

---

一个给用户提供自由扩展空间的控制器  
一个用户可自己设计外围模块的控制器

**SYNTRON**

# SC-U系列控制器

# 使用手册

( V 6 . 0 )

北京和利时电机技术有限公司

<http://www.syn-tron.com>

TEL: 010-62932100

## 前言

- 本资料包括 SC-U 系列控制器的硬件安装配线和使用要领。
- 在安装使用前，必须确认已仔细研究过本手册。
- 若有对本手册不理解的地方，请与我们联系，我们将给您详尽的解释。
- 有关该程控器标准装置的各种指令的解释及编程方法请参照《SC 系列控制器编程手册》
- 有关梯形图编辑环境的使用方法请参照《EasyLad 梯形图集成编辑器使用手册》

### 声明

在不于通知的情况下保留对本手册进行修改的权利。

## 目 录

第一章 概述.....	1
1.1 主要特点.....	1
1.2 型号命名方式.....	2
1.3 订货选型规格.....	3
1.4 编程连接和通讯地址设置.....	4
1.5 技术指标.....	5
第二章 主机硬件配置.....	7
2.1 外观尺寸.....	7
2.1.1 安装尺寸.....	7
2.1.2 外观介绍.....	7
2.1.3 通讯口管脚定义.....	8
2.1.4 主机端子定义.....	9
2.1.5 I/O 点的编号规则.....	10
2.2 输入滤波时间.....	10
2.3 输入电路的正确配线.....	11
2.4 输出电路的正确配线.....	13
2.4.1 继电器输出电路.....	13
2.4.2 晶体管输出电路.....	15
2.5 掉电检测电路.....	17
第三章 内部软元件介绍.....	18
3.1 输入/输出继电器 X、Y.....	18
3.2 辅助继电器 M.....	18
第四章 扩展功能模块.....	20
4.1 内置的通讯扩展口 COMM1.....	20
4.1.1 主要特点.....	20
4.1.2 如何进行通讯.....	20
4.1.3 通讯程序例子.....	29
4.2 16 路开关量输入模块 SC-U-16DI.....	30
4.2.1 主要特点.....	30
4.2.2 端子排列图.....	30
4.2.3 安装.....	30
4.2.4 输入电路的正确配线.....	31
4.3 16 路开关量输入输出模块 SC-U-16DIO.....	33
4.3.1 主要特点.....	33
4.3.2 端子排列图.....	33
4.3.3 安装.....	33
4.3.4 输入电路的正确配线.....	34
4.3.5 输出电路的正确配线.....	36
4.4 5 轴扩展定位模块 SC-U-5PTO.....	38
4.4.1 主要特点.....	38
4.4.2 端子排列图.....	38
4.4.3 安装.....	39

4.4.4	输入电路的正确配线.....	40
4.4.5	输出电路的正确配线.....	41
4.4.6	应用.....	42
4.5	4 通道模拟量输入模块 SC-U-4AI.....	52
4.5.1	主要特点.....	52
4.5.2	端子排列图.....	52
4.5.3	安装.....	53
4.5.4	读入模拟量值.....	53
4.6	4 通道模拟量输出模块 SC-U-4AO.....	55
4.6.1	主要特点.....	55
4.6.2	端子排列图.....	55
4.6.3	安装.....	56
4.6.4	输出模拟量的值.....	56
4.7	4 通道热电偶输入模块 SC-U-4EI.....	58
4.7.1	主要特点.....	58
4.7.2	端子排列图.....	58
4.7.3	安装.....	59
4.7.4	读入温度值.....	59
4.8	4 通道 PT100 热电阻输入模块 SC-U-4PT100.....	61
4.8.1	主要特点.....	61
4.8.2	端子排列图.....	61
4.8.3	安装.....	62
4.8.4	读入温度值.....	62
4.9	5 个通讯口扩展模块 SC-U-5COM.....	65
4.9.1	主要特点.....	65
4.9.2	端子排列图.....	65
4.9.3	模块安装和通讯口地址.....	66
4.9.4	如何进行通讯.....	66
4.9.5	扩展通讯口通讯程序例子.....	75
4.9.6	扩展通讯口 ModBus-RTU 主机模式通讯.....	75
4.9.7	扩展通讯口 ModBus-RTU 从机模式通讯.....	83
4.10	内置的特殊功能扩展板.....	85
4.10.1	主要特点.....	85
4.10.2	型号规格表.....	85
4.10.3	端子排列图.....	85
4.10.4	设置模拟板的 SPI 地址.....	86
4.10.5	如何输入输出模拟量值.....	86
4.11	远程 IO 扩展模块.....	87
4.11.1	主要特点.....	87
4.11.2	端子排列图.....	87
4.11.3	远程 I/O 点的编号规则.....	88
4.11.4	安装.....	88
4.11.5	设置远程 IO 扩展分机的分机地址.....	88
4.11.6	如何刷新远程 IO 模块.....	88

4.11.7 刷新远程 IO 模块梯形图例子 .....90

# 第一章 概述

SC-U 系列控制器是采用先进的电子技术、自动化技术、可靠性技术和开放式的设计思想设计而成的新一代控制器，具有极高的可靠性、极高的性能/价格比和极其灵活的扩展能力等特点，可适用于各行各业，各种场合中的检测、监测及控制的自动化。使用范围可覆盖从替代继电器的简单控制到更复杂的自动化控制。

SC-U 系列控制器价格低廉，功能强大，易学易用，其扩展组合方便、灵活。其本地开关量 I/O 点数从 14 点可扩展至 86 点（使用远程 I/O 模块可扩展至数百点），并具备可自动重装的高速计数及其高速输出、标准的 RS-485 接口、脉冲测量、实时时钟、程序加密及浮点运算等先进的功能，以及基于 WINDOWS 的全中文文化的梯形图编程环境。

SC-U 系列控制器广泛用于冶金、交通、化工、轻工、造纸、水处理等行业，以及食品与机械设备、制冷设备、医疗设备、电梯等等。

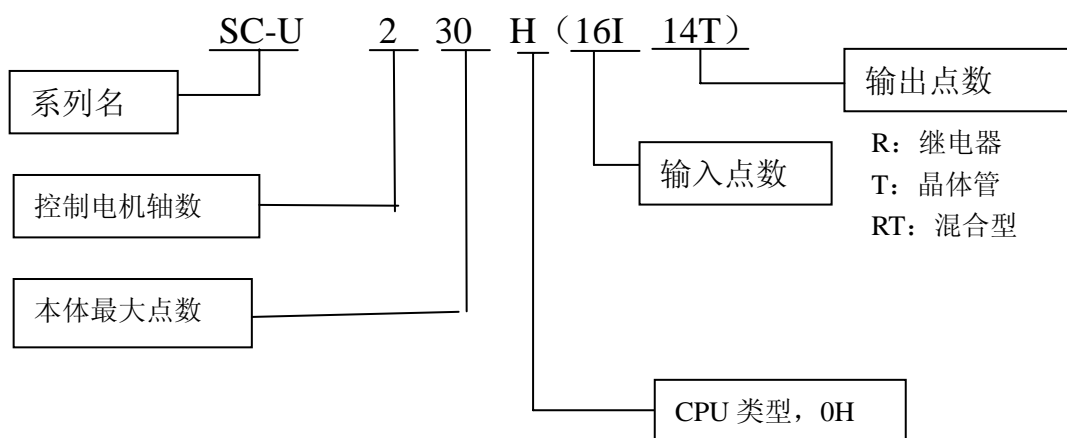
## 1.1 主要特点

- 具有 12K（CPU-0A）或 14K（CPU-0H）步的超大程序容量及独特的程序加锁保护功能，保密性极高，绝不可解密，并且无须电池维护。
- 采用结构化的梯形图编程语言，使您的程序更易编，易读，易懂。可进行变量、数组、结构体的声明和自动分配，在梯形图中直接使用算术表达式和函数来表示变量、常数之间的运算关系，使您很容易编写出复杂的数据运算程序。
- CPU-0A 系列内核具有 256 个字的非易失性数据存储器（掉电保持），并且不用电池，用户可在程序运行中使用指令或由通讯口对其进行读写。
- CPU-0H 系列内核具有 16K 个字的非易失性数据存储器（掉电保持），并且不用电池，用户可在程序运行中使用指令或由通讯口对其进行读写，最少可进行 100 万次的写入操作，数据可保存至少 100 年。也可选无读写次数限制的存储器。
- 具有大容量非易失性数据存储器扩展接口，可扩展非易失性数据存储器最大到 2M 字，用户可在程序运行中使用函数对其进行读写。（选件）
- 具有 7 个中断源：定时中断，高速计数器 C0 中断，高速计数器 C1 中断，高速计数器 C2 中断，外部输入 X3 边缘触发中断，外部输入 X4 边缘触发中断，PTO 脉冲串输出完成中断。其外部输入边缘触发中断可选择为正边缘触发或负边缘触发。
- 主机有 14 点（8 入 6 出）、30 点（16 入 14 出）两种机型。

- 丰富的扩展模块，有 16DI、16DIO-R、16DIO-T、模拟量模块、5 轴定位扩展模块、5 个通讯口扩展模块、高速计数器模块、远程 I/O 扩展模块等。
- 电源为交流 220V 的主机，还可以内置模拟量模块或特殊功能模块。
- 主机具有 3 路高速计数器，且均具有高速输出和中断功能，并可直接测量 3 路脉冲信号的脉冲宽度、周期和频率。
- 主机提供 2 路 PWM 或 2 路具有加减速功能的脉冲串定位控制输出（最高输出频率：CPU-0A 为 20KHz，CPU-0H 为 100KHz）。
- 主机提供 2 个 RS485 通讯口（1 个编程口，1 个可选的扩展口），内置有 Modbus-RTU 通讯协议和自由口通讯模式。
- 浮点运算功能。
- 内置有实时时钟（作为选件）。可提供年、星期、月、日、小时、分钟等信息。电池使用寿命可达 10 年。

## 1.2 型号命名方式

型号命名的基本格式为



### 1.3 订货选型规格

#### (1) 主机型号规格

型号	说明
SC-U214H (8I6T)	CPU-0H 主机, 晶体管输出, 8 入/6 出
SC-U230H (16I14T)	CPU-0H 主机, 晶体管输出, 16 入/14 出

#### (2) 扩展模块型号规格

型号	说明
SC-U-16DI	16 路开关量输入模块, 可内置 4 路模拟量 (选件)
SC-U-16DIO-T	8 路开关量输入/8 路晶体管输出模块, 可内置 4 路模拟量 (选件)
SC-U-5PTO	5 轴定位模块, 每轴最高输出频率: 100KHz (选件)
SC-U-5COM	5 个 RS485 (或 RS232) 通讯口扩展模块 (选件)
SC-U-4AI-V	4 路模拟量电压输入模块, 可内置另外 4 路模拟量 (选件)
SC-U-4AI-I	4 路模拟量电流输入模块, 可内置另外 4 路模拟量 (选件)
SC-U-4AO-V	4 路模拟量电压输出模块, 可内置另外 4 路模拟量 (选件)
SC-U-4EI	4 路热电偶输入模块, 可内置另外 4 路模拟量 (选件)
SC-U-4PT100	4 路 PT100 热电阻输入模块, 可内置另外 4 路模拟量 (选件)

#### (3) 输出形式的选择

继电器输出: 可用于交流负载或直流负载, 为有触点输出。

晶体管输出: 用于直流负载的高速无触点输出。

混合型输出: 在一个单元内提供继电器和晶体管两种形式的输出, 低半部分编号的为晶体管输出, 高半部分编号的为继电器输出, 晶体管都为 6 点。

特殊情况需用户订货提出。

#### (4) 输入形式的选择

各个单元输入形式均为直流 24V 输入, 输入信号可为无源触点或 NPN 型集电极开路输出。

其他形式的输入用户需订货提出。

#### (5) 内置特殊功能板选择

每个扩展模块 (5PTO 和 5COM 除外) 和电源为直流 24V 的主机, 则还有可选的内置模拟量板或特殊功能板, 内置有 **4AIB-V** (4 路电压输入)、**4AIB-I** (4 路电流输入)、**4EI**



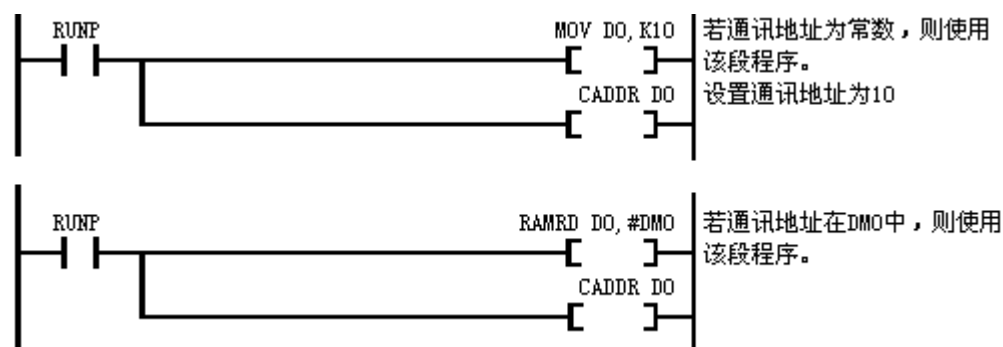
**B**（4 路热电偶输入）、**2EI2AOB**（2 路热电偶输入 2 路电压输出）、**2PT100**（2 路 PT100 热电阻输入）、**1PTO**（单轴扩展定位控制）、**COMM-232**（RS232 扩展通讯口）、**COMM-485**（RS485 扩展通讯口）等板选择。

## 1.4 编程连接和通讯地址设置

用户可通过以下设备编辑并向控制器下载程序：

- 1) 一台个人计算机(PC)或笔记本电脑和一套 EasyLad 梯形图集成编辑软件及 WIN95 以上的操作系统。
- 2) 一个具有全透明零延时转换的 RS232/RS485 转换器

若个人计算机和控制器为点对点连接，则在 EasyLad 中可把通讯的分机号设为 0，此时不管控制器的通讯地址如何，都可进行通讯。若个人计算机和控制器为点对多点连接即总线连接，则在 EasyLad 中分机号必须与要连接的控制器的通讯地址设为一致，而在控制器的用户程序中必须保证运行后执行过设置通讯地址的指令（CADDR）。下面是用梯形图设置通讯地址的例子：



用梯形图设置通讯地址

## 1.5 技术指标

CPU-0A 内核:

项 目	指 标	
操作控制方式	反复扫描程序	
I/O 控制方式	批处理（执行 END 指令时），并有 I/O 刷新指令	
程序扫描时间	约 15ms/K 步（位和字混合指令）	
编程语言	支持表达式和函数的梯形图 + 步进指令	
程序容量	12K 步（无需备用电池）	
指令数	逻辑：28 条，步进：3 条，功能：100 多条	
输入继电器（光电隔离）	24V DC/7mA	最大 48 点，X0-X57（8 进制）
输出继电器	继电器	250V AC/3A
	晶体管	24V DC/0.4A
辅助继电器	208 点，M0-M207（注：任何 1 个数据存储寄存器都可作为 16 个辅助继电器）	
定时器	10ms	24 点，T0-T23
	100ms	208 点，T0-T23，T272-T455（与 C272-C455 共用）
	200ms	184 点，T272-T455（与 C272-C455 共用）
	1s	184 点，T272-T455（与 C272-C455 共用）
	定时中断	1 个，时间：1ms-255ms
计数器	普通	208 点，C0-C23，C272-C455（与 T272-T455 共用）
	高速	10KHz:2 点，C0-C1；15KHz:1 点，C2
数据寄存器	临时寄存器	16 点，D0-D15
	特殊寄存器	1 点，D16
数据存储寄存器	非易失性	256 字，DM0~DM255，至少 100 年的数据保持时间
	易失性	200 字，DM256~DM455（注：DM256~DM271 为临时存储器）
子程序和自定义函数个数	无限制	
跳转标记个数	无限制	
中断源	7 个	
通讯	RS-485，波特率：19.2K、9600，内置 Modbus-RTU 协议	
工作环境温度	0~55℃	
电源	220V ± 20%	

CPU-0H 内核:

项 目	指 标	
操作控制方式	反复扫描程序	
I/O 控制方式	批处理（执行 END 指令时），并有 I/O 刷新指令	
程序扫描时间	约 2ms/K 步（位和字混合指令）	
编程语言	支持表达式和函数的梯形图 + 步进指令	
程序容量	14K 步（无需备用电池）	
指令数	逻辑：28 条，步进：3 条，功能：100 多条	
输入继电器（光电隔离）	24V DC/7mA	最大 48 点，X0-X57（8 进制）
输出继电器	继电器	250V AC/3A
	晶体管	24V DC/0.4A
辅助继电器	208 点，M0-M207（注：任何 1 个数据存储单元都可作为 16 个辅助继电器）	
定时器	10ms	1732 点，T0-T23，T272-T1979（与 C272-C1979 共用）
	100ms	1732 点，T0-T23，T272-T1979（与 C272-C1979 共用）
	200ms	1708 点，T272-T1979（与 C272-C1979 共用）
	1s	1708 点，T272-T1979（与 C272-C1979 共用）
	定时中断	1 个，时间：1ms-255ms
计数器	普通	1732 点，C0-C23，C272-C1979（与 T272-T1979 共用）
	高速	100KHz:3 点，C0、C1、C2
数据寄存器	临时寄存器	16 点，D0-D15
	特殊寄存器	1 点，D16
数据存储单元	非易失性	16K 字
	易失性	1724 字，DM256~DM1979（注：DM256~DM271 为临时存储器）
子程序和自定义函数个数	无限制	
跳转标记个数	无限制	
中断源	7 个	
通讯	RS-485，波特率：115.2K、38.4K、19.2K、9600，内置 Modbus-RTU 协议	
工作环境温度	0~55℃	
电源	220V ± 20%	

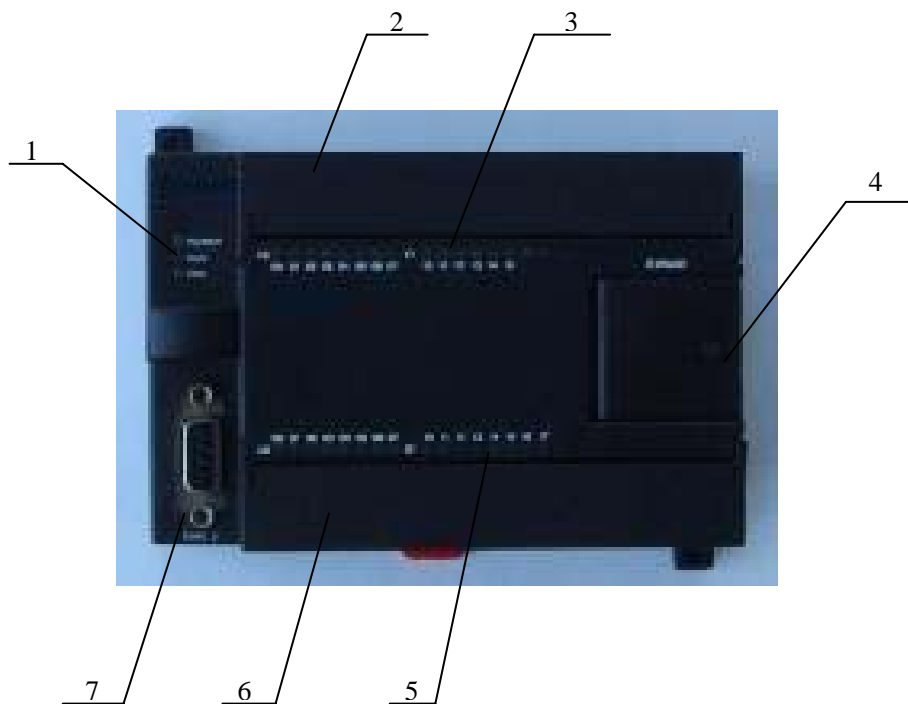
## 第二章 主机硬件配置

### 2.1 外观尺寸

#### 2.1.1 安装尺寸

主机：80mm×120mm×62mm。

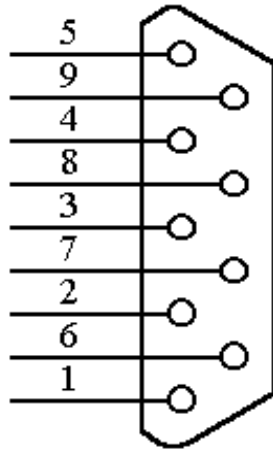
#### 2.1.2 外观介绍



- (1) — 状态指示灯（POWER、ERROR、STOP），亮为有效。
- (2) — 输出和电源端子排。
- (3) — 输出指示灯，亮为接通、灭为断开。
- (4) — 扩展插座，用于连接扩展单元来扩展 I/O 点数。
- (5) — 输入指示灯，亮为接通、灭为断开。
- (6) — 输入端子排。
- (7) — 通讯口，DB9 孔型插座。

### 2.1.3 通讯口管脚定义

通讯口采用 DB9 孔型插座，其管脚排列如下：



通讯口的管脚排列

- (1)——空。
- (2)—— GND，通讯地，该端与 CPU 共地。
- (3)—— D+，RS485 编程通讯口数据+（A 端）。
- (4)—— D1-，RS485 扩展通讯口 1 数据-（B 端）。
- (5)—— GND，通讯地，该端与 CPU 共地。
- (6)—— D1+，RS485 扩展通讯口 1 数据+（A 端）。
- (7)—— 空。
- (8)—— D-，RS485 编程通讯口数据-（B 端）。
- (9)—— 空。

### 2.1.4 主机端子定义

主机端子排列如下：

COM0	Y0	Y1	Y2	Y3	Y4	Y5	COM1	Y6	Y7	Y10	Y11	Y12	Y13	Y14	Y15	FG	L	N
X0	X1	X2	X3	X4	X5	X6	X7	X10	X11	X12	X13	X14	X15	X16	X17	COM	0V	+24

X0~X17: 开关量输入端。与 COM 端可接无源触点或 NPN 型集电极开路输出的传感器。

COM: 开关量输入的公共端。

0V: +24V 电源的 0V 端。

+24: +24V 电源输出端。

COM0: 输出端 Y0~Y5 的公共端。

Y0~Y5: 输出端。

COM1: 输出端子 Y6~Y15 的公共端。

Y6~Y15: 输出端。

FG: 接大地端。

L: 交流 220V 火线。

N: 交流 220V 零线。

### 2.1.5 I/O 点的编号规则

SC-U 系列控制器的 I/O 点均采用八进制编号，每 8 个为一组，编号中的最低位为该组中具体的 I/O 点号，编号中除了最低位的其他位为该组的组号，例如 X16：X 表示输入点，1 为该点所在的组号，6 为该点在该组中的点号。主机单元提供的 I/O 点具有固定的编号，开关量扩展单元的 I/O 点采用相对的编号，即该单元的 I/O 编号由该单元在 I/O 扩展链中的位置决定，具体来说扩展单元 I/O 编号的顺序为最靠近基本单元的编号为最小并从 X20 和 Y20 开始编号，以后依次递增。如下例：



扩展单元的 I/O 编号举例

若主机单元为 SC-U214H(8I6T)，扩展单元 1 为 SC-U-16DIO，扩展单元 2 为 SC-U-16DIO，则基本单元的 I/O 编号固定为 X0~X7、Y0~Y5，扩展单元 1 的 I/O 编号按顺序为 X20~X27、Y20~Y27，扩展单元 2 的 I/O 编号按顺序为 X30~X37、Y30~Y37。

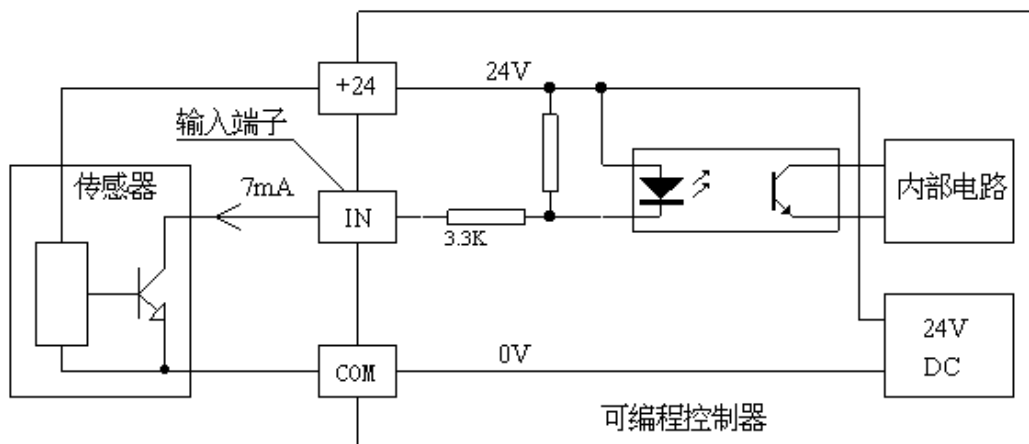
## 2.2 输入滤波时间

SC-U 系列控制器为每个数字量输入点均设置有硬件滤波器，对于噪声干扰具有更好的滤波效能。该滤波器的额定响应时间为 10ms，但 X0~X4 若用作高速输入功能如高速计数、脉冲测量、外部中断等时该滤波器自动设置为高速响应模式，此时的响应时间大约为 40us。

## 2.3 输入电路的正确配线

### (1) 输入电路特性

控制器信号输入电路如下：



输入电路

- 输入端子：
 

输入端子与 0V 端之间可接无源触点或 NPN 型集电极开路的输出方式的传感器。+24 端可作为传感器电源如上图所示。
- 输入电路：
 

输入电路的 1 次电路与 2 次电路用光电耦合器隔离，2 次电路中设有硬件滤波器，这是为了防止由于输入触点的颤振、输入线混入的噪声而引起误动作设计的，因此，输入从 ON→OFF，OFF→ON 变化时，控制器内部有约 10ms 的响应滞后。
- 输入灵敏度：
 

控制器的输入电流为 DC24V、7mA，引起输入动作的最小电流为 3~3.5mA，但为了确实启动，必须取 5mA 以上；为了确实切断，必须取 1.5mA 以下。因此，当输入触点串有二极管或电阻（不能完全启动），或者有并联电阻，或漏电流（不能完全切断）时，务请按下面的联接注意事项行事。
- 传感器用外部电路：
 

控制器的输入电流是有控制器内部的 DC24V 电源供给的。因此，光电开关等传感器用外部电源驱动时，该外部电源需为 DC24V±4V，传感器的输出晶体管需为 NPN 集电极开路型。



## (2) 联接注意事项

- 输入设备的选择

控制器的输入电流是 DC24V（内部供电）7mA，请使用与弱电流相适应的小型输入器件。若使用大电流的开关，将发生不正确的触点动作。

- 输入触点有串联二极管时，串联二极管的电压降应在约 4V 以下。因此如果串接带 LED 指示的开关时，串接数不要超过 2 个。

## 2.4 输出电路的正确配线

### (1) 输出特性

- 输出端子

在输出共用一个公共点的端子范围内，必须用同一电压类型和同电压等级，但不同的公共点组，可使用不同电压的负载。

负载驱动电源请用平滑的 DC5~26.4V。

- 电路隔离

控制器的内部电路与输出晶体管之间用光电耦合器进行电气隔离，且各输出公共点之间也相互隔离。

- 响应时间

从输出继电器的线圈通电或断电到输出触点 ON 或 OFF 的响应时间均为 50us。

- 输出电流

每个输出点：最大 0.4A。

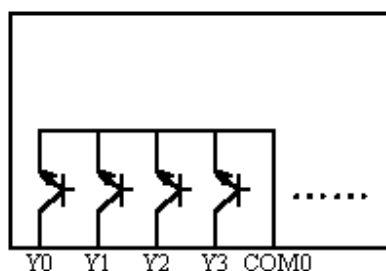
每个公共端：最大 4A。

注：输出晶体管为 NPN 型达林顿结构，其导通时约有 1~1.5V 的压降。

控制器内部没有输出短路保护，为了防止负载短路，用户需在外部设置保险丝。

### (2) 输出电路的构成

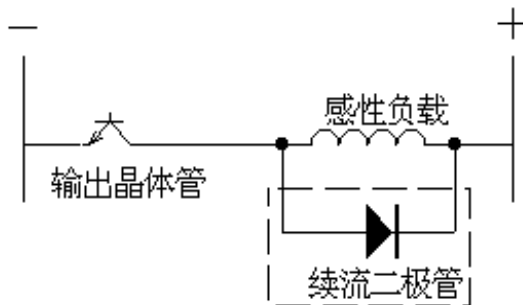
- 输出电路结构



晶体管输出电路结构

- 直流感性负载

晶体管输出内部电路包括了用于噪声吸收的齐纳二极管，但对于大电感或频繁开关的感性负载，还需在负载旁边并接续流二极管来防止击穿晶体管。



晶体管输出的直流感性负载

## 2.5 掉电检测电路

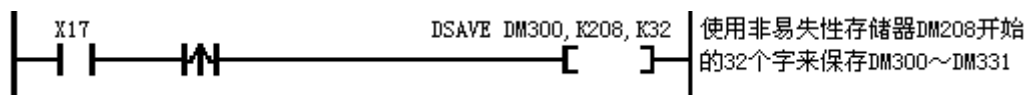
SC-U 系列控制器内置有掉电检测电路,用户可使用该电路在电源掉电时执行一些动作,如保存现场运行数据等。

该电路在控制器内部与 X17 连接(通过内部的拔码开关也可断开),当**电源掉电**时使 X17 接通(ON),当电源电压正常时使 X17 断开(若 X17 输入端没用)。注意因为使用普通输入进行掉电检测,其反应速度受用户程序的扫描周期影响。

出厂时,掉电检测电路设置为与 X17 连接,用户若要断开,则可打开上盖,找到拔码开关 S1,把第 2 位拔到 OFF 即可。

掉电检测应用例子:

要使易失性存储器 DM300~DM331 掉电保持,其梯形图例子(使用 X17 掉电检测):



使用 X17 检测电源掉电

## 第三章 内部软元件介绍

### 3.1 输入/输出继电器 X、Y

- 输入继电器 (X0~X57)

输入继电器采用八进制编号。

X0~X17 为主机开关量输入，X20~X57 为扩展模块开关量输入。

输入继电器既可以采用直接寻址方式如 X10，也可以采用寄存器位寻址方式如 RX 0.8。

输入继电器不能用程序驱动。

- 输出继电器 (Y0~Y47)

输出继电器采用八进制编号。

Y0~Y15 为主机开关量输出，Y20~Y47 为扩展模块开关量输出。

输出继电器的电子常开和常闭触点使用次数不限，在控制器中可自由使用，然而外部触点（输出元件）与内部触点的动作有所不同。

输出继电器既可以采用直接寻址方式如 Y10，也可以采用寄存器位寻址方式如 RY 0.8。

- 输入、输出继电器的动作时序

- (1) 输入处理

程序执行前控制器的全部输入端子的通/断状态读入输入映像寄存器。

在程序执行中即使输入状态变化，输入映像寄存器的内容也不变。直到下一扫描周期的输入处理阶段才读入这变化。另外，输入触点从通（ON）→断（OFF）或从断（OFF）→通（ON）变化，到处于确定状态止，输入滤波器还有一响应延迟时间（约 10ms）。

- (2) 程序处理

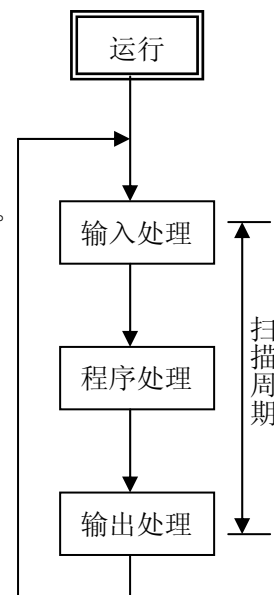
对应用户程序存储器所存的指令，从输入映像寄存器和其他软元件的映像寄存器中将有关软元件的通/断状态读出，从 0 步开始顺序运算，每次结果都写入有关的映像寄存器，因此，各软元件的映像寄存器的内容随着程序的执行在不断变化。

输出继电器的内部触点的动作由输出映像寄存器的内容决定。

- (3) 输出处理

全部指令执行完毕，将输出 Y 的映像寄存器的通/断状态向输出锁存寄存器传送，成为控制器的实际输出。

控制器内的外部输出触点对输出软元件的动作有一个响应时间，即要有一个延迟才动作。



## 3.2 辅助继电器 M

辅助继电器采用十进制编号。

辅助继电器为 208 点，其中通用辅助继电器 176 点，特殊辅助继电器 32 点。

辅助继电器为内部标志位存储器，可作为控制继电器存储中间操作状态或其他的控制信息，其线圈与输出继电器一样，由控制器内部各软元件的触点驱动。辅助继电器的电子常开和常闭触点使用次数不限，在控制器内可以自由使用。但是这些触点不能直接驱动外部负载，外部负载的驱动必须由输出继电器实行。

辅助继电器既可以采用直接寻址方式如 M8，也可以采用寄存器位寻址方式如 RM0.8。

- 通用辅助继电器 (M0~M175)

其功能可由用户自己定义。

- 特殊辅助继电器 (M176~M207)

这类继电器均有特殊的用途。其功能由控制器内部所定义，用户自己不能定义。

M176~M179 为系统保留，用户不要使用。

每一个特殊辅助继电器均有一个名称（标识符）来表示。如 RUN 表示运行继电器 M207。

特殊辅助继电器表：（1—ON，0—OFF）

名称	编号	功 能
RUN	M207	当控制器为 RUN 状态时始终 1 (ON)，为 STOP 状态时为 0 (OFF)
ERR	M206	控制器正确/错误,当控制器出现错误时为 1，否则为 0
RUNP	M205	RUN 脉冲，仅在程序开始运行后的第 1 个扫描周期为 1
CLK50	M204	50HZ 时钟，周期为 20ms
CLK5	M203	5HZ 时钟，周期为 200ms
CF	M202	进位/借位继电器
OV	M201	溢出位继电器
HC01M	M200	为 1 时，选 C0、C1 为 A-B 相高速计数；为 0 时，选 C0、C1 为单相高速计数
HSOE2	M199	高速 C2 高速输出控制继电器，为 1 时允许高速 C2 执行高速输出，为 0 时禁止
HSOE1	M198	高速 C1 高速输出控制继电器，为 1 时允许高速 C1 执行高速输出，为 0 时禁止
HSOE0	M197	高速 C0 高速输出控制继电器，为 1 时允许高速 C0 执行高速输出，为 0 时禁止
DF2M	M196	高速 C2 测频闸门时间选择，若 DF2M1 为 0，则为 0 时选 500ms，为 1 时选 50ms
HC2M	M195	高速 C2 自动重载/不重载选择，为 1 选择自动重载功能，为 0 选择不重载功能
HC2F	M194	当高速 C2 为自动重载计数器时，M194 为计数次数到标志，即每当高速 C2 减为 0 或刷新频率测量值时接通 M194，高速 C2 复位并不使 M94 为 0
SPIM1	M193	SPI 接口接收方式控制继电器 1
SPIM0	M192	SPI 接口接收方式控制继电器 0
LT	M191	小于标记，执行比较指令时若第 1 操作数小于第 2 操作数则为 1，否则为 0
EQ	M190	相等标记，执行比较指令时若两操作数相等则为 1，不相等则为 0
GT	M189	大于标记，执行比较指令时若第 1 操作数大于第 2 操作数则为 1，否则为 0
INT5M	M188	中断源 5 (X4 边缘触发中断) 模式，0 为正边缘触发，1 为负边缘触发
INT4M	M187	中断源 4 (X3 边缘触发中断) 模式，0 为正边缘触发，1 为负边缘触发
DF2M1	M186	高速 C2 测频闸门时间选择 1，为 1 时：若 DF2M 为 0 选 1s，为 1 选 100ms

HC2CK	M185	高速 C2 计数时钟选择，为 0 选择外时钟 (X2)，为 1 选择内时钟 (0.3us)
GF0	M184	通用标记 0，由用户设置或复位，可用作临时标记位
CBUSY	M183	通讯忙标记，当执行 NETR 或 NETW 通讯函数后为 1，通讯完成后为 0
CLK10	M182	10HZ 时钟，周期为 100ms
PTO1F	M181	当 Y1 的设定个数的脉冲串输出完成后置 1，由用户软件清 0
PTO0F	M180	当 Y0 的设定个数的脉冲串输出完成后置 1，由用户软件清 0
CLK1	M179	1HZ 时钟，周期为 1s

注：为了产品的兼容性，建议用户使用特殊辅助继电器的名称（标识符），而不要使用其编号。其他的特殊元件也如此。

其他的内部软元件见《SC 系列编程手册》。



## 第四章 扩展功能模块

### 4.1 内置的通讯扩展口 COMM1

#### 4.1.1 主要特点

- 内置在控制器主机内部。
- RS485 接口。
- 波特率：2400、4800、9600、19200、38400、57600、115200。数据位：7 位或 8 位。停止位：1 位。校验位：无校验、偶校验、奇校验。
- 模块内置 3 种数据块校验计算：累加和 (SUM)、异或和 (XOR)、CRC-16 (与 ModBus-RTU 校验方式兼容)。
- 提供有 ModBus-RTU 协议主站和从站通讯函数库。

#### 4.1.2 如何进行通讯

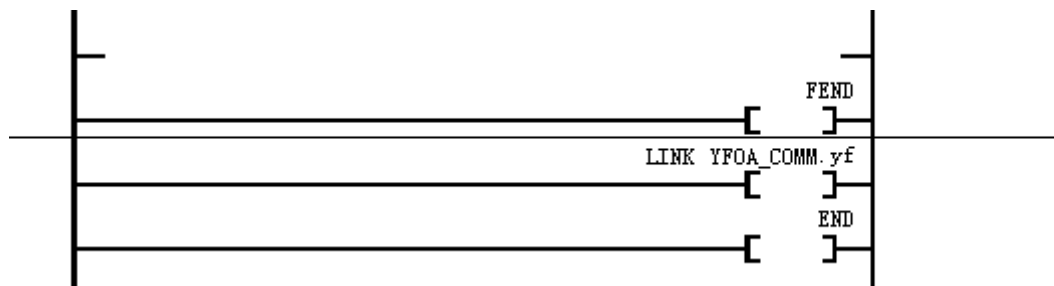
该扩展通讯口内置有 64 字节发送缓冲区(地址编号为 0~63)和 64 字节接收缓冲区(地址编号为 0~63)，二者完全独立，每帧数据最大可发送和接收 64 字节，CPU 模块通过向发送缓冲区写数据来设置要发送的内容，通过读接收缓冲区来读取接收到的数据。

用户需在自己的程序中连接扩展通讯口函数库 YF0A\_COMM.yf。在该函数库中提供了使用该模块所需要的函数。在编程软件 EasyLad 中的操作如下：

- 点击“工具”菜单，弹出以下菜单内容：



- 点击“添加函数库连接(L)”，弹出“打开文件”对话框，找到文件 YF0A\_COMM.yf 打开即可，连接后的程序如下：



函数库连接

函数库连接指令 (LINK) 必须位于主程序结束指令 (FEND) 的下面。

函数库文件 YF0A\_COMM.yf 与用户的梯形图程序文件应在同一个文件夹中。

**注：该扩展通讯口使用 SPI 接口扩展模块的地址 13，当控制器内置有扩展通讯口，则其他的特殊功能扩展模块的地址不要再设置为 13。**

(用户也可使用多模块扩展通讯口函数库 YF\_COMMx.yf，将该函数库中的各个函数的模块地址参数设置为 13 即可。函数库 YF\_COMMx.yf 使用说明见 4.10 节“5 个通讯口扩展模块 SC-U-5COM”)

在函数库 YF0A\_COMM.yf 中提供的函数如下：

- **通讯设置函数：Comm1Set**

*函数定义：*

FUN I, Mode As D2, RxTim As D3

*函数功能：*

设置工作模式、波特率和接收字符间隔超时时间。

*输入参数：*

**Mode:** 工作模式和波特率选择，**位 2--0:** 波特率选择，000—2400，001—2400，010—4800，011—9600，100—19200，101—38400，110—57600，111—115200；**位 4--3:** 校验位选择，00—无校验，01—偶校验，10—奇校验；**位 5:** 数据位位数，0 为 8 位数据，1 为 7 位数据。停止位固定为 1 位。若没有执行过该函数，则模块默认的波特率为 19200，无校验。

**RxTim:** 接收字符间隔超时时间 (0~255ms)，当设置为 0 时表示没有接收字符间隔超时检测。当接收到数据时将启动接收字符间隔超时检测，若在该时间内没有接收到下个数据，则置位接收字符间隔超时标记。

该函数调用例子：Comm1Set (H03, 10)

- **读通讯模块的状态字：Comm1Flag**

*函数定义：*

FUN I, Void As D0

*函数功能：*

读取通讯模块内的标记位：接收缓冲区校验比较结果标记、发送数据标记、接收字符间隔超时标记等。

*输入参数：*

**Void:** 无意义，可为任何值。

*返回值：*

通讯模块内的各个标记位（状态字）：

位 0：发送数据标记，为 1 表示正在发送数据块，为 0 表示发送端口空闲，数据块发送完成或无数据发送。

位 1：接收缓冲区数据块校验比较结果标记，当执行 Rx1Sum、Rx1Xor 或 Rx1CRC 函数后，为 1 表示接收缓冲区数据块的校验结果与收到的校验数据不相等，为 0 表示相等。该标记在执行 Rx1Sum、Rx1Xor 或 Rx1CRC 函数后才有效。

位 2：接收字符间隔超时标记。若接收字符间隔超时时间不为 0，则当接收到数据时将启动接收字符间隔超时检测，若在该时间内没有接收到下个数据，则该标记置位。当执行接收数据指针复位函数 Rx1RST(0)后该标记复位。

该函数调用例子：D0 = Comm1Flag ( 0 )

- **读校验计算结果寄存器的值：Comm1VerW**

*函数定义：*

FUN I, Void As D0

*函数功能：*

读取通讯模块内的校验计算结果寄存器的值。

*输入参数：*

**Void:** 无意义，可为任何值。

*返回值：*

校验计算结果寄存器的值（字）：

*说明：*

当执行了数据块校验计算函数(Tx1Sum、Tx1Xor、Tx1CRC、Rx1Sum、Rx1Xor、Rx1CRC)

后，会把计算结果（SUM 和 CRC-16 校验为 1 个字有效；XOR 校验为低字节有效、高字节为 0）放到校验计算结果寄存器中，用户使用该函数即可获得校验计算结果。

该函数调用例子：D0 = Comm1VerW ( 0 )

- **发送缓冲区写字节数据（字符）：Tx1Char**

*函数定义：*

FUN I, TxBufAddr As D2, Val As D3

*函数功能：*

向发送缓冲区中指定的地址写一个字节的的数据。

*输入参数：*

**TxBufAddr：**发送缓冲区中要写的地址（0~63）。

**Val：**要写入的数据，其低字节被写入到发送缓冲区中。

该函数调用例子：Tx1Char( 0, 123 )

- **发送缓冲区写字数据（2 个字节）：Tx1Word**

*函数定义：*

FUN I, TxBufAddr As D2, Val As D3

*函数功能：*

向发送缓冲区中指定的地址写一个字（2 个字节）的数据。

*输入参数：*

**TxBufAddr：**发送缓冲区中要写的首地址（0~63）。

**Val：**要写入的数据，其高字节被写入到 TxBufAddr 的地址，低字节被写入到 TxBufAddr+1 的地址。

该函数调用例子：Tx1Word ( 0, 1234 )

- **发送缓冲区写双字数据（4 个字节）：Tx1DINT**

*函数定义：*

FUN I, TxBufAddr As D2, Val As LDM256

*函数功能：*

向发送缓冲区中指定的地址写一个双字（4 个字节）的数据。

*输入参数：*

**TxBufAddr：**发送缓冲区中要写的首地址（0~63）。

**Val：**要写入的双字数据，其高字的高字节被写入到 TxBufAddr 的地址，高字的低字节

被写入到 TxBufAddr+1 的地址，低字的高字节被写入到 TxBufAddr+2 的地址，低字的低字节被写入到 TxBufAddr+4 的地址。

该函数调用例子：Tx1DINT ( 0, 12345678 )

- **发送缓冲区求累加和校验：Tx1Sum**

*函数定义：*

FUN I, TxBufAddr As D2, Len As D3, OutAddr As D4

*函数功能：*

把发送缓冲区中指定的数据块进行累加和校验计算，并把计算结果放入到发送缓冲区中指定的地址和校验计算结果寄存器中。

*输入参数：*

**TxBufAddr:** 发送缓冲区中要参与累加和计算的数据块的首地址（0~63）。

**Len:** 发送缓冲区中要参与累加和计算的数据块的长度。

**OutAddr:** 校验计算结果（低字节）要放入到发送缓冲区中指定的地址（0~63）。

该函数调用例子：Tx1Sum ( 0, 10, 10 )

- **发送缓冲区求异或和校验：Tx1Xor**

*函数定义：*

FUN I, TxBufAddr As D2, Len As D3, OutAddr As D4

*函数功能：*

把发送缓冲区中指定的数据块进行异或和校验计算，并把计算结果放入到发送缓冲区中指定的地址和校验计算结果寄存器中。

*输入参数：*

**TxBufAddr:** 发送缓冲区中要参与异或和计算的数据块的首地址（0~63）。

**Len:** 发送缓冲区中要参与异或和计算的数据块的长度。

**OutAddr:** 校验计算结果（低字节）要放入到发送缓冲区中指定的地址（0~63）。

该函数调用例子：Tx1Xor ( 0, 10, 10 )

- **发送缓冲区求 CRC-16 校验：Tx1CRC**

*函数定义：*

FUN I, TxBufAddr As D2, Len As D3, OutAddr As D4

*函数功能：*

把发送缓冲区中指定的数据块进行 CRC-16（ModBus-RTU 兼容）校验计算，并把计算

结果放入到发送缓冲区中指定的地址和校验计算结果寄存器中。

*输入参数:*

**TxBufAddr:** 发送缓冲区中要参与 CRC-16 计算的数据块的首地址 (0~63)。

**Len:** 发送缓冲区中要参与 CRC-16 计算的数据块的长度。

**OutAddr:** 校验计算结果 (1 个字) 要放入到发送缓冲区中指定的地址 (0~63), 高字节放入到 OutAddr 的地址, 低字节放入到 OutAddr+1 的地址。

该函数调用例子: Tx1CRC ( 0, 10, 10 )

- **发送数据: Tx1Out**

*函数定义:*

FUN I, TxBufAddr As D2, Len As D3

*函数功能:*

把发送缓冲区中指定的数据块的数据从通讯端口发送出去。

*输入参数:*

**TxBufAddr:** 发送缓冲区中要发送的数据块的首地址 (0~63)。发送时从首地址开始依次按字节发送。

**Len:** 发送缓冲区中要发送的数据块的长度 (按字节)。

该函数调用例子: Tx1Out ( 0, 12 )

- **读接收端口接收到的字节数: Rx1Num**

*函数定义:*

FUN I, Void As D0

*函数功能:*

读取接收端口接收到的字节数。

*输入参数:*

**Void:** 无意义, 可为任何值。

*返回值:*

接收端口接收到的字节数, 为 0 或小于 0 表示还没接收到数据, 为其他表示已经接收到的数据的字节数。该数据可通过执行 Rx1RST 函数复位为 0。

*说明:*

接收端口接收到的数据按先后顺序依次存放在从地址 0 开始的接收缓冲区中。

该函数调用例子: D0 = Rx1Num ( 0 )

- **接收端口接收数据指针复位函数：Rx1RST**

*函数定义：*

FUN I, Void As D0

*函数功能：*

把接收数据存放指针和接收字符间隔超时标记复位，同时也把接收到的字节数复位为 0，表示接收缓冲区要从地址 0 开始重新接收数据。

*输入参数：*

**Void:** 无意义，可为任何值。

该函数调用例子： Rx1RST(0)

- **按字节读接收缓冲区中的数据：Rx1Char**

*函数定义：*

FUN I, RxBufAddr As D2

*函数功能：*

按字节读取接收缓冲区中指定位置的数据。

*输入参数：*

**RxBufAddr:** 接收缓冲区中要读的数据的地址（0~63）。

*返回值：*

接收缓冲区中指定位置的数据（1 个字节）。

该函数调用例子： D0 = Rx1Char(0)

- **按字读接收缓冲区中的数据：Rx1Word**

*函数定义：*

FUN I, RxBufAddr As D2

*函数功能：*

按字读取接收缓冲区中指定位置的数据。

*输入参数：*

**RxBufAddr:** 接收缓冲区中要读的字数据的首地址（0~63）。

*返回值：*

接收缓冲区中指定位置的数据（1 个字），地址 RxBufAddr 中的数据为高字节，地址 RxBufAddr+1 中的数据为低字节。

该函数调用例子： D0 = Rx1Word(10)

- 按双字读接收缓冲区中的数据：**Rx1DINT**

*函数定义：*

FUN I, RxBufAddr As D2

*函数功能：*

按双字读取接收缓冲区中指定位置的数据。

*输入参数：*

**RxBufAddr:** 接收缓冲区中要读的双字数据的首地址（0~63）。

*返回值：*

接收缓冲区中指定位置的数据（双字），地址 **RxBufAddr** 中的数据为其高字的高字节，地址 **RxBufAddr+1** 中的数据为高字的低字节，地址 **RxBufAddr+2** 中的数据为低字的高字节，地址 **RxBufAddr+3** 中的数据为低字的低字节。

该函数调用例子：`LDM300 = Rx1DINT ( 10 )`

- 接收缓冲区求累加和校验：**Rx1Sum**

*函数定义：*

FUN I, RxBufAddr As D2, Len As D3, CMPAddr As D4

*函数功能：*

把接收缓冲区中指定的数据块进行累加和校验计算，并把计算结果放入到校验计算结果寄存器中，同时和接收缓冲区中指定的位置的接收到校验数据进行比较。

*输入参数：*

**RxBufAddr:** 接收缓冲区中要参与累加和计算的数据块的首地址（0~63）。

**Len:** 接收缓冲区中要参与累加和计算的数据块的长度。

**CMPAddr:** 校验计算结果（低字节）要比较的校验数据的地址（0~63）。比较结果可由 `Comm1Flag` 函数读取。

该函数调用例子：`Rx1Sum ( 0, 10, 10 )`

- 接收缓冲区求异或和校验：**Rx1Xor**

*函数定义：*

FUN I, RxBufAddr As D2, Len As D3, CMPAddr As D4

*函数功能：*

把接收缓冲区中指定的数据块进行异或和校验计算，并把计算结果放入到校验计算结果寄存器中，同时和接收缓冲区中指定的位置的接收到校验数据进行比较。



*输入参数:*

**RxBufAddr:** 接收缓冲区中要参与异或和计算的数据块的首地址 (0~63)。

**Len:** 接收缓冲区中要参与异或和计算的数据块的长度。

**CMPAddr:** 校验计算结果 (低字节) 要比较的校验数据的地址 (0~63)。比较结果可由 Comm1Flag 函数读取。

该函数调用例子: Rx1Xor ( 0, 10, 10 )

- **接收缓冲区求 CRC-16 校验: Rx1CRC**

*函数定义:*

FUN I, RxBufAddr As D2, Len As D3, CMPAddr As D4

*函数功能:*

把接收缓冲区中指定的数据块进行 CRC-16 (ModBus-RTU 兼容) 校验计算, 并把计算结果放入到校验计算结果寄存器中, 同时和接收缓冲区中指定的位置的接收到校验数据进行比较。

*输入参数:*

**RxBufAddr:** 接收缓冲区中要参与 CRC 计算的数据块的首地址 (0~63)。

**Len:** 接收缓冲区中要参与 CRC 计算的数据块的长度。

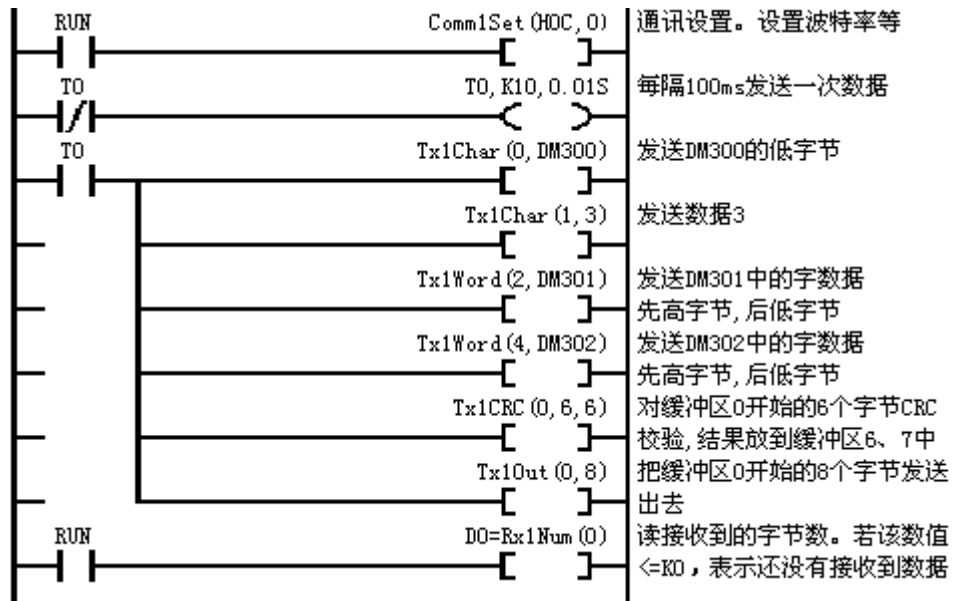
**CMPAddr:** 校验计算结果 (字) 要比较的校验数据 (2 个字节) 的首地址 (0~63)。

比较结果可由 Comm1Flag 函数读取。

该函数调用例子: Rx1CRC ( 0, 10, 10 )

### 4.1.3 通讯程序例子

梯形图的通讯例子：



扩展通讯口通讯例子

## 4.2 16 路开关量输入模块 SC-U-16DI

### 4.2.1 主要特点

- 每个模块可提供 16 路开关量输入。
- 每个模块还可以内置 4 路模拟量（选件），有 4AI-V（4 路电压输入）、4AI-I（4 路电流输入）、2AI2AO-V（2 路电压输入 2 路电压输出）、2AI2AO-I（2 路电流输入 2 路电压输出）、4EI（4 路热电偶输入）、2EI2AO（2 路热电偶输入 2 路电压输出）、2PT100（2 路 PT100 热电阻输入）等选择。
- 开关量输入采用光电隔离。
- 模块尺寸（W×H×D）为 71mm×80mm×62mm。

### 4.2.2 端子排列图

端子排列如下：

+24	X10	X11	X12	X13	X14	X15	X16	X17	MC1	MP2	MP3
+24	X0	X1	X2	X3	X4	X5	X6	X7	MC0	MP0	MP1

+24：为输入端 X0~X17（输入电路）提供的+24V 电源端，范围：12V~30V（2 个+24 端子都要接）。

X0~X17：开关量输入端。当该端与+24V 电源的 0V 端接通时为 ON，断开时为 OFF。该端与 0V 端之间可接无源触点或 NPN 型集电极开路输出方式的传感器。

MC0、MC1、MP0、MP1、MP2、MP3：内置的模拟量等特殊功能接线端，具体意义要看内置模拟量板的说明。

### 4.2.3 安装

#### (1) DIN 导轨安装方式

按 DIN46277 标准（宽 35mm）的 DIN 导轨中原封不动装入。

取出主体时，要从上方拉出 DIN 导轨钩子。

**注意：在有较强振动场合不要使用 DIN 导轨，而应直接安装**

#### (2) 直接安装方式

可使用单元上的 2 个安装孔（M4 螺丝孔）来进行直接安装。

### (3) 扩展模块的安装位置

开关量扩展模块应安装在 CPU 模块的右边，模拟量扩展模块应安装在开关量扩展模块的右边。

#### 注意事项

安装环境：

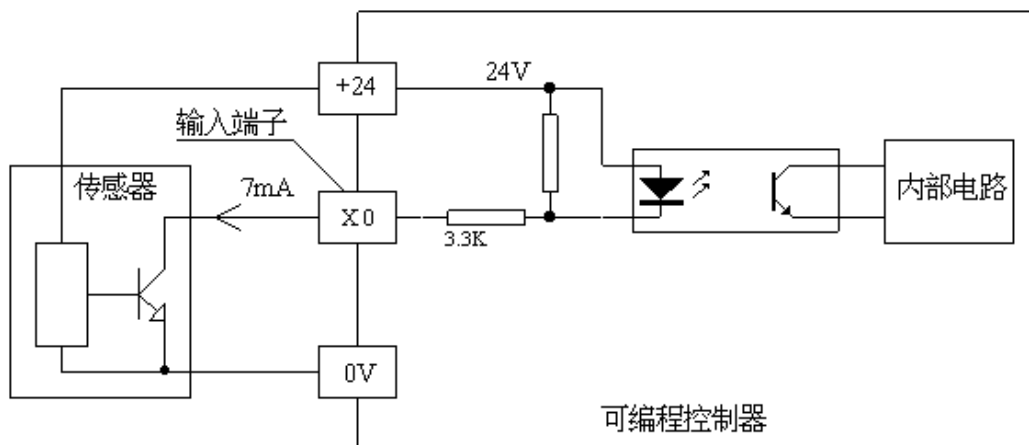
- 不宜安装在有灰尘、油烟、导电性粉尘、腐蚀性气体和可燃性气体的场合
- 不宜直接安装在有剧烈振动、冲击的场合
- 不要安装在高温、结露、雨淋的场合

安装施工

- 加工安装孔和进行配线施工时不要让粉末和电线头等杂物掉入可编程控制器内
- 单元与其他机器、构造物之间要留 50mm 以上的空间，并尽量与高压线、高压机器、动力机器隔开一定距离
- 信号线与动力线应分开配线，并尽量隔开一定距离

## 4.2.4 输入电路的正确配线

### (1) 输入电路特性



输入电路如下：

#### 输入电路

- 输入端子：

输入端子与+24V 电源的 0V 端之间可接无源触点或 NPN 型集电极开路的输出方式的传感器。当 X0 与 0V 端接通时表示为 ON；断开时表示为 OFF。

- 输入电路:

输入电路的 1 次电路与 2 次电路用光电耦合器隔离, 2 次电路中设有硬件滤波器, 这是为了防止由于输入触点的颤振、输入线混入的噪声而引起误动作设计的, 因此, 输入从 ON→OFF, OFF→ON 变化时, 控制器内部有约 10ms 的响应滞后。

输入电路的 1 次电路的电源电压范围为 12V~30V, 在此范围内均能正确工作, 但若传输线很长时, 为提高抗干扰性, 推荐采用 24V。

- 输入灵敏度:

控制器的输入电流为 DC24V、7mA, 引起输入动作的最小电流为 2.5~3mA, 但为了确实启动, 必须取 3mA 以上; 为了确实切断, 必须取 1.5mA 以下。

因此, 当输入触点串有二极管或电阻 (不能完全启动), 或者有并联电阻, 或漏电流 (不能完全切断) 时, 务请按下面的联接注意事项行事。

- 传感器用外部电路:

控制器的输入电流是由输入电路的电源供给的。

因此, 光电开关等传感器用外部电源驱动时, 该外部电源应与控制器输入电路的电源一致, 或者传感器的输出晶体管为 NPN 集电极开路型。

## (2) 联接注意事项

- 输入设备的选择

控制器的输入电流是 DC24V (内部供电) 7mA, 请使用与弱电流相适应的小型输入器件。若使用大电流的开关, 将发生不正确的触点动作。

### 4.3 16 路开关量输入输出模块 SC-U-16DIO

#### 4.3.1 主要特点

- 每个模块可提供 8 路开关量输入和 8 路开关量输出，开关量输出可为继电器或 NPN 型晶体管。
- 最大输出电流：24V/0.4A（晶体管输出），220VAC/2A（继电器输出）。
- 每个模块还可以内置 4 路模拟量（选件），有 4AI-V（4 路电压输入）、4AI-I（4 路电流输入）、2AI2AO-V（2 路电压输入 2 路电压输出）、2AI2AO-I（2 路电流输入 2 路电压输出）、4EI（4 路热电偶输入）、2EI2AO（2 路热电偶输入 2 路电压输出）、2PT100（2 路 PT100 热电阻输入）等选择。
- 开关量输入采用光电隔离，开关量输出采用继电器或光电隔离。
- 模块尺寸（W×H×D）为 71mm×80mm×62mm。

#### 4.3.2 端子排列图

端子排列如下：

COM	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	MC1	MP2	MP3
+24	X0	X1	X2	X3	X4	X5	X6	X7	MC0	MP0	MP1

**+24:** 为输入端 X0~X7（输入电路）提供的+24V 电源端，范围：12V~30V。

**X0~X7:** 开关量输入端。当该端与+24V 电源的 0V 端接通时为 ON，断开时为 OFF。该端与 0V 端之间可接无源触点或 NPN 型集电极开路输出方式的传感器。

**COM:** 输出端子 Y0~Y7 的公共端，该端最大电流为 8A。

**Y0~Y7:** 输出继电器的接线端。当输出继电器为 ON 时，则对应的端子与 COM 之间接通；当输出继电器为 OFF 时，则对应的端子与 COM 之间断开。

**MC0、MC1、MP0、MP1、MP2、MP3:** 内置的模拟量等特殊功能接线端，具体意义要看内置模拟量板的说明。

#### 4.3.3 安装

##### (1) DIN 导轨安装方式

按 DIN46277 标准（宽 35mm）的 DIN 导轨中原封不动装入。

取出主体时，要从上方拉出 DIN 导轨钩子。

**注意：在有较强振动场合不要使用 DIN 导轨，而应直接安装**

(2) 直接安装方式

可使用单元上的 2 个安装孔（M4 螺丝孔）来进行直接安装。

(3) 扩展模块的安装位置

开关量扩展模块应安装在 CPU 模块的右边，模拟量扩展模块应安装在开关量扩展模块的右边。

**注意事项**

安装环境：

- 不宜安装在有灰尘、油烟、导电性粉尘、腐蚀性气体和可燃性气体的场合
- 不宜直接安装在有剧烈振动、冲击的场合
- 不要安装在高温、结露、雨淋的场合

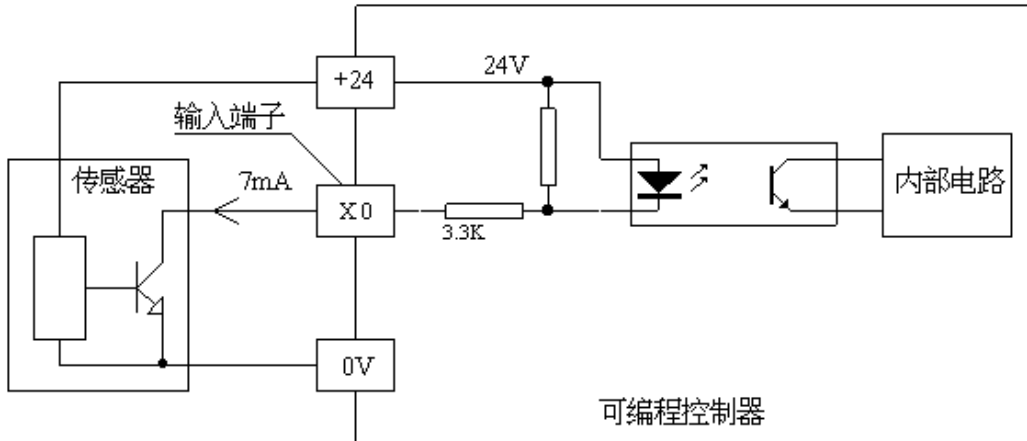
安装施工

- 加工安装孔和进行配线施工时不要让粉末和电线头等杂物掉入可编程控制器内
- 单元与其他机器、构造物之间要留 50mm 以上的空间，并尽量与高压线、高压机器、动力机器隔开一定距离
- 信号线与动力线应分开配线，并尽量隔开一定距离

#### 4.3.4 输入电路的正确配线

(1) 输入电路特性

输入电路如下：



输入电路

- 输入端子:

输入端子与+24V电源的0V端之间可接无源触点或NPN型集电极开路的输出方式的传感器。当X0与0V端接通时表示为ON；断开时表示为OFF。

- 输入电路:

输入电路的1次电路与2次电路用光电耦合器隔离，2次电路中设有硬件滤波器，这是为了防止由于输入触点的颤振、输入线混入的噪声而引起误动作设计的，因此，输入从ON→OFF，OFF→ON变化时，控制器内部有约10ms的响应滞后。

输入电路的1次电路的电源电压范围为12V~30V，在此范围内均能正确工作，但若传输线很长时，为提高抗干扰性，推荐采用24V。

- 输入灵敏度:

控制器的输入电流为DC24V、7mA，引起输入动作的最小电流为2.5~3mA，但为了确实启动，必须取3mA以上；为了确实切断，必须取1.5mA以下。

因此，当输入触点串有二极管或电阻（不能完全启动），或者有并联电阻，或漏电流（不能完全切断）时，务请按下面的联接注意事项行事。

- 传感器用外部电路:

控制器的输入电流是由输入电路的电源供给的。

因此，光电开关等传感器用外部电源驱动时，该外部电源应与控制器输入电路的电源一致，或者传感器的输出晶体管为NPN集电极开路型。

## (2) 联接注意事项

- 输入设备的选择



控制器的输入电流是 DC24V（内部供电）7mA，请使用与弱电流相适应的小型输入器件。若使用大电流的开关，将发生不正确的触点动作。

### 4.3.5 输出电路的正确配线

#### (1) 输出特性

- 输出端子

输出为每 8 点共用一个公共点，在输出共用一个公共点的端子范围内，必须用同一电压类型和同电压等级，但不同的公共点组，可使用不同电压的负载。负载驱动电源请用平滑的 DC5~26.4V。

- 电路隔离

控制器的内部电路与输出晶体管之间用光电耦合器进行电气隔离，且各输出公共点之间也相互隔离。

- 响应时间

从输出继电器的线圈通电或断电到输出触点 ON 或 OFF 的响应时间均为 50us。

- 输出电流

每个输出点：最大 0.4A。

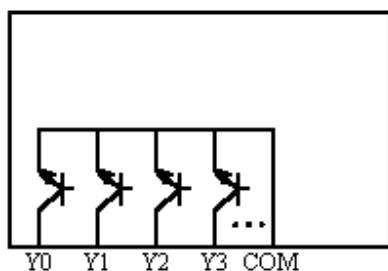
每个公共端：最大 8A。

注：输出晶体管为 NPN 型达林顿结构，其导通时约有 1~1.5V 的压降。

控制器内部没有输出短路保护，为了防止负载短路，用户需在外部设置保险丝。

#### (2) 输出电路的构成

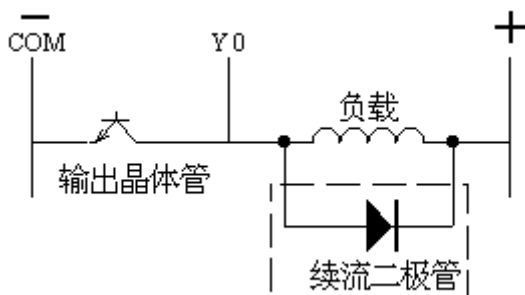
- 输出电路结构



晶体管输出电路结构

- 直流感性负载

晶体管输出内部电路包括了用于噪声吸收的齐纳二极管，但对于大电感或频繁开关的感性负载，还需在负载旁边并接续流二极管来防止击穿晶体管。



晶体管输出的负载接线方式

## 4.4 5 轴扩展定位模块 SC-U-5PTO

### 4.4.1 主要特点

- 一个模块上提供 5 轴定位控制输出(脉冲+方向), 每轴都有 2 路原点捕捉输入、3 种回原点方式。
- 每轴最大输出频率为 100KHz。
- 具有绝对定位、相对定位、回原点、点动、运行中变速等功能。
- 输出形式为 NPN 集电极开路输出, 最大输出电流: 24V/0.1A。
- 输入采用光电隔离, 输出采用光电隔离。
- 模块尺寸 (W×H×D) 为 71mm×80mm×62mm。

### 4.4.2 端子排列图

端子排列如下:

COM	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	MC1	MP2	MP3
+24	X0	X1	X2	X3	X4	X5	X6	X7	MC0	MP0	MP1

**+24:** 为输入端 X0~X7、MP0、MP1 (输入电路) 提供的+24V 电源端, 范围: 12V~30V。

**X0~X7、MP0、MP1:** 开关量输入端。当该端与+24V 电源的 0V 端接通时为 ON, 断开时为 OFF。该端与 0V 端之间可接无源触点或 NPN 型集电极开路输出方式的传感器。其中

X0: 第 1 轴回零的零位信号输入。

X1: 第 1 轴回零的零位信号使能输入 (或近零信号输入)。

X2: 第 2 轴回零的零位信号输入。

X3: 第 2 轴回零的零位信号使能输入 (或近零信号输入)。

X4: 第 3 轴回零的零位信号输入。

X5: 第 3 轴回零的零位信号使能输入 (或近零信号输入)。

X6: 第 4 轴回零的零位信号输入。

X6: 第 4 轴回零的零位信号使能输入 (或近零信号输入)。

MP0: 第 5 轴回零的零位信号输入。

MP1: 第 5 轴回零的零位信号使能输入 (或近零信号输入)。

COM: 输出端子 Y0~Y7、MP2、MP3 的公共端 (0V 端), 该端最大电流为 2A。

Y0~Y7、MP2、MP3: 输出接线端。为 NPN 集电极开路输出。其中

Y0: 第 1 轴的脉冲输出。

Y1: 第 1 轴的方向输出。

Y2: 第 2 轴的脉冲输出。

Y3: 第 2 轴的方向输出。

Y4: 第 3 轴的脉冲输出。

Y5: 第 3 轴的方向输出。

Y6: 第 4 轴的脉冲输出。

Y7: 第 4 轴的方向输出。

MP2: 第 5 轴的脉冲输出。

MP3: 第 5 轴的方向输出。

MC1: 为输出 Y0~Y7、MP2、MP3 提供的电源端, 范围: 5V~26.4V。

### 4.4.3 安装

#### (1) DIN 导轨安装方式

按 DIN46277 标准 (宽 35mm) 的 DIN 导轨中原封不动装入。

取出主体时, 要从上方拉出 DIN 导轨钩子。

**注意: 在有较强振动场合不要使用 DIN 导轨, 而应直接安装**

#### (2) 直接安装方式

可使用单元上的 2 个安装孔 (M4 螺丝孔) 来进行直接安装。

#### (3) 扩展模块的安装位置

该扩展模块应安装在 CPU 模块的右边。

#### (4) 模块地址

该模块的 SPI 地址固定占用 2、3、4、5、6, 因此用户的其他扩展模块的 SPI 地址不要再设置为这些数。

## 注意事项

安装环境：

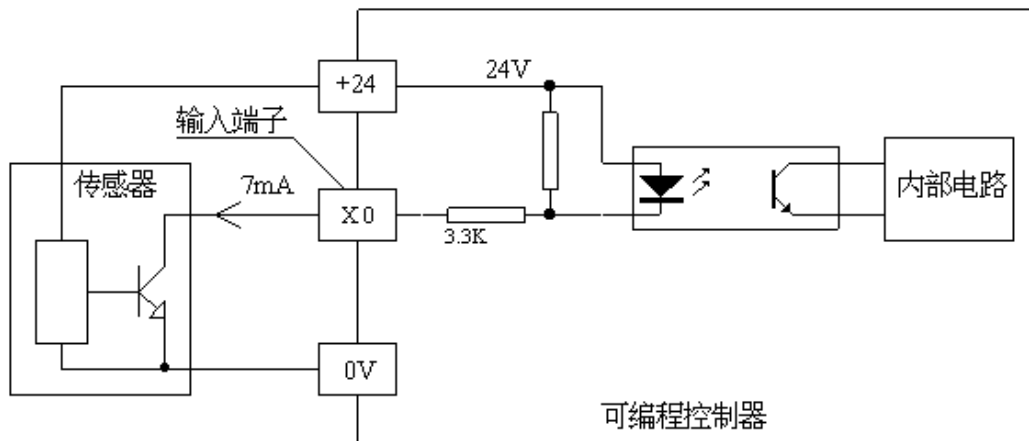
- 不宜安装在有灰尘、油烟、导电性粉尘、腐蚀性气体和可燃性气体的场合
- 不宜直接安装在有剧烈振动、冲击的场合
- 不要安装在高温、结露、雨淋的场合

安装施工

- 加工安装孔和进行配线施工时不要让粉末和电线头等杂物掉入可编程控制器内
- 单元与其他机器、构造物之间要留 50mm 以上的空间，并尽量与高压线、高压机器、动力机器隔开一定距离
- 信号线与动力线应分开配线，并尽量隔开一定距离

## 4.4.4 输入电路的正确配线

(1) 输入电路特性



输入电路如下：

## 输入电路

- 输入端子：

输入端子与+24V 电源的 0V 端之间可接无源触点或 NPN 型集电极开路的输出方式的传感器。当 X0 与 0V 端接通时表示为 ON；断开时表示为 OFF。

- 输入电路：

输入电路的 1 次电路与 2 次电路用光电耦合器隔离，1 次电路中设有硬件滤波器，这是为了防止由于输入触点的颤振、输入线混入的噪声而引起误动作设计的，响应频率约 3KHz。

输入电路的 1 次电路的电源电压范围为 12V~30V, 在此范围内均能正确工作, 但若传输线很长时, 为提高抗干扰性, 推荐采用 24V。

- 输入灵敏度:

控制器的输入电流为 DC24V、7mA, 引起输入动作的最小电流为 2.5~3mA, 但为了确实启动, 必须取 3mA 以上; 为了确实切断, 必须取 1.5mA 以下。

因此, 当输入触点串有二极管或电阻 (不能完全启动), 或者有并联电阻, 或漏电流 (不能完全切断) 时, 务请按下面的联接注意事项行事。

- 传感器用外部电路:

控制器的输入电流是由输入电路的电源供给的。

因此, 光电开关等传感器用外部电源驱动时, 该外部电源应与控制器输入电路的电源一致, 或者传感器的输出晶体管为 NPN 集电极开路型。

(2) 联接注意事项

- 输入设备的选择

控制器的输入电流是 DC24V (内部供电) 7mA, 请使用与弱电流相适应的小型输入器件。若使用大电流的开关, 将发生不正确的触点动作。

#### 4.4.5 输出电路的正确配线

(1) 输出特性

- 输出端子

所有输出共用一个公共点 COM, 必须用同一电压类型和同电压等级。

负载驱动电源请用平滑的 DC5~26.4V。

- 电路隔离

控制器的内部电路与输出晶体管之间用光电耦合器进行电气隔离。

- 输出电流

每个输出点: 最大 0.1A。

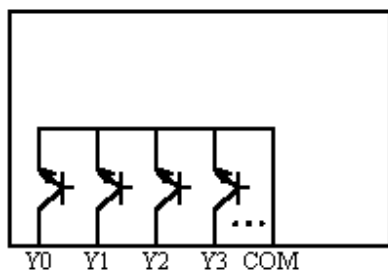
每个公共端: 最大 2A。

**注:** 输出晶体管为 NPN 型结构, 其导通时约有 0~0.7V 的压降。

控制器内部没有输出短路保护, 为了防止负载短路, 用户需在外部设置保险丝。

(2) 输出电路的构成

- 输出电路结构



晶体管输出电路结构

#### 4.4.6 应用

使用多模块脉冲串输出函数库 EPTO\_S.yf (多模块 S 曲线加减速脉冲串输出) 可以很方便的完成定位控制, 该函数库可支持 8 个模块、16 个轴的点到点定位控制。

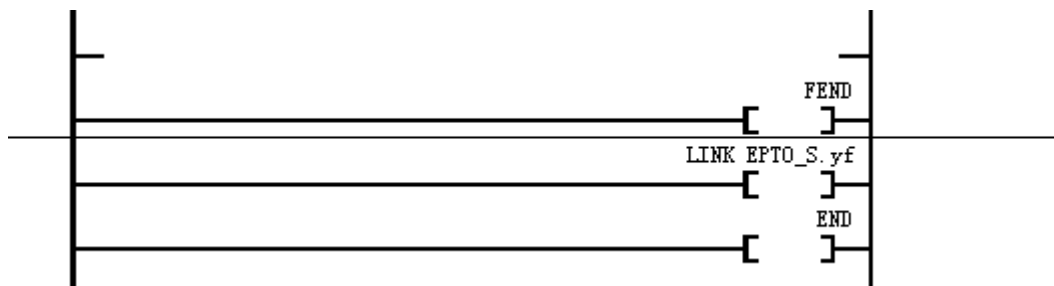
##### 第 1 步: 添加函数库连接

若要使用该函数库, 用户必须在自己的程序中添加函数库连接, 连接要使用的函数库文件。在编程软件 EasyLad 中的操作如下:

- 点击鼠标右键, 弹出以下菜单内容:



- 点击“添加函数库连接”，弹出“打开文件”对话框，找到文件 EPTO\_S.yf 打开即可，连接后的程序如下：



函数库连接

函数库连接指令（LINK）必须位于主程序结束指令（FEND）的下面。

## 第 2 步：定义函数库所需要的全局符号

若要使用函数库 EPTO\_S.yf，则用户必须定义以下全局符号：

**StopPosE[16]**：数组，类型为 LDM 或 DINT。各个轴的电机停止时的绝对位置（单位：脉冲个数）。不管用了几个轴，都要占用 16 个双字。StopPosE [0]为 SPI 地址 0 的 Y0 轴的，StopPosE [2] 为 SPI 地址 0 的 Y1 轴的，StopPosE [4]为 SPI 地址 1 的 Y0 轴的，StopPosE [6] 为 SPI 地址 1 的 Y1 轴的，StopPosE [8]为 SPI 地址 2 的 Y0 轴的，StopPosE [10]为 SPI 地址 2 的 Y2 轴的，……，StopPosE [28]为 SPI 地址 7 的 Y0 轴的，StopPosE [30]为 SPI 地址 7 的 Y1 轴的。SC-U-5PTO 模块使用 SPI 地址 2、3、4、5、6 的 Y0 轴。

**PTO\_DirE**：类型为 DM 或 INT。各个位为各个轴的电机方向，为 OFF 远离原点，为 ON 接近原点（坐标  $0 \sim +\infty$ ）。PTO\_DirE .0 为 SPI 地址 0 的 Y0 轴的，PTO\_DirE .1 为 SPI 地址 0 的 Y1 轴的，PTO\_DirE .2 为 SPI 地址 1 的 Y0 轴的，PTO\_DirE .3 为 SPI 地址 1 的 Y1 轴的，PTO\_DirE .4 为 SPI 地址 2 的 Y0 轴的，PTO\_DirE .5 为 SPI 地址 2 的 Y2 轴的，……，PTO\_DirE .14 为 SPI 地址 7 的 Y0 轴的，PTO\_DirE .15 为 SPI 地址 7 的 Y1 轴的。SC-U-5PTO 模块使用 SPI 地址 2、3、4、5、6 的 Y0 轴。

用户可使用全局符号表来定义上述全局符号（点击鼠标右键，从弹出菜单中选择“全局符号表”即可打开全局符号表），如下：

	符号名	类型或地址	描述
1	StopPosE[16]	LDM400	EPTO_S函数库所需全局符号：扩展定位模块各个轴的电机停止时的绝对位置
2	PTO_DirE	DM432	EPTO_S函数库所需全局符号：扩展定位模块各个轴的方向，为OFF远离原点

用户若需要在停电后记住电机的绝对位置，则应对 **StopPosE[16]**进行掉电保存（必须在电机停止后断电才能准确的记住绝对位置）。



EPTO\_S.yf 中有 8 个函数供用户定位控制使用。如下：

### ● 绝对定位函数 DRVAE

*函数定义：*

FUN I, SPI\_Addr As D0, Axis As D1, Pos As LDM256, RunFre As LDM258, StartFre As D2,  
AccelTim As D3

*函数功能：*

对某个模块中指定的轴电机按绝对位置来进行定位。

*输入参数：*

**SPI\_Addr:** SPI 地址 (2~6)，2、3、4、5、6 依次为第 1、第 2、第 3、第 4、第 5 轴。

**Axis:** 该 SPI 地址中指定的轴号，对于 SC-U-5PTO 模块该参数固定为 0。

**Pos:** 电机要移动到的绝对位置 (脉冲个数)。

**RunFre:** 电机的恒速运行脉冲频率 (62~100000Hz)。

**StartFre:** 电机的起始运行脉冲频率 (62~32767Hz)。

**AccelTim:** S 曲线加 (减) 速段每台阶的运行时间 (为值×128us)。

*调用形式：*

DRVAE(SPI\_Addr, Axis, Pos, RunFre, StartFre, AccelTim)

*说明：*

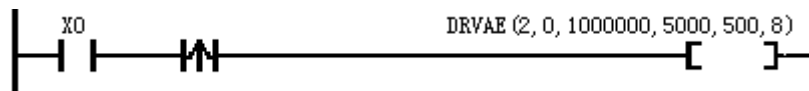
执行该函数后脉冲串输出完成标记被复位，到位置后被置位。

注：执行该函数时电机必须处于停止状态 (无脉冲输出)，执行该函数后电机立刻运行 (输出脉冲)。

例如：

DRVAE(2,0,1000,5000,500,8)表示把第 1 轴电机移动到绝对位置 1000 处，恒速运行脉冲频率为 5000Hz，起始频率为 500Hz，每台阶运行时间大约为 1ms。

梯形图例子如下：



## ● 相对定位函数 DRVIE

*函数定义:*

FUN I, SPI\_Addr As D0, Axis As D1, PNum As LDM256, RunFre As LDM258, StartFre As D2, AccelTim As D3

*函数功能:*

对某个模块中指定的轴电机按相对位置来进行定位。

*输入参数:*

**SPI\_Addr:** SPI 地址 (2~6), 2、3、4、5、6 依次为第 1、第 2、第 3、第 4、第 5 轴。

**Axis:** 该 SPI 地址中指定的轴号, 对于 SC-U-5PTO 模块该参数固定为 0。

**PNum:** 电机要移动到的相对位置 (相对于当前位置的脉冲个数), 为正数表示电机远离原点, 为负数表示接近原点。

**RunFre:** 电机的恒速运行脉冲频率 (62~100000Hz)。

**StartFre:** 电机的起始运行脉冲频率 (62~32767Hz)。

**AccelTim:** S 曲线加 (减) 速段每台阶的运行时间 (为值×128us)。

*调用形式:*

DRVIE(SPI\_Addr, Axis, PNum, RunFre, StartFre, AccelTim)

*说明:*

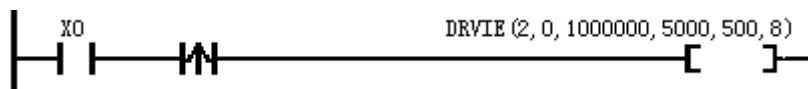
执行该函数后脉冲串输出完成标记被复位, 到位置后被置位。

注: 执行该函数时电机必须处于停止状态 (无脉冲输出), 执行该函数后电机立刻运行 (输出脉冲)。

例如:

DRVIE(2,0,1000,5000,500,8)表示把第 1 轴电机移动到相对于当前位置 1000 处, 恒速运行脉冲频率为 5000Hz, 起始频率为 500Hz, 每台阶运行时间大约为 1ms。

梯形图例子如下:



## ● 电机停止函数 DRVSE

*函数定义:*

FUN I, SPI\_Addr As D0, Axis As D1

### 函数功能:

通过程序来使某个模块中指定的轴电机停止，执行该函数后电机立刻进行减速停止。

### 输入参数:

**SPI\_Addr:** SPI 地址 (2~6)，2、3、4、5、6 依次为第 1、第 2、第 3、第 4、第 5 轴。

**Axis:** 该 SPI 地址中指定的轴号，对于 SC-U-5PTO 模块该参数固定为 0。

### 调用形式:

DRVSE(SPI\_Addr, Axis)

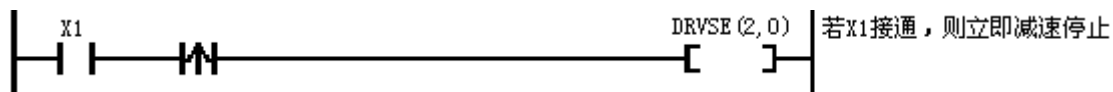
### 说明:

当电机减速停止后脉冲串输出完成标记被置位。

例如:

DRVSE(2,0)表示使第 1 轴电机减速停止

梯形图例子如下:



## ● 读出电机的当前位置函数 RdCPosE

### 函数定义:

FUN L, SPI\_Addr As D0, Axis As D1

### 函数功能:

返回某个模块中指定的轴电机的当前绝对位置（脉冲个数），无论电机处于运行还是停止状态。

### 输入参数:

**SPI\_Addr:** SPI 地址 (2~6)，2、3、4、5、6 依次为第 1、第 2、第 3、第 4、第 5 轴。

**Axis:** 该 SPI 地址中指定的轴号，对于 SC-U-5PTO 模块该参数固定为 0。

### 返回值:

该轴电机的当前绝对位置（脉冲个数）。

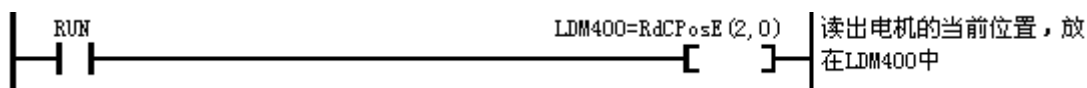
### 调用形式:

RdCPosE(SPI\_Addr, Axis)

例如:

LDM300=RdCPosE(2,0)表示把第 1 轴电机的当前位置读到 LDM300 中。

梯形图例子如下：



## ● 回零点函数 ZRNE

*函数定义：*

FUN I, SPI\_Addr As D0, Axis As D1, RunFre As LDM256, StartFre As D2, AccelTim As D3,  
ZRNPos As D4

*函数功能：*

使某个模块中指定的轴电机通过找原点输入信号回到原点。

*输入参数：*

**SPI\_Addr:** SPI 地址 (2~6)，2、3、4、5、6 依次为第 1、第 2、第 3、第 4、第 5 轴。

**Axis:** 该 SPI 地址中指定的轴号和回原点方式，H00 为 Y0 单输入方式 (X0 上升沿开始减速并走 ZRNPos 个脉冲停止)，H10 为 Y0 双输入方式 1 (X1 上升沿使能 X0 检测原点，此后当 X0 上升沿开始减速并走 ZRNPos 个脉冲停止)，H30 为 Y0 双输入方式 2 (X1 上升沿先减速，然后下降沿再使能 X0 检测原点，此后当 X0 上升沿后继续走 ZRNPos 个脉冲停止)。

**RunFre:** 电机的恒速运行脉冲频率 (62~100000Hz)。

**StartFre:** 电机的起始运行脉冲频率 (62~32767Hz)。

**AccelTim:** S 曲线加 (减) 速段每台阶的运行时间 (为值×128us)。

**ZRNPos:** 回原点中过零位传感器后的停机位置 (脉冲个数，范围：0~16383)，以该位置为原点。当回原点为单输入方式或双输入方式 1 时，该值应大于减速段的脉冲个数 (减速段脉冲个数可由编程软件 EasyLad 中的 PTO 定位控制计算工具算出)。

*调用形式：*

ZRNE(SPI\_Addr, Axis, RunFre, StartFre, AccelTim, ZRNPos)

*说明：*

执行该函数后脉冲串输出完成标记被复位，到位置后被置位。

若当前位置为正数或 0，则电机向反方向运行，若当前位置为负数，则电机向正方向运行。

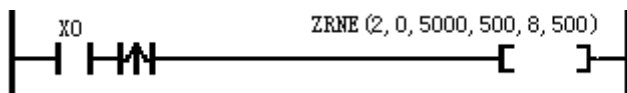
注：执行该函数时电机必须处于停止状态 (无脉冲输出)，执行该函数后电机立刻运行

(输出脉冲)。

例如：

ZRNE(2,0,5000,500,8,500)表示模块 2 中的第 1 轴电机以恒速运行脉冲频率为 5000Hz，起始频率为 500Hz，每台阶运行时间大约为 1ms 找原点，电机最终停在过零位传感器 500 个脉冲处（即以该位置为原点）。

梯形图例子如下：



## ● 点动函数 JOGE

*函数定义：*

FUN I, SPI\_Addr As D0, Axis As D1, JGF As D2.0, JGR As D3.0, RunFre As LDM256,  
StartFre As D4, AccelTim As D5

*函数功能：*

对某个模块中指定的轴电机来进行点动。

*输入参数：*

**SPI\_Addr:** SPI 地址 (2~6)，2、3、4、5、6 依次为第 1、第 2、第 3、第 4、第 5 轴。

**Axis:** 该 SPI 地址中指定的轴号，对于 SC-U-5PTO 模块该参数固定为 0。

**JGF:** 正转点动输入，ON 时正转，OFF 时停止。

**JGR:** 反转点动输入，ON 时反转，OFF 时停止。

**RunFre:** 电机的恒速运行脉冲频率 (62~100000Hz)。

**StartFre:** 电机的起始运行脉冲频率 (62~32767Hz)。

**AccelTim:** S 曲线加 (减) 速段每台阶的运行时间 (为值×128us)。

*调用形式：*

JOGE(SPI\_Addr, Axis, JGF, JGR, RunFre, StartFre, AccelTim)

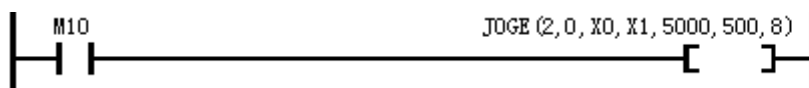
*说明：*

JGF 和 JGR 同时只能有一个为 ON，当都为 OFF 时才使电机停止。

例如：

JOGE(2,0,X0,X1,5000,500,8)表示对第 1 轴电机点动，X0 为正向点动，X1 为反向点动，运行频率为 5000Hz，起始频率为 500Hz，加减速每台阶时间大约为 1ms。

梯形图例子如下：（当 M10 为 ON 时允许点动）



## ● 变速函数 DRVVE

*函数定义：*

FUN I, SPI\_Addr As D0, Axis As D1, VelFre As LDM256, AccelTim As D2

*函数功能：*

在电机运行过程中改变运行速度。

*输入参数：*

**SPI\_Addr:** SPI 地址（2~6），2、3、4、5、6 依次为第 1、第 2、第 3、第 4、第 5 轴。

**Axis:** 该 SPI 地址中指定的轴号，对于 SC-U-5PTO 模块该参数固定为 0。

**VelFre:** 新的运行速度频率（62~100000Hz）。

**AccelTim:** 加（减）速每台阶的运行时间（为值×128us），若为 0 则没有加减速，若不  
为 0 则有加速或减速：若新速度大于原速度则加速，若新速度小于原速度则减速。

*调用形式：*

DRVVE(SPI\_Addr, Axis, VelFre, AccelTim)

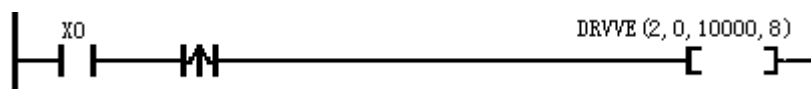
*说明：*

若该变速执行加速过程，则电机停止时会自动减速到该次变速前的速度停止。若该变速  
执行减速过程，则电机停止时不再自动减速停止。

例如：

DRVVE(2,0,10000,8)表示把正在运行的第 1 轴电机的速度变为 10000Hz，带加减速，加  
减速每台阶时间大约为 1ms。

梯形图例子如下：



## ● 读出电机的状态标记函数 RdDRVF

函数定义:

FUN I, SPI\_Addr As D0

函数功能:

返回电机的当前状态标记。

输入参数:

**SPI\_Addr:** SPI 地址 (2~6), 2、3、4、5、6 依次为第 1、第 2、第 3、第 4、第 5 轴。

返回值:

该 SPI 地址的电机的当前状态标记。**位 0:** 运行/停止标记, ON 为停止, OFF 为运行。

**位 4:** ON 时表捕捉到原点信号, 但并不表示电机已停止。**位 8:** 零位信号输入的状态。**位**

**10:** 零位信号使能输入 (近零信号输入) 的状态。

调用形式:

RdDRVF(SPI\_Addr)

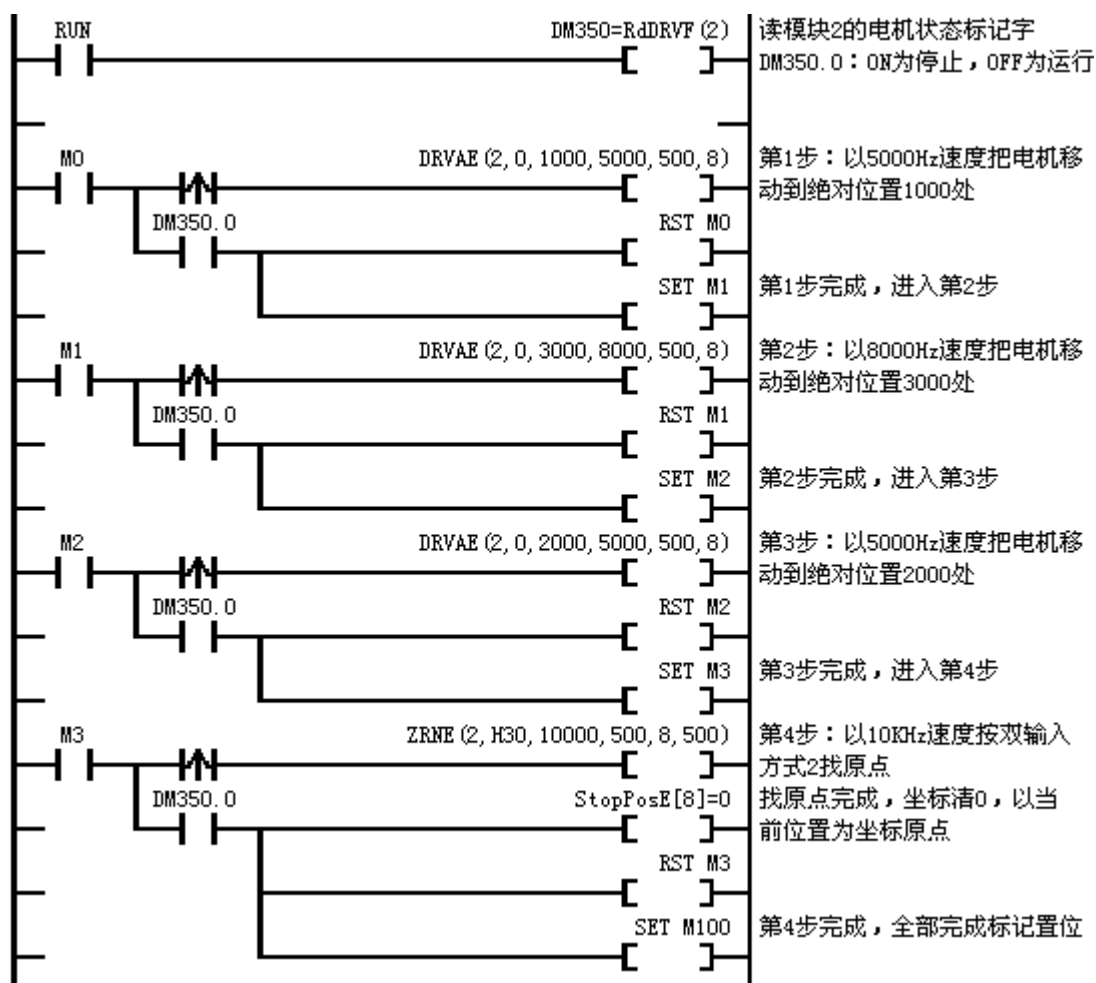
例如:

DM300=RdDRVF(2)表示把第 1 轴的状态标记读到 DM300 中。

梯形图例子如下:



扩展定位模块的定位控制梯形图例子：



扩展定位模块的定位控制例子



## 4.5 4 通道模拟量输入模块 SC-U-4AI

### 4.5.1 主要特点

- 每个模块可测量 4 通道 0~10V 电压或 0 (4) ~20mA 电流。
- 4 通道采样周期可选为：0.1 秒、50ms、20ms、10ms（须在定货时指明）。采样周期越长，则分辨率越高，抗干扰越好。
- 当采样周期为 0.1 秒时，分辨率高于 1/15000。
- 内置有微处理器，直接输出电压或电流值。
- 每个模块还可以内置另外 4 路模拟量（选件），有 4AI-V（4 路电压输入）、4AI-I（4 路电流输入）、2AI2AO-V（2 路电压输入 2 路电压输出）、2AI2AO-I（2 路电流输入 2 路电压输出）、4EI（4 路热电偶输入）、2EI2AO（2 路热电偶输入 2 路电压输出）、2PT100（2 路 PT100 热电阻输入）等选择。
- 模块尺寸（W×H×D）为 71mm×80mm×62mm。

### 4.5.2 端子排列图

端子排列如下：

COM	A0+	A0-	A1+	A1-	A2+	A2-	A3+	A3-	MC1	MP2	MP3
SPI 地址开关			X2	X3	X4	X5	X6	X7	MC0	MP0	MP1

**A0+**：模拟量输入通道 0 的输入电压正端（或电流输入端）。

**A0-**：模拟量输入通道 0 的输入电压负端。

**A1+**：模拟量输入通道 1 的输入电压正端（或电流输入端）。

**A1-**：模拟量输入通道 1 的输入电压负端。

**A2+**：模拟量输入通道 2 的输入电压正端（或电流输入端）。

**A2-**：模拟量输入通道 2 的输入电压负端。

**A3+**：模拟量输入通道 3 的输入电压正端（或电流输入端）。

**A3-**：模拟量输入通道 3 的输入电压负端。

**COM**：模拟电路的公共地。

**MC0、MC1、MP0、MP1、MP2、MP3**：内置的另外 4 路模拟量等特殊功能接线端，

具体意义要看内置模拟量板的说明。

### 4.5.3 安装

#### (1) DIN 导轨安装方式

按 DIN46277 标准（宽 35mm）的 DIN 导轨中原封不动装入。

取出主体时，要从上方拉出 DIN 导轨钩子。

**注意：在有较强振动场合不要使用 DIN 导轨，而应直接安装**

#### (2) 直接安装方式

可使用单元上的 2 个安装孔（M4 螺丝孔）来进行直接安装。

#### (3) 扩展模块的安装位置

开关量扩展模块应安装在 CPU 模块的右边，模拟量扩展模块应安装在开关量扩展模块的右边。

#### (4) 模块地址

在模块上的 4 位 SPI 地址拨码开关即为本模块的地址开关。可设置范围为 2~12、14。

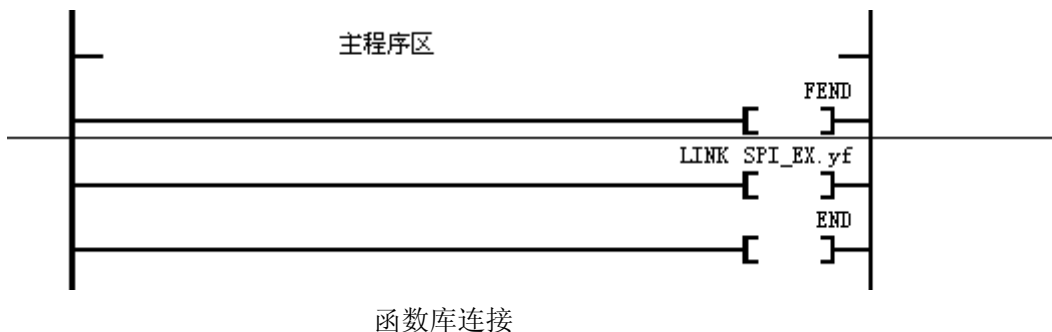
### 4.5.4 读入模拟量值

由于该模块采用 SPI 接口，因此用户需在自己的程序中连接 SPI 扩展模块函数库 SPI\_EX.yf。在该函数库中提供有基于 SPI 接口的扩展模块所需要的函数，如数字量输入、数字量输出、模拟量输入、模拟量输出等。在编程软件 EasyLad 中的操作如下：

- 点击“工具”菜单，弹出以下菜单内容：



- 点击“添加函数库连接 (L)”，弹出“打开文件”对话框，找到文件 SPI\_EX.yf 打开即可，连接后的程序如下：



函数库连接指令 (LINK) 必须位于主程序结束指令 (FEND) 的下面。

用户也可在函数库连接指令中省去路径名，如“LINK SPI\_EX.yf”，此时只要函数库文件 SPI\_EX.yf 与用户的梯形图程序文件在同一个文件夹中即可。

在该函数库中用于模拟量输入的函数如下：

- **模拟量输入函数：SPI\_AI**

*函数定义：*

FUN I, SPI\_Addr As D0, AI\_Addr As D1

*函数功能：*

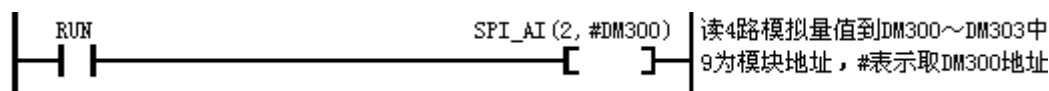
读入模拟量输入模块中的 4 路模拟量值存储到指定的数据块中。

*输入参数：*

**SPI\_Addr:** 模块地址。

**AI\_Addr:** 数据块的首地址 (DM 存储器)，占用 4 个字单元。若输入范围为 0~10V，则存储器中的值即为电压值 (单位：mV)。若输入范围为 0~20mA，则存储器中的值即为电流值 (单位：uA)。

该函数的调用例子如下：



模拟量输入函数 SPI\_AI 使用例子

**函数 SPI\_AI 应在主程序中使用 RUN 触点调用，以保证每个扫描周期都被执行。**

在上面的程序例子中，DM300 中为 A0 通道的电压值 (单位：mV)，DM301 中为 A1 通道的电压值 (单位：mV)，DM302 中为 A2 通道的电压值 (单位：mV)，DM303 中为 A3 通道的电压值 (单位：mV)。例如，若 DM300 的值为 1234，则表示 A0 通道的电压测量值为 1234mV。

## 4.6 4 通道模拟量输出模块 SC-U-4AO

### 4.6.1 主要特点

- 每个模块具有 4 路模拟量电压输出，输出范围为 0~10V。
- 分辨率为 8 位。
- 内置有微处理器，直接输出电压值。
- 每个模块还可以内置另外 4 路模拟量（选件），有 4AI-V（4 路电压输入）、4AI-I（4 路电流输入）、2AI2AO-V（2 路电压输入 2 路电压输出）、2AI2AO-I（2 路电流输入 2 路电压输出）、4EI（4 路热电偶输入）、2EI2AO（2 路热电偶输入 2 路电压输出）、2PT100（2 路 PT100 热电阻输入）等选择。
- 模块尺寸（W×H×D）为 71mm×80mm×62mm。

### 4.6.2 端子排列图

端子排列如下：

COM	A0+	A0-	A1+	A1-	A2+	A2-	A3+	A3-	MC1	MP2	MP3
SPI 地址开关			X2	X3	X4	X5	X6	X7	MC0	MP0	MP1

**A0+**：模拟量输出通道 0 的输出电压正端。

**A0-**：模拟量输出通道 0 的输出电压负端。

**A1+**：模拟量输出通道 1 的输出电压正端。

**A1-**：模拟量输出通道 1 的输出电压负端。

**A2+**：模拟量输出通道 2 的输出电压正端。

**A2-**：模拟量输出通道 2 的输出电压负端。

**A3+**：模拟量输出通道 3 的输出电压正端。

**A3-**：模拟量输出通道 3 的输出电压负端。

**COM**：模拟电路的公共地。

**MC0、MC1、MP0、MP1、MP2、MP3**：内置的另外 4 路模拟量等特殊功能接线端，具体意义要看内置模拟量板的说明。

### 4.6.3 安装

#### (1) DIN 导轨安装方式

按 DIN46277 标准（宽 35mm）的 DIN 导轨中原封不动装入。

取出主体时，要从上方拉出 DIN 导轨钩子。

**注意：在有较强振动场合不要使用 DIN 导轨，而应直接安装**

#### (2) 直接安装方式

可使用单元上的 2 个安装孔（M4 螺丝孔）来进行直接安装。

#### (3) 扩展模块的安装位置

开关量扩展模块应安装在 CPU 模块的右边，模拟量扩展模块应安装在开关量扩展模块的右边。

#### (4) 模块地址

在模块上的 4 位 SPI 地址拨码开关即为本模块的地址开关。可设置范围为 2~12、14。

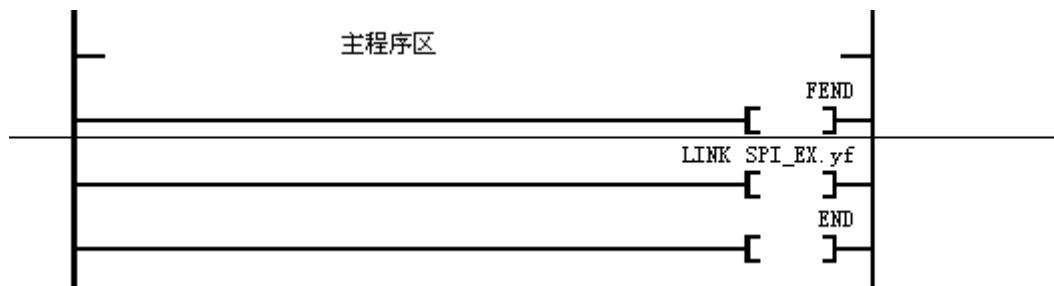
### 4.6.4 输出模拟量的值

由于该模块采用 SPI 接口，因此用户需在自己的程序中连接 SPI 扩展模块函数库 SPI\_EX.yf。在该函数库中提供有基于 SPI 接口的扩展模块所需要的函数，如数字量输入、数字量输出、模拟量输入、模拟量输出等。在编程软件 EasyLad 中的操作如下：

- 点击“工具”菜单，弹出以下菜单内容：



- 点击“添加函数库连接 (L)”，弹出“打开文件”对话框，找到文件 SPI\_EX.yf 打开即可，连接后的程序如下：



函数库连接

函数库连接指令 (LINK) 必须位于主程序结束指令 (FEND) 的下面。

用户也可在函数库连接指令中省去路径名，如“LINK SPI\_EX.yf”，此时只要函数库文件 SPI\_EX.yf 与用户的梯形图程序文件在同一个文件夹中即可。

在该函数库中用于模拟量输出的函数如下：

- **模拟量输出函数：SPI\_AO**

*函数定义：*

FUN I, SPI\_Addr As D0, AO\_Addr As D1

*函数功能：*

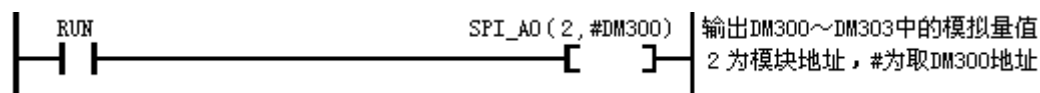
把指定的数据块中所存的模拟量值 (mV) 输出到模拟量输出模块的 4 路输出通道中。

*输入参数：*

**SPI\_Addr:** 模块地址。

**AO\_Addr:** 数据块的首地址 (DM 存储器)，占用 4 个字单元。分别对应于模拟量输出通道 0~3 要输出的值,范围：0~10000，单位：mV。

该函数的调用例子如下：



输出模拟量的值

**函数 SPI\_AO 应在主程序中使用 RUN 触点调用，以保证每个扫描周期都被执行。**

在上面的程序例子中，DM300 中为 A0 通道的电压输出值 (单位：mV)，DM301 中为 A1 通道的电压输出值 (单位：mV)，DM302 中为 A2 通道的电压输出值 (单位：mV)，DM303 中为 A3 通道的电压输出值 (单位：mV)。例如，若 DM300 的值为 1234，则表示 A0 通道的电压输出值为 1234mV。

## 4.7 4 通道热电偶输入模块 SC-U-4EI

### 4.7.1 主要特点

- 每个模块可测量 4 通道 K 型热电偶，分辨率为 0.1℃。
- 4 通道采样周期可选为：0.1 秒、50ms、20ms、10ms（须在定货时指明）。采样周期越长，则分辨率越高，抗干扰越好。
- 内置热电偶冷端自动补偿和热电偶断线报警。
- 内置有微处理器，直接输出线性化后的温度值。
- 每个模块还可以内置另外 4 路模拟量（选件），有 4AI-V（4 路电压输入）、4AI-I（4 路电流输入）、2AI2AO-V（2 路电压输入 2 路电压输出）、2AI2AO-I（2 路电流输入 2 路电压输出）、4EI（4 路热电偶输入）、2EI2AO（2 路热电偶输入 2 路电压输出）、2PT100（2 路 PT100 热电阻输入）等选择。
- 模块尺寸（W×H×D）为 71mm×80mm×62mm。

### 4.7.2 端子排列图

端子排列如下：

COM	A0+	A0-	A1+	A1-	A2+	A2-	A3+	A3-	MC1	MP2	MP3
SPI 地址开关			X2	X3	X4	X5	X6	X7	MC0	MP0	MP1

**A0+**：模拟量输入通道 0 的热电偶正端。

**A0-**：模拟量输入通道 0 的热电偶负端。

**A1+**：模拟量输入通道 1 的热电偶正端。

**A1-**：模拟量输入通道 1 的热电偶负端。

**A2+**：模拟量输入通道 2 的热电偶正端。

**A2-**：模拟量输入通道 2 的热电偶负端。

**A3+**：模拟量输入通道 3 的热电偶正端。

**A3-**：模拟量输入通道 3 的热电偶负端。

**COM**：模拟电路的公共地。

**MC0、MC1、MP0、MP1、MP2、MP3**：内置的另外 4 路模拟量等特殊功能接线端，

具体意义要看内置模拟量板的说明。

### 4.7.3 安装

#### (1) DIN 导轨安装方式

按 DIN46277 标准（宽 35mm）的 DIN 导轨中原封不动装入。

取出主体时，要从上方拉出 DIN 导轨钩子。

**注意：在有较强振动场合不要使用 DIN 导轨，而应直接安装**

#### (2) 直接安装方式

可使用单元上的 2 个安装孔（M4 螺丝孔）来进行直接安装。

#### (3) 扩展模块的安装位置

开关量扩展模块应安装在 CPU 模块的右边，模拟量扩展模块应安装在开关量扩展模块的右边。

#### (4) 模块地址

在模块上的 4 位 SPI 地址拨码开关即为本模块的地址开关。可设置范围为 2~12、14。

### 4.7.4 读入温度值

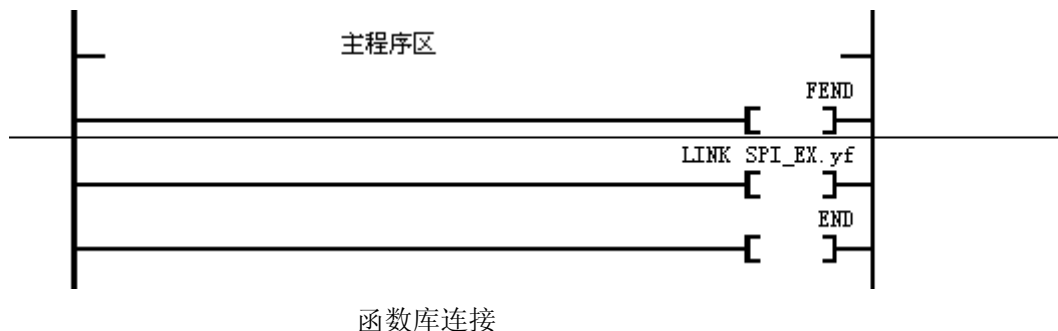
由于该模块采用 SPI 接口，因此用户需在自己的程序中连接 SPI 扩展模块函数库 SPI\_EX.yf。在该函数库中提供有基于 SPI 接口的扩展模块所需要的函数，如数字量输入、数字量输出、模拟量输入、模拟量输出等。在编程软件 EasyLad 中的操作如下：

- 点击“工具”菜单，弹出以下菜单内容：





- 点击“添加函数库连接 (L)”，弹出“打开文件”对话框，找到文件 SPI\_EX.yf 打开即可，连接后的程序如下：



函数库连接指令 (LINK) 必须位于主程序结束指令 (FEND) 的下面。

用户也可在函数库连接指令中省去路径名，如“LINK SPI\_EX.yf”，此时只要函数库文件 SPI\_EX.yf 与用户的梯形图程序文件在同一个文件夹中即可。

在该函数库中用于模拟量输入的函数如下：

- **模拟量输入函数：SPI\_AI**

*函数定义：*

FUN I, SPI\_Addr As D0, AI\_Addr As D1

*函数功能：*

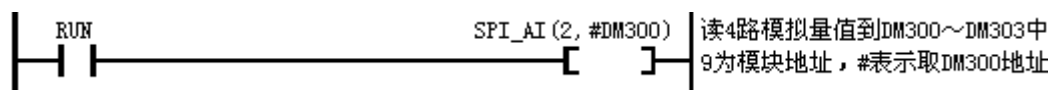
读入模拟量输入模块中的 4 路模拟量值存储到指定的数据块中。

*输入参数：*

**SPI\_Addr:** 模块地址。

**AI\_Addr:** 数据块的首地址 (DM 存储器)，占用 4 个字单元。

该函数的调用例子如下：



模拟量输入函数 SPI\_AI 使用例子

**函数 SPI\_AI 应在主程序中使用 RUN 触点调用，以保证每个扫描周期都被执行。**

在上面的程序例子中，DM300 中为 A0 通道的温度值 (单位：0.1℃)，DM301 中为 A1 通道的温度值 (单位：0.1℃)，DM302 中为 A2 通道的温度值 (单位：0.1℃)，DM303 中为 A3 通道的温度值 (单位：0.1℃)。例如，若 DM300 的值为 1234，则表示 A0 通道的温度测量值为 123.4℃。若温度值为 -1000，则说明该通道的热电偶断线。

## 4.8 4 通道 PT100 热电阻输入模块 SC-U-4PT100

### 4.8.1 主要特点

- 每个模块可测量 4 通道 PT100 传感器。
- 4 通道采样周期可选为：0.1 秒、0.2 秒（须在定货时指明）。采样周期越长，则分辨率越高，抗干扰越好。
- 采用三线制接线方式，可消除导线电阻的影响。
- 内置有微处理器，直接输出线性化后的温度值。
- 测量范围为 $-100^{\circ}\text{C} \sim 750^{\circ}\text{C}$ ，分辨率为 $0.1^{\circ}\text{C}$ 。
- 每个模块还可以内置另外 4 路模拟量（选件），有 4AI-V（4 路电压输入）、4AI-I（4 路电流输入）、2AI2AO-V（2 路电压输入 2 路电压输出）、2AI2AO-I（2 路电流输入 2 路电压输出）、4EI（4 路热电偶输入）、2EI2AO（2 路热电偶输入 2 路电压输出）、2PT100（2 路 PT100 热电阻输入）等选择。
- 模块尺寸（W×H×D）为 $71\text{mm} \times 80\text{mm} \times 62\text{mm}$ 。

### 4.8.2 端子排列图

端子排列如下：

COM			R0+	R0-	R0G	R1+	R1-	R1G	MC1	MP2	MP3
SPI 地址开关			R2+	R2-	R2G	R3+	R3-	R3G	MC0	MP0	MP1

**R0:** PT100 输入通道 0。R0+为 PT100 电阻一端，R0-和 R0G 为 PT100 电阻另一端。

**R1:** PT100 输入通道 1。R1+为 PT100 电阻一端，R1-和 R1G 为 PT100 电阻另一端。

**R2:** PT100 输入通道 2。R2+为 PT100 电阻一端，R2-和 R2G 为 PT100 电阻另一端。

**R3:** PT100 输入通道 3。R3+为 PT100 电阻一端，R3-和 R3G 为 PT100 电阻另一端。

**COM:** 模拟电路的公共地。

**MC0、MC1、MP0、MP1、MP2、MP3:** 内置的另外 4 路模拟量等特殊功能接线端，具体意义要看内置模拟量板的说明。

### 4.8.3 安装

#### (1) DIN 导轨安装方式

按 DIN46277 标准（宽 35mm）的 DIN 导轨中原封不动装入。

取出主体时，要从上方拉出 DIN 导轨钩子。

**注意：在有较强振动场合不要使用 DIN 导轨，而应直接安装**

#### (2) 直接安装方式

可使用单元上的 2 个安装孔（M4 螺丝孔）来进行直接安装。

#### (3) 扩展模块的安装位置

开关量扩展模块应安装在 CPU 模块的右边，模拟量扩展模块应安装在开关量扩展模块的右边。

#### (4) 模块地址

在模块上的 4 位 SPI 地址拨码开关即为本模块的地址开关。可设置范围为 2~12、14。

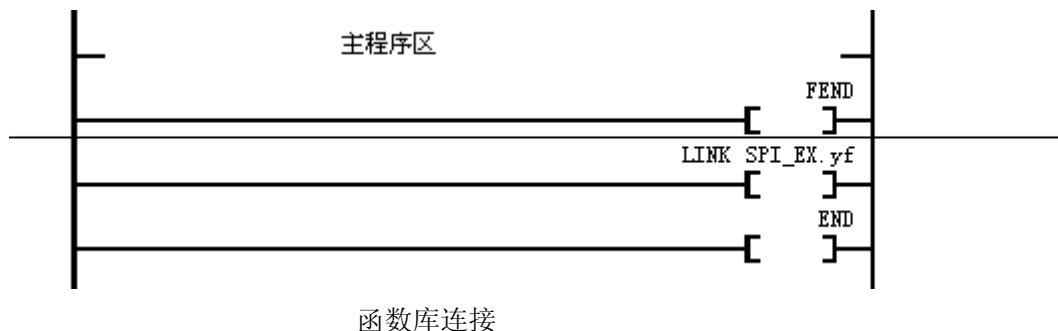
### 4.8.4 读入温度值

由于该模块采用 SPI 接口，因此用户需在自己的程序中连接 SPI 扩展模块函数库 SPI\_EX.yf。在该函数库中提供有基于 SPI 接口的扩展模块所需要的函数，如数字量输入、数字量输出、模拟量输入、模拟量输出等。在编程软件 EasyLad 中的操作如下：

- 点击“工具”菜单，弹出以下菜单内容：



- 点击“添加函数库连接 (L)”，弹出“打开文件”对话框，找到文件 SPI\_EX.yf 打开即可，连接后的程序如下：



函数库连接指令 (LINK) 必须位于主程序结束指令 (FEND) 的下面。

用户也可在函数库连接指令中省去路径名，如“LINK SPI\_EX.yf”，此时只要函数库文件 SPI\_EX.yf 与用户的梯形图程序文件在同一个文件夹中即可。

在该函数库中用于模拟量输入的函数如下：

- **模拟量输入函数：SPI\_AI**

*函数定义：*

FUN I, SPI\_Addr As D0, AI\_Addr As D1

*函数功能：*

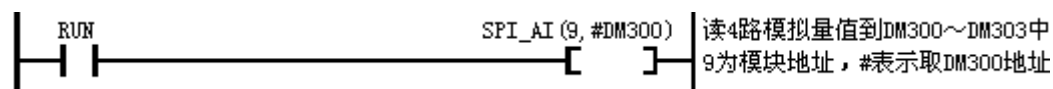
读入模拟量输入模块中的 4 路模拟量值存储到指定的数据块中。

*输入参数：*

**SPI\_Addr:** 模块地址。

**AI\_Addr:** 数据块的首地址 (DM 存储器)，占用 4 个字单元。

该函数的调用例子如下 (SC-U0A-4RTD 的模块地址为 9)：



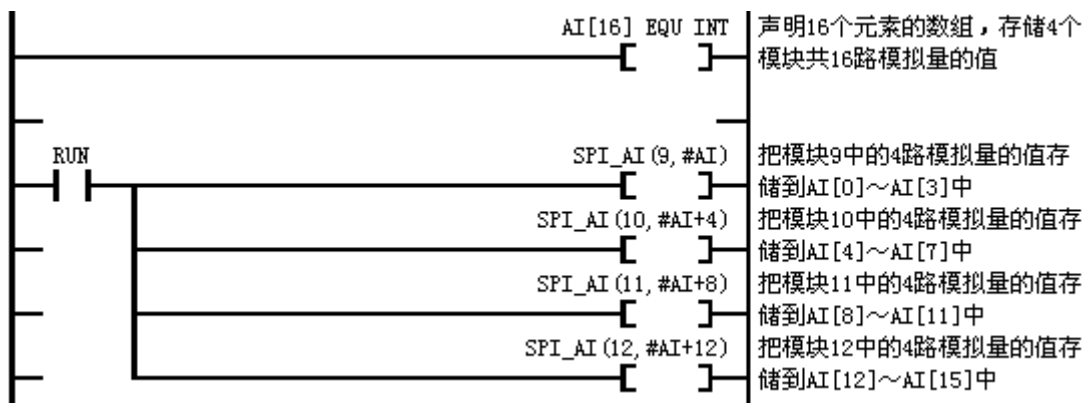
模拟量输入函数 SPI\_AI 使用例子

**函数 SPI\_AI 应在主程序中使用 RUN 触点调用，以保证每个扫描周期都被执行。**

在上面的程序例子中，DM300 中为 R0 通道的温度值 (扩大 10 倍)，DM301 中为 R1 通道的温度值 (扩大 10 倍)，DM302 中为 R2 通道的温度值 (扩大 10 倍)，DM303 中为 R3 通道的温度值 (扩大 10 倍)。例如，若 DM300 的值为 1234，则表示 R0 通道的温度测量值为 123.4℃。

若系统中使用了多个本模块，则程序例子如下：

假设使用了 4 个模拟量输入模块，其模块地址依次为 9、10、11、12。梯形图例子如下（使用数组定义方式）：



多个模拟量输入模块的程序例子

在上面的程序例子中，模块地址 9 中的 4 路模拟量 R0~R3 的值分别存储到 AI[0]~AI[3]中，模块地址 10 中的 4 路模拟量 R0~R3 的值分别存储到 AI[4]~AI[7]中，模块地址 11 中的 4 路模拟量 R0~R3 的值分别存储到 AI[8]~AI[11]中，模块地址 12 中的 4 路模拟量 R0~R3 的值分别存储到 AI[12]~AI[15]中。

## 4.9 5 个通讯口扩展模块 SC-U-5COM

### 4.9.1 主要特点

- 5 个 RS485（或 RS232）通讯口扩展模块，有端子和 DB9 两种连接方式（注：DB9 连接只能提供 4 个通讯口）。
- 波特率：2400、4800、9600、19200、38400、57600、115200。数据位：7 位或 8 位。停止位：1 位。校验位：无校验、偶校验、奇校验。
- 模块内置 3 种数据块校验计算：累加和（SUM）、异或和（XOR）、CRC-16（与 ModBus-RTU 校验方式兼容）。
- 提供有 ModBus-RTU 协议主站和从站通讯函数库。
- 模块尺寸（W×H×D）为 71mm×80mm×62mm。

### 4.9.2 端子排列图

端子排列如下：

G	D1-	D1+		G	D2-	D2+		G	D3-	D3+	
G	D4-	D4+		G	D5-	D5+					

**G**：通讯地。

**D1-**：第 1 个通讯口数据负端（RS485）或 RX（RS232）。

**D1+**：第 1 个通讯口数据正端（RS485）或 TX（RS232）。

**D2-**：第 2 个通讯口数据负端（RS485）或 RX（RS232）。

**D2+**：第 2 个通讯口数据正端（RS485）或 TX（RS232）。

**D3-**：第 3 个通讯口数据负端（RS485）或 RX（RS232）。

**D3+**：第 3 个通讯口数据正端（RS485）或 TX（RS232）。

**D4-**：第 4 个通讯口数据负端（RS485）或 RX（RS232）。

**D4+**：第 4 个通讯口数据正端（RS485）或 TX（RS232）。

**D5-**：第 5 个通讯口数据负端（RS485）或 RX（RS232）。

**D5+**：第 5 个通讯口数据正端（RS485）或 TX（RS232）。

### 4.9.3 模块安装和通讯口地址

#### (1) DIN 导轨安装方式

按 DIN46277 标准（宽 35mm）的 DIN 导轨中原封不动装入。

**注意：在有较强振动场合不要使用 DIN 导轨，而应直接安装**

#### (2) 直接安装方式

可使用单元上的 2 个安装孔（M4 螺丝孔）来进行直接安装。

#### (3) 扩展模块的安装位置

该扩展模块应安装在 CPU 模块的右边。

#### (4) 模块地址和通讯口地址

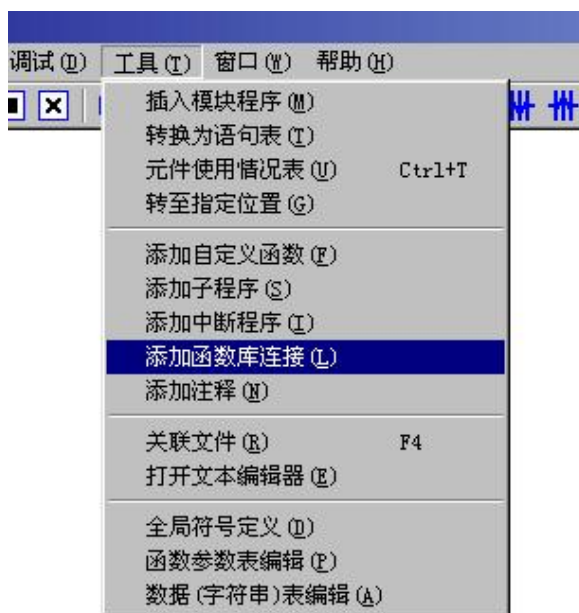
该模块的 SPI 地址固定占用 2、3、4、5、6，因此用户的其他扩展模块的 SPI 地址不要再设置为这些数。通讯口 1 的 SPI 模块地址为 2，通讯口 2 的 SPI 模块地址为 3，通讯口 3 的 SPI 模块地址为 4，通讯口 4 的 SPI 模块地址为 5，通讯口 5 的 SPI 模块地址为 6。

### 4.9.4 如何进行通讯

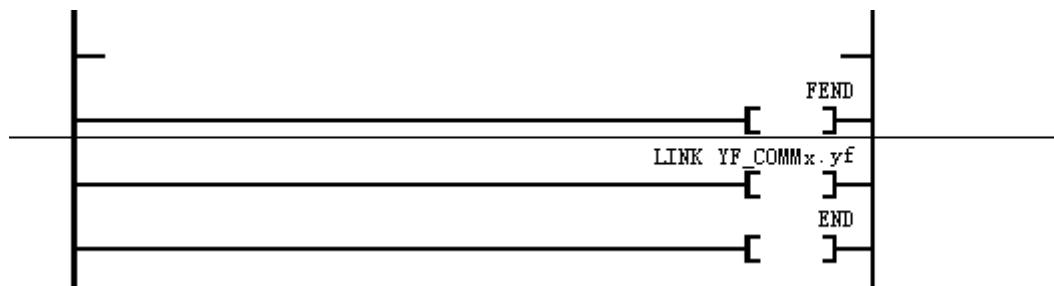
每个通讯口内置有 64 字节发送缓冲区（地址编号为 0~63）和 64 字节接收缓冲区（地址编号为 0~63），二者完全独立，每帧数据最大可发送和接收 64 字节，CPU 模块通过向发送缓冲区写数据来设置要发送的内容，通过读接收缓冲区来读取接收到的数据。

用户需在自己的程序中连接多模块扩展通讯口函数库 YF\_COMMx.yf。在该函数库中提供了使用该模块所需要的函数。在编程软件 EasyLad 中的操作如下：

- 点击“工具”菜单，弹出以下菜单内容：



- 点击“添加函数库连接(L)”，弹出“打开文件”对话框，找到文件 YF\_COMMx.yf 打开即可，连接后的程序如下：



函数库连接

函数库连接指令 (LINK) 必须位于主程序结束指令 (FEND) 的下面。

函数库文件 YF\_COMMx.yf 与用户的梯形图程序文件应在同一个文件夹中。

在该函数库中提供的函数如下：

- **通讯设置函数：COMMxSet**

*函数定义：*

FUN I, SPI\_Addr As D0, Mode As D2, RxTim As D3

*函数功能：*

设置工作模式、波特率和接收字符间隔超时时间。

*输入参数：*

**SPI\_Addr:** 模块地址 (0~15)。

**Mode:** 工作模式和波特率选择，**位 2--0:** 波特率选择，000—2400，001—2400，010—4800，011—9600，100—19200，101—38400，110—57600，111—115200；**位 4--3:** 校验位选择，00—无校验，01—偶校验，10—奇校验；**位 5:** 数据位位数，0 为 8 位数据，1 为 7 位数据。停止位固定为 1 位。若没有执行过该函数，则模块默认的波特率为 19200，无校验。

**RxTim:** 接收字符间隔超时时间 (0~255ms)，当设置为 0 时表示没有接收字符间隔超时检测。当接收到数据时将启动接收字符间隔超时检测，若在该时间内没有接收到下个数据，则置位接收字符间隔超时标记。

该函数调用例子：COMMxSet (2, H03, 10)

- **读通讯模块的状态字：COMMxFlag**

*函数定义：*

FUN I, SPI\_Addr As D0

*函数功能：*



读取通讯模块内的标记位：接收缓冲区校验比较结果标记、发送数据标记、接收字符间隔超时标记等。

*输入参数：*

**SPI\_Addr**：模块地址（0~15）。

*返回值：*

通讯模块内的各个标记位（状态字）：

位 0：发送数据标记，为 1 表示正在发送数据块，为 0 表示发送端口空闲，数据块发送完成或无数据发送。

位 1：接收缓冲区数据块校验比较结果标记，当执行 Rx1Sum、Rx1Xor 或 Rx1CRC 函数后，为 1 表示接收缓冲区数据块的校验结果与收到的校验数据不相等，为 0 表示相等。该标记在执行 Rx1Sum、Rx1Xor 或 Rx1CRC 函数后才有效。

位 2：接收字符间隔超时标记。若接收字符间隔超时时间不为 0，则当接收到数据时将启动接收字符间隔超时检测，若在该时间内没有接收到下个数据，则该标记置位。当执行接收数据指针复位函数 Rx1RST(0)后该标记复位。

该函数调用例子：D0 = COMMxFlag ( 2 )

- **读校验计算结果寄存器的值：COMMxVerW**

*函数定义：*

FUN I, SPI\_Addr As D0

*函数功能：*

读取通讯模块内的校验计算结果寄存器的值。

*输入参数：*

**SPI\_Addr**：模块地址（0~15）。

*返回值：*

校验计算结果寄存器的值（字）：

*说明：*

当执行了数据块校验计算函数（TXxSum、TXxXor、TXxCRC、RXxSum、RXxXor、RXxCRC）后，会把计算结果（SUM 和 CRC-16 校验为 1 个字有效；XOR 校验为低字节有效、高字节为 0）放到校验计算结果寄存器中，用户使用该函数即可获得校验计算结果。

该函数调用例子：D0 = COMMxVerW ( 2 )

- **发送缓冲区写字节数据（字符）：TXxChar**

*函数定义:*

FUN I, SPI\_Addr As D0, TxBufAddr As D2, Val As D3

*函数功能:*

向发送缓冲区中指定的地址写一个字节的的数据。

*输入参数:*

**SPI\_Addr:** 模块地址 (0~15)。

**TxBufAddr:** 发送缓冲区中要写的地址 (0~63)。

**Val:** 要写入的数据, 其低字节被写入到发送缓冲区中。

该函数调用例子: TXxChar(2, 0, 123 )

● **发送缓冲区写字数据 (2 个字节): TXxWord**

*函数定义:*

FUN I, SPI\_Addr As D0, TxBufAddr As D2, Val As D3

*函数功能:*

向发送缓冲区中指定的地址写一个字 (2 个字节) 的数据。

*输入参数:*

**SPI\_Addr:** 模块地址 (0~15)。

**TxBufAddr:** 发送缓冲区中要写的首地址 (0~63)。

**Val:** 要写入的数据, 其高字节被写入到 TxBufAddr 的地址, 低字节被写入到 TxBufAddr+1 的地址。

该函数调用例子: TXxWord (2, 0, 1234 )

● **发送缓冲区写双字数据 (4 个字节): TXxDINT**

*函数定义:*

FUN I, SPI\_Addr As D0, TxBufAddr As D2, Val As LDM256

*函数功能:*

向发送缓冲区中指定的地址写一个双字 (4 个字节) 的数据。

*输入参数:*

**SPI\_Addr:** 模块地址 (0~15)。

**TxBufAddr:** 发送缓冲区中要写的首地址 (0~63)。

**Val:** 要写入的双字数据, 其高字的高字节被写入到 TxBufAddr 的地址, 高字的低字节被写入到 TxBufAddr+1 的地址, 低字的高字节被写入到 TxBufAddr+2 的地址, 低字的低字

节被写入到 TxBufAddr+4 的地址。

该函数调用例子：TXxDINT (2, 0, 12345678 )

● **发送缓冲区求累加和校验：TXxSum**

*函数定义：*

FUN I, SPI\_Addr As D0, TxBufAddr As D2, Len As D3, OutAddr As D4

*函数功能：*

把发送缓冲区中指定的数据块进行累加和校验计算，并把计算结果放入到发送缓冲区中指定的地址和校验计算结果寄存器中。

*输入参数：*

**SPI\_Addr:** 模块地址 (0~15)。

**TxBufAddr:** 发送缓冲区中要参与累加和计算的数据块的首地址 (0~63)。

**Len:** 发送缓冲区中要参与累加和计算的数据块的长度。

**OutAddr:** 校验计算结果 (低字节) 要放入到发送缓冲区中指定的地址 (0~63)。

该函数调用例子：TXxSum (2, 0, 10, 10 )

● **发送缓冲区求异或和校验：TXxXor**

*函数定义：*

FUN I, SPI\_Addr As D0, TxBufAddr As D2, Len As D3, OutAddr As D4

*函数功能：*

把发送缓冲区中指定的数据块进行异或和校验计算，并把计算结果放入到发送缓冲区中指定的地址和校验计算结果寄存器中。

*输入参数：*

**SPI\_Addr:** 模块地址 (0~15)。

**TxBufAddr:** 发送缓冲区中要参与异或和计算的数据块的首地址 (0~63)。

**Len:** 发送缓冲区中要参与异或和计算的数据块的长度。

**OutAddr:** 校验计算结果 (低字节) 要放入到发送缓冲区中指定的地址 (0~63)。

该函数调用例子：TXxXor (2, 0, 10, 10 )

● **发送缓冲区求 CRC-16 校验：TXxCRC**

*函数定义：*

FUN I, SPI\_Addr As D0, TxBufAddr As D2, Len As D3, OutAddr As D4

### 函数功能:

把发送缓冲区中指定的数据块进行 CRC-16 (ModBus-RTU 兼容) 校验计算, 并把计算结果放入到发送缓冲区中指定的地址和校验计算结果寄存器中。

### 输入参数:

**SPI\_Addr:** 模块地址 (0~15)。

**TxBufAddr:** 发送缓冲区中要参与 CRC-16 计算的数据块的首地址 (0~63)。

**Len:** 发送缓冲区中要参与 CRC-16 计算的数据块的长度。

**OutAddr:** 校验计算结果 (1 个字) 要放入到发送缓冲区中指定的地址 (0~63), 高字节放入到 OutAddr 的地址, 低字节放入到 OutAddr+1 的地址。

该函数调用例子: TXxCRC (2, 0, 10, 10)

### ● 发送数据: TXxOut

#### 函数定义:

FUN I, SPI\_Addr As D0, TxBufAddr As D2, Len As D3

#### 函数功能:

把发送缓冲区中指定的数据块的数据从通讯端口发送出去。

#### 输入参数:

**SPI\_Addr:** 模块地址 (0~15)。

**TxBufAddr:** 发送缓冲区中要发送的数据块的首地址 (0~63)。发送时从首地址开始依次按字节发送。

**Len:** 发送缓冲区中要发送的数据块的长度 (按字节)。

该函数调用例子: TXxOut (2, 0, 12)

### ● 读接收端口接收到的字节数: RXxNum

#### 函数定义:

FUN I, SPI\_Addr As D0

#### 函数功能:

读取接收端口接收到的字节数。

#### 输入参数:

**SPI\_Addr:** 模块地址 (0~15)。

#### 返回值:

接收端口接收到的字节数, 为 0 或小于 0 表示还没接收到数据, 为其他表示已经接收到

的数据的字节数。该数据可通过执行 Rx1RST 函数复位为 0。

*说明:*

接收端口接收到的数据按先后顺序依次存放在从地址 0 开始的接收缓冲区中。

该函数调用例子: D0 = RXxNum ( 2 )

- **接收端口接收数据指针复位函数: RXxRST**

*函数定义:*

FUN I, SPI\_Addr As D0

*函数功能:*

把接收数据存放指针和接收字符间隔超时标记复位,同时也把接收到的字节数复位为 0,表示接收缓冲区要从地址 0 开始重新接收数据。

*输入参数:*

**SPI\_Addr:** 模块地址 (0~15)。

该函数调用例子: RXxRST ( 2 )

- **按字节读接收缓冲区中的数据: RXxChar**

*函数定义:*

FUN I, SPI\_Addr As D0, RxBufAddr As D2

*函数功能:*

按字节读取接收缓冲区中指定位置的数据。

*输入参数:*

**SPI\_Addr:** 模块地址 (0~15)。

**RxBufAddr:** 接收缓冲区中要读的数据的地址 (0~63)。

*返回值:*

接收缓冲区中指定位置的数据 (1 个字节)。

该函数调用例子: D0 = RXxChar (2, 0)

- **按字读接收缓冲区中的数据: RXxWord**

*函数定义:*

FUN I, SPI\_Addr As D0, RxBufAddr As D2

*函数功能:*

按字读取接收缓冲区中指定位置的数据。

输入参数:

**SPI\_Addr:** 模块地址 (0~15)。

**RxBufAddr:** 接收缓冲区中要读的字数据的首地址 (0~63)。

返回值:

接收缓冲区中指定位置的数据 (1 个字), 地址 RxBufAddr 中的数据为高字节, 地址 RxBufAddr+1 中的数据为低字节。

该函数调用例子: D0 = RXxWord (2, 10)

- **按双字读接收缓冲区中的数据: RXxDINT**

函数定义:

FUN I, SPI\_Addr As D0, RxBufAddr As D2

函数功能:

按双字读取接收缓冲区中指定位置的数据。

输入参数:

**SPI\_Addr:** 模块地址 (0~15)。

**RxBufAddr:** 接收缓冲区中要读的双字数据的首地址 (0~63)。

返回值:

接收缓冲区中指定位置的数据 (双字), 地址 RxBufAddr 中的数据为其高字的高字节, 地址 RxBufAddr+1 中的数据为高字的低字节, 地址 RxBufAddr+2 中的数据为低字的高字节, 地址 RxBufAddr+3 中的数据为低字的低字节。

该函数调用例子: LDM300 = RXxDINT (2, 10)

- **接收缓冲区求累加和校验: RXxSum**

函数定义:

FUN I, SPI\_Addr As D0, RxBufAddr As D2, Len As D3, CMPAddr As D4

函数功能:

把接收缓冲区中指定的数据块进行累加和校验计算, 并把计算结果放入到校验计算结果寄存器中, 同时和接收缓冲区中指定的位置的接收到校验数据进行比较。

输入参数:

**SPI\_Addr:** 模块地址 (0~15)。

**RxBufAddr:** 接收缓冲区中要参与累加和计算的数据块的首地址 (0~63)。

**Len:** 接送缓冲区中要参与累加和计算的数据块的长度。

**CMPSum:** 校验计算结果（低字节）要比较的校验数据的地址（0~63）。比较结果可由 Comm1Flag 函数读取。

该函数调用例子：RXxSum (2, 0, 10, 10)

● **接收缓冲区求异或和校验：RXxXor**

*函数定义：*

FUN I, SPI\_Addr As D0, RxBufAddr As D2, Len As D3, CMPAddr As D4

*函数功能：*

把接收缓冲区中指定的数据块进行异或和校验计算，并把计算结果放入到校验计算结果寄存器中，同时和接收缓冲区中指定的位置的接收到校验数据进行比较。

*输入参数：*

**SPI\_Addr:** 模块地址（0~15）。

**RxBufAddr:** 接收缓冲区中要参与异或和计算的数据块的首地址（0~63）。

**Len:** 接收缓冲区中要参与异或和计算的数据块的长度。

**CMPSum:** 校验计算结果（低字节）要比较的校验数据的地址（0~63）。比较结果可由 Comm1Flag 函数读取。

该函数调用例子：RXxXor (2, 0, 10, 10)

● **接收缓冲区求 CRC-16 校验：RXxCRC**

*函数定义：*

FUN I, SPI\_Addr As D0, RxBufAddr As D2, Len As D3, CMPAddr As D4

*函数功能：*

把接收缓冲区中指定的数据块进行 CRC-16（ModBus-RTU 兼容）校验计算，并把计算结果放入到校验计算结果寄存器中，同时和接收缓冲区中指定的位置的接收到校验数据进行比较。

*输入参数：*

**SPI\_Addr:** 模块地址（0~15）。

**RxBufAddr:** 接收缓冲区中要参与 CRC 计算的数据块的首地址（0~63）。

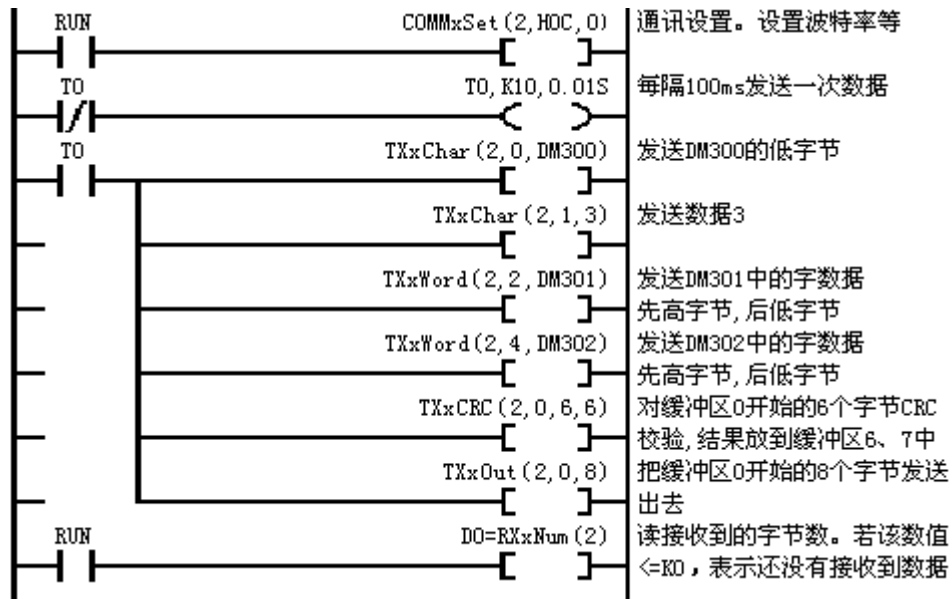
**Len:** 接收缓冲区中要参与 CRC 计算的数据块的长度。

**CMPSum:** 校验计算结果（字）要比较的校验数据（2 个字节）的首地址（0~63）。比较结果可由 Comm1Flag 函数读取。

该函数调用例子：RXxCRC (2, 0, 10, 10)

#### 4.9.5 扩展通讯口通讯程序例子

梯形图的通讯例子：



多扩展通讯口通讯例子

#### 4.9.6 扩展通讯口 ModBus-RTU 主机模式通讯

若用户需要使扩展通讯口作为 ModBus-RTU 主机模式进行通讯，则用户需在自己的程序中连接多模块扩展通讯口函数库 YF\_COMMx.yf 和 ModBus-RTU 主机模式通讯函数库 ModBus\_M.yf。

若要使用函数库 ModBus\_M.yf，则用户必须定义以下全局符号：

**COMMxReg[16]：**数组，类型为 DM 或 INT。函数库内部使用。不管用了几个扩展通讯口，都要占用 16 个字。

**COMMxBusy：**类型为 DM 或 INT。各个位为各个扩展通讯口的通讯忙标记，为 OFF 表示通讯空闲，为 ON 表示通讯忙。COMMxBusy.0 为模块地址 0 的扩展通讯口的通讯忙标记，COMMxBusy.1 为模块地址 1 的扩展通讯口的通讯忙标记，COMMxBusy.2 为模块地址 2 的扩展通讯口的通讯忙标记，……，COMMxBusy.15 为模块地址 15 的扩展通讯口的通讯忙标记。

用户可使用全局符号表来定义上述全局符号（点击鼠标右键，从弹出菜单中选择“全局符号表”即可打开全局符号表），如下：

符号名	类型或地址	描述
1 COMMxReg[16]	DM400	ModBus_M函数库所需全局符号：函数库内部使用
2 COMMxBusy	DM416	ModBus_M函数库所需全局符号：各个扩展通讯口的通讯忙标记



在函数库 ModBus\_M.yf 中提供有以下函数供用户使用。

## 1、读多寄存器函数 ModRHRx

*函数定义:*

FUN I, SPI\_Addr As D0, CommAddr As D3, DM\_Addr As D1, Len As D2

*函数功能:*

按指定的从机地址，读取从机中的某块 DM 的值（使用功能码“03”），并存入到本机中指定的 DM 数据块中。

*输入参数:*

**SPI\_Addr:** 模块地址（0~15）。

**CommAddr:** 从机地址。

**DM\_Addr:** 本机中 DM 数据块（数组）的地址（DM256 以后的单元），数组中的[0]为从机寄存器地址，[1]~…为本机中存储读入的从机数据块的数据。

**Len:** 本机中 DM 数据块（数组）的长度。

*返回值:*

无

*调用形式:*

ModRHRx(SPI\_Addr, CommAddr, DM\_Addr, Len)

*说明:*

若 COMMxBusy.n（n 为对应的模块地址 0~15）为 ON，则该函数不会执行。当 COM MxBusy.n 为 OFF 时，若调用该函数成功，则 COMMxBusy.n 变为 ON，表示本机作为主机正在与从机进行通讯，当通讯完成后 COMMxBusy.n 自动被复位；若调用不成功，则 COM MxBusy.n 依旧保持为 OFF，此时通常是由于其他主机正在对本机进行访问或错误的函数参数引起的。

*例:*

ModRHRx(2,1,#DM300,8) 其中 DM300 为从机寄存器地址,DM301~DM307 存储读入的从机数据块的数据（读入 7 个字）。

## 2、写多寄存器函数 ModPMRx

*函数定义:*

FUN I, SPI\_Addr As D0, CommAddr As D3, DM\_Addr As D1, Len As D2

#### 函数功能:

按指定的从机地址，把本机中的 DM 数据块的数值写入到从机中的某块寄存器中（使用功能码“16”）。

#### 输入参数:

**SPI\_Addr:** 模块地址（0~15）。

**CommAddr:** 从机地址。

**DM\_Addr:** 本机中 DM 数据块（数组）的地址（DM256 以后的单元），数组中的[0]为从机寄存器地址，[1]~…为主机中要写到从机中的数据。

**Len:** 本机中 DM 数据块（数组）的长度。

#### 返回值:

无。

#### 调用形式:

ModPMRx(SPI\_Addr, CommAddr, DM\_Addr, Len)

#### 说明:

若 COMMxBusy.n（n 为对应的模块地址 0~15）为 ON，则该函数不会执行。当 COM MxBusy.n 为 OFF 时，若调用该函数成功，则 COM MxBusy.n 变为 ON，表示本机作为主机正在与从机进行通讯，当通讯完成后 COM MxBusy.n 自动被复位；若调用不成功，则 COM MxBusy.n 依旧保持为 OFF，此时通常是由于其他主机正在对本机进行访问或错误的函数参数引起的。

#### 例:

ModPMRx (2,1,#DM300,8) 其中 DM300 为从机寄存器地址，DM301~DM307 为主机中要写到从机中的数据（共写入 7 个字）。

### 3、读线圈状态函数 ModRCSx

#### 函数定义:

FUN I, SPI\_Addr As D0, CommAddr As D3, DM\_Addr As D1, CoilNum As D2

#### 函数功能:

按指定的从机地址，读取从机中的某些线圈的状态（使用功能码“01”），并存入到本机中指定的 DM 存储器中。

*输入参数:*

**SPI\_Addr:** 模块地址 (0~15)。

**CommAddr:** 从机地址。

**DM\_Addr:** 本机中 DM 数据块 (数组, 占 2 个字) 的地址 (DM256 以后的单元), 数组中的[0]为从机线圈块首地址, [1]为本机中存储读入的线圈的状态。

**CoilNum:** 要读入的线圈的个数 (最多为 16 个)。

*返回值:*

无

*调用形式:*

ModRCSx(SPI\_Addr, CommAddr, DM\_Addr, CoilNum)

*说明:*

若 COMMxBusy.n (n 为对应的模块地址 0~15) 为 ON, 则该函数不会执行。当 COMMxBusy.n 为 OFF 时, 若调用该函数成功, 则 COMMxBusy.n 变为 ON, 表示本机作为主机正在与从机进行通讯, 当通讯完成后 COMMxBusy.n 自动被复位; 若调用不成功, 则 COMMxBusy.n 依旧保持为 OFF, 此时通常是由于其他主机正在对本机进行访问或错误的函数参数引起的。

*例:*

ModRCSx(2,1,#DM300,10) 其中 DM300 为从机线圈块首地址, DM301 存储读入的 10 个线圈的状态。

#### 4、强制多线圈函数 ModFMCx

*函数定义:*

FUN I, SPI\_Addr As D0, CommAddr As D3, DM\_Addr As D1, CoilNum As D2

*函数功能:*

按指定的从机地址, 强制从机中的某些线圈为指定的状态 (使用功能码 “15”)。

*输入参数:*

**SPI\_Addr:** 模块地址 (0~15)。

**CommAddr:** 从机地址。

**DM\_Addr:** 本机中 DM 数据块 (数组, 占 2 个字) 的地址 (DM256 以后的单元), 数组中的[0]为从机线圈块首地址, [1]为要把从机中的某些线圈强制为的状态, 位 0 对应于从

机线圈块的首个线圈。

**CoilNum:** 要强制的线圈的个数（最多为 16 个）。

*返回值:*

无

*调用形式:*

ModFMCx(SPI\_Addr, CommAddr, DM\_Addr, CoilNum)

*说明:*

若 COMMxBusy.n (n 为对应的模块地址 0~15) 为 ON, 则该函数不会执行。当 COMMxBusy.n 为 OFF 时, 若调用该函数成功, 则 COMMxBusy.n 变为 ON, 表示本机作为主机正在与从机进行通讯, 当通讯完成后 COMMxBusy.n 自动被复位; 若调用不成功, 则 COMMxBusy.n 依旧保持为 OFF, 此时通常是由于其他主机正在对本机进行访问或错误的函数参数引起的。

*例:*

ModFMCx(2,1,#DM300,10) 其中 DM300 为从机线圈块首地址, DM301 中的位 0--位 9 为从机中的 10 个线圈要强制的状态。

## 5、强制单线圈函数 ModFSCx

*函数定义:*

FUN I, SPI\_Addr As D0, CommAddr As D3, CoilAddr As D1, Val As D2.0

*函数功能:*

按指定的从机地址, 强制从机中的某个线圈为指定的状态 (使用功能码 “05”)。

*输入参数:*

**SPI\_Addr:** 模块地址 (0~15)。

**CommAddr:** 从机地址。

**CoilAddr:** 从机中要强制的线圈的地址。

**Val:** 要强制的状态, 0 为 OFF, 1 为 ON。

*返回值:*

无。

*调用形式:*

ModFSCx(SPI\_Addr, CommAddr, CoilAddr, Val)

*说明:*

若 COMMxBusy.n (n 为对应的模块地址 0~15) 为 ON, 则该函数不会执行。当 COM MxBusy.n 为 OFF 时, 若调用该函数成功, 则 COMMxBusy.n 变为 ON, 表示本机作为主机正在与从机进行通讯, 当通讯完成后 COMMxBusy.n 自动被复位; 若调用不成功, 则 COM MxBusy.n 依旧保持为 OFF, 此时通常是由于其他主机正在对本机进行访问或错误的函数参数引起的。

*例:*

ModFSCx(2,1,100,M10) 把从机中线圈 100 强制为本机中 M10 的状态。

## 6、调整单寄存器函数 ModPSRx

*函数定义:*

FUN I, SPI\_Addr As D0, CommAddr As D3, RegAddr As D1, Val As D2

*函数功能:*

按指定的从机地址, 把指定的数据写入到从机中某个寄存器中 (使用功能码 “06”)。

*输入参数:*

**SPI\_Addr:** 模块地址 (0~15)。

**CommAddr:** 从机地址。

**RegAddr:** 从机中要调整的寄存器的地址。

**Val:** 要写入的值。

*返回值:*

无。

*调用形式:*

ModPSRx(SPI\_Addr, CommAddr, RegAddr, Val)

*说明:*

若 COMMxBusy.n (n 为对应的模块地址 0~15) 为 ON, 则该函数不会执行。当 COM MxBusy.n 为 OFF 时, 若调用该函数成功, 则 COMMxBusy.n 变为 ON, 表示本机作为主机正在与从机进行通讯, 当通讯完成后 COMMxBusy.n 自动被复位; 若调用不成功, 则 COM MxBusy.n 依旧保持为 OFF, 此时通常是由于其他主机正在对本机进行访问或错误的函数参数引起的。

*例:*

ModPSRx(2,1,300,DM400) 把本机中 DM400 的值写入到从机中寄存器 300 中。

## 7、通讯事件扫描处理函数 OnCOMMx

*函数定义:*

FUN I, SPI\_Addr As D0

*函数功能:*

对扩展通讯口按 ModBus-RTU 主机模式进行通讯事件扫描处理。

*输入参数:*

**SPI\_Addr:** 模块地址 (0~15)。

*返回值:*

无

*调用形式:*

OnCOMMx(SPI\_Addr)

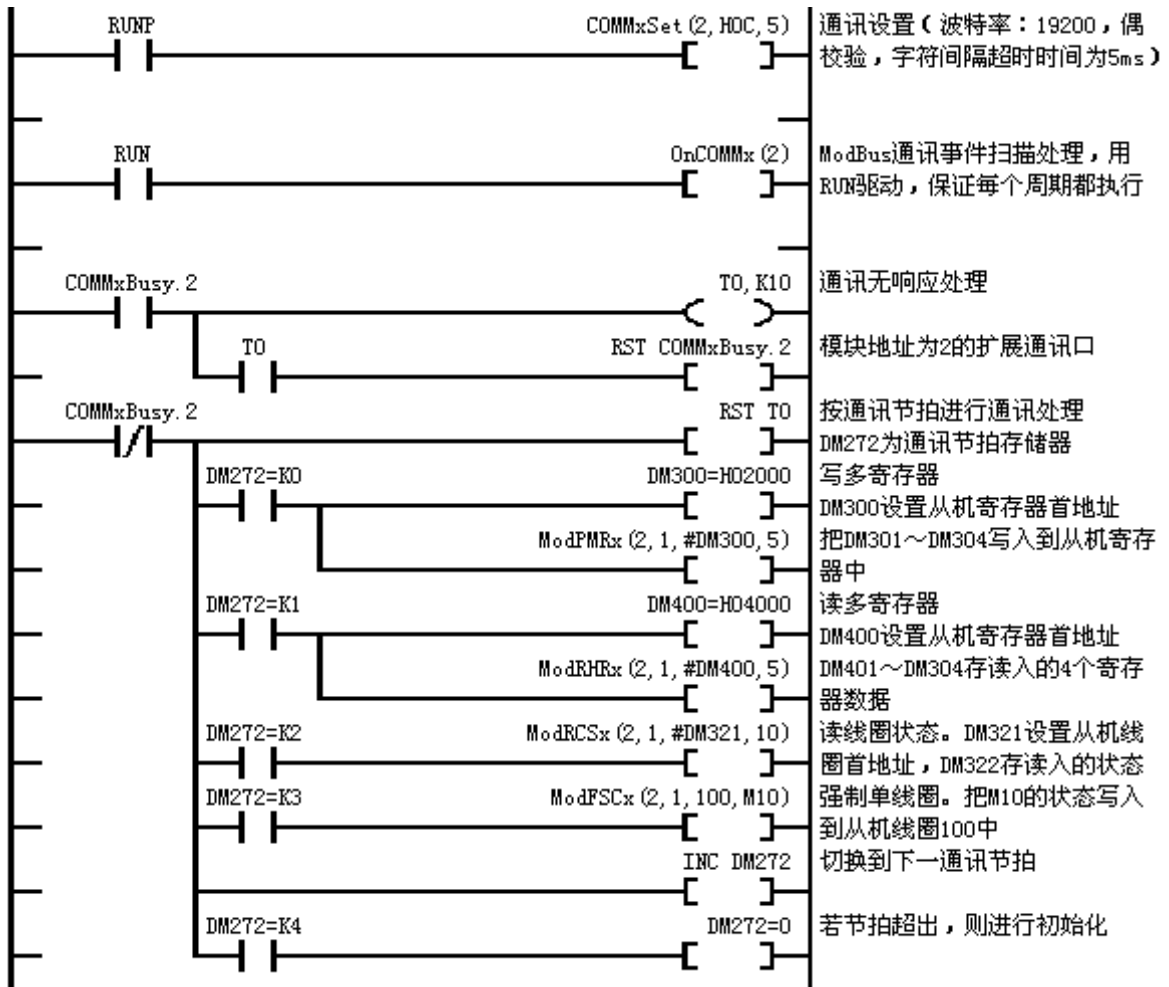
*说明:*

该函数要保证每个扫描周期至少被扫描 1 次。

*例:*

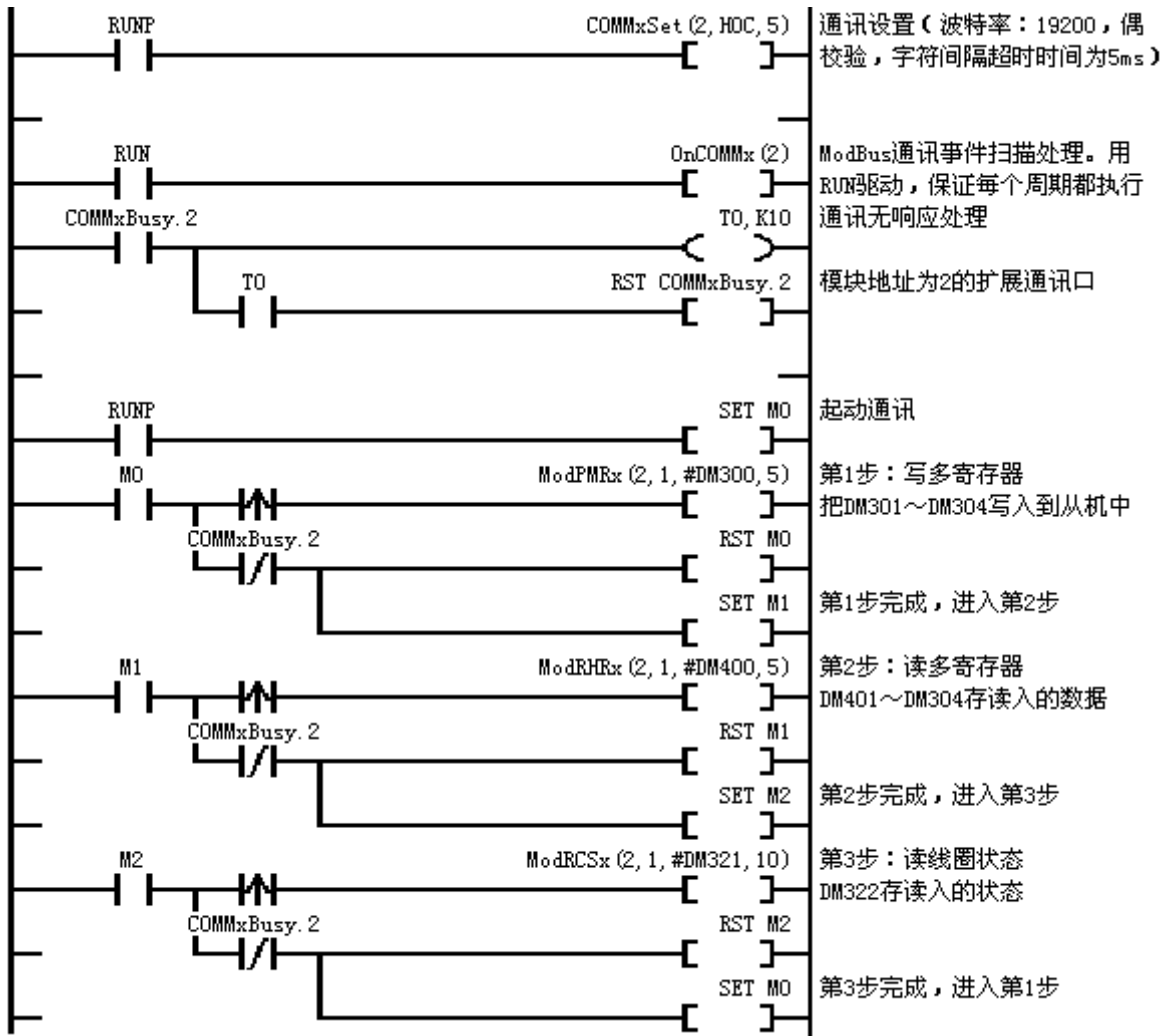
OnCOMMx (2) 对模块地址为 2 的扩展通讯口进行通讯事件处理。

扩展通讯口 ModBus-RTU 主机通讯的梯形图例子 1:



扩展通讯口 ModBus-RTU 主机通讯例子 1

扩展通讯口 ModBus-RTU 主机通讯的梯形图例子 2:



扩展通讯口 ModBus-RTU 主机通讯例子 2

#### 4.9.7 扩展通讯口 ModBus-RTU 从机模式通讯

若用户需要使扩展通讯口作为 ModBus-RTU 从机模式进行通讯, 则用户需在自己的程序中连接多模块扩展通讯口函数库 YF\_COMMx.yf 和 ModBus-RTU 从机模式通讯函数库 ModBus\_S.yf (该函数库支持 ModBus 功能码: 01、03、05、06、15。一个帧中的最大寄存器个数为 24 个字)。

在函数库 ModBus\_S.yf 中提供有以下函数供用户使用。

##### 1、ModBus-RTU 从机模式通讯处理函数 ModBus\_S

函数定义:

FUN I, SPI\_Addr As D8, SlaveAddr As D7

函数功能:



对扩展通讯口按 ModBus-RTU 从机模式进行通讯事件扫描处理。

*输入参数:*

**SPI\_Addr:** 模块地址 (0~15)。

**SlaveAddr:** 从机模式下的从机地址。

*返回值:*

无

*调用形式:*

ModBus\_S(SPI\_Addr, SlaveAddr)

*说明:*

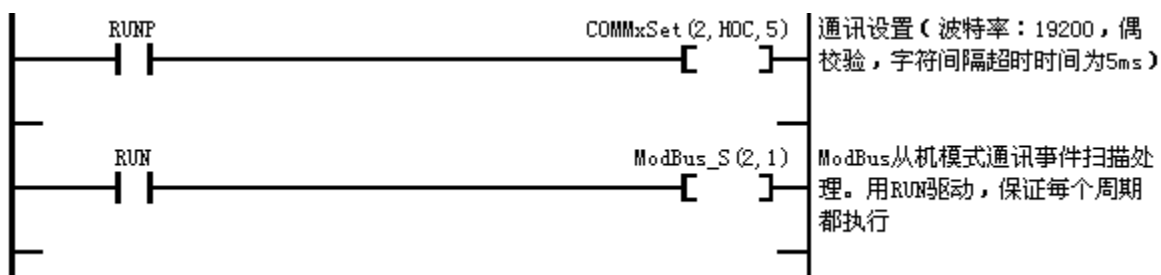
该函数要保证每个扫描周期至少被扫描 1 次。

该函数支持 ModBus 功能码: 01、03、05、06、15。一个帧中的最大寄存器个数为 24 个字。

*例:*

ModBus\_S(2,1) 其中 2 为扩展通讯口的模块地址, 1 为 ModBus 的从机地址。

扩展通讯口 ModBus-RTU 从机通讯的梯形图例子:



扩展通讯口 ModBus-RTU 从机通讯例子

## 4.10 内置的特殊功能扩展板

### 4.10.1 主要特点

- 该板内置于开关量扩展模块或模拟量扩展模块内部，为这些模块在单模块内再提供 4 路额外的模拟量或其他特殊功能模块。
- 内置有微处理器，直接输入输出模拟量值。
- 该系列扩展板有 4AI-V（4 路电压输入）、4AI-I（4 路电流输入）、2AI2AO-V（2 路电压输入 2 路电压输出）、2AI2AO-I（2 路电流输入 2 路电压输出）、4EI（4 路热电偶输入）、2EI2AO（2 路热电偶输入 2 路电压输出）、2PT100（2 路 PT100 热电阻输入）、COMM-232（RS232 扩展通讯口）、COMM-485（RS485 扩展通讯口）等选择。

### 4.10.2 型号规格表

特殊功能扩展板型号规格

型号	说明
SC-U-4AIB-V	4 路模拟量电压输入板
SC-U-4AIB-I	4 路模拟量电流输入板
SC-U-4AOB-V	4 路模拟量电压输出板
SC-U-2AI2AOB-V	2 路电压输入、2 路电压输出板
SC-U-2AI2AOB-I	2 路电流输入、2 路电压输出板
SC-U-4EIB	4 路热电偶输入板
SC-U-2EI2AOB	2 路热电偶输入、2 路电压输出板
SC-U-2PT100B	2 路 PT100 热电阻输入板
SC-U-COMM-232	RS232 扩展通讯板
SC-U-COMM-485	RS485 扩展通讯板

### 4.10.3 端子排列图

端子排列如下：

									MC1	MP2	MP3
									MC0	MP0	MP1

**MC0:** 模拟量板的模拟电路的公共地，扩展通讯板的通讯地。

**MC1:** 模拟电路的公共地。

**MP0:** 模拟量板的模拟量通道 0，RS485 通讯板的 D-，RS232 通讯板的 TX。

**MP1:** 模拟量板的模拟量通道 1，RS485 通讯板的 D+，RS232 通讯板的 RX。

**MP2:** 模拟量板的模拟量通道 2。

**MP3:** 模拟量板的模拟量通道 3。

#### 4.10.4 设置模拟板的 SPI 地址

在扩展模块的扩展连接插座旁边有一 4 位拨码开关即为内置模拟量扩展板的 SPI 地址开关，可设置范围为 2~12、14。

#### 4.10.5 如何输入输出模拟量值

输入输出模拟量的方法与模拟量扩展模块的完全一样，可参照对应的模拟量扩展模块。

## 4.11 远程 IO 扩展模块

### 4.11.1 主要特点

- 控制器主机中的 1 个扩展通讯口最多可连接 8 个远程 IO 扩展分机，每个远程 IO 扩展分机最大数字 IO 数量为 32 入/32 出，因此 1 个扩展通讯口最多可扩展 256 入/256 出个远程数字 IO。
- 单个远程 IO 扩展分机 32 入/32 出连续刷新时间在 10ms 以内，可使用多个扩展通讯口来连接多个远程 IO 扩展分机以提高远程 IO 刷新速度。
- 远程 IO 扩展分机有 16I（16 路开关量输入）、8I8R（8 路开关量输入/8 路继电器输出）、8I8T（8 路开关量输入/8 路晶体管输出）、16T（16 路晶体管输出）等选择，并使用控制器主机的 IO 扩展模块来扩展至 32 入/32 出。
- 通讯接口为 RS485，最大通讯距离为 500 米。
- 每个分机都内置有 120 欧终端电阻，并通过拨码开关选择使用和不使用。
- 模块尺寸（W×H×D）为 71mm×80mm×62mm。

### 4.11.2 端子排列图

端子排列如下（以 8I8R 为例）：

COM	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	0V	0V	+24
+24	X0	X1	X2	X3	X4	X5	X6	X7	G	D-	D+

**+24:** 为输入端 X0~X7（输入电路）提供的+24V 电源端，范围：12V~30V。

**X0~X7:** 开关量输入端。当该端与+24V 电源的 0V 端接通时为 ON，断开时为 OFF。该端与 0V 端之间可接无源触点或 NPN 型集电极开路输出方式的传感器。

**COM:** 输出端子 Y0~Y7 的公共端，该端最大电流为 8A。

**Y0~Y7:** 输出继电器的接线端。当输出继电器为 ON 时，则对应的端子与 COM 之间接通；当输出继电器为 OFF 时，则对应的端子与 COM 之间断开。

**D-:** RS485 编程通讯口数据-（B 端）。

**D+:** RS485 编程通讯口数据+（A 端）。

**G:** 通讯地

### 4.11.3 远程 I/O 点的编号规则

远程 IO 扩展模块的数字量 I/O 点均采用八进制编号，每 8 个为一组，编号中的最低位为该组中具体的 I/O 点号，编号中除了最低位的其他位为该组的组号，例如 X16：X 表示输入点，1 为该点所在的组号，6 为该点在该组中的点号。远程 IO 扩展分机单元提供的 I/O 点具有固定的编号，开关量扩展模块的 I/O 点采用相对的编号，即该模块的 I/O 编号由该模块在 I/O 扩展链中的位置决定，具体来说扩展模块 I/O 编号的顺序为最靠近扩展分机单元的编号为最小，以后依次递增。如下例：



远程 IO 扩展模块的 I/O 编号举例

若远程 IO 扩展分机单元为 8I8R，扩展模块 1 为 SC-U-16DIO，扩展模块 2 为 SC-U-16DIO，则扩展分机单元的 I/O 编号固定为 X0~X7、Y0~Y7，扩展模块 1 的 I/O 编号按顺序为 X10~X17、Y10~Y17，扩展模块 2 的 I/O 编号按顺序为 X20~X27、Y20~Y27。

### 4.11.4 安装

#### (1) DIN 导轨安装方式

按 DIN46277 标准（宽 35mm）的 DIN 导轨中原封不动装入。

取出主体时，要从上方拉出 DIN 导轨钩子。

**注意：在有较强振动场合不要使用 DIN 导轨，而应直接安装**

#### (2) 直接安装方式

可使用单元上的 2 个安装孔（M4 螺丝孔）来进行直接安装。

### 4.11.5 设置远程 IO 扩展分机的分机地址

在扩展分机的扩展连接插座旁边有一 4 位拨码开关即为分机地址设置开关（8、4、2、1 编码），当设置为 0 时分机地址为 1，当设置为 7 时分机地址为 8，最大分机地址为 8。

### 4.11.6 如何刷新远程 IO 模块

若要刷新远程 IO 模块，则用户需在自己的程序中连接多模块扩展通讯口函数库

YF\_COMMx.yf 和远程数字 IO 模块通讯函数库 RDIO.yf。

在函数库 RDIO.yf 中提供有以下函数供用户使用：

### ● 远程数字量 IO 扫描刷新函数 RDIO

*函数定义：*

FUN I, SPI\_Addr As D2, MaxCommAddr As D3, DM\_Addr As D0

*函数功能：*

使用扩展通讯口对远程数字量 IO 进行扫描刷新处理。

*输入参数：*

**SPI\_Addr:** 主机上用于刷新远程数字量 IO 的扩展通讯口所在的模块地址。

**MaxCommAddr:** 主机所连接的远程数字量 IO 分机的最大通讯地址（1~8）。最小通讯地址总是从 1 开始，并且通讯地址必须连续分配。

**DM\_Addr:** 主机中所存的远程数字量 IO 的状态。共占用 35 个字。**[0]~[15]: 远程输入继电器状态**，其中[0]为分机 1 的 X00~X17 状态，[1]为分机 1 的 X20~X37 状态，[2]为分机 2 的 X00~X17 状态，[3]为分机 2 的 X20~X37 状态，……，[14]为分机 8 的 X00~X17 状态，[15]为分机 8 的 X20~X37 状态。**[16]~[31]: 远程输出继电器状态**，其中[16]为分机 1 的 Y00~Y17 状态，[17]为分机 1 的 Y20~Y37 状态，[18]为分机 2 的 Y00~Y17 状态，[19]为分机 2 的 Y20~Y37 状态，……，[30]为分机 8 的 Y00~Y17 状态，[31]为分机 8 的 Y20~Y37 状态。**[32]~[34]: 内部使用。**

*返回值：*

无。

*调用形式：*

RDIO(SPI\_Addr, MaxCommAddr, DM\_Addr)

*说明：*

推荐在主程序中使用 RUN 触点调用该函数，以保证每个扫描周期都能被扫描到。

*例：*

RDIO(13,8,#DM400) 表示使用模块地址 13 的扩展通讯口刷新远程数字量 IO，该通讯口共连接 8 个远程数字量 IO 分机，通讯地址从 1 到 8，占用 DM400~DM434。这 8 个分机的输入输出状态在 DM400~DM431 中，其中 **DM400** 为分机 1 的 X00~X17 状态，**DM401** 为分机 1 的 X20~X37 状态，**DM402** 为分机 2 的 X00~X17 状态，**DM403** 为分机 2 的 X20~X37 状态，……，**DM414** 为分机 8 的 X00~X17 状态，**DM415** 为分机 8 的 X20~X37 状态，

DM416 为分机 1 的 Y00~Y17 状态，DM417 为分机 1 的 Y20~Y37 状态，DM418 为分机 2 的 Y00~Y17 状态，DM419 为分机 2 的 Y20~Y37 状态，……，DM430 为分机 8 的 Y00~Y17 状态，DM431 为分机 8 的 Y20~Y37 状态，DM432~DM434 被函数内部使用。

#### 4.11.7 刷新远程 IO 模块梯形图例子

刷新远程 IO 模块的梯形图例子如下：

