Koyo

Value & Technology

可编程序控制器 **SR21 系列**

用户手册

[第二版]

光洋电子(无锡)有限公司

前　言

本手册较为系统的介绍了 SR—21/22，SE—22 的系统规格、系统构成、编程、I/O 模块及其特性、安装和维护等方面的知识，为用户熟悉并应用该产品提供所需的信息。

SR—21/22，SE—22 是小型模块式可编程序控制器。这三种类型的 PLC 具有很好的兼容性，除 CPU 模块不同外，其电源框架，I/O 模块、编程器完全兼容。特殊模块，外部设备及编程语言大部分都兼容。故本手册以 SR—21PLC 为主进行讲述。

一、SR—21/22，SE—22PLC 之间的差异。

1、 SR—21 与 SR—22PLC 完全兼容，其区别是用户存储器容量由 1.7K 语（SR—21）增加到 3.7K 语（SR—22）。

2、 SE—22PLC 在 SR—21/22PLC 的基础上增加了：

1） CPU 模块具有 RS—232C 通讯接口（CCM 协议（子局），固定协议或无协议）。

2） 增加了级式（stage）编程语言，梯形图编程和级式编程可混合使用。

3） 增加了矩形扫描输入/矩阵扫描输出功能。

4） 采用 FLASHROM 用户存储器，用户程序容量为 2.5K 语。

二、SR—21/22，SE—22PLC 的特点：

1、 SR—21/22，SE—22PLC 是模块式 PLC 具有配置灵活，维护方便的特点。

2、 可外部设定定时器，计数器的预置值。

1） SR，SE 系列 PLC 具有拨盘接口模块 E—01D，可外接 4 组，每组 4 位 BCD 码的拨盘，可对指定的 674~677 4 个定时器/计数器进行预置值的外部设定。

2） 可以外接数据设定显示单元 C—02DS，用来设定或改变定时器/计数器的预置值和数据寄存器中的内容，并且可对 I/O 状态，T/C 经过值，数据寄存器内容进行监视。

3、 具有丰富的特殊模块。

SR，SE 系列 PLC 具有模拟量输入模块、模拟量输出模块、热电偶输入模块、高速计数器等特殊模块。

4、 具有步进电机单轴定位模块。

5、 具有定时中断扫描功能。

三、由于产品改进等原因，有时资料所刊内容会与实际产品有所差别，请注意！在使用时若有疑义，请与本公司驻各地办事处或与我们直接联系！

目　　录

第一章 系统构成

§1—1 系统构成

一、基本 PLC 框图

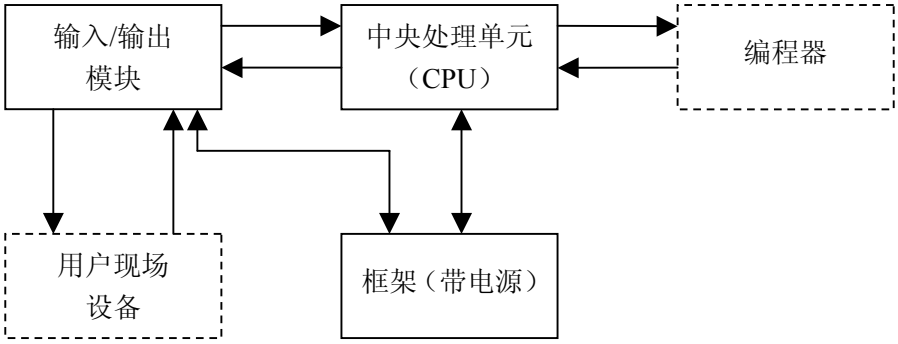


图 1.1 基本 PLC 框图

二、系统构成

SR—21/22、SE—22PLC 是组件结构，它由下列组件构成。

1、 **编程器**——它是用来输入用户程序、监视 CPU 内部状态的专用工具。

2、 **CPU 模块**——它是 PLC 的“大脑”，它读入输入模块的状态，根据已存入的用户逻辑进行判断，然后控制输出模块驱动控制对象。由编程器输入的用户程序存放在 CPU 中。

3、 **框架**——此框架本身带电源，用来安放 CPU 和输入/输出模块，并提供 PLC 机 5V、9V、24VDC 电源。

4、 **输入模块**——该模块是把外部设备的状态转换成 CPU 模块可接受的信号并送到 CPU 的存储器中。

5、 **输出模块**——该模块是根据 CPU 控制指令去驱动控制对象。

6、 **外围设备：**

（1）**数据通讯单元**（E—02DM，E—03DM）— 提供 SR—21/22PLC 与上位计算机或其它高档 PLC 之间进行通讯，传送 PLC 的 I/O 信息，SR 系列产品在通讯中只能作从机。

（2）**数据设定显示单元**（C—02DS）—该单元用来对 PLC 的 T/C 通断状态，T/C 经过值和数据寄存器的内容进行监视，可以用来改变或设定 T/C 的预置值和数据寄存器中的内容，并有自定义显示工作方式。可以选择 2 位、4 位、8 位 BCD 数据显示方式。该单元通过电缆与 PLC 编程口连接使用。

（3）**EPROM 写入单元**— 通过该单元可以把 CMOSRAM 中的用户程序固化在 EPROM 芯片中。

（4）**C—11D/12D 显示器**—它与 16 点晶体管输出模块（例如 E—15T）相连接，根据要求控制相应用户程序，可显示 PLC 内部数据。C—11D 为 4 位 BCD 码显示，C—12D 为 8 位 BCD 码显示。

（5）**C—10D 显示器**—先用编程器设定好所要监视的 T/C 经过值或数据寄存器内容后，关掉电源，在编程口用电缆接上该显示器，再次通电运行后，该显示器就显示由编程器所设定的监视的内容。

§1—2 性能指标

一、一般规格

项　　目	SR—21/22	SE—22
电源	AC115~230V±15%　47~63Hz	DC20~28V±10%
工作温度	0℃~60℃	
环境湿度	5~95%（无凝露）	
绝缘电阻	20MΩ以上　DC500V 电源—外部端子之间	10 MΩ　DC500V
耐电压	AC1500V 50—60Hz 1 分钟 电源—外部端子之间	AC1000V　1 分钟
耐振动	JIS C0911 ⅡB ，3 级标准	符合 GB2423.10—81FC 试验规定
耐冲击	JIS C0912 标准，三轴 10G	三个垂直轴的每一个轴偶然振幅 15g，11ms，半正弦波。
抗干扰	NEMA（ISC3—304）标准， 1μs，1KV 矩形波。	静态放电：　8KV
		辐射电磁场：　10V/M
		快速瞬态共模干扰：2KV
		衰减震荡波串模干扰：1KV
		抗三次谐波干扰,10%额定电压 0°和 180°
环境气氛	无腐蚀性气体	

二、性能规格

项　　目	SR—21/22	SE—22
程序运行方式	存储程序，循环扫描方式	←
程序语言	梯形图	梯形图、级式并用
指令数	71 条	83 条
处理速度	8ms/0.5K	2.4ms/0.5K
程序存储容量	0.7K/1.7K/3.7K(CMOS RAM) 0.7K/3.7K　(EPROM)	2.5K　(FLASHROM)
I/O 点数	168 点(184 点)	←
级	—	128 级
内部线圈	124 点（其中 28 点可停电记忆）	364 点(其中 268 点可停电记忆)
特殊功能线圈	12 点	28 点
移位寄存器	128 步	←
定时器	64 点（0.1~999.9s） (0.01~99.99s) (0.001~9.999s**)	64 点 (0.1~9999.9s) (0.01~99.99s)
计数器	64 点(1~9999)	←
鼓形控制器	64 个(10000 步)	←
数据寄存器	128 个(8 位)	336 个（8 位）　512 个（16 位）
高速计数器	10KHz	←
自诊断功能	CPU 异常、程序存储器奇偶校验、电池异常、语法检查	
监视功能	I/O 监视、ON/OFF 监视、T/C 经过值监视、数据寄存器监视	
保密口令	由编程器设定 4 位 BCD 码	←
通讯功能	RS—232/422　CCM 协议	RS—232　CCM 协议

*　常规 I/O 点为 168 点，如果不使用 E—01D 及 E—01Z 等模块时，可把内部线圈 160—177 的定义号用作 I/O 点，则 I/O 最大可达 184 点。

**　0.001 秒定时器实际分辨率为 3ms。

§1—3 框架

SR-21PLC 的框架有五槽、八槽和十槽三种。每个框架右边有一个电源，靠近电源的第一个槽中安放 CPU 模块，其余槽安放 I/O 模块。框架电源有两种，一种用 AC110/220 供电（E—01BJ；E—02B；E—05B；E—03B；E—04B），一种用 DC24V 供电（E—02B—C；E—03B—C；E—04B—C）。它们为插入框架的模块提供工作电源。

§1—3—1 基本框架

一、五槽框架（E—01BJ、E—02B、E—02B—C）

其中 E—01BJ 只能单独使用。该框架不带选择开关。接线端子也不一样。

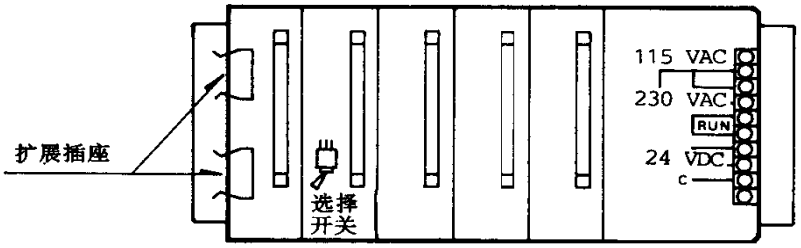


图 1.2 E-02B 五槽框架

二、八槽框架（E—05B）

该框架只能做 CPU 框架

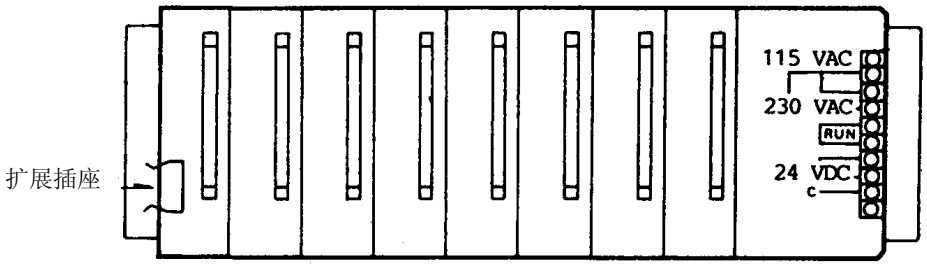


图 1.3 E-05B 八槽框架

三、十槽框架（E—03B、E—03B—C，E—04B，E—04B—C）

其中 E—03B，E—03B—C 只能做 CPU 框架，不能做扩展框架。

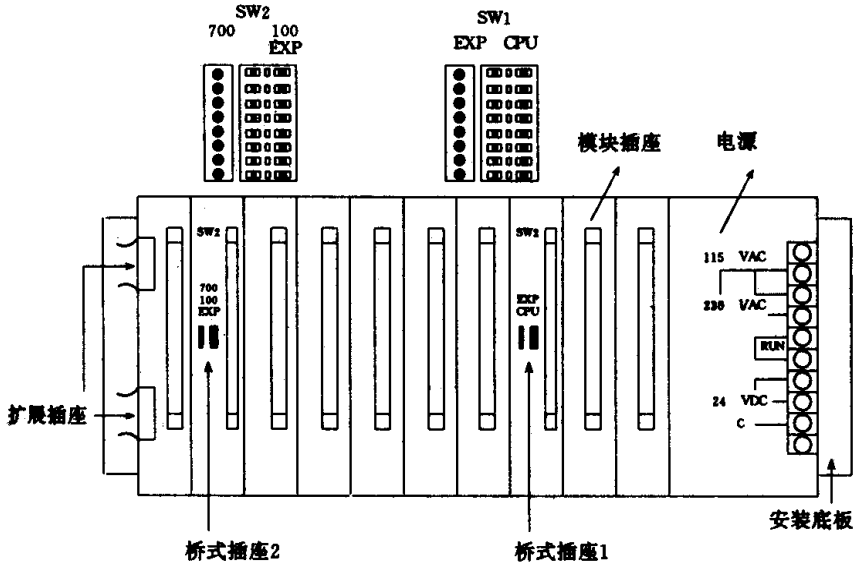
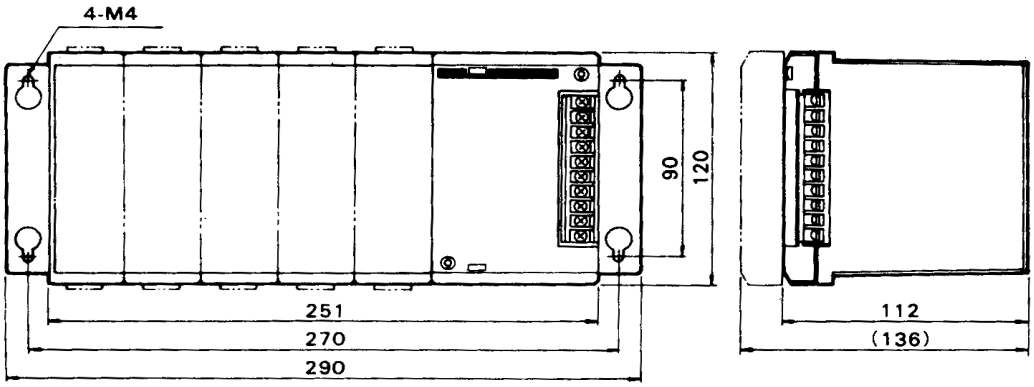


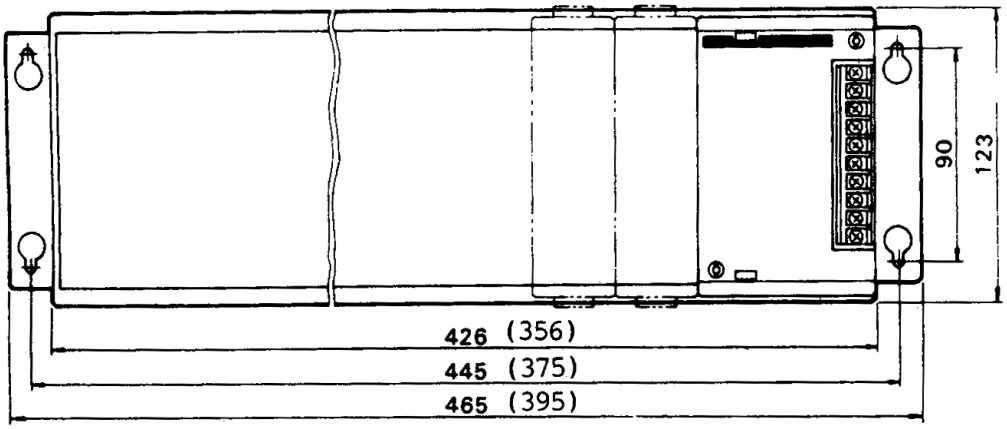
图 1.4 E-04B 十槽框架

四、框架安装尺寸（单位：毫米）

1、E—01BJ/E—02B/E—02B—C



2、E—03B/E—03B—C/E—04B/E—04B—C/E—05B



括号中数字为 E—05 框架安装尺寸

图 1.5 框架安装尺寸图

§1—3—2 扩展框架

如果 I/O 模块数超过一个电源框架所能容纳的数量，则可以另加一个到两个框架。有 CPU 模块的第一个框架称为基本框架或 CPU 框架，将另外两个（或一个）框架称为扩展框架。基本框架和扩展框架可以用同样的电源框架，框架之间用 420mm 的 I/O 扩展电缆连接，这根电缆的一端标有“CPU”，另一端标有“扩展”。“CPU”端应插进基本框架的底部扩展插座，“扩展”端插进扩展框架的顶部扩展插座。如果使用了第三个框架，则两个扩展框架之间也用这样的电缆连接，电缆的“CPU”端插进第二个框架（即第一个扩展框架）的底部扩展插座，“扩展”端插进第三个框架的顶部扩展插座。在扩展框架中，不需要再插入 CPU 模块，而全部插入 I/O 模块，I/O 模块的数量和类型是任意的。使用扩展框架应注意选择开关和桥式插座的位置。五槽框架（E—02B/E—02B—C），十槽框架（E—04B）可作为扩展框架。

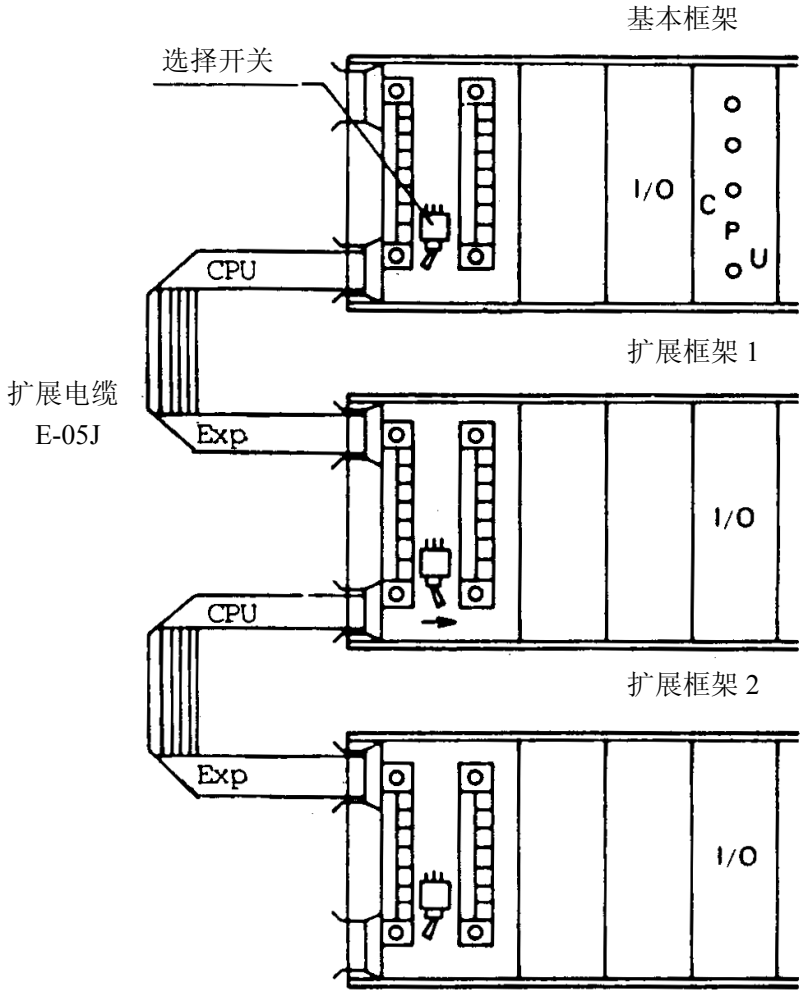


图 1.6 三个 E—02B（—C）五槽框架

一、五槽框架选择开关的位置

在五槽框架(E—02B/E—02B—C)内部 4 槽与 5 槽 之间有一框图架选择开关。在基本框架(CPU 框架)，这开关必须打向左侧，第二框架（第一扩展框架）此开关必须朝右侧，如果使用第三框架（第二扩展框架）开关必须朝左（见图 1.6 ）。

如果五槽框架作为八槽框架和十槽框架的扩展框架时，五槽框架中此开关必须打向左边。

二、十槽框架桥式插座的位置

十槽框架 E—04B/E—04B—C 在底板上有二个桥式插座，SW1 桥式插座位于 3 槽和 4 槽之间共有 EXP 和 CPU 两个位置。如果在 CPU 位置，则此十槽框架为 CPU 框架。如果在 EXP 位置，则此框架为扩展框架。

SW2 桥式插座位于 9 槽和 10 槽之间其有 $^{100}_{EXP}$ 和 700 两个位置,如果十槽框架作为 CPU 框架，则此桥式插座选择分配给 CPU 框架第 10 槽和扩展框架槽的定义号。如果在 $^{100}_{EXP}$ 位置，则 CPU 框架第十槽的定义号为 100，十槽扩展框架的第一槽到第三槽的定义号从 110 开始分配到 137 为止。如果在 700 位置，则第 10 槽的定义号为 700—707。如果有扩展框架，则扩展框架 I/O 的定义号从 710 开始分配。十槽框架作为扩展框架，则 SW2 开关一定在 $^{100}_{EXP}$ 位置。图 1.7 表示了十槽框架作为扩展框架时，SW1 和 SW2 的位置。

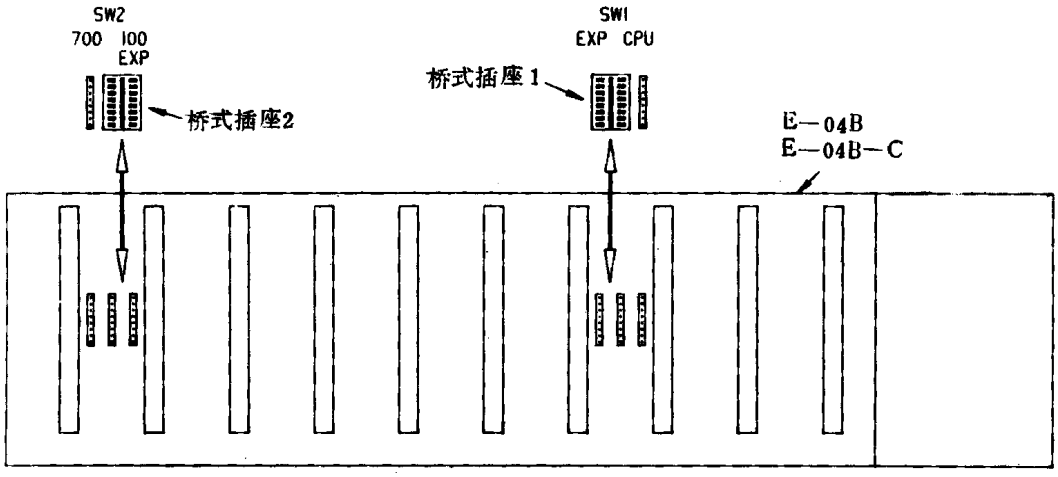


图 1.7　E-04B 十槽框架为扩展框架时 SW1，SW2 位置图

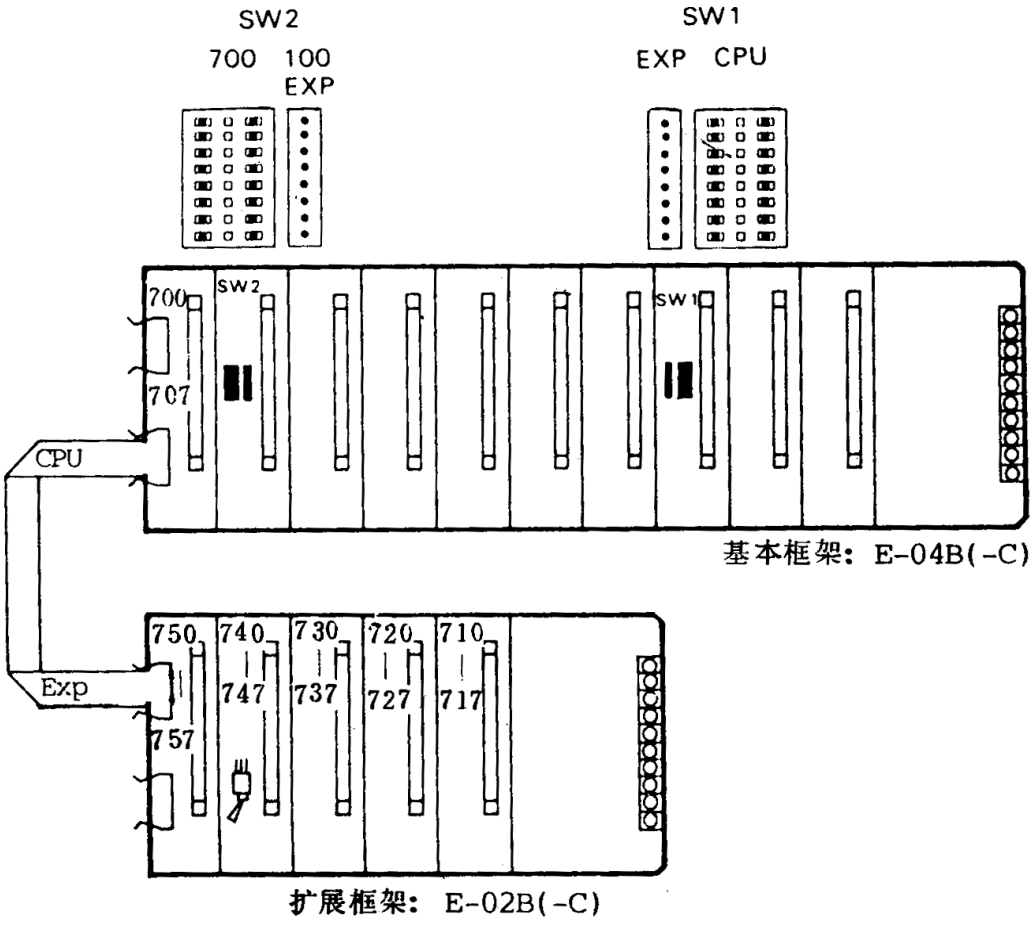


图 1.8　一个 E—04B（—C）扩展一个 E—02B（—C）框架

§1—3—3 框架电源

一、框架电源接线端子图

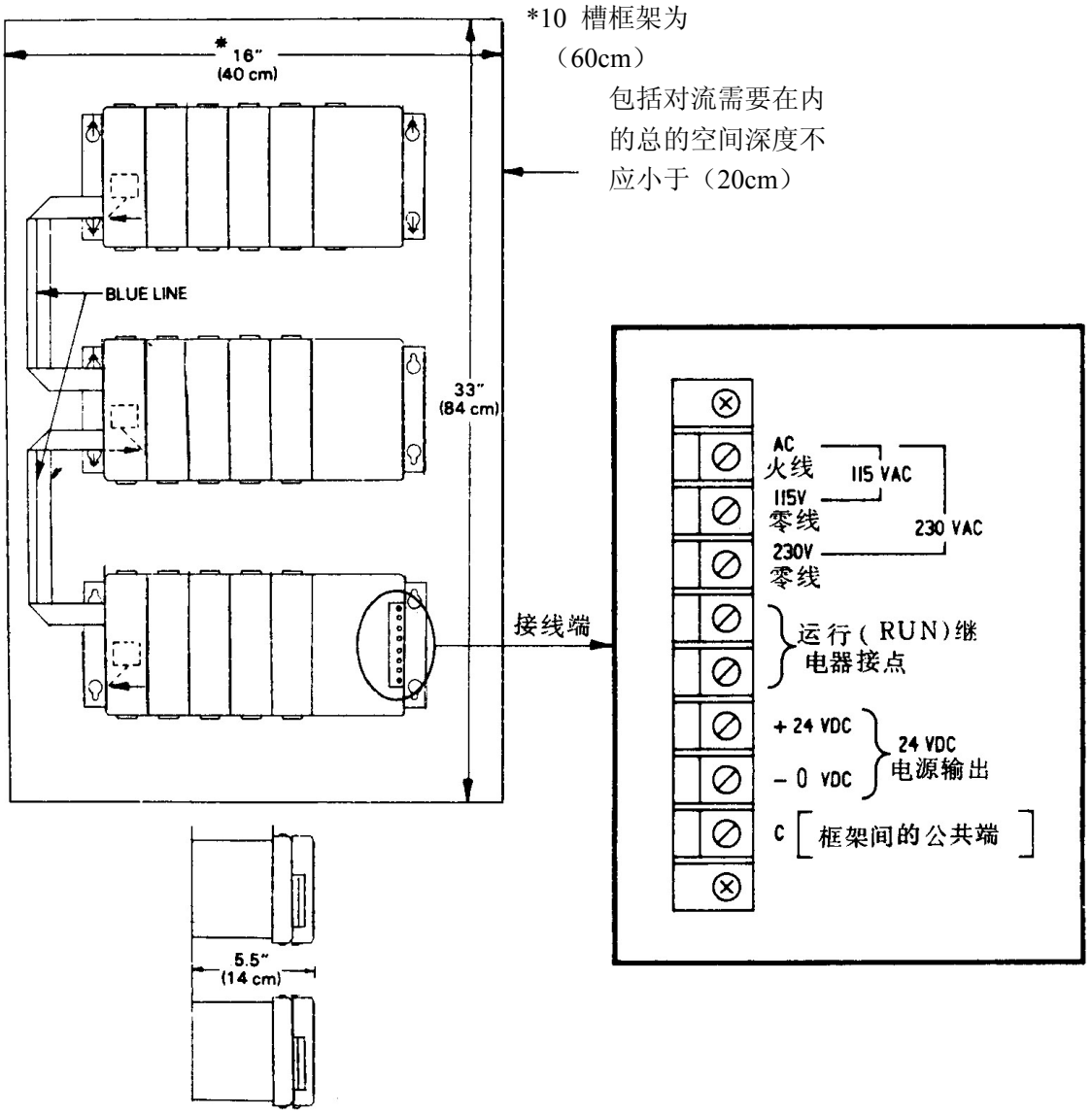


图 1.9 框架电源接线端子图

二、框架电源的电源连接图

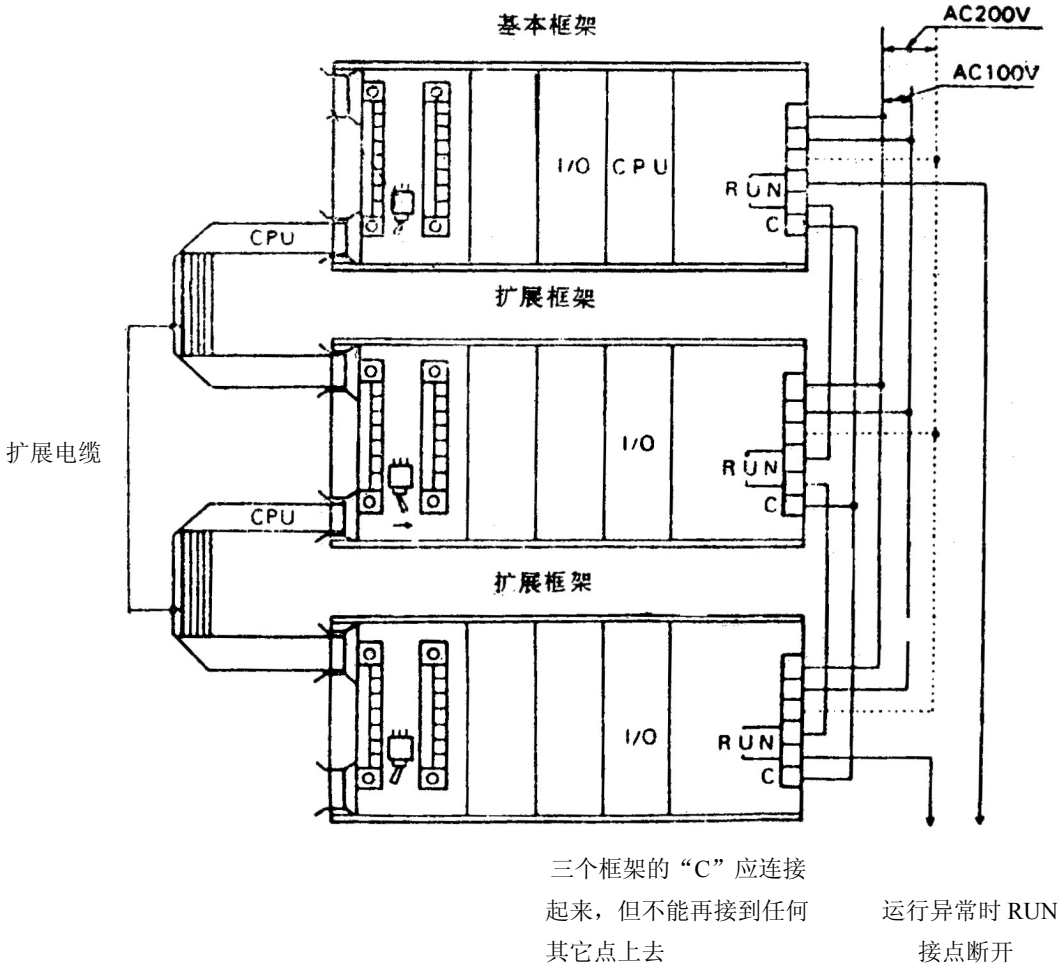


图 1.10 框架电源的电源连接

§1—3—4 框架安装的注意事项

一、模块在框架中的安装

CPU 模块总是插在第一个框架中紧靠电源的槽中。在 CPU 上有一个插座，当 CPU 安装在框架上时，编程器可插在它的上面，为了操作更灵活，在 CPU 和编程器之间可安装一根 1.5 米的编程电缆。框架的其余槽中可插入 I/O 模块，I/O 模块的数量、类型可根据实际应用的需要任意配置。所有的模块及编程器都有两个塑料弹性锁扣，用它将模块固定在电源框架中，编程器固定在 CPU 模块上。

二、散热

PLC 是通过空气对流来散热的，不需要用风扇。因此为保证散热，在框架的顶部和底部不应阻止自由空气的流通。在框架安装时，四边应留出的空间尺寸见图 1.11。在安装框架时，不要颠倒或竖起来安装。如不按图 1.11 的要求安装，那就只能考虑降低最高环境温度。另外，在框架附近最好不要装大容量的发热源，特别是框架的下方。为了 PLC 的可靠工作，进入框架底部的空气不应超过 60℃。

I/O 模块及电源的连线应整理好，以免阻塞空气的流通。

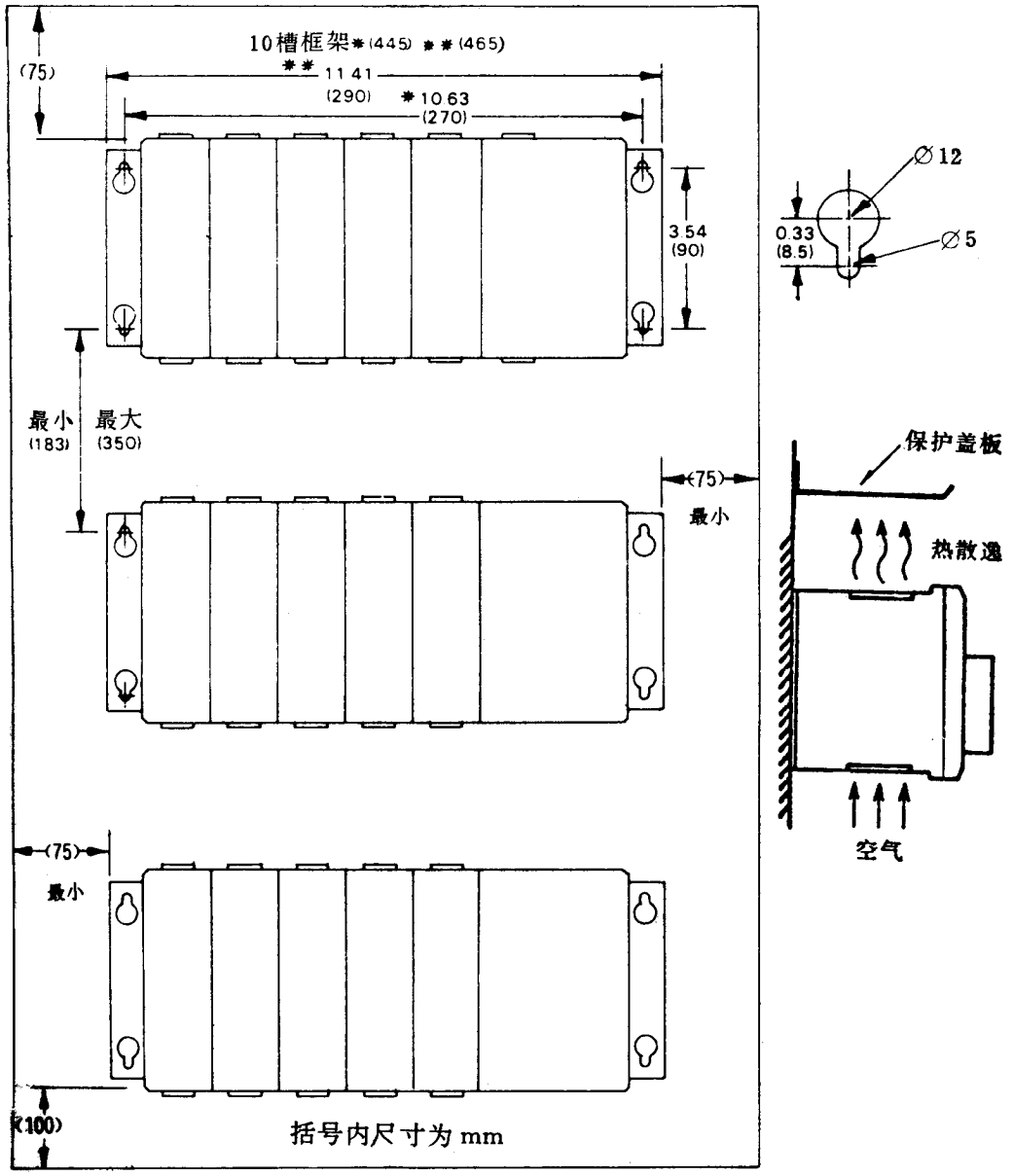


图 1.11　为了保证散热正确的框架安装尺寸

三、框架接地

电源地线接到两个安装螺栓中的任一个螺栓上。跨接线可用来将框架与安装架地线互连见图 1.12 ,电源上的“C”接线柱是在框架超过一个时，用来互连的。另外，在单机架系统时，“C”接线柱不接到任何地方。

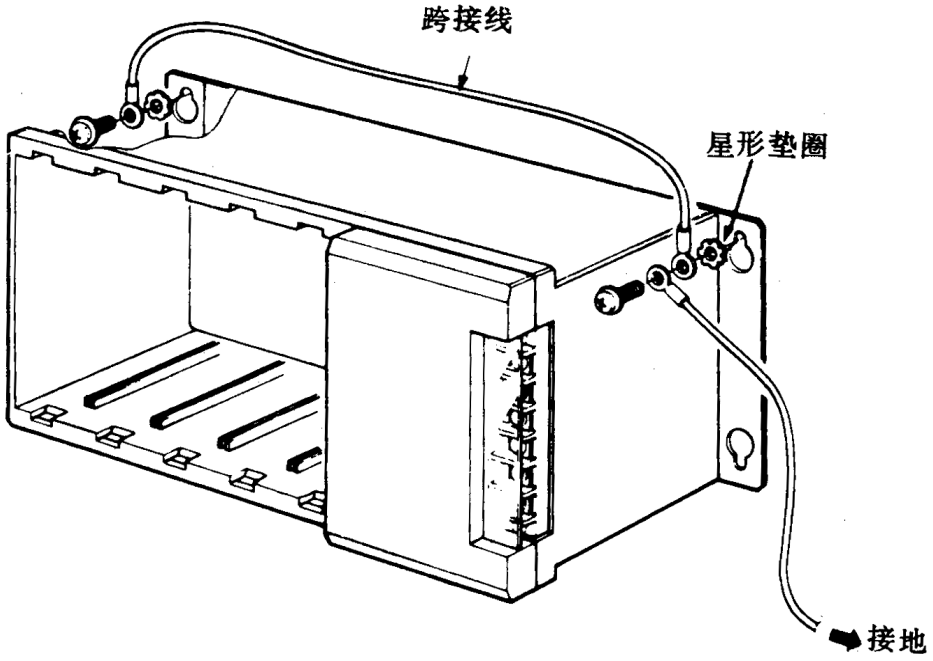


图 1.12 建议框架的接地方法

四、务必要正确连接框架电源的交流电源

框架电源的第一、第二个接线端子接交流 115V 电源，第一、第三线端子接交流 220V 电源，特别要注意不要把交流 220V 电源接到第一、二个接线端子，不然就要烧坏电源。

五、框架电源的负载限制

如果 CPU 框架或扩展框架中电源超载，那么不可预见的系统动作可能发生，为了保证这种情况不发生，放在框架中模块的总电流能力不得超过框架电源电流的支承能力。

每个模块所使用的电源是用单位负载来表达的，此处一单位负载等于 10 毫安。根据输入和输出都接通的最坏的情况来进行计算。表 1.1 列出了每个框架提供的单位负载，表 1.2 列出了每个模块所使用的单位负载。当配置一个框架时，记下所提供的单位负载，于是加出你已选用的模块所使用的总单位负载。模块的总单位负载必需不超过框架所提供的总单位负载。如果已超过，应该重新设计系统。按上面方法计算出的模块所使用的单位负载，是最坏的情况，实际使用中，除 E—01B 外，其它框架基本上都能满足要求。

表 1.1 框架电源所能提供的电源负载一览表

订 货 号	框 架 名 称	框架电源所提供的电源负载			
		+5V	+9V	+24V	备 注
E—01BJ	五槽框架（只能单独使用）	40	60	20	交流电源电压下降 10%以内
		40	30	20	交流电源电压下降 15%以内
E—02B	五槽框架	140	80	40	总的 2.2A
E—02B—C	五槽直流电源框架	140	80	50	总的 2.7A
E—03B	十槽框架（只能作 CPU 框架）	140	170	60	周围温度低于 50℃
		100	140	50	周围温度低于 55℃
E—03B—C	十槽直流框架（只能作 CPU 框架）	140	170	50	周围温度低于 50℃
		100	150	50	周围温度低于 55℃
E—04B	十槽框架	140	170	60	周围温度低于 50℃
		100	140	50	周围温度低于 55℃
E—04B—C	十槽直流电源框架	140	170	50	周围温度低于 50℃
		100	150	50	周围温度低于 55℃
E—05B	八槽框架（只能用 CPU 框架）	140	170	50	周围温度低于 50℃
		100	140	50	周围温度低于 55℃

六、新老框架对照表

由于产品改进、更新等原因，KOYO 推出了一些新的框架，以替代原有老框架，各新老框架的对应关系如表所示：

老 框 架	新 框 架
E—02B	D3—05B—1
E—02B—C	D3—05BDC
E—05B	D3—08B—1
E—04B	D3—10B—1
E—04B—C	D3—10BDC

1.2 组件所使用的单位负载

订 货 号	组 件 名 称	模块使用的单位负载		
		+5V	+9V	+24V
SR-21-EX	CPU	25	—	—
R-21P-EX	编程器	6	5	—
E-01N	24V 直流输入（8 点）	—	1	10
E-10NT	24V 直流输入/输出（4 点/4 点）	—	2	7
E-01NT	24V 直流输入/继电器输出（4 点/4 点）	—	20	6
E-01D	拨盘接口	—	1	9
E-05N	24V 直流输入（16 点）	—	3	23
E-01Z	高速计数器	—	7	—
E-02N	24V 交流/直流源输入（8 点）	—	1	—
E-55N	24V 交流/直流源输入（16 点）	—	13	—
E-01S	I/O 模拟器（8 点）输入	—	1	11
E-20N	115V 交流输入（8 点）	—	1	1
E-21N	115V 交流隔离输入（4 点）	—	1	—
E-22N	230V 交流输入（8 点）	—	1	—
E-25N	100V 交流输入（16 点）	—	1	—
E-10T	24V 直流汇点输出（8 点）	—	2	3
E-12T	24V 直流 2A 输出（4 点）	—	1	1
E-52T	24V 直流汇点/源输出（4 点）	—	1	10
E-50T	24V 直流源输出（8 点）	—	3	—
E-15T	24V 直流汇点输出带 LED（16 点）	—	4	10
E-55T	24V 直流源输出带 LED（16 点）	—	20	—
E-20T	115V/230V 交流输出（8 点）	—	16	—
E-21T	115V/230V 交流隔离输出（4 点）	—	16	—
E-01T	继电器输出（8 点）	—	34	—
E-05T	继电器输出（16 点）	—	48	—
E-02DM-R1	数据通讯单元	30	—	—
E-02RM	I/O 连接（本地）模块	60	—	—
E-02RS	I/O 连接（远程）模块	60	—	—
C-31P-R1	打印机接口单元	26	—	—
C-22P-1	PROM 写入器单元	50	—	—

单位负载=10mA，根据最坏情况（即所有输入和输出接通）计算。
应按§1—3—2 节说明正确安置选择开关和桥式插座的位置。

§1—3—5 现场连线的注意事项

当进行现场连线时，最好能做到以下几点：

1、 低电平信号连线应该与其它现场连线分开。

2、 交流电源连线应该与直流现场连线分开。

3、 连线不应敷设在靠近产生电气干扰设备的地方。

4、 如果存在严重的干扰问题，可能需要附加电源滤波或者一个隔离变压器。

5、 为了尽量减少对人员的危险。应该提供正确的接地。

6、 把所有 I/O 连线标上标记。

7、 应保证接到接线柱上的连线裸头不要超出螺钉下压板的范围、最好在连线头上加一个可收缩的套管罩住。

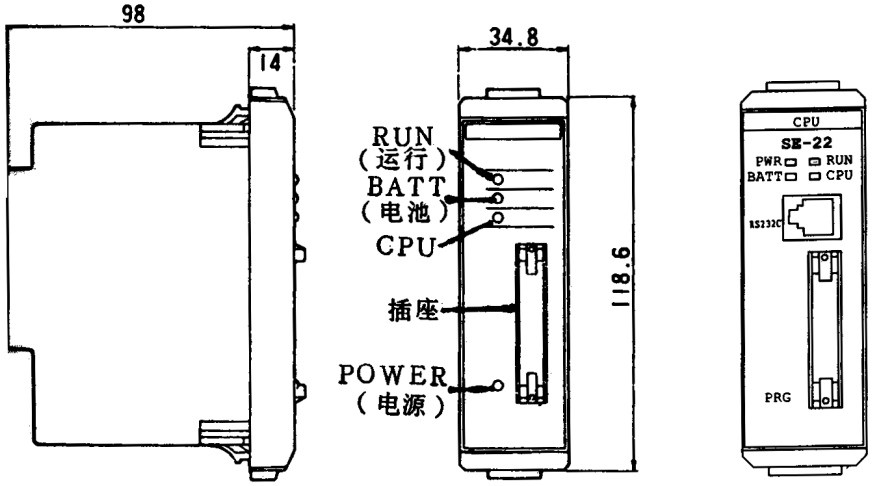
8、 连线应捆绑在一起，为了在不破坏 I/O 接线情况下能拔出 I/O 模块，连线要留一个足够的长度。在检查 I/O 连线后，重新盖上塑料盖。

§1—4 CPU 模块

CPU 是 PLC“大脑”，它一定要安装在基本框架的紧贴电源的第一个槽中。

§1—4—1 CPU 面板介绍

CPU 模块的面板图如图 1.13 所示：



a．SR—21/22 b．SE—22

图 1.13 CPU 面板及外形尺寸图

CPU 模块面板上的插座是安装编程器或编程电缆用的。

CPU 模块面板上还有 4 个指示灯指示 CPU 模块的工作状态。

1、 状态指示灯：

1） RUN（运行）灯——CPU 正常工作时此灯亮。

2） BATT（电池）灯——当锂电池电压低于 2.7V 时此灯亮。则在一周之内需要更换锂电池。

3） CPU 灯——当 CPU 发生故障时此灯亮。

4） POWER（电源）灯——当 5V 供电正常时，此灯亮。

2、 编程器插座：用来安装编程器或其它外围设备。

3、 通讯口（RS—232C）（仅 SE—22）——标准 RS—232C 通讯口，与上位计算机或上位 PLC 进行通讯用。

§1—4—2 SR-21/22 CPU 模块上开关的设定

SR-21 CPU 模块面板下方有两个开关（见图 1.14）SW1 和 SW2。ON 是开关拨向面板方向（即向上）。出厂时 SW1 开关设置在 OFF 位置。SW2 开关设置在 ON 位置。

SW1 开关是停电记忆功能选择开关，此开关 ON 时，停电记忆线圈 340—373 有停电记忆功能。如果 OFF 时，340—373 线圈无停电记忆功能。计数器和移位寄存器不受此开关影响，总是有停电记忆功能的。

SW2 开关与 2 组短接针配合使用来确定 CPU 中存储器的类型和大小。

SW2 开关 ON 时采用 CMOSRAM 存储器。SW2 开关 OFF 时采用 EPROM 存储器，具体设置见 CPU 模块的硬件环境和存储器配置表。

出厂时 SW1 在 OFF 位置，SW2 在 ON 位置。

§1—4—3 CPU 模块的存储器配置

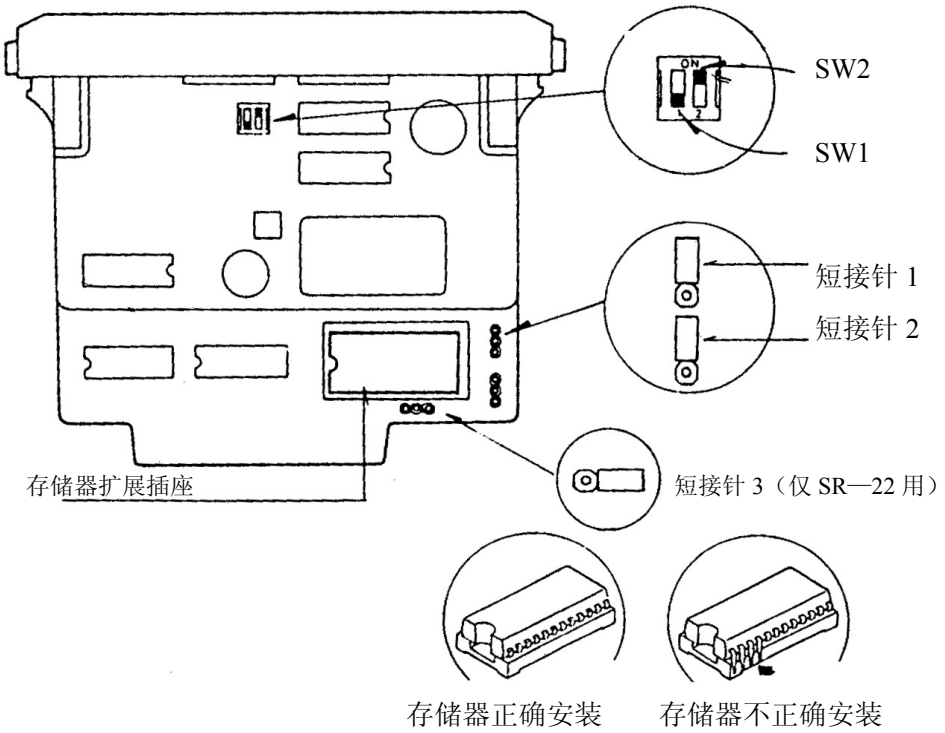
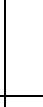
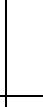
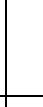
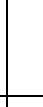
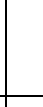


图 1.14 SR-21/22 CPU 模块开关及短接针图

SR-21 CPU 出厂时所配备的用户存储器是CMOSRAM，用户程序容量为 0.7K 字。为了扩展 SR-21 CPU 的用户程序容量，CPU 模块上有一存储器扩展插座，可插 RAM/ROM 芯片。在此插座上可插入 CMOSRAM 芯片 HM6264LP-15 使 SR-21 的用户程序容量达到了 1.7 K 字。另外还可在此插座上插上 EPROM 芯片 HN27256G-25，SR-21CPU 的容量为 1.7 K，SR-22 的容量为 3.7 K 字。SR-22 CPU 模块出厂时其用户存储器是 CMOSRAM，用户程序容量为 3.7 K 字。

CPU 模块的用户存储器扩展时，除了在存储器扩展插座上插上相应芯片外，同时还必需设置好 SW2 开关，短接针 1，短接针 2，短接针 3（仅 SE—22 用）的位置。详细情况见下表：

SR-21/22 CPU 模块的硬件环境和存储器配置表：

存储器类型和容量 / 开关 短接针	标准容量 RAM（701 字）（SR—21）	扩展 RAM（1724 字）（SR—21）	EPROM（1724 字）（SR—21）	扩展 RAM（3701 字）（SR—22）	EPROM（3701 字）（SR—22）
存储器芯片	不安装	HM6264LP-15	HN27256G-25	HM62256LP-15	HN27256G-25
SW2					
短接针 1					
短接针 2					
短接针 3	无	无	无		

如果 CPU 硬件环境和存储器配置不符合上表所列，则给出出错代码 30。

注意：在安装 EPROM 之前应拨掉短接针 EF；如果短接针 EF 不断开的话，电池将在一个短时间放电。

SE-22 CPU 的用户存储器采用 FLASHROM，用户程序容量为 2.5K 字。

§1—4—4 SE-22 CPU 模块的通讯

一、SE-22 CPU 通讯的功能

SE-22 CPU 模块上具有一个 RS232C 规格的通用通讯口，可以和下述机器进行通讯。

1、上位通讯连接用（CCM 协议，从机功能）

（1）上位个人计算机（CCM 用通讯软件）

（2）上位 PLC 机（SU—6B，SU—6M，SG—8B，采用通讯模块 U—01DM，G—01DM）

2、汉字显示设定单元 CL—02DS。

3、工业触摸屏图形显示器 GP 系列。

4、无协议串行收发信

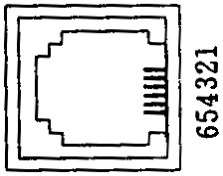
（1）串行数据发信器（条形码读入机器等）

（2）串行数据收信器（打印机等）

（3）串行数据收发信机（温控仪等）

二、SE-22 通讯口的引脚分配

SE-22 CPU 模块通讯口采用 6 芯电话插头，其引脚分配如下表：

端口 1	针号	信号名
	1	0V
	2	+5V
	3	RXD
	4	TXD
	5	RTS
	6	CTS

三、SE-22 CPU RS—232C 通讯口的性能

项 目	规 格	系统参数设定
传送规格	RX—232C	———
传送距离	MAX 15m	———
传送方式	半双工非同期，7/8 位，奇/偶校验，1/2 停止位，HEX/ASCII 方式	022，023
传送速度	300，600，1200，2400，4800，9600，19200 参数选择	021
错误检查	奇偶校验，LRC	022
通讯协议	CCM 协议（仅用子局 1-90）或固定协议或无协议	020

四、SE-22 CPU RS-232C CCM 通讯。

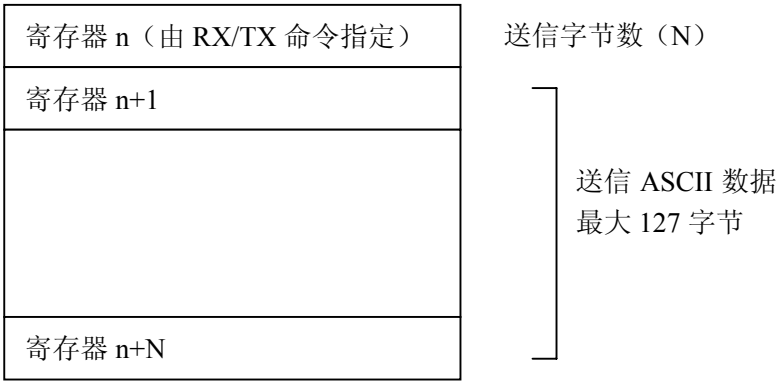
参照 G—01DM 技术资料。

五、SE—22 CPU RS—232C 固定协议通讯。

1、串行送信

子局号设定为 96。

数据的传送由 RX/TX 命令进行。

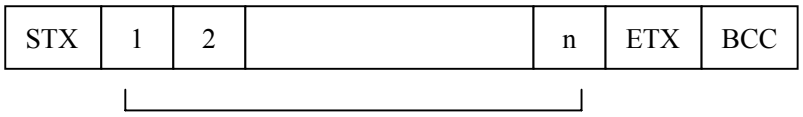


2、串行受信

A 型串行受信：

子局号设定为 97。

数据格式：



ASCII 数据，最大 127 字节

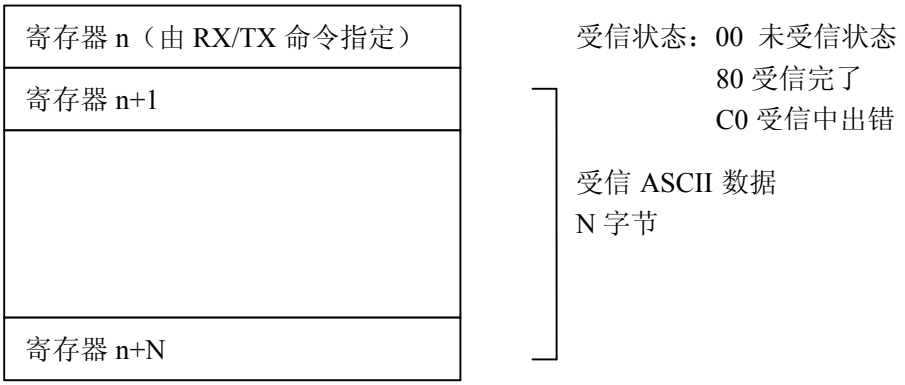
BCC=STX—ETX 间的异或和

在数据接受完毕后，对数据格式，BCC 进行检查。

正常：对终端设备发送“ACK”

异常：对终端设备发送“NAK”

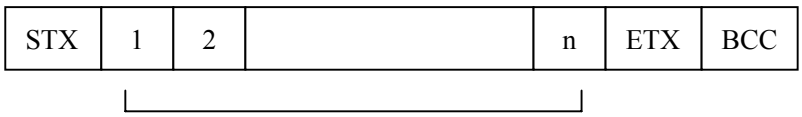
受信数据存放于 TX R#×××指定的寄存器中，格式如下：



B 型串行受信：

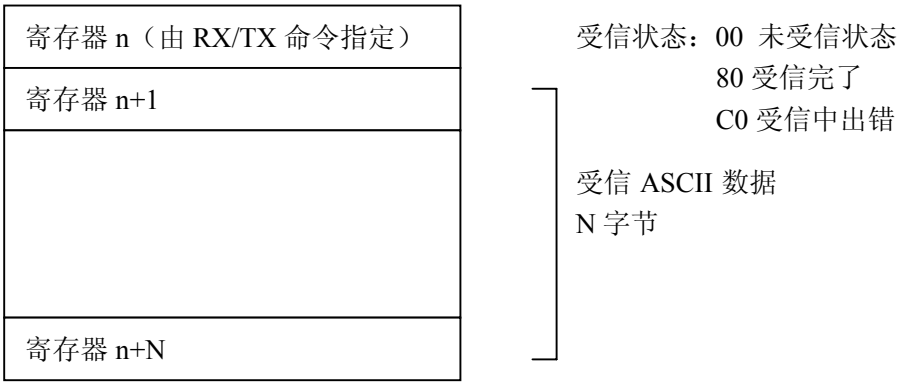
子局号设定为 98。

数据格式：

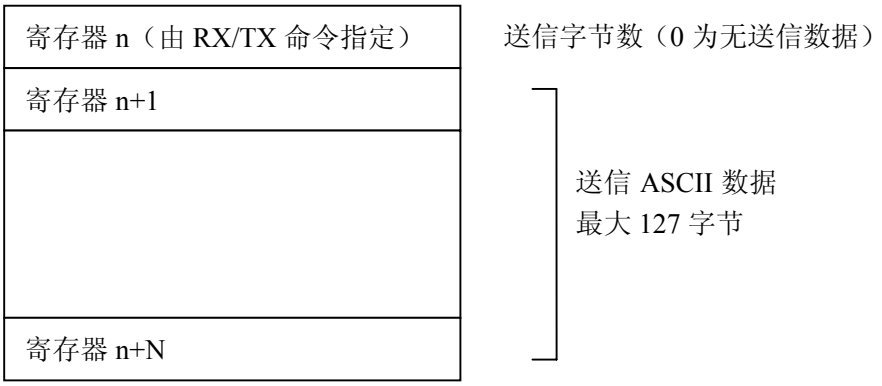


ASCII 数据，最大 127 字节

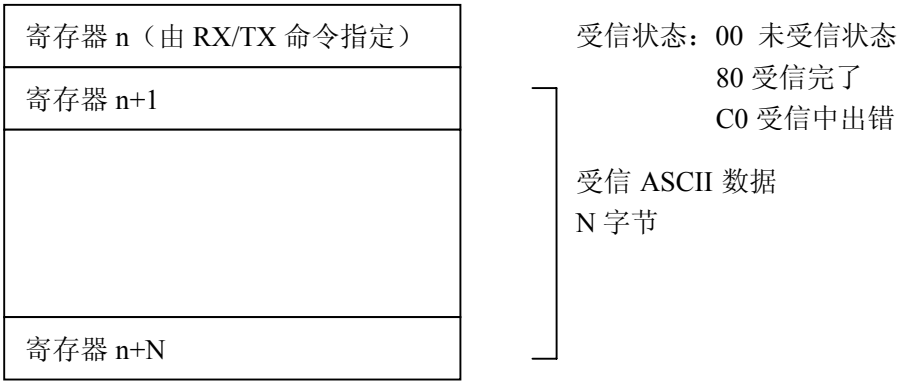
BCC=1——ETX 间的异或和
在数据接受完毕后，对数据格式，BCC 进行检查。
正常：对终端设备发送“ACK”
异常：对终端设备发送“NCK”
受信数据存放于 TX R#×××指定的寄存器中，格式如下：



3、串行送受信
子局号设定为 100。
数据的送受信由 RXTX 命令进行，格式为：



受信数据的终了码由系统参数 014，015 指定。
例如：014=0DH，015=0AH，则受信数据的终了码为 0D0AH
014=00H，015=0DH，则受信数据的终了码为 0DH
受信数据的格式为：



六、SE-22 CPU RS-232C 无协议通讯

子局号设定为 102。

无协议通讯通过对环形送信缓冲区和环形受信缓冲区的操作完成。环形送信缓冲区有数据时，系统自动将数据送住 RS-232C 端口。RS-232C 端口有数据时，系统自动将数据接受下来存入环形受信缓冲区。程序可以通过 READ，WRITE 指令对环形送信缓冲区和环形受信缓冲区进行操作。

WRITE 指令 SE-22

送信数据——→环形送信缓冲区——→RS-232 端口

READ 命令 SE-22

受信数据←——环形受信缓冲区←——RS-232 端口数据

环形送信缓冲区和环形受信缓冲区的大小为 256 字节。

环形送信缓冲区和环形受信缓冲区中的数据由寄存器 R746—R747 可以得知。

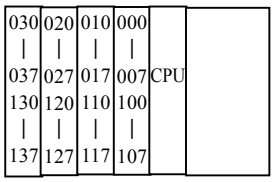
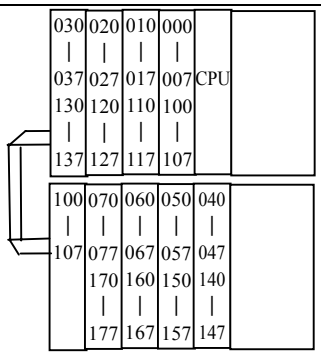
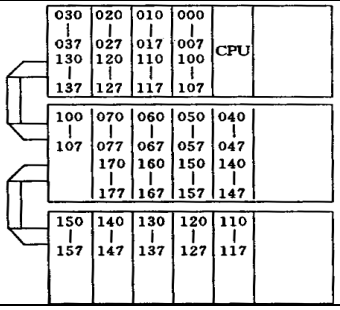
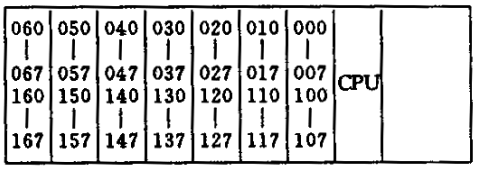
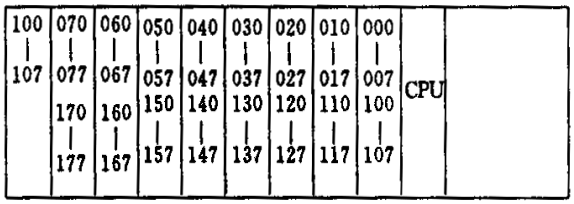
七、SE-22 功能存储器号码变换表：

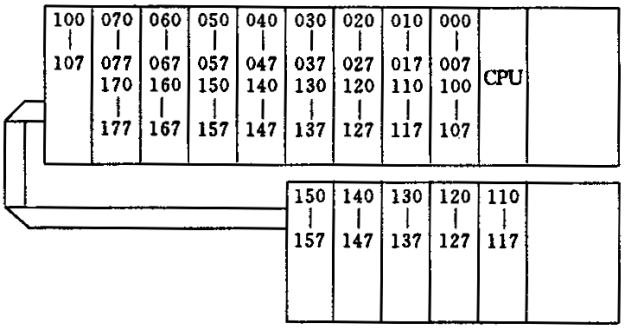
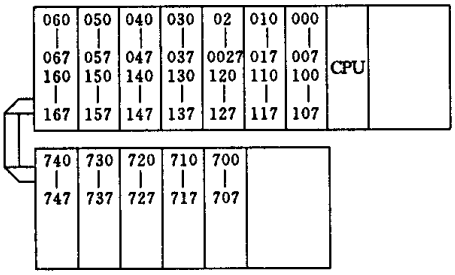
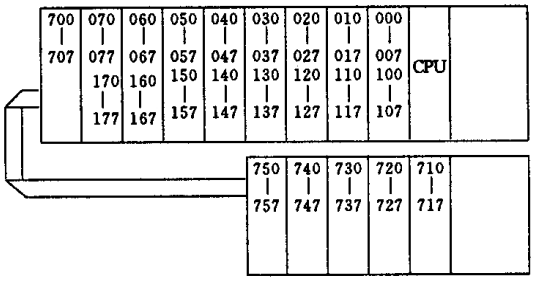
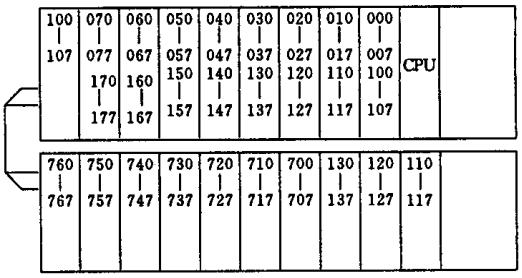
功能码	功　　能	SE-22	内部定义号
31h	读写数据寄存器	600—677 400—577 160—377 700—777 P000—P777	01h—040h 41 h—80 h 81 h—0c8 h 0c9 h—0e8 h 0801 h—0A00 h
33 h	读写（bit）I/O，内部寄存器，移位寄存器、级	000—157	01—70h
36 h	读写运行方式	00h	Run0101h Prg8080h 机种：0808h
37 h	读写用户程序	000—2560	000—9ffh
39 h	状态读写	…	…

§1-5 系统配置

§1-5-1 系统配置表

这些例子表示可用的框架和 I/O 点的许多配置中的一些例子，配置取决于实际使用所需的 I/O 点数和 I/O 模块的类型（4、8 或 16 点，以及所使用模块的负载单位）。

框架型号	I/O 模块类型		I/O 点	配置
	8 点模块	16 点模块		
E-01BJ E-02B E-02B-C 使用这些框架中的一个框架	4	—	32	
	3	1	40	
	2	2	48	
	1	3	56	
	—	4	64	
E-02B E-02B-C 使用这些框架中的二个框架	9	—	72	
	8	1	80	
	7	2	88	
	6	3	96	
	5	4	104	
	4	5	112	
	2	6	112	
	1	7	120	
	—	8	128	
E-02B E-02B-C 使用这些框架中的三个框架	14	—	112	
	12	1	112	
	10	2	112	
	8	3	112	
	6	4	112	
	4	5	112	
	0	8	128	
	12	2	128	
E-05B 使用一个框架	7	—	56	
	6	1	64	
	5	2	72	
	4	3	80	
	3	4	88	
	2	5	96	
	1	6	104	
	—	7	112	
E-03B E-03B-C E-04B E-04B-C 使用这些框架中的一个框架	9	—	72	
	8	1	80	
	7	2	88	
	6	3	96	
	5	4	104	
	4	5	112	
	2	6	112	
	1	7	120	
	—	8	128	

框架型号	I/O 模块类型		I/O 点	配　　置
	8 点模块	16 点模块		
E-03B / E-03B-C 使用这些框架中的一个框架；E-02B / E-02B-C 使用这些框架中的一个框架	14	—	112	
	12	1	112	
	10	2	112	
	8	3	112	
	6	4	112	
	4	5	112	
	—	8	128	
	12	2	128	
E-05B + E-02B	12	—	96	
	11	1	104	
	10	2	112	
	9	3	120	
	8	4	128	
	7	5	136	
	6	6	144	
	5	7	152	
E-04B / E-04B-C 使用这些框架中的一个框架 + E-02B / E-02B-C 使用这些框架中的一个框架	14	—	112	
	13	1	120	
	12	2	128	
	11	3	136	
	10	4	144	
	9	5	152	
	8	6	160	
	6	8	176	
E-04B 使用二个框架	19	—	152	
	18	1	160	
	17	2	168	
	15	3	168	
	13	4	168	
	11	5	168	
	8	7	176	
	7	8	184	

注：1、在 E-01BJ 框架上安装 16 点 I/O 模块，必需特别注意电源的容量和电流消耗之间的关系。

2、表中 I/O 定义号为所有情况的组合，当使用 16 点 I/O 模块时，写在槽下部的 I/O 定义号（16 点模块后 8 点定义号）被占用了。应该注意，被 16 点模块后 8 点所占用的定义号的那个槽，因该槽 I/O 定义号（写在槽的上部）已被占用，所以不能安装 8 点模块。

3、在存储器插座安装扩展 CMOSRAM 芯片或 EPROM 芯片时，芯片缺口应与插座芯片缺口的方向相同。

4、当不使用（E-01D）拨盘接口模块，和（E-01Z）高速计数模块时，160-177 定义号不作内部线圈使用，而可作为外部 I/O 定义号使用，使 I/O 的最大配置由 168 点扩大到 184 点。

§1—5—2 系统估算

要设计一个 PLC 系统，首先要计算出输入（一般为按钮、开关、过载继电器等）和输出（一般为电磁阀、马达启动器、指示灯）在计算输入/输出点数时，中间继电器、时间继电器及计数器一般不要考虑。根据输入/输出点数后确定框架及存储器的容量。

下面是估算框架和模块需要量的步骤：

1、区分不同类型、不同电压的输入和输出，并计算各自的总数（如：4 点 115VAC 和 5 点 24VDC 输入，3 点 115VAC 和 1 点 24VDC 输出）。

2、根据点数，算出各种类型的模块的需要量（用 4 点，8 点还是 16 点模块，输出模块需要多大的负载能力，这些问题都要考虑）。

3、是否需要用高速计数器，拨盘接口模块，模块量模块？

4、算出总共需要多少个 I/O 模块，如模块数为 1—4 个可用一个 5 槽框架；5 一 9 个，可用两个 5 槽框架或一个 10 槽框架；10—14 个，可用三个五槽框架，或一个 10 槽框架加一个 5 槽框架，或两个 10 槽架；在 SR—21 中，用两个 10 槽框架时，最多可用 19 个 I/O 模块。

5、如框架数多于一个，需要用扩展电缆，两个框架用一根，三个框架用两根，如框架有空的槽，应用填充模块（E-DMY）盖上。

CPU 模块是每个 PLC 系统必需的，SR—21 的 CPU 的标准用户存储器容量都是 700 字的 CMOS RAM，可以扩充到 1724 字。如果采用 EPROM 存储器，容量也是 1724 字。对于一般复杂程序的系统，I/O 点数在 64 点以内时，700 字是足够的，I/O 点达到 184 点时，1724 字也足够用了。如果您的系统很复杂，您可以找一个典型环节，试编一段程序，算一下所用的语句数，从而推算出整个系统所需的存贮容量。如果这样还无法估计容量，则需进行实际的编程，以确定所需存储器的容量。如果用户程序超过 1.7K，则选用 SR—22PLC 或 SE—22PLC。SE—22 的用户存贮器是容量为 2.5K 的 FLASH—ROM。

当配置一个系统时，您还可以考虑是否需要下面的硬件：

1、EPROM 存储器及其写入单元

实际上，对于大多数需要设计的 PLC 系统，并没有继电器控制逻辑图可供参考，在这种情况下，I/O 点数可用系统的控制要求来计算。输入是 PLC 为执行它的指定功能所要求的信号，因此输入点数基本上就是由现场提供给 PLC 的信号数（如按钮、开关等）。输出则是 PLC 所要控制的对象。（电磁阀、马达启动器、接触器、指示灯等）。您可以将输入输出信号及其电压类型、电流大小分别出一张表，即可方便地算出 I/O 模块需要量。

当然，当您确定系统配置时，也应该考虑到 PLC 本身所具备的内部功能对您的系统影响。

第二章 PLC 操 作

本章介绍 PLC 的操作，主要介绍编程器的面板组成部件及其功能。通过本章学习将使您对编程器键和各种编程器操作所要求的键操作顺序有所了解。

§2—1 编程器

编程器（图 2.1）是用来输入一个新的程序或检查预先输入的程序，如果需要，可对一个预先输入的程序进行修改，监控输入或输出点的状态，显示寄存器内容和显示定时器或计数器的累积值。另外 SR—21PLC 用编程器可输入一个通行字来保护程序的内容。在未经允许或无意中不能输入或改变原有程序。它的面板组成部件和功能在下面章节中介绍。编程器功能的详细说明参看第 3 章编程。

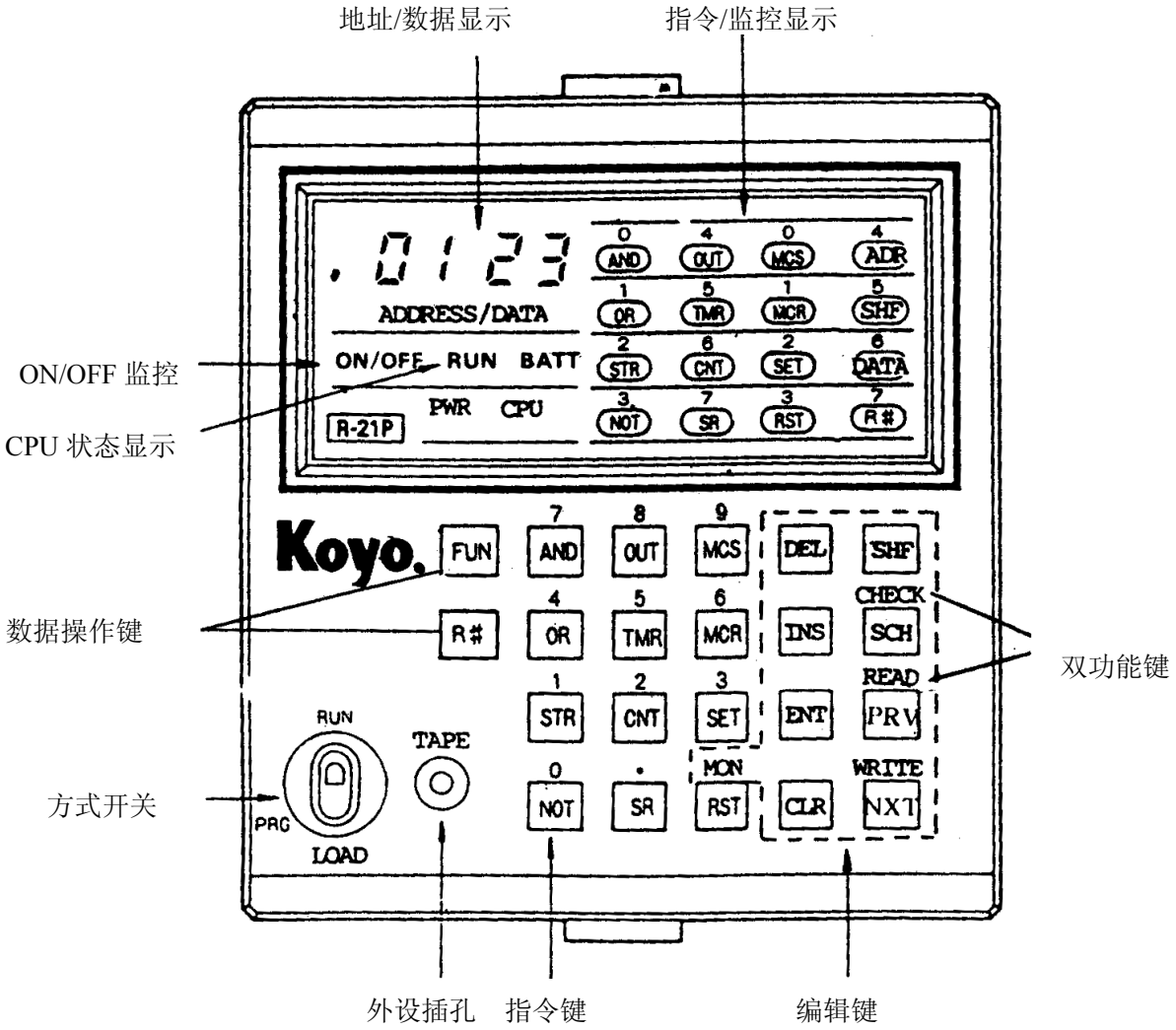


图 2.1 编程器面板图

§2—1—1 编程器面板组成部件

本节介绍图 2.1 所示的编程器面板组成部件、各部件的名称。下面对各部做具体介绍。

1、**方式开关**

这是一个用来选择 PLC 操作方式的三位置开关。该开关在不中断 AC 电源的情况下，根据需要随时可以改变位置。上边位置（RUN）是允许执行程序并输出。CPU 将对已存入的程序进行扫描，并可以显示定时器/计数器和继电器触点的状态。但在 RUN 方式时不允许改变程序，可改变定时器计数器的预置值。在中间位置（PRG），可以输入新的程序，也可以修改原来输入的程序，但它不执行程序 。下边位置（LOAD）把编程器与外部装置相连接（例如录音机）。当在 LOAD 位置时不执行程序。

警 告

SR—21CPU 安装完毕经调试正常运行以后，可编程序控制器总是处在 RUN（运行）方式，再装上编程器时编程器一定要放在 RUN（运行）方式，否则 PLC 停止工作。

2、**地址、数据显示器**

这是一个四位显示器。以十进制形式来指明存在 PLC 存贮器中的用户程序地址，或者显示作为逻辑组成部分的数据。为了表明正在显示的是地址，在每位数字的右下方有一点（例如 0.1.2.3）。同时右上方 ADR（地址）后面的发光二极管将亮。

3、**状态显示器**

由五个发光二极管来显示 SR—21 下列的功能或状态。

ON/OFF：当在 RUN（运行）方式时，这个发光二极管指示编程器所显示指令的离散定义号（I/O，内部线圈和移位寄存器）的状态。当定义号激励时它点亮，不激励时它熄灭。此功能在用编程器调试程序时很有用。

RUN：CPU 处在运行方式并在执行逻辑时灯亮。

BATT：当内部锂电池的电压低于规定值时灯亮，此时该电池在一星期内需要更换（见第 6 章）。锂电池是用于在没有电源情况下保持存在 CMOS RAM 存贮器中的程序。当电池电压是正常的或不接电池时，该灯是灭的。

CPU：当内部自检检查出 PLC 硬件有故障时该灯亮。

PWR：当内部电源产生正常直流电源时灯亮，如果灯不亮，应检查框架电源。

当安装上编程器后，面板上面的 4 个发光二极管（RUN、BATT、PWR 和 CPU）重复 CPU 指示灯的动作。

4、**指令/监控显示**

在编程器上的 16 个发光二极管是用来显示存储器的逻辑指令。在编程时，他们反映进入 CPU 存储器以前所编的程序。前 12 个发光二极管的功能详见下面第 5 条（指令键）。右边 4 个发光二极管为专用功能见下面。

ADR：当显示一个地址值时该灯亮。地址是由十进制数显示。从 0000 开始。到用户程序最后一个地址时结束。

SHF：当操作者选用 SHF 功能转换键时灯亮，此时编程器上所有键就执行第二个功能，该功能一直到按 ENT 或 CLR 键才结束。

DATA：在监控寄存器的内容及用地址/数据来显示寄存器中所包含的数据（4 位 BCD 值）时灯亮，并且可显示 2 个连续寄存器的内容。

REG：在监控寄存器的内容及用地址/数据来显示所选寄存器的寄存器号时该灯亮。按压 MON 键可以轮流使 REG 和 DATA 灯亮。

当使用监控功能时这 16 个发光二极管也显示连续 16 个 I/O 的状态，英文字母上面的数字是在此功能时用的。

5、指令键

当在 PRG（编程）方式时，这 12 键被用来选择所要求的功能和输入逻辑。当在其前面按 SHF 功能转换键后，此时逻辑键将用来输入上部所标的数字值。当数字值输入时，其数值可在地址/数据显示器上显示出来。小数点仅用于输入定时器的预置值。监控功能将在后面作为轮换功能键介绍，这些键所标志的功能将在下面讨论。

AND：把两个触点或两组触点串联联接。

OR： 把两个触点或两组触点并联联接。

STR：起始新的逻辑行，以及把现行逻辑结果存入先进后出的下压堆栈中。

NOT：把状态取反，使常开触点变为常闭触点。AND NOT 是表示串联一个常闭触点，OR NOT 表示并联一个常闭触点。

OUT：定义一个输出线圈并使一个逻辑行结束。

TMR：用来规定一个定时器功能，并表示一个逻辑行的结束。

CNT：用来规定一个计数器功能，并表示一个逻辑行的结束。

SR ：移位寄存器，由三个逻辑行组成，即输入，时钟和复位。移位寄存器定义号由起始位和终止位级成。并表示逻辑行的结束。

MCS：主控开始，用于建立一条主控母线，用于由一组逻辑来控制许多线圈。

MCR：主控复位，结束主控功能，主控母线以主控开始作起始以主控复位作结束，因此在程序中 MCS 和 MCR 的数据必须相同。

SET：用于锁存线圈。一经接通后则锁存线圈，该线圈的接通状态将不受内部线圈 376（禁止输出）的影响。

RST：用于解除 SET 功能的锁定，即将被 SET 接通的锁存线圈断开（复位）。

6、编辑键

该 8 个编辑键用于修改 CPU 存贮器中的内容或更改显示地址中的内容。也可修改先前编入的逻辑（删除或插入）。为了防止你无意中打的指令进入 CPU 存贮器中，在 SR—21 中为了确保操作的正确性。在修改程序前需打入预先约定的操作。然后进入编辑状态，每个键的详细说明如下：

DEL：删除键，在编程器上显示逻辑后，该键用来实现从 CPU 中删去某个逻辑。在双键程序中只要删除第一次按键，在按 DEL 键后必须再按 PRV 键才能执行删除功能。

INS：插入键，用于在已完成的程序之间插入逻辑功能。在确定了新功能要放置的位置后，显示所要插入功能的下一条功能或功能地址，然后接着按插入键（不是写入键）和确认键 NXT（下移）键，这样新的逻辑就会建立在所显示的功能或地址之前。

ENT：写入键，用来在开始建立 CPU 程序时，完成逻辑输入，一般在输入一条逻辑指令结束时按此键。

CLR：当选择 CLR（清除）键时，它清除编程器预先写入的未按 ENT 键的指令；如果错误代码被显示，这个键将证实错误并显示错误处的地址，同时清除错误代码，使编程器恢复到正常状态；当监控一个程序时，按清除键将代替指令而显示该指令的存贮器地址。当按下述顺序逐个按 CLR、SHF、3、4、8、DEL、NXT 键时，将存贮器的内容全部清除。

提　　示

使用清除程序操作时应该谨慎，该操作仅用于清除你输入的全部程序。如果仅是清除单个或部分指令时，应选用 DEL 和 PRV 键来逐条删除指令。

SHF：这个键（功能转换）用来实现第二个（上部标志）功能，并使在显示区的 SHF 发光二极管亮。功能转换操作是利用 CLR 键或 ENT 键结束的。在选择数字之前先按功能转换键然后按数字键。例如，输入一个定义号为 12 的常开触点来作为一逻辑行的第一个触点，此时键的操作顺序应是 STR，SHF，1，2，ENT。

SCH：这键（搜索）允许搜索已输入程序中某一个逻辑功能。先输入需要搜索逻辑功能的指令然后按搜索键（不是 ENT 键）。存贮器将从现行位置开始搜索直到找到相符的逻辑功能或者搜索完所有的存贮器为止。连续按这个键将会对存贮器进行重复搜索工作。如果搜索不到所需逻辑，将显示错误代码 E—99。

PRV：当编程器在显示逻辑或监控 I/O 状态时，使用 PRV 键将显示上一个逻辑功能或 I/O 状态。连续按这键将使地址朝减少方向显示，直到存贮器地址是 0 为止。

NXT：当编程器在显示逻辑或监控 I/O 状态时，使用 NXT 键将显示下一条逻辑功能或 I/O 状态。连续按这键将使地址朝增加方向显示直到存贮器结束为止。当编程器显示地址时按 NXT 键则使编程器显示该地址的指令。按 CLR 键，编程器显示该指令的地址。

7、功能转换

当采用功能转换键时将使大多数改变他们原来的功能，执行编程器面板上键所对应上方的功能。在没按功能转换键时这些键将仍执行原来的功能。在功能转换情况下这些键大多数是完成数据操作（数字 0—9 及小数点）。当开始编程时，首先先按正常键而不用功能转换（如 AND，OR，NOT，TMR 等等）。而功能转换键用于在程序中需输入数据的场合。

在功能转换下的其它 4 个键介绍如下：

MON：监控在运行方式进行，该功能允许用户监控 2 组（8 点为一组）连续定义号的 I/O 状态在选择监控 I/O 点定义号时，仅需指定第一组 8 点的定义号，此时 PLC 能自动显示该组和紧接着下一组共 16 点 I/O 的实时状态。例如你想监控 10 至 17 号 I/O 时，可指定 10 号或 10-17 号中任一个定义号。这是因为只有选择该组定义号，PLC 才能显示这组定义号中 I/O 的状态。例如，输出定义号为 043，则本组（040—047）中的 I/O 状态将被显示。该 I/O 状态可看编程器上的第一组 8 个 LED 逻辑指示灯（AND，OR，STR 等等）。下面 8 个定义号（即 050—057）也显示出来，他们的状态由后面的 8 个发光二极管将相应接通或断开。当 I/O 状态改变时，发光二极管的状态也相应地改变。

如需显示另一组的 I/O 定义号，可以用 PRV 或 NXT 键来向前（40，30，20 等）或向后（50，60，70 等）显示有前定义号中的状态。显示定义号的最大值为 577。当达到 570 后将回到 000（NXT）或 000 到 570（PRV）。定时器或计数器的现行值也能利用输入 SHF，6XX（定时器或计数器定义号）及按 MON（监控键）来实现监控。

WRITE：该功能可与音频盒式磁带录音机或某些其它外围设备连用。把 CPU 中的逻辑有效地传送到外围设备中。每个编程器提供一根与外围设备连接的音频电缆。操作时，先把设备接通同时把编程器上的开关打到 LOAD 位置，按 WRITER 键，传送开始，这时就把 CPU 中的逻辑写进盒式磁带或其它外围设备中。

READ：这功能用于连接音频盒式磁带录音机或其它外围设备，把盒式磁带或其它存贮设备中的内容装载到 CPU 的存贮器中，每个编程器提供与外围设备连接的音频电缆，通过该电缆连接编程器和外围设备。把编程器的开关打到 LOAD 位置并按 READ 键，这就开始把磁带中存储的程序装进 CPU 存贮器中。

CHECK：这功能用于校对音频盒式磁带录音机或其它外围设备的程序传送是否有效。在传送进行完以后，应该像 READ 那样读回来，但这时应用 CHECK 键而不用 READ 键。此校对操作既不改变 CPU 的逻辑也不改变外围设备中的数据，仅在两个信息源之间进行数据比较确保在录制过程中不出错。

8、**数据操作键**

这两键是在 SR—21PLC 编程输入数据操作时使用。这两个键的功能介绍如下：

F：这个键为数据操作功能键。该数据操作键 F 必须与后面二位表示相应数据操作功能的数字键连用（例如 F50，F62，F74，F82 等），数据操作和他们对应功能号的表格见第四章编程篇。在此数字值之前不需要按功能转换（SHF）键，这功能转换（SHF）键在 SR—21 编程时用于其它数字值时是必须要按的。

R：这个键有两个功能，它既可用于选择数据寄存器或定时器/计数器累积值寄存器，也可用于在编程数据操作时选择一组定义号。在 SR—21PLC 中，寄存器是以字（16 位）进行操作的。当确定了一个三位数据寄存器定义号后（400 到 577），实际选中了两个连续的 8 位寄存器，把数据写入被选中的数据寄存器或者从其中读出数据。例如，如果按 R 键后接着输入定义号 410，则在数据操作中表示，寄存器 410 和 411 被选中，如果输入 R610，在数据操作中对应那个定义号的定时器/计数器累加值寄存器将被选中。

一个寄存器可用为一个定时器或计数器的预置值。在此情况下，编程时，规定寄存器号（R）直接写在该定时器或计数器的后面。

一组定义号是对应于 8 个连续离散定义号为一组的一组数或二组数。这些定义号可以是实际的 I/O 点，内部线圈或移位寄存器。例如，如果按 R 键后接着输入 37，一组定义号就是 370 到 377。如果输入 R5，二组定义号就是 I/O 点 050 到 067（即 050—057 和 060—067 二组定义号）。

注：是一组数或是二组数由数据运算指令确定。

9、**外设接口**

这是一个允许连接音频盒式磁带录音机的串行接口，用于录制（保存）用户存贮中的程序并校核磁带的内容与用户存储器的内容是否一致，并且可把存有用户存贮器内容的磁带保存起来。反之，可从录有用户的磁带把用户程序送到 CPU 的用户存贮器中。

§2—2 程序检查和错误代码

当用编程器输入梯形图逻辑程序时，CPU 自动地执行对编程器所选用的数据和操作进行检查。检查键操作顺序、定义号范围等功能输入是否正确。在检查期间如检出错误，则将在数据显示器上字母 E 后跟两位数代码（01—99）显示出来。当方式开关拨到 RUN 方式的时候，CPU 将实时检查程序是否有错。在输入一个程序以后，应用编程器检查进入的程序是否有错误。

表 2.1 讨论了每个错误代码，它的原因和消除错误的方法。

表 2.1 错误代码定义

代码	方式			含义	原因	改正方法
	RUN	PROG	LOAD			
E1	×	×	×	不正确的操作	操作员试图执行非法的操作例如在 RUN 方式改变程序	检查操作，按 CLR，重新开始正确的操作
E2	×			程序结构有错	在 RUN 方式时，CPU 检出程序中有错误，例如输入模块定义号被用作线圈定义号	拨到编程方式，按 CLR 将显示故障逻辑地址，按 NEXT 显示内容，然后进行修改。
E3	×			超出堆栈容量	压入堆栈操作超过 8 级	拨到编程方式，按 CLR，编程器将显示第 9 个 STR 错误的位置，检查逻辑，必要时修改程序
E5	×			线圈定义号重复	线圈（输出、内部线圈，定时器或计数器）重复用作输出	拨到编程方式，按 CLR，编程器将显示使用同一定义号两个线圈中第二个线圈的位置，输入另一个线圈定义号。
E6	×			主控不完整	程序中 MCR 比 MCS 多	拨到编程方式，按 CLR，编程器将显示一个不匹配的 MCR，用删去 MCR 或加入 MCS 来修改程序
E7	×			移位寄存器或计数器不完整	计数器或移位寄存器少一个或多一个逻辑行	拨到编程方式，按 CLR，编程器将显示错误功能，加入所要求的复位，时钟将消除逻辑行
E8		×		操作不正确	操作员试图在一个需要占用 2 个字指令中第 2 字插入逻辑	按 CLR
E9	×			逻辑行不完整	继电器梯形图逻辑行没有接线图，继电器触点左边不完整或悬挂。	拨到编程方式，按 CLR，编程器将显示第一个没结束的逻辑元素，紧接着这个元素把逻辑行完整，或者删去元素以去掉不完整的逻辑行。

代码	方 式			含 义	原 因	改正方法
	RUN	PROG	LOAD			
E11		×		存贮器溢出	CPU 中存贮器已存满此时再加逻辑	按 CLR，重新组织程序，使得逻辑不超过限制。
E14	×	×		F89 指令不配对（仅 SE—22）	F89 指令不配对	F89 指令应成对出现
E21	×	×		奇偶出错	CPU 已检出它的内部存贮器中奇偶结构出错	拨到 LOAD 方式，按 CLR，把预先录制好的磁带重新加载存贮器或者清除存贮器并用人工进行重加载。如果 BATT 灯不接通，故障不能清除。更换 CPU 模块。
E22		×		保密口令出错	输入了不正确的保密口令	输入正确的保密口令
E25			×	比较出错	外部装置（例如磁带中）的内容与 CPU 存贮器中的内容不一致	按 CLR，检查出正确的程序编号或磁带，如果正确，重录磁带或重新加载 CPU
E28			×	录制信号太弱	放音信号电平低，例如录音机音量低于容许的电平	调整录音机或其它外围装置的音量，如果延长接通时间重新开始操作，这样可得到可靠工作。
E29	×			通讯错误（仅 SE—22）		
E30		×		CPU 存储器配置不对	用户存储器开关及短接针设置不正确	正确设置用户存储器开关及短接针
E31		×		存放编译代码的 RAM 不足	用户程序太长	缩短程序或修改程序结构
E32		×		用户 EPROM 检查出错	用户程序写入 EPROM 时不正确	用紫外线擦除 EPROM，把用户程序从新写入 EPROM 存储器
E41	×	×		快速子程序指令溢出	快速子程序指令超出 70 条	缩短快速子程序指令，使它不超过 70 条。
E42	×	×		快速子程序中 CNT 指令超出	快速子程序中 CNT 指令超过 4 条	使快速子程序中 CNT 指令不超过 4 条。

代码	方式			含义	原因	改正方法
	RUN	PROG	LOAD			
E43	×	×		快速子程序中，CNT 预置值设置不正确	快速子程序中，指令 CNT6××，R×××中的R×××未使用R600—R677。	使指令 CNT6××，R×××中的 R××× 改用 R600—R677 之间的定义号。
E44	×	×		快速子程序中使用了禁止指令	快速子程序中使用了禁止使用的指令	修改快速子程序，删除禁止使用的指令。
E50	×			环形受信缓冲区满（仅 SE—22）		
E51	×			环形发信缓冲区满（仅 SE—22）		
E99	×	×		无效搜索	找不到要检索的功能	按 CLR（对其它功能进行搜索）

§2—3 操作顺序

为了有效的和高效率地输入梯形图程序，必需了解基本 PLC 操作顺序，熟悉每个键单独的用法和其它键一起使用的顺序。编程器是一个输入、编辑和监控程序的极好的工具。表 2.2 列出了各种操作，输入那些操作需按的按键，以及能在那些操作的方式开关位置执行。在表中用字母 R（RUN），P（PROGRAM），和 L（LOAD）来表示方式开关的每个位置。

表 2.2 PLC 操作顺序表

操作		按键或操作顺序	方式开关位置		
			R	P	L
清除所有用户存贮器内容		CLR SHF 348 DEL NXT		×	
回到程序存贮器中 0000 号地址		SHF NXT	×	×	
显示现行地址		CLR	×	×	
显示现行功能		NXT	×	×	
显示下一步功能		NXT（在上面键操作后）	×	×	
显示上一步功能		PRV	×	×	
到指定地址		SHF（地址）NXT	×	×	
到下一个地址		在上面操作后，按 NXT	×	×	
清除显示的功能		CLR	×	×	
写入一指令	一语句指令	（功能）SHF（定义号）ENT		×	
	二语句指令	（功能）SHF（定义号）ENT 或 FUN（功能号）ENT 数据或 SHF（定义号）ENT 或 R#（寄存器号）ENT		×	
改写一指令		用搜索指令搜索出要改写的指令，然后用上述方法输入新指令		×	
搜索一个指定功能		（功能）SHF（定义号）SCH NXT	×	×	
搜索一个指定的定义号（触点）		SHF（定义号）SCH NXT	×	×	
在显示的功能（或地址）之前插入功能		（功能）SHF（定义号）INS NXT。如果是二语句指令，只要插入第一语句，第二语句自动跟入。		×	
删去一功能		显示删去的功能或地址，然后按 DEL PRV。如果是二语句指令，只要删除第一语句，第二语句同时删去。		×	

操　　作	按键或操作顺序	方式开关位置		
		R	P	L
插入一 END 指令（即在程序中插入一空地址，空地址后的指令不起作用）	CLR　SHF　INS　NXT		×	
检查程序是否有错，如果没检出错误，显示下一个可使用的地址（搜索空地址）	CLR　SCH	×	×	
改变定时器或计数器的预置值	SHF（预置值）ENT	×		
监视一组 16 个连续定义号，（I/O，内部线圈，移位寄存器）的通断状态。	SHF（开始的定义号）MON	×		
监视定时器/计数器触点的通断状态	TMR（CNT）SHF6××MON	×		
监视任何线圈或触点的通/断状态。	由编程器上 ON/OFF 发光二极管的接通或断开来显示（编程器上显示）定义号的状态	×		
监控定时器或计数器的累积值	SHF（6××）MON 为了监控上一个或下一个 T/C 按 PRV 或 NXT	×		
监控数据寄存器内容*	R（寄存器定义号）MON	×		
监控下一个寄存器内容	NXT	×		
监控上一个寄存器内容	PRV	×		
输入或改变所显示的寄存器的内容能够输入 4 位 BCD 值（16 位）	SHF××××ENT。当数据内容被显示时，这个操作能够改写这寄存器的现行内容。	×		
强制一定义号接通（将不考虑用户逻辑）	SET SHF（定义号）ENT 定义号能够为输入，输出，移位寄存器，定时器或计数器。	×		
强制一定义号断开（将不考虑用户逻辑）	RST SHF（定义号）ENT	×		
强制定时器/计数器接通	SET SHF 6××ENT（T/C 触点）	×		
强制定时器/计数器断开	RST SHF 6××ENT（T/C 触点）	×		
从程序存储器传送到磁带上	（选择程序识别字）SHF　WRITE			×

操　　作	按键或操作顺序	方式开关位置		
		R	P	L
从磁带读到程序存贮器	（选择程序识别号）SHF READ			×
校对磁带或 PROM 与程序存贮器中的数据	（选择程序识别号）SHF CHECK			×
输入或改变一个通行字**	CLR SHF 9876 DEL SHF ××××ENT NXT	×	×	
联机顺序	CLR SHF 5678 DRL SHF ××××ENT NXT	×	×	
离机顺序	CLR SHF 1234 DEL NXT	×	×	
搜索数据寄存器定义号	CLR R#（寄存器定义号×××）CHECK	×	×	
搜索整数常数	CLR SHF ××××. CHECK （CNT 指令的设定值，F 指令中的常数）	×	×	
搜索小数常数	CLR SHF ×××. ×CHECK （0.1sTMR 指令中的常数）	×	×	
	CLR SHF××.××CHECK （0.01sTMR 指令中的常数）	×	×	
	CLR SHF .××××CHECK （特殊 TMR 指令中的常数）	×	×	
RAM 数据缓冲区的监视	CLR SHF ××××NXT ××××≥5000 为相对序号，5000 相对于 RAM 数据缓冲区序号（地址）0，5001 相对于 RAM 数据缓冲区序号 1……	×		
修改 RAM 数据缓冲器中的数据	SHF××××ENT	×		
ROM 数据缓冲区的监视	CLR SHF××××PRV （××××为 0000—4999）	×		
把 RAM 数据缓冲器录音到磁带上	99××　WRITE 99 为标示符，××为文件号，范围为 00—99			×
从磁带装入到 RAM 数据缓冲器中	99××　READ 99 为标示符，××为文件号，范围为 00—99			×

操　　作	按键或操作顺序	方式开关位置		
		R	P	L
校核磁带和 RAM 数据缓冲器	99××　CHECK 99 为标示符，××为文件号，范围为 00—99			×
把 ROM 数据缓冲器录音到磁带上	98××　WRITE 98 为标示符，××为文件号，范围为 00—99			×
校核磁带和 ROM 数据缓冲器	98××　CHECK 98 为标示符，××为文件号，范围为 00—99			×
将记录到磁带上的 ROM 数据缓冲器装入到 CPU 内的 RAM 数据缓冲器	96××READ 96 为标示符，××为文件号，应和 ROM 数据缓冲器录音到磁带上时所使用的文件号一致，范围为 00—99			×
校核磁带上的 ROM 数据缓冲器和 CPU 内的 RAM 数据缓冲器	96××CHECK 96 为标示符，××为文件号，应和 ROM 数据缓冲器录音到磁带上时所使用的文件号一致，范围为 00—99			×
RAM 数据缓冲器全清	CLR SHF 348　DEL NXT		×	
16 位数据寄存器的监视*** （仅 SE—22）	R8××××MON	×	×	
16 位数据寄存器的强制操作 （仅 SE—22）	16 位数据显示后，按 SHF××××ENT	×	×	
系统参数的监控（仅 SE—22）	R．×××MON	×	×	
系统参数的修改（仅 SE—22）	系统参数显示后，按 SHF××ENT	×	×	
程序存储器←→磁带 （仅 SE—22） （设定系统参数 006=00）	写入：（程序识别号）SHF WRITE 读出：（程序识别号）SHF READ			×
SE—22PLC 之间程序转录（使用 RS—232C 口）系统参数 006≠00	发送端按 READ 键 接收端按 WRITE 键			×

*当按监控键时，地址数据显示指定的寄存器定义号。为了显示寄存器的数据内容，再次按监控键。

**9876 是固定数字，通行字（××××）可以是 0001 到 9999 之间的任何数值。0000 值相当于没有通行字。当通行字是 0000 时所有编程器的功能是可操作的。

***按下 R8 键显示为 P。F 键=SHF SHF 键，R 键=SHF SCH 键。

用户一般按照上表所列的操作顺序即可操作，为了用户进一步了解操作，在下表例出了常用操作的简要说明。

常用操作简要说明

操　作	按键或操作顺序	方式开关位置		
		R	P	L
检查 PLC 逻辑	显示零地址后，按 NXT 键显示存储器零地址的逻辑内容。连续按 NXT 键可逐步检查，最后显示 END。按 PRV 键可检查上一条存储器的逻辑内容。按 CLR 键显示正在观察的逻辑地址。	×	×	
搜索 PLC 逻辑	显示零地址后，输入要搜索的逻辑功能，按 SCH 键，开始搜索。连续按 SCH 键，可连续搜索该逻辑功能。按 NXT 键可显示已搜索到逻辑功能的地址。	×	×	
修改一个逻辑单元	显示零地址后，用 NXT，PRV 或搜索功能显示要修改的单元，输入新的逻辑，按 ENT。如果输入有错误，按 CLR 清除。		×	
删除一个逻辑单元	显示零地址后，用 NXT，PRV 或搜索功能显示要删除的单元，按 DEL PRV。如需去掉删除操作，按 CLR 键（在按 DEL 键之前）。		×	
清除全部存储器内容	显示零地址后，按 CLR SHF 348 DEL NXT 如需取消该操作在按 NXT 前按 CLR 键。		×	
插入一个逻辑单元	显示零地址后，用 NXT，PRV 或搜索功能显示要插入的单元的下一个单元，输入一个新逻辑，再按 INS NXT。		×	
监控 I/O 状态 （16 点为一组进行监控）	显示零地址后， 按 SHF×××（要监控的 I/O 定义号）MON	×		
监控定时器/计数器状态	显示零地址后， 按 SHF×××（要监控的 T/C 定义号）MON	×		
显示一指定地址	显示零地址后， 按 SHF××××（存储器地址）NXT 按 SHF NXT 显示零地址	×	×	
监控数据寄存器内容	显示零地址后， R××××（寄存器字节号）MON MON	×		

操　作	按键或操作顺序	方式开关位置		
		R	P	L
改变一数据存储器内容	显示零地址后，按 R×××（寄存器字节号） MON MON 后显示寄存器内容再按 SHF××××（4 位 BCD 修改值）ENT	×		
输入或改变一经过字	CLR SHF 9876 DEL SHF××××ENT ENT	×	×	
经过字起作用时访问编程器功能	连机：CLR SHF5678DEL SHF ××××ENT NXT （××××为经过字） 离机：CLR SHF 1234 DEL NXT	×	×	

提　示

如果用户由于某种原因忘记了已输入 SR—21PLC 的经过字，将不能使用编程器的编程功能。如果发生这种情况，请与光洋电子（无锡）有限公司销售部联系。以便得到帮助。

强置 I/O 定义号

强置 I/O 定义号，是指用编程器操作的办法强迫某一个输入或输出定义号（或其它内部功能定义号）处于接通或断开状态，它是检查用户逻辑和现场接线的一种有力工具。I/O 强置操作的结果视被强制的定义号而定。与外部设备相连的输入点的定义号是在用户逻辑的一个扫描周期内被强迫 ON 或 OFF。所有其它的 I/O 定义号也可被强迫 ON 或 OFF ，但用户逻辑执行后，优先取用户逻辑决定的状态。

用户可使用 I/O 强制功能来检查输出的现场接线，由于用户逻辑优先于 I/O 强置功能，所以最好在用户程序存储器中无程序时进行这种检查。强迫一个输出点接通，检查被控对象动作是否正常。

下面是强置 I/O 定义号的按键顺序：

.强置某一 I/O 定义号为 ON：

SET，SHF，×××（定义号），ENT

.强置某一 I/O 定义号为 OFF

RST，SHF， ×××（定义号），ENT

警　　告

当强置输入点定义号 ON 或 OFF 时。我们可发现输入点的物理状态是优先的。但如果强置操作发生在用户逻辑执行中、同一扫描中的 I/O 检查之前，则强置状态可能会引起输出点在不应该的时候转为接通或断开。

第三章 编 程 原 理

§3—1 扫描方式

在具体讨论编程原理和定义号使用之前，有必要先介绍一下 CPU 的内部操作原理，在不考虑 CPU 内部结构的情况下，就能解决大多数的应用和程序开发。但是，如果在编程时了解一些 CPU 操作的知识，对某些应用可解决得更好一些。所有 PLC 的基本操作是扫描工作方式。在一个程序中可能要执行几百个逻辑判定，但 CPU 并不能同时作出全部判定。类似于许多微处理器，它是每次执行一个操作。然而，由于内部执行速度很快，可以在全部操作处理完后同时送出结果。

§3—1—1 循环扫描

一、循环扫描是 PLC 的一般处理方法，其扫描顺序为：

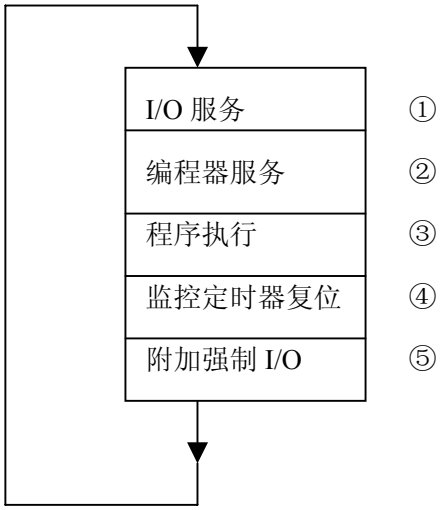


图 3.1 CPU 扫描顺序图

I/O 服务将模块外部输入点的状态读和 CPU 内部的功能存储器，根据程序执行的结果将输出点的功能存储器状态传送到外部模块。

CPU 从首地址开始按次逐个扫描所有的存储器地址，执行用户输入的全部功能，如继电器接点，记忆继电器，计数器，鼓形控制器，移位寄存器等，直到用户程序的结束。在扫描期间线圈功能存储器的状态，取决于输入的用户逻辑程序指令。这此状态可以立即被下面的逻辑功能使用。

如果连接着编程器，还要进行编程器服务和附加的强制 I/O 操作。

之后，CPU 回到扫描的开始，继续进行 I/O 服务和从首地址开始按次扫描逻辑功能。

§3—1—2 快速扫描子程序

一、SR—21/22PLC 定时扫描方式

SR—21/22PLC 的用户程序被执行时，扫描周期约需几十毫秒左右，对于一些要求较高速度输入/输出的场合 SR—21/22PLC 提供了一种快速处理输入/输出的方法，称为快速扫描子程序。其扫描周期固定为 2.5ms。它可以支持一些较快的输入/输出模块，可使用的指令为：

1） 接点处理类指令；

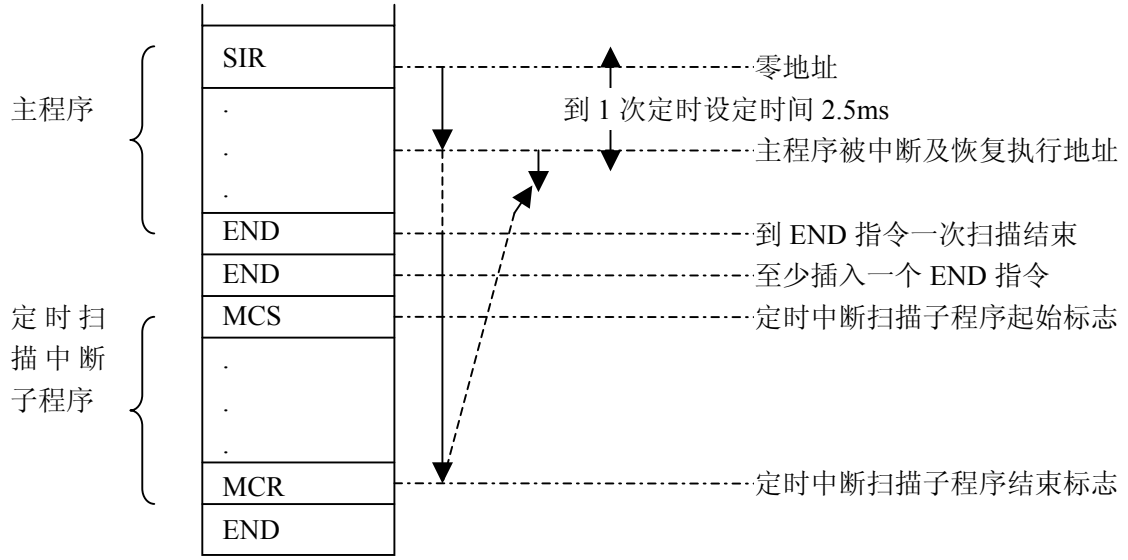
2） 特殊定时器，定时单位为 0.001s（即 1ms）；

3）特殊计数器；

1、快速扫描子程序的构造

为了区别起见，将以往的 PLC 用户程序称为标准程序，而将支持快速处理的用户程序（梯形图）称为快速扫描子程序。

快速扫描子程序的构造是在标准用户程序之后，插入至少一个 END 指令，再以 MCS 指令开始，MCR 指令结束的一段用户程序，参见下图：



SR—21/22 定时中断扫描过程

2、扫描周期：等周期扫描为 2.5ms

3、快速扫描子程序允许的长度

允许使用指令数为 70 条，推荐使用 40 条以下，否则过长的快速扫描子程序将过分影响标准程序的扫描时间。

在快速扫描子程序中，允许使用的特殊定时器或特殊计数器指令总和为 4 条，而且每条特殊定时器或计数器指令相当于 10 条单字指令（STR 等；TMR，CNT 为双字指令）。

4、快速扫描子程序中禁止使用的指令

标准程序中的许多指令可以在快速扫描子程序中使用，也有一些是禁止使用的。如果使用的话，会给出出错信息 E44。在快速扫描子程序中禁止使用的指令如下表所示：

命令	定义号	命令	定义号
STR	600—677	MCR	———
STR NOT	600—677	SET OUT	0—157，700—770
AND	600—677	SET OUT RST	0—157，160—373，376，700—770
AND NOT	600—677	TMR	600—677
OR	600—677	TMR	6.00—6.77
OR NOT	600—677	CNT	100—107
MCS	———	SR	400—577
		F50 等	F50 等 F 类命令

5、快速输入/输出点的定义号范围

定义号 0—157 均可以作为快速输入/输出点使用（即直接输入，输出功能）。而 700—767 不可作为快速输入/输出点处理，但在快速扫描子程序中可以作为普通 I/O 点处理。

对于快速输入/输出点而言，要求输入信号的脉冲宽度≥2.5ms，根据需要可能要使用快速输入模块。

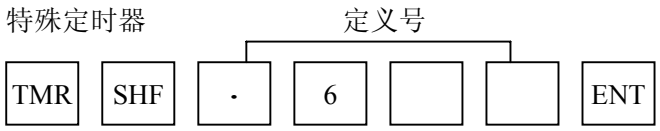
5、 特殊定时器 TMR.600—TMR.673

① 特殊定时器是为了区别其它定时器(定时单位为 0.1s 或 0.01s)而言的。它的定时单位为 1ms，定时范围为 0—9.999s，实际分辨率为 3ms。

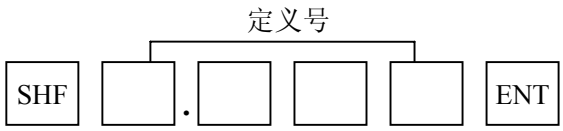
② 特殊定时器可使用的定义号范围为 600—673。在编程时，为了和其它的定时器区分开来，采用.600—.673 来描述其定义号。

③ 对于特殊定时器的定义号.600—.673，如果使用了其中某个定义号，例如.600（即使用了 TMR.600）则下一个定时器（TMR.601）也同时被占用，而且其内容可以被标准程序读出，即 TMR.601 的内容可被读出。

④ TMR.600—.673 命令的预置值可以是常数，也可以是寄存器号 R600—R677，而且所用定义号未被其它 TMR 或 CNT 命令所使用。见下图：

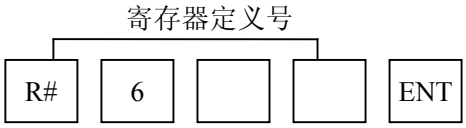


600—673



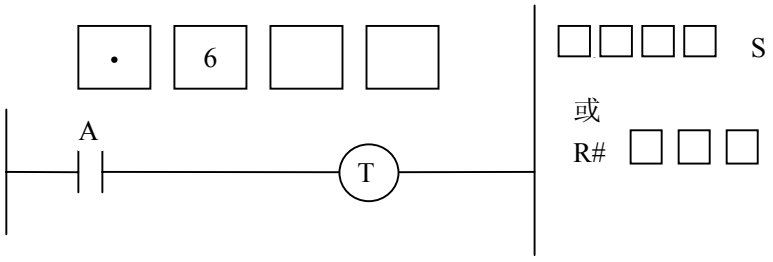
0.001—9．999s

或者



600—677（未被其它 TMR、CNT 所使用）

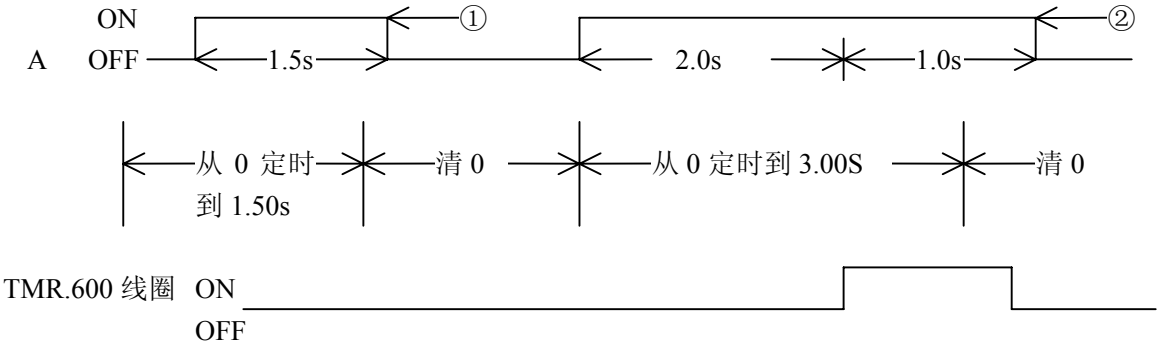
梯形图



⑤ 特殊定时器 TMR.600—.673 的时序

TMR.600—.673 的动作时序和标准定时器 TMR600—677 或 TMR6.00—6.73 的不同，其时序如下图所示。当特殊定时器被清零之瞬间，将其经过值保存到下一个定时器中，直到再一次被清零为止。

动作时序：



在①或②处清 0 之瞬间,将 TMR.600 的经过值保存到 TMR.601 中。

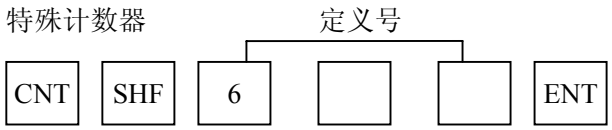
7、特殊计数器 CNT600—673

①在快速扫描子程序中使用的 CNT600—CNT673 和标准程序中的 CNT600—CNT673 不同，称其为特殊计数器，计数范围为 0—9999。

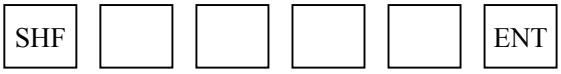
②定义号范围为 600—673。

③对于特殊计数器 CNT600—CNT673，如果使用了其中某个定义号（例如 600）则下一个计数器（CNT601）也被占用，且其内容可以被标准程序读出。

④CNT600—673 指令的预置值可以是常数，也可以是寄存器号 R600—677，而且所用定义号未被其它 TMR、CNT 指令使用，见下图：

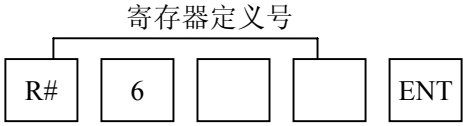


600—673



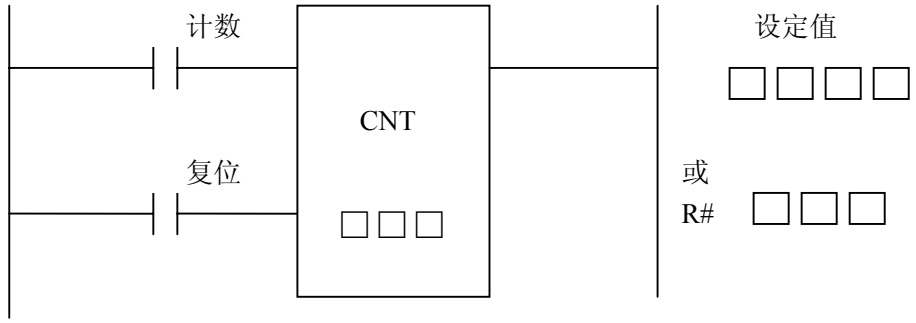
0—9999

或者



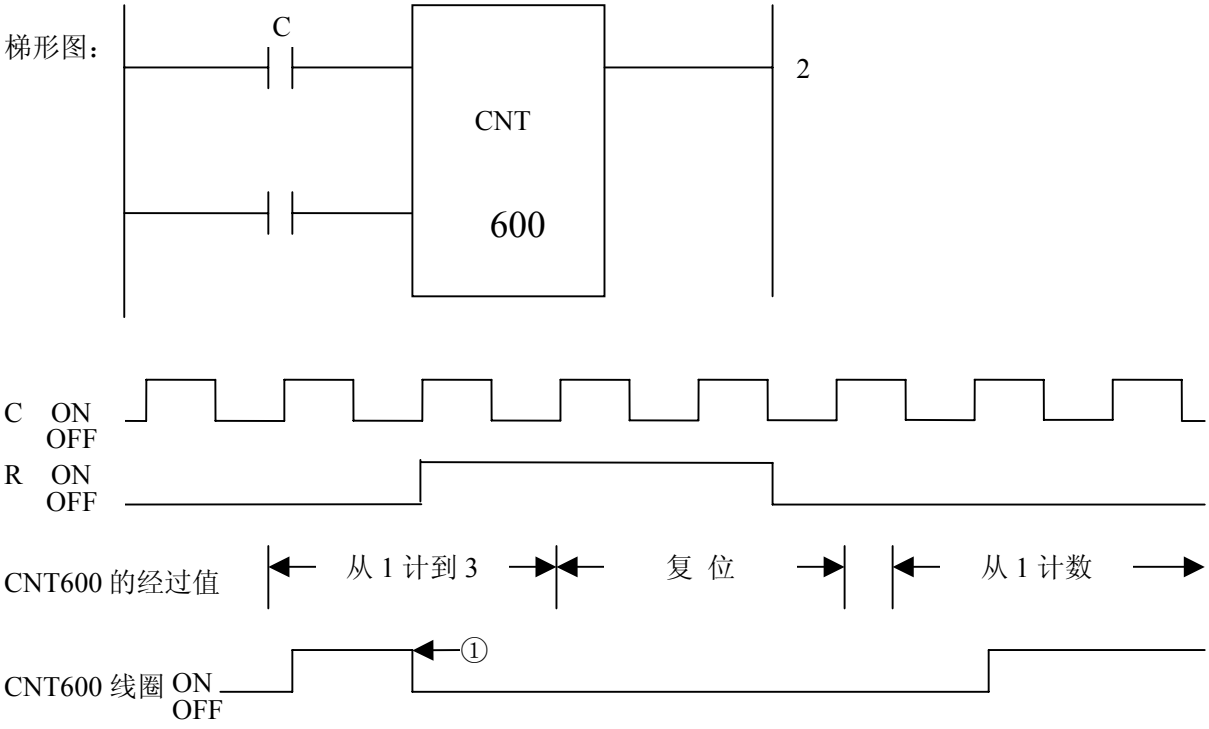
600—677（未被其它 TMR、CNT 所使用）

梯形图



⑤特殊计数器 CNT600—673 的时序

特殊 CNT600—673 的动作时序和标准程序序中的 CNT600—673 不同，其时序如下图所示。当特殊计数器被复位瞬间，将经过值保存到下一个计数中，直到再一次被复位时为止。

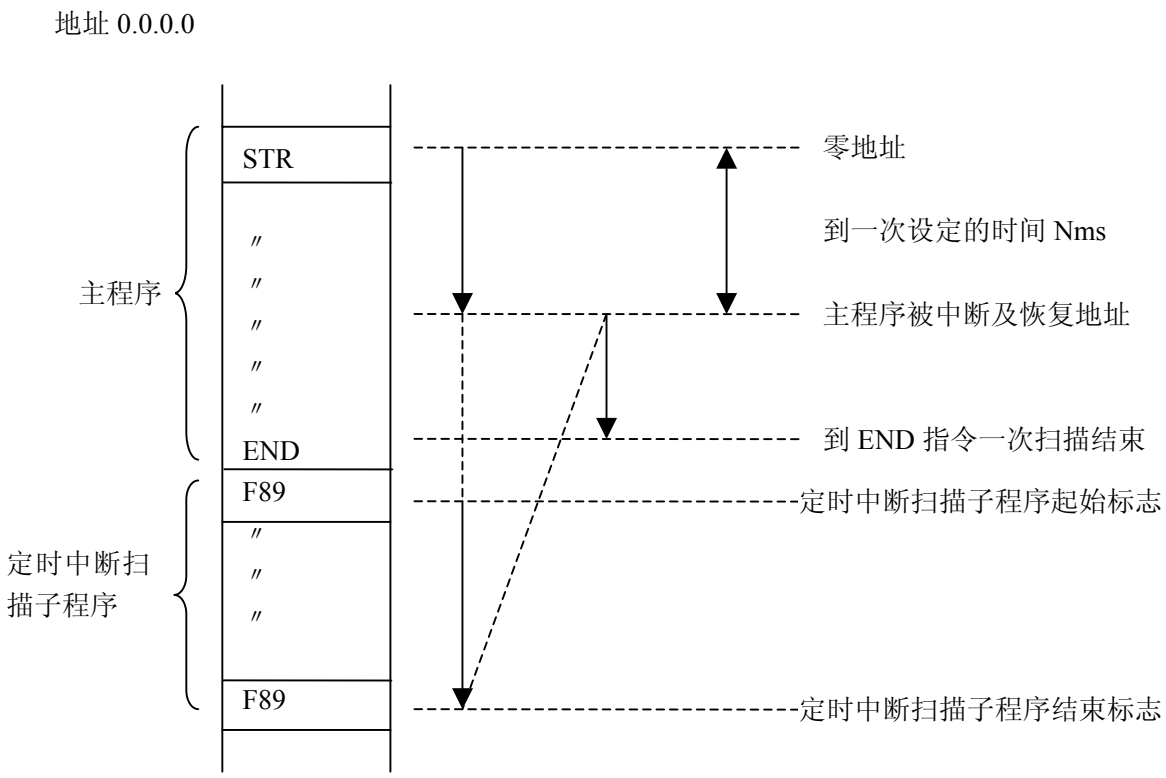


在①处 CNT600 被复位瞬间，将其经过值存入 CNT601 中 。

注：特殊计数器 CNT600—673 仅用在 SR—21/22 的快速扫描子程序中，PLC 的系统程序能自动区别是常用计数器还是特殊计数器。

二、SE—22 定时扫描方式

SE—22 定时扫描方式可以每隔 Nms 对指定的程序进行扫描。在 SE—22PLC 中，定时扫描子程序的执行是在主程序的运行过程即循环扫描过程中，如果特殊功能线圈 SP1002 为 OFF（定时扫描许可），每到定时设定时间，即中断主程序的执行，去执行一遍定时扫描子程序，然后再返回继续原来的主程序的扫描执行。定时扫描方式过程如下图所示：



SE—22PLC 定时扫描中断示意图

1、定时中断扫描子程序开始和结束标志，采用指令功能键 F89，F89 应成对使用。

2、定时中断扫描时间的设定，在系统参数 000 中设定毫秒数。

3、特殊继电器 SP1002 的设定，当 SP1002 为 ON 时，定时中断扫描禁止，当 SP1002 为 OFF 时，定时中断扫描许可。

4、 当 SP1002 为 OFF 时，到达在系统参数 000 中设定的定时中断扫描时间（Nms）时，执行定时中断扫描子程序一次，然后返回到主程序中断处继续执行主程序，再次到达设定的定时中断扫描时间时，再次执行定时中断扫描子程序。

5、 定时中断扫描子程序不宜过长，否则过分影响主程序的执行和响应。

6、 在定时中断扫描子程序中禁止使用的指令。

STR（NOT） 600—677
AND（NOT） 600—677
OR （NOT） 600—677
MCS/MCR
SET OUT
SET OUT RST
TMR/CNT
数据操作指令

§3—1—3 矩阵扫描（仅 SE—22PLC）

（一）、**矩阵扫描输入**

SE—22PLC 具有矩阵扫描输入功能，采用一块 8 点输入模块和一块 8 点输出模块可构成 64 点矩阵输入。

（一）、硬件连接：

1、采用的输入模块及输出模块要能构成回路，建议采用 E—01N 输入模块及 E—10T 输出模块。

2、矩阵输入各点之间需要用二极管隔离。

3、硬件连接。

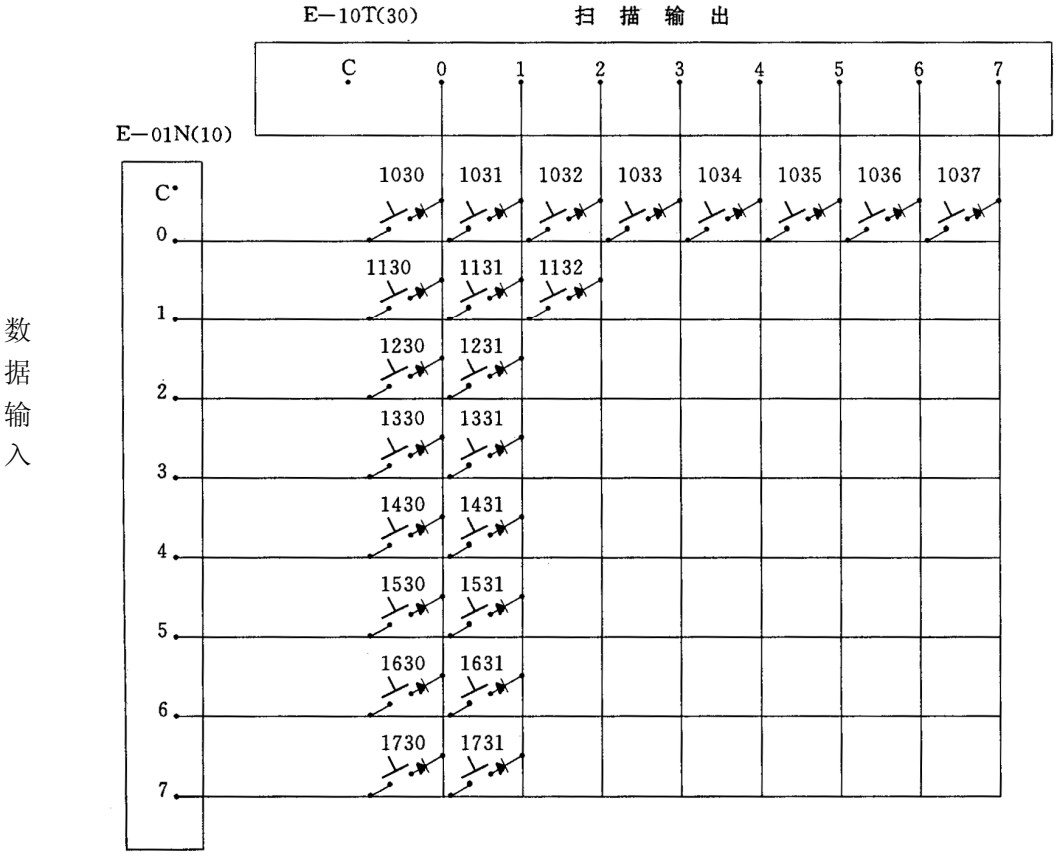


图 3.3 矩阵扫描输入硬件连接图

模块所需的 24V 电源由框架提供，E—01N 和 E—10T 的 24V 电源的 0V 在电源框架内相连接，因此 E—01N 和 E—10T 的公共点已不需要相连。

（二）、矩阵扫描输入的原理

矩阵扫描输入是通过依次轮流接通（扫描）输出模块的 8 个输出点 0、1…7、0，当输出模块 E—10T 的 0 号输出接通时，就把 1030—1730 的 8 个输入点的状态读入矩阵扫描数据区存放起来，……，当输出模块 E—10T 的 7 号输出接通时，就把 1037—1737 的 8 个输入点的状态读入矩阵扫描数据区存放起来，从上可知，当扫描输出模块依次扫描一次则可以读入 64 个输入点的状态，当下一次扫描时，读入新的输入状态，刷新老的输入状态。

（三）矩阵扫描输入的编程方法

假定输入模块 E—01N 安装在 2#槽（定义号 10—17），输出模块 E—10T 安装在 4#槽（定义号 30—37）。

1、用系统参数区 R.001 号设定矩阵扫描输入行选择用线圈开始定义号，即输出模块 E—10T 安装在框架中的定义号，在本例中为 30。

2、用系统参数区 R.002 号设定矩阵扫描输入数据输出用线圈，即输入模块 E—01N 安装在框架中的定义号，在本例中为 10。

3、用系统参数区 R.003 号设定矩阵扫描输入数据的开始定义号，这实际上是设定一个 64 位（bit）的区域用来存放 64 个矩阵扫描输入的数据，设定值一定要用开始定义号。

例如本例中设定 R.003 号为 200，则使用了 200—277 共 64 个内部线圈来存放矩阵输入的数据。

200—207 当输出模块 E—10T 的 0 号（定义号 30）输出接通时，存放 1030—1730 的 8 个输入点的状态。

210—217 当输出模块 E—10T 号字 1 号（定义号 31）输出接通时，存放 1031—1731 的 8 个输入点的状态。

220—227 当输出模块 E—10T 的 2 号（定义号 32）输出接通时，存放 1032—1732 的 8 个输入点的状态。

•
•
•

270—277 当输出模块 E—10T 的 7 号（定义号 37）输出接通时，存放 1037—1737 的 8 个输入点的状态。

建议使用内部线圈比较好，这样就可用内部线圈触点按控制要求控制输出。

4、用系统参数区 R.016 号设定矩阵扫描时间间隔，时间间隔设定为 2.5×Nms

当 R.016 号设定为 1 时，即为 2.5ms，一次扫描总共需 8×2.5ms=20 ms，此为最小扫描时间即矩阵扫描输入重新刷新的最小时间为 20 ms，即按钮（或开关）接通的时间应大于 20 ms，不然要丢失数据。

5、系统参数区寄存器的设定方法

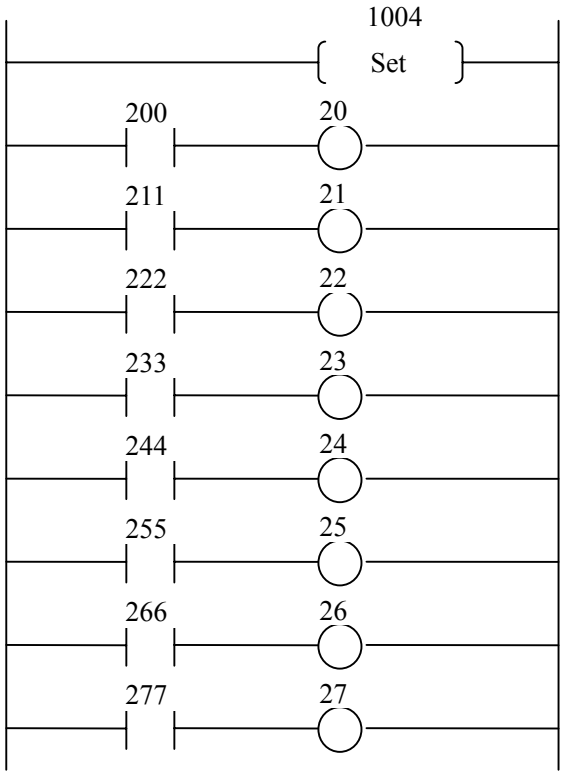
例：设定 R.001 的数值为 30

按[R][.][1][MON]，这时编程器显示 R.001 中的数值

接着按[SHF][3][0][ENT]

6、使特殊继电器 1002 为 ON 时进行矩阵扫描输入。

7、按控制要求绘制梯形图并输入



矩阵扫描输入打开

二、矩阵扫描输出

SE—22 具有矩阵扫描输出功能，采用二块输出模块可构成 64 点矩阵输出。

（一）硬件连接

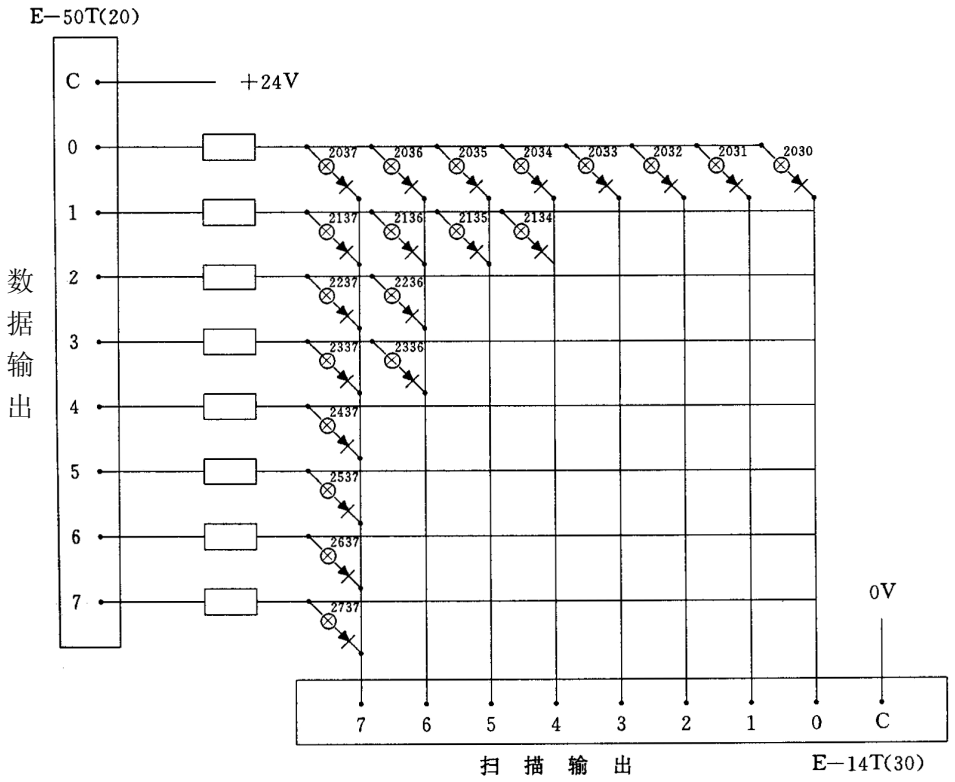


图 3.3 矩阵扫描输出硬件连接图

1、 二输出模块要构成回路，因此需采用一块 8 点源输出模块，一块 8 点汇点输出模块。

2、 因扫描的刷新周期为 2.5ms×8ms,输出动作频繁,因此需采用晶体管输出模块。

3、 扫描输出模块可能同时接通 8 个灯，因此该模块要选用负载电流大的 8 点汇点输出模块，建议：数据输出采用 E—50T 源输出模块，扫描输出采用 E—14T 汇点输出模块。

4、 矩阵输出各点应串联二极管隔离，防止相互影响。

5、 在某些场合要串联限流电阻，防止灯烧坏。

（二）矩阵扫描输出的原理

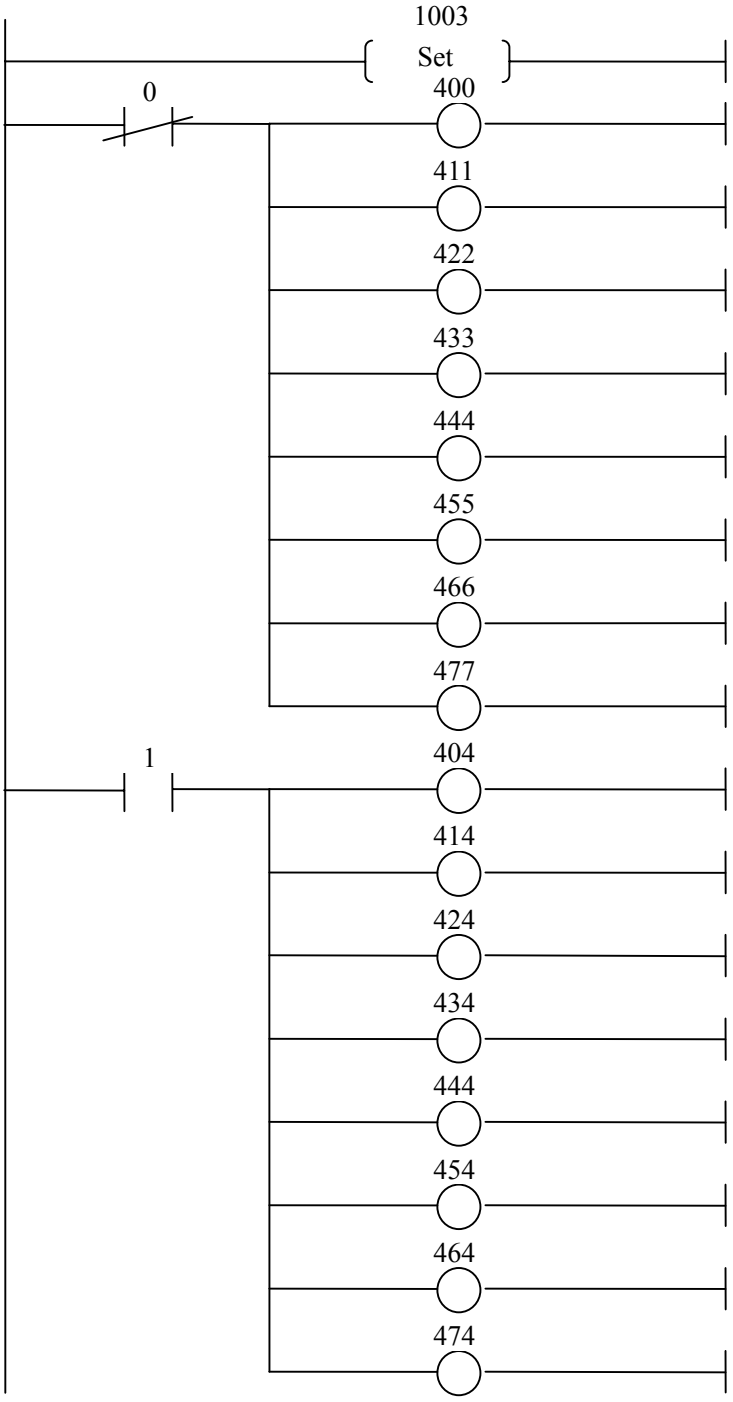
根据设定的矩阵输出数据区，例如设定系统参数区 R.013 为 400，则设定了矩阵扫描输出数据区为 400—407，410—417，420—427，430—437，440—447，450—457，460—467，470—477 共 64 个内部线圈，当扫描输出开始扫描时，E—14T 输出模块 0 号输出（定义号 30）接通，同时把 400—407 的状态送到数据输出模块 E—50T，400—407 中接通的内部线圈（由程序决定）使数据输出模块 E—50T 相对应的输出点接通，例如 400，404 点接通则 E—50T 输出模块的 0 号（定义号 20）和 4 号（定义号 24）2 个输出点接通，这时矩阵输出指示灯 2030，2430 接通，经过 2.5ms 以后，扫描输出模块 1 号（定义号 31）接通，这时把 410—417 的状态送到数据输出模块 E—50T，如果 411 和 414 点接通，则 E—50T 输出模块的 1 号（定义号 21）和 4 号（定义号 24）2 个输出点接通，则相应的矩阵扫描输出指示灯 2131 和 2431 接通……依次循环，在使用矩阵扫描输出时，矩阵扫描输出

时间间隔应设定为最快的 2.5ms，这时利用人的眼睛反应迟钝矩阵输出指示灯不闪烁，如果设定为 5ms 则矩阵输出指示灯闪烁，如果循环扫描时间长则矩阵输出指示灯轮流接通。

（三）矩阵扫描输出的编程方法

假定输出模块 E—50T 安装在 3#槽（定义号 20—27），输出模块 E—14T 安装在 4#槽（定义号 30—37）。

1、 设定矩阵扫描输出行选择开始定义号 R.011 为 30
2、 设定矩阵扫描输出数据的开始定义号 R.012 为 20
3、 设定矩阵扫描输出数据的开始定义号 R.013 为 400
4、 设定矩阵扫描输出时间间隔 R.017 为 1
5、 使特殊继电器 1003 为 ON，进行矩阵扫描输出。
6、 根据输出要求编制程序



§3—1—4 级式编程语言(仅 SE—22PLC)

SE—22PLC 在原有的梯形图指令基础上增加了级式语言，在编程序时可使用梯形图语言，也可使用级式语言。

级式语言的级就是把动作按工序一步一步进行分解并编程，然后按工序执行顺序和条件连接起来完成控制目的，在级式编程中，原来互相联锁的动作可以编写在不同时动作的级中，各个动作的联锁可以由级的状态自动控制，因此它可以使编程变得简单。

这种以级为单位的程序（块），根据级的状态为 ON 或是 OFF，而执行或者不执行该级内的程序，这种级的执行和流程可以由编程者进行控制，包括并行运行，跳转，根据不同条件分支，合流等，因为只有那些状态为 ON 的级内的程序才会执行，跳过 OFF 的级，这样它可以使程序运行速度加快。

状态为 ON 的级可以由程序中的 JMP，NJMP，SET，RST 以及 ISG 指令来设定或控制，包括：

（1）由 ISG 指令指定的初始级，在 PLC 投入运行开始时状态自动为 ON。

（2）由 JMP，NJMP 指令执行后进入的目的级状态为 ON。

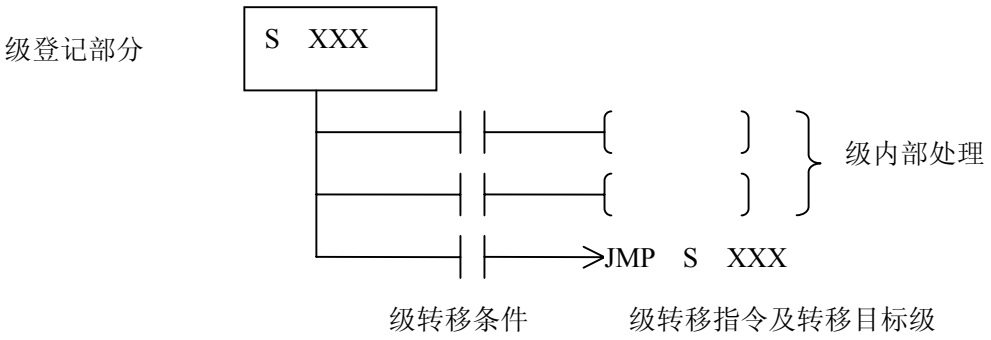
（3）由 SET 指令执行后，被置位的指定级状态为 ON。

状态为 ON 的级可以同时有一个至多个，在级号允许范围内没有限制，由于程序上的先后顺序，PLC 依据循环扫描的原理来从前往后扫描所有状态为 ON 的级。

级式语言编程方法是在梯形图指令系统的基础上增加了一些级的定义和控制指令，级内的程序由梯形图指令构成，但有些梯形图指令在级式语言编程中的作用有些变化。

具体请参阅原华光电子工业有限公司《级式语言编程指导》。

级的构成：



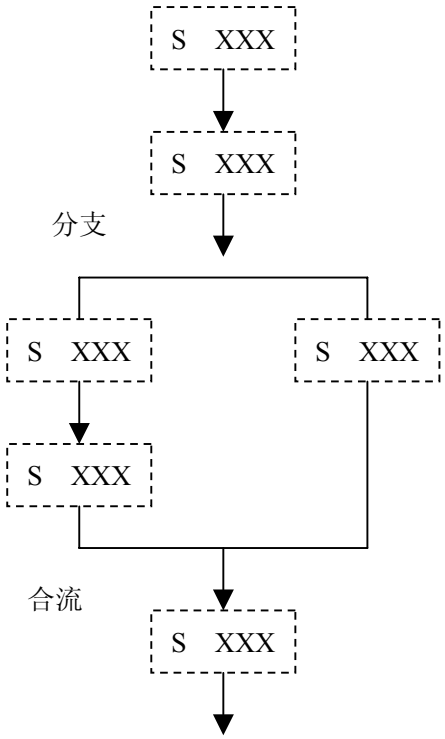
通常，级登记部分是必需的，它表示一个级的开始，初始级用 ISG 指令指定，其它级用 SG 指令指定。

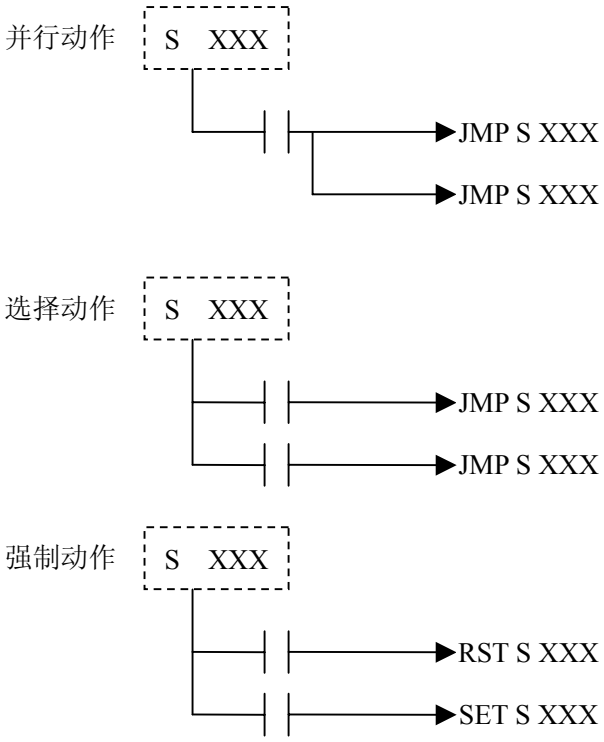
级内部处理是根据控制要求的处理程序。使用梯形图指令。

级转移条件和级转移目标级，是本级内容完成后的处理目标。

那么，整个程序就可以由这样一些级联结构成。

级的流向：


S XXX
S XXX
分支
S XXX
S XXX
S XXX
合流
S XXX


并行动作
S XXX
JMP S XXX
JMP S XXX
选择动作
S XXX
JMP S XXX
JMP S XXX
强制动作
S XXX
RST S XXX
SET S XXX

§3—2 功能存储器及定义号

功能存储器是 PLC 存储各种功能（例如输入，输出，内部线圈，定时器，计数器，移位寄存器，特殊线圈等）的通断状态和存储数据（数据寄存器、T/C 经过值寄存器等）的存储器。定义号“是每个功能存储器的编号，每种功能存储器都规定了不同的编号和范围，则每种功能存储器都有自己的定义号。定义号采用 8 进制。表 1 为功能存储器及定义号一览表。

定义号是 PLC 编程中非常重要的部分，无论何种 PLC，编程都与定义号有关。使用何种功能一定要使用该功能的定义号，一定不能搞错。

表 1、功能存储器及定义号一览表

功　能	简称	SR—21/22		SE—22		备　注
		点数	定义号	点数	定义号	
输入/输出	I/O	184	000—177 700—767	184	000—177 700—767	
级	SG			128	1400—1577	SE—22PLC 根据系统参数 004 的千位状态可以停电记忆
内部线圈	IR	124	200—337 340—373*	364	200—337 340—373* 1020—1377*	*有停电记忆功能 SE—22PLC，340—373 根据系统参数 004 的个位状态可以停电记忆，1020—1377 根据系统参数 004 的百位状态可以停电记忆。
特殊线圈	SP	12	374—377 770—777	28	374—377 770—777 1000—1017	
移位寄存器	SR	128	400—577*	128	400—577*	*SR—21 有停电记忆功能 SE—22 根据系统参数 004 的十位状态可以停电记忆，可以作内部线圈使用。
定时器/计数器	T/C	64	600—677*	64	600—677*	作定时器使用无停电保持 作计数器使用或未使用定义号停电保持
输入/输出寄存器	R	23	000—017 070—076	23	000—017 070—076	
级寄存器	R			16	140—157	
内部线圈寄存器	R	15.5	020—033 034—037*	45.5	020—033 034—037* 102—137*	*有停电记忆功能
特殊线圈寄存器	R	1.5	037 077	3.5	037 077 100—101	
移位寄存器	R	16	040—057*	16	040—057*	*有停电记忆功能
定时器/计数器寄存器	R	8	060—067*	8	060—067*	*计数器有停电记忆功能
数据寄存器(8 位)	R	128	400—577*	336	160—577* 700—777*	*有停电记忆功能
T/C 经过值寄存器(16 位)	R	64	600—677*	64	600—677*	*计数器有停电记忆功能
16 位数据寄存器	P			512	000—777*	*有停电记忆功能
系统参数区	R#			32	000—037	R# 000—037 为 FLASHROM 区

§3—2—1 输入/输出定义号分配

SR—21PLC 系列的 I/O 模块可按您的需要任意组合安装，其定义号不是取决于 I/O 模块，而是取决于安装 I/O 模块框架的槽号，框架的每个槽都有固定的定义号，如图 3.4 所示，装 CPU 的槽为 0 号槽。靠近 CPU 的槽为 1 号槽，其定义号为 0~7，往左为 2 号槽，其定义号为 10~17。……依次类推，如果某槽中安装 16 点模块，这 16 点模块的前 8 点定义号为该槽定义号，后 8 号定义号为该槽定义号+100。因此模块一旦安装好后，该模块的定义号也就确定了。例如在 1 号槽中安装了 E——05NT16 点输入模块，该模块前 8 点输入的定义号为 0—7，后 8 点输入的定义号为 100—107。注意：这时 9 号槽就不能安装模块了，因 9 号槽的定义号 100—107 已经被 0 号槽中的 16 点输入模块中的后 8 点定义号占用了。

每个 I/O 点的定义号还得根据模块上的回路号而定，回路号就是 I/O 模块与外部设备相连的接线端子号。所有的 I/O 定义号见 §1—5—1 系统配置表。

9	8	7	6	5	4	3	2	1	0	
110 | 107	70 | 77 170 | 177	60 | 67 160 | 167	50 | 57 150 | 157	40 | 47 140 | 147	30 | 37 130 | 137	20 | 27 120 | 127	10 | 17 110 | 117	10 | 17 100 | 107	C P U	电 源

图 3.4 I/O 定义号举例

§3—2—2 内部线圈

一、内部线圈

内部线圈是在用户程序中作为控制中间逻辑用的控制继电器（相当于中间继电器），内部的状态不能直接输出给输出模块。SR—21 系列 PLC 中的内部线圈 340—373 具有停电记忆功能，则当 PLC 处于运行状态，当 AC 电源断电时，该线圈能记忆住停电时的状态（ON 或 OFF）。在 CPU 模块上有一个开关（SW1），当它在 ON 位置时，340—373 内部线圈有停电记忆功能，在 OFF 位置则这些内部线圈无停电记忆功能。

SR—21 系列 PLC，其移位寄存器定义号 400—577 也可作为内部线圈使用，并具有停电记忆功能。

SE— 22PLC 中的内部线圈 340—373 根据系统参数 004 的个位状态决定是否停电记忆，内部线圈 1020—1377 根据系统参数 004 的百位状态决定是否停电记忆。

二、特殊继电器

CPU 的特殊继电器是用途已被定义的内部继电器，在程序中通常用来表示某种状态或者设定功能。

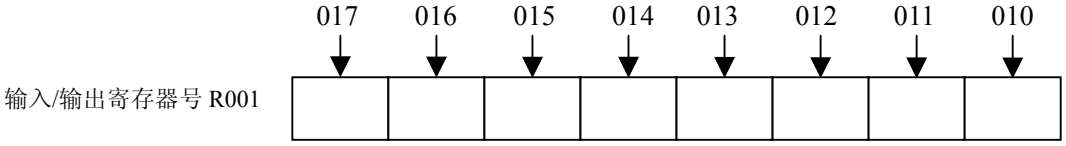
表 2 特殊继电器一览表

特殊继电器号	SE—22	SR—21/22
374	初期复位继电器，运行开始时的第 1 个扫描周期 ON	
375	0.1 秒时钟,0.05ON/0.05OFF	
376	PAUSE 暂停输出线圈,ON 时外部输出全部 OFF(SETOUT,SET 指令除外)	
377	电池异常,电池电压低时为 ON	
770	0.01 秒时钟,0.005ON/0.005OFF(仅 SE—22)	
771	外部诊断	
772	＞标志（ACC 大）	
773	＝标志（ACC 一致）	
774	＜标志（ACC 小）	
775	C 标志（进位或借位）	
776	Z 标志（零）	
777	E 标志（溢出）	
1000	ON 为固定协议通讯中	—
1001	ON 为通讯错误	—
1002	ON 定时扫描禁止，OFF 定时扫描允许	—
1003	ON 矩阵扫描输出开放，OFF 正常扫描输出	—
1004	ON 矩阵扫描输入开放，OFF 正常扫描输出	—
1005	ON 环行送信缓冲区满，OFF 环行送信缓冲区未满	—
1006	ON 环行受信缓冲区满，OFF 环行送信缓冲区未满	—
1007	1 秒时钟，0.5 秒 ON/0.5 秒 OFF	—
1010	1 分钟时，30 秒 ON/30 秒 OFF	—
1011	扫描时钟，1 扫描 ON/1 扫描 OFF	—
1012	RUN 方式 ON	—
1013—1017	备用	—

§3—2—3 数据寄存器和特殊寄存器

数据寄存器用于存储数据，数据可用数据操作指令写入数据寄存器或从数据寄存器中读出，SR—21 系列的数据寄存器是 8 位数据寄存器（一个字节）可存放 2 位 BCD 数据，它们的定义号为 400—577。

输入、输出，内部线圈，移位寄存器、定时器、计数器等主要是进行逻辑运算，其定义号是单个位的（bit），实际上它们也是按顺序以 8 个定义号（8 bit）为一组存放在相对应的寄存器中，其对应数据寄存器的定义号，是在原定义号中去掉最后一位，在前面加个 0，例如：8 个输入，其定义号为 010—017，实际上这 8 个输入的状态作为一组存放在定义号为 001 的输入/输出寄存器中，其对应关系为：



如果某一个输入接通，则对应定义号的那一位置 1。这样，如果进行逻辑运算（逻辑指令）则从其存储的寄存器中取出其定义号的单个位（bit）进行逻辑运算。如果进行数据操作（数据操作指令），则把这 8 个输入作为一个整体组成 2 位 BCD 码，采用输入寄存器定义号（而不是单个位定义号）则可与数据寄存器一样进行数据操作。

SE—22PLC 除了 8 位数据寄存器外，还有 16 位数据寄存器。

T/C 经过值寄存器存放对应的定时器/计数器的经过值，为 16 位（存放 4 位 BCD 码）寄存器。

SR—21/SE—22 系列 PLC 的某些数据寄存器已定义有特殊功能称特殊功能寄存器，如下表所示：

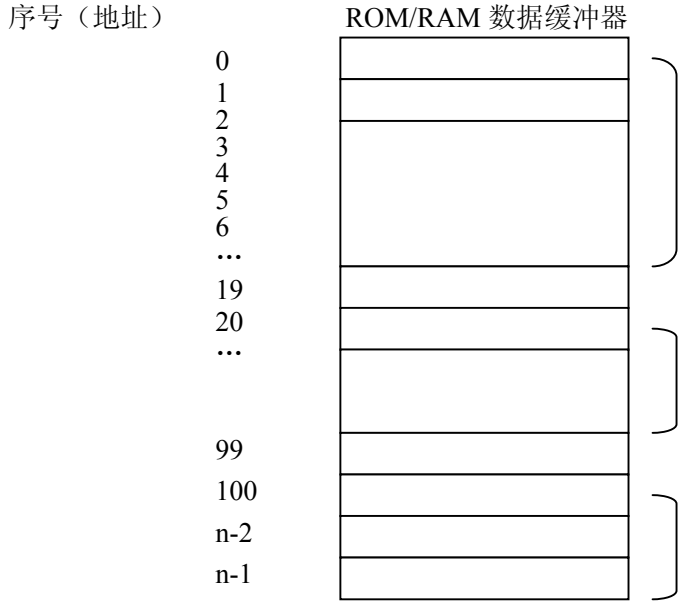
表 3 特殊寄存器一览表

寄存器号	用途	SR—21/22	SE—22
R0564 R0565 R0566 R0567 R0570 R0571 R0572 R0573	拨盘接口模块 E—01D 的 4 位 BCD×4 的设定存储用寄存器	×	×
R0574 R0575	外部诊断寄存器	×	×
R0576 R0577	辅助 ACC	×	×
R0740	最短扫描时间		×
R0741	最长扫描时间		×
R0742	现在扫描时间		×
R0743	语法检查错误号（ASCII 型 BCD）		×
R0744	程序出错的首地址（ASCII 型 BCD 下位）		×
R0745	程序出错的首地址（ASCII 型 BCD 上位）		×
R0746	环行送信缓部区字节数		×
R0747	环行受信缓冲区字节数		×
R0750—R0777	备用		×

§3—2—4 数据缓冲器（仅 SP—21/22）

SR—21 系列 PLC 除了 128 个数据寄存器外，还有功能和数据寄存器一样的数据区称为数据缓冲器，它是把 RAM 或 ROM 的用户存储器的空余部分作为数据缓冲区，因此它有两种类型，一种为可写/可读的 RAM 数据缓冲器，另一种为只读的 ROM 数据缓冲器，其大小随用户程序长度不同而改变，最大长度为 5000 字，每个字对应 4 位 BCD 数（16 位），其长度存放在 0 号数据缓冲器中。

数据缓冲器的构造如下图所示。



数据缓冲器的序号（地址）从 0 开始至 n—1（按十进制编号 1≤n≤5000），每个序号内存放 4 位 BCD 数。有关数据缓冲器的操作均是按序号，即以 4 位 BCD 数为对象进行的，这一点和 PLC 原有的数据寄存器操作不同。

① 序号 0—19 为只读区域，它主要是用来表示系统的状态，目前被使用的有：
0 号：存放数据缓冲器的长度，即（n—1），n 为数据缓冲器的长度。
1 号：存放系统软件版本号。
2 号：存放系统存储器配置代码。

② 序号 20—99 具有特殊意义，在使用时应注意，但目前还未规定其特殊含义。
这块区域对 RAM 而言可读写；对 ROM 而言为只读。

③ 序号 100—n—1 为用户可随意使用的数据区域。若为 RAM 数据缓冲器时，可读可写；否则为只读。

④ 在编程状态下，对 RAM 数据缓冲器可以进行全清、显示、修改操作；对 ROM 数据缓冲器只可进行显示操作。

⑤ 在运行状态下，对 RAM 数据缓冲器可以进行显示、修改操作；对 ROM 数据缓冲器只可进行显示操作。

⑥ 在 LOAD 状态下，可以对 RAM 数据缓冲器进行录音、装入和校核；对 ROM 数据缓冲器可以进行录音、校验操作。

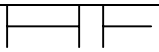
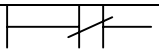
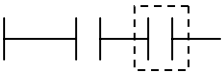
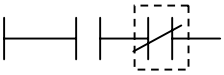
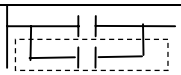
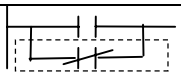
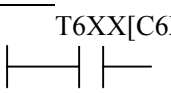
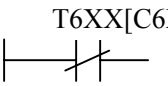
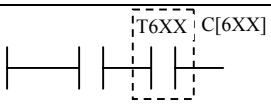
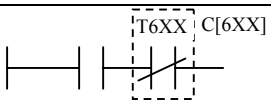
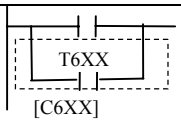
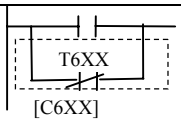
§3—2—5 系统参数区（仅 SE—22）

系统参数区是用来指定 SE—22CPU 的功能和使用范围的区域，仅在 RUN 方式时有效。

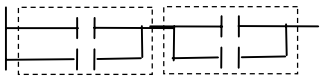
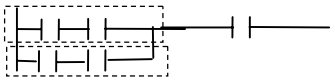
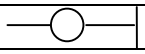
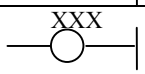
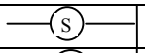
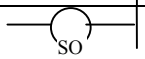
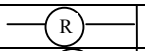
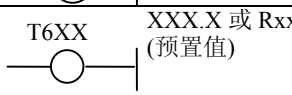
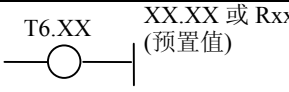
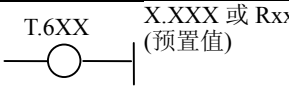
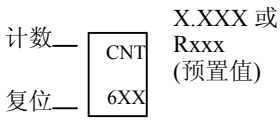
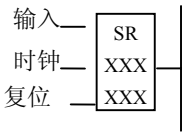
000	N ms 定时扫描功能的设定	
001	矩阵扫描输入行选择用线圈开始定义号	
002	矩阵扫描输入数据输出用线圈开始定义号	
003	矩阵扫描输入数据区的开始定义号	
004	停电保持的有效/无效指定	参见功能存储器表的说明
005	备用	
006	程序转录端口的指定	00 通过磁带端口转录 01—99 通过 RS232C 端口转录
007 010	备用	
011	矩阵扫描输出行选择开始定义号（8 点）	
012	矩阵扫描输出数据开始定义号（8 位）	
013	矩阵扫描输出数据区的开始定义号（8 个）	
014	固定协议时的通信终了码	
015	固定协议时的通信终了码	
016	矩阵扫描输入时间间隔（2.5×Nms）指定	扫描周期为 8×2.5×Nms
017	矩阵扫描输出时间间隔（2.5×Nms）指定	扫描周期为 8×2.5×Nms
020	通讯局的指定	1—90
021	通讯波特率的指定	01：300bps 02：600 bps 03：1200 bps 04：2400 bps 05：4800 bps 06：9600 bps 07：19200 bps
022	奇偶校验停止位的指定	下位：0=无奇偶校验 1=奇校验 2=偶校验 上位：1=1 位停止位 2=2 位停止位
023	数据长度，HEX/ASCII 的设定	下位：7=7 位数据长度 8=8 位数据长度 上位：0=HEX 1=ASCII
024	应答延迟，帧延时的设定	下位：0= 0ms 应答延迟 1=10ms 应答延迟 2=100ms 应答延迟 3=500ms 应答延迟 上位：0=0ms 帧延时 1=10ms 帧延时
025—037	备用	

§3—3 指令一览

一、顺序指令

分类	指令名称	指　令	符　　号	语数
接点指令	逻辑运算开始 NO 接点。压入堆栈	STR		1
	逻辑运算开始 NC 接点	STR NOT		1
	逻辑与运算 NO 接点	AND		1
	逻辑与运算 NC 接点	AND NOT		1
	逻辑或运算 NO 接点	OR		1
	逻辑或运算 NC 接点	OR NOT		1
T/C 接点指令	逻辑运算开始 NO 接点	STR TMR		2
		STR CNT		2
	逻辑运算开始 NC 接点	STR NOT TMR		2
		STR NOT CNT		2
	逻辑与运算 NO 接点	AND TMR		2
		AND CNT		2
	逻辑与运算 NC 接点	AND NOT TMR		2
		AND NOT CNT		2
	逻辑与运算 NO 接点	OR TMR		2
		OR CNT		2
	逻辑或运算 NC 接点	OR NOT TMR		2
		OR NOT CNT		2

续前

分类	指令名称	指 令	符 号	语数
逻辑组连接	逻辑组串联	AND STR		
	逻辑组并联	OR STR		
母线指令	主控开始	MCS		
	主控复位	MCR		
输出指令	线圈 ON 动作	OUT		1
	移位寄存器作内部线圈使用时 ON 动作	OUT 400-577		1
	线圈 ON 保持动作	SET		1
	输出禁止线圈 376 接通时不受其影响	SET OUT		1
	线圈 OFF 保持动作	RST		1
	单脉冲线圈 ON 动作	SET OUT RST		1
定时器指令	0.1 秒定时器	TMR600-673		2
	0.01 秒定时器	TMR 6.00-6.73		2
	0.001 秒定时器(仅 SR-21/22)	TMR 600-.673		2
	计数器	CNT CNT XXX *1		2
	移位寄存器	SR		2

*1：在 SR—21、SE—22PLC 中

CNT100、CNT102、CNT104、CNT106 为高速计数模块设定值输入专用指令。

CNT101、CNT103、CNT105、CNT107 为高速计数模块比较值输入专用指令。

二、数据处理指令

分类	指令名称	指　令	符　　号	语数
数据读入指令	16 位读入指令	D．STR （F50）	D．STR Rxxx 或 D．STR xxxx	2
	8 位读入指令	D．STR1 （F51）	D．STR1 Rxxx	2
	上 4 位读入指令	D．STR2 （F52）	D．STR2 Rxxx	2
	下 4 位读入指令	D．STR3 （F53）	D．STR3 Rxxx	2
	16 点模块读入指令	D．STR5 （F55）	D．STR5 Rxxx	2
	任意位 RAM 数据缓冲器内容读入指令 （仅 SR—21/22）	F56 nnnn	F56　nnnn	2
	任意位 RAM 数据缓冲器内容间址读指令 （仅 SR—21/22）	F56 R×××	F56　Rxxx	2
	RAM 数据缓冲器内容送到数据寄存器块传送指令 （仅 SR—21/22）	F57 R×××	F50 xx　*1 F60　R576 F50 xxxx　*2 F57 Rxxx　*3	2
	任意位 ROM 数据缓冲器内容读入指令 （仅 SR—21/22）	F58 nnnn	F58　nnnn	2
	任意位 ROM 数据缓冲器内容间接读指令 （仅 SR—21/22）	F58 R×××	F58　Rxxx	2
	ROM 数据缓冲器内容送到数据寄存器块传送指令 （仅 SR—21/22）	F59 R×××	F50 xx　*1 F60　R576 F50 xxxx　*2 F59 Rxxx　*3	2

注：*1 给定传送字数 N　0<N<64.

*2 给定 RAM 数据缓冲器的开始编号 NO　0≤NO≤RAM 缓冲器的最大序号。

*3 Rxxx 为数据寄存器 R400~R576。

续前表

分类	指令名称	指　令	符　号	语数
数据写出指令	16 位写出指令	D．OUT （F60）	D．OUT Rxxx	2
	8 位写出指令	D．OUT1 （F61）	D．OUT1 Rxxx	2
	上 4 位写出指令	D．OUT2 （F62）	D．OUT2 Rxxx	2
	下 4 位写出指令	D．OUT3 （F63）	D．OUT3 Rxxx	2
	16 点模块写出指令	D．OUT5 （F65）	D．OUT5 Rxxx	2
	将累加器内容写到 RAM 数据缓冲区任意写出指令 （仅 SR—21/22）	F66 nnnn	F66　nnnn	2
	将累加器内容写到 RAM 数据缓冲区间址写指令 （仅 SR—21/22）	F66 R×××	F66　Rxxx	2
算术运算指令	16 位比较指令	CMPR ＞＝＜ （F70）	CMPR Rxxx 或 CMPR xxxx 4 位 BCD	2
	4 位 BCD 加法指令	+ （F71）	+　Rxxx 或 +　xxxx 4 位 BCD	2
	4 位 BCD 减法指令	— （F72）	—　Rxxx 或 —　xxxx 4 位 BCD	2

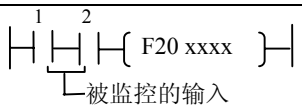
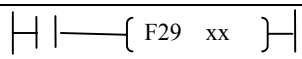
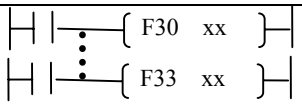
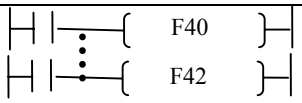
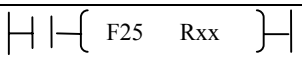
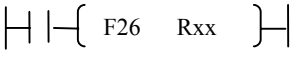
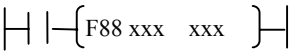
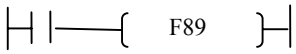
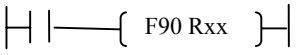
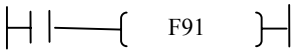
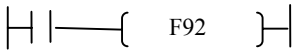
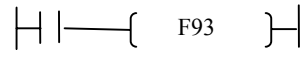
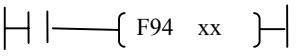
续前表

分类	指令名称	指　令	符　　号	语数
算术运算	4 位 BCD 乘法指令	× （F73）	× Rxxx 或 × xxxx 4 位 BCD	2
	4 位 BCD 除法指令	÷ （F74）	÷ Rxxx 或 ÷ xxxx 4 位 BCD	2
逻辑处理	16 位“与”	D．AND （F75）	D．AND Rxxx 或 D．AND xxxx	2
	16 位“或”	D．OR （F76）	D．OR Rxxx 或 D．OR xxxx	2
	HEX→ASCII （仅 SE—22）	F78	F78	1
	ASCII→HEX （仅 SE—22）	F79	F79	1
	数据寄存器内容+1 指令 （仅 SR—21/22）	F77 R×××	F77 Rxxx	2
	数据寄存器内容—1 指令 （仅 SR—21/22）	F78 R×××	F78 Rxxx	2

续前表

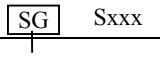
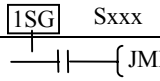
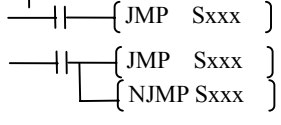
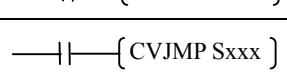
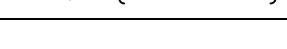
分类	指令名称	指　令	符　　号	语数
逻辑变化指令	右移指令	SFTR （F80）	├┤├{ F80 xx }├┤	2
	左移指令	SFTL （F81）	├┤├{ F81 xx }├┤	2
	译码指令	DECO （F82）	├┤├{ F82 }├┤	1
	编码指令	ENCO （F83）	├┤├{ F83 }├┤	1
	取反指令	INV （F84）	├┤├{ INV }├┤	1
	BCD→BIN 变换指令	BIN （F85）	├┤├{ BIN }├┤	1
	BIN→BCD 变换指令	BCD （F86）	├┤├{ BCD }├┤	1
	SEG 变换 （仅 SE—22）	SEG （F87）	├┤├{ F87 }├┤	1

续前表

分类	指令名称	指　令	符　　号	语数
特殊指令	外部故障诊断指令	FALT（F20）		1
智能模块控制指令	智能模块读取数据存放领域的指定	F29		2
	向智能模块传送数据	F30~F33		2
	从智能模块读取数据	F40~F42		1
	C—02DS 的状态指定命令	F25		2
其它命令（仅 SE—22）	寄存器间址写指令	F26		2
	寄存器间址读指令	F27		2
	指定范围 I/O 传送	IORF（F88）		3
	定时扫描程序开始、结束	F89		1
	固定协议通讯开始	RXTX（F90）		2
	环形受信缓冲区的 1 字节读出	READ（F91）		1
	环形送信缓冲区的 1 字节写入	WRITE（92）		1
	注释指令	CMMT（93）		1
	轴定位调用指令	F94		2

注：F25 指令的使用请参阅 C—02DS 用户手册。F29—F42 指令的使用请参阅《E—02PP 轴定位模块使用说明书》

三、级式指令（仅 SE—22）

分类	指令名称	指　令	符　　号
级式指令	级登记指令	SG（F10）	
	初始级登记指令	ISG（F11）	
	级转移指令	JMP/NJMP（F12/F13）	
	级合流登记指令	CV（F14）	
	级合流转移指令	CVJMP（F15）	

级式语言请参阅《级式语言编程手册》

第四章 编　程

§4—1 顺序指令

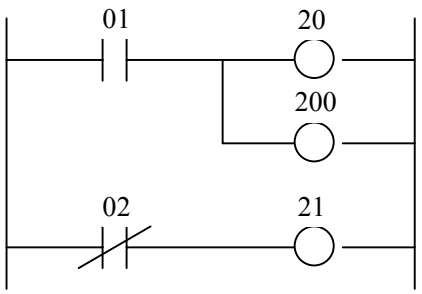
§4—1—1 基本逻辑指令

一、STR、STR NOT、OUT

可使用的触点定义号：I/O、S、IR、SP、SR。可使用的输出定义号 Q、IR、SP、SR。

STR 逻辑行开始的常开接点，STR NOT 逻辑行开始的常闭接点。OUT 输出指令。

指　令		注　释
指令符	定义号	
STR	01	01 为常开接点
OUT	20	20 为输出线圈
OUT	200	200 为内部线圈
STR NOT	02	02 为常闭接点
OUT	21	21 为输出线圈



另外，指令 STR TMR、STR NOT TMR、STR CNT、STR NOT CNT 的使用方法与 STR、STR NOT 相同，只是定义号不同，为定时器/计数器定义号。

注：键入梯形图程序时编程器方式开关必需在编程（PRG）位置。

1、 清除所有存储器内容，按键顺序是：CLR、SHF、3、4、8、DEL、NXT，在完成上述操作后，用户存储器内容全部清零，程序首地址为 0、0、0、0。

2、 键入程序必需从零地址开始键入。

3、 必需根据梯形图从上到下，从左到右的顺序一个逻辑行，一个逻辑行键入，中间不能有空地址。

4、 在键入数字（定义号等）之前必需按“SHF”键（第二功能），指令结束时一定要按“ENT”键。例如上例梯形图，键入程序为：

STR　　SHF　1　ENT

OUT　　SHF　20　ENT

OUT　　SHF　200　ENT

STR　NOT　SHF　2　ENT

OUT　　SHF　21　ENT

在数字键之前要按 SHF 键，在指令结束时按 ENT 键，这是规定，在以后讲述编程时，为了简明，省略了“SHF”键和“ENT”键，如上例所示，但在键入程序时不要忘记键入这两个键。

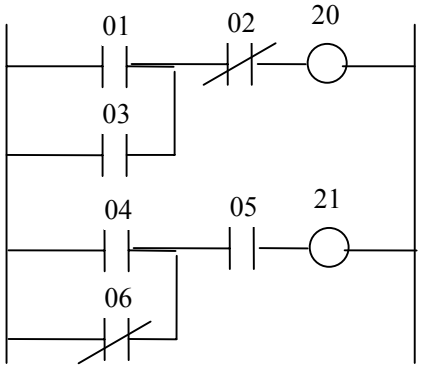
二、AND、AND NOT、OR、OR NOT

可使用的触点定义号：I/O、S、IR、SP、SR。

AND、AND NOT 表示在原先接点上串联一个常开或常闭接点。

OR、OR NOT 表示在原先接点上并联一个常开或常闭接点。

指　令		注　释
指令符	定义号	
STR	01	两个常开接点并联再
OR	03	与一个常闭接点串联
AND NOT	02	
OUT	20	控制20号输出线圈一
STR	04	个常开接点与一个常
OR NOT	06	闭接点并联再与一个
AND	05	常开接点串联
OUT	21	控制 21 号输出线圈



另外，AND TMR、AND CNT、AND NOT TMR、AND NOT CNT 指令使用方法与 AND、AND NOT 相同，只是定义号使用定时器/计数器定义号。

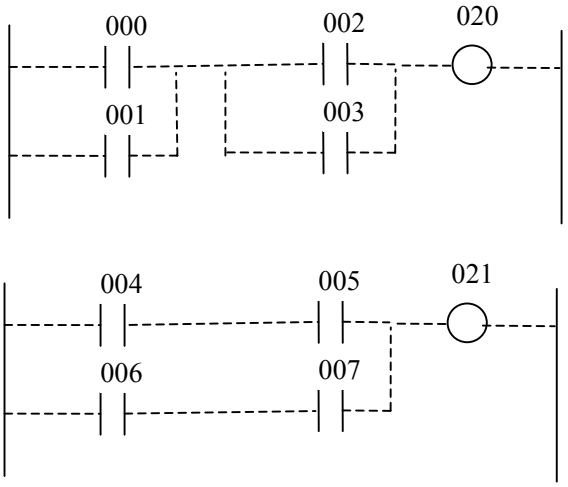
OR TMR、OR CNT、OR NOT TMR、OR NOT CNT 指令使用方法与 OR、OR NOT 相同，只是定义号使用定时器/计数器定义号。

三、AND STR、OR STR

逻辑组串联或并联指令

遇到逻辑组第一个触点，要用 STR（压入堆栈）指令。

指　令		注　释
指令符	定义号	
STR	000	
OR	001	
STR	002	压入堆栈
OR	003	逻辑组串联
AND STR		
OUT	020	
STR	004	
AND	005	
STR	006	
AND	007	压入堆栈
OR STR		逻辑组并联
OUT	021	

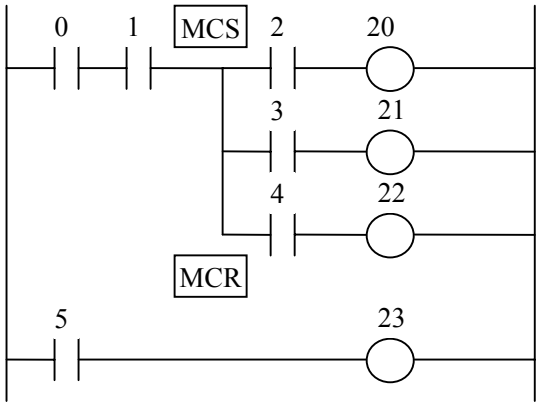


在 SR—21 中，“AND STR”和“OR STR”只有连续七级。请注意“连续”这两个字。在整个程序中，可以有无数次用“AND STR”和“OR STR”，但不能连续超过七级。

四、MCS、MCR

主控功能：MCS 主控开始，MCR 主控结束。

指　令		注　释
指令符	定义号	
STR	0	
AND	1	
MCS		主控开始
STR	2	
OUT	20	
STR	3	
OUT	21	
STR	4	
OUT	22	主控结束
MCR		
STR	5	
OUT	23	



在编程中，我们经常会遇到许多线圈同时受一个接点（或一组接点）的控制。如果在每个线圈的控制逻辑中都编入该逻辑条件，那么势必要增加许多接点，浪费用户存储器。遇到这种情况，在 SR—21 中，可以用“主控继电器”功能来解决。

首先在梯形图的左边编入共同的逻辑条件，然后编入 MCS（主控开始）功能。接着分别编入各自的控制逻辑。最后是 MCR（主控结束）功能。当共同逻辑条件为“1”（ON）时，MCS 和 MCR 之间的逻辑可正常操作，如共同逻辑条件为“0”（OFF）时，则 MCS 和 MCR 之间的逻辑行，所包含的所有线圈均为断开，定时器复零，计数器停止计数（但不复零）。

主控嵌套最多七级，如超过七级，将会出错。

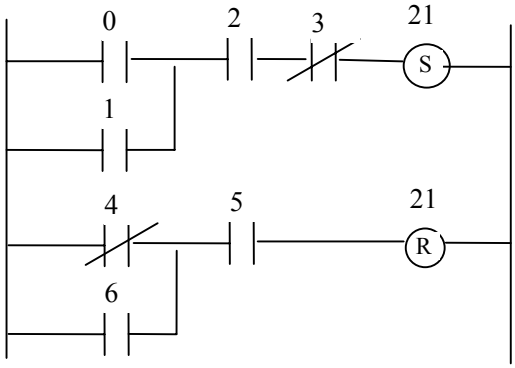
五、SET、RST

置位、复位指令。

用 SET 指令设定的线圈（Q、IR、SR），一旦其控制逻辑接通，则被置位，并保持接通状态（即使控制逻辑断开），必需当设定同一线圈的 RST 功能有效时才能使被 SET 置位的线圈断开。

SET/RST 指令一般成对出现（可在程序的任何位置），在实际应用中，有时也可能只用其中之一，来实现某些功能，这有待用户灵活运用。

指　令		注　释
指令符	定义号	
STR	0	
OR	1	
AND	2	21#线圈
AND NOT	3	置位
SET	21	
STR NOT	4	
OR	6	21#线圈
AND	5	复位
RST	21	



六、停电记忆线圈的使用

在某些应用场合，PLC 断电（关机或失电）时的一些状态需要保持，以便当 PLC 恢复运行时能保证设备工作的连续性。

在 PLC 内，一般的内部线圈，输出线圈，不管它在断电以前是什么状态，在断电后再恢复通电时，其状态总被清零，就无法维持断电以前的状态。为此，在 SR—21 中，设置了 28 个带停电记忆功能的内部线圈，它们的定义号为 340—373。（必须注意的是，只有当 CPU 模块上的开关 SW1（见 1—14）处于 ON 位置时，它们才具有停电记忆功能；否则，与 160—337 线圈的功能一样。当 CPU 的开关 1 处 ON 位置时，这 28 个线圈能在 PLC 断电后再通电时，维持断电以前的状态）。另外，计数器与移位寄存器线圈也有停电记忆功能（与 CPU 的开关 1 状态无关）。

停电记忆的内部线圈，在 PLC 正常运行时，用法及功能与无停电记忆功能的内部线圈完全一样，它唯一的区别就是前者在 PLC 断电扣再复电时，维持停电前的状态；而后者在复电时，状态总是断开的。

SE—22PLC 的内部线圈 340—373，1020—1377 有停电记忆功能，当系统参数 004 的个位状态为“1”时，340—373 有停电记忆功能，当系统参数 004 的百位状态为“1”时，1020—1377 有停电记忆功能。

停电记忆线圈与 SET 指令共同使用使编程较方便。

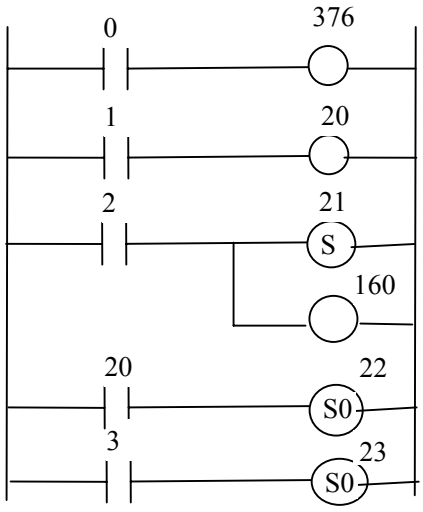
七、输出的禁止

在 SR—21 中，专设一个特殊功能的内部线圈 376 号线圈，当该线圈置“1”（ON）时，禁止硬件输出，而不影响程序的执行。

SET OUT

“SET OUT”设定的输出，在一般情况下用“OUT”设定的输出完全一样，但当 376 线圈接通时，用“SET OUT”设定的输出（包括输出点）的状态不受 376 线圈影响。SET 指令设定的输出线圈也不受 376 线圈的影响。

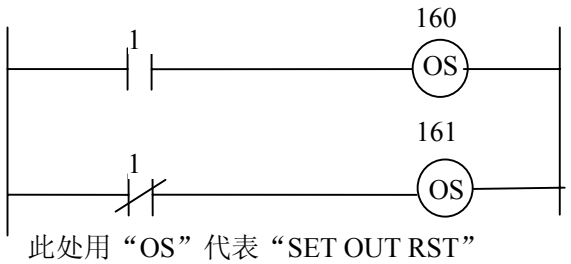
指　令		注　释
指令符	定义号	
STR	0	禁止输出线圈
OUT	376	376 通
STR	1	
OUT	20	
STR	2	
SET	21	
OUT	160	
STR	20	
SET OUT	22	21、22、23 号线圈输出不受 376 影响
STR	3	
SET OUT	23	



八、SET OUT RST（单触发指令）

在 SR—21 中，用“SET OUT RST”指令定义的线圈。在控制条件成立时，在一个扫描周期内接通。欲使它再次接通，需控制条件再次成立方可。SET OUT RST 指令只接通一个扫描周期。

指　令		注　释
指令符	定义号	
STR	1	当“1”为 ON，160 接通。当“1”为 OFF 时 161 接通，各为一个扫描周期。
SET OUT RST	160	
STR NOT	1	
SET OUT RST	161	



此处用“OS”代表“SET OUT RST”

§4—1—2 定时器和计数器

定时器和计数器是顺序控制系统中常用的功能元件，也是所有 PLC 都具有的功能。在 SR—21 系列中，共有 64 个定时器和计数器，它们的定义号是 600~677。类似于“OUT”功能。定时器，计数器线圈也是在逻辑行的末端设定，任一定时器/计数器线圈的接点（常开与/或常闭），可在程序中不受限制地使用，而线圈只能在程序中设定一次。当定时器或计数器的控制逻辑接通时，进行计时或计数，当计到预置值时，它们的接点动作（常开闭合，常闭断开）。定时器和计数器的预置值用四位十进制数表示，每个定时器的预置值为 0.1 秒到 999.9 秒，每个计数器的预置值为 1~9999。600~677 这个 64 个线圈，每个都既可用作定时器，也可用作计数器，例如，您可将 600 设定为定时器，601 用作计数器，但是，一个线圈在一段程序中不能同时用作定时器和计数器，因此，在 SR—21 中 ，可有 0~64 个定时器，可有 0~64 个计数器，但两者加起来总数不能多于 64 个，在这 64 个定时器/计数器中，600~673 这 60 个，它们的预置值是在程序中设定的。但它们的预置值也可以不在程序中直接写出，可用某两个数据存贮单元（共 16 位）的内容作为它们的预置值，在程序中只写出这个预置值所在的数据存贮单元。另外有四个定时器与/或计数器即 674~677，必须由外部的拨盘开关来设定，用拨盘开关设定的好处是，可以根据工艺的要求，随时地更改定时器/计数器的预置值，方便直观。连接拨盘开关的是拨盘接口模块 E—01D。

E—01D 模块的输入数据除了提供 674~677 定时器和计数器使用以外，同时送入数据寄存器 R564~R573，内容如下：

E—01D 的 0 通道：R564 低 2 位，R565 高 2 位

E—01D 的 1 通道：R566 低 2 位，R567 高 2 位

E—01D 的 2 通道：R570 低 2 位，R571 高 2 位

E—01D 的 3 通道：R572 低 2 位，R573 高 2 位

下面我们分别介绍定时器和计数器的编程。

一、定时器

定时器（图 4.1）只有一个控制逻辑行，当该逻辑行的控制逻辑接通时，定时器开始计时，计时是从 000.0 秒开始，当累积时间达到预置值时，它的接点将动作（常开接通，常闭断开），如此时控制逻辑仍然接通，则定时器继续往上计时，直到计到 999.9 秒，监控时地址/数据显示在 999.9 但仍计时下去，而其接点一直保持动作状态（即使到计了 999.9 秒，以后一直显示 999.9）。不论计时计到何时，只要它的控制逻辑断开或 CPU 中止运行，则定时器计时值马上复零，如其接点已动作，则立即复位。在 SR—21 中，定时器预置值是 0.1 秒的倍数，它也可设定为 0.01 秒,0.001 秒的倍数。定时器的时钟信号由 CPU 内部晶振产生，不受电源频率影响。

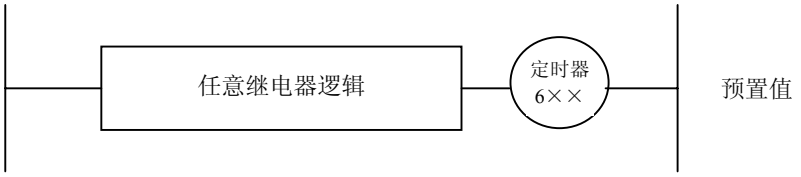


图 4.1 定时器逻辑行

1、 定时单位为 0.1 秒的定时器指令 TMR6××

在逻辑行中，定时器线圈和接点“T6××”表示，编程时的助记符为“TMR6××”在设定线圈（定时器）时，应在线圈之后，立即写上并键入预置值（674—677 除外）

定时器可编成自复位，即用该定时器的一个常闭接点用为它的控制逻辑。当定时器计时到预置值时，其常闭接点断开，使定时器复零，常闭接点重新闭合，定时器又重新开始计时，利用这样的功能，可产生任意周期的时钟信号，供程序使用。

指　令		注　释
指令符	定义号	
STR	0	定时器 T600，每 5 秒接通一次，接通时间为一扫描周期
AND NOT TMR	600	
TMR	600	
	5	
STR	1	“1”或“2”接通 15 秒后，20 号线圈输出，21 号线圈断开
OR	2	
TMR	601	
	15	
STR TMR	601	
OUT	20	
STR NOT TMR	601	
OUT	21	
STR	3	“3”接通 6、7 秒后，22 号线圈输出
TMR	602	
	6.7	
STR TMR	602	
OUT	22	

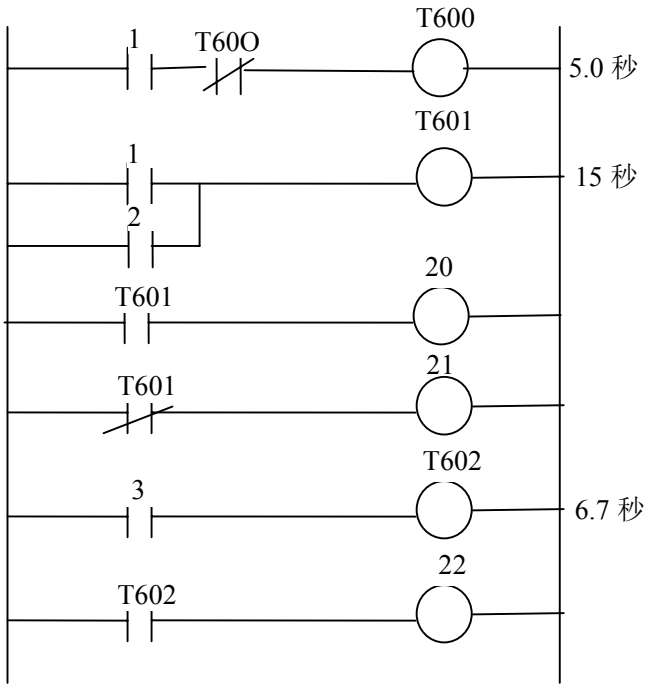


图 4.2 定时器逻辑举例

2、 定时单位为 0.01s 的定时器 TMR6.XX

SR—21PLC 除了 0.1s 定时器外，还提供了定时单位为 0.01s 和 0.001s 两种定时器。其中 0.01s 定时器的助记符为 TMR6.XX。功能和动作时序与 TMR600—673 相同。但是，TMR6.XX 不可在快速扫描子程序中使用。

3、定时单位为 0.001s 的特殊定时器指令 TMR. 6XX

详见快速扫描子程序中的说明。

二、计数器指令 CNT6XX

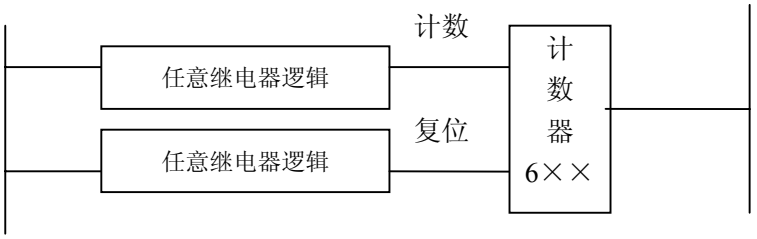
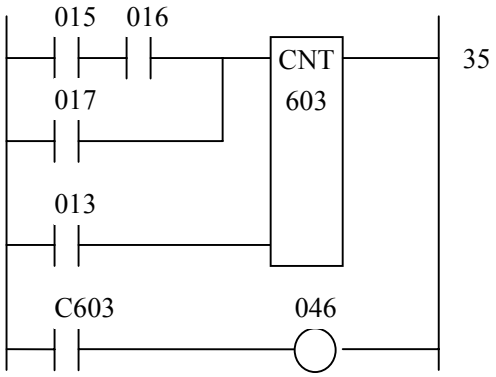


图 4.3 计数器逻辑

计数器的操作除了需要两个逻辑行控制外，与定时器类似，如图 4.3 所示。上面一个逻辑行控制计数器的递增计数，当该行从断（OFF）到通（ON）时，计数器的计数值就加 1。当该逻辑行断开，并再次从断到通（OFF—ON）时，计数值再加 1。所有计数器都是向预置值作递增计数。下面一个逻辑行（第二个 STR 功能）控制计数器的复位。无论何时，只要这一行接通，计数器的计数值就复位（回到零），在这一行接通的情况下，上一个逻辑行不起作用，计数器和定时器的另一个区别就是在 CPU 失电时，计数器有停电记忆功能。在 SE—2PLC 中，当系统参数 004 的十位状态为“1”时，计数器有停电记忆功能。

计数器举例

指　令		注　释
指令符	定义号	
STR	015	计数端
AND	016	
OR	017	
STR	013	复位端
CNT	603	计数器计数（递增）
——	035	
STR CNT	603	当计数器计到 35 时
OUT	046	46 线圈输出



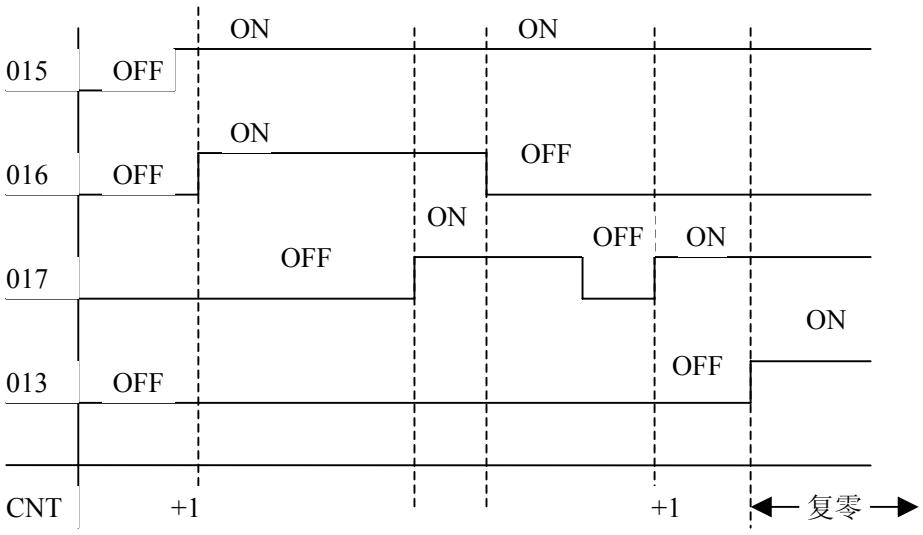


图 4.4 计数器时序图

注：在 SR—21 中，每个定时器的最大预置值是 999.9 秒，每个计数器的最大预置值为 9999。如果需要的预置值超过这个数，我们可以通过几个定时器和计数器的串联来扩充预置值范围。

§4—1—3 鼓形控制器

在顺序控制中，很多场合都是按步操作的，即第一步某些机构动作，第二步另一些机构动作，依次顺序动作。在以往，可以用机械式的鼓形控制器（亦称凸轮控制器）来满足这样的控制要求。在 SR—21 中，可以用计数器来实现这种功能。

图 4.5 是一个鼓形控制器的例子。鼓形控制器的步进由用户选定的开关控制，在本例中用了一个按钮。按钮每按一次，就进入下一步，比如，在第一步时，进行动作 C，按一次按钮，进入第二步，进行动作 A 和 D……。用机械凸轮来进行顺序控制。步数和输出点数受到很大的限制，而且成本较高。采用 PLC 来模拟这种机构，由于是软件实现的，因此编程、修改灵活方便，总步数和输出数几乎可满足所有场合的要求。

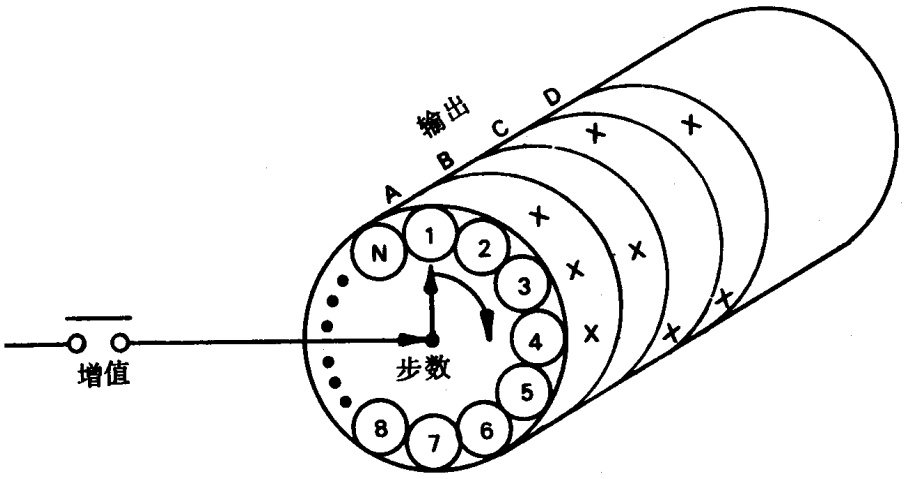


图 4.5 鼓形控制器

为了更清楚地说明 SR—21 的鼓形控制器功能，我们结合一个具体例子来介绍，一般只按工艺要求，画出基于时间条件的动作图或表。图 4.6 就是一个例子。在这个例子中，有六个输出点按时间顺序改变通/断状态，图中的水平线表示输出接通。

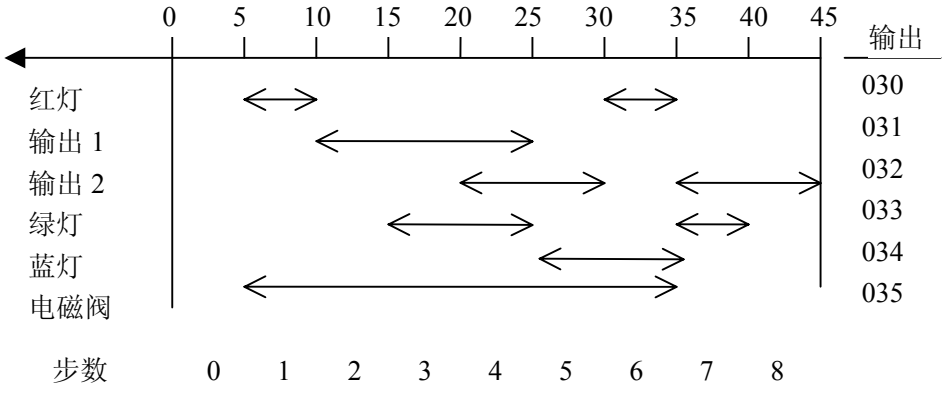
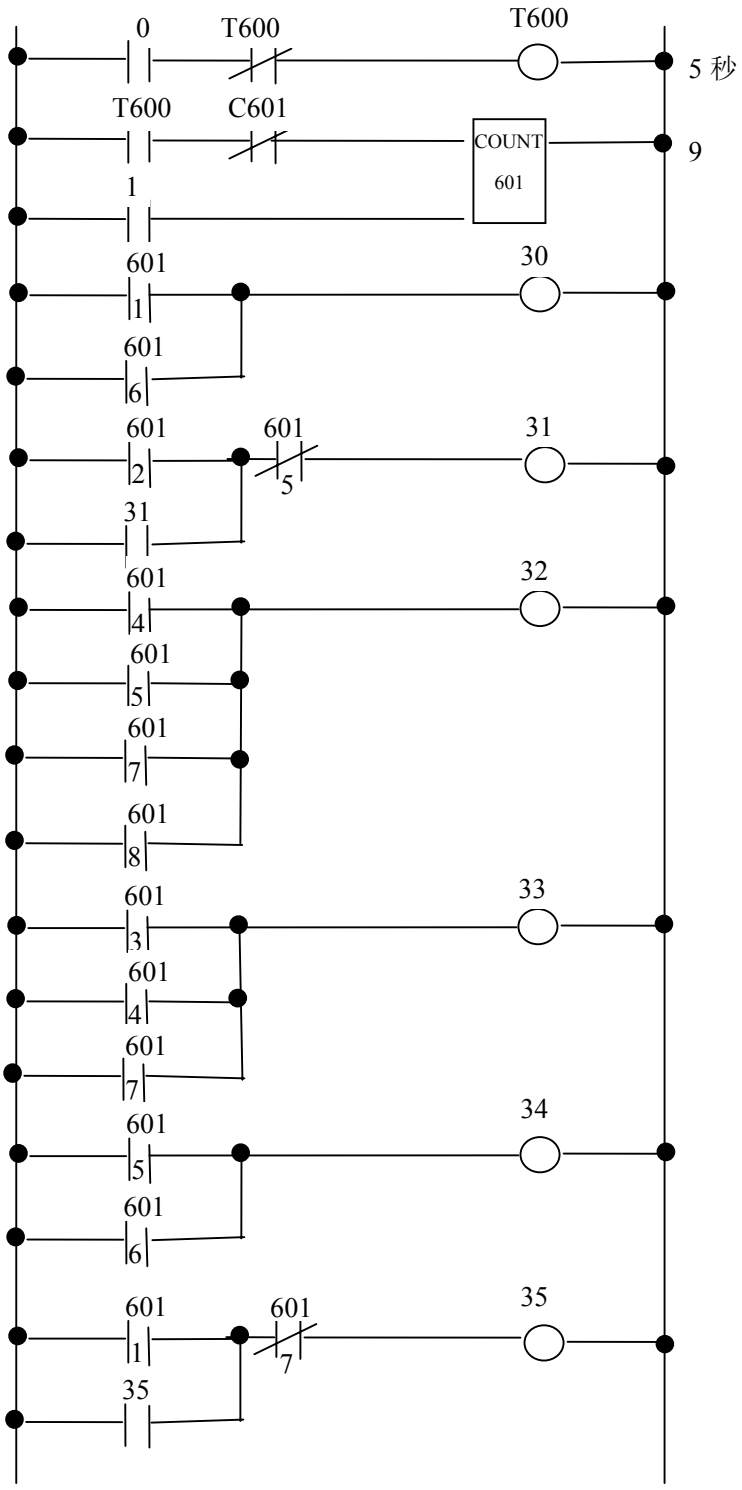


图 4.6 鼓形控制器动作图举例

用 SR—21 的计数器作鼓形控制器，首先要设定一个计数器，使它基于工艺要求（时间或条件）进行计数，计一次数即为一个工步，然后只需简单地按动作图，编出每个输出应该接通的步数。图 4.7 就是实现图 4.6 的动作图的梯形图程序。




```
STR              0
AND NOT TMR 600
TMR              600
…                5
STR TMR          600
AND NOT CNT 601
STR              001
CNT              601
…                9
STR              601
…                1
OR               601
…                6
OUT              30
STR              601
…                2
OR               031
AND NOT          601
…                5
OUT              31
STR              601
…                4
OR               601
…                5
OR               601
…                7
OR               601
…                8
OUT              32
STR              601
…                3
OR               601
…                4
OR               601
…                7
OUT              33
STR              601
…                5
OR               601
…                6
OUT              34
STR              601
…                1
OR               35
AND NOT          601
…                7
OUT              35
```

图 4.7 鼓形控制器编程

有了前面几节的知识，我们不难读懂这样的梯形图程序，并据此了解鼓形控制器的编程方法。这里仅是刚接触了类似这样的接点：

（a）接点表示当 601 计数值为 1 时，该接点接通，（b）接点表示当 601 计数值为 5 时该接点断开，当计数值不是 1 时，（a）接点总是断开的，当计数值不是 5 时，（b）接点总是接通的，把它们写成语句时，一定要注意不能在 601 之前写“CNT”，同时要在后面写出步数。用这方法来表示计数器的一种特殊用法，以区别于计数器线圈的接点。

SR—21 共有 64 个定时器/计数器，因此最多可有 64 个这样的鼓形控制器；每个计数器的计数范围为 0~9999，因此，每个鼓形控制数最多可编 10000 步。

§4—1—4 移位寄存器

一、功能

移位寄存器是一组受时钟信号控制，与时钟信号同步动作的存储单元，图 4.8 为一个典型的移位寄存器。每个存储单元可以存储表示物理量的两个状态的单位数据信号（如 1 或 0），每当时钟信号由 OFF 到 ON 时，该数据信息随时钟信号进入第一存储单元，同时每单元的内容依次向后移 1 位，最后级的内容丢失。

在 SR—21 中，有 128 个带停电记忆功能的存储单元作移位寄存器使用，它们可以组成若干个移位寄存器（位数不定），但总数不能超过 128。

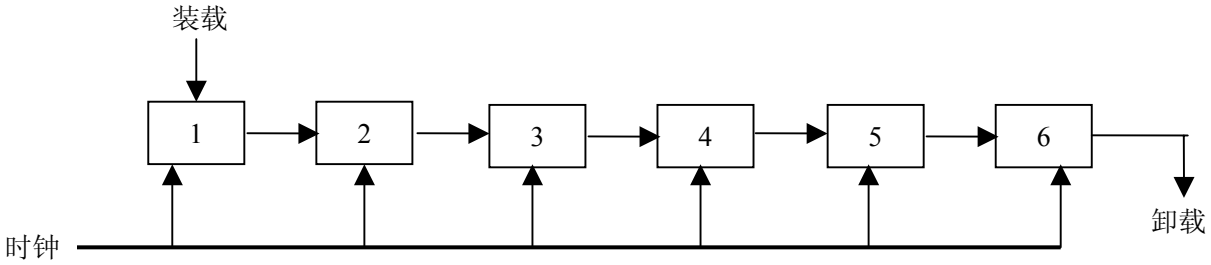


图 4.8 典型的移位寄存器

二、移位寄存器指令 SR

移位寄存器的设定，需要三个逻辑行，每一行都可以是任何串并联的逻辑组合。第一行是数据输入端，第二行为时钟信号端，第三行是复位端。

指　令		注　释
指令符	定义号	
STR	1	数据输入
STR	2	时钟信号输入
STR	3	复位输入
SR	400	
	407	
STR	401	
OUT	20	
STR NOT	403	
OUT	21	

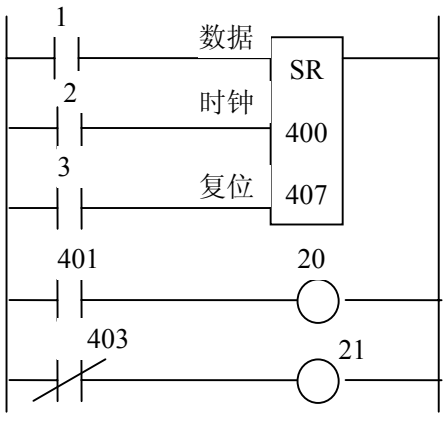


图 4.9　移位寄存器梯形图举例

当时钟信号端的逻辑从 OFF 变到 ON 转变时，移位寄存器进行移位，在图 4.9 的移位寄存器中当 2 号输入从 OFF 到 ON 时，数据输入端的状态移入 400，400 的状态移到 401，401 的状态移到 402，……，406 的状态移到 407，407 的状态就丢失了。进行移位时，第一位的状态（例中的 400），取决于时钟端从 OFF 到 ON 的时刻数据输入端的状态，如这时数据输入端是接通的（例中的 1 号输入是 ON），则进入第一位（400）的状态为 1（表示通）；否则，则进入 0（表示断）。无论何时，只要复位端（例中的 3 号输入）接通，则移位寄存器的各位均被清零。移位寄存器中的每一位都可用它的常开或常闭接点来控制其它线圈。如例中，用 401 的常开接点控制 20 号输出，用 403 的常闭接点控制 21 号输出。

移位寄存器还可以根据需要，编成其它形式的移位寄存器，例如构成双向移位寄存器，环形移位寄存器等，它也可构成鼓形控制器，使用非常灵活。

三、移位寄存器举例

图 4.10 为一条假想的装配生产机械。该机共有 12 个工位，工件可以是杯子或其它容器，该机的工作过程是固定的，完成对产品的装配。机上有三个独立的工作区，完成零件装配，焊接、夹紧、上螺母、油漆、贴商标等工作。为避免无零件时机械空操作，在第一个位置装有传感器，来检查有没有零件装入。右边的检验位置将正品和不合格品分开。机械上的每个位置都有相应的定义号。注意，由于受机械结构的限制，3、5、7、9、10、12 号位置仅用于传送工作。

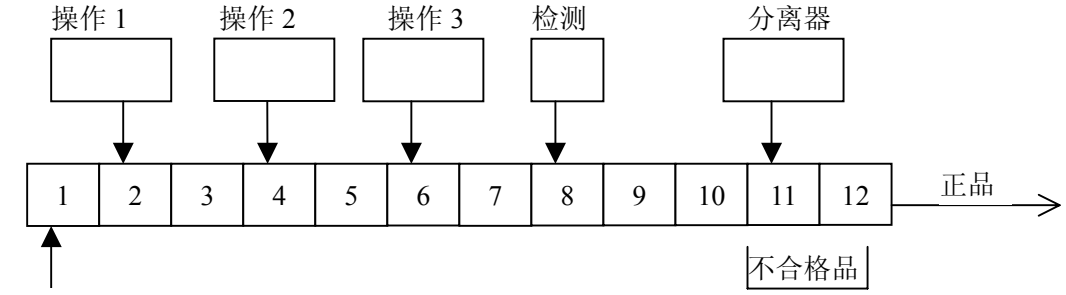


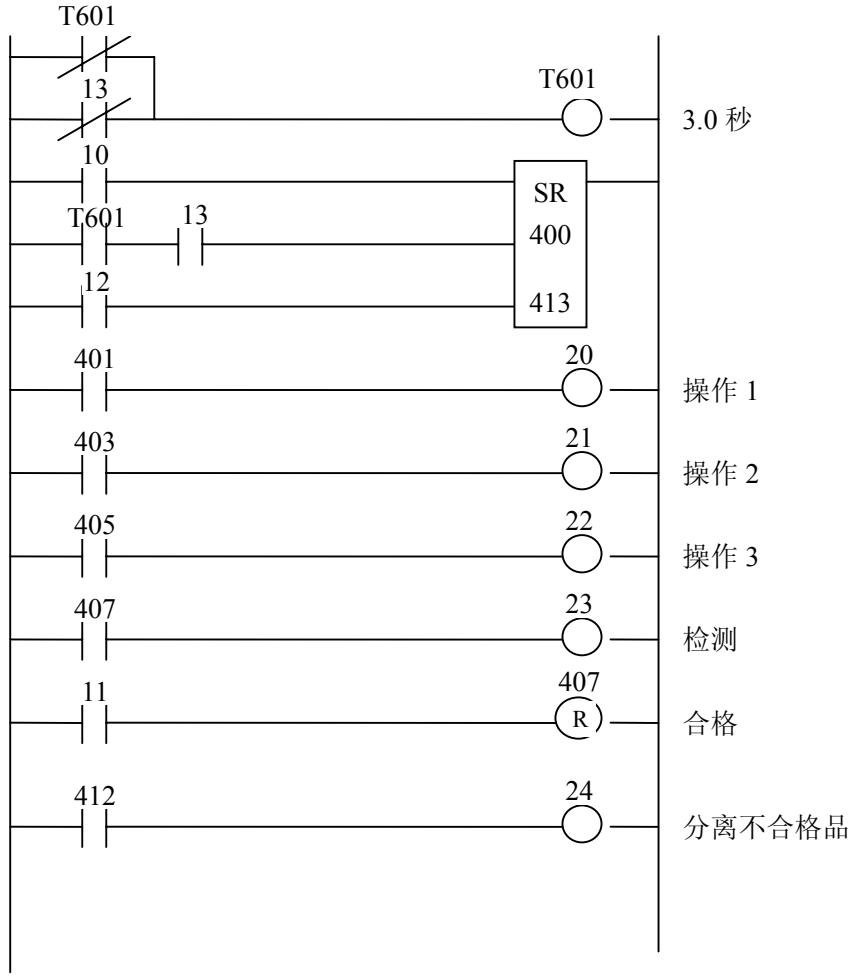
图 4.10 装配线示意图

假定 I/O 定义号分配为：

零件装入信号（输入）	10
检测合格（输入）	11
总复位（输入）	12
允许工作信号（输入）	13
操作 1（输出）	20
操作 2（输出）	21
操作 3（输出）	22
检测（输出）	23
分离器（输出）	24

该机械每 3 秒钟移一个工位。当允许工作时，定时器 601 每 3 秒产生一个输出脉冲，作为移位寄存器的时钟信号。

梯形图：




```
STRNOT   TMR   601
OR NOT         13
TMR            601
               3
STR            10
STR TMR        601
AND            13
STR            12
SR             400
               413
STR            401
OUT            20
STR            403
OUT            21
STR            405
OUT            22
STR            407
OUT            23
STR            11
RST            407
STR            412
OUT            24
```

图 4.11 装配机械控制程序

在正常情况下，零件连续不断在位置 1 装入，同时，零件装入信号（ON）进入 400。待零件移至位置 2，零件装入号也移至 401，并接通 20 输出，对零件作第一种装配操作。三秒后，该信号移入 402，又三秒后移入 403（此时零件正好进入位置 4），对零件作第 2 种装配操作。过两个 3 秒后，在位置 6 作第 3 种操作。当装配好的产品送到检测位置时，它对应的零件装入信号移到 407，输出 23 接通，对产品进行检测。如检测结果产品合格，则 11 接通，使用权 407 复位，如检测出产品不合格，407 不复位，当这个不合格品送到位置 11 时，正好一个“1”移入 412，使分离器动作，将不合格品剔出。

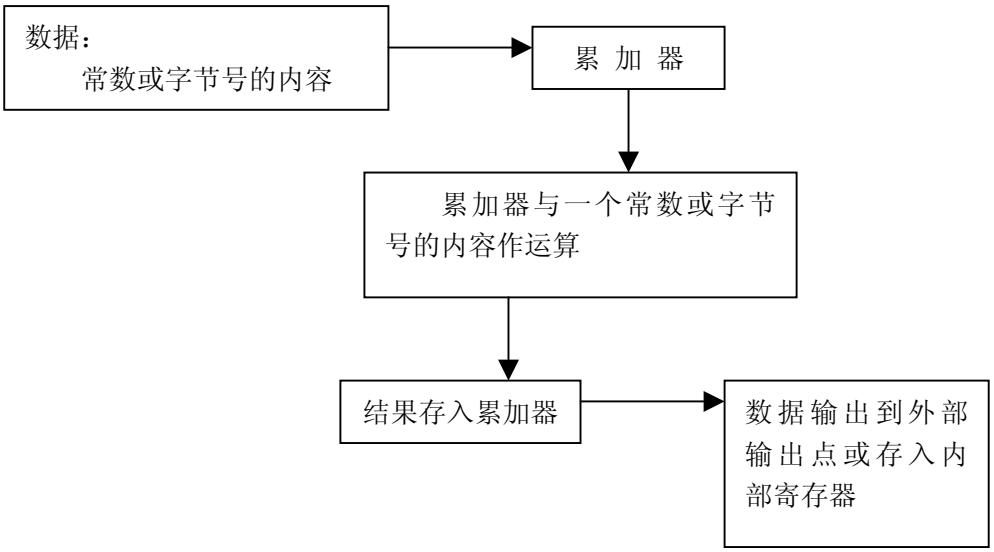
如果我们在该程序中加入某些其它逻辑，还可以提高程序的控制能力。例如，可以进行操作超时报警，记录正品，不合格的数量，班产、日产等。

§4—2 数据操作

§4—2—1 数据操作

SR—21 除了基本继电器逻辑功能外，另外还有数据传送、算术运算、逻辑操作，变换及外部故障诊断等数据操作功能。这些操作是多位数据的处理。

数据操作指令的数据处理均在一个被称为累加器的 16 位寄存器中进行。累加器中可存贮四位 BCD 数值（每位 BCD 数在累加器中占 4 位）或 16 位二进制数。当对累加器的内容执行指定的操作后，处理结果还存贮在累加器中。累加器中的数据也可以被传送到外部输出模块或内部的存贮单元中。



数据操作指令可以分以下五类：

（1）数据传送：将一个数据传送到累加器中，或将累加器中的数据传送到数据寄存器或输出单元。

（2）算术运算：将累加器中 BCD 数据与另一个 BCD 数据作四则运算或比较。

（3）逻辑操作：将累加器的数据与另一个数据的相对应的各位作逻辑运算。

（4）数据变换：将累加器中的数据进行变换（BCD、二进制、移位、编译码、取反）。

（5）故障诊断：对受 PLC 控制的外部设备进行故障诊断。

第（5）类指令是 SR—21 的一种特殊功能，与前几类功能不一样，我们将单独另立一节进行叙述。

第（4）类指令的操作数只有一个，即累加器中的数据。

第（2）、（3）类指令都涉及到两个操作数，一个数据为累加器中的数据，另一个数据由用户程序指定，可以是一个常数，也可以是一个间接数。

第（1）类指令中，一种是将一个常数或一个间接数送入累加器，另一种是将累加器中的数据送到指定单元。

第（1）、（2）、（3）类指令都由两条语句组成，第一条语句规定操作的类型，第二条语句为操作数，正如前面所述，操作数分为常数和间接数。

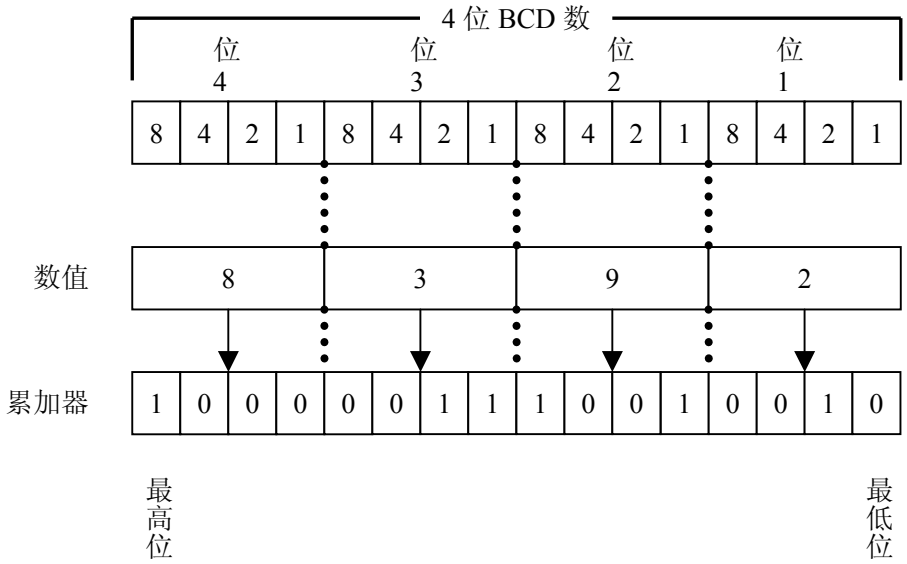
（一） 操作数为常数

如操作数为常数，指令形式如下：

指 令	ENT	SHT	×	×	×	×	ENT

这里，××××表示 9999 以内的常数。

当数据操作指令的操作数为常数时，在指令书写和键入时，只需用十进制数形式表示。但它在累加器或其它存储单元的存贮形式是 BCD 数，例如：一个四位十进制数 8392（八千三百九拾二），在累加器的存贮形式为：



（二） 操作数为间接数

这里讲的间接数，是指在指令中不直接写出操作数的数值，而是写出操作数所在的存贮单元。

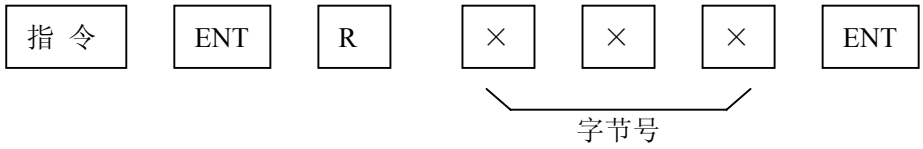
在前面介绍的开关量逻辑中，我们已知道，输入、输出及存贮中间状态的单元的编号为定义号，每个定义号所表述的状态是通或断，仅需占 1 位，即 1（ON）和 0（OFF）。而数据操作指令处理的是 16 位、8 位或 4 位的数据，它需占用 16 位，8 位或 4 位（4 位，2 位或 1 位十进制数）。用字节号表示的存储单元都是八位（除定时器/计数器累积值寄存器外），下面是 SR—21/SE—22 字节号一览表：

表 4.1 SR—21/SE—22 字节号一览表

单 元 类 型	SR—21/22	SE—22
I/O	R000~R017，R070~R076	←
内部继电器	R020~R033	←
停电记忆内部线圈	R034~R037	R034~R037，R102~R137
移位寄存器	R040~R057	←
定时器/计数器 累积值寄存器（16 位）	R600~R677	←
数据寄存器（8 位）	R400~R577	R160~R577，R700~R777
级线圈		R140~R157

注意：上表中的字节号也是八进制数。如果 64 个定时器没有全部使用，600~677 中的剩余的累积值寄存器也可作数据寄存器用，但它的 16 位内容须同时存取。

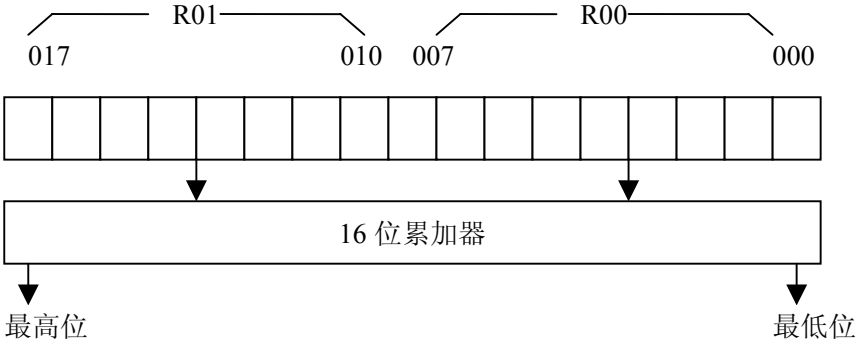
当操作数为间接数时，指令形式如下：



在数据操作指令中，字节号之前要写“R”，以便与常数数据区分。

一个字节号所表示的数据是八位，而大部分数据操作都有是 16 位数据处理，因此，这样的指令就处理连续的两个字节号的存贮内容（600~677 除外）。例如，要将 16 位累加器的内容存入数据寄存器，那就需要用两个数据寄存器，如指令中写的字节号为 R400，那么实际上是将累加器的内容存放到了 400 和 401 两个数据寄存器中，400 中存的是低 8 位，401 中存的是高 8 位。

除了数据寄存器（400~577）和定时/计数器累积值寄存器（600~677）以外，其他每个字节号实际上都是八个定义号组成的。如 R00，即为 000~007 这 8 个 I/O 点定义号。例如，要将 000~017 这 16 个输入点的状态读入累加器，在指令中写出字节号 R00 即可，存贮形式如下：



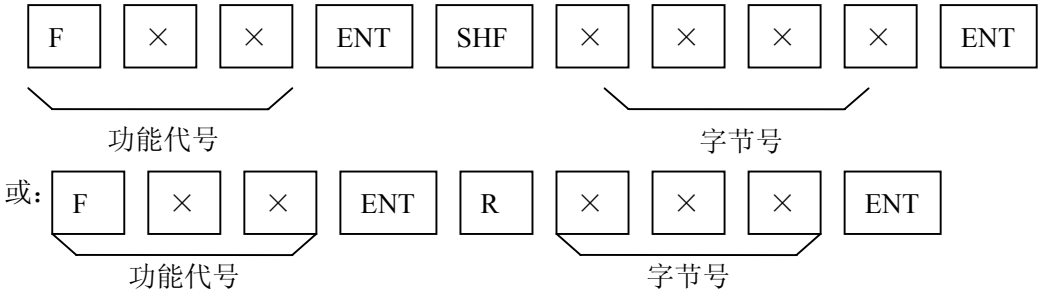
如果 000 点是接通的，则在累加器的最低位存入的状态为“1”再如 017 点是断开的，则在累加器的最高位存入的状态为“0”。

为了后面叙述方便，我们将 16 位累加器的每一位都编上序号，如下所示：

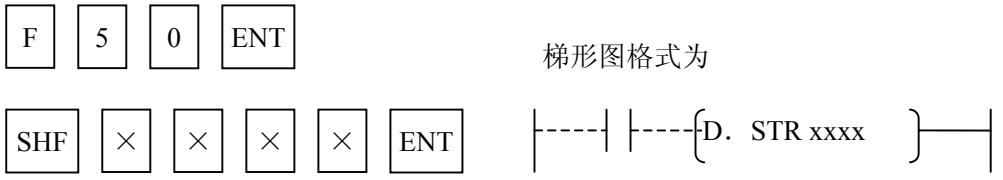
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

第 0 位为最低位，15 位是最高位。

在 SR—21 中，每种数据操作指令都用一个功能代号（F××）表示，表 4.2 为功能代号一览表，当键入数据操作指令时，操作顺序如下：



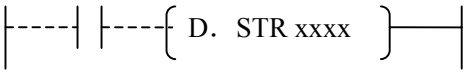
例如：执行 4 位 BCD 数送入累加器，其指令为 D.STR，功能代码为 F50，则编程器键入格式（指令符）为



或 [F] [5] [0] [ENT]

梯形图格式为

[R] [×] [×] [×] [ENT]



在以后讲述每条指令时就不叙述了，只要彷照上例输入即可。

表 4.2 数据操作指令的功能代号

指　　令	功能代号	指　　令	功能代号
D．STR（数据读入）	F50	×（乘）	F73
D．STR1	F51	÷（除）	F74
D．STR2	F52	D．AND（数据与）	F75
D．STR3	F53	D．OR（数据或）	F76
D．STR5	F55	右移	F80
D．OUT（数据输出）	F60	左移	F81
D．OUT1	F61	译码	F82
D．OUT2	F62	编码	F83
D．OUT3	F63	INV（取反）	F84
D．OUT5	F65	BIN（BCD→BIN）	F85
比较	F70	BCD（BIN→BCD）	F86
+（加）	F71	外部故障诊断	F20
—（减）	F72		
SEG 变换（仅 SE—22）	F87	寄存器间址读/写命令（仅 SE—22）	F27/F26
RAM 数据→累加器（仅 SR—21/22）	F56	寄存器内容—1（仅 SR—21/22）	F78
RAM 数据→数据寄存器(仅 SR—21/22）	F57	参数监视，设定命令	F25
ROM 数据→累加器（仅 SR—21/22）	F58	智能模块控制命令	F29
ROM 数据→数据寄存器(仅 SR—21/22）	F59	智能模块控制命令	F30~F33
累加器内容→RAM（仅 SR—21/22）	F66	智能模块控制命令	F40~F42
寄存器内容+1（仅 SR—21/22）	F77	通讯指令（仅 SE—22）	F88~F94

在本节后面部分分别介绍了每一条指令的具体信息。在介绍时我们一般都指下面的格式

指令的名称

功能代号

指令在累加器中完成的相应操作

在指令中可用的字节号(有效字节号)

某些指令的编程例子

另外，SR—21 中有 6 个作标志使用的内部线圈。当有些数据操作完成后，使某些相应的标志线圈在一定情况下接通。这些标志线圈的接点可用来驱动输出或完成某些功能。这些特殊线圈的定义号（注意，它们是表示通/断两个状态的开关量）及其意义如下：

线圈定义号	内容	线圈定义号	内容
772	>、大于	775	进位或借位
773	=、等于	776	零
774	<、小于	777	溢出

§4—2—2 数据传送

数据传送包括两类：

① 将一个数据送入累加器（数据存贮）

② 将累加器数据送到某存储单元（数据输出）

一、数据存贮

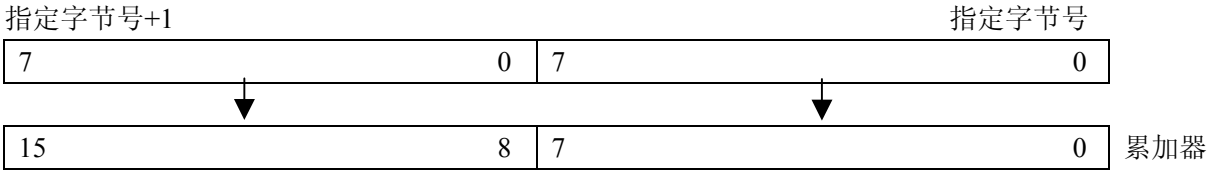
数据存贮指令共分五种：D.STR，D.STR1，D.STR2，D.STR3，D.STR5

D．STR（F50）

执行 D.STR（数据存贮）指令，是将一个四位数 BCD 常数或两个指定字节号的内容送入累加器。

提　　示

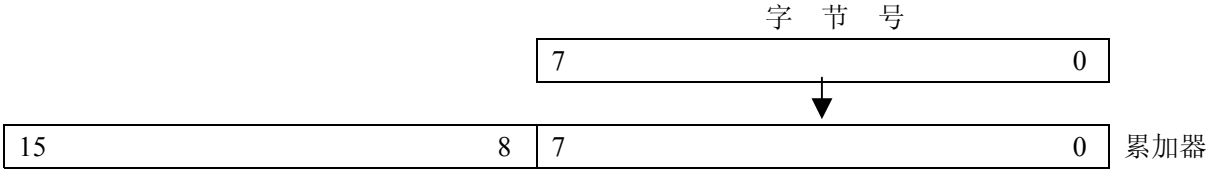
存入的数据如需进行算术运算，应为 BCD 数据，否则，在运算前要先将它转成 BCD 数据。如为非 BCD 数，则溢出标志（777）接通。



有　效　字　节　号	
R000~R016	I/O 点
R070~R075	I/O 点
R020~R036	内部线圈
R040~R056	移位寄存器
R400~R576	数据寄存器（8 位）
R600~R677	定时/计数器累积值寄存器（16 位）
0000~9999	BCD 常数

D．STR1（F51）

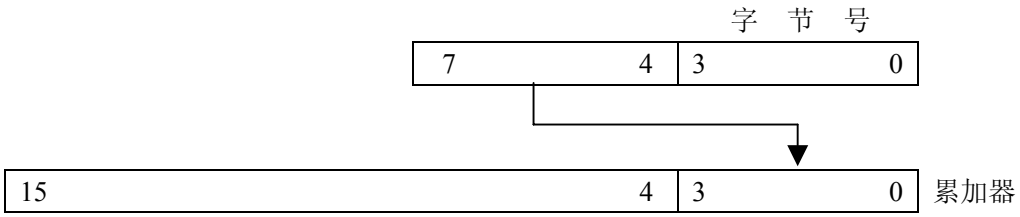
执行 D．STR1 指令是把一指定字节号的内容存贮到累加器的低 8 位中，累加器高 8 位均为零。



有　效　字　节　号	
R000~R017	I/O 点
R070~R076	I/O 点
R020~R037	内部线圈
R040~R057	移位寄存器
R400~R577	数据寄存器（8 位）

D．STR2（F52）

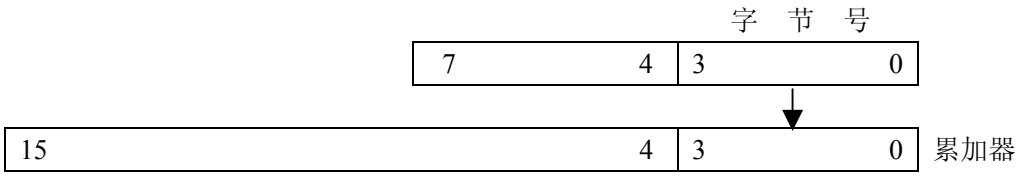
执行该指令是把指定字节号的高 4 位的内容存贮到累加器的低 4 位中。累加器的高 12 位均为零。



有效字节号与 D.STR1（F51）相同

D．STR3（F53）

执行该指令是把指定字节号的低 4 位的内容存贮到累加器的低 4 位中。累加器的高 12 位均为零。



有效字节号与 D.STR1（F51）相同

D．STR3（F55）

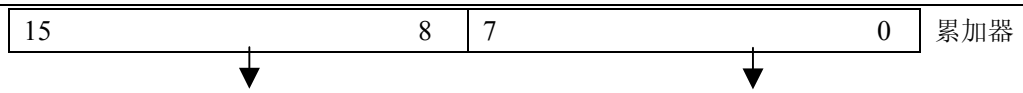
执行该指令时把一块 16 点输入模块上的内容存入到累加器中。

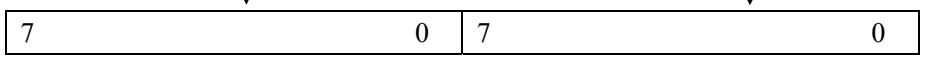
有 效 字 节 号	
000~007	16 点输入模块

二、数据输出

D．OUT（F60）

执行 D.OUT(数据输出)指令是把累加器（16 位）内容传送到二个指定的字节号中，字节号可以是输出、内部继电器、移位寄存器线圈、数据寄存器或 T/C 累积值寄存器。



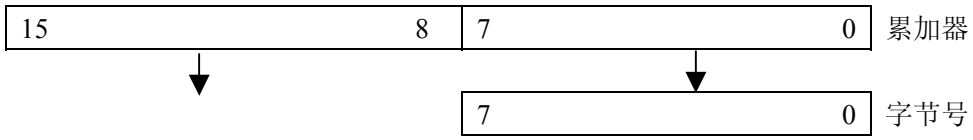


字节号+1　　字节号

有　效　字　节　号	
R000~R016	I/O 点
R070~R075	I/O 点
R020~R036	内部线圈
R040~R056	移位寄存器
R400~R576	数据寄存器（8 位）
R600~R677	T/C 累积值寄存器（16 位）

D．OUT1（F61）

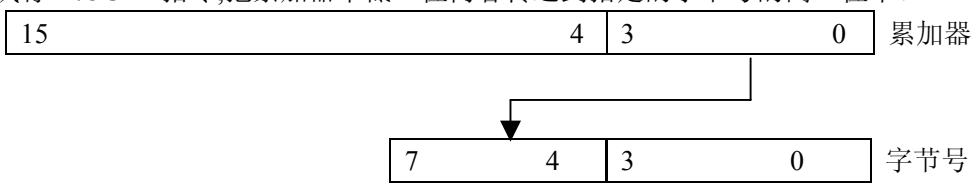
执行该指令是把累加器中低 8 位（0~7）内容传送到一个指定的字节号中。



有效字节号与 D.STR1(F51)相同

D．OUT2（F62）

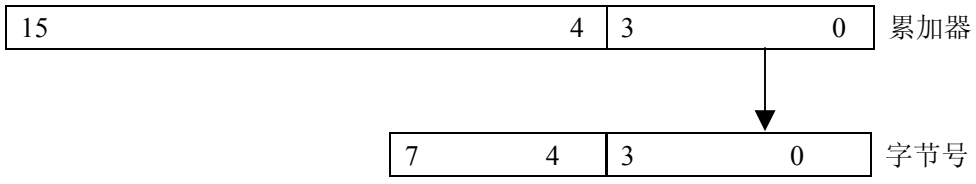
执行 D.OUT2 指令,把累加器中低 4 位内容传送到指定的字节号的高 4 位中。



有效字节号与 D.STR1(F51)相同

D．OUT3（F63）

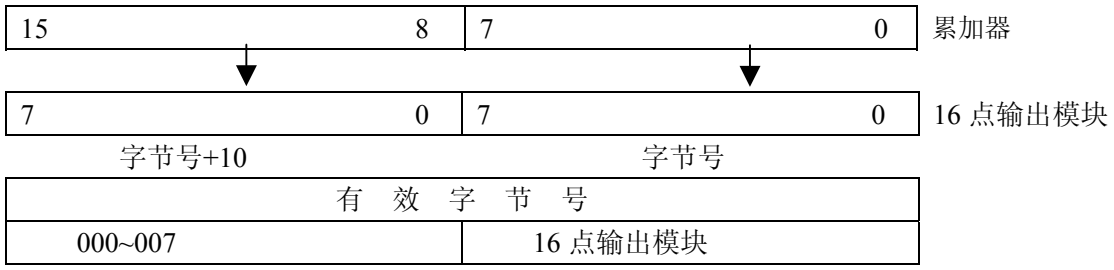
该指令是把累加器中低 4 位内容传送到指定字节号的低 4 位中。



有效字节号与 D.STR1(F51)相同

D．OUT5（F65）

该指令把累加器内容传送到一块 16 点输出模块中。



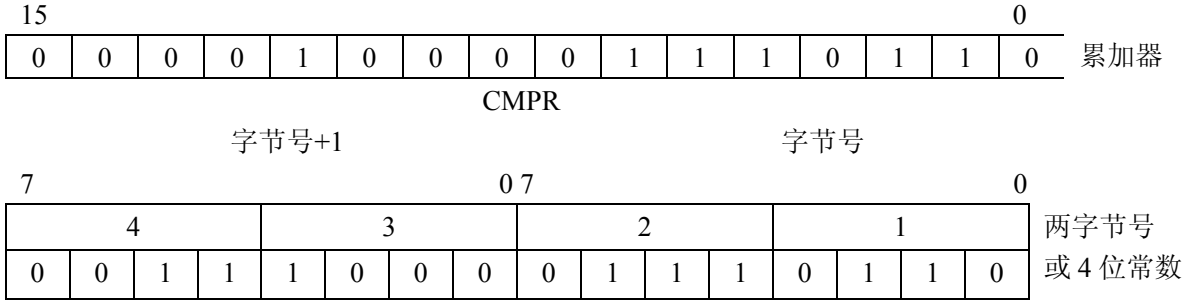
§4—2—3 算术运算

SR—21 的算术运算指令包括比较、+（加）、—（减），×（乘）÷（除）。它们都是两个 BCD 数据之间的运算，为便于理解，您可将之看成是两个 4 位十进制的运算。

CMPR（F70）比较

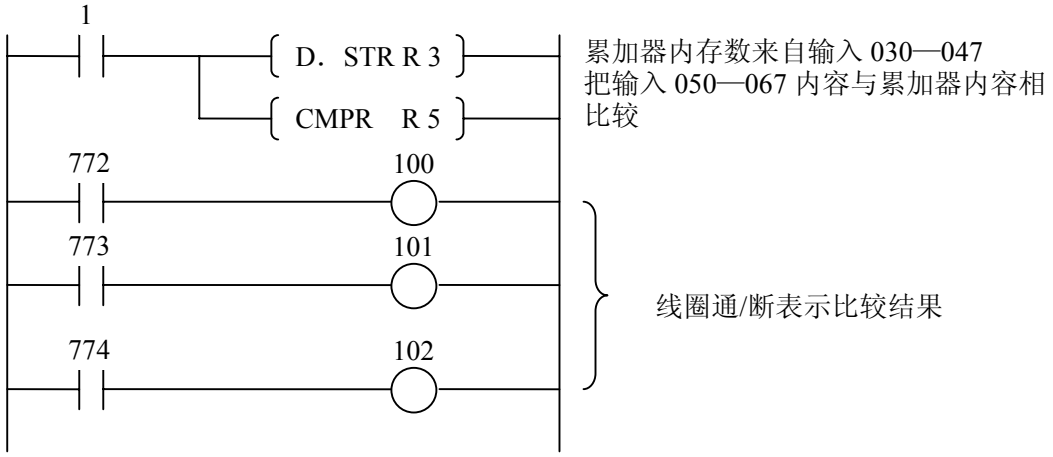
把指定的两字节号的内容（BCD 数）或 4 位 BCD 常数与累加器中内容（BCD 数）进行比较。比较结果将使三个特殊内部线圈中的一个接通，哪个接通取决于累加器的数值是大于(＞)、等于(=)、还是小于（＜）指定两字节号的数值或常数。

有效字节号与 D.STR（F50）相同



- 如果（累加器）＞（字节号），则 772 接通。
- 如果（累加器）=（字节号），则 773 接通。
- 如果（累加器）＜（字节号），则 774 接通。

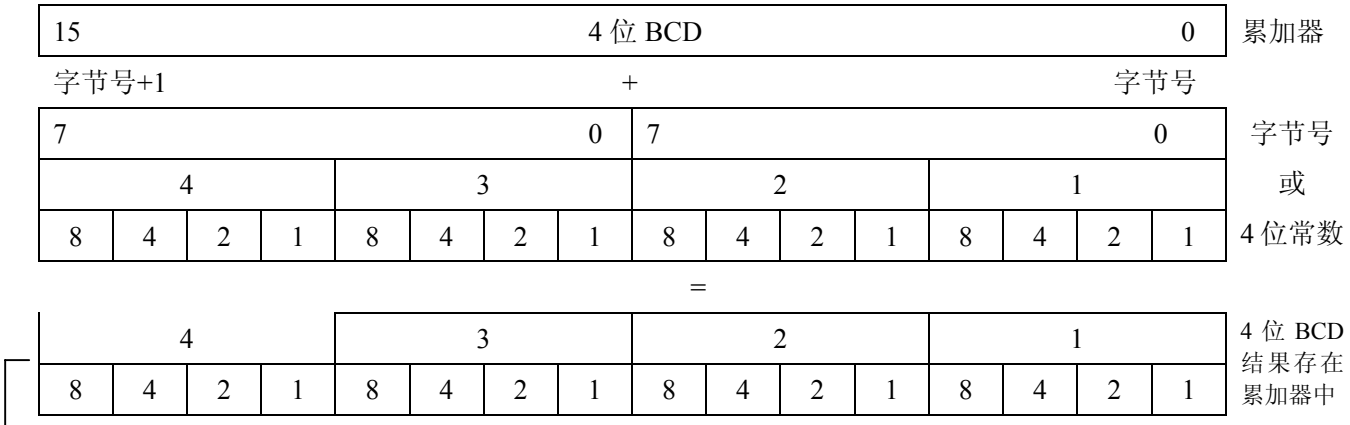
比较编程举例：



加法（+）、4 位 BCD 码（F71）

该指令把一个 4 位常数或指定的二字节号中的内容（BCD 数）和累加器中内容（BCD 数）相加，其结果以 4 位 BCD 形式存贮在累加器中。如果结果大于 9999，进位标志线圈（775 内部线圈）接通，如结果为零，则零标志线圈（776）接通。在程序中可以用标志线圈的接点来控制指示灯或其它标志装置的输出。

有效字节号与 D.STR（F50）相同



进位	零
775	776

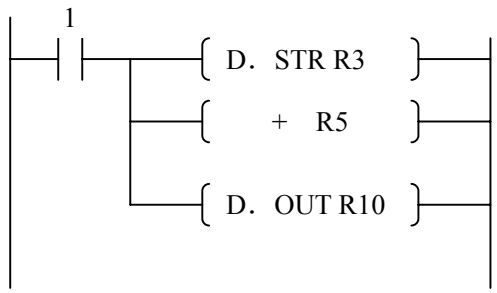
如果结果是 0000，零标志接通。
如果结果＞9999，进位标志接通。
如果结果＞9999，而且 BCD 数是 0000，则零标志和进位标志接通。

举　　例

```
   7777
  +2223
 ------
  10000
 ------
```

编程举例

进位标志接通 ◄ ► 零标志接通

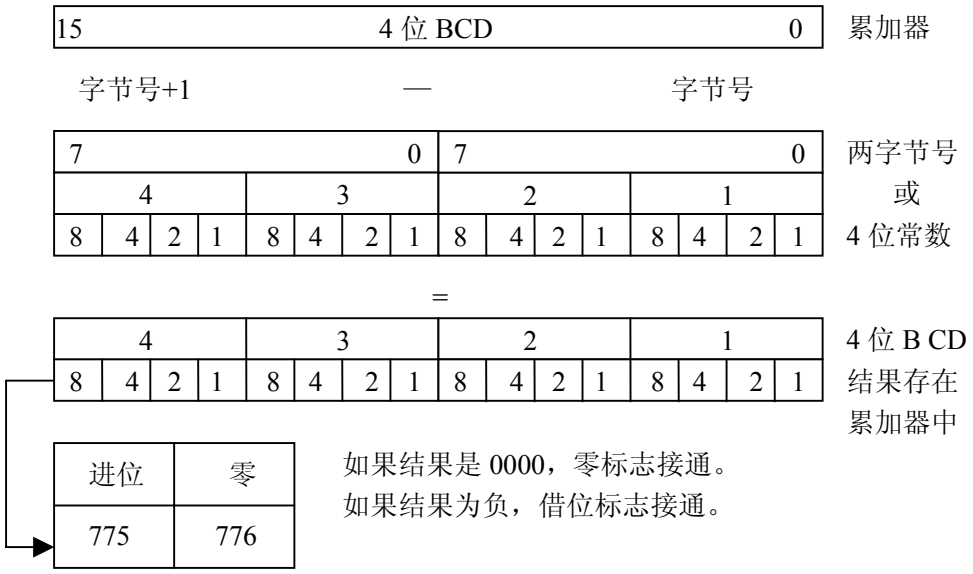
梯　形　图	指 令 符	注　　释
	STR1 F50 R3 F71 R5 F60 R10	●累加器中 4 位 BCD 码来自输入 030—047。 ●累加器的内容与来自输入 050—067 的内容相加。 ●运算结果存贮在累加器中，并写到输出 100—117 中。 ●如果 D.OUT 指定把结果送到数据寄存器或内部线圈中，这样运算结果存在这些单元中以备后用。

减法（—）、4 位 BCD（F72）

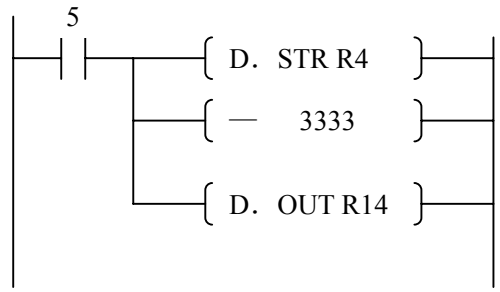
该指令把累加器的内容（BCD 数）减去指定两字节号中的内容（BCD 数）或 4 位 BCD 常数。如果为正它就直接写入累加器。如果是负的，则借位标志（775）接通，差的绝对值被 10000 去减然后结果写入累加器中（补码）。如减的结果是零，零标志（776）接通。

有效字节号与 D.STR（F50）相同

●减法顺序



编程举例

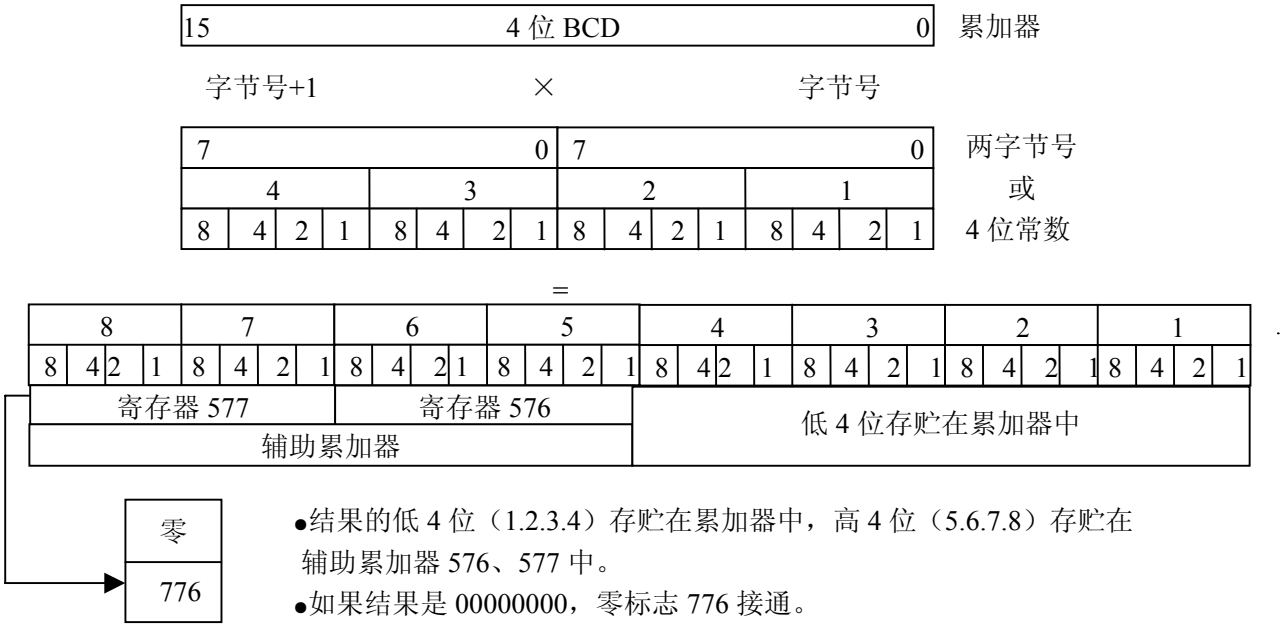
梯 形 图	指 令 符	注 释
	STR 5 F50 R4 F72 3333 F60 R14	●累加器中的数值来自输入 040~057 ●累加器的数值减去常数 3333。 ●运算结果存入累加器并写入 140~157 输出中。

乘法（×）4 位 BCD（F73）

该指令把累加器中内容和指定字节号的内容或一个 4 位 BCD 常数相乘。其运算结果可能是 8 位 BCD 数。低 4 位的结果存贮在累加器中，其 5~8 位存贮在被称为辅助累加器的 576 和 577 号数据寄存器中。如其结果为零，零标志（776）接通。

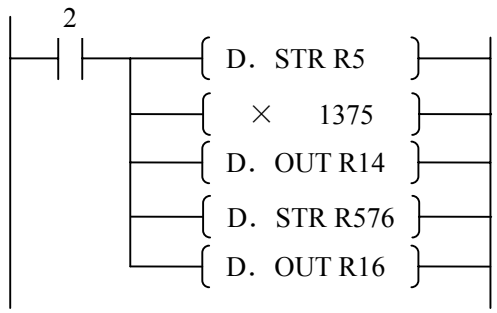
有效字节号与 D.STR(F50)相同

●乘法顺序



●结果的低 4 位（1.2.3.4）存贮在累加器中，高 4 位（5.6.7.8）存贮在辅助累加器 576、577 中。

●如果结果是 00000000，零标志 776 接通。

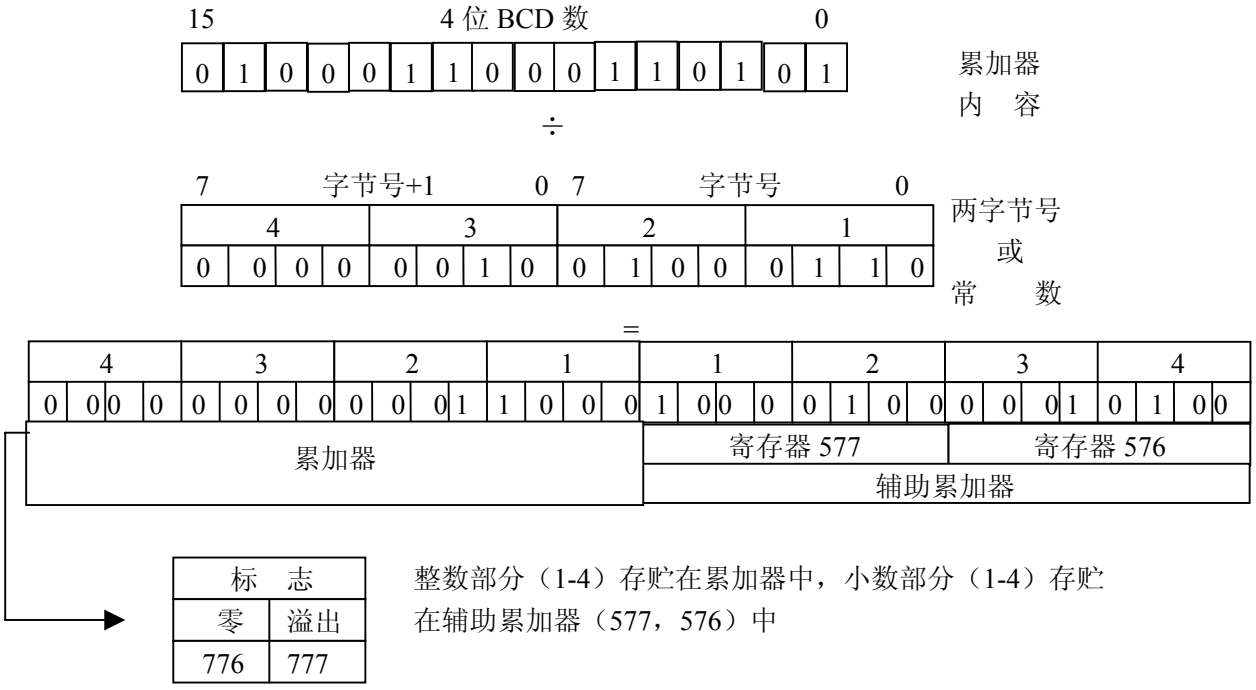
梯 形 图	指 令 符	注 释
4 位数乘 4 位数结果是 8 位数 	STR　2 F50　R5 F73　1375 F60　R14 F50　R576 F60　R16	●累加器中 4 位 BCD 码来自输入 050—067。 ●累加器的内容与常数 1375 相乘。 ●假定运算结果是一个八位BCD数值。其低 4 位存贮在累加器中并传到 140—157 输出中，高 4 位存在辅助累加器 576、577 中并传送到 160~177 输出中。

除法（÷）4 位 BCD（F74）

该指令把累加器的内容除以指定两字节号中的内容或一个 4 位 BCD 常数，其运算结果（商）的整数部分存贮在累加器中，小数部分存贮在辅助累加器（576、577）中，如果除数或被除数为零，零标志（766）接通，如果除数为零，溢出标志（777）也接通。

有效字节号与 D.STR（F50）相同

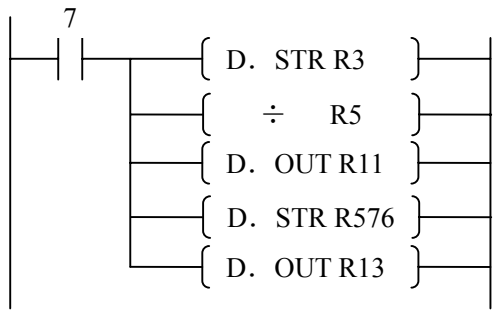
●除法顺序



标 志	
零	溢出
776	777

整数部分（1-4）存贮在累加器中，小数部分（1-4）存贮在辅助累加器（577，576）中

编程举例

梯 形 图	指 令 符	注 释
	STR 7 F50 R3 F74 R5 F60 R11 F50 R576 F60 R13	●累加器内容来自输入 030—047。 ●累加器的内容除以 050—067 的内容。 ●运算结果的整数（4 位）存贮在累加器，并写入到输出 110—127 中。 ●运算结果的小数（4 位）存入辅助累加器（577，576），并写入到输出 130—147 中。

§4—2—4 逻辑处理

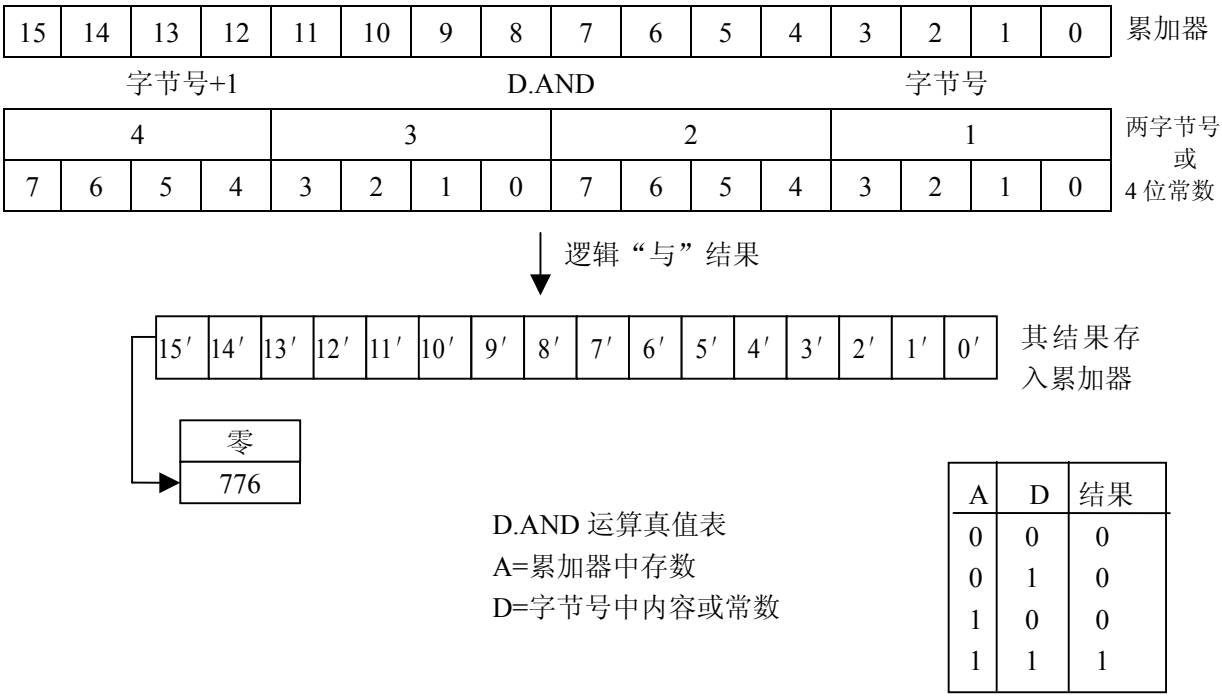
逻辑处理指令包括数据“与”（D.AND）和数据“或”（ D.OR）。

D.AND、（数据与）（F75）

D.AND 指令是把累加器中的内容与指定的两字节号内容或一个 4 位 BCD 常数的相对应每一位分别进行逻辑“与”，其结果存贮在累加器中，如结果为 0000，零标志（766）接通。

有效字节号与 D.STR（F50）相同

D.AND 顺序



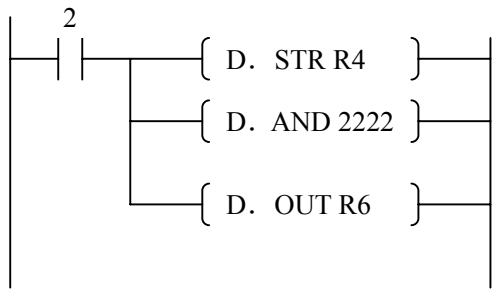
D.AND 运算真值表

A=累加器中存数

D=字节号中内容或常数

A	D	结果
0	0	0
0	1	0
1	0	0
1	1	1

编程举例

梯 形 图	指 令 符	注 释
	STR 4 F50 R4 F75 2222 F60 R6	●假如累加器取自输入 040~057 的常数是 1234。 ●累加器的内容和常数 2222 进行逻辑“与”。 ●结果为 0220 存贮到累加器中并传送到输出 060~077 中。

运算操作过程如下所示：

15 0

0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0

累加器内容

D．AND（F75）

7 0 7 0 字节号内

0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0

容或常数

15 D．OUT（F60） 0

0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0

操作后的累加器内容

D．OR（数据或）（F76）

D．OR 指令是累加器内容与指定的两字节号的内容或一个 4 位 BCD 常数进行逻辑“或”运算。其结果送回累加器中，如结果为 0000，零标志线圈 776 接通。

有效字节号与 D.STR(F50) 相同。

．D．OR 顺序：

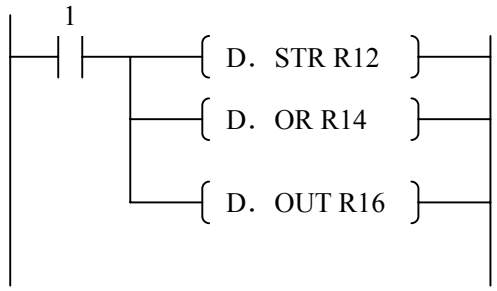
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	累加器
D．OR																
4				3				2				1				两字节号或
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	4 位常数
逻辑“或”↓																
15′	14′	13′	12′	11′	10′	9′	8′	7′	6′	5′	4′	3′	2′	1′	0′	累加器

零
776

●D．OR 操作真实表
●A=累加器中内容
●D=字节号中内容或常

A	D	结果
0	0	0
0	1	1
1	0	1
1	1	1

编程举例

梯 形 图	指 令 符	注 释
	STR 1 F50 R12 F76 R14 F60 R16	●假定把 120~137 输入的内容 3210 送入累加器。 ●输入 140~157 的内容 7531 与累加器内容进行逻辑“或”。 ●运算结果 7731 存入累加器并送到 160~177 输出中。

上述运算操作过程如下所示：

15 0

0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0

累加器内容

D．OR

7 0 7 0 两字节号

0	1	1	1	0	1	0	1	0	0	1	1	0	0	0	1

或 4 位常数

15 D.OUT 0

0	1	1	1	0	1	1	1	0	0	1	1	0	0	0	1

操作后累加器内容

§4—2—5 数据变换

数据变换指令只涉及累加器中的数据，经过处理后的结果还存在累加器中。

右移（F80）

该指令把累加器的内容向右移指定的位数，移动位数可为 1-15，移位后左边的空位以零填入，如果有“1”移出累加器，进位标志（775）接通。如果移位后累加器内容全为零，零标志（776）接通。

●FUN80。右移顺序（以右移 6 位为例）

0	1	1	1	1	0	0	1	0	0	0	1	0	1	0	1

移位前累加器内容右移 6 位

FUN80 6

0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

移位后累加器内容

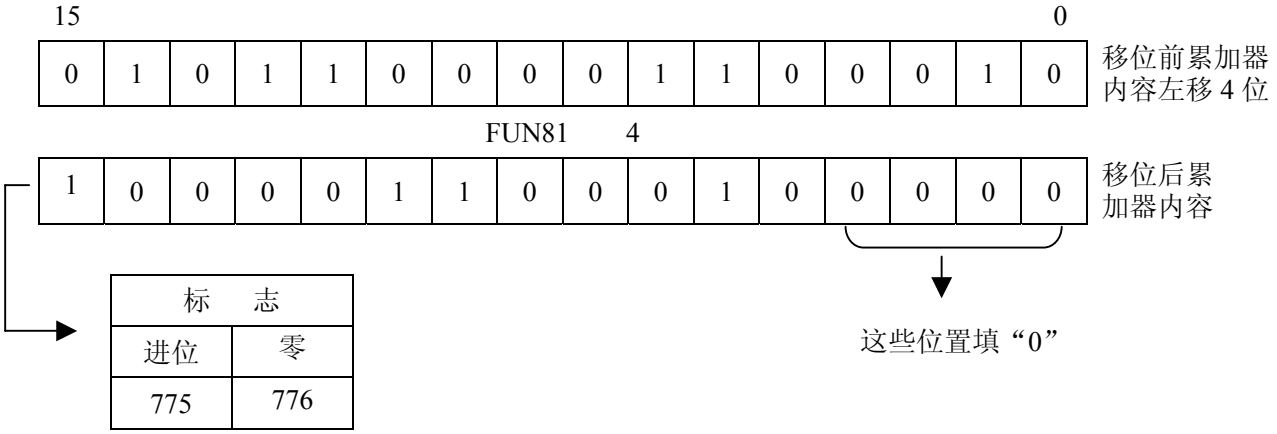
这些位置填“0”

标 志	
进位	零
775	776

左移（F81）

该指令是把累加器中的内容向左移过指定的位数，移动位数可以是 1—15，移位后右边的空位以“0”填入。如果有“1”移出累加器，则进位标志（775）接通。如果移位后累加器内容全为零，零标志（776）接通。

●FUN81。左移顺序（以左移 4 位为例）

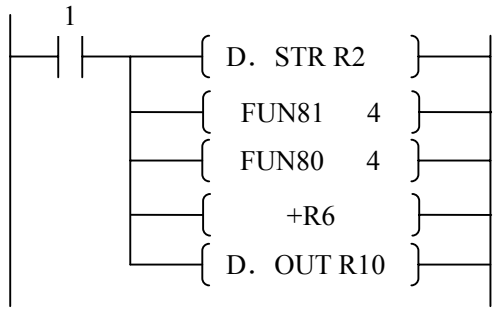


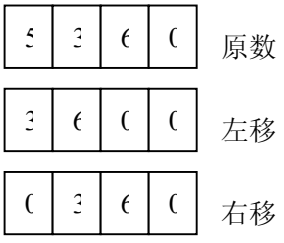
标 志	
进位	零
775	776

F80●F81 编程举例：

该例表示删去存入累加器中不必要的数字的一种方法。这例子是将累加器的 BCD 数的第 4 位内容清零。

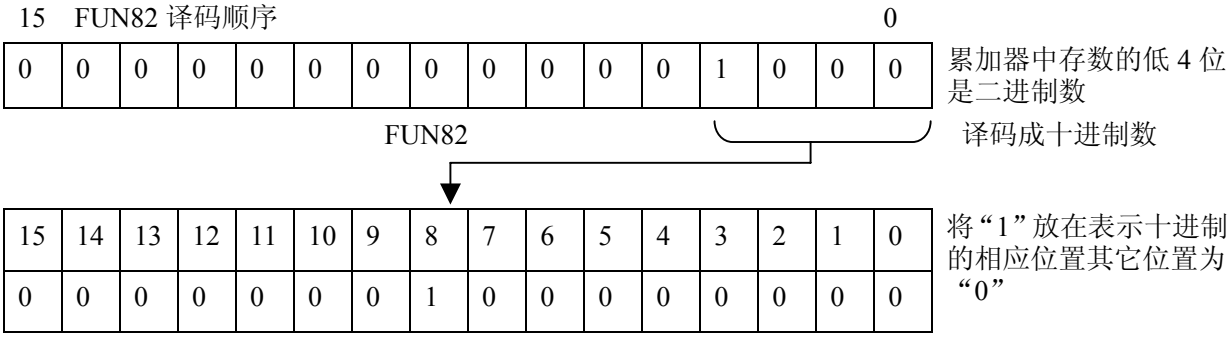
编程举例：

梯 形 图	指 令 符	注 释
	STR 1 F50 R2 F81 4 F80 4 F71 R6 F60 R10	●累加器中存数来自输入 020—037。 ●左移 4 位。 ●右移 4 位第 4 位（5）被删除 ●移位后累加器中内容和输入 060~070 的内容相加。 ●结果存贮在累加器中并写到输出 100~117 中



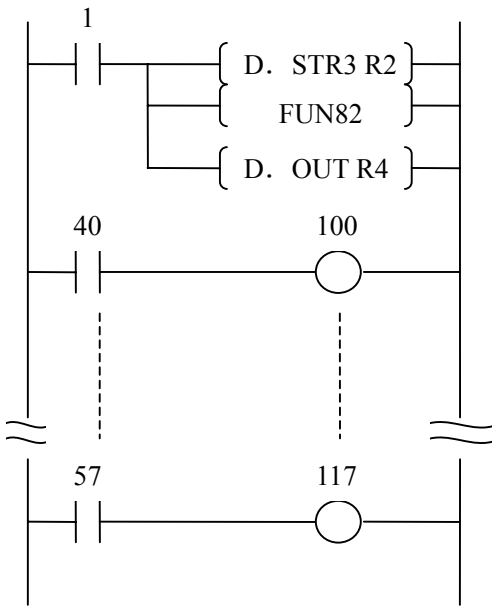
译码（F82）

该指令把累加器中低 4 位的二进制数译成 0—15 的十进制数，在累加器中对应十进制数的某一位置上置“1”。原累加器中高 12 位不进行该操作。



如果累加器中 BCD 值为 8 则译码后累加器 8 号位置“1”其它位置“0”

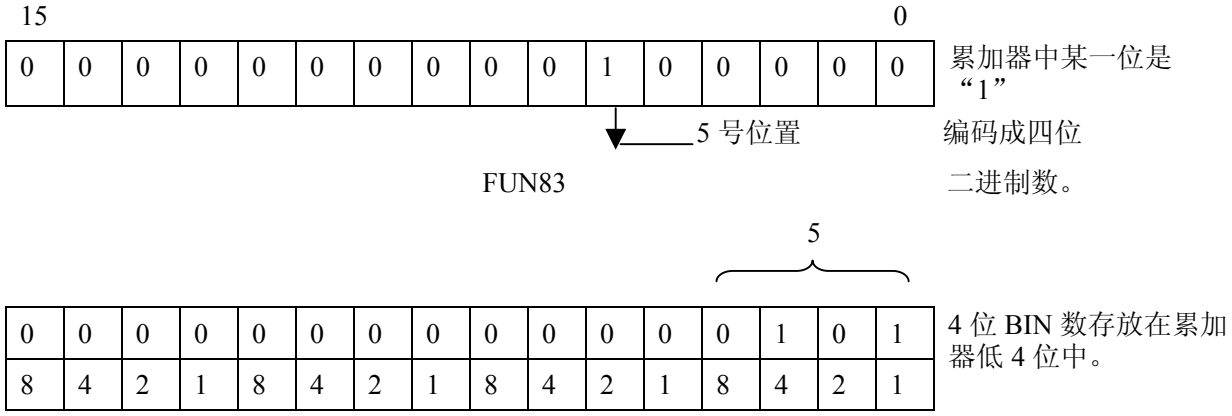
编程举例：

梯 形 图	指 令 符	注 释
	STR 1 F53 R2 F82 F60 R4 STR 40 OUT 100 … … … … STR 57 OUT 117	●把输入 20、21、22、23 组成的 BCD 数装入累加器的低 4 位（例如 6） ●把这 BCD 译成 1~15 的十进制数。 ●把该数存入累加器相应位置 ●输出 040~057 与累加器中的 16 位相对应，当累加器中某一位置“1”输出 100~117 中相应有一线圈接通（输出 106 接通）。

编码（F83）

该指令把累加器中（0~15）置“1”的某一位进行编码，编码成 4 位二进制的数放在累加器的低 4 位中。在 F83 执行前累加器如果有 2 位以上为 ON 或没有接通的位，则溢出标志（777）接通。

●FUN83 编码顺序

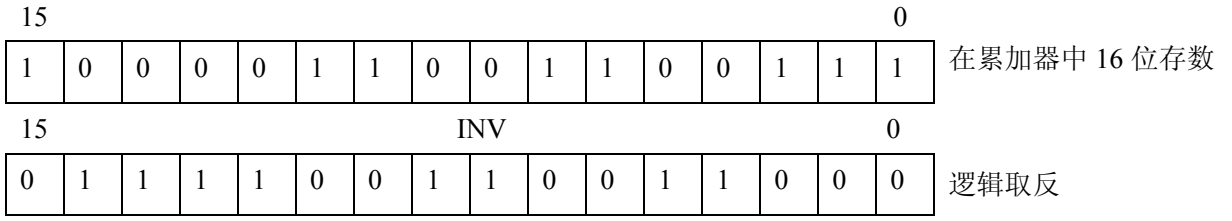


FUN83 指令执行后累加器高 12 位均为零。

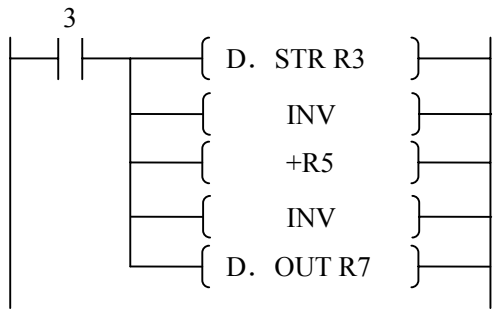
INV（取反）（F84）

该指令把累加器中 16 位内容取反如 1→0，0→1

取反顺序



编程举例：

梯 形 图	指 令 符	注 释
	STR 3 F50 R3 F84 F71 R5 F84 F60 R7	●把输入 030——047 内容存入累加器并取反。 ●取反后的数据再和输入 050~067 的内容相加，其结果存贮在累加器中，再把该内容取反。 ●取反结果写到输出 070~107 中。

BIN（BCD 转换成 BIN）（F85）

该指令把累加器中 BCD 数据转换成相应二进制数（BIN）

二进制转换顺序

5				4				2				3			
0	1	0	1	0	1	0	0	0	0	1	0	0	0	1	1

4 位 BCD 数存贮在累加器中。

BCD

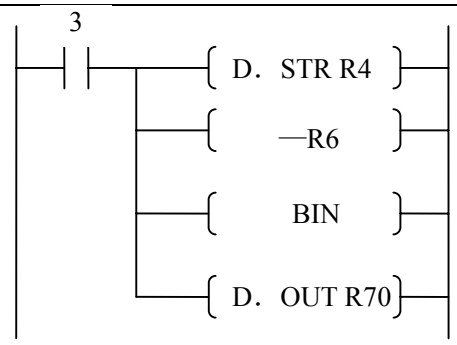
↓

BIN

0	0	0	1	0	1	0	1	0	0	1	0	1	1	1	1

转换成二进制数

编程举例：

梯 形 图	指 令 符	注 释
	STR 3 F50 R4 F72 R6 F85 F60 R70	●输入 040——057 内容存贮在累加器中。 ●累加器的内容减去输入 060—077 的内容，其结果存贮在累加器中。 ●把累加器中二进制数写到输出 070—107 中。

BCD（BIN 转换成 BCD）（F86）

该指令把累加器中的二进制码转换成 BCD 数，如果 BCD 数＞9999，则溢出标志 777 接通。

BCD 转换顺序

0	0	1	0	0	1	0	0	0	0	1	1	0	1	1	1

存贮在累加器中的二进制数

BIN

↓

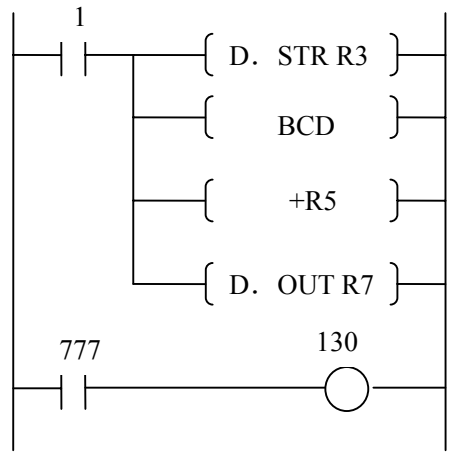
BCD

9				2				7				1			
1	0	0	1	0	0	1	0	0	1	1	1	0	0	0	1

转换成一个 4 位数 BCD 数

溢 出
777

编程举例：

梯 形 图	指 令 符	注 释
	STR 1 F50 R3 F86 F71 R5 F60 R7 STR 777 OUT 130	●输入 030—047 的内容（二进制数）存入累加器。并转换为一个 4 位 BCD 数。 ●累加器内容与输入 050—067 的内容（BCD 数）相加，结果存入累加器。 ●把累加器中的 BCD 数写到输出 070—107 中。 ●如出现溢出现象，溢出标志 777 接通 130 输出。

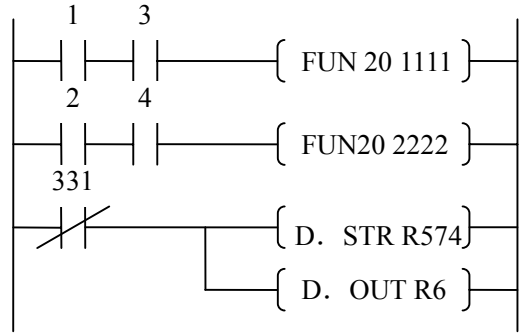
§4—2—6 外部故障诊断

这里讲的故障诊断是对受 PLC 控制的输入设备，控制负载的故障进行检测、显示、报警。在实际应用场合，外部设备出现不该产生的情况是难免的，如：输入 1 和输入 2 不该同时接通而实际同时接通了，那么这是一种故障。检测故障的方法就是当 PLC 接受到不应有的信号即故障信号时，显示一个用户事先编制的代表这一故障的代号（四位十进制数），以告知操作者故障的原因（性质和发生点）。这里用的编程指令是 FUN20（功能 20）

当发生故障时，PLC 执行 F20 功能，将 F20 指令中的四位 BCD 数在编程器上显示出来。同时为使用户能用外接的数据显示装置能显示此编号，PLC 同时将此编号送入了 574 和 575 数据寄存器中，并且使 771 号内部线圈接通。

在一个用户程序中，可以多次使用 F20 指令，以检测多种故障。每个监测点（每个 F20 指令中），赋予不同的四位常数作为故障编号，当出现一个故障时，显示其相应的编号。如果同时出现多种故障，那么，显示这几种故障中在程序中检测点靠后的那一个故障的编号，并将此编号存贮在 574、575 寄存器中。在编程器上显示的故障编号，可以按 CLR 键将之清除。

编程举例：

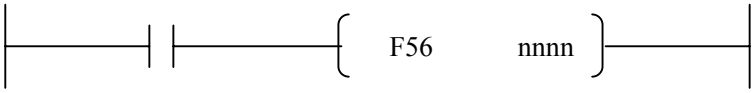
梯 形 图	指 令 符	注 释
	STR 1 AND 3 F20 1111 STR 2 AND 4 F20 2222 STR NOT 331 F50 R574 F60 R6	●当输入 1 和 3 同时接通时，编程器上显示 1111，同时此数存入 574，575 寄存器中。 ●如输入 2 和 4 同时接通，编程器上显示 2222，同时此数存入 574，575 号寄存器中。 ●将 574、575 的内容读入累加器后，送到 060~077 输出上。 ●故障排除后，按 CLR 键可清除显示。

§4—2—7 SR—21/22 数据缓冲器传送指令

F56 指令

1、F56 nnnn

功能：将编号为 nnnn 的 RAM 数据缓冲器内容读到 PLC 累加器中。如果 nnnn 超过 RAM 数据缓冲器的最大序号，则将 0 送入 PLC 累加器中。



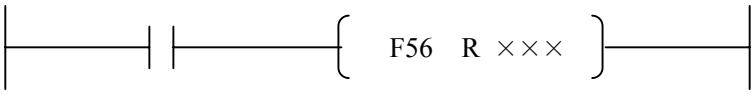
nnnn：0—RAM 数据缓冲器的最大序号

指令执行：条件成立时执行，否则不执行。

1、F56 Rxxx

功能：将数据寄存器 Rxxx 及 Rxxx+1 中的 BCD 数作为 RAM 数据缓冲器的编号，然后把该编号内的数据读到 PLC 累加器中。

如果 Rxxx 及 Rxxx+1 中的数据不是 BCD 数，或者该 BCD 数超过 RAM 数据缓冲器的长度，则将 0 送入 PLC 累加器中。



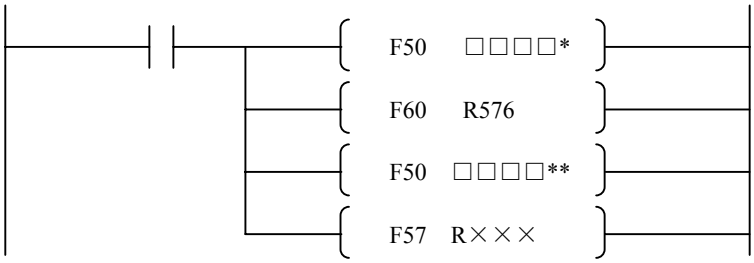
Rxxx

有效字节号与 D.OUT（F60）相同。

命令执行：条件成立时执行，否则不执行。

F57 Rxxx 指令

功能：将 RAM 数据缓冲器的内容传送到数据寄存器（块传送），RAM 数据缓冲器的开始编号在累加器中；传送字数在辅助累加器 R576 中。

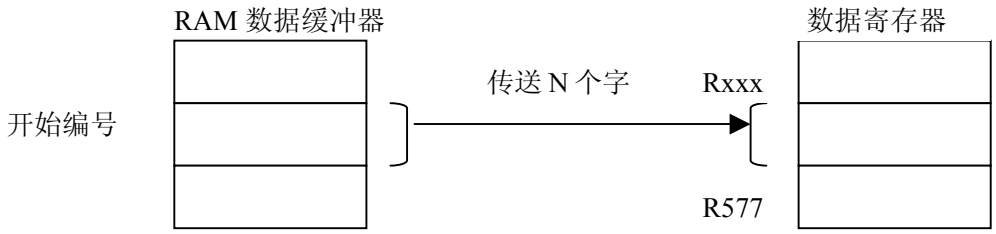


□□□□* ：给定传送字数 N：0＜N＜64；当 N≥64 时，以 64 取余。

□□□□**：给定 RAM 数据缓冲器的开始编号 NO：0≤NO≤RAM 缓冲器的最大序号

Rxxx ：为数据寄存器 R400—R576

传送示意图如下：



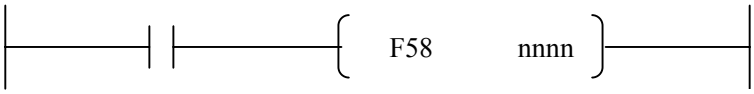
指令执行：条件成立时执行。

F58 指令

1、F58 nnnn

功能：将编号为 nnnn 的 ROM 数据缓冲器内容读到 PLC 累加器中。

如果 nnnn 超过 ROM 数据缓冲器的长度，则将 0 送入 PLC 累加器中。



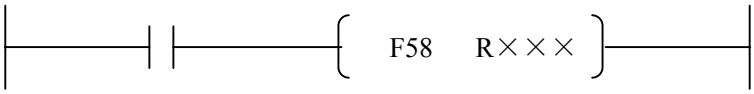
nnnn 的范围：0—ROM 数据缓冲器的长度

指令执行：条件成立时执行。

2、F58 Rxxx

功能：将数据寄存器 Rxxx 及 Rxxx+1 中的 BCD 数作为 ROM 数据缓冲器的编号，然后把该编号内的数据读到 PLC 累加器中。

如果 Rxxx 及 Rxxx+1 中的数据不是 BCD 数或者该 BCD 数超过 ROM 数据缓冲器的长度，则将 0 送入 PLC 累加器中。



R×××

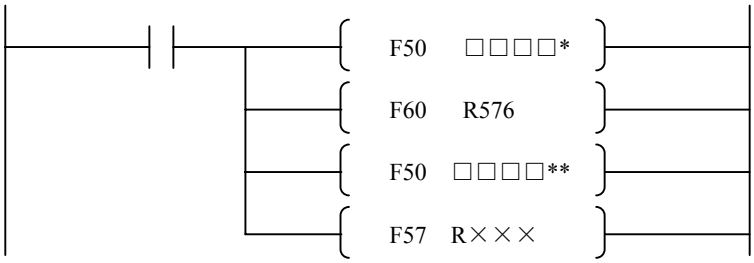
有效字节号与 D.OUT（F60）相同。

指令执行：条件成立时执行

F59 R×××指令

功能：将 RAM 数据缓冲器的内容传送到数据寄存器中（块传送）

RAM 数据缓冲器的开始编号在累加器中；传送字数在辅助累加器 R576 中。



□□□□* ：给定传送字数 N：0＜N＜64；当 N≥64 时，以 64 取余。

□□□□**：给定 RAM 数据缓冲器的开始编号 NO：0≤NO≤RAM 缓冲器的最大序号

Rxxx ：为数据寄存器 R400—R576

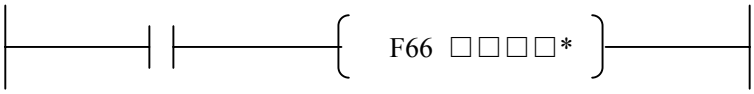
注：如从数据寄存器 Rxxx 开始存不下 N 个字，则传送数据寄存器 R577 为止

指令执行，条件成立时执行。

F66 指令

1、F66 nnnn

功能：将累加器中内容写入编号为 nnnn 的 RAM 数据缓冲器中。

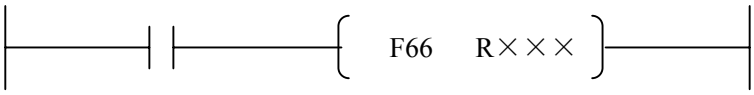


□□□□* ：给定 RAM 数据缓冲器的编号 NO：20≤NO≤RAM 缓冲器最大序号

指令执行：条件成立时执行。

2、F66 Rxxx

功能：以 Rxxx 及 Rxxx+1 中的 BCD 数为 RAM 数据缓冲器的编号，将累加器中的内容写入该 RAM 数据缓冲器内。



R×××：

有效字节号与 D.OUT（F60）相同。

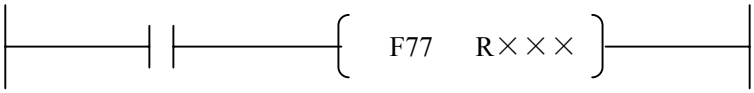
Rxxx 中的 BCD 数范围为：20—RAM 缓冲器最大序号

指令执行：条件成立时执行

§4—2—8 SR—21/22 其它指令

F77 Rxxx 指令

功能：数据寄存器内容加 1



R×××：

有效字节号与 D.OUT（F60）相同。

指令执行：条件成立时执行

指令执行后，不影响累加器的内容，但影响标志位 775、776、777

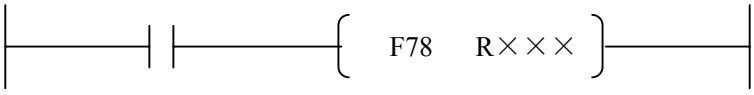
775：加 1 以产生溢出；

776：加 1 以后结果为 0；

777：Rxxx 中的数据不是 BCD 数；

F78 Rxxx 指令

功能：数据寄存器内容减 1



R×××：

有效字节号与 D.OUT(F60)相同

指令执行：条件成立时执行。

指令执行后，不影响累加器的内容，但影响标志位 775、776、777

775：减 1 以后产生借位；

776：减 1 以后结果为 0；

777：Rxxx 中的数据不是 BCD 数；

F25 Rxxx

该指令是为设定显示单元 C—02DS 提供的，详细说明参阅 C—02DS 用户手册。

智能模块控制命令

F29—F42 命令是支持智能模块 E—02PP 等用的，关于 F29—F42 指令的详细说明请参阅《E—02PP 轴定位模块使用说明书》。

§4—2—9 SE—22 追加指令

一、级式指令

SG S□□□□

指令功能键：F10

说 明：（1）级登记指令

SG 是规定在此指令后面所属的编号登记指令。

（2）指定的级的状态为 ON 时，执行该级所属的指令，OFF 则不执行。级的状态从 ON 变为 OFF 时，该级所属的 OUT，TMR 指令等 OFF 或复位。

（3）以下条件使级的状态为 ON

JMP/NJMP S××××指令，SET S××××指令执行时，这些指令指定的级变为 ON。

（4）以下条件使级的状态为 OFF

JMP/NJMP S××××指令执行时所属的该级变为 OFF。

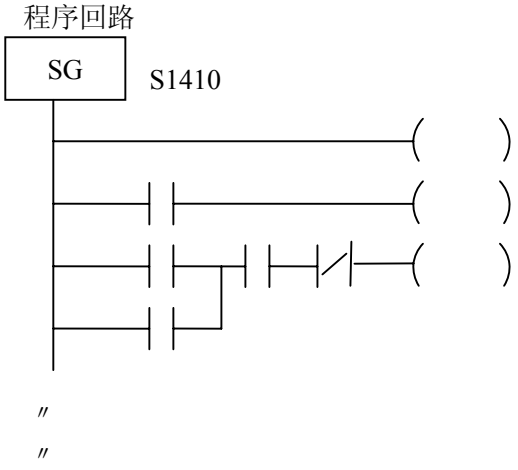
RST S××××指令执行时，指定的级为 OFF。

（5）级号可以任意分配，但同一级号不能被重复登记。

级的范围：1400—1577

（6）根据停电记忆参数的设定，断电时级的 ON/OFF 状态可记忆。

（7）回路例：



通常先写无条件处理部分，再写带条件处理部分。

ISG S□□□□

指令功能键：F11

说 明：（1）初始级登记指令

ISG 是规定在此指令后面所属的初始级的编号登记指令。

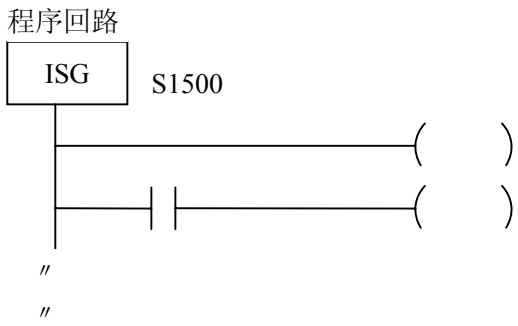
电源投入或开始 RUN 时，级的状态为 ON。

程序执行的关系及功能同 SG 相同。

（2）级号可以任意分配，但同一组号不能被重复登记，也不能与 SG 登记的级号重复，级的范围：1400—1577

（3）根据停电记忆参数的设定，断电时级的 ON/OFF 状态可记忆。

（4）回路例：



助记符 键操作

ISG S1500 F11 1500

〃 〃

CV S□□□□

指令功能键：F14

说 明：（1）登记同时顺序合流的级号的指令。

被合流的级有必要用 CV 指令进行连续登记，由 CV 指令登记的称为合流级。

当所有被合流指定的级的状态全为 ON 时，执行最后一个 CV 指令后的指令。当 CVJMP 指令执行了向其它级的转移指令扣，合流级群的状态都置为 OFF。

（2）CV 指令之后，必须用 CVJMP 指令。

（3）从第一个 CV 到最后一个 CV 之间不能使用其它指令。

（4）CV 指令一次最大可登记 16 个级。

（5）回路例见 CVJMP 指令。

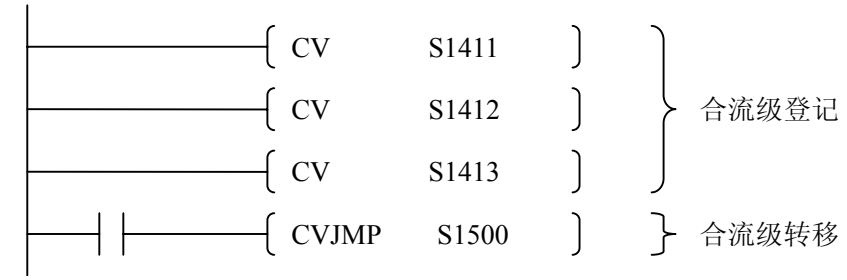
CVJMP S□□□□

指令功能键：F15

说 明：（1）合流级群成立后转移至指定的级。

（2）合流级群所有级的状态为 ON 时，如果没有转移条件，则执行 CVJMP 指令，如果 CV 指令后有转移条件则转移条件成立时，执行 CVJMP 指令指定的级的状态为 ON，然后将合流级群所指定的级状态全为 OFF。

（3）回路例：



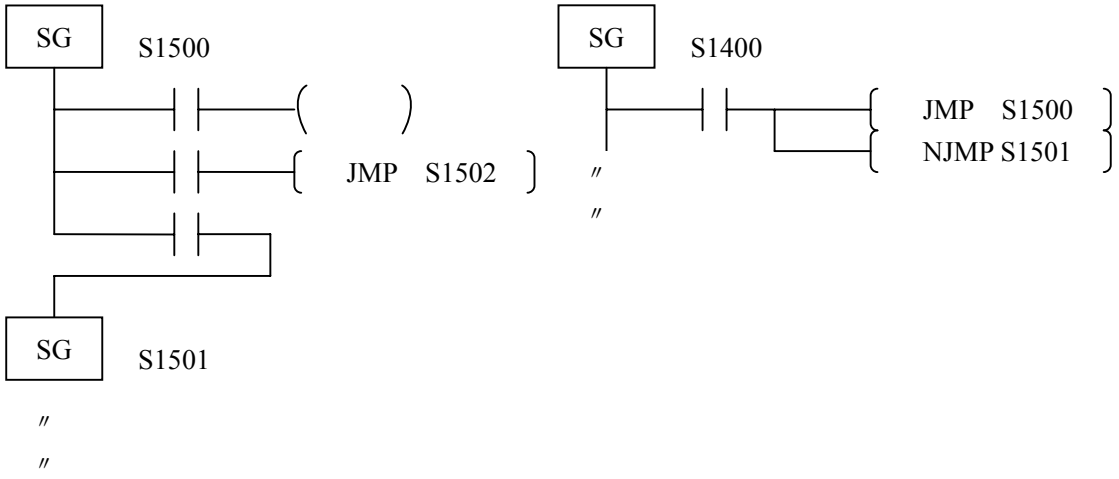
JMP S□□□□

指令功能键：F12

说　　明：（1）所属级的状态为 ON，在跳转条件成立时向指定的级跳转。

（2）该指令执行后，所属的级复位，该指令指定的级接通。

（3）回路例：



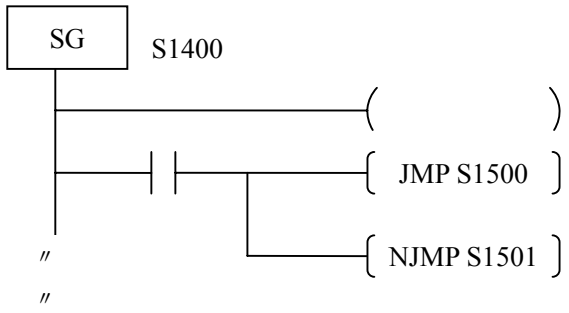
NJMP S□□□□

指令功能键：F13

说　　明：（1）所属级的状态为 ON，在跳转条件不成立时向指定的级跳转。

（2）该指令执行后，所属的级复位，该指令指定的级接通。

（3）回路例：



二、新增数据处理指令

寄存器间址写命令（F26）

说　　明（1）将 ACC 中的 16 位数据传送到 P 寄存器，（P 为 16 位数据寄存器）。

P 寄存器号由该指令后的操作数据寄存器和下一个寄存器的内容间接指定。

（2）F26 R××

（3）16 位数据寄存器 P 的范围是 000—777，八进制。

操作数指定的数据寄存器和下一个数据寄存器的内容超出 000—777 的范围，该指令不动作。

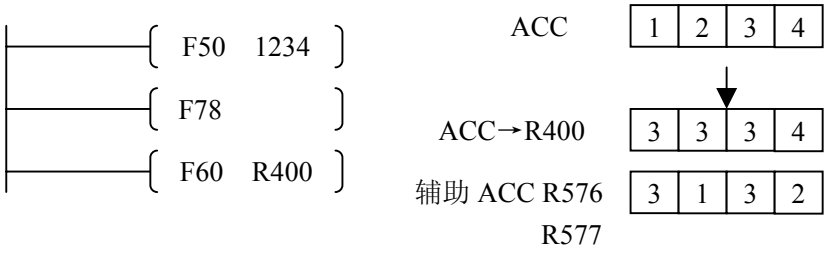
寄存器间址读指令（F27）

说　　明：（1）将 P 寄存器中的 16 位数据传送到 ACC，（P 为 16 位数据寄存器）。
P 寄存器号由该指令后的操作数据寄存器和下一个寄存器的内容间接指定。
（2）F27 R××

（3）16 位数据寄存器 P 的范围是 000—777，八进制。
操作数指定的数据寄存器和下一个数据寄存器的内容超出 000—777 的范围，该指令不动作。

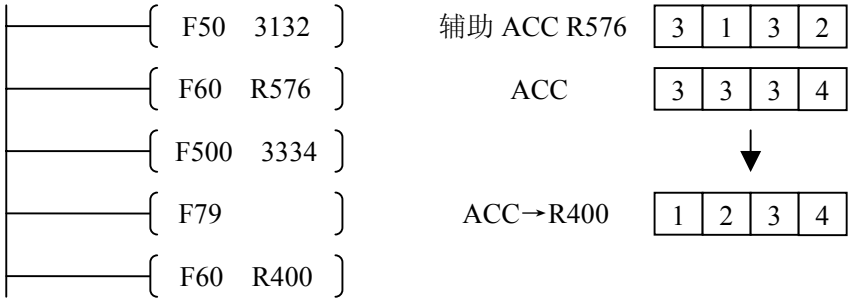
HEX → ASCII 转换指令（F78）

说　　明：（1）HTA：将 ACC 中的十六进制数转换成 ASCII 码→辅助 ACC，ACC。
（2）十六进制数为 0—9，A—F 的范围，执行该指令后被转换成相应的 ASCII 码为 30—39H，41—46H ，（H 表示十六进制数）。
（3）该指令是按每 4 bits 二进制位进行转换，不会出现数据超出范围。
（4）回路例：



ASCII → HEX 转换指令（F79）

说　　明（1）ATH：将辅助 ACC，ACC 中的 ASCII 码转换成十六进制数→ACC。
（2）ASCII 码在 30—39H，41—46H 的范围，执行该指令后被转换成相应的十六进制数为 0—9，A—F，（H 表示十六进制数）。
（3）ACC 或辅助 ACC 中的数据超出范围后的处理，ACC 中的内容不确定。
（4）回路例：



SEG 变换指令（F87）

说　　明（1）SEG 指令，将 ACC 中的下 8 位数据变为 7 段代码显示用数据→ACC。

（2）下表所列出的是在执行 SEG 指令后的代码。

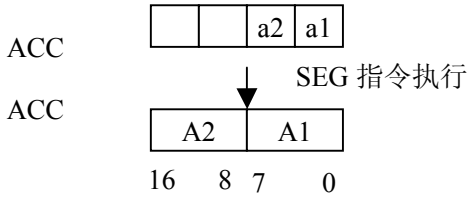
（3）使用举例：

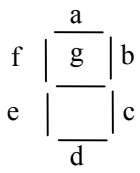
在使用 C—02DS 显示控制单元时，自定义显示的段码，根据需要显示字符，将转换后的十六进制代码送到特殊定义的数据寄存器。

ACC 数据	7 段 码								HEX 代码
	h	g	F	e	d	c	b	a	
0	0	0	1	1	1	1	1	1	3F
1	0	0	0	0	0	1	1	0	06
2	0	1	0	1	1	0	1	1	5B
3	0	1	0	0	1	1	1	1	4F
4	0	1	1	0	0	1	1	0	66
5	0	1	1	0	1	1	0	1	6D
6	0	1	1	1	1	1	0	1	7D
7	0	0	1	0	0	1	1	1	27
8	0	1	1	1	1	1	1	1	7F
9	0	1	1	0	1	1	1	1	6F
A	0	1	1	1	0	1	1	1	77
B	0	1	1	1	1	1	0	0	7C
C	0	0	1	1	1	0	0	1	39
D	0	1	0	1	1	1	1	0	5E
E	0	1	1	1	1	0	0	1	79
F	0	1	1	1	0	0	0	1	71

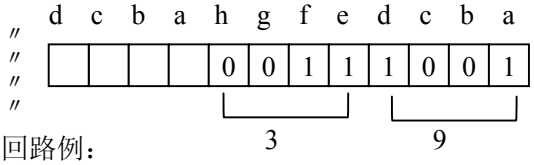
变换前：ACC 的下 8 位=a2 a1
是 2 位十六进制数据

变换后：ACC 为相应的代码

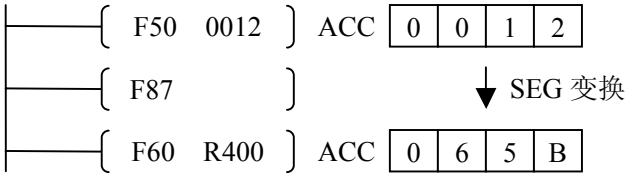




例如：ACC 中原始十六进制数为 a1=C
转换后 ACC 如下



回路例：



IORF 指定范围的 I/O 传送指令（F88）

说　　明：（1）指定范围的 I/O 定义号立即传送。

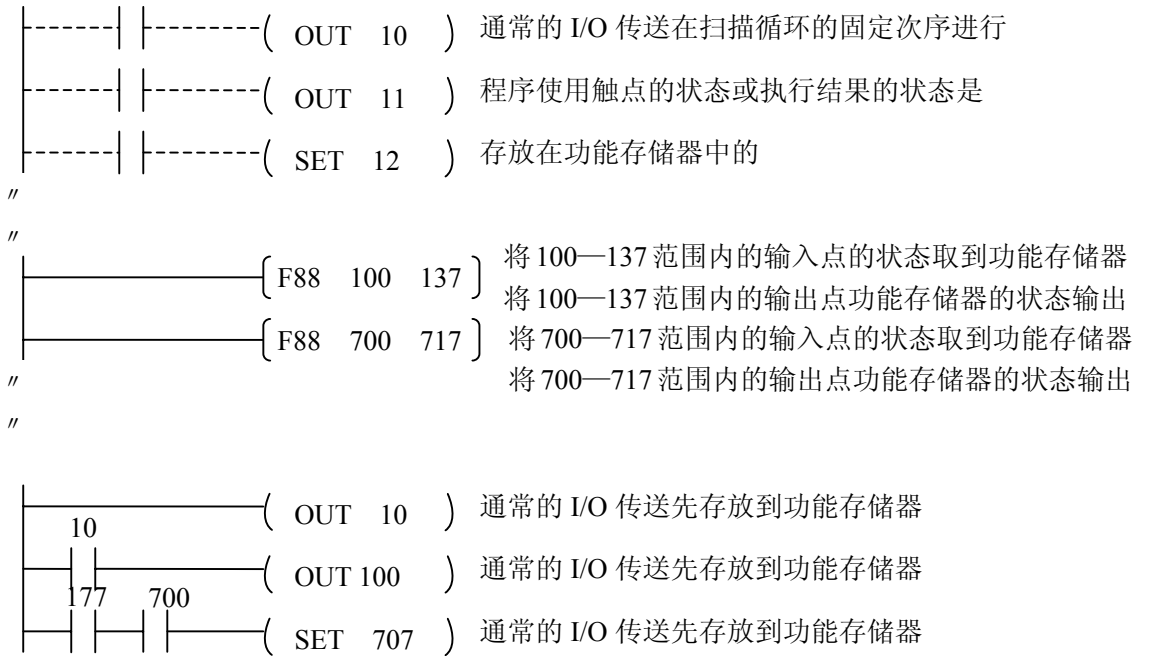
（2）在此指令执行时，对指令指定的范围内的 I/O 定义号进行一次立即传送。

输入点，将外部输入模块的状态重新读入功能存储器。

输出点，将功能存储器的状态立即传送到外部输出模块。

（3）通常的 I/O 传送是在 I/O 扫描期间，一次读取或传送，程序用到时从功能存储器读取状态或改变状态。遇到 IORF 指令时，该指令在功能存储器与外部模块再次进行状态交换。

（4）回路例：



定时扫描程序开始/结束的指定（F89）

说　　明：（1）定义定时扫描程序的开始和结束。F89 应成对使用。

（2）定时扫描功能的设定，在系统参数 000 中设定毫秒数。

（3）特殊继电器 SP1002 的设定，当 SP1002 为 ON 时，定时扫描禁止；当 SP1002 为 OFF 时，定时扫描许可。

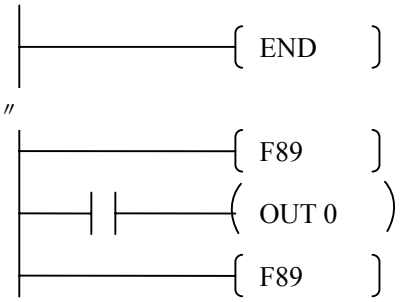
（4）定时扫描子程序，在 SP1002 为 OFF，定时设定时间到，执行定时扫描子程序一次，然后返回到原来的主程序继续执行，下一次定时设定时间到，再次执行。

（5）定时扫描子程序不宜过长，否则会影响主程序的执行和响应。

（6）对定时扫描子程序内所使用的指令有所限制，以下指令不能使用。

STR（NOT） 600—677　　SET OUT RST
AND（NOT）600—677　　TMR/CNT
OR （NOT）600—677　　SR
MCS/MCR　　数据操作指令
SET OUT

（7）回路例：



注释指令（F93）

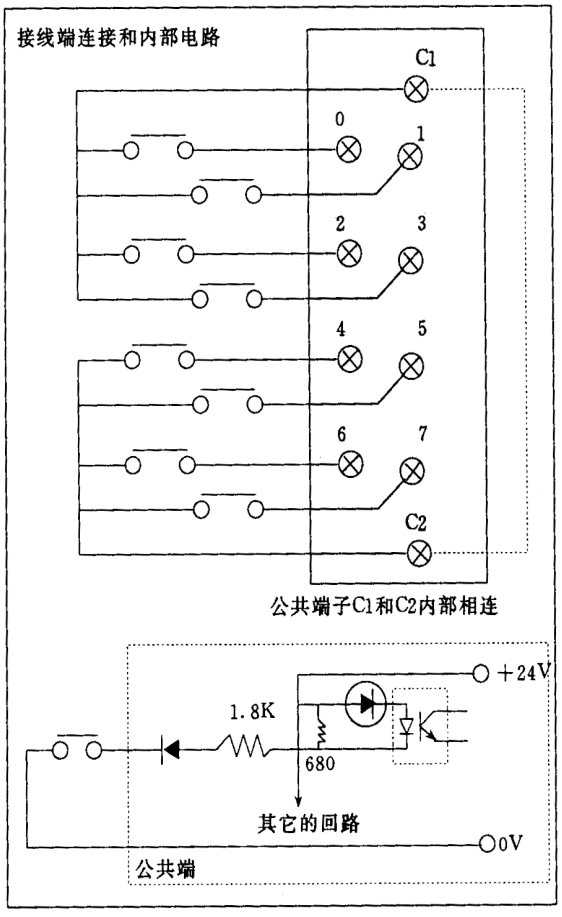
说　　明：（1）在程序中仅起注释作用的空操作操作，不引起任何动作，在搜索时起标志作用。

第五章 I/O 模块及接线

§5—1 输入模块

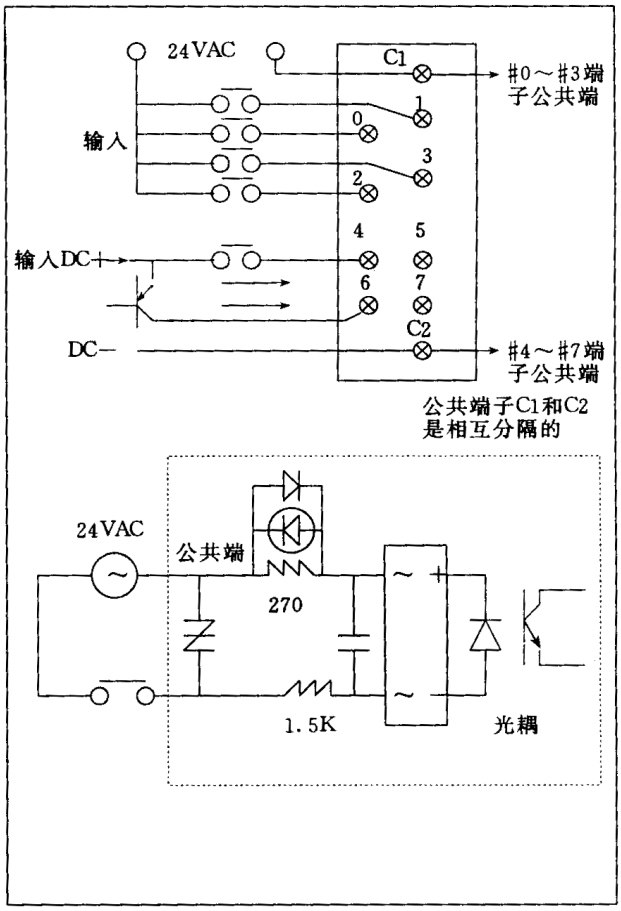
24VDC 输入模块（8 点）E—01N

项 目	特 性
回路数	8 点
适合的输入信号源	ON/OFF 信号来自无电压的触点或开路集电极三极管 耐压：DC24V 开闭容量：17mA 以上
ON 电流	＞7mA
OFF 电流	＜3mA
ON 电压	＜3V
OFF 电压	随供电电压和输入数变化
输入端开路电压	18—36VDC
OFF→ON 响应	4—15ms
ON→OFF 响应	4—15ms
内部 24V 消耗电流	14 mA /1 回路
内部 9V 消耗电流	10 mA /1 模块
重量	120g



24VAC/DC 源输入模块（8 点） E—02N

项 目	特 性	
回路数	8 点	
输入信号	AC 输入电压	DC 输入电压
	20~28VAC（50/60Hz）	20~28VDC
ON 电流	＞10mA	
OFF 电流	＜2mA	
ON 电压	＞20V	
OFF 电压	＜6V	
OFF→ON 响应	5—50 ms	6—30 ms
ON→OFF 响应	5—60 ms	5—60 ms
输入电流	通常 13 mA （24V/50Hz） 最大 19 mA	通常 13 mA 最大 19 mA
绝缘耐压	2KVAC 1 分钟	
内部 9V 消耗电充	10 mA	
重量	120g	

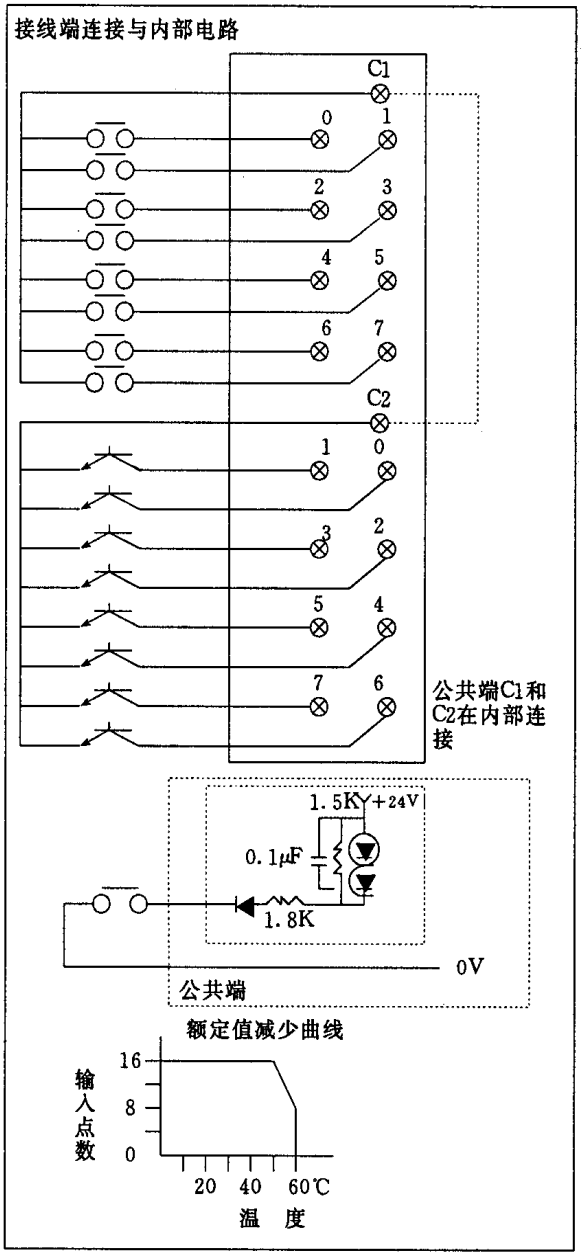


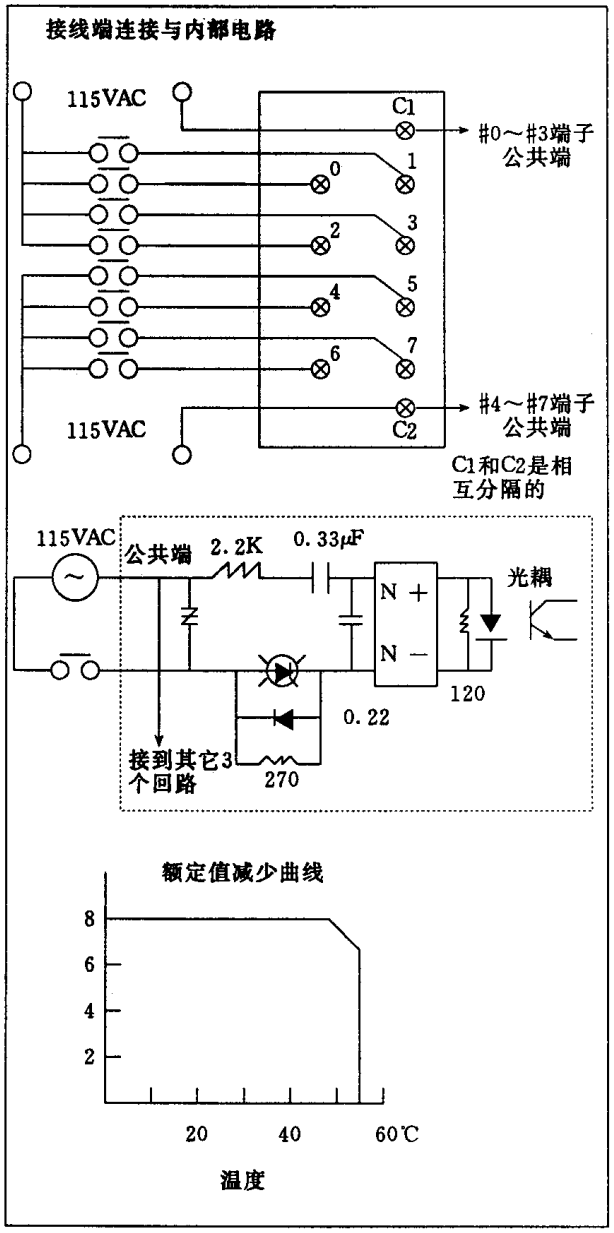
24VDC 输入模块（16 点） E—05N

项 目	特 性
回路数	16 点
ON 电流	＞15mA
OFF 塌流	＜1mA
ON 电压	＜13V（24V），7V（18V） 25V（36V）
OFF 电压	＜19V（24V），13V（19V） 31V（36V）
输入端开路电压	18—36VDC
OFF→ON 响应	3—15ms
ON→OFF 响应	4—15ms
内部 24V 消耗电流	14mA/1 回路
内部 9V 消耗电流	25mA/1 模块
重量	180g

100VAC 输入模块（8 点） E—20N

项 目	特 性
回路数	8 点
输入电压	100VAC（允许变动范围 85—132VAC）50/60Hz
输入电流	11 mA（100v，50Hz）， 13 mA（100v，60Hz）
输入阻抗	10KΩ
ON 电压	＞80VAC
OFF 电压	＜20VAC
OFF→ON 响应	10—30ms
ON→OFF 响应	10—60ms
内部 9V 消耗电流	10mA
重量	140g



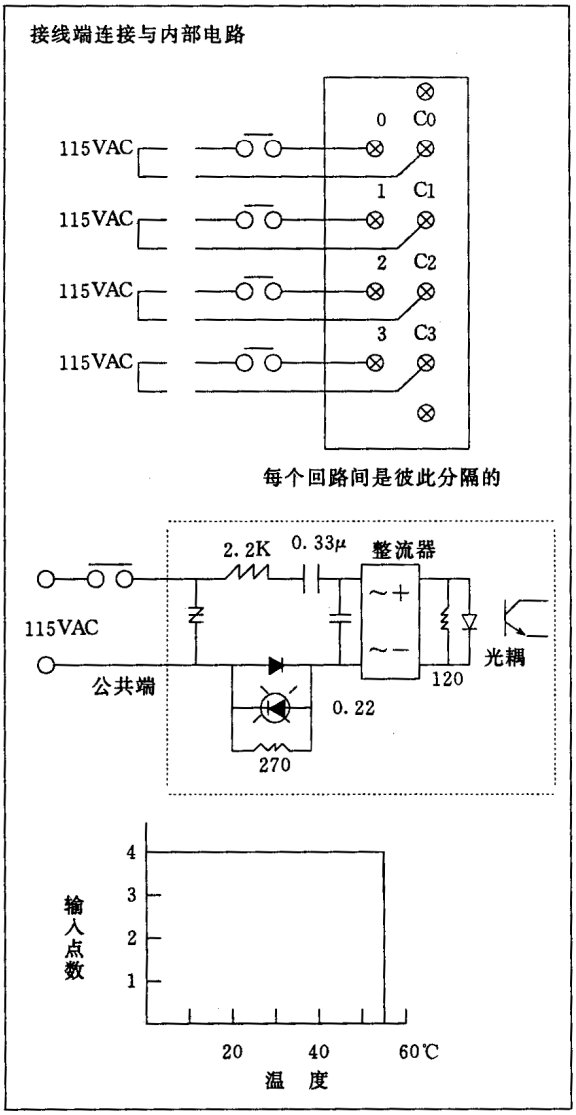


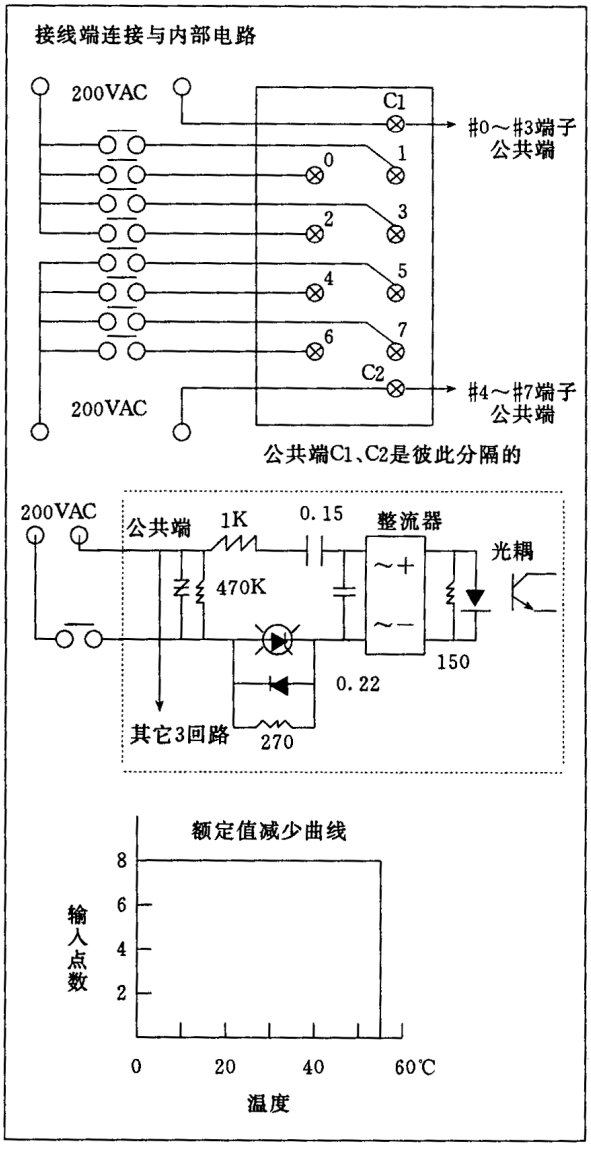
100VAC 输入模块（4 点分隔式） E—21N

项　　目	特　　性
回路数	4 点（彼此分隔）
输入电压	100VAC（允许变动范围 80—132VAC）50/60Hz
输入电流	8—13.5mA(80—132V,50Hz)
	9.5—16mA(80V—132V, 60Hz)
ON 电压	＞80VAC
OFF 电压	＜20VAC
OFF→ON 响应	10—30ms
ON→OFF 响应	10—60ms
内部 9V 消耗电流	10mA
重量	120g

200/220VAC 输入模块(8 点)　E22N

项　　目	特　　性
回路数	8 点
输入电压	200/220VAC(允许变动范围 180—265VAC，50/60Hz)
输入电流	11mA（230v，50Hz），最大 18 mA
输入阻抗	18KΩ（60Hz）
ON 电压	＞180V
OFF 电压	＜40V
OFF 电流	2mA 以内
OFF→ON 响应	5—50ms
ON→OFF 响应	5—60ms
内部 9V 消耗电流	10mA
重量	140g



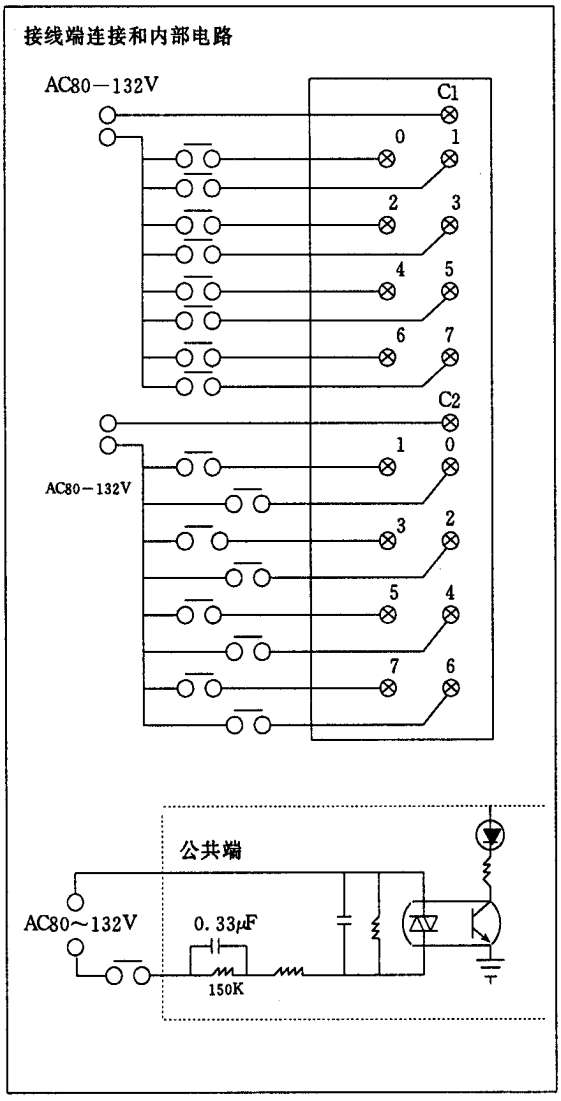


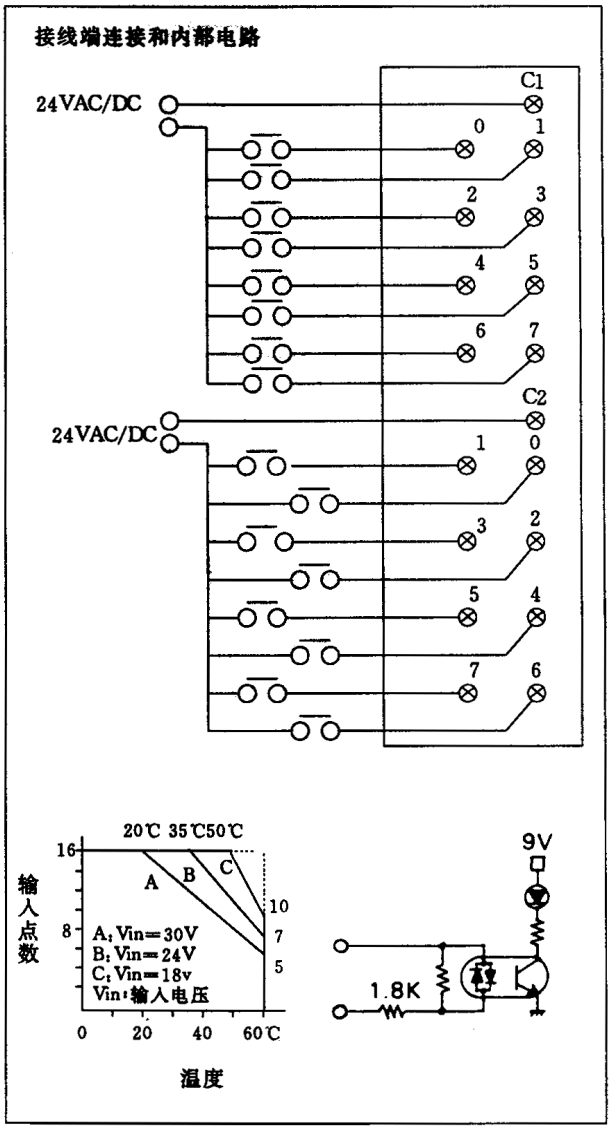
100VAC 输入模块（16 点） E—25N

项 目	特 性
回路数	16 点
输入电压	AC100/115V（允许电压变动范围 AC85 — 132V ）50/60Hz
输入电流	10mA（100V，50Hz）最大 25 mA
输入阻抗	8 KΩ（60Hz）
ON 电压	80V 以上
OFF 电压	15V 以下
ON 电流	8mA
OFF 电流	1.5 mA
OFF→ON 响应	5—50ms
ON→OFF 响应	5—60ms
内部 9V 消耗电流	10mA
重量	140g

24VAC/DC 源输入模块（16 点） E—55N

项 目	特 性	
回路数	16 点，8 回路一个公共端	
输入信号	DC 电压输入	AC 电压输入
额定输入电压	14—30VDC	14—30VAC 50/60Hz
ON 电流	7mA 以上	
OFF 电流	2mA 以下	
ON 电压	14V 以上	
OFF 电压	3V 以下	
输入阻抗	1.8 KΩ	
输入电流	12mA/DC24V	12mA/AC24V
OFF→ON 响应	5—25 ms	5—30 ms
ON→OFF 响应	5—25 ms	5—30 ms
内部 9V 消耗电流	最大 130mA/模块 典型 4.5mA/ON 点+2.5mA	
重量	170g	





§5—2 输出模块

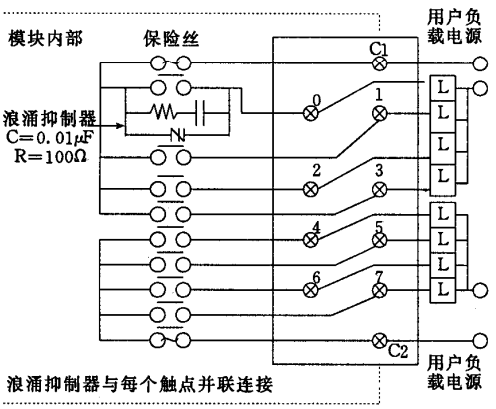
继电器输出模块（8 点） E—01T

项 目	特 性
回路数	8 个常开触点
最大开关容量	(220VAC,4A 阻性负载(寿命 10 万次以上) 200VAC,0.5A(寿命 10 万次以上) 220VAC,0.05A(寿命 80 万次以上) 110VAC,0.5A(寿命 10 万次以上) 110VAC,0.1A(寿命 65 万次以上) 灯/感性负载 24VDC,5A 阻性负载(寿命 10 万次以上) 24VDC,0.5A 灯负载(寿命 10 万次以上)
触点间的漏电流	1 mA 以下（220V，60Hz）
最小开关容量	5 mA，5V
内部 9V 消耗电流	45mA /回路
保险丝	10A（4 点一个，可更换）
重量	200g

注:1。由于浪涌抑制器，漏电流可能启动要求很小电流的负载。例如氖灯。

2．感性负载：上升时功率因素为 0.65，下降时功率因素为 0.35 时，电流为正常电流的 10 倍。

3．灯负载：上升和下降时的功率因素为 1 时，电流为正常时的 10 倍。



附加电阻的计算

继电器输出模块，用在某些超小负载情况下，须接附加电阻。

下面介绍关于附加电阻的电阻值及功率的计算方法：

计算公式：

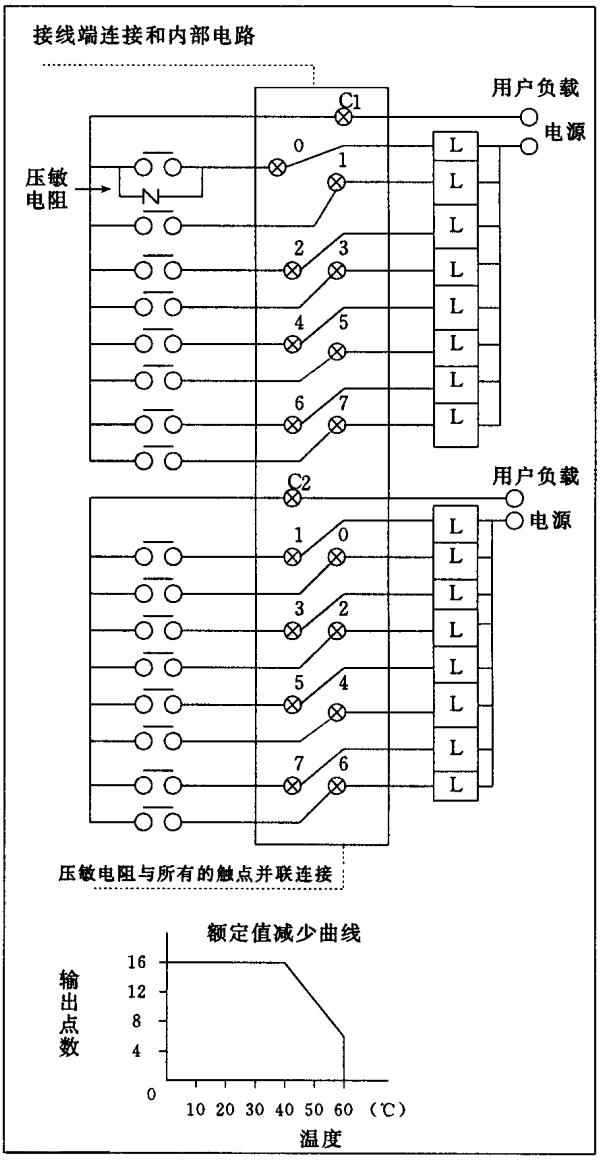
找出附加电阻值最大阻值公式

$R=V1 \times 500$ V1 为正常接通负载的最小电压，找出附加电阻的最小功率公式

$Pr = \frac{V2 \times V2 \times 2}{R}$ V2 为接通负载时常施加的电压，R 为上面所计算出的电阻值，选择电阻时，取此计算电阻稍低一点的但又是最接近的标准电阻值。选择功率值时，取此计算功率值高一点的但又最接近的标准功率值。

继电器输出模块（16 点） E—05T

项 目	特 性
回路数	16 点
输出形式	继电器触点输出
最大开闭容量	220VAC,2A 阻性负载(寿命 10 万次以上) 200VAC,0.25A(寿命 10 万次以上) 220VAC,0.03A(寿命 80 万次以上) 110VAC,0.25A(寿命 10 万次以上) 灯/感性负载 24VDC,2A 阻性负载(寿命 10 万次以上) 24VDC,0.25A 灯负载(寿命 10 万次以上)
触点间漏电流	0.1 mA 以下(220V,60Hz)
最小开关容量	5mA，5V
OFF→ON 响应	12ms 以下
ON→OFF 响应	12ms 以下
内部 9V 消耗电流	30mA /回路
保险丝	无 建议外接 2A 的保险丝
重量	240g



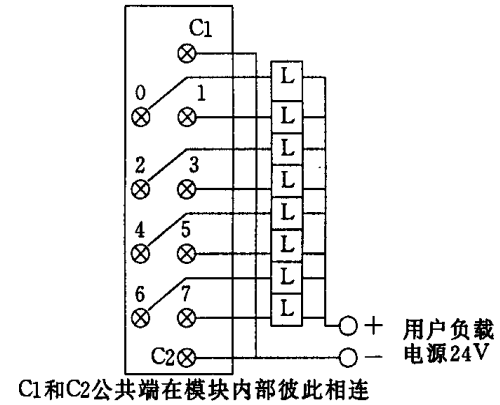
24VDC 汇点输出模块(8 点) E—10T

项 目	特 性
回路数	8 点
输出形式	NPN 开路集电极输出
开闭容量	24VDC，0.5A
漏电流	100μA 以下(加 40V 电压时)
ON 时电压降	0.65V(1A),0.7V(0.25A),0.8V(0.5A)
允许峰值电压	45V(包含波纹,电源负载的峰值)
OFF→ON 响应	0.1ms
ON→OFF 响应	0.1ms
内部 9V 消耗电流	20 mA
内部 24V 消耗电流	3mA/回路
保险丝容量	3A(焊接型)
重量	120g

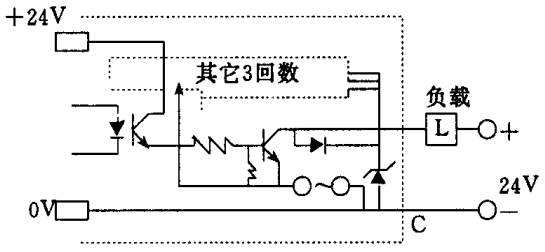
24VDC2A 汇点/源输出模块(4 点) E—12T

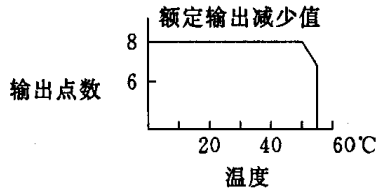
项 目	特 性
回路数	4 点
输出形式	与 NPN 开路集电极输出相当
开闭容量	24V，2A（环境温度 16℃，时间常数为 100ms 的感性负载）最大 6A
漏电流	0.4mA 以下(40VDC 时)
ON 时电压降	0.15V(1A),0.3V(2A) 0.6V(4A),0.9V(6A)
允许峰值电压	45V(包含波纹,电源负载的峰值)
OFF→ON 响应	0.1ms
ON→OFF 响应	0.1ms
内部 9V 消耗电流	12 mA
内部 24V 消耗电流	5mA
保险丝容量	5A(可更换)
重量	120g

接线端连接与内部电路

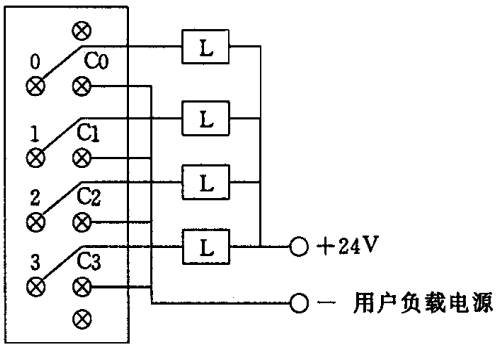


C1和C2公共端在模块内部彼此相连

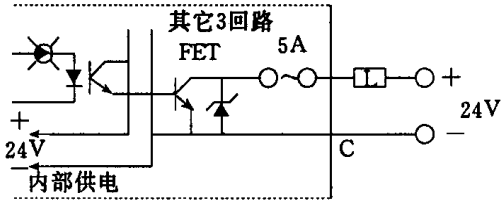




接线端连接与内部电路



负载电源0V连接到内部0V电源线



开闭频度的限制

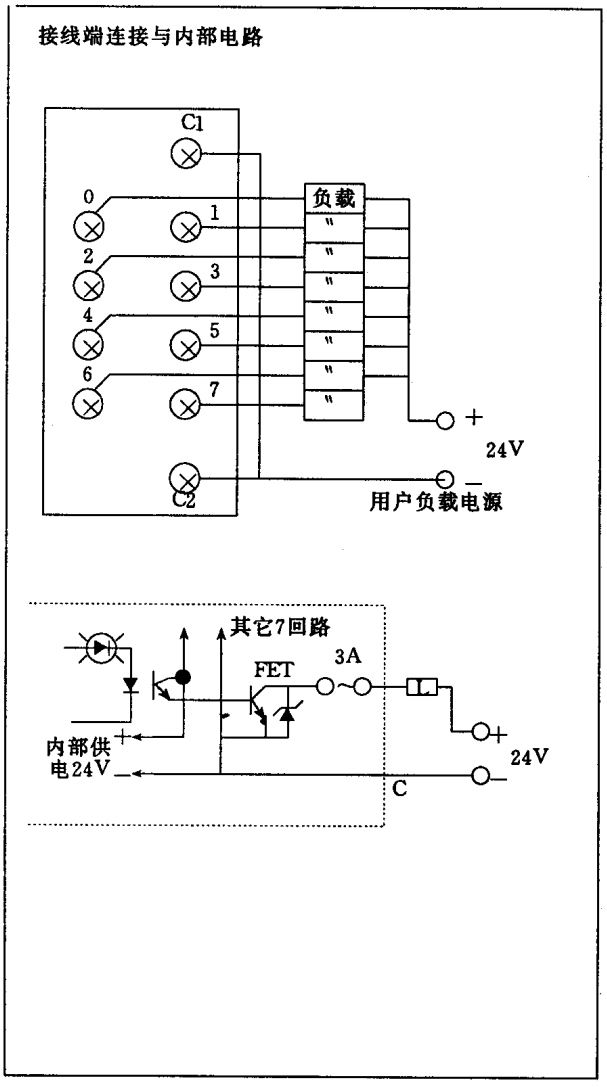
每回路产生1.5W发热量时的开闭频度如下表所示。当开闭频度增加时，发热量成比例增加，星号(*)表示一个二极管应该与负载并联连接。

负载型式	负载的时间常数	负载电流(在24V时)			
		1A	2A	4A	6A
电阻负载	—	开关频度无限制			
感性负载	7ms	1000次/分	500次/分	250次/分	20次/分
	40ms	180次/分	90次/分	*	*
	100ms	70次/分	15次/分	*	*
灯负载		0.8最大(额定)			

24VDC2A 汇点输出 模块(8 点) E—14T

项 目	特 性
回路数	8 点
输出形式	与 NPN 开路集电极输出相当
开闭容量	24V，2A（环境温度 60℃，时间常数，100ms 的感性负载）最大 6A
漏电流	0.4mA 以下(40VDC 时)
ON 时电压降	0.3VDC(2A)
允许峰值电压	45V(包含波纹,电源负载的峰值)
OFF→ON 响应	0.1ms
ON→OFF 响应	0.1ms
内部 9V 消耗电流	30 mA
内部 24V 消耗电流	10mA
保险丝容量	3A(可更换)（每点一个）
重量	135g

接线端连接与内部电路

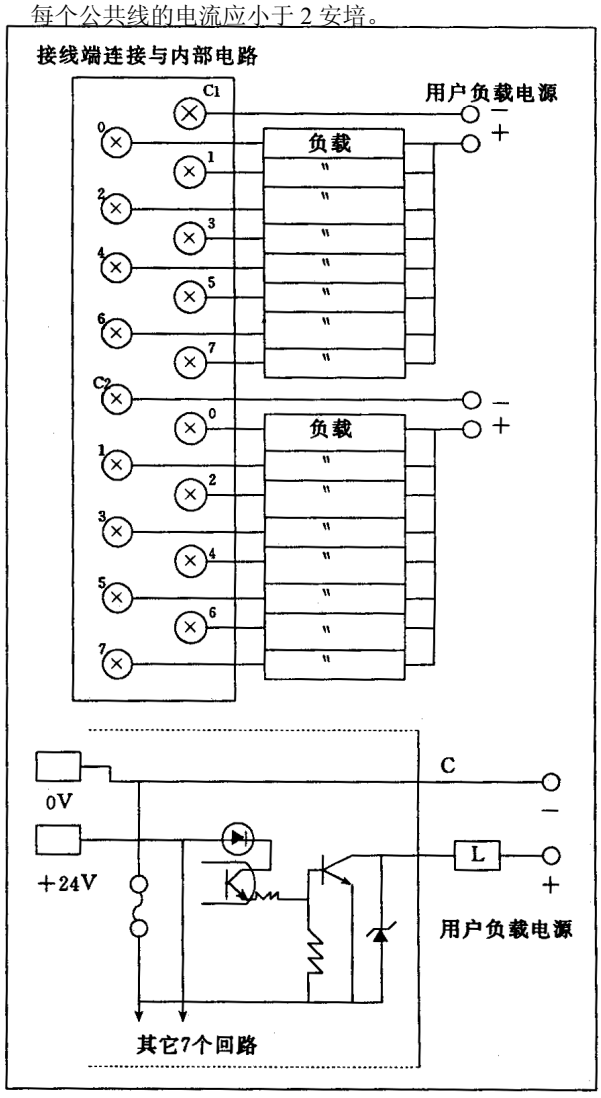


24VDC 汇点输出模块(16 点) E—15T

项 目	特 性
回路数	16 点
输出形式	NPN 开路集电极输出
开闭容量	24VDC，0.5A
漏电流	100μA 以下(40V 电压时)
ON 时电压降	2.0V(0.5)
允许峰值电压	45V(包含波纹,用户负载的峰值电压)
OFF→ON 响应	0.1ms
ON→OFF 响应	0.1ms
内部 9V 消耗电流	3 mA+2.3 mA/回路
内部 24V 消耗电流	6mA/回路
保险丝容量	3A(不可更换)（每 8 点一个）
重量	160g

每个公共线的电流应小于 2 安培。

接线端连接与内部电路



115/230VAC 输出模块（8 点） E—20T

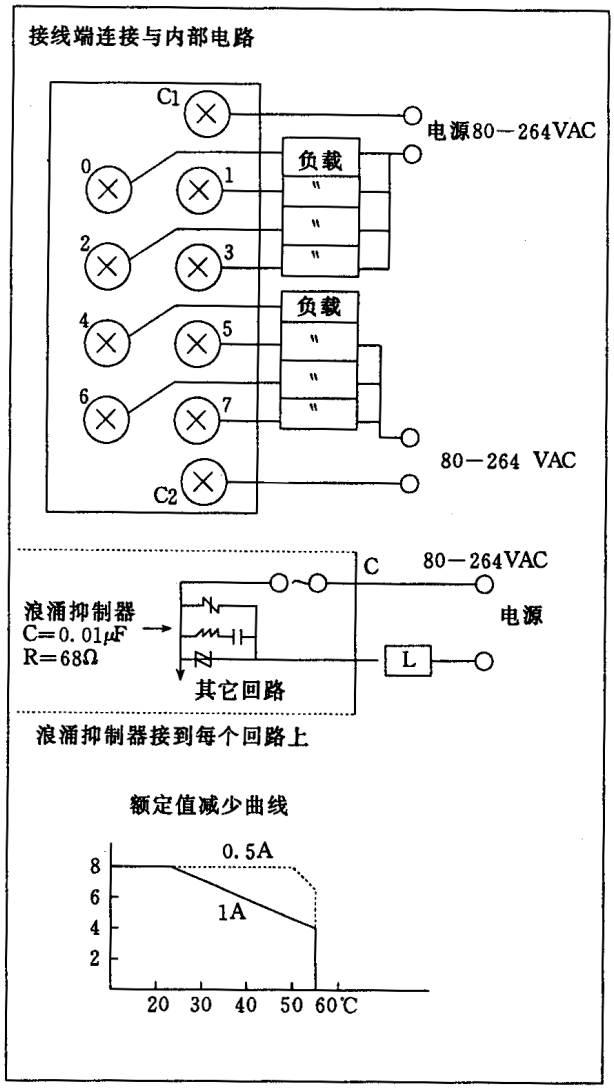
项 目	特 性
回路数	8 点
最大开闭容量	80—264VAC，1A（峰值 10A，1 周期内）
最小开闭容量	80—264VAC，25Ma
漏电流	0.45mA(50Hz,110V),1mA(50Hz,220V) 0.52 mA(60Hz,110V),1.2mA(60Hz,220V)
剩余电压	0.8V(0.5A),0.9V(1A)
OFF→ON 响应	1ms
ON→OFF 响应	1/2 周期以内
内部 9V 消耗电流	20mA/回路
保险丝	5A（焊接型）
重量	180g

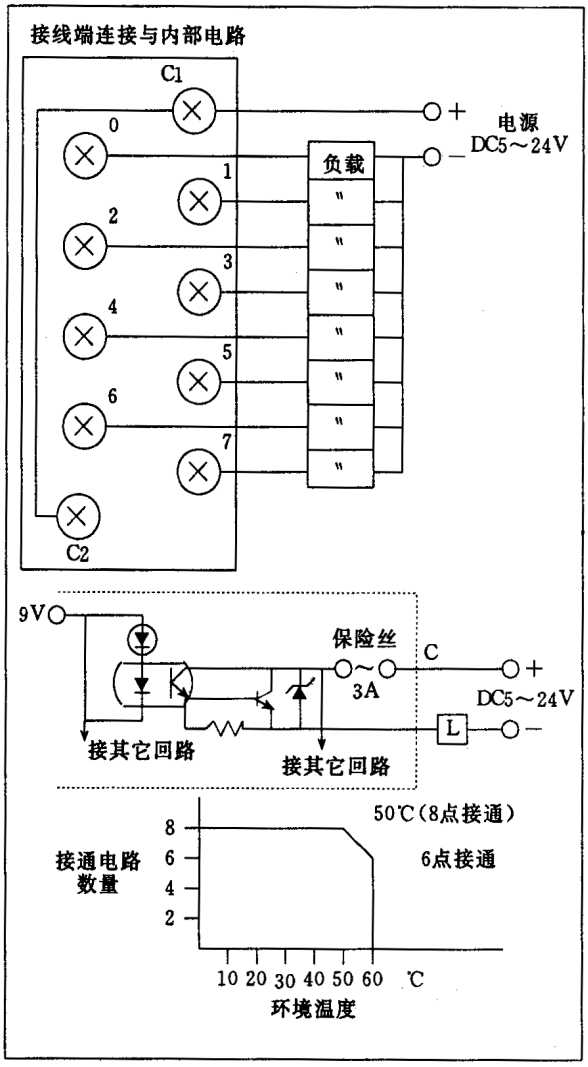
注:1、由于浪涌抑制器，漏电流可能启动要求很小电流的负载，例如氖灯。

2、4 个回路有一公共端，两公共端是分隔的。

24VDC 源输出模块（8 点） E—50T

项 目	特 性
回路数	8 点
输出形式	NPN 晶体管发射极跟随器
开关电压	5—24VDC
峰值电压	40VDC
最大电流	0.5A
ON 时电压降	1.0V(0.5A),0.75V(0.1A)
漏电流	0.1mA 以下(24VDC 时)
最小推荐负载	1mA
OFF→ON 响应	0.1ms
ON→OFF 响应	1ma
内部 9V 消耗电流	30Ma
保险丝	3A（每 4 点一个）
重量	200g





115/230VAC 分隔式输出模块(4 点) E—21T

项 目	特 性
回路数	4 点
最大开闭容量	80—264VAC，2A（连续）
最小开闭容量	80—264VAC,10mA
漏电流	3mA（50Hz，110V），6mA（50Hz，220V） 3.5mA(60Hz,110V),7mA(60Hz,220V)
ON 时电压降	0.8V(2A),1V(1A),1.5V(10A)
OFF→ON 响应	1ms 以内
ON→OFF 响应	1/2 周期以内
内部 9V 消耗电流	12mA/回路
保险丝	3A(可更换)
重量	160g

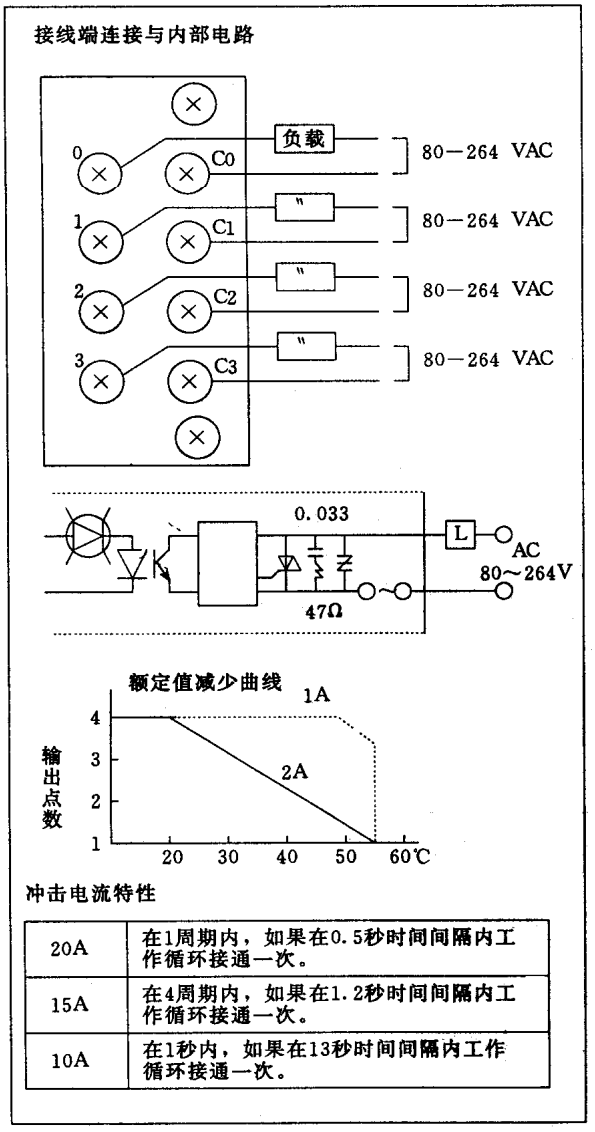
注:1、在电感或灯负载情况下，冲击电流约是正常电流的 10 倍，检查冲击电流和保险丝，保险丝应低于 0.5 秒以内为 6A 和 6 分钟以内为 4A。

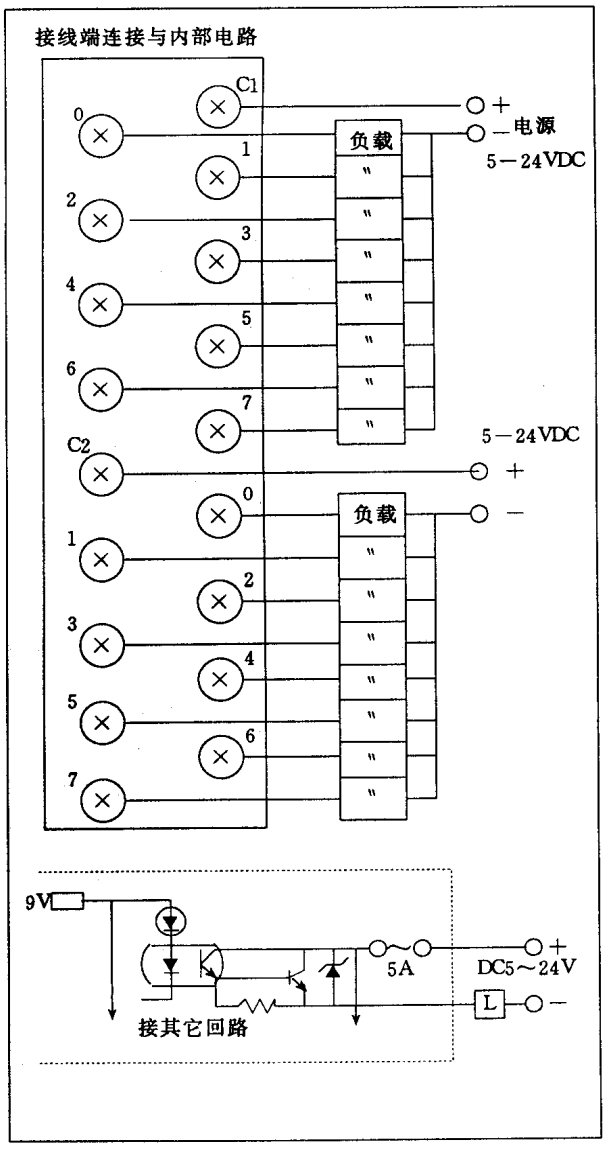
2、负载电流小于 10mA 可能引起不可靠的工作，在这种情况下，在负载上并联电阻，使负载电流增加到 10mA 可更高。

3、带有二极管的负载也可使用。

24VDC 源输出模块（16 点） E—55T

项 目	特 性
回路数	16 点
输出形式	NPN 晶体管发射极跟随器
剩余电压	最大 1.5V 0.7V(正常),(负载电流=0.1A) 0.75V(正常),(负载电流=0.25A) 0.8V(正常),(负载电流=0.5A)
开关电压	5—24VDC
用户负载的峰值电压	40V
额定电流	0.5A(电阻负载) 0.1A(灯负载) 0.5A,7ms(时间常数) 0.3A,40ms(时间常数) 0.2A,100ms(时间常数) 感性负载
OFF→ON 响应	0.1ms 以下(电阻负载)
ON→OFF 响应	1ms 以下(电阻负载)
漏电流	10μA 以下(40V 电压)
内部 9V 消耗电流	7.5mA(典型)/ON 点数,最大 180mA/全部点
保险丝	5A（焊接型）（每 8 点一个）
重量	200g

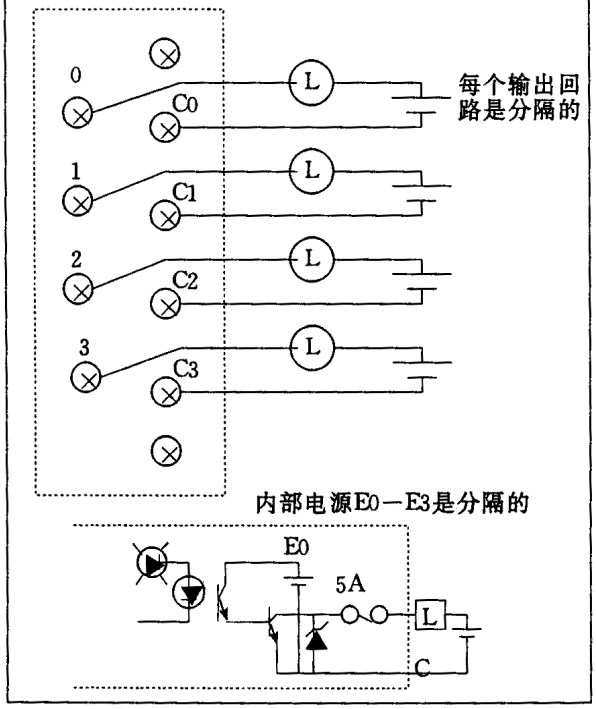




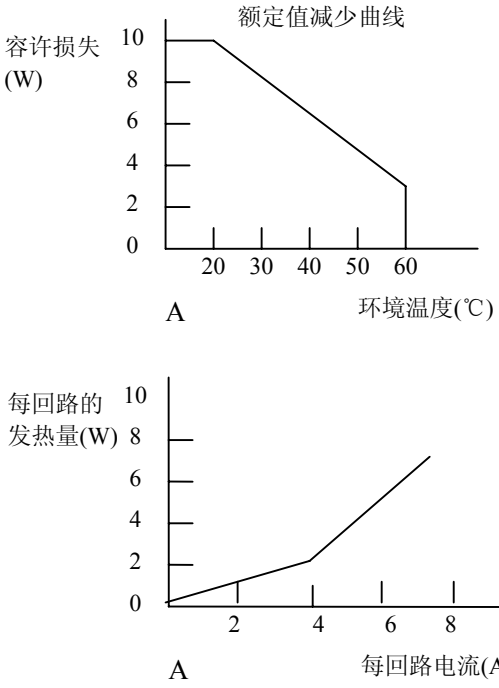
24VDC 汇点/源输出模块(4 点) E—52T

项 目	特 性
回路数	4 点，独立分隔
开闭容量	24V，2A（连续），6A（最大瞬时电流）
漏电流	0.4mA 以下(输出端 40V 电压)
剩余电压	0.15V(1A),0.3V(2A),0.6(4A),1V(6A)
允许峰值电压	45V(输出端间)
OFF→ON 响应	0.1ms
ON→OFF 响应	0.1ms
内部 9V 消耗电流	12mA 以下
内部 24V 消耗电流	30mA 以下
绝缘耐压	1.5KVAC
保险丝	5A(可更换)
重量	120g

接线端连接与内部电路



特性曲线



容许损失

每块模块的容许损失能够通过下面两种情况的数值的和来确定。

(1) 由每四路电流确定的损失。

(2) 在感性负载下的损失。

在感性负载最大开闭频率场合时，发热量定为 1.5W。这个损失与频率成正比。

通过并联连接能够增加电流，感性负载的遮断特性由表示每一个回路的最大开闭频度表来确定。

在灯负载的情况下，串联连接也是可能的，灯负载的额定电流应小于 0.8A。

在每个输出电路中使用了场效应晶体管，但是电路图表示了使用一个 NPN 晶体管的等效电路。

由于每个输出回路彼此是分隔的，因此各种回路的自由组合，无论是汇点还是源输出是可以的。

最大开闭度表

负载型式	负载的时间常数	负载电流（在 24V 时）			
		1A	2A	4A	6A
电阻负载	—	无关频度无限制			
感性负载	7ms	1000 次/分	500 次/分	250 次/分	20 次/分
	40m	180 次/分	90 次/分	*	*
	100ms	70 次/分	15 次/分	*	*

*表示负载需要并联二极管

§5—3 输入/输出模块

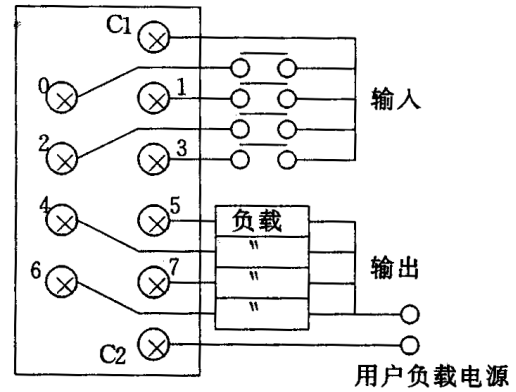
24VDC 输入/继电器输出模块（4/4 点） E—01NT

项 目	特 性
回路数	输入：4 点（24VDC） 输出：4 点（常开触点）
输入电路	与 E—01N 相同
输出电路	与 E—01N 相同
内部 9V 消耗电流	输出：45mA/回路+20 mA
内部 24V 消耗电流	输入：14 mA/回路
保险丝	10A（对于输出，可更换）
重量	170g

24VDC 输入/继电器源输出模块（8/8 点） E—05NT

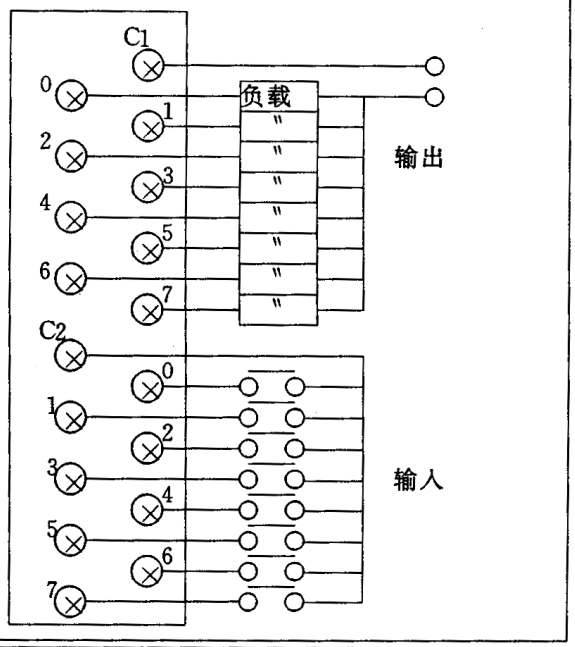
项 目	特 性
回路数	输入 8 点（24VDC） 输出 8 点（继电器输出）
输入电路	与 E—01N 相同
输出电路	与 E—05T 相同，继电器输出
内部 9V 消耗电流	输入 10mA/模块，输出 30 mA/点以上
内部 24V 消耗电流	输入 14mA/点
保险丝	无，建议每个负载外接 2A 的保险丝

接线端连接与内部电路



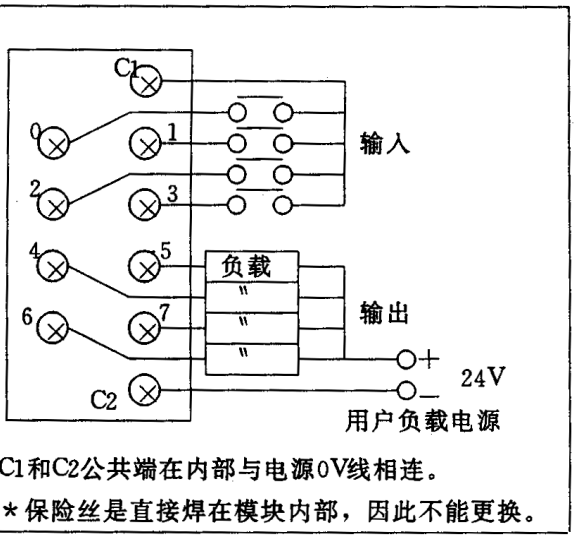
公共端C1与内部24V电源0V端相连接

接线端连接与内部电路



24VDC输入/开路集电极输出模块(4/4点)E—10NT

项目	特性
回路数	输入4点(24VDC) 输出4点(24VDC，0.5A)
输入电路	与E—01N相同
输出电路	与E—10T相同，开路集电极输出
内部9V消耗电流	20mA
内部24V消耗电流	输入:14mA/回路 输出:3mA/回路
保险丝	3A(对于输出，焊接型)
重量	130g



C1和C2公共端在内部与电源0V线相连。

＊保险丝是直接焊在模块内部，因此不能更换。

§5—4 拨盘接口（E—01D）

该模块提供 SR—21 最多四组拨盘开关的接口给用户来使用。每组拨盘开关允许操作者对一个定时器或计数器设定预置值。事实上，假如没有该接口模块，就无法使用 674~677 这四个定时器或计数器。用于检测这些拨盘开关状态的电压（24VDC）由 CPU 所在的电源框架提供，该接口必须与 CPU 安装在同一个框架上，并且必需安装在靠近 CPU 的 4 个槽的任一个槽中，在每个系统中，只能用一个接口。拨盘开关及它们的联接线由用户提供。拨盘开关采用标准的 BCD 编码，并有二极管隔离，大多数拨盘开关都可选用。

只要该接口安装在 CPU 电源框架中，8 个离散定义号就分配给该模块所在的槽，该接口的操作并不需要这此定义号。它们可用作内部线圈，但不能给其它硬件 I/O 使用。每次扫描，全部四个预置值被读入 SR—21PLC。为该接口模块的 20 个接线端的接线图见拨盘开关接线图。对不使用的电路，也并不需要特殊的接线。

每次扫描把拨盘值读入 CPU，并保证迅速响应新的数值。当你改变拨盘开关的数值时应小心，因为在一次扫描或几次扫描期间，可能会将中间数值读入并使用。例如，如果拨盘开关设置值为 095，您需要的新的数值为 105，如先改百位数，则 CPU 检测到并使用的结果是 195，直到十位数从 9 变到 0 为止。无论怎样，如果首先改变十位数，在 105 设置之前的值 005 将被读入。实际结果取决于应用和进入的特定逻辑。

通常，较高的数比低的数据更容易接收，从左即高位数输入新值的自然趋势将导致较大的值。（通常，人的习惯总是先从左边即高的位数开始改变数值，这将引起较大的数值变化）。

该模块内部功耗： 24VDC90mA（9 个负载单位）

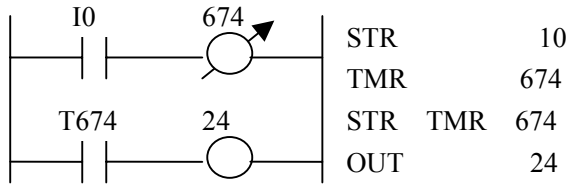
9VDC100mA（10 个负载单位）

注：E—01D 所接拨盘开关的数据不仅作为 674~677 的预置值，而且被送入下列数据寄存器中

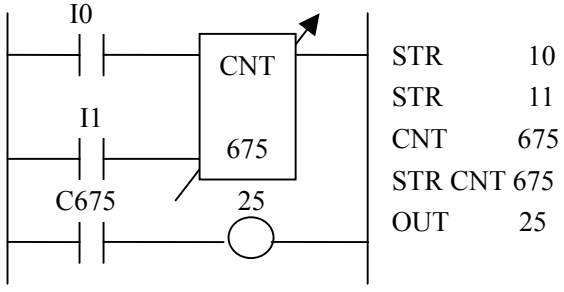
第一组拨盘数据	送入 R565，R564
第二组拨盘数据	送入 R567，R566
第三组拨盘数据	送入 R571，R570
第四组拨盘数据	送入 R573，R572

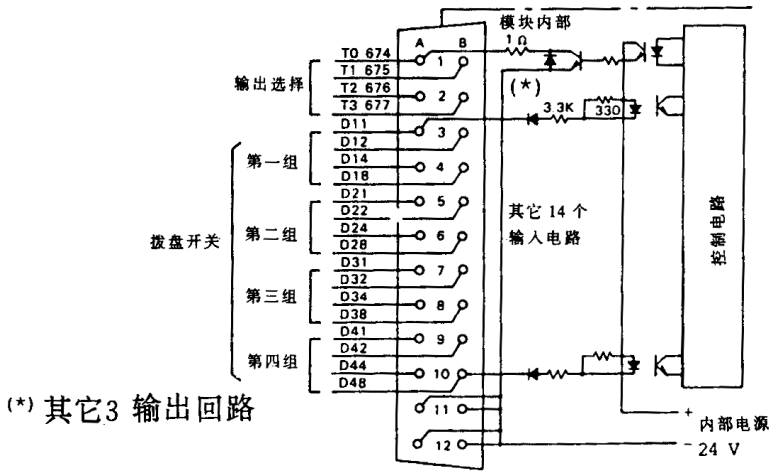
附：采用 E—01D 拨盘接口模块时 674、675、676、677，作为定时器或计数器编程时，在输入定时器或计数器指令后，不要输入预置值。

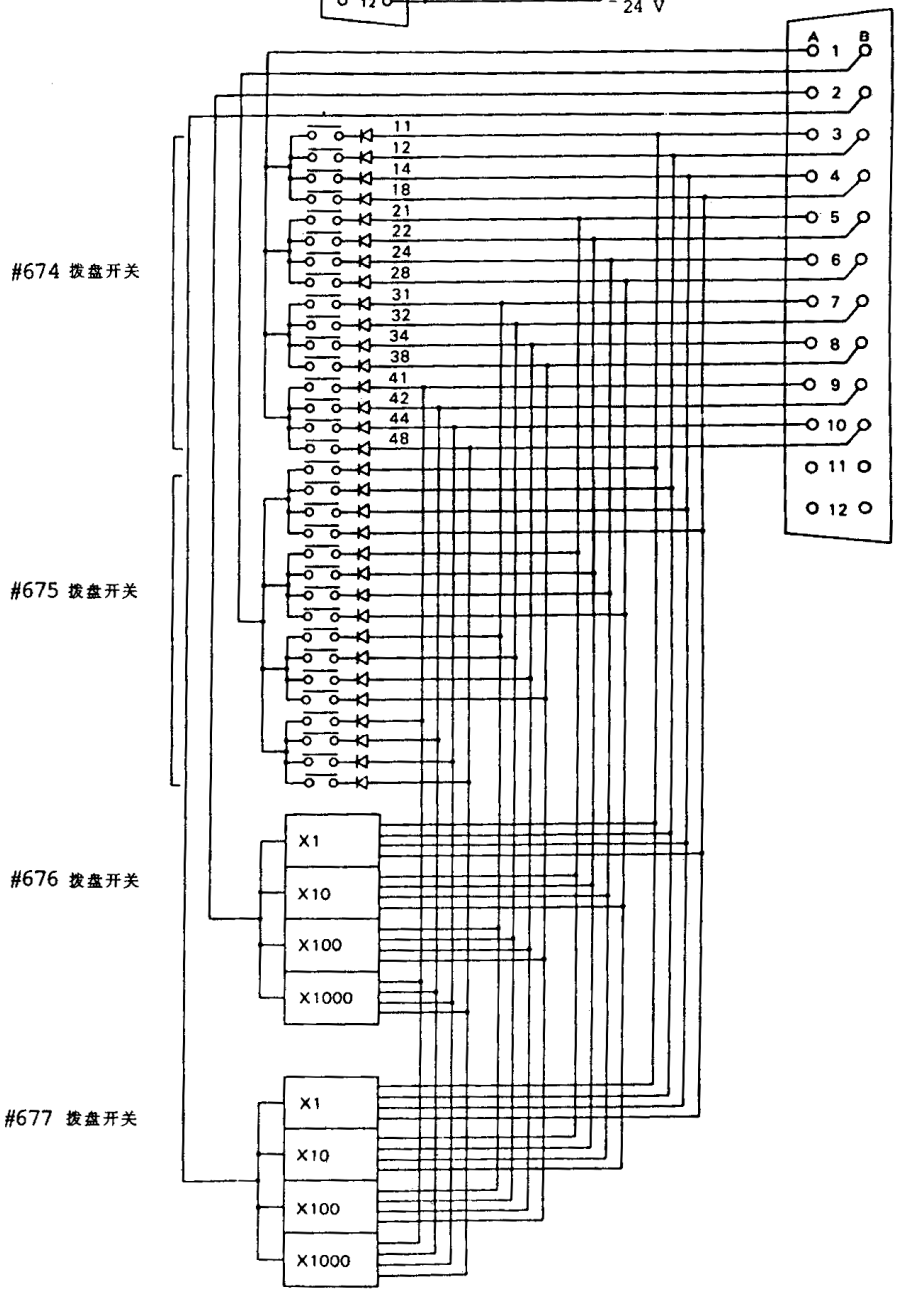
定时器编程举例



计数器编程举例







四组拨盘开关是并行连接的，因此各组拨盘开关间需安装隔离二极管。

拨盘接口接线

§5—5 模块量输入模块

模块量输入模块 E—01AD

一、概述

模拟量输入模块有四个独立的输入通道，这些通道能将模拟量输入信号转换成数字量信号供可编程序控制器处理。在 SR—21 中，除了基本功能外，还具有数据操作（包括运算功能），因此在用模拟量输入模块时可执行其它类型的处理。每个模拟量输入模块需要 16 个 I/O 定义号作为其地址。

二、硬件特性

用户可以通过跨接片的设置，来选用电压输入（1—5VDC）或电流输入（4—20mA），该模块出厂时设置成电压输入。分辩率是 8 位的，即转换的最大数值为 255。在模块面板上有 8 个 LED 作为输入的八位二进制显示。通过按面板上的一个按钮来选择显示的通道。每按一次按钮，就选择一个通道，例如通道 1、2、3、4、1 等。

用户现场接线接到模块面板上的一个可拆式接线端上。在接线端板上有一个塑料罩子以保护接线端，罩子有一可插的标志纸，可用来记录电路的信息。

最大转换时间为 2ms，不会增加 CPU 扫描时间。每次转换四个通道中的一个，每个扫描周期转换一个通道，下一次扫描依次转换下一个通道。

三、电源要求

该模块需要一个 24VDC 外接电源。您可以用 SR—21 高容量框架提供的 24VDC 电源，但每个框架提供的电流限于 100mA 以内，因此，框架只能供给一个模拟量输入模块使用，因为这个模块需要的操作电流最大为 65mA。

四、一般特性和电气技术规格

该模块的技术规格见下表。

表 1 模拟量输入模块技术规格

项　　目	规　　格
通 道 数	4（独立）
输入范围	1—5VDC 或 4—20mA（每个通道由跨接线选择）
分 辩 率	八位二进制（1/256）
数字输出	八位二进制数据输出 四位通道状态
需要 I/O 点	16 个（分配给每个槽的 16 点，如 0—7 和 100—107 等）
LED 显示（16 个 LED）	八位：读入所选通道的数据 四位：数据显示的通道选择 四位：被扫描的通道
工作环境温度	0—60℃
湿　　度	5%~95%（无凝露）
输入形式	差动输入
输入阻抗	≥1MΩ（电压输入） 250Ω（电流输入）
安全的最大额定值	电压输入，0—10VDC 电流输入，0—30mA
转换方法	逐次逼近方式
转换时间	最大 2ms
线性误差	最大±0.8%
精　　度	25℃时最大±1%
精度与温度关系	±70PPM（1/百万）/1℃
外部电源	+24VDC±10%，65mA（最大）
9VDC 功耗	25—55mA（由框架电源提供）
重　　量	200g

一、 I/O 定义号

模拟量输入模块用 16 个 I/O 点，从分配给该模块所占 I/O 槽的第一个 I/O 定义号开始。在表中，假定模拟量输入模块放置在 5 槽或 10 槽框架的第三个（2 号）槽中（定义号为 10—17，110——117），采用这些定义号只是举个例子。对于其它可以有 16 个定义号的 I/O 槽，可见前面几章的介绍。下表定义了每个 I/O 点的使用情况。

表 2 I/O 点定义

I/O 点	定　义	权	I/O 类型
10	二进制数据位	1	输入
11	二进制数据位	2	输入
12	二进制数据位	4	输入
13	二进制数据位	8	输入
14	二进制数据位	16	输入
15	二进制数据位	32	输入
16	二进制数据位	64	输入
17	二进制数据位	128	输入

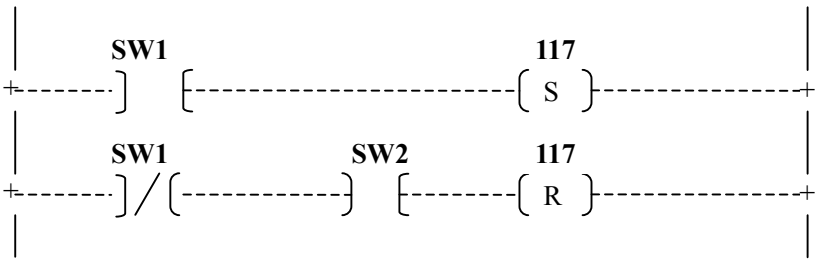
表 2 I/O 点定义（续）

I/O 点	定　　义	I/O 类型
110	通道 1，该位用于梯形图中，指示该通道接收输入的数据	输入
111	通道 2，同上	输入
112	通道 3，同上	输入
113	通道 4，同上	输入
114 115	114 和 115 这两个输出组合起来用于设置选择只扫描一个通道，而不是全部四个。如果扫描选择输出是接通的，这些位就可忽略。在梯形图中设定这两个输出，可选定您希望的通道，对应于各通道的状态如下： 输出 114　输出 115　通道号 0　0　1 1　0　2 0　1　3 1　1　4	输出 输出
116	不用	
117	扫描选择输出。为使模块读入数据输入，该位必须设置，除非使用了选择通道扫描输出。这个输出在梯形图中设置。	输　出

二、 梯形图举例

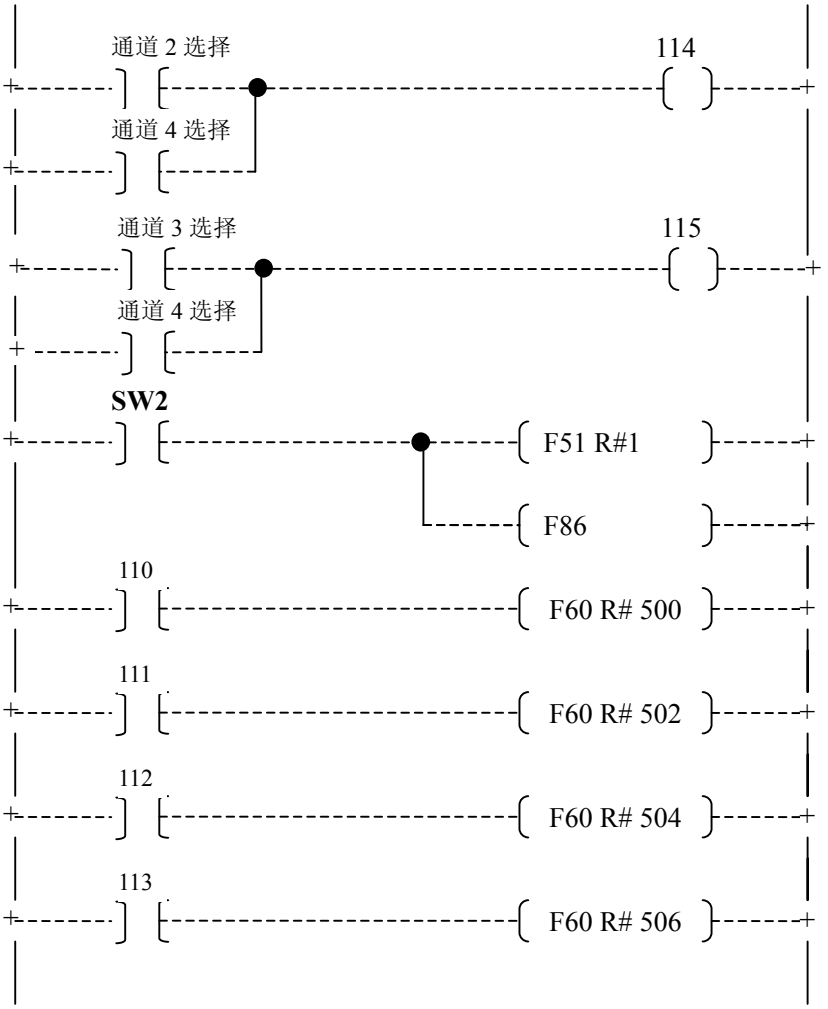
下面例子中应用的 I/O 定义号与表 2 定义的 I/O 点相同。这是 SR—21 的编程例子。

例 1：扫描全部四个通道，输出 117 必须 ON。



SW1 和 SW2 为用户开关

例 2：该梯形图例子为选择被扫描通道。



读入数据

转换成 BCD

第一通道数据存于 500 501 寄存器

第二通道数据存于 502 503 寄存器

第三通道数据存于 504 505 寄存器

第四通道数据存于 506 507 寄存器

上面例子中使用的功能代码是：

F51=D.STR1(数据存贮 1)，F86=BINtoBCD，F60=D.OUT(数据输出)

三、 模拟量输入模块部件

图 5.1 是模块量输入模块的面板说明，标明了其器件和用户接线端。

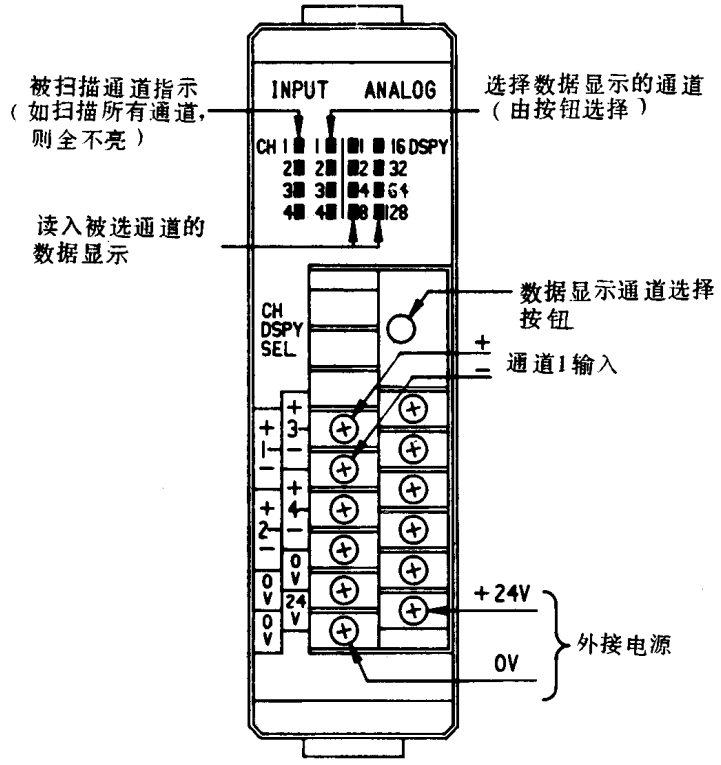
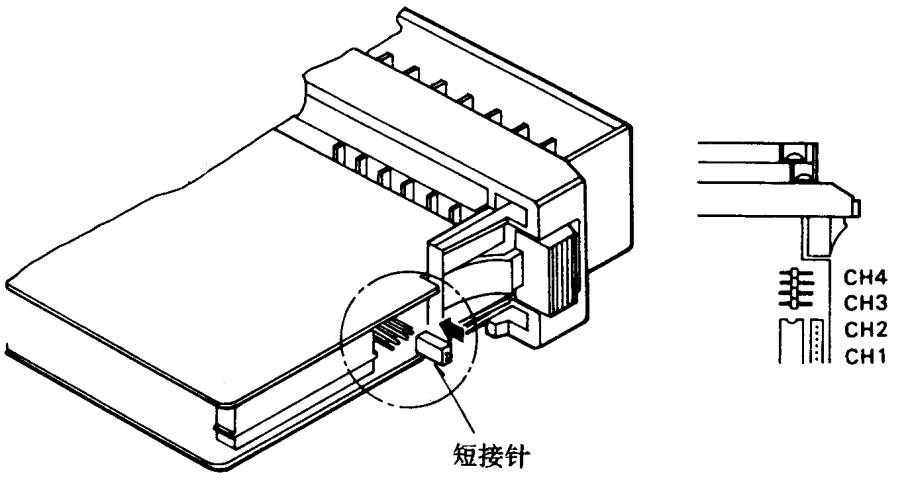


图 5.1 模拟量输入模块面板

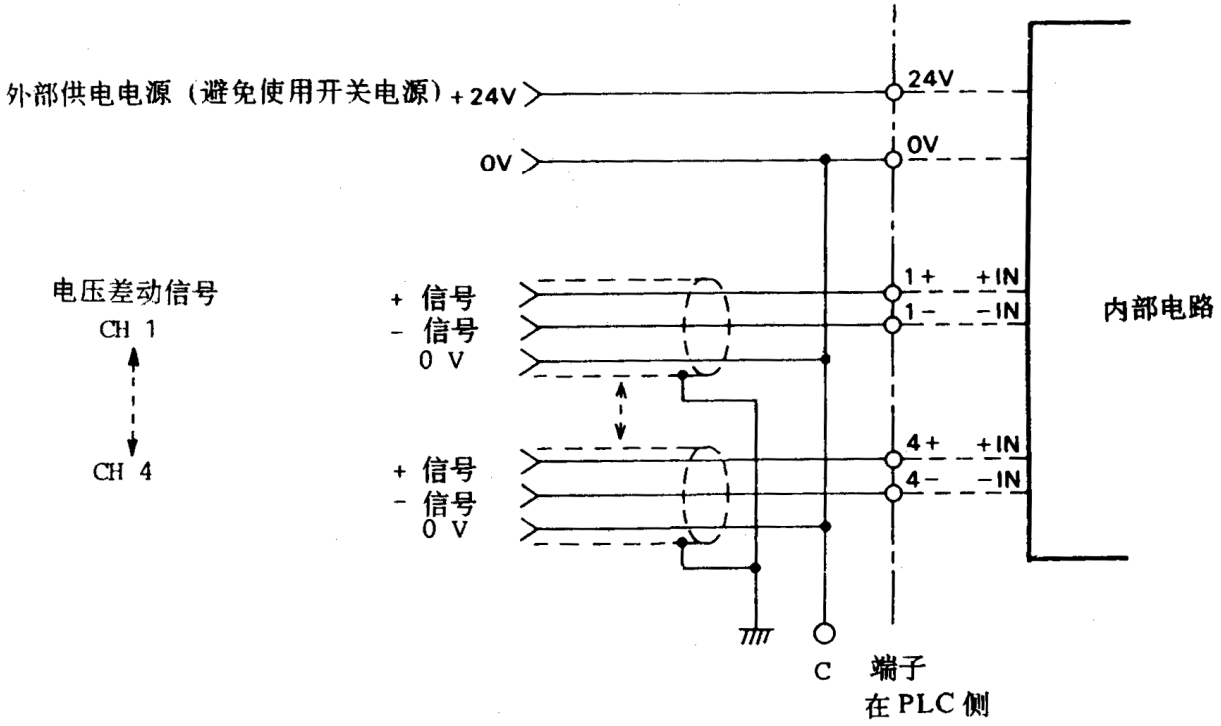
八、工作方式的选择

工作方式是用 1—5VDC 还是用 4—20mA，由位于底板旁边跨接器引脚的连接或不连接决定。有四组引脚，（每组两脚，）每个通道一组，第一组引脚标有 CH1，最后一组引脚标有 CH4。出厂时的设置是 1—5VDC 工作方式，即跨接器未连接。对任一通道，如果选择 4—20mA 工作方式，就需将一跨接器对应于该通道的两引脚连接起来，并使其可靠地连接。

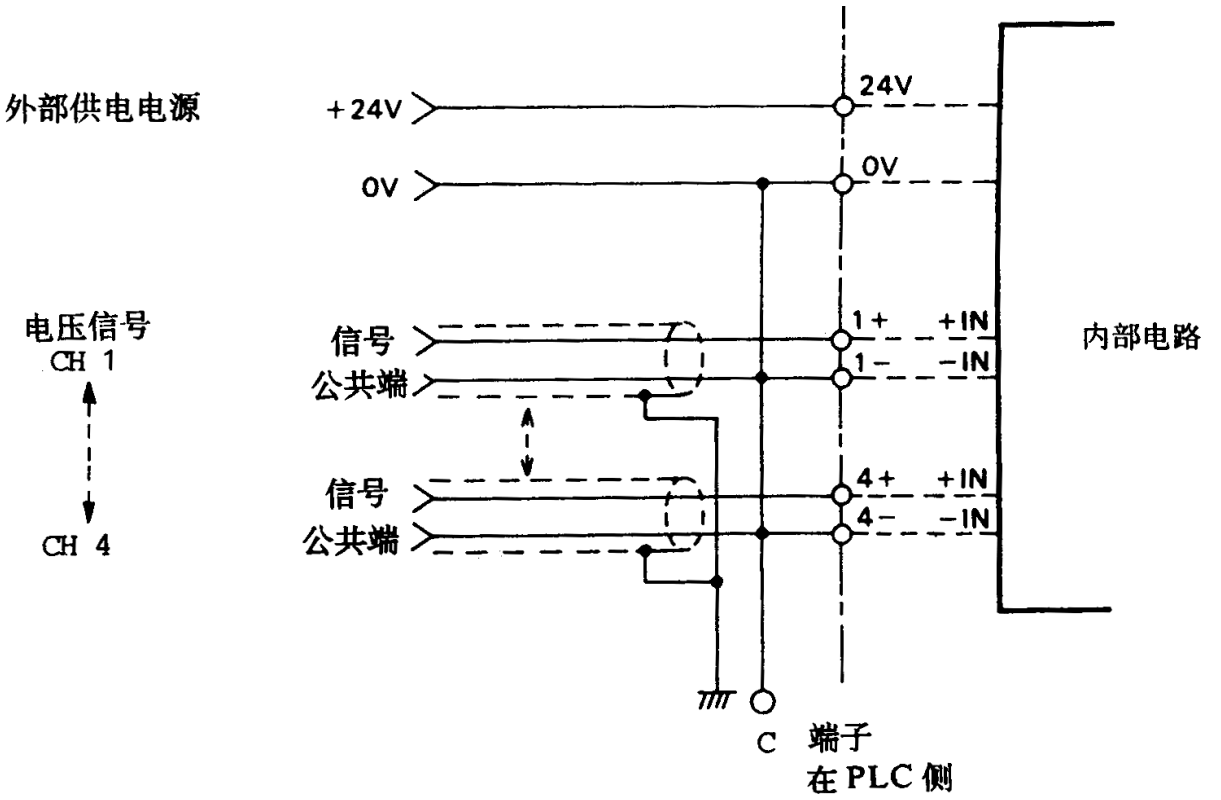


九、外部连线举例

（1）电压信号差动输入



（2）电压信号公共输入



模拟量输入模块 E—02AD

一、概述

E—02AD 模拟量输入模块有一个输入通道，该模块把模拟量输入信号转换成 12 位数字信号供可编程序控制器处理。

该模块可使用的模块量输入信号范围为 1—5V，4—20mA，0—10V。另外，当使用 0—10V 输入信号时，通过外接附加电阻，可设定模拟量输入为任何范围。

二、规格

项 目	规 格		
点用 I/O 点数	16 点		
通道数	1 路		
分辨率	1/4096（12 位）		
输入范围	电压输入	电压输入	电流输入
	1—5V	0—10V	4—20mA
绝对输入信号范围	±15V 最大	±15V 最大	±30mA 最大
输入阻抗	约 1MΩ	约 1MΩ	250Ω
输入型式	差动输入 在 0—10V 电压输入时，如果外接附加电阻，则可用作 0—2mA，0—5mA，0—10 mA 和 0—20 mA 电流输入信号		
线性误差 温度精度 总合精度	±0.1%最大 ±0.007/℃最大 ±0.4%(25℃时)	在最大信号情况下	
转换方法	积分方式		
转换时间	41ms 最大（循环重复时间为 50ms+PLC 扫描时间）		
正常方式波动除去比	约 6dB 200HZ		
公共方式波动除去比	＞60dB 60HZ		
动作显示	12 位，显示数字转换的二进制输出值		
绝缘方式	光耦		
内部 9V 消耗电流	55mA 最大		
外部电源	电压±15V±5%以内，容许波动 1%以下，电流＜70mA		
重量	200 克		

三、地址分配

地址	数据	地址	数据
n+0	1	n+100	256
n+1	2	n+101	512
n+2	4	n+102	1024
n+3	8	n+103	2048
n+4	16	n+104	—
n+5	32	n+105	—
n+6	64	n+106	—
n+7	128	n+107	—

注：n 表示模块安装位置所确定的第一个 I/O 定义号。

四、方框图

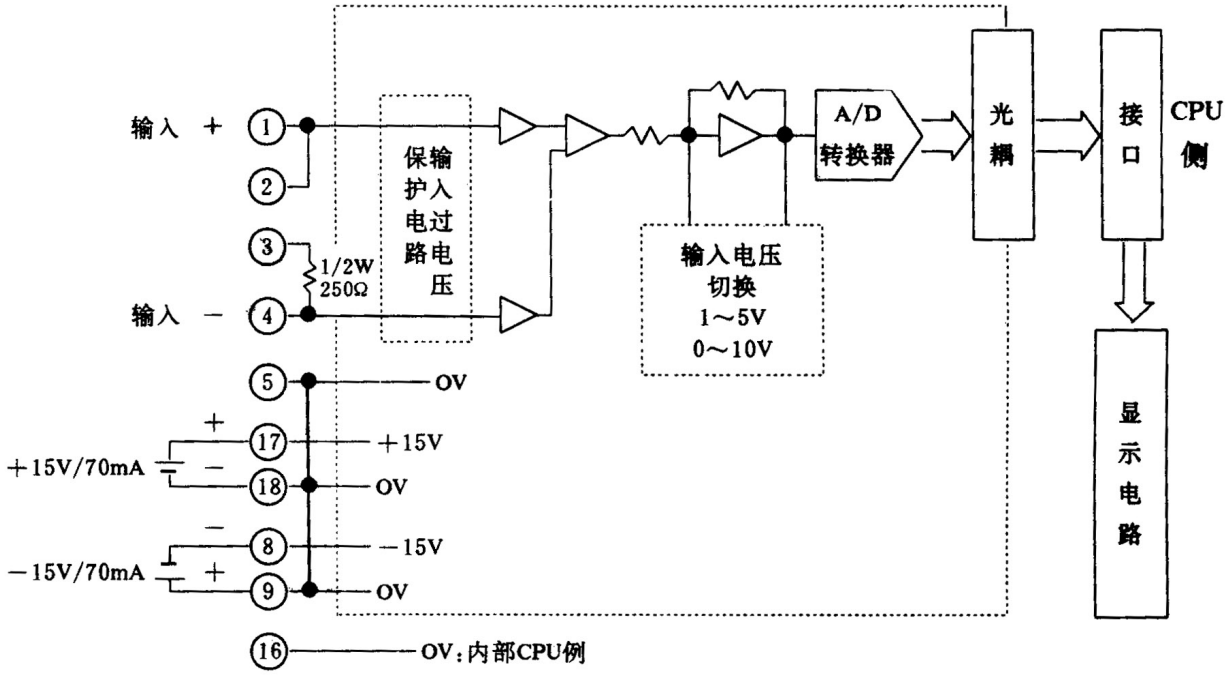


图 5.2 模拟量输入模块 E—02AD 方框图

接线端子号的排列是左侧从顶到底是 1 到底，右侧从顶到底是 10 到 18。

五、使用注意事项：

1．请尽量不要使用开关电源。

2．避免和其它负载使用同一±15V 电源，联合使用电源将对 E—02AD 有坏的影响。

3．接线端子（16）是连到内部 CPU 的 OV 端子上通常不要接它。有时把端子（16）和±150V 电源的 OV 端子（端子（18）和（9））连接可能增加 E—02AD 的 S/N 比或 SR—21 的抗干扰性。

4．请不要触摸印刷板上的调整用电位计。

六、外部连接和面板

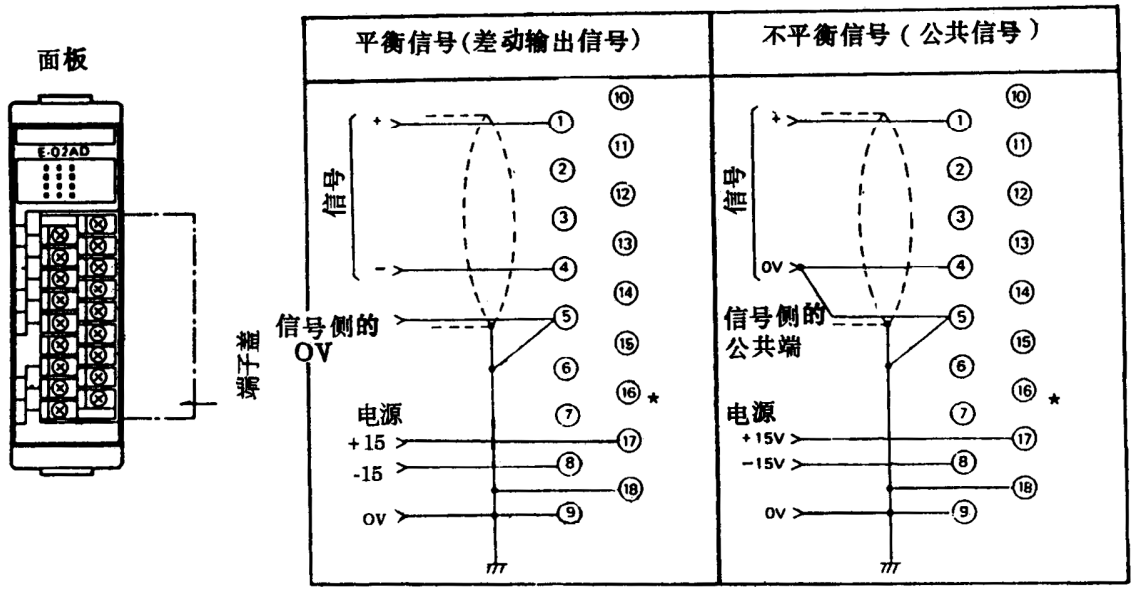


图 5.3 外部连接和面板图

七、按输入规格设定短接针和端子连接的位置

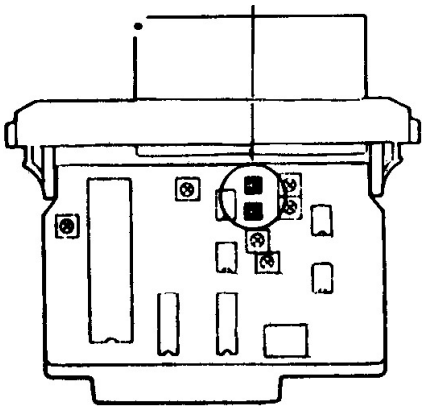
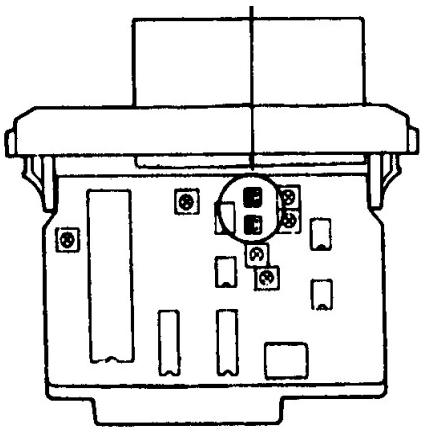
输入规格	端子(2)~(3)之间	外接辅助电阻	短接针设置位置
4~20mA	短接	——	设置短接针在右边 
1~5V	不接	——	
0~2 mA 0~5 mA 0~10 mA 0~20 mA	不接	5KΩ ±0.1%1/4W 2KΩ ±0.1%1/4W 1KΩ ±0.1%1/4W 500Ω ±0.1%1/2W 在端子 1 和 4 之间加接上述电阻,随着 0V~10 的产生可用作电流输入	设置短接针在左边 
0~10V	不接	——	

图 5.4 设定短接针和端子连接位置

§5—6 模拟量输出模块

模拟量输出模块 E—01DA

一、概述

模拟量输出模块提供两个独立的输出通道，每一通道都能将八位二进制数据换成一个模拟量输出。在 SR—21 中除了基本功能外，还具有数据操作（包括运算）功能，因此在用模拟量输出模块时允许进行多种类型的操作，每一个模拟量输出模块需要 16 个 I/O 定义号作为地址。

二、硬件特性

每个通道能提供一个电压输出（0—+10VDC）或电流输出（4—20mA）。用户通过选定如何将现场连接到面板上螺丝接线端可选定每个通道是电压还是电流输出。分辩率是八位，即允许转换的最大数值为 255。在面板上，每个通道都有八个 LED，作为每个通道数据输出的八位二进制显示。

用户的现场接线连接到模块面板上的可拆式接线端板上。在接线端板上盖一个塑料罩，以保护接线端。接线端罩内有一可移动的标记，用于记录线路信息。

最大转换时间为 100μs，不会增加 CPU 的扫描时间。每次扫描可对两个通道一起转换。

三、电源要求

该模块需要一个外接 24VDC 电源，SR—21 框架上的 24VDC 电源输出可以利用，但框架提供的电流是 100mA，只可以给一个模块量输出通道供电，因为每个通道所需的工作电流最大为 85mA，两个通道的最大负荷为 179mA。

四、通用特性和电气指标

该模块的技术指标见表 3。

表 3 模拟量输出模块技术指标

通道数：	2 个（独立）
模拟量输出范围：	0—+10VDC 或 4—20mA，（在面板接线端上对每个通道作选择）
分辨率	八位二进制数 1/256
数字输入数据	来自 CPU 的八位二进制数
需要 I/O 点	16 个（分配给每个槽的 16 个点，如 10—17，110—117）等
LED 显示（16 个 LED，每个通道 8 个）	每个通道输出数据在八位二进制数据显示
工作环境温度	0—60℃
相对湿度	5%~95%（无凝露）
输出阻抗	≤0.5Ω（电压输出）
输出电流	最大 10 mA（电压输出）
外接电阻	最大 550Ω，最小 5Ω（电流输出）
在 CPU 扫描的开始	转换开始
转换时间	最大 100μs
精　　度	25℃时为 0.4%
精度与温度关系	±50PPM（1/百万）/1℃
外部电源	+24VDC，170mA（最大）±5%稳定性
9VDC 功耗	最大 80mA（由框架电源提供）

五、I/O 定义号

模拟量输出模块占用 16 个 I/O 点，它们从分配给该模块所在 I/O 槽中的第一个 I/O 定义号开始。在表 4 中，假定模拟量输出模块放在 5 槽框架中的第 2 个槽中（定义号为 0—7 和 100—107）。使用这些定义号只举个例子，可分配 16 个定义号的其它 I/O 槽，见前面几章的介绍。下表定义了每个 I/O 的使用情况。

表 4　I/O 点定义

I/O 点	定　　义		I/O 类型
	通道　数据	权	
0	通道 1——数据位 1	1	输出
1	通道 1——数据位 2	2	输出
2	通道 1——数据位 3	4	输出
3	通道 1——数据位 4	8	输出
4	通道 1——数据位 5	16	输出
5	通道 1——数据位 6	32	输出
6	通道 1——数据位 7	64	输出
7	通道 1——数据位 8	128	输出

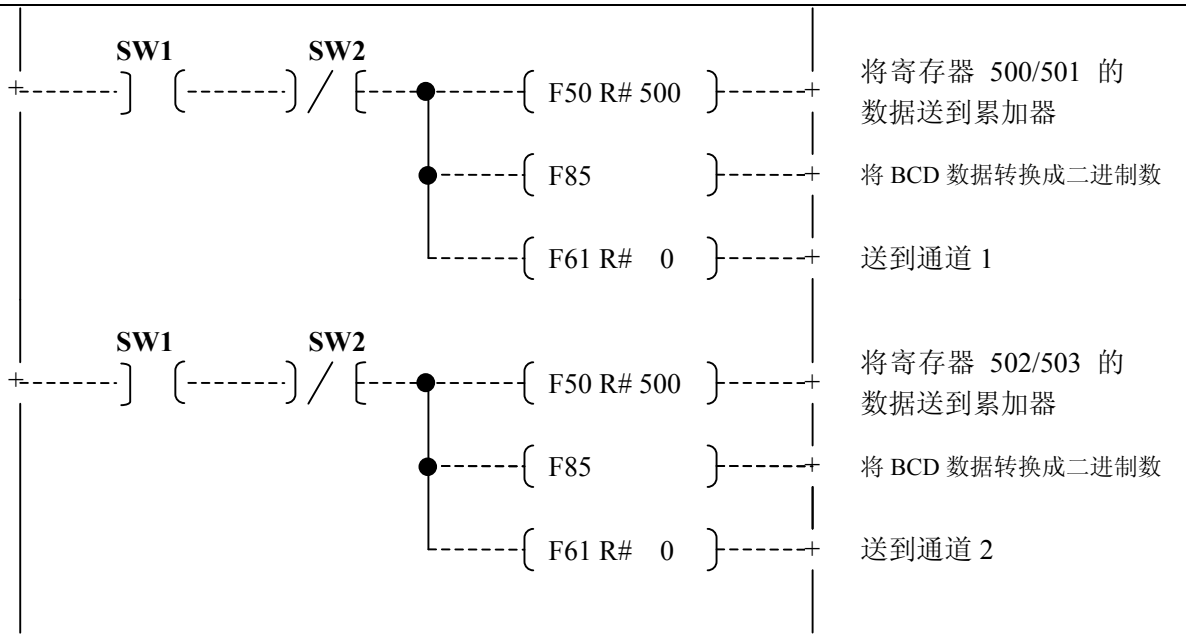
表 4 I/O 点定义（续）

I/O 点	定　　义		I/O 类型
	通道　数据	权	
100	通道 2——数据位 1	1	输出
101	通道 2——数据位 2	2	输出
102	通道 2——数据位 3	4	输出
103	通道 2——数据位 4	8	输出
104	通道 2——数据位 5	16	输出
105	通道 2——数据位 6	32	输出
106	通道 2——数据位 7	64	输出
107	通道 2——数据位 8	128	输出

六、 梯形图逻辑举例

下面的例子提供了将输出数据传送到通道 1 和通道 2 的方法。这是 SR—21 使用模拟量输出模块的编程例子。

例：输出数据到通道 1 和通道 2。这里利用 I/O 的定义号与表 4 相同，即 0—7 为通道 1，100—107 为通道 2。



七、模拟量输出模块部件

下图是模拟量输出模块的面板说明，标出了其部件及用户接线端。

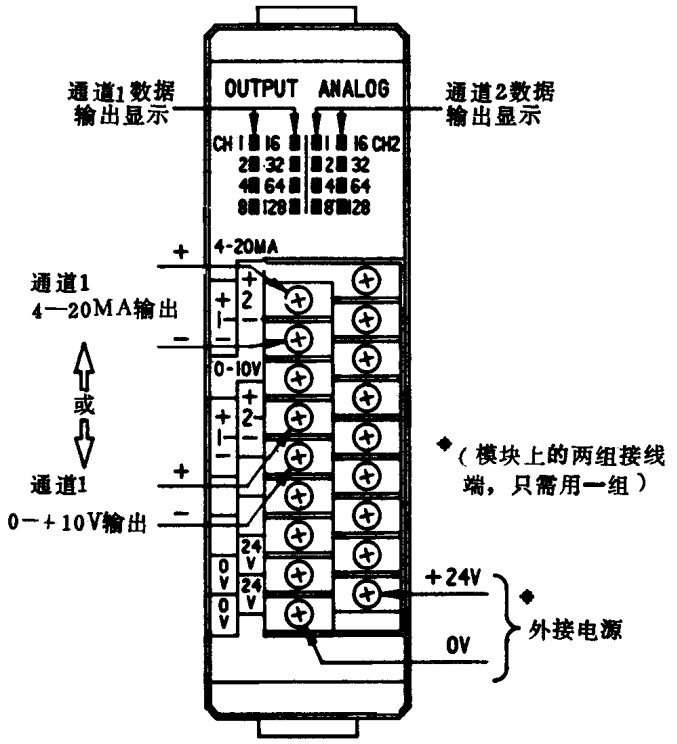


图 5.5 模拟量输出模块面板

模拟量输出模块 E—02DA

一、概述

E—02DA 模拟量输出模块有一个输出通道，该模块把 PLC 的 12 位二进制信号转换成 0—10V 和 4mA—20mA 的模拟量输出信号。

二、规格

项　　目	规　　格	
占用 I/O 点数	16 点	
通道数	1 路	
分辩率	1/4096（12 位）	
输出信号	电压输出 0—10V	电流输出：4—20mA
输出电压		11V 最小
输出阻抗	0.5Ω最大	
输出电流	5 mA 最大	
负载电阻	2KΩ最小	5—550Ω
输出型式	不平衡输出	不平衡输出
线性误差 温度精度 总合精度	±0.1%最大 ±0.007%/ ℃ ±0.2%最大(25℃)	在全量程条件下
转换时间	1mS 最大	
动作显示	LED 显示，以 12 位二进制码显示工作状态	
绝缘方式	光耦	
外部电源	电压　±15V±5%以内，容许波纹小于 1% 电流消耗小于 80mA	
内部 9V 消耗电流	80mA 最大	
重量	200 克	

三、地址分配

地址	数据	地址	数据
n+0	1	n+100	256
n+1	2	n+101	512
n+2	4	n+102	1024
n+3	8	n+103	2048
n+4	16	n+104	
n+5	32	n+105	
n+6	64	n+106	
n+7	128	n+107	

注：n 表示模块安装位置所确定的第一个 I/O 定义号

四、方框图

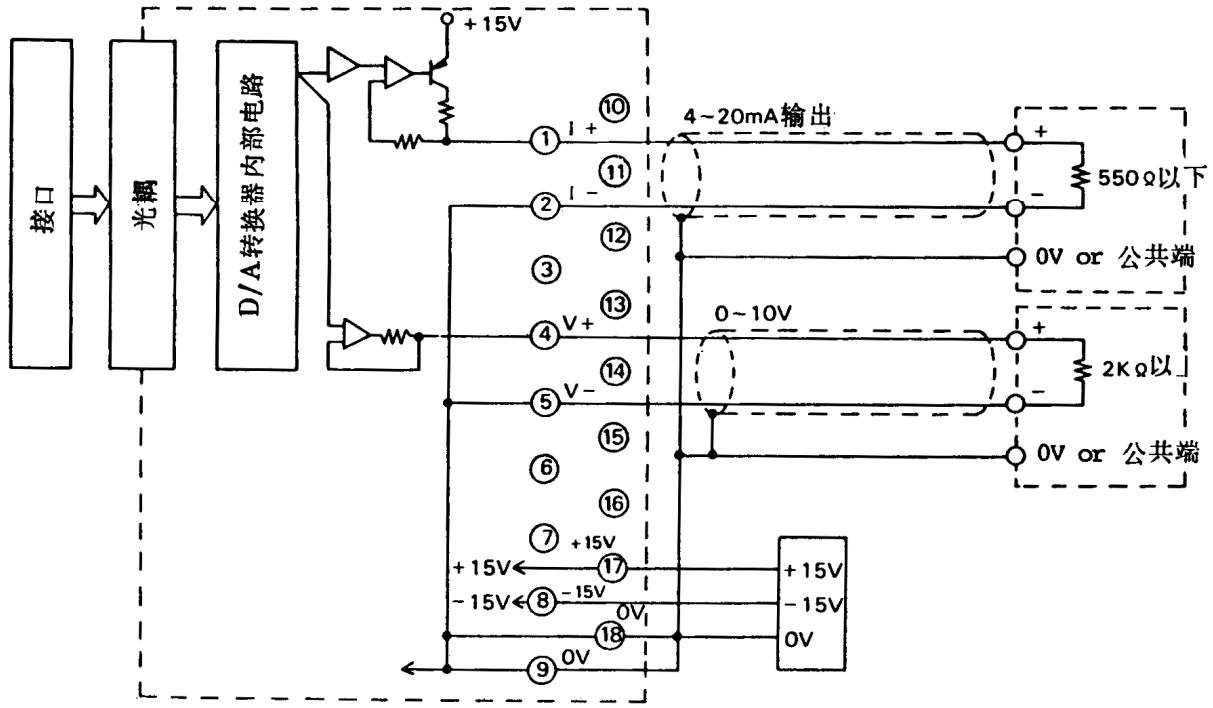


图 5.6 模拟量输出模块 E—02DA 方框图

五、使用注意事项：

1、电流输出（4~20mA）在信号接收侧加装一个负载电阻（250Ω ±0.1%）后就可作为 1~5V 的电压输出使用。

2、避免和其它负载使用同一±15V 电源，联合使用电源将对 E—02AD 有坏的影响。

3、请尽量不要使用开关电源。

4、请不要触摸印印制板上的调整用电位计。

六、面板图

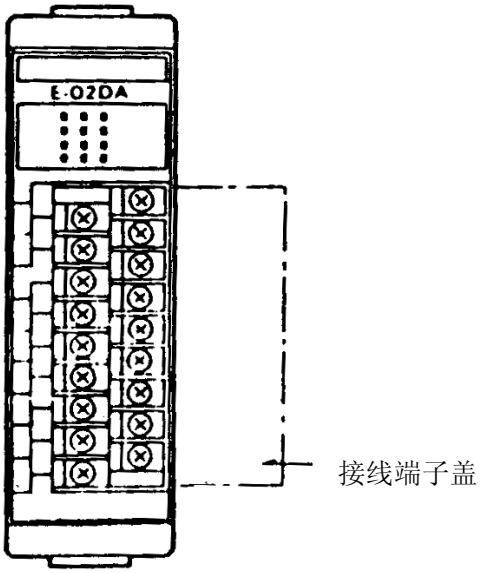
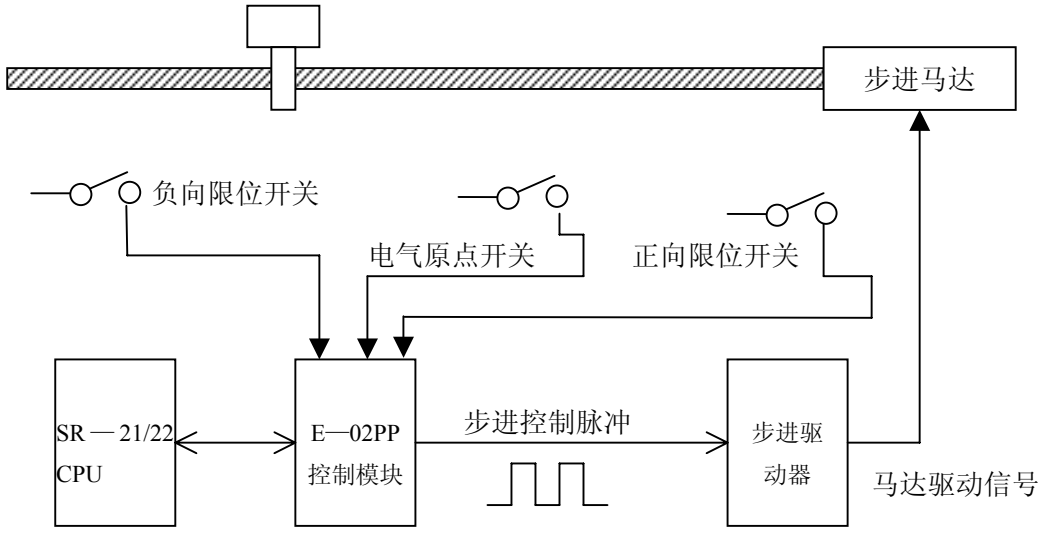


图 5.7 面板图

§5—7 步进电机单轴定位模块

步进电机单轴定位模块 E—02PP 安装在 SR 系列的电源框架中。每台 PLC 可带 E—02PP 模块 2 块。SR—21/22 PLC 可以通过 I/O 总线对 E—02PP 模块根据 PLC 的命令以及电机当时的运动状态，输出相应的速度控制信号，使步进电机按要求运转，达到位置控制的目的。



E—02PP 模块主要技术指标

项　目	技　术　指　标
占用 I/O 点数	8 点
控制轴数	单轴
位置指定方式	绝对、相对、间接指定
位置指定范围	—8388608~+8388607CT
定位目标带宽度	0~255 可编程
速度控制脉冲	NPN 晶体管 OC 输出、耐压 48V、电流 200mA，3 种输出方式
可编程速度	1~60000vu，即 1.67Hz~100KHz
可编程加速度	1~500000au
运动方式	a．调整运动，b.自动找原点,c.定位运动,d.可变速运动
限位开关输入	4 点.DC24V 源输入
定位命令存储器	可存放 256 条命令
子程序存储器	可存放 10 个子程序最多包含 512 条命令

详细资料请参阅“E—02PP 步进电机单轴定位模块用户手册”。

第六章 维　护

§6—1 维护过程概述

SR—21 设计成可长期不间断地工作。但是，偶尔有的地方也需要对动作进行修正，迅速找到这个场所并修正它们是很重要的。修正发生在 PLC 以外的动作需要许多时间。

§6—2 查找故障的设备

SR—21 的指示灯及机内装备，有益于对 PLC 整个控制系统查找故障。编程器是主要的诊断工具，它能方便地插到 PLC 上面。在编程器上可以观察整个控制系统的状态，当您去查找以 SR—21 为核心的控制系统的故障时，作为一个习惯，你应带一个编程器。

§6—3 基本的查找故障顺序

提出下列问题，并根据发现的合理动作逐个否定。一步一步地更换 SR—21 中的各种模块，直到故障全部排除。所有主要的修正动作能通过更换模块来完成。除了一把螺丝刀和一个电压表或万用表外，并不需要特殊的工具。不需要示波器，高级精密电压表（数字电压表）或特殊的测试程序。图 6.1 为几个参考指示灯的位置。

1.PWR（电源）灯亮否？如果不亮，在采用交流电源的框架的电压输入端（98—162VAC 或 195—252VAC）检查电源电压；对于需要直流电源的框架，测量+24V 和 0V 端之间的直流电压，如果不是合适的 AC 或 DC 电源，则问题发生在 SR—21PLC 之外。如 AC 和 DC 电源电压正常，但 PWR 灯不亮，检查保险丝，如必要的话，就更换 CPU 框架。

2.CPU 灯亮否？如亮了，检查显示的出错代码，对照表 2.1 的错误代码定义，作相应的修正。

3.RUN（运行）灯亮否？如不亮，检查编程器是不是处于 PRG 或 LOAD 位置，或者是不是程序出错。如 RUN 灯不亮，而编程器并未插上，或者编程器处于 RUN 方式且没有显示出错代码，则要更换 CPU 模块。

4.BATT（电池）灯亮否？如亮，要更换锂电池。由于 BATT 灯只是个报警信号，即使电池电压过低，程序也可能尚未改变。更换电池以后，检查程序或让 PLC 试运行。如程序已有错，在完成系统编程初始化后，将录在磁带上的程序重新装入 PLC。

5.在多框架系统中，如果 CPU 是工作的，可用 RUN 继电器来检查其它几个电源的工作。如果 RUN 继电器未闭合（高阻态），按上面讲的第一步检查 AC 或 DC 电源。如 AC 或 DC 电源正常而继电器是断开的，则需要更换框架。

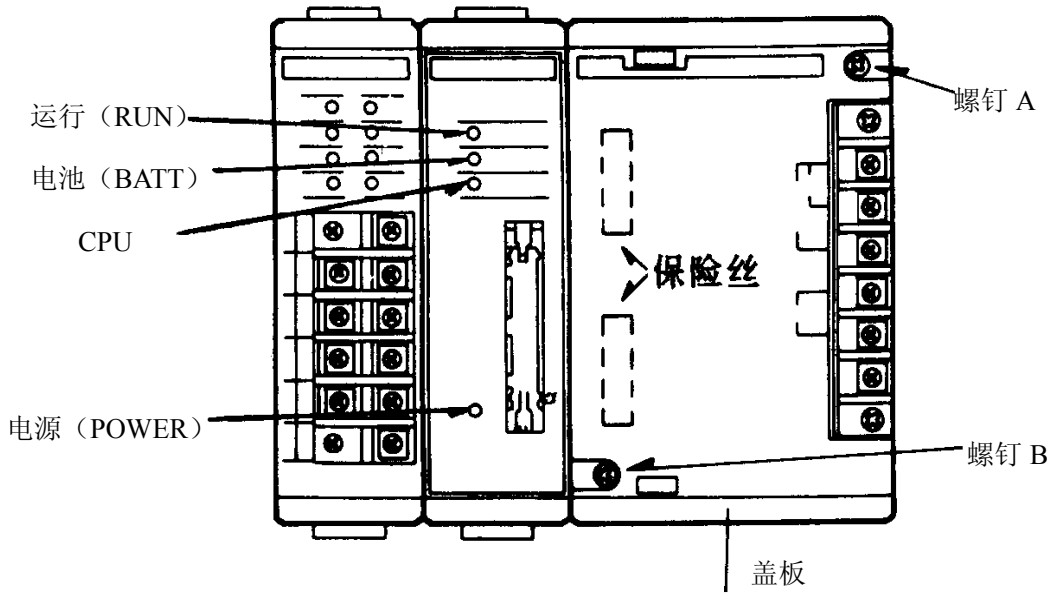


图 6.1 查找故障指示灯

§6—4 一般查找故障步骤

其它步骤与用户的逻辑知识有关。下面的一些步骤，实际上只是较普通的，对于您遇到的特定的应用问题，尚需修改或调整。查找故障的最好工具就是您的感觉和经验。首先，插上编程器，并将开关打到 RUN（运行）位置，然后按下列步骤进行。

1．如果 SR—21PC 停止在某些输出被激励的地方，一般是处于中间状态，则查找引起下一步操作发生的信号（输入，定时器，线圈，鼓轮控制器等）。编程器会显示那个信号的 ON/OFF 状态。

2．如果输入信号，将编程器显示的状态与输入模块的 LED 指示作比较，，结果不一致，则更换输入模块。如发现在扩展框架上有多个模块要变换，那么，在您更换模块之前，应先检查 I/O 扩展电缆和它的连接情况。

3．如输入状态和输入模块的 LED 灯显示一致，就要比较一下发光二极管与输入装置（按钮、限位开关等）的状态。如二者不同，测量一下输入模块电压（见第 5 章，I/O 接线）如发现有问题，需要更换 I/O 装置，现场接线或电源；否则，要更换输入模块。

4．如果信号是一个连接现场设备的输出线圈，则要比较它的状态与输出模块的 LED 显示的状态。如果它们不相同，检查现场电源的激励电压是否正常，如现场电源不正常，检查电源和它的接线；如现场电源正常，但 I/O 模块输出的状态有错，则要更换输出模块或检查框架提供给模块的电源是否正常。

5．如信号是线圈，没有输出或输出与线圈的状态不相同，就得用编程器检查输出的驱动逻辑，并检查程序清单。检查应从右向左进行，找出第一个不接通的触点，如没有通的那个信号是输入，就按第 2 和 3 步检查该输入点，如是线圈，就按第 4 和 5 步检查。要确使主控继电器不影响逻辑操作。

6．如果信号是定时器，而且停在小于 999.9 的非零值上，则要更换 CPU 模块。

7．如果该信号控制一个计数器，首先检查控制复位的逻辑，然后是计数信号。按上述 2 到 5 步进行。

§6—5 组件的更换

下面是更换 SR—21PC 系统组件的步骤。

一、更换框架

1．切断 AC 电源；如装有编程器，拔掉编程器。

2．从框架右端的接线端板上，拔下塑料盖板，拆去电源接线。

3．拔掉所有的 I/O 模块。如果原先在安装时有多个工作回路的话，不要搞乱 I/O 的接线，并记下每个模块的框架中的位置，以便重新插上时不至于搞错。

4．如果 CPU 框架，拔出 CPU 组件和填充模块。将它放在安全的地方，以便以后重新安装。

5．卸去底部的两个固定框架的螺丝，松开上部两个螺丝，但不用拆掉。

6．将框架向上推移一下，然后把框架向下拉出来放在旁边。

7．将新的框架从顶部螺丝上套进去。

8．装上底部螺丝将四个螺丝都拧紧。

9．插入 I/O 模块，注意位置要与拆下时一致。

警　告

如果模块插错位置，将会引起控制系统危险的或错误的操作，但不会损坏模块。

10．插入卸下的 CPU 的和填充模块。

11．在框架右边的接线端上重新接好电源接线，再盖上电源接线端的塑料盖。

12．检查一下电源接线是否正确，然后再通上电源。仔细地检查整个系统的工作，确保所有的 I/O 模块位置正确，程序没有变化。

二、CPU 模块的更换

1．切断电源，如插有编程器的话，把编程器拔掉。

2．向中间挤压 CPU 模块面板的上、下紧固扣，使它们脱出卡口。

3．把模块从槽中垂直拔出。

4．如果 CPU 上装着 EPROM 存储器，把 ERROM 拔下，装在新的 CPU 上。

5．首先将印刷线路板对准底部导槽。将新的 CPU 的模块插入底部导槽。

6．轻微地晃动 CPU 模块，使 CPU 模块对准顶部导槽。

7．把 CPU 模块插进框架，直到两个弹性锁扣扣进卡口。

8．重新插上编程器，并通电。

9．在对系统编程初始化后，把录在磁带上的程序重新装入。检查一下整个系统的操作。

三、I/O 模块的更换

1．切断框架和 I/O 系统的电源。

2．卸下 I/O 模块接线端上的塑料盖。拆下有故障模块的现场接线。

3．拆去 I/O 接线端的现场接线或卸下可拆卸式接插座，这要视模块的类型而定。给每根线贴上标签或记下安装连线的标记，以便于将来重新连接。

4．向中间挤压 I/O 模块面板上的上、下弹性锁扣，使它们脱出卡口。

5．垂直向上拔出 I/O 模块。

6．插入新的 I/O 模块，应首先使印刷板对准底部导槽。

7．轻微晃动模块，使 I/O 模块进入导槽。

8．把模块插进框架，直到两个弹性锁扣扣进卡口。

9．重新接上现场接线或装上可拆卸式接插座，然后装好塑料盖。

10．接通框架电源，然后接通 I/O 系统电源。检查系统操作，特别要注意更换的 I/O 模块。

四、更换锂电池

（一）、SR—21/22，锂电池的更换

如果 CMOS 存储器的保持电池需要更换，就遵循下列步骤。图 6.2 为 CPU 上电池、电池插头和电池紧固扣的位置。

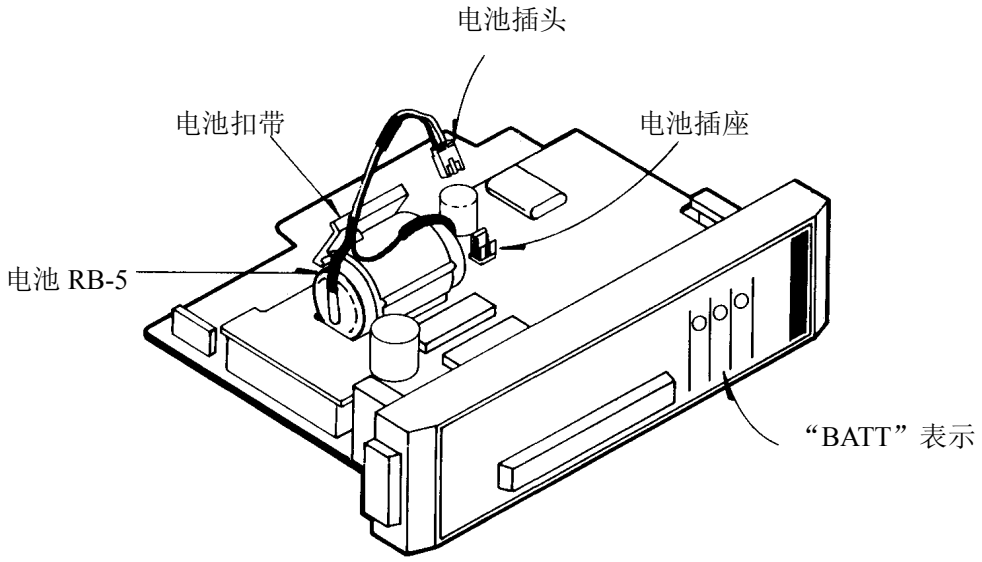


图 6.2SR—21/22 锂电池位置和连接

1.按前面讲的步骤拔出 CPU。

2.解开固定电池用的塑料紧固扣。

3.拔下电池,在没有电池的情况下,系统有足够的能力维持 CMOS 存储器的记忆约 20 分钟。

警　告

在处理锂电池时小心谨慎。不要随便丢弃失效的电池，不要试图给电池充电，不要使用电池短路，否则，可能会引起电池爆炸，燃烧释放有害物质。

4．接上新的电池（订货号为 RB—5），将它放进印刷板上合适的位置。

5．系上新的紧固扣或绝缘线。

6．把 CPU 插进框架。

7．检查一下 BATT 灯是不是灭了，如果必要的话，在系统编程初始化后，将录在磁带上的程序重新装入 CPU。然后再检查一下整个系统的操作。

8．如果 CPU 的两块印刷板分开了，必须使它们可靠地重新接合，然后装入框架，再通电，否则，逻辑可能总维持在高电流消耗方式，使电池过多地消耗。

（二）、SE—22 锂电池的更换

SE—22CPU 采用 FLASHROM 作为用户存储器，用户存储器不需锂电池进行停电保持。但是，如果内部线圈、数据寄存器等需具有停电保持功能的话，则需在 CPU 模块上装上 RB—9 锂电池。

如果出现了锂电池电压低的报警，CPU 面板上的 BATT 灯点亮，内部特殊继电器 SP377 为 ON 时，需要换锂电池。

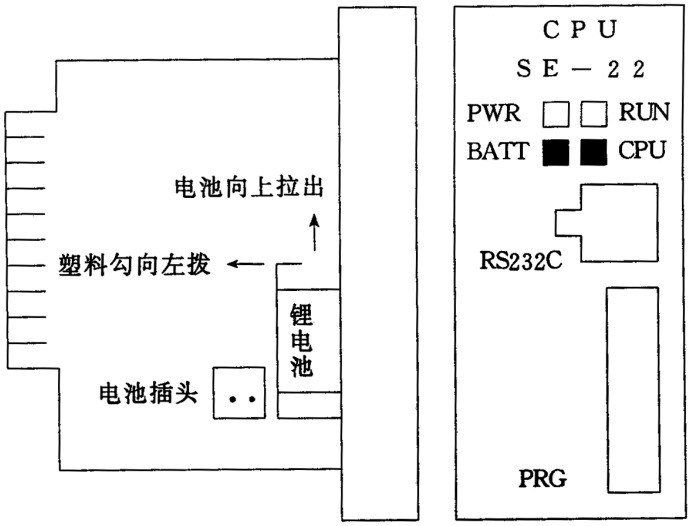


图 6.3　SE—22 锂电池位置和连接

电池更换操作的有关说明：

1、切断电源后拔下 CPU。

2、按上图所示方向和位置，朝左轻拨塑料勾，电池就可以向上拉出。

3、电池连接线可以拨出后，更换新电池，上面相反的方向装回电池。

4、在没有电池的情况下，系统有足够的能力维持 CMOS 存储器的记忆 10 分钟以上。

5、只有以下内容都不需要停电保持时，才可以拆除锂电池。

（1）计数器的计数经过值。

（2）级线圈和内部线圈中可以停电记忆的部分。

（3）数据寄存器可以停电记忆的部分，即除了位映射数据寄存器外的部分。

（4）系统参数区 040—077 部分 CMOS 领域的部分。

以下内容不需要靠锂电池维持

（1）定时器的定时经过值，无记忆。

（2）级线圈和内部线圈中不能停电记忆的部分无记忆。

（3）数据寄存器不能停电记忆的部分，即位映射数据寄存器中不需要停电记忆的部分。

（4）系统参数区 000—037 部分 FLASHROM 领域的部分。

6、系统运行方式与电池无关。

在安装了编程器时为编程器的开关位置方式。

没有安装编程器时自动为 RUN 运行方式。

§6—6 增设存储器

当对 SR—21PLC 增设存储器时，应遵循下列步骤。可以增设 CMOS RAM 存储器，使存储器容量从 700 字增加到 1724 字，也可以安装一块含有预先写入了程序的非易失的 PROM 存储器。

1．在安装增设的存储器之前，建议先将原先的程序录在磁带上。如果不这么做，在增设了存储器并对所有的存储器进行清零操作后，原先的程序就丢失了。

2．按前面的介绍，拔出 CPU。

3．附加存储器插座的位置在 CPU 大印刷电路板的尾部。

4．SR—21 用的是 HM6264LP—15，2K×8 位 CMOS RAM（或者是它们的等效型号）如果增设 PROM 存储器，SR—21 用 HN27256G—25（或等效的 PROM）。必须把可靠性好的 CMOS RAM 或 EPROM 用于 PLC。

注 意

当拿 CMOS 存储器 IC 时，总得拿着它的壳体，而不能拿在引脚上。在引脚上的静电能损坏内部线路。而这种损坏往往在几天或几个星期的操作中不会被发现。

5．IC 的方向，应是它一端的缺口对着存储器座的缺口。

6．在安装存储器时，为了方便，可能有必要将小印刷电路板拔出 3mm。不要把板分开。在存储器安装后，再使小板恢复到原先位置。

7．在把 IC 插进插座时应小心，用力均匀，不要弄弯了引脚。用肉眼仔细看一下，确保所有的引脚都对准位置，然后按下去，使 IC 插牢。如果必要，还得按表 3.2 调整跨接器与/或开关 2。

8．如果 CPU 的两块印刷电路板曾分开过、必须确保它们重新接合，然后才能插进框架并通电，否则，逻辑可能总处于高电流消耗方式，使电池损耗过多。

9．按先前的介绍安装 CPU 模块。

10．接通 CPU 电源，把方式开关置于 PRG 位置，进行清除全部存储内容操作（CLR SHF 348 DEL NXT）。这样，整个存储器的全部数据都清零了，把原先录在磁带上的程序重新装入 CPU 或输入一段新的程序。

附录 位映射数据寄存器

定义号及寄存器号是八进制的

16 位 T/C 经过值寄存器号与 T/C 定义号相同

方括号包括的部分表示可以停电记忆，*表示特殊功能线圈。

类型	八 进 制 定 义 号								寄存器号
输入及输出	007	006	005	004	003	002	001	000	R000
	017	016	015	014	013	012	011	010	R001
	027	026	025	024	023	022	021	020	R002
	037	036	035	034	033	032	031	030	R003
	047	046	045	044	043	042	041	040	R004
	057	056	055	054	053	052	051	050	R005
	067	066	065	064	063	062	061	060	R006
	077	076	075	074	073	072	071	070	R007
	107	106	105	104	103	102	101	100	R010
	117	116	115	114	113	112	111	110	R011
	127	126	125	124	123	122	121	120	R012
	137	136	135	134	133	132	131	130	R013
	147	146	145	144	143	142	141	140	R014
	157	156	155	154	153	152	151	150	R015
	167	166	165	164	163	162	161	160	R016
	177	176	175	174	173	172	171	170	R017
	707	706	705	704	703	702	701	700	R070
	717	716	715	714	713	712	711	710	R071
	727	726	725	724	723	722	721	720	R072
	737	736	735	734	733	732	731	730	R073
	747	746	745	744	743	742	741	740	R074
	757	756	755	754	753	752	751	750	R075
	767	766	765	764	763	762	761	760	R076

类型	八 进 制 定 义 号								寄存器号
输入及输出	207	206	205	204	203	202	201	200	R020
	217	216	215	214	213	212	211	210	R021
	227	226	225	224	223	222	221	220	R022
	237	236	235	234	233	232	231	230	R023
	247	246	245	244	243	242	241	240	R024
	257	256	255	254	253	252	251	250	R025
	267	266	265	264	263	262	261	260	R026
	277	276	275	274	273	272	271	270	R027
	307	306	305	304	303	302	301	300	R030
	317	316	315	314	313	312	311	310	R031
	327	326	325	324	323	322	321	320	R032
	337	336	335	334	333	332	331	330	R033
	347	346	345	344	343	342	341	340	R034
	357	356	355	354	353	352	351	350	R035
	367	366	365	364	363	362	361	360	R036
	377*	376*	375*	374*	373	372	371	370	R037
	777*	776*	775*	774*	773*	772*	771*	770*	R077
	1007*	1006*	1005*	1004*	1003*	1002*	1001*	1000*	R100
	1017*	1016*	1015*	1014*	1013*	1012*	1011*	1010*	R101
	1027	1026	1025	1024	1023	1022	1021	1020	R102
	1037	1036	1035	1034	1033	1032	1031	1030	R103
	1047	1046	1045	1044	1043	1042	1041	1040	R104
	1057	1056	1055	1054	1053	1052	1051	1050	R105
	1067	1066	1065	1064	1063	1062	1061	1060	R106
	1077	1076	1075	1074	1073	1072	1071	1070	R107

类型	八 进 制 定 义 号								寄存器号
内部线圈	1107	1106	1105	1104	1103	1102	1101	1100	R110
	1117	1116	1115	1114	1113	1112	1111	1110	R111
	1127	1126	1125	1124	1123	1122	1121	1120	R112
	1137	1136	1135	1134	1133	1132	1131	1130	R113
	1147	1146	1145	1144	1143	1142	1141	1140	R114
	1157	1156	1155	1154	1153	1152	1151	1150	R115
	1167	1166	1165	1164	1163	1162	1161	1160	R116
	1177	1176	1175	1174	1173	1172	1171	1170	R117
	1207	1206	1205	1204	1203	1202	1201	1200	R120
	1217	1216	1215	1214	1213	1212	1211	1210	R121
	1227	1226	1225	1224	1223	1222	1221	1220	R122
	1237	1236	1235	1234	1233	1232	1231	1230	R123
	1247	1246	1245	1244	1243	1242	1241	1240	R124
	1257	1256	1255	1254	1253	1252	1251	1250	R125
	1267	1266	1265	1264	1263	1262	1261	1260	R126
	1277	1276	1275	1274	1273	1272	1271	1270	R127
	1307	1306	1305	1304	1303	1302	1301	1300	R130
	1317	1316	1315	1314	1313	1312	1311	1300	R131
	1327	1326	1325	1324	1323	1322	1321	1320	R132
	1337	1336	1335	1334	1333	1332	1331	1330	R133
	1347	1346	1345	1344	1343	1342	1341	1340	R134
	1357	1356	1355	1354	1353	1352	1351	1350	R135
	1367	1366	1365	1364	1363	1362	1361	1360	R136
	1377	1376	1375	1374	1373	1372	1371	1370	R137

类型	八 进 制 定 义 号								寄存器号
移位寄存器	407	406	405	404	403	402	401	400	R040
	417	416	415	414	413	412	411	410	R041
	427	426	425	424	423	422	421	420	R042
	437	436	435	434	433	432	431	430	R043
	447	446	445	444	443	442	441	440	R044
	457	456	455	454	453	452	451	450	R045
	467	466	465	464	463	462	461	460	R046
	477	476	475	474	473	472	471	470	R047
	507	506	505	504	503	502	501	500	R050
	517	516	515	514	513	512	511	510	R051
	527	526	525	524	523	522	521	520	R052
	537	536	535	534	533	532	531	530	R053
	547	546	545	544	543	542	541	540	R054
	557	576	555	574	573	572	571	550	R055
	567	566	565	564	563	562	561	560	R056
	577	576	575	574	573	572	571	570	R057
定时计数线圈	607	606	605	604	603	602	601	600	R060
	617	616	615	614	613	612	611	610	R061
	627	626	625	624	623	622	621	620	R062
	637	636	635	634	633	632	631	630	R063
	647	646	645	644	643	642	641	640	R064
	657	656	655	654	653	652	651	650	R065
	667	666	665	664	663	662	661	660	R066
	677	676	675	674	673	672	671	670	R067

作计数器使用或未使用时可停电记忆

作定时器使用不可停电记忆

类型	八 进 制 定 义 号								寄存器号
级线圈	1407	1406	1405	1404	1403	1402	1401	1400	R140
	1417	1416	1415	1414	1413	1412	1411	1410	R141
	1427	1426	1425	1424	1423	1422	1421	1420	R142
	1437	1436	1435	1434	1433	1432	1431	1430	R143
	1447	1446	1445	1444	1443	1442	1441	1440	R144
	1457	1456	1455	1454	1453	1452	1451	1450	R145
	1467	1466	1465	1464	1463	1462	1461	1460	R146
	1477	1476	1475	1474	1473	1472	1471	1470	R147
	1507	1506	1505	1504	1503	1502	1501	1500	R150
	1517	1516	1515	1514	1513	1512	1511	1510	R151
	1527	1526	1525	1524	1523	1522	1521	1520	R152
	1537	1536	1535	1534	1533	1532	1531	1530	R153
	1547	1546	1545	1544	1543	1542	1541	1540	R154
	1557	1556	1555	1554	1553	1552	1551	1550	R155
	1567	1566	1565	1564	1563	1562	1561	1560	R156
	1577	1576	1575	1574	1573	1572	1571	1570	R157

光洋电子(无锡)有限公司
Koyo ELECTRONICS （WUXI） CO., LTD.
地址：江苏省无锡市蠡溪路 118 号　邮编：214072
电话：0510-5167888　传真：0510-5161393
http：//www.koyoele.com.cn

KEW-M1211B

2001 年 9 月