 2 个 PPI/自由通讯口, 1 个以太网通讯口, 2 路 50KHz 输出 (Pulse/Dir 或 PTO/PWM)	CTH2 216-2AX33-0X40
CPU H226XL 72KB 程序空间/110KB 数据空间, 220VAC 电源, 24DI/16DO 继电器输出, 2A, 2 个 PPI/自由通讯口, 1 个以太网通讯口	CTH2 216-2BX33-0X40
CPU H228XL 96KB 程序空间/10KB 数据空间, 220VAC 电源, 36DI/24DO 继电器输出, 2A, 2 个 PPI/自由通讯口, 1 个以太网通讯口	CTH2 218-3BX33-0X60

产品名称与规格	产品订货号
高性能型升级版	
CPU H224X 16KB 程序空间/108KB 数据空间，24VDC 电源，14DI/10DO 晶体管源型输出，0.5A，1 个 PPI，1 个自由通讯口，1 个以太网通讯口，2 路 20KHz（PTO/PWM）或 2 路 100KHz 输出（Pulse/Dir），数据永久保存，支持模拟量和 CAN 扩展板，支持西门子 S7 协议	CTH2 214-1AX35-0X24
CPU H224X 16KB 程序空间/108KB 数据空间，220VAC 电源，14DI/10DO 继电器输出，2A，1 个 PPI，1 个自由通讯口，1 个以太网口，数据永久保存，支持模拟量和 CAN 扩展板，支持西门子 S7 协议	CTH2 214-1BX35-0X24
CPU H226XM 72KB 程序空间/110KB 数据空间，24VDC 电源，14DI/10DO 晶体管源型输出，0.5A，2 个 PPI/自由口，1 个以太网口，2 路 20KHz（PTO/PWM），3 路 100KHz 输出（Pulse/Dir），数据永久保存，支持模拟量、CAN 和 RS485 扩展板，支持西门子 S7 协议	CTH2 216-1AX35-0X24
CPU H226XM 72KB 程序空间/110KB 数据空间，220VAC 电源，14DI/14DO 继电器输出，2A，2 个 PPI/自由口，1 个以太网口，数据永久保存，支持模拟量、CAN 和 RS485 扩展板，支持西门子 S7 协议	CTH2 216-1BX35-0X24
CPU H226XL 72KB 程序空间/110KB 数据空间，24VDC 电源，24DI/16DO 晶体管源型输出，0.5A，2 个 PPI/自由通讯口，1 个以太网通讯口，2 路 20KHz（PTO/PWM）或 2 路 100KHz 输出（Pulse/Dir），数据永久保存，支持模拟量、CAN 和 RS485 扩展板，支持西门子 S7 协议	CTH2 216-2AX35-0X40
CPU H226XL 72KB 程序空间/110KB 数据空间，220VAC 电源，24DI/16DO 继电器输出，2A，2 个 PPI/自由通讯口，1 个以太网口，数据永久保存，支持模拟量、CAN 和 RS485 扩展板，支持西门子 S7 协议	CTH2 216-2BX35-0X40
CPU H226IM 256KB 程序空间/164KB 数据空间，24VDC 电源，14DI/10DO 晶体管源型输出，0.5A，2 个 PPI/自由口，1 个以太网口，2 路 20KHz（PTO/PWM），3 路 100KHz 输出（Pulse/Dir），数据永久保存，支持模拟量、CAN 和 RS485 扩展板，支持西门子 S7 协议	CTH2 216-1AD46-0X24
CPU H226IM 256KB 程序空间/164KB 数据空间，220VAC 电源，14DI/10DO 继电器输出，2A，2 个 PPI/自由口，1 个以太网口，数据永久保存，支持模拟量、CAN 和 RS485 扩展板，支持西门子 S7 协议	CTH2 216-1BD46-0X24
CPU H226IL 256KB 程序空间/164KB 数据空间，24VDC 电源，24DI/16DO 晶体管源型输出，0.5A，2 个 PPI/自由口，1 个以太网口，2 路 20KHz（PTO/PWM），3 路 100KHz 输出（Pulse/Dir），数据永久保存，支持模拟量、CAN 和 RS485 扩展板，支持西门子 S7 协议	CTH2 216-2AD46-0X40
CPU H226IL 256KB 程序空间/164KB 数据空间，220VAC 电源，24DI/16DO 继电器输出，2A，2 个 PPI/自由通讯口，1 个以太网口，数据永久保存，支持模拟量、CAN 和 RS485 扩展板，支持西门子 S7 协议	CTH2 216-2BD46-0X40
CPU H226IH 256KB 程序空间/164KB 数据空间，24VDC 电源，14DI/10DO 晶体管漏型输出，0.5A，2 个 PPI/自由口，1 个以太网口，4 路 200KHz 输出（Pulse/Dir，）数据永久保存，支持模拟量、CAN 和 RS485 扩展板，支持西门子 S7 协议	CTH2 216-1AH46-2B24

扩展模块	产品订货号
SM221 数字量输入模块, 8 点输入, 24VDC	CTH2 221-1BF32
SM221 数字量输入模块, 16 点输入, 24VDC	CTH2 221-1BH32
SM221 数字量输入模块, 32 点输入, 24VDC	CTH2 221-1BL32
SM222 数字量输出模块, 8 点晶体管输出, 24VDC, 0.5A(输出保护)	CTH2 222-1BF32
SM222 数字量输出模块, 8 点晶体管 NPN 型输出, 24VDC, 0.5A	CTH2 222-1DF32
SM222 数字量输出模块, 16 点晶体管输出, 24VDC, 0.5A(输出保护)	CTH2 222-1BH32
SM222 数字量输出模块, 32 点晶体管输出, 24VDC, 0.5A(输出保护)	CTH2 222-1BL32
SM222 数字量输出模块, 8 点继电器输出, 2A	CTH2 222-1HF32
SM222 数字量输出模块, 16 点继电器输出, 2A	CTH2 222-1HH32
SM223 数字量输入/输出模块, 4 点 24VDC 输入, 4 点晶体管输出, 24VDC, 0.5A(输出保护)	CTH2 223-1BF32
SM223 数字量输入输出模块, 8 点 24VDC 输入, 8 点晶体管输出, 24VDC, 0.5A(输出保护)	CTH2 223-1BH32
SM223 数字量输入输出模块, 16 点 24VDC 输入, 16 点晶体管输出, 24VDC, 0.5A(输出保护)	CTH2 223-1BL32
SM223 数字量输入输出模块, 4 点 24VDC 输入, 4 点继电器输出, 2A	CTH2 223-1HF32
SM223 数字量输入输出模块, 8 点 24VDC 输入, 8 点继电器输出, 2A	CTH2 223-1PH32
SM223 数字量输入输出模块, 16 点 24VDC 输入, 16 点继电器输出, 2A	CTH2 223-1PL32
SM231 模拟量输入模块, 4 点, 0~20 mA 电流或±5V, ±2.5V, 0~10V, 0~5V 电压输入, 隔离型 12 位精度	CTH2 231-0HC32
SM231 高精度模拟量输入模块, 8 点输入, 电压输入, 光偶隔离 16BIT 精度	CTH2 231-0HF32
SM231 高精度模拟量输入模块, 8 点输入, 电流输入, 光偶隔离 16BIT 精度	CTH2 231-1HF32
SM231 模拟量电压输入模块, 8 点, ±2.5V, 0~10V, 0~5V 电压输入且两通道可选 0~20mA 电流输入, 隔离型 12 位精度	CTH2 231-5HF32
SM231 热电阻温度输入模块, 2 点 RTD, 隔离型 16 位精度	CTH2 231-7PB32
SM231 热电阻温度输入模块, 4 点 RTD, 隔离型 16 位精度	CTH2 231-7PC32
SM231 热电偶温度输入模块, 4 点 TC, J/K/R/S/T/E/N, 隔离型 16 位精度	CTH2 231-7PD32
SM231 热电偶温度输入模块, 8 点 TC, J/K/R/S/T/E/N, 隔离型 16 位精度	CTH2 231-7PF32
SM231 热电偶 PID 模块, 4 点 J/K 型, 带智能 PID, 隔离型 16 位精度	CTH2 231-7TD32
SM231 热电偶 PID 模块, 8 点 J/K 型, 带智能 PID, 隔离型 16 位精度	CTH2 231-7TF32
SM231 8 路电流输入, 0-20mA/4-20mA 带智能 PID, 隔离型 16 位精度	CTH2 231-7HF32
SM231 混合温度输入模块, 2 点 NTC 或 PT100, 2 点 0~20mA 电流或±5V, ±10V, 0~10V, 0~5V 电压输入, 隔离型 16 位精度	CTH2 231-7ND32
SM231 热电阻温度输入模块, 8NTC/PT100, 隔离型 16 位精度	CTH2 231-7NF32
SM231 称重模块, 1 路传感器信输入, 采样频率 50Hz, 0.01%模块精度, 6VDC 每路 150MA 激励电源输出, 隔离型 16 位精度	CTH2 231-7WA32
SM232 模拟量输出模块, 2 点, ±10V 电压或 0~20mA 电流输出, 隔离型电压 12 位精度或电流 11 位精度	CTH2 232-0HB32

扩展模块	产品订货号
SM232 模拟量输出模块, 4 点, $\pm 10V$ 电压或 $0\sim 20mA$ 电流输出, 隔离型电压 12 位精度或电流 11 位精度	CTH2 232-0HD32
SM235 模拟量输入输出模块, 4 点电压或电流输入/1 点电压或电流输出, 隔离型电压 12 位精度或电流 11 位精度	CTH2 235-0KD32
SM253 运动控制模块, 2 路单相或 AB 相 HSC 输入, 200KHz, 2 轴 PTO/PWM 输出, 200KHz, 合信运动控制库	CTH2 253-1BH32
SM277A Profibus DP 从站接口模块, 12M 通信速率, 光电隔离	CTH2 277-0AA32
SM277B Profibus DP 从站模块, 1.5M 通信速率, 光电隔离	CTH2 277-0AB32
SM277C CAN 从站模块, 自带 8DI/6DO, 光电隔离, 可扩展 7 个模块	CTH2 277-0AC32
SM277 Profinet 从站模块, 100M 通信速率, 可扩展 8 个模块	CTH2 277-0PN32

产品名称与规格	产品订货号
扩展板	
EBH AMS-03 模拟量输入输出扩展板, 2 点*12 位精度电压输入, 1 点*12 位精度电压/电流输出	CTH2 AMS-03S1-EB
EBH-AMS-06 模拟量输入输出扩展板, 4 点*12 位精度电压输入, 2 点*12 位精度电压输出	CTH2 AMS-06S1-EB
EBH-AMS-06 模拟量输入输出扩展板, 4 点*12 位精度电压输入, 2 点*12 位精度电流输出	CTH2 AMS-06S2-EB
EBH CAN 主站通信扩展板, 1Mbps, 光电隔离, 支持 EasyCAN 组态	CTH2 CAN-01S1-EB
EBH CAN 主站通信扩展板, 1Mbps, 光电隔离, 支持 CANfree 和 CAN 硬件组态及第三方 EDS 文件导入	CTH2 CAN-01S2-EB
EBH PWM 04 充电导引扩展板, 双通道电压输入, 单通道可调脉宽 PWM 输出	CTH2 PWM-04S1-EB
一路 RS485 通信扩展板隔离型, 适用于全线高性能升级版型系列产品	CTH2 RS485-01S1-EB
配件	
PLC 锂电池	CTH2 291-8BA33
RS485 编程通讯电缆	CTS7 191-USB30
64K 存储卡(仅 H224X/ H226XL/ H226XM/ H228XL/ H226IM/ H226IL/ H226IH 支持)	CTS7 291-MC064
256K 存储卡(仅 H224X/ H226XL/ H226XM/ H228XL H226IM/ H226IL/ H226IH 支持)	CTS7 291-MC256
便携式编程卡	CTS7 291-PC001
便携式编程卡(新版 V2)	CTS7 291-PC002



服务热线
400-700-4858

-  **深圳市合信自动化技术有限公司**
-  深圳市南山区深圳国际创新谷 6 栋 A 座 9 层
-  0755-86226822
-  sales@co-trust.com
-  www.co-trust.com



官方微信公众号