PLC电气控制与组态设计

严俊

第一章 可编程控制器的基本知识

第一节 可编程控制器的产生和发展

一、可编程控制器的产生

可编程序控制器问世于1969年。是美国汽车制造工业激烈竞争的结果。更新汽车型号必然要求加工生产线改变。正是从汽车制造业开始了对传统继电器控制的挑战。1968年美国General Motors公司，要求制造商为其装配线提供一种新型的通用程序控制器，并提出10项招标指标。这就是著名的GM 10条。

GM10 条是可编程序控制器出现的直接原因：

1. 编程简单，可在现场修改程序；
2. 可靠性高于继电器控制柜；
3. 体积小于继电器控制柜；
4. 维护方便，最好是插件式；
5. 可将数据直接送入管理计算机；
6. 在成本上可与继电器控制柜竞争；
7. 输入可以是交流 115V；
8. 输出为交流 115V、2A 以上，能直接驱动电磁阀等；
9. 在扩展时，原系统只需很小变更；
10. 用户程序存贮器容量至少能扩展到 4K 。

a. 可编程控制器的发展及定义

1969 年，美国数据设备公司 (DEC) 研制出世界上第一台可编程控制器，并成功地应用在 GM 公司的生产线上这一时期它**主要用于顺序控制，只能进行逻辑运算，故称为可编程逻辑控制器，简称** PLC(Programmable Logic Controller)。

70 年代后期，随着微电子技术和计算机技术的迅猛发展，使 PLC 从开关量的逻辑控制扩展到数字控制及生产过程控制领域，**真正成为一种电子计算机工业控制装置，** 故称为可编程控制器，简称 PC(Programmable Controller)。但由于 PC 容易和个人计算机 (Personal Computer) 相混淆，故人们仍**习惯地用 PLC 作为可编程控制器的缩写。**

1985 年 1 月国际电工委员会的定义：

“可编程序控制器是一种数字运算的电子系统，**专为工业环境下应用而设计**。它采用可编程序的存储器，用来在内部存储执行逻辑运算、顺序控制、定时、计数和算术运算等操作的指令，并通过数字式、模拟式的输入和输出，控制各种类型的机械或生产过程。可编程序控制器及其有关设备，都应按**易于与工业控制系统联成一个整体，易于扩充的原则设计**”。

•PLC 与传统的继电器逻辑相比

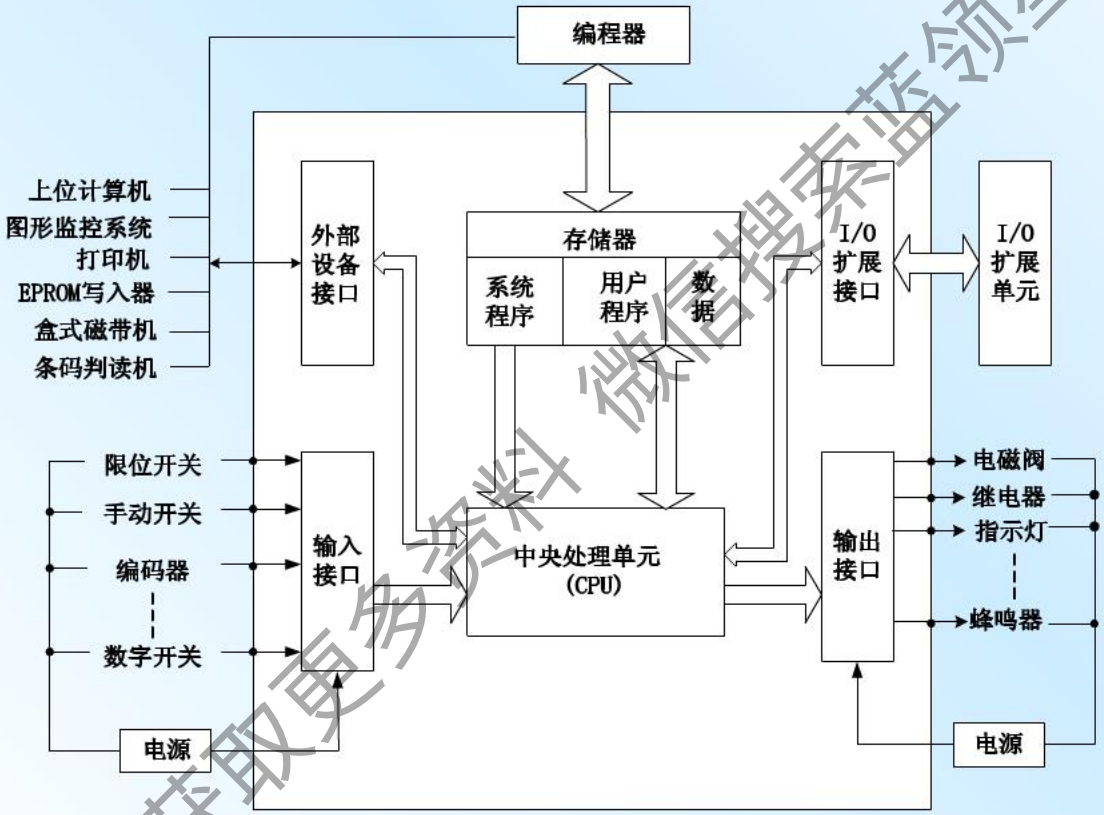
- 可靠性高、逻辑功能强、体积小。
- 在需要大量中间继电器、时间继电器及计数继电器的场合， PLC 无需增加硬设备。
- **随着要求的变更 PLC 对程序修改方便。继电器线路要想改变控制功能，必须变更硬接线，灵活性差。**
- 具有网络通讯功能，可附加高性能模块对模拟量进行处理，实现各种复杂控制功能。

a. PLC 与工业控制计算机相比

1. **PLC 继承了继电器系统的基本格式和习惯**，对于有继电器系统方面知识和经验的人来说，尤其是现场的技术人员，学习起来十分方便。
2. PLC 一般是由电气控制器的制造厂家研制生产，**各厂家的产品不通用。工业控制机**是由通用计算机推广应用发展起来的，一般由微机厂、芯片及板卡制造厂开发生产。它在硬件结构方面的突出优点是**总线标准化程度高，产品兼容性强。**
3. PLC 的运行方式与工业控制机不同，**微机的许多软件不能直接使用。**工业控制机可使用通用微机的各种编程语言，对要求快速、实时性强、模型复杂的工业对象的控制占有优势。但它要求使用者具有一定的计算机专业知识。

1. PLC 和工业控制机都是专为工业现场应用环境而设计的。**都具有很高的可靠性。**
2. PLC 一般具有模块结构，**可以针对不同的对象进行组合和扩展。**

b. PLC 的系统结构



- PLC 各部分的作用

1.CPU

a. 诊断 PLC 电源、内部电路的工作状态及编制程序中的语法错误。

b. 采集现场的状态或数据，并送人 PLC 的寄存器中。

c. 逐条读取指令，完成各种运算和操作。

d. 将处理结果送至输出端。

e. 响应各种外部设备的工作请求。

a. PLC 各部分的作用

–存储器

a. **系统程序存储器：** 用以存放系统管理程序、监控程序及系统内部数据。 PLC 出厂前已将其固化在只读存储器 ROM 或 PROM 中，用户不能更改。

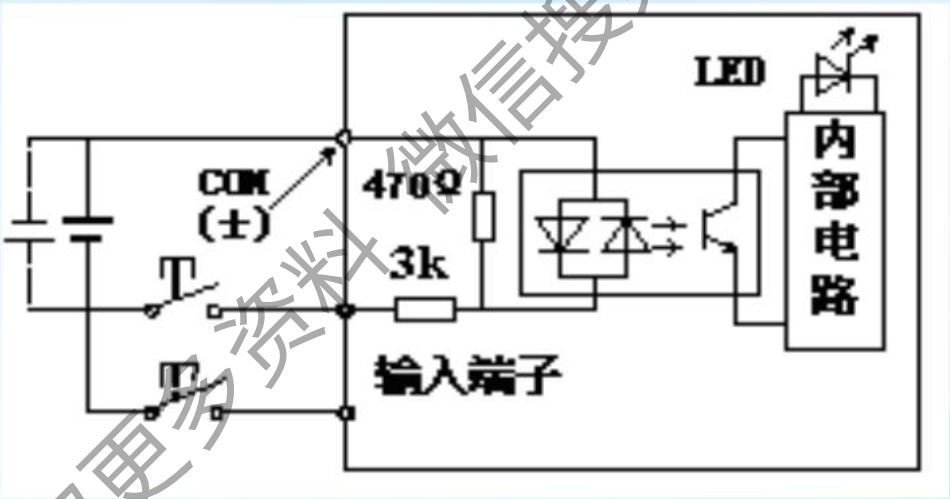
b. **用户存储器：** 包括用户程序存储区及工作数据存储区。这类存储器一般由低功耗的 CMOS-RAM 构成，其中的存储内容可读出并更改。

注意： PLC 产品手册中给出的“存储器类型”和“程序容量”是针对用户程序存储器而言的。

a. PLC 各部分的作用

1. 输入输出接口电路

输入接口电路： 采用光电耦合电路，将限位开关、手动开关、编码器等现场输入设备的**控制信号转换**成 CPU 所能接受和处理的**数字信号**。

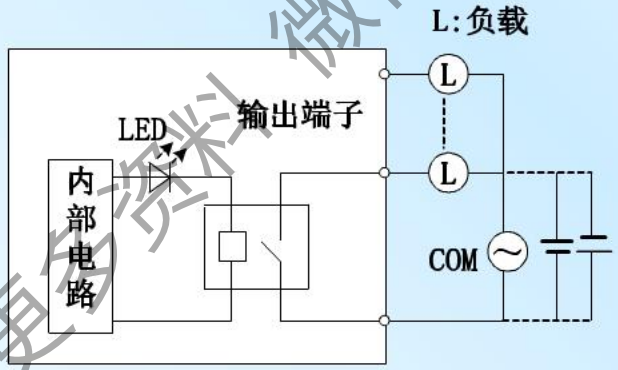


PLC 的输入接口电路（直流输入型）

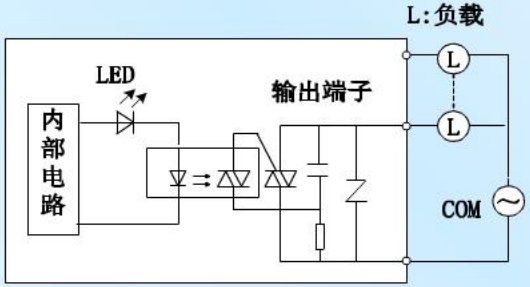
•PLC 各部分的作用

输出接口电路： 采用光电耦合电路，**将 CPU 处理过的信号转换成现场需要的强电信号输出，** 以驱动接触器、电磁阀等外部设备的通断电。**有三种类型：**

a. **继电器输出型：** 为有触点输出方式，用于接通或断开**开关频率较低的直流负载或交流负载回路**。

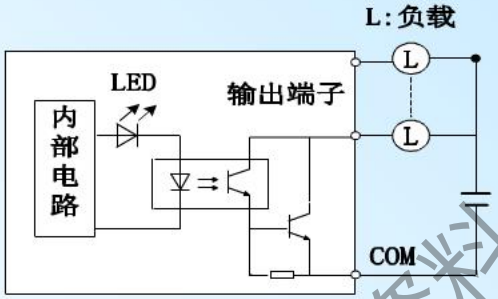


继电器输出型

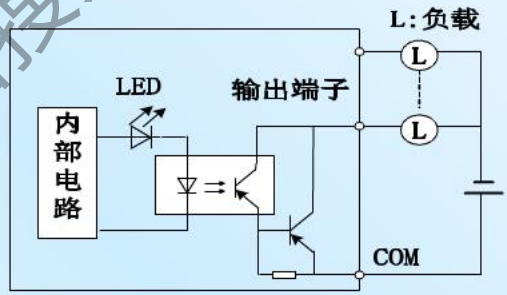


（b） 晶闸管输出型

- **晶闸管输出型：**为无触点输出方式，用于接通或断开**开关频率较高的交流电源负载。**



（c） 晶体管输出型
（NPN 集电极开路）



（d） 晶体管输出型
（PNP 集电极开路）

- **晶体管输出型：**为无触点输出方式，用于接通或断开**开关频率较高的直流电源负载。**

– 电源

PLC 的电源是指将外部输入的交流电处理后转换成满足 PLC 的 CPU、存储器、输人输出接口等内部电路工作需要的**直流电源电路或电源模块**。许多 PLC 的直流电源采用直流开关稳压电源，不仅可提供多路独立的电压**供内部电路使用，** 而且**还可为输入设备提供标准电源。**

－手持编程器

手持编程器采用**助记符语言编程**，具有编辑、检索、修改程序、进行系统设置、内存监控等功能。可一机多用，具有使用方便、价格低廉的特点。

缺点：不够直观

可通过 PLC 的 RS232 外设通讯口（或 RS422 口配以适配器）与计算机联机，**利用专用工具软件**（NPST－GR、FPSOFT、FPWIN－GR）**对 PLC 进行编程和监控**。利用计算机进行编程和监控比手持编程工具更加直观和方便。

1. 输入输出 I / O 扩展接口

若主机单元的 I / O 点数不能满足需要时，可通过此接口用扁平电缆线将 I / O 扩展单元与主机相连，以增加 I / O 点数。PLC 的最大扩展能力主要受 CPU 寻址能力和主机驱动能力的限制。

第三节 可编程控制器的原理及技术性能

c. PLC 的基本工作原理

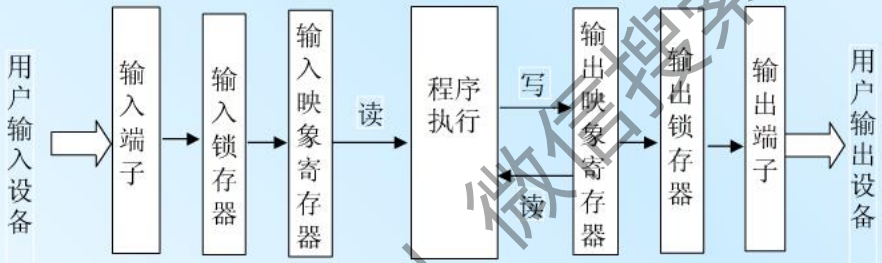
微机：**等待命令**的工作方式

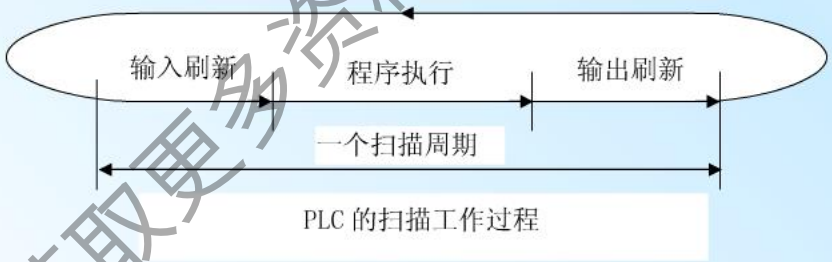
PLC：**循环扫描**的工作方式

CPU 从第一条指令开始按指令步序号作周期性的循环扫描，如果**无跳转指令，则从第一条指令开始逐条顺序执行用户程序，直至遇到结束符后又返回第一条指令，周而复始不断循环，每一个循环称为一个扫描周期。**

一个扫描周期主要分为三个阶段：

1. 输入刷新阶段
2. 程序执行阶段
3. 输出刷新阶段





a. PLC 的基本工作原理

由于输入刷新阶段是紧接输出刷新阶段后马上进行的，所以亦将这**两个阶段统称为 I / O 刷新阶段**。实际上，除了执行程序和 I / O 刷新外，PLC 还要进行各种错误检测（自诊断功能）并与编程工具通讯，这些操作统称为“监视服务”。一般在程序执行后进行。

扫描周期的长短主要取决于程序的长短。

由于每一个扫描周期只进行一次 I / O 刷新，故使系统**存在输入、输出滞后现象。**这对于一般的开关量控制系统不但不会造成影响，**反而可以增强系统的抗干扰能力。**但对于控制时间要求较严格、响应速度要求较快的系统，就需要精心编制程序，必要时采用一些特殊功能，以减少因扫描周期造成的响应滞后。

a. PLC 的主要技术指标

- 输入 / 输出点数（ I/O 点数）

–内存容量

注意： **“内存容量”实际是指用户程序容量，不包括系统程序存储器的容量。**

4. 扫描速度 （单位： ms / k 或 μs/ 步。 ）

5. 指令条数

6. 内部继电器和寄存器数目

7. 编程语言及编程手段

8. 高级模块

主控模块可实现基本控制功能，**高级模块可实现一些特殊的专门功能。**如 A / D 和 D / A 转换模块等。

a. PLC 的内存分配及 I / O 点数

- **I / O 继电器区：** I / O 区的寄存器**可直接与 PLC 外部的输入、输出端子传递信息**，具有“继电器”的功能，有自己的“线圈”和“触点”。故常称为“I / O 继电器区”。
- **内部通用继电器区：****只能在 PLC 内部使用，**其作用与中间继电器相似，在程序控制中可存放中间变量。
- **数据寄存器区：****只能按字使用**，不能按位使用。一般只用来**存放各种数据。**
- **特殊继电器、寄存器区：****被系统内部占用**，专门用于某些特殊目的，一般**不能由用户任意占用。**
- **系统寄存器区：**用来**存放各种重要信息和参数。**通过用户程序，不能读取和修改系统寄存器的内容。

第四节 PLC的分类及功能

b. PLC 的分类

1. 按结构形式分类

a. 整体式

b. 模块式

2. 按功能分类

a. 低档机

b. 中档机

c. 高档机

3. 按 I / O 点数和程序容量分类

分 类	I / O 点数	程序容量
超小型机	64 点以内	256 ~ 1000 字节
小型机	64 ~ 256	1 ~ 3.6K 字节
中型机	256 ~ 2048	3.6 ~ 13K 字节
大型机	2048 以上	13K 字节以上

a. PLC 的主要功能

1. 条件控制功能
2. 定时／计数控制功能
3. 数据处理功能
4. 步进控制功能
5. A／D与D／A转换功能
6. 运动控制功能
7. 过程控制功能
8. 扩展功能
9. 远程I／O功能
10. 通信联网功能
11. 监控功能

第五节 PLC的特点、应用场合和发展趋势

b. PLC 的主要特点

- 可靠性高、抗干扰能力强。主要有以下几个方面：
- **隔离（采用光电耦合器）**
- 滤波
- 对 PLC 的内部电源采取了屏蔽、稳压、保护等措施。
- 设置了连锁、环境检测与诊断、 Watchdog 等电路。
- 利用系统软件定期进行系统状态、用户程序、工作环境和故障检测。
- 对用户程序及动态工作数据进行电池备份。
- **采用密封、防尘、抗振的外壳封装结构。**
- 以集成电路为基本元件，内部处理过程不依赖于机械触点。**采用循环扫描的工作方式**，也提高了抗干扰能力。

1. 可实现三电一体化

将电控(逻辑控制)、电仪(过程控制)和电结(运动控制)集于一体，可以方便、灵活地组合成各种不同规模和要求的控制系统。

3. 编程简单、使用方便、柔性好

4. 体积小、重量轻、功耗低

a. PLC 的应用场合

- **逻辑控制** ：可取代传统继电器系统和顺序控制器。如各种机床、自动电梯、装配生产线、电镀流水线、运输和检测等的控制。
- **运动控制** ：可用于精密金属切削机床、机械手、机器人等设备的控制。
- **过程控制** ：通过配用 A / D 、 D / A 转换模块及智能 PID 模块实现对生产过程中的温度、压力、流量、速度等连续变化的模拟量进行闭环调节控制。
- **数据处理**
- **多级控制** ：利用 PLC 的网络通信功能模块及远程 I / O 控制模块实现多台 PLC 之间、 PLC 与上位计算机的链接，以完成较大规模的复杂控制。

a. 可编程控制器的发展趋势

1. 在系统构成规模上**向大、小两个方向发展**；
2. 功能不断增强，各种应用模块不断推出；
3. 产品更加**规范化、标准化**。

第六节 PLC的几种编程语言

不采用微机的编程语言，采用梯形图语言、指令助记符语言、控制系统流程图语言、布尔代数语言等。其中**梯形图、指令助记符语言最为常用。**

PLC的设计和生产至今尚无国际统一标准，**不同厂家所用语言和符号也不尽相同。**但它们的梯形图语言的基本结构和功能是**大同小异**的。

a. 梯形图语言

梯形图是在原继电器—接触器控制系统的继电器梯形图基础上演变而来的**一种图形语言**。**它是目前用得最多的 PLC 编程语言。**

注意：梯形图表示的并**不是一个实际电路而只是一个控制程序，**其间的连线表示的是它们之间的逻辑关系，即所谓“软接线”。

常开触点 ：⊣ ⊢

常闭触点：⊣/⊢

线圈： ⊣ ⊢

注意：它们并非是物理实体，而是“软继电器”。每个“软继电器”仅对应 PLC 存储单元中的一位。该位状态为“ 1”时，对应的继电器线圈接通，其常开触点闭合、常闭触点断开；状态为“ 0”时，对应的继电器线圈不通，其常开、常闭触点保持原态。

a. 指令助记符语言

助记符语言类似于计算机汇编语言，用一些简洁易记的文字符号表达 PLC 的各种指令。同一厂家的 PLC 产品，其助记符语言与梯形图语言是相互对应的，可互相转换。

助记符语言常用于手持编程器中，梯形图语言则多用于计算机编程环境中。

第二章 松下电工可编程控制器

产品－FP1 介绍

FP1 是一种功能很强的小型机，它的某些技术性能是一些同档次机型的小型机所不具备的。**具有通常只在大型 PLC 中才具备的功能。**

通过主机上配有的 RS422 或 RS232 接口，可实现 PLC 与 PC 机之间的通信，将 PC 机上的梯形图程序直接传送到可编程控制器中去。

有近 200 条的指令。**数据处理功能比一般小型机强。**

b. FP1系列产品类型及构成

在FP系列产品中，**FP1属于小型PLC产品。**该产品系列有C14、C16、C24、C40、C56和C72型等多种规格。扩展单元有E8～E40四种规格。

以C字母开头代表主控单元（或称主机），以E字母开头代表扩展单元（或称扩展机）。后面跟的数字代表I/O点数。

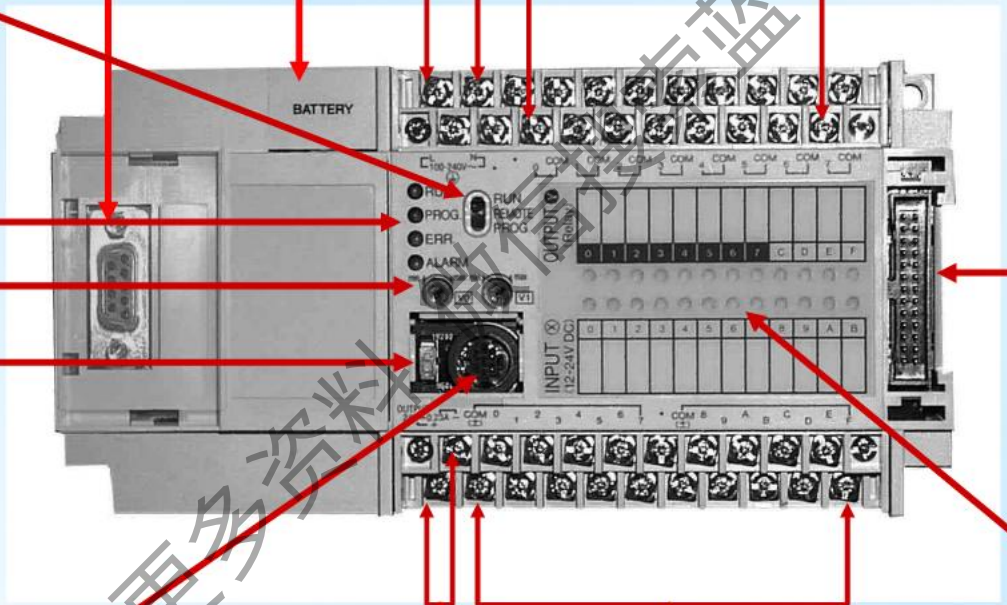
例如C24表示输入和输出点数之和为24。

表 2-1 FP1 系列 PLC 主要产品规格

简表

品名	类型	I / O 点数	内部寄存器	工作电压	输出形式
C14	标准型	8 / 6	EEPROM	DC24V 或 AC100 ~ 240V	继电器、晶体管（NPN、PNP）
C16	标准型	8 / 8			
C24 C24C	标准型 带 RS232 口和时钟 / 日历	16 / 8	RAM		
C40 C40C	标准型 带 RS232 口和时钟 / 日历	24 / 16			
C56 C56C	标准型 带 RS232 口和时钟 / 日历	32 / 24			
C72 C72C	标准型 带 RS232 口和时钟 / 日历	40 / 32			
E8		8 / 0 4 / 4 0 / 8	/	/	继电器、晶体管（NPN、PNP）
E16		16 / 0 8 / 8 0 / 16	/		
E24		16 / 8	/	DC24V 或 AC100 ~ 240V	
E40		24 / 16	/		

FP1 系列 C24 型 PLC 控制单元的外形图



1. RS232 口

只有 C24 、 C40 、 C56 和 C72 的 C 型机才配有。该口能与 PC 机通讯编程，也可连接其它外围设备。

2. 运行监视指示灯

- 当运行程序时，“ RUN”指示灯亮；
- 当控制单元中止执行程序时，“ PROG”指示灯亮；
- 当发生自诊断错误时，“ ERR”指示灯亮；
- 当检测到异常的情况时或出现“ Watchdog”定时故障时，“ ALARM”指示灯亮。

3. 电池座

4. 电源输入端子

FP1 型主机有交、直流电

- **工作方式选择开关**

  有三个工作方式档位，即“RUN”、“REMOTE”和“PROG”。

- **输出端子**

  该端子板为两头带螺丝可拆卸的板。**带“.”标记的端子不能作为输出端子使用。**

- **直流电源输出端子**

  在FP1系列**主机内部均配有一个供输入端使用的24V直流电源。**

- **输人端子**

  该端子板为两头带螺丝可拆卸的板。输入电压范围为直

- 编程工具连接插座（RS422 口）

可用此插座经专用外设电缆连接编程工具。

- 波特率选择开关
- 电位器（V0 、 V1）

这两个电位器可用螺丝刀进行手动调节，实现外部设定。当调节该电位器时， PLC 内部对应的特殊数据寄存器 DT9040 和 DT9041 的内容在 0 ～ 255 之间变化，相当于输入外部可调的模拟量。

- I / O 状态指示灯

用来指示输人 / 输出的通断状态。

- I / O 扩展单元接口插座

用于连接 FP1 扩展单元及 A / D 、 D / A 转换单元、链接单元。

a. FP1 系列可编程控制器的技术性能

可编程控制器的功能是否强大，很大程度上取决于它的技术性能。

表 2-2　FP1 系列 PLC 控制单元技术性能一览表

项　目		C14	C16	C24	C40	C56	C72
主机 I / O 点数		8 / 6	8 / 8	16 / 8	24 / 16	32 / 24	40 / 32
最大 I / O 点数		54	56	104	120	136	152
运行速度		1.6μs / 步					
程序容量		900 步		2720 步		5000 步	
程序存储器类型		EEPROM(无电池)		RAM（备用电池）和 EPROM			
指令数	基本	41		80		81	
	高级	85		111		111	
内部继电器 (R)		256 点		1008 点			
特殊内部继电器 (R)		64 点		64 点			

定时器 / 计数器 (T / C)		128 点	144 点	
数据寄存器 (DT)		256 字	1660 字	6144 字
特殊数据寄存器 (DT)		70 字	70 字	
索引寄存器 (IX 、 IY)		2 字	2 字	
主控指令 (MC / MCE) 点数		16 点	32 点	
跳转标记数 (LBL) 个数 (用于 JMP 、 LOOP 指令)		32 点	64 点	
微分点数 (DF 或 DF /)		点数不限制		
步进数		64 级	128 级	
子程序个数		8 个	16 个	
中断个数		/	9 个程序	
输入滤波时间		1 ~ 128ms		
自诊断功能		看门狗定时器，电池检测，程序检测		
特殊功能	高速计数	X0, X1 为计数输入，可加 / 减计数。单相输入时计数最高频率为 10KHZ ，两路两相输入时最高频率为 5KHZ 。 X2 为复位输入		
	手动拨盘寄存器	1 点	2 点	4 点
	脉冲捕捉输入	4 点	共 8 点	
	中断输入	/	共 8 点	
	定时中断	/	10ms ~ 30s 间隔	
	脉冲输出	1 点 (Y7)		2 点 (Y6 、 Y7)
		脉冲输出频率 : 45HZ ~ 4.9KHZ		
	固定扫描	2.5ms × 设定值 (160ms 或更小)		

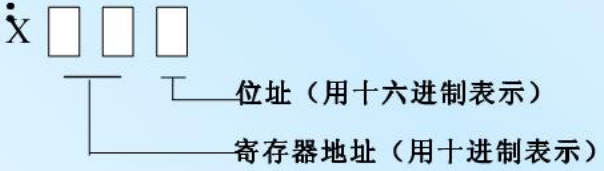
第二节　FP1 的内部寄存器及 I / O 配置

在使用 FP1 的 PLC 之前，了解 PLC 的 I / O 分配　以及内部寄存器的功能和配置是十分重要的。

X、WX 为 I / O 区的输入继电器，可直接与输入端子传递信息。Y、WY 为 I / O 区的输出继电器，可向输出端子传递信息。

名　称	符号（位 / 字）	编　号		
		C14 、 C16	C24 、 C40	C56 、 C72
输入继电器	X(bit)	208 点 : X0 ~ X12F		
	WX(word)	13 字 : WX0 ~ WX12		
输出继电器	Y(bit)	208 点 : Y0 ~ Y12F		
	WY(word)	13 字 : WY0 ~ WY12		
内部继电器	R(bit)	256 点 : R0 ~ R15F	1008 点 : R0 ~ R62F	
	WR(word)	16 字 : WR0 ~ WR15	63 字 : WR0 ~ WR62	
特殊内部继电器	R(bit)	64 点 : R9000 ~ R903F		
	WR(word)	4 字 : WR900 ~ WR903		
定时器	T(bit)	100 点 : T0 ~ T99		
计数器	C(bit)	28 点 : C100 ~ C127	44 点 : C100 ~ C143	
定时器 / 计数器设定值寄存器	SV(word)	128 字 : SV0 ~ SV127	144 字 : SV0 ~ SV143	
定时器 / 计数器经过值寄存器	EV(word)	128 字 : EV0 ~ EV127	144 字 : EV0 ~ EV143	
通用数据寄存器	DT(word)	256 字 : DT0 ~ DT255	1660 字 : DT0 ~ DT1659	6144 字 : DT0 ~ DT6143
特殊数据寄存器	DT(word)	70 字 : DT9000 ~ DT9069		
系统寄存器	(word)	No.0 ~ No.418		
索引寄存器	IX(word)	IX 、 IY 各一个		
	IY(word)			
十进制常数寄存器	K	16 位常数（字） : K - 32768 ~ K32767		
		32 位常数（双字） : K - 2147483648 ~ K2147483647		
十六进制常数寄存器	H	16 位常数（字） : H0 ~ HFFFF		
		32 位常数（双字） : H0 ~ HFFFFFFFF		

X 和 Y 是按位寻址的，而 WX 和 WY 只能按“字”寻址。 X 与的地址编号规则完全相同，下面以 X 为例说明如下：



如： X110 表示寄存器 WX11 中的第 0 位， X11F 表示寄存器 WX11 中的第 F 号位。图示如下：

WX11	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0

X11F　　X110

注意：字地址为 0 时可省略字地址数字，只给位地址 即可。

例：若 X4 为“ ON”，则 WX0 的第四位为“ 1”。

若 WY1=5 ，则表明 Y10 和 Y12 两个触点“ ON”。

表中 R 和 WR 的编号规则与 X 、 WX 和 Y 、 WY 相同。

a. 输入继电器

输入继电器的作用是将外部开关信号或传感器的信号输入到 PLC 。

注意：输入继电器只能由外部信号来驱动，而不能由内部指令来驱动，其触点也不能直接输出去驱动执行元件。

b. 输出继电器

输出继电器的作用是将 PLC 的执行结果向外输出，驱动外设（如接触器、电磁阀）动作。

注意：输出继电器必须是由 PLC 控制程序执行的结果来驱动。

c. 内部继电器

PLC 的内部寄存器供用户存放中间变量，**其作用与继电器－接触器控制系统中的中间继电器相似，**因此称为内部继电器（软继电器）。

a. 特殊内部继电器

R9000 ～ R903F 为特殊内部继电器，均有专门的用途，用户不能占用。这些继电器**不能用于输出，只能做内部触点用。**其主要功能是：

- 标志继电器
- 特殊控制继电器：例如，初始闭合继电器R9013，它的功能是只在运行中第一次扫描时闭合，从第二次扫描开始断开并保持打开状态。
- 信号源继电器

b. 定时器／计数器（T／C）

定时器（T）触点的通断由定时器指令（TM）的输出决定。如果定时器指令定时时间到，则与其同号的触点动作。

计数器（C）的触点是计数器指令（CT）的输出。如果计数器指令计数完毕，则与其同号的触点动作。

a. 定时器／计数器的设定值寄存器（SV）与经过值寄存器（EV）

SV 是存储定时器／计数器指令**预置值**的寄存器； EV 是存贮定时器／计数器**经过值**的寄存器。**EV 的值随着程序的运行而递减变化，** 当它的内容变为 0 时，定时器／计数器的触点动作。

每个定时器／计数器的编号都有一组 SV 和 EV 与之相对应 (表 2-4)

表 2-4　T／C 与 SV、EV 对应示意表

定时器／计数器编号	设定值寄存器 SV	经过值寄存器 EV
T0 ⋮ T99 C100 ⋮ C143	SV0 ⋮ SV99 SV100 ⋮ SV143	EV0 ⋮ EV99 EV100 ⋮ EV143

a. 通用数据寄存器（DT） 和特殊数据寄存器（DT）

通用数据寄存器用来存储各种数据。**它是纯粹的寄存器，不带任何触点。**

特殊数据寄存器是具有特殊用途的寄存器。每个数据寄存器由一个字（16-bit）组成。

b. 索引寄存器（IX 、 IY）

在 FP1 系列的 PLC 内部有两个 16 位的索引寄存器IX和IY 。其作用有以下两类：

1. 作数据寄存器使用

作为数据寄存器使用时，可作为 16-bit 寄存器单独使用；

当用作 32-bit 寄存器时，IX作低 16-bit， IY 作高 16-bit；

作为 32-bit 操作数编程时，如果指定IY为低 16-bit，则

a. 索引寄存器（IX 、 IY）

2. 其它操作数的修正值

b. 地址修正值功能（适用于 WX 、 WY 、 WR 、 SV 、 EV 和 DT）

例：有指令为 [FO　MV ， DT1 ， IXDT100]，执行后的结果为：

当IX＝ K30 时， DT1 中的数据被传送至 DT130 。

当IX＝ K50 时， DT1 中的数据被传送至 DT150 。

② 常数修正值功能（对 K 和 H）

例：有指令为 [FO　MV ， IXK30 ， DT100]，执行后的结果为：

当IX＝ K20 时，传送至 DT100 内容为 K50 。

当IX＝ K50 时，传送至 DT100 内容为 K80

注意：索引寄存器不能用索引寄存器来修正；当索引寄存器用作地址修正值时，要确保修正后的地址不要超出有效范围；当索引寄存器用作常数修正值时，修正后的值可能上溢或下溢。

a. 常数寄存器（K、H）

常数寄存器主要用来存放 PLC 输入数据，**十进制常数以数据前加字头 K 来表示，十六进制常数用数据前加字头 H 来表示。**

控制单元、初级扩展单元、次级扩展单元、I / O 链接单元和智能单元 (A / D 转换单元和 D / A 转换单元) 的 I / 0 分配是固定的。

FP1 系列 PLC 的 I / O 点数共有 416 点（输入 X0 ～ X12F 共 208 点，输出 Y0 ～ Y12F 也是 208 点），但受外部接线端子和主机驱动能力的限制，**最多可扩展 152 点（C72 型），其余的可作内部寄存器使用。**

品　种	型　号		输入端编号	输出端编号
控制单元	C14		X0 ~ X7	Y0 ~ Y4 , Y7
	C16		X0 ~ X7	Y0 ~ Y7
	C24		X0 ~ XF	Y0 ~ Y7
	C40		X0 ~ XF , X10 ~ X17	Y0 ~ YF
	C56		X0 ~ XF , X10 ~ X1F	Y0 ~ YF , Y10 ~ Y17
	C72		X0 ~ XF , X10 ~ X1F X20 ~ X27	Y0 ~ YF , Y10 ~ Y1F
初级扩展单元	E8	输入类型	X30 ~ X37	/
		I / O类型	X30 ~ X33	Y30 ~ Y33
		输出类型	/	Y30 ~ Y37
	E16	输入类型	X30 ~ X3F	/
		I / O类型	X30 ~ X37	Y30 ~ Y37
		输出类型	/	Y30 ~ Y3F
	E24	I / O类型	X30 ~ X3F	Y30 ~ Y37
	E40	I / O类型	X30 ~ X3F , X40 ~ X47	Y30 ~ Y3F

续上表

品　　种	型　　号		输入端编号	输出端编号
次级扩展单元	E8	输入类型	X50 ~ X57	/
		I / O类型	K50 ~ X53	Y50 ~ Y53
		输出类型	/	Y50 ~ Y57
	E16	输入类型	X50 ~ X5F	/
		I / O类型	X50 ~ X57	Y50 ~ Y57
		输出类型	/	Y50 ~ Y5F
	E24	I / O类型	X50 ~ X5F	Y50 ~ Y57
	E40	I / O类型	X50 ~ X5F, X60 ~ K67	Y50 ~ Y5F
I / O链接单元			X70 ~ X7F(WX7) X80 ~ X8F(WX8)	Y70 ~ Y7F(WY7) Y80 ~ Y8F(WY8)
A / D转换单元		通道0	X90 ~ X9F(WX9)	/
		通道1	X100 ~ X10F(WX10)	/
		通道2	X110 ~ X11F(WX11)	/
		通道3	X120 ~ X12F(WX12)	/
D / A转换单元	单元号0	通道0	/	Y90 ~ Y9F(WY9)
		通道1	/	Y100 ~ Y10F(WY10)
	单元号1	通道0	/	Y110 ~ Y11F(WY11)
		通道1	/	Y120 ~ Y12F(WY12)

第三章　FP1 的指令系统

第一节　概述

一、继电器系统与 PLC 指令系统

可编程控制器来源于继电器系统和计算机系统，可以将其理解为计算机化的继电器系统。继电器在控制系统中主要起两种作用：

1）逻辑运算。运用继电器触点的串、并联接等完成逻辑与、或、非等功能，从而可完成较复杂的逻辑运算。

2）弱电控制强电。即通过有关的触点的通断，控制继电器的电磁线圈，从而来控制强电的断通。

对于简单控制功能的完成，采用继电器控制系统具有简单、可靠、方便等特点，因此，继电器控制系统得到了广泛应用。

注意：

PLC 内部的硬件资源多数是以继电器的概念出现的。注意，只是概念上的继电器，并非物理继电器。这里所指的继电器均为软继电器，是由 PLC 内部的存储单元构成的。

二、FP1指令系统分类

表 3-1　FP1 系列可编程控制器指令统计

分类名称		C14/C16	C24/C40	C56/C72
基本指令	顺序指令	19	19	19
	功能指令	7	7	8
	控制指令	15	18	18
	条件比较指令	0	36	36
高级	数据传输指令	11	11	11
	数据运算及比较指令	36	41	41

FP1 的指令按照**功能**可分为两大类

- **基本指令**
- **高级指令**

按照在手持编程器上的**输入方式**可为三种

- **键盘指令。**可以直接在键盘上输入的指令（即各种指令在手持编程器上有相应的按键）。
- **非键盘指令。**键盘上找不到，输入时需借助于“ SC”和“ HELP”键，指令方可输入。
- **扩展功能指令。**也是键盘上找不到的，但可通过输入其功能号将其输入，即用“ FN”键加上数字键输入该类指令。这类指令在指令表中都各自带有功能编号，在显示器上显示为“ FN ×××”，其中 N 是功能编号，×××是指令的助记符。输入功能编号后，助记符可自动显示，不必由用户输入。

第三章　FP1 的指令系统

第二节　FP1 的基本指令系统

基本指令可分为四大类，即

- **基本顺序指令：** 主要执行以位 (bit) 为单位的逻辑操作，是继电器控制电路的基础。
- **基本功能指令：** 有定时器、计数器和移位寄存器指令。
- **控制指令：** 可根据条件判断，来决定程序执行顺序和流程的指令。
- **比较指令：** 主要进行数据比较。

基本指令多数是构成继电器顺序控制电路的基础，所以借用继电器的线圈和触点来表示。同时，该类指令还是可编程控制器使用中最常见、也是用得最多的指令，因此，属于必须熟练掌握和运用的内容。

一、基本顺序指令

基本顺序指令主要是对继电器和继电器触点进行逻辑操作的指令。

FP1 的指令表达式比较简单，由操作码和操作数构成，格式为：

地址　操作码　　操作数

其中，操作码规定了 CPU 所执行的功能。

例如： AN X0 ，表示对 X0 进行与操作

操作数包含了操作数的地址、性质和内容。操作数可以没有，也可以是一个、两个、三个甚至四个，随不同的指令而不同。如 / 指令就没有操作数。

表 3-3 基本顺序指令的操作数

指令助记符	继电器			定时/计数器触点	
	X	Y	R	T	C
ST、ST/					
OT	×			×	×
AN、AN/					
OR、OR/					
SET、RST	×			×	×
PLS	×			×	×

表中对应项目为“×”表示该项不可用，为空则表示可用。

例如：OT 指令对应继电器 X 项为“×”，说明 OT 指令的操作数不能为 X 继电器。

1. 输入输出指令：ST、ST/、OT

ST　　加载　　用A类触点（常开触点）开始逻辑运算的指令。

ST/　　加载非　　用B类触点（常闭触点）开始逻辑运算的指令。

OT　　输出　　输出运算结果到指定的输出端，是继电器线圈的驱动

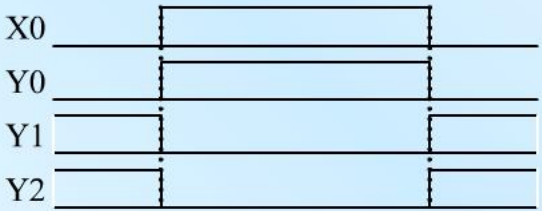
例 3-1

梯形图

指令表

地址	指令	数据
0	ST	X0
1	OT	Y0
2	/	
3	OT	Y1
4	ST/	X0
5	OT	Y2

时序图



例题说明：

当 X0 接通时， Y0 接通；当 X0 断开时，Y1 接通、 Y2 接通。

由例中可见， Y0 和 Y1 都受控于 X0 ，但是因为 Y1 前面有非指令，因此与 Y0 的状态正好相反，这与继电器系统明显不同，在继电器系统中， X0 断开， Y1 回路就不可能导通。

此外，对于输出 Y2 ，也是当输入触点 X0 断开时， Y2 接通，与 Y1 的控制方式

/ 指令为逻辑取反指令，可单独使用，

注意事项

但是一般都是与其它指令组合形成新指令使用，如 ST/ 。

OT 不能直接从左母线开始，但是必须以右母线结束。

OT 指令可以连续使用，构成并联输出，也属于分支的一种，可参见堆栈指令。

一般情况下，对于某个输出继电器只能用一次 OT 指令，否则，可编程控制器按

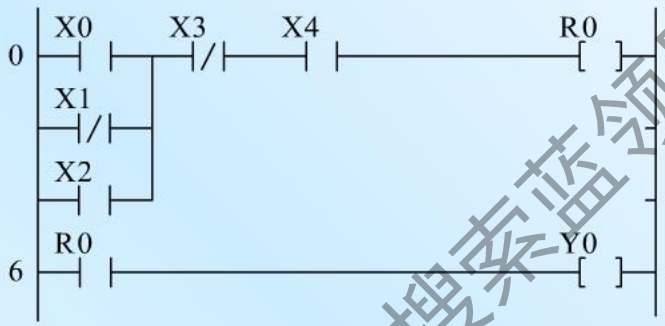
2. 逻辑操作指令：

AN、AN/、OR、OR/

AN	与	串联一个A类(常开)触点。
AN/	与非	串联一个B类(常闭)触点。
OR	或	并联一个A类(常开)触点。
OR/	或非	并联一个B类(常闭)触点。

例 3-2

梯形图



指令表

地址	指令	数据
٠	ST	X٠
١	OR/	X١
٢	OR	X٢
٣	AN/	X٣
٤	AN	X٤
٥	OT	R٠
٦	ST	R٠
٧	OT	Y٠

时序图

例题说明：

当 X0 、 X4 接通且 X3 断开时， R0 接通； R0 同时又是 Y0 的控制触点， R0 接通时 Y0 也接通。

由于 X0 、 X1 和 X2 三个触点并联， X2 与 X0 同为常开触点，所以 X2 和 X0 具有同样的性质；而 X1 为常闭触点，与 X0 的性质正好相反。 X2 和 X1 的时序图也与 X0 相同或相反，故这里略去。

注意事项

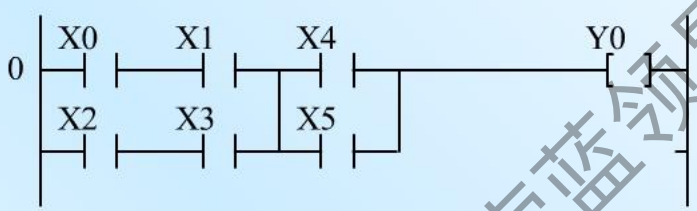
AN 、 AN/ 、 OR 、 OR/ 可连续使用。

3. 块逻辑操作指令：ANS、ORS

ANS	组与	执行多指令块的与操作，即实现多个逻辑块相串联。
ORS	组或	执行多指令块的或操作，即实现多个逻辑块相并联。

例 3-3

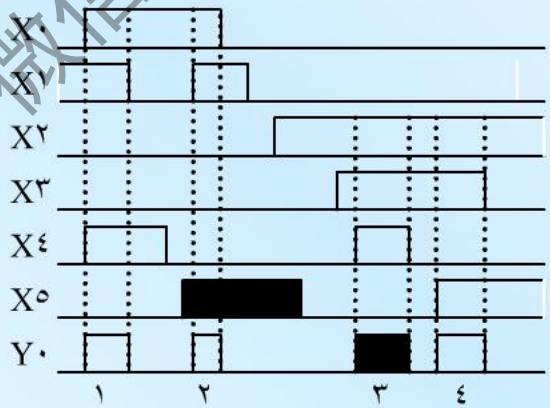
梯形图



指令表

地址	指令	数据
٠	ST	X٠
١	AN	X١
٢	ST	X٢
٣	AN	X٣
٤	ORS	
٥	ST	X٤
٦	OR	X٥
٧	ANS	
٨	OT	Y٠

时序图



例题说明：

从时序图上看，该例的逻辑关系显得比较复杂，但是仔细分析就可发现 Y0 有四个接通段，分别代表了该例子的四种有效组合。

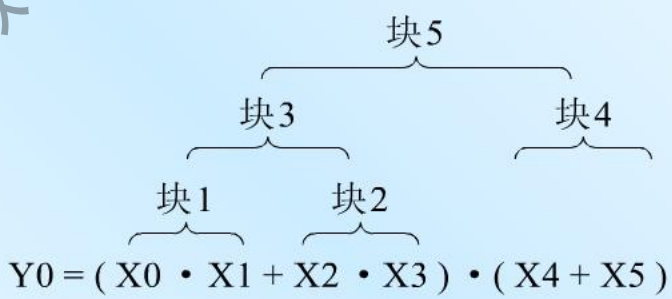
- 当 X0 、 X1 接通且 X4 接通时， Y0 接通，对应图中第 1 段接通情况。
- 当 X0 、 X1 接通且 X5 接通时， Y0 接通，对应图中第 2 段接通情况。
- 当 X2 、 X3 接通且 X4 接通时， Y0 接通，对应图中第 3 段接通情况。
- 当 X2 、 X3 接通且 X5 接通时， Y0 接通，对应图中第 4 段接通情况。

注意事项

掌握 ANS、 ORS 的关键主要有两点：一是要理解好串、并联关系，二是要形成块的观念。针对例 3-3 ，在下面的图中，分别从程序和逻辑关系表达式两方面对此加以具体说明。

从图中可见， X0 和 X1 串联后组成逻辑块 1， X2 和 X3 串联后组成逻辑块 2，用 ORS 将逻辑块 1 和逻辑块 2 并联起来，组合成为逻辑块 3；然后由 X4 和 X5 并联后组成逻辑块 4，再用 ANS 将逻辑块 3 和逻辑块 4 串联起来，组合成为逻辑块 5，结果输出给 Y0 。

地址	指令	数据	
0	ST	X0	块1
1	AN	X1	
2	ST	X2	块2
3	AN	X3	
4	ORS		块3
5	ST	X4	
6	OR	X5	块4
7	ANS		块5
8	OT	Y0	



4. 堆栈指令：PSHS、RDS、POPS

PSHS　推入堆栈　存储该指令处的操作结果。

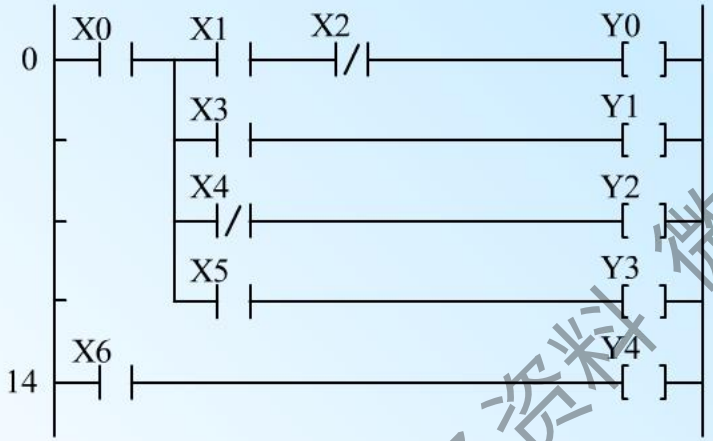
RDS　读取堆栈　读出 PSHS 指令存储的操作结果。

POPS　弹出堆栈　读出并清除由 PSHS 指令存储的操作结果。

堆栈指令主要用于构成具有分支结构的梯形图，使用时必须遵循规定的 PSHS、RDS、POPS 的先后顺序。

例 3-4

梯形图



指令表

地址	指令	数据
0	ST	X0
1	PSHS	
2	AN	X1
3	AN/	X2
4	OT	Y0
5	RDS	
6	AN	X3
7	OT	Y1
8	RDS	
9	AN/	X4
10	OT	Y2
11	POPS	
12	AN	X5
13	OT	Y3
14	ST	X6
15	OT	Y4

例题说明：

当 X0 接通时，程序依次完成下述操作。

- 存储 PSHS 指令处的运算结果（这里指 X0 的状态），这时 X0 接通，则当 X1 也接通且 X2 断开时， Y0 输出。
- 由 RDS 指令读出存储的结果，即 X0 接通，则当 X3 接通时， Y1 输出。
- 由 RDS 指令读出存储的结果，即 X0 接通，则当 X4 断开时， Y2 输出。
- 由 POPS 指令读出存储的结果，即 X0 接通，则当 X5 接通时， Y3 输出；然后将 PSHS 指令存储的结果清除，即解除与 X0 的关联，后续指令的执行将不再受 X0 影响。
- 当 X6 接通时， Y4 输出。此时与 X0 的状态不再相关。

本例中连用了两个 RDS 指令，目的是为了说明该指令只是读存储结果，而不影响存储结果；在执行了 POPS 后，就结束了堆栈指令，不再与 X0 的状态相关，如例中， Y4 的状态只受 X6 控制。

注意事项

- 当程序中遇到 PSHS 时，可理解为是将左母线到 PSHS 指令（即分支点）之间的所有指令存储起来，推入堆栈，提供给下面的支路使用。换个角度，也可理解为左母线向右平移到分支点，随后的指令从平移后的左母线处开始。
- RDS 用于 PSHS 之后，这样，当每次遇到 RDS 时，该指令相当于将 PSHS 保存的指令重新调出，随后的指令表面上是接着 RDS，实际上相当于接着堆栈中的指令来写。在功能上看，也就是相当于将堆栈中的那段梯形图与 RDS 后面的梯形图直接串联起来。
- POPS 相当于先执行 RDS 的功能，然后结束本次堆栈，因此，用在 PSHS 和 RDS 的后面，作为分支结构的最后一个分支回路。
- 从上面对构成堆栈的三个指令的分析可知，最简单的分支，即两个分支，可只由 PSHS 和 POPS 构成；而三个以上的分支，则通过反复调用 RDS 指令完成，这点可参见例题。也就是说，一组堆栈指令中，有且只有一个 PSHS 和一个 POPS，但是可以没有或有多个 RDS。
- 注意区分分支结构和并联输出结构梯形图。二者的本质区别在于：分支结构中，分支点与输出点之间串联有触点，而不单纯是输出线圈。
- 堆栈指令的复杂应用还包括嵌套使用。

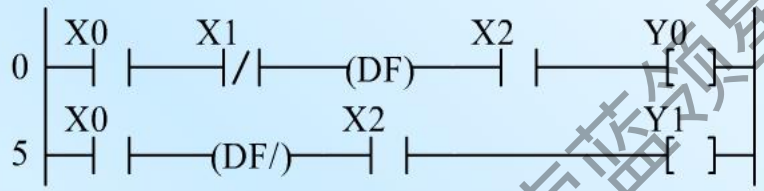
5. 微分指令：DF 、DF/

DF　　上升沿微分 检测到触发信号上升沿，使触点接通一个扫描周期。

DF/　　下降沿微分 检测到触发信号下降沿，使触点接通一个扫描周期。

例 3-5

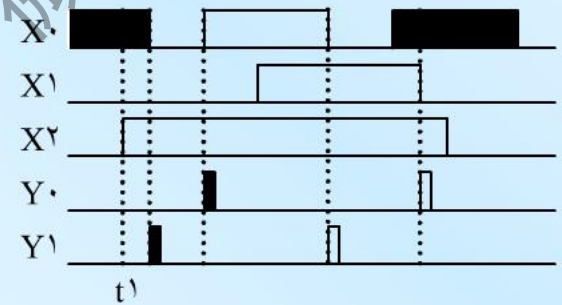
梯形图



指令表

地址	指令	数据
٠	ST	X٠
١	AN/	X١
٢	DF	
٣	AN	X٢
٤	OT	Y٠
٥	ST	X٠
٦	DF/	
٧	AN	X٢
٨	OT	Y١

时序图



例题说明：

当检测到触发信号的上升沿时，即 X1 断开、 X2 接通且 X0 由 OFF→ON 时， Y0 接通一个扫描周期。另一种情况是 X0 接通、 X2 接通且 X1 由 ON→OFF 时， Y0 也接通一个扫描周期，这是由于 X1 是常闭触点的缘故。

当检测到触发信号的下降沿时，即 X2 接通且 X0 由 ON→OFF 时， Y1 接通一个扫描周期。

注意事项

DF 和 DF/ 指令的作用都是在控制条件满足的瞬间，触发后面的被控对象（触点或操作指令），使其接通一个扫描周期。这两条指令的区别在于：前者是当控制条件接通瞬间（上升沿）起作用，而后者是在控制条件断开瞬间（下降沿）起作用。这两个微分指令在实际程序中很有用，可用于控制那些只需触发执行一次的动作。在程序中，对微分指令的使用次数无限制。

这里所谓的“触发信号”，指的是 DF 或 DF/ 前面指令的运算结果，而不是单纯的某个触点的状态，如例中 X0 与 X1 的组合；也不是后面的触点状态，如在时序图中的 t1 时刻，X0 和 X1 都处于有效状态，X2 的上升沿却不能使 Y0 接通。

6. 置位、复位指令：SET、RST

SET　置位　保持触点接通，为 ON 。

RST　复位　保持触点断开，为 OFF 。

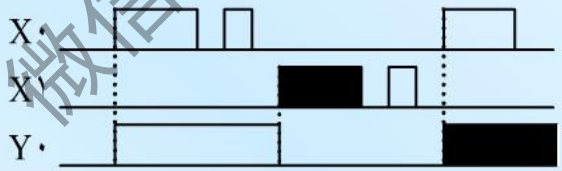
例 3-6

梯形图

指令表

地址	指令	数据
·	ST	X·
\	SET	Y·
٤	ST	X\
٥	RST	Y·

时序图



例题说明：

该程序执行的结果是，当 X0 接通时，使 Y0 接通，此后不管 X0 是何状态，Y0 一直保持接通。而当 X1 接通时，将 Y0 断开，此后不管 X1 是何状态，Y0 一直保持断开。

7. 保持指令：KP

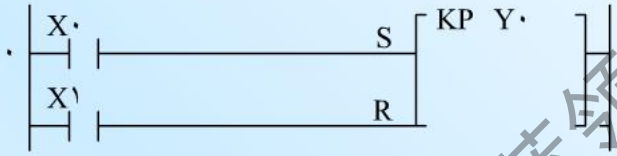
KP　保持　使输出为 ON，并保持。

KP 指令的作用是将输出线圈接通并保持。该指令有两个控制条件，一个是置位条件（S）、另一个是复位条件（R）。当满足置位条件，输出继电器（Y 或 R）接通，一旦接通后，无论置位条件如何变化，该继电器仍然保持接通状态，直至复位条件满足时断开。

S 端与 R 端相比，R 端的优先权高，即如果两个信号同时接通，复位信号优先有效。

例 3-7

梯形图



指令表

地址	指令	数据
·	ST	X·
1	ST	X1
2	KP	Y·

时序图

例题说明：

当 X0 接通时， Y0 接通；当 X1 接通时， Y0 断开，而不论 X0 状态如何。

注意事项

该指令与 SET 、 RST 有些类似，另外， SET 、 RST 允许输出重复使用，而 KP 指令则不允许。

8. 空操作指令：NOP

NOP　空操作　　空操作。

PLC 执行 NOP 指令时，无任何操作，但是要消耗一定的时间。

当没有输入程序或进行清理内存操作时，程序存储器各单元均自动为空操作指令。

可用 NOP 作为查找时的特殊标记，人为插入若干个 NOP 指令，对程序进行分段，便于检查和修改。如程序中某一点插入的 NOP 指令的数量超出 1 个，编程系统会自动对其进行编号，因此，该指令常在调试程序时使用，此时，程序的大小有所增加，但是对运算结果没有影响。

二、基本功能指令

基本功能指令主要包括一些具有定时器、计数器和移位寄存器三种功能的指令。其中，定时和计数本质上是同一功能。根据指令功能分类，将高级指令中的可逆计数指令F118(UDC)、左右移位指令F119(LRSR)以及辅助定时器指令F137(STMR)也包括在内。

表 3-5 基本功能指令的操作数

指令名称	可用寄存器										
	继电器			定时/计数器		寄存器	索引寄存器		常数		索引修正值
	WX	WY	WR	SV	EV	DT	IX	IY	K	H	
TM 预置值	×	×	×		×	×	×	×		×	×
CT 预置	×	×	×		×	×	×	×		×	×

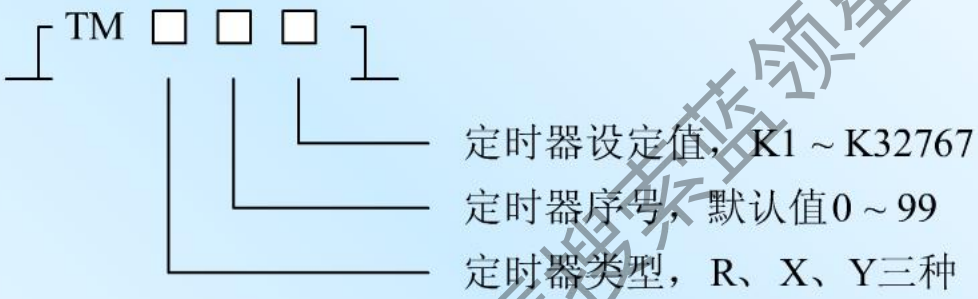
1．定时器指令：TM、F137(STMR)

TMR　以 0.01s 为最小时间单位，设置延时接通的定时器。

TMX　以 0.1s 为最小时间单位，设置延时接通的定时器。

TMY　以 1.0s 为最小时间单位，设置延时接通的定时器。

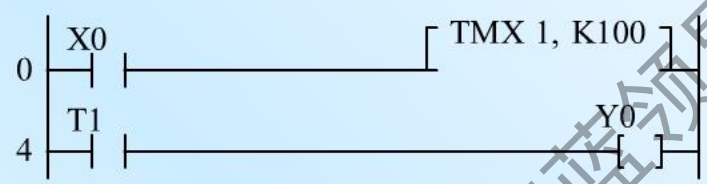
定时器的工作原理为：定时器为减 1 计数。当程序进入运行状态后，输入触点接通瞬间定时器开始工作，先将设定值寄存器 SV 的内容装入过程值寄存器 EV 中，然后开始计数。每来一个时钟脉冲，过程值减 1，直至 EV 中内容减为 0 时，该定时器各对应触点动作，即常开触点闭合、常闭触点断开。而当输入触点断开时，定时器复位，对应触点恢复原来状态，且 EV 清零，但 SV 不变。若在定时器未达到设定时间时断开其输入触点，则定时器停止计时，其过程值寄存器被清零，且定时器对应触点不动作，直至输入触点再接通，重新开始定时。



简单的说，当定时器的执行条件成立时，定时器以R、X、Y所规定的时间单位对预置值作减计数，预置值减为0时，定时器导通。其对应的常开触点闭合，常闭触点断开。

例 3-8

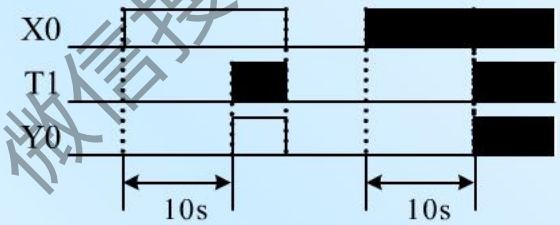
梯形图



指令表

地址	指令	数据
0	ST	X0
1	TMX	1
		K100
4	ST	T1
5	OT	Y0

时序图



例题说明：

当 X0 接通时，定时器开始定时， 10 秒后，定时时间到，定时器对应的常开触点 T1 接通，使输出继电器 Y0 导通为 ON ；当 X0 断开时，定时器复位，对应的常开触点 T1 断开，输出继电器 Y0 断开为 OFF 。

注意事项

1) TM 指令是减法计数型预置定时器，参数有两个，一个是时间单位，即定时时钟，可分为 3 种， R=0.01s ， X=0.1s， Y=1.0s ；另一个是预置值，只能用十进制，编程格式为 K 加上十进制数，因此，取值范围可表示为 K1 ~ K32767 。这样，定时时间就可以根据上述两个参数直接计算出来，即

定时时间 = 时间单位 × 预置值

也正是由于这个原因， TM R1 K1000 、 TM X1 K100 、 TM Y1 K10 这三条指令的延时时间是相同的，都是 10 秒，差别仅在于定时的时间精度不同。对于这个例子，由于只用到定时结果，采用上述任何一种写法都可以。

2) 定时器的设定值和过程值会自动存入相同编号的专用寄存器 SV 和 EV 中，因此可通过察看同一编号的 SV 和 EV 内容来监控该定时器的工作情况。采用不同的定时时钟会影响精度，也就是说，过程值 EV 的变化过程不同。

3) 同输出继电器的概念一样，定时器也包括线圈和触点两个部分，采用相同编号，但是线圈是用来设置，触点则是用于引用。因此，在同一个程序中，相同编号的定时器只能使用一次，即设置一次，而该定时器的触点可以通过常开或常闭触点的形式被多次引用。

4) 在 FP1-C24 中，初始定义有 100 个定时器，编号为 T0 ~ T99，通过系统寄存器 No.5 可重新设置定时器的个数。

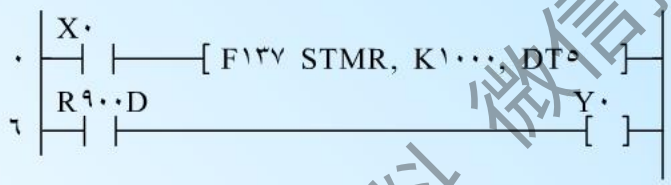
5) 由于定时器在定时过程中需持续接通，所以在程序中定时器的控制信号后面不能串联微分指令。

6) 在实际的 PLC 程序中，定时器的使用是非常灵活的，如将若干个定时器串联或是将定时器和计数器级联使用可扩大定时范围，或将两个定时器互锁使用可构成方波发生器，还可以在程序中利用高级指令 F0(MV) 直接在 SV 寄存器中写入预置值，从而实现可变定时时间控制。

F137(STMR)　　以 0.01s 为最小时间单位设置延时接通的定时器。该定时器与 TMR 类似，但是设置方式上有所区别。下面举例说明。

例 3-9

梯形图



指令表

地址	指令	数据
٠	ST	X٠
١	F١٣٧(STMR)	
		K١٠٠٠
		DT٥
٦	ST	R٩٠٠D
٧	OT	Y٠

例题说明：

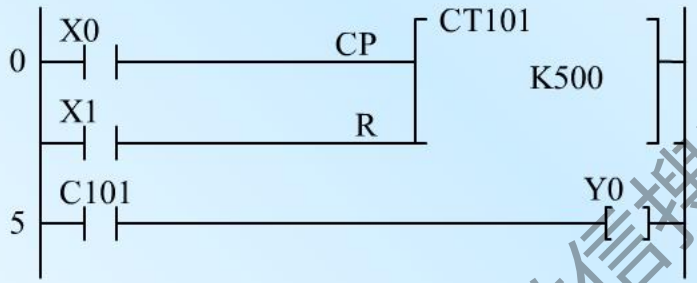
该例与上例中使用 TMX 实现的定时结果类似，但是当用 R900D 作为定时器的触点编程时，务必将 R900D 编写在紧随 F137(STMR) 指令之后。此外，这里的 DT5 起到与经过值寄存器 EV 类似的作用。

2．计数器指令：CT、F118(UDC)

CT 指令是一个减计数型的预置计数器。其工作原理为：程序一进入“运行”方式，计数器就自动进入初始状态，此时 SV 的值被自动装入 EV，当计数器的计数输入端 CP 检测到一个脉冲上升沿时，预置值被减 1，当预置值被减为 0 时，计数器接通，其相应的常开触点闭合，常闭触点断开。计数器的另一输入端为复位输入端 R，当 R 端接收到一个脉冲上升沿时计数器复位，计数器不接通，其常开触点断开，常闭触点闭合；当 R 端接收到脉冲下降沿时，将预置值数据再次从 SV 传送到 EV 中，计数器开始工作。计数器 CT 指令的梯形图符号如下图所示。

例 3-10

梯形图



指令表

地址	指令	数据
0	ST	X0
1	ST	X1
2	CT	101
		K500
5	ST	C101
6	OT	Y0

例题说明：

程序开始运行时，计数器自动进入计数状态。当检测到 X0 的上升沿 500 次时，计数器对应的常开触点 C101 接通，使输出继电器 Y0 导通为 ON；当 X1 接通时，计数器复位清零，对应的常开触点 C101 断开，输出继电器 Y0 断开为 OFF 。

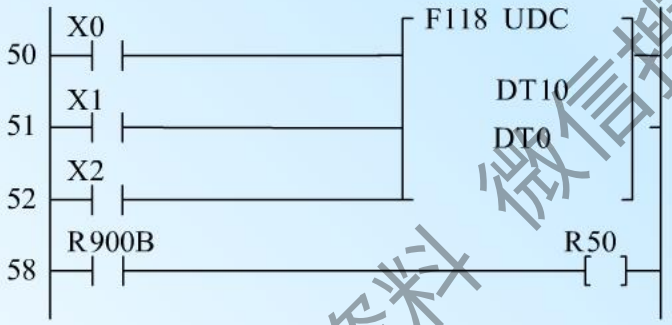
注意事项

- FP1-C24 中，共有 44 个计数器，编号为 C100 ~ C143。此编号可用系统寄存器 No.5 重新设置。设置时注意 TM 和 CT 的编号要前后错开。
- 计数器与定时器有密切的关系，编号也是连续的。定时器本质上就是计数器，只不过是对固定间隔的时钟脉冲进行计数，因此两者有许多性质是类似的。
- 与定时器一样，每个计数器都有对应相同编号的 16 位专用寄存器 SV 和 EV，以存储预置值和过程值。
- 同一程序中相同编号的计数器只能使用一次，而对应的常开和常闭触点可使用无数次。
- 计数器有两个输入端，即计数脉冲输入端 CP 和复位端 R，分别由两个输入触点控制，R 端比 CP 端优先权高。
- 计数器的预置值即为计数器的初始值，该值为 0 ~ 32767 中的任意十进制数，书写时前面一定要加字母“K”。

F118(UDC) 指令，也起到计数器的作用。与 CT 不同的是，该指令可以根据参数设置，分别实现加 / 减计数的功能，下面举例说明。

例 3-11

梯形图



指令表

地址	指令	数据
50	ST	X0
51	ST	X1
52	ST	X2
53	F118(UDC)	
		DT10
		DT0
58	ST	R900B
59	OT	R50

例题说明： 使用 F118(UDC) 指令编程时，一定要有加 / 减控制、计数输入和复位触发三个信号。

当检测到复位触发信号 X2 的下降沿时， DT10 中的数据被传送到 DT0 中，计数器开始工作；当检测到 X2 的上升沿时，即复位信号有效，DT0 被清 0 ，计数器停止工作。

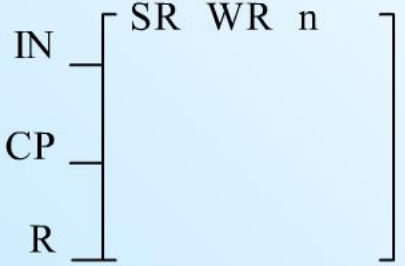
X0 为加 / 减控制信号，当其为 ON 时，进行加计数，为 OFF 时，进行减计数。

X1 为计数输入信号，检测到其上升沿时，根据 X0 的状态，执行加 1 或减 1 计数。

这里， DT10 相当于 CT 指令中的预置值寄存器 SV ， DT0 相当于经过值寄存器 EV 。当 DT0 中的结果为 0 时，特殊内部寄存器 R900B 接通，内部寄存器 R50 有输出。

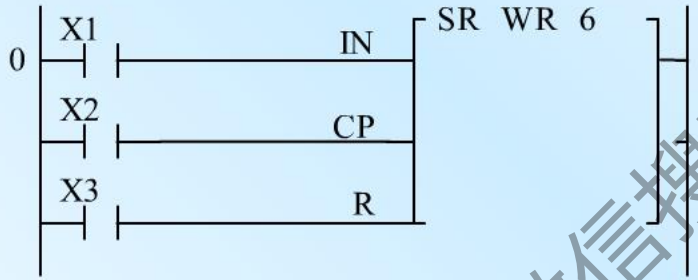
3．移位指令：SR、F119(LRSR)

SR 为左移移位指令。其功能为：当 R 端为 OFF 状态时，该指令有效。这时，每检测到一个 CP 端的上升沿 (OFF→ON)， WRn 中的数据就从低位向高位依次左移一位，其中， WRn 的最低位用数据输入端 IN 的状态补入，最高位数据丢失。当 R 为 ON 状态时，该指令复位，WRn 中的数据被清零。此外，需要指出的是，该指令的操作数只能用内部字继电器 WR， n 为 WR 继电器的编号。



例 3-12

梯形图



指令表

地址	指令	数据
0	ST	X1
1	ST	X2
2	ST	X3
3	SR	WR6

例题说明：

当复位信号 X3 为 OFF 状态时，每当检测到移位信号 X2 的上升沿， WR6 寄存器的数据左移 1 位，最高位丢失，最低位由当时数据输入信号 X1 的状态决定：如果当时 X1 处于接通状态，则补 1 ，否则，补 0 。

如果 X3 接通， WR6 的内容清 0 ，这时 X2 信号无效，移位指令停止工作。

F119（LRSR）指令为左/右移位寄存器指令，使 16-bit 内部继电器中的数据向左或向右移动 1-bit。F119（LRSR）指令可以使用作为数据区的寄存器和常数见下表。

D1：移位区内首地址寄存器；

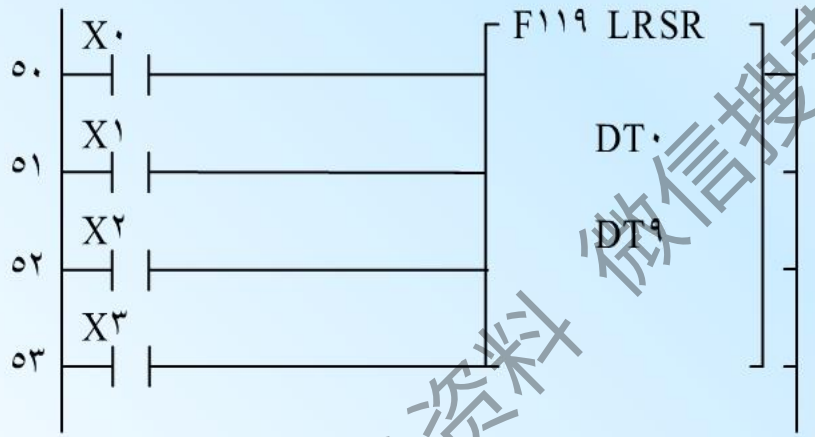
D2：移位区内末地址寄存器；

注意：移位区内的首地址和末地址要求是同一种类型的寄存器，并满足 D1≤D2。

操作数	可用寄存器										
	继电器			定时/计数器		寄存器	索引寄存器		常数		索引修正值
	WX	WY	WR	SV	EV	DT	IX	IY	K	H	
D1	×						×	×	×	×	×
D2	×						×	×	×	×	×

例 3-13

梯形图



指令表

地址	指令	数据
٥٠	ST	X٠
٥١	ST	X١
٥٢	ST	X٢
٥٣	ST	X٣
٥٤	F١١٩(LRSR)	
		DT٠
		DT٩

例题说明：

F119(LRSR) 指令需要有 4 个输入信号，即左 / 右移位信号、数据输入、移位信号和复位触发信号，分别对应例中 X0 ~ X3 共 4 个触点。 DT0 指定移位区首地址， DT9 指定末地址。

当 X3 为 ON 时，复位信号有效， DT0 和 DT9 均被清 0 ，移位寄存器停止工作。

当 X3 为 OFF 时，移位寄存器正常工作。这时，由移位触发信号 X2 的上升沿触发移位操作，移动的方向由 X0 决定，若 X0 为 ON ，表示进行数据左移，为 OFF ，表示进行数据右移。至于移入的数据为 1 还是为 0 ，则取决于 X1 的状态，若 X1 接通，移入数据为 1 ，否则，移入数据为 0 。

这里， DT0 ~ DT9 构成了连续的 16 位寄存器区，移位操作使所有位同时进行，整个区域按照高位在左侧低位在右侧的顺序排列。

三、控制指令

从程序的执行步骤和结构构成上看，基本顺序指令和基本功能指令是按照其地址顺序执行的，直到程序结束为止；而控制指令则可以改变程序的执行顺序和流程，产生跳转和循环，构成复杂的程序及逻辑结构。

PLC 指令的执行特点是采用扫描执行方式，这里就存在扫描和执行的关系的问题：对于一段代码，**扫描并执行**是正常的步骤，但是也存在另外一种情况，就

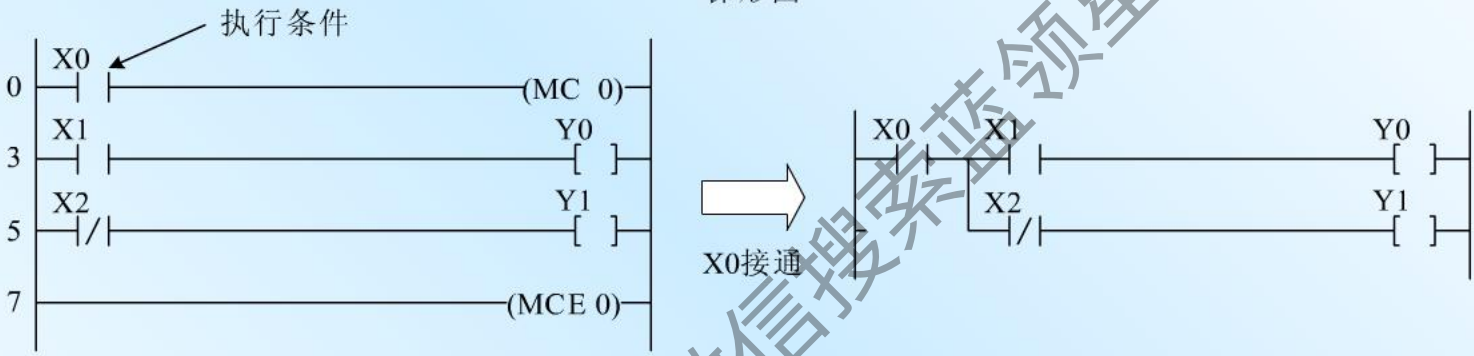
1．主控继电器指令：MC、MCE

MC：主控继电器指令。
MCE：主控继电器结束指令。

功能：用于在程序中将某一段程序单独界定出来。当 MC 前面的控制触点闭合时，执行 MC 至 MCE 间的指令；当该触点断开时，不执行 MC 至 MCE 间的指令。

例 3-14

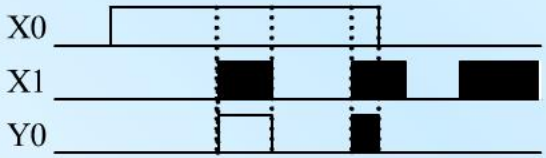
梯形图



指令表

地址	指令	数据
0	ST	X0
1	MC	0
3	ST	X1
4	OT	Y0
5	ST/	X2
6	OT	Y1
7	MCE	0

时序图



例题说明：

当控制触点 X0 接通时，执行 MC0 到 MCE0 之间的程序，这时，从上图中的梯形图可以看出，效果等同于右侧的简化梯形图。否则，不执行 MC0 到 MCE0 之间的程序。

值得注意的是，当主控继电器控制触点断开时，在 MC 至 MCE 之间的程序，遵循扫描但不执行的规则，可编程控制器仍然扫描这段程序，不能简单地认为可编程控制器跳过了这段程序。而且，在该程序段中不同的指令状态变化情况也有所不同，具体情况参见下表。

指令或寄存器	状态变化
OT(Y 、 R 等)	全部 OFF 状态
KP 、 SET 、 RST	保持控制触点断开前对应各继电器的状态
TM 、 F137(STMR)	复位，即停止工作
CT 、 F118(UDC)	保持控制触点断开前经过值，但停止工作

注意事项

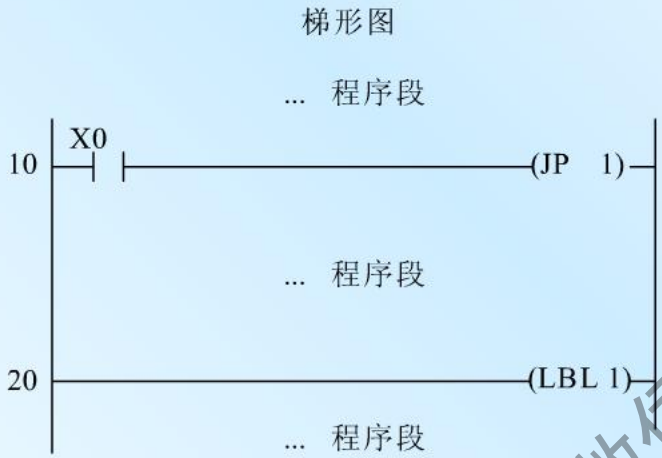
- MC 和 MCE 在程序中应成对出现，每对编号相同，编号范围为 0 ~ 31 之间的整数。而且，同一编号在一个程序中只能出现一次。
- MC 和 MCE 的顺序不能颠倒。
- MC 指令不能直接从母线开始，即必须有控制触点。
- 在一对主控继电器指令 (MC 、 MCE) 之间可以嵌套另一对主控继电器指令。

2．跳转指令：JP、LBL

JP：跳转指令。
LBL：跳转标记指令。

当控制触点闭合时，跳转到和 JP 指令编号相同的 LBL 处，不执行 JP 和 LBL 之间的程序，转而执行 LBL 指令之后的程序。与主控指令不同，遵循不扫描不执行的原则，在执行跳转指令时，JP 和 LBL 之间的指令略过，所以可使整个程序的扫描周期变短。

例 3-15



指令表

	地址	指令	数据
		…	
X0接通：该段不执行	10	ST	X0
	11	JP	1
		…	
X0断开：该段正常执行		…	
	20	LBL	1
		…	

例题说明：

在 JP1 指令的前面、 JP1 与 LBL1 中间、以及 LBL1 的后面都可能有其它的指令程序段，如图所示。当控制触点 X0 断开时，跳转指令不起作用， JP1 与 LBL1 中间的指令正常执行，与没有跳转指令一样；当控制触点 X0 接通时，执行跳转指令，跳过 JP1 与 LBL1 中间的程序段，直接执行 LBL1 的后面的程序段。

注意事项

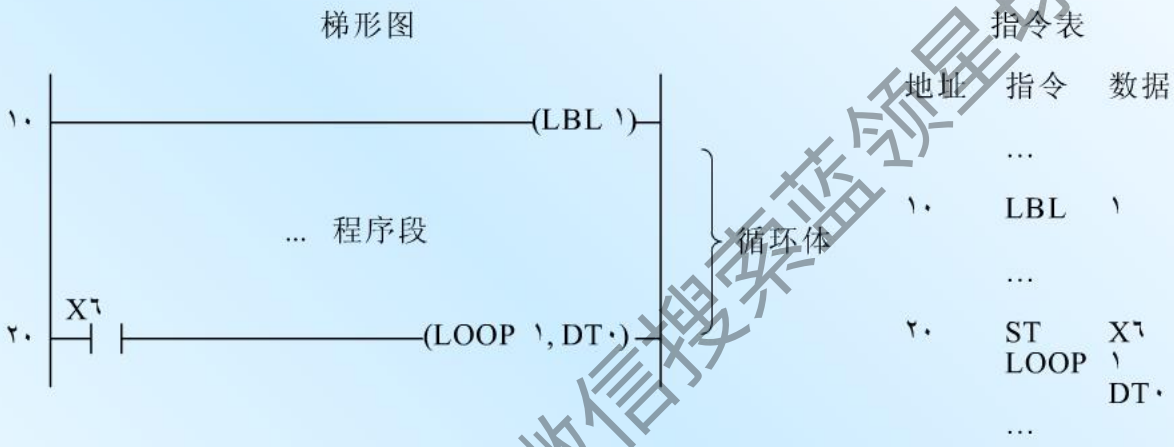
- 可以使用多个编号相同的 JP 指令，即允许设置多个跳向一处的跳转点，编号可以是 0 ~ 63 以内的任意整数，但不能出现相同编号的 LBL 指令，否则程序将无法确定将要跳转的位置。
- LBL 指令应该放在同序号的 JP 指令的后面，当然，放在前面也可以，不过这时扫描不会终止，而且可能发生瓶颈错误，详细内容请参见手册。
- JP 指令不能直接从母线开始，即前面必须有触发信号。
- 在一对跳转指令之间可以嵌套另一对跳转指令。
- 不能从结束指令 ED 以前的程序跳转到 ED 以后的程序中去；不能在子程序或中断程序与主程序之间跳转；不能在步进区和非步进区进行跳转。

3．循环跳转指令：LOOP、LBL

LOOP：循环指令。
LBL：循环标记指令。

循环指令的功能为：当执行条件成立时，循环次数减 1，如果结果不为 0，跳转到与 LOOP 相同编号的 LBL 处，执行 LBL 指令后的程序。重复上述过程，直至结果为 0，停止循环；当执行条件不成立时，不循环执行。

例 3-16



例题说明：

当 X6 接通时，数据寄存器 DT0 的预置值减 1，若结果不为 0， LOOP 指令跳转到 LBL1 处，执行 LBL1 之后的程序。重复执行相同的操作直至 DT0 中的内容变为 0，结束循环。

当 X6 断开时，不执行循环。

注意事项

- 可以使用多个编号相同的 LOOP 指令，编号可以是 0 ~ 63 以内的任意整数，但不能出现相同编号的 LBL 指令，否则程序将无法确定循环区间。此外，该指令可以与 JP 指令共用相同编号的 LBL 指令，但为了程序清晰，尽量避免。
- LBL 指令与同编号的 LOOP 指令的前后顺序不限，但工作过程不同。一般将 LBL 指令放于 LOOP 指令的上面，此时，执行循环指令的整个过程都是在一个扫描周期内完成的，所以整个循环过程不可太长，否则扫描周期变长，影响了 PLC 的响应速度，有时甚至会出错。
- LOOP 指令不能直接从母线开始，即必须有触发信号。当某编号的 LOOP 对应的触发信号接通时，与同编号的 LBL 即构成一个循环。
- 循环跳转指令可以嵌套使用。
- 不能从结束指令 ED 以前的程序跳转到 ED 以后的程序中去；也不能在子程序或中断程序与主程序之间跳转；不能在步进区和非步进区进行跳转。

4．结束指令：ED、CNDE

ED：结束指令，表示主程序结束。

CNDE：条件结束指令，当控制触点闭合时，可编程控制器不再继续执行程序，结束当前扫描周期，返回起始地址；否则，继续执行该指令后面的程序段。

例 3-17

例题说明：

当控制触点 X0 闭合时，条件结束指令 CNDE 起作用，返回程序起始地址，当前的扫描结束，进入下一次扫描；否则，控制触点 X0 断开，继续执行下面的指令扫描，当遇到 ED 指令，才结束当前的扫描。

5. 步进指令：SSTP、NSTP、NSTL、CSTP、STPE

SSTP：步进开始指令，表明开始执行该段步进程序。

NSTP、NSTL：转入指定步进过程指令。这两个指令的功能一样，都是当触发信号来时，程序转入下一段步进程序段，并将前面程序所用过的数据区清除，输出 OT 关断、定时器 TM 复位。区别在于触发方式不同，前者为脉冲式，仅当控制触点闭合瞬间动作，即检测控制触点的上升沿，类似于微分指令；后者为扫描式，每次扫描检测到控制触点闭合都要动作。

CSTP：复位指定的步进过程。

STPE：步进结束指令，结束整个步进过程。

除了用于生产过程的顺序控制，步进指令还可用于选择分支控制、并行分支控制等，

例 3-18

梯形图

指令表

地址	指令	数据
	...	
١٠	ST	X٠
١١	NSTP	١
١٤	SSTP	١
١٧	OT	Y١٠
١٨	ST	X١
١٩	NSTL	٢
٢٢	SSTP	٢
	...	
١٠٠	ST	X٣
١٠١	CSTP	٥٠
١٠٤	STPE	

例题说明：

当检测到 X0 的上升沿时，执行步进过程 1(SSTP1~SSTP2)；当 X1 接通时，清除步进过程 1，并执行步进过程 2；当 X3 接通时，清除步进过程 50，步进程序执行完毕。

注意事项

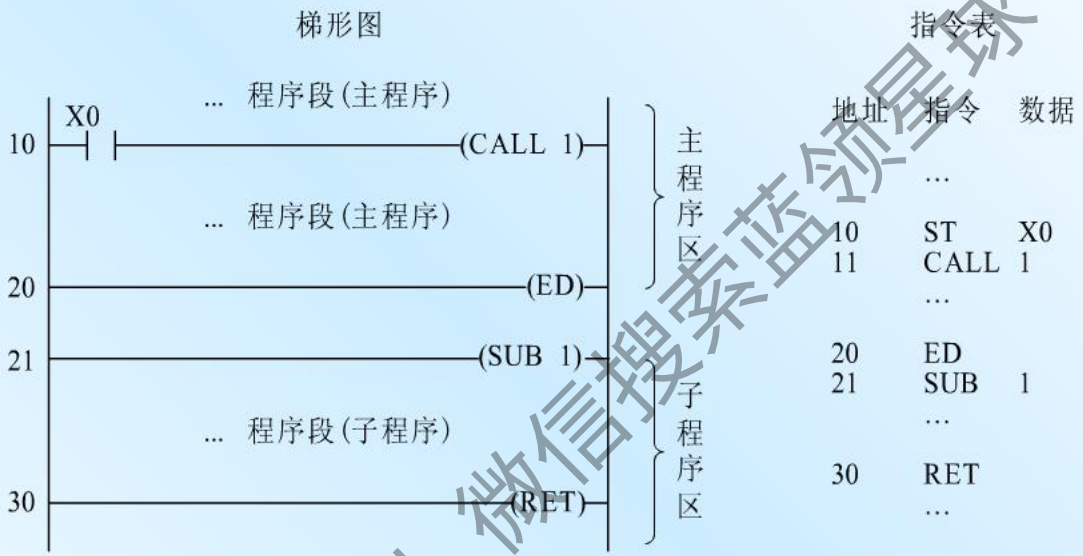
- 步进程序中允许输出 OT 直接同左母线相连。
- 步进程序中不能使用 MC 和 MCE、JP 和 LBL、LOOP 和 LBL、ED 和 CNDE 指令。
- 在步进程序区中，识别一个过程是从一个 SSTP 指令开始到下一个 SSTP 指令，或一个 SSTP 指令到 STPE 指令，即步进程序区全部结束。
- 当 NSTP 或 NSTL 前面的控制触点接通时，程序进入下一段步进程序。这里的控制触点和步进控制程序区结束指令 STPE 都是必需的。

6．子程序调用指令：CALL、SUB、RET

CALL：子程序调用指令，执行指定的子程序。
SUB：子程序开始标志指令，用于定义子程序。
RET：子程序结束指令，执行完毕返回到主程序。

子程序调用指令的功能：当 CALL n 指令的执行条件成立时，程序转至子程序起始指令 SUB n 处，执行 SUB n 到 RET 之间的第 n 号子程序。遇到 RET 指令，子程序结束并返回到 CALL n 的下一条指令处，继续执行主程序。

例 3-19



例题说明：

当 X0 接通时，程序从主程序转到编号为 1 的子程序的起始地址 SUB 1 处，开始执行子程序；当执行到 RET 处时，子程序执行完毕，返回到主程序调用处，从 CALL 1 指令的下一条指令继续执行随后的主程序。

当 X0 断开时，不调用子程序，继续执行主程序。

注意事项

- ✓ FP1-C24 可用子程序的个数为 16 个，即子程序编号范围为 SUB0 ~ SUB15，且两个子程序的编号不能相同。
- ✓ 子程序必须编写在主程序的 ED 指令后面，由子程序入口标志 SUB 开始，最后是 RET 指令，缺一不可。
- ✓ 子程序调用指令 CALL 可以在主程序、子程序或中断程序中使用，可见，子程序可以嵌套调用，但最多不超过 5 层。
- ✓ 当控制触点为 OFF 时，子程序不执行。这时，子程序内的指令状态如下表所示。

指令或寄存器	状态变化
OT 、 KP 、 SET 、 RST	保持控制触点断开前对应各继电器的状态
TM 、 F137(STMR)	不执行
CT 、 F118(UDC) ； SR 、 F119(LRSR)	保持控制触点断开前经过值，但停止工作
其它指令	不执行

7．中断指令：INT、ICTL、IRET

ICTL：中断控制指令，用于设定中断的类型及参数。
INT：　中断程序开始标志。
IRET：中断程序结束标志。

为了提高 PLC 的实时控制能力，提高 PLC 与外部设备配合运行的工作效率以及 PLC 处理突发事件的能力，FP1 设置了中断功能。中断就是中止当前正在运行的程序，去执行为要求立即响应信号而编制的中断服务程序，执行完毕再返回原先被中止的程序并继续运行。

FP1 的中断类型

FP1-C24 以上机型均有中断功能，其中断功能有两种类型，一种是外部中断，又叫**硬件中断**，一种是定时中断，又叫**软件中断**。

1) 外部中断共有 8 个中断源 X0 ~ X7，对应中断入口为

X0 – INT0　X4 – INT4
X1 – INT1　X5 – INT5
X2 – INT2　X6 – INT6
X3 – INT3　X7 – INT7

其优先级别为 INT0 最高， INT7 最低。 FP1 规定中断信号的持续时间应≥ 2ms 。

2)　内部定时中断是通过软件编程来设定每间隔一定的时间去响应一次中断服务程序，定时中断的中断入口为 INT24 。

中断的实现

1) 对于内部定时中断，是通过编程来实现的，定时中断的时间，由中断命令控制字设定。

٢) 对于外部中断，应先设定系统寄存器 No.403 的值，然后再设定中断控制字，并按中断程序的书写格式编写程序。

此外，与普通微机不同，PLC 的中断是非嵌套的，也就是说，在执行低级中断时，若有高级中断到来，并不立即响应高级中断，而是在执行完当前中断后，才响应高级中断。

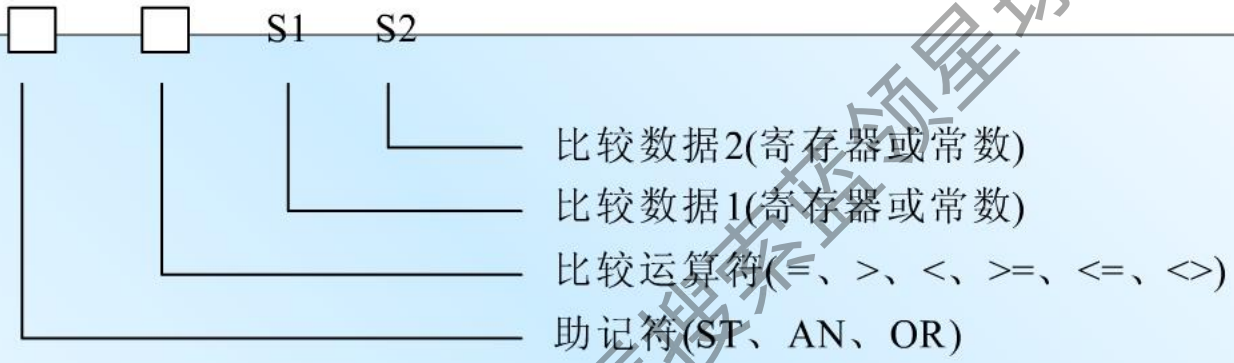
中断控制字的设置

ICTL 是中断控制字指令，有二个操作数 S1 和 S2 。它可以是常数 H，也可以是某个寄存器的数据。其中 S1 设置中断类型，S2 设置中断参数。具体设置方法参见手册。

注意事项

- 使用外部中断之前，首先设置系统寄存器 No.403 。
- ICTL 指令应和 DF 指令配合使用。
- 中断子程序应放在主程序结束指令 ED 之后。
- INT 和 IRET 指令必须成对使用。
- 中断子程序中不能使用定时器指令 TM 。
- 中断子程序的执行时间不受扫描周期的限制。
- 中断子程序中可以使用子程序调用指令。

四、比较指令



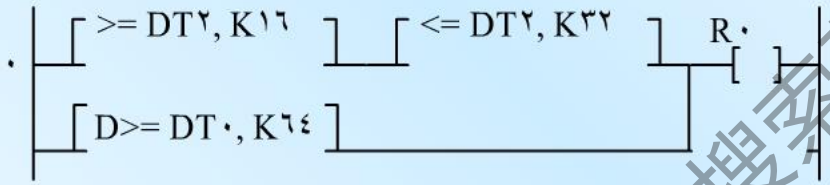
比较指令由 3 部分组成

第一部分为助记符， 分别由 ST 、 AN 、 OR 开始，用于指定条件满足后要进行的操作是开始，还是逻辑与、逻辑或；

第二部分为比较运算符， 主要有等于（=）、大于（>）、小于（<）、大于等于（>=）、小于等于（<=）和不等于（<>）共 6

例 3-20

梯形图



指令表

地址	指令	数据
٠	ST>=	
		DT٢
		K١٦
٥	AN<=	
		DT٢
		K٣٢
١٠	ORD>=	
		DT٠
		K٦٤
١٩	OT	R٠

例题说明：

该程序的功能为： 根据 DT2 中的数据范围，或 (DT1， DT0) 中的内容，来决定 R0 的输出状态。设 DT2 中数据用 x 表示，(DT1， DT0) 中数据用 y 表示，则当 16≤x≤32，或者 y≥64 时， R0 导通，输出为 ON；否则， R0 断开，输出为 OFF 。

从该例可以看出，比较指令实际上相当于一个条件触点，根据条件是否满足，决定触点的通断。

注意事项

单字比较为 16 位数据，双字比较为 32 位数据，用寄存器寻址时，后者采用两个相邻寄存器联合取值，如例中（DT1，DT0），表示由 DT1 和 DT0 联合构成 32 位数据。

在构成梯形图时，ST、AN、OR 与基本顺序指令中用法类似，区别仅在于操作数上，前者为寄存器（16-bit 或 32-bit），后者为继电器（1-bit）。

单字指令步数为 5 步，而双字指令步数为 9 步。

第三章　FP1 的指令系统

第三节 高级指令概述

一、高级指令的类型

数据传送指令： 16 位、32 位数据，以及位数据的传送、拷贝、交换等功能。

算术运算指令： 二进制数和 BCD 码的加、减、乘、除等算术运算。

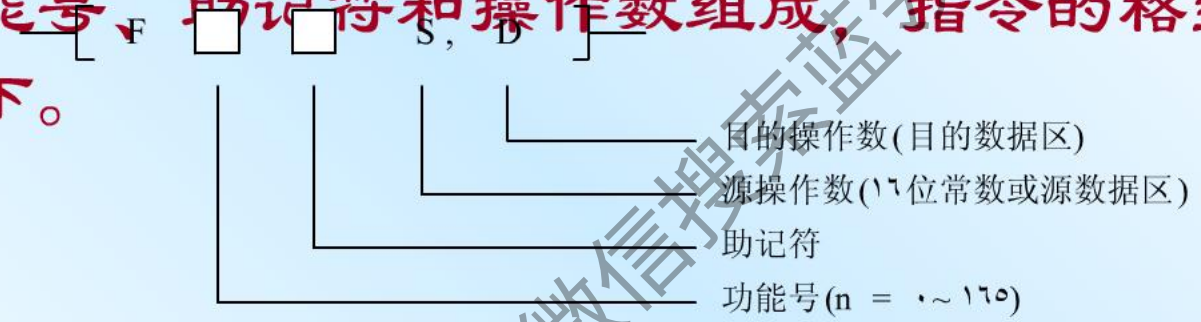
数据比较指令： 16 位或 32 位数据的比较。

逻辑运算指令： 16 位数据的与、或、异或和异或非运算。

数据转换指令： 16 位或 32 位数据按指定的格式进行转换。

二、高级指令的构成

高级指令由大写字母“F”、指令功能号、助记符和操作数组成，指令的格式如下。



Fn 是指令功能号， Fn = F0 ~ F165 。不同的功能号规定 CPU 进行不同的操作。指令的助记符用英文缩写表示，一般可据此大致推测出该指令的功能。

S 是源操作数或源数据区， D 是目的操作数或目的数据区，分别指定操作数或其地址、性质和内容。

操作数可以是一个、二个或者三个，取决于所用的指令，可以是单字 (16-bit) 和双字 (32-bit) 的数据，若为位操作指令，还可以是位 (1-bit) 数据。

三、高级指令的操作数

1．进位制

- 二进制系统 (BIN)
- 十进制常数 (K 常数)
- 十六进制常数 (H 常数)
- 二进制表示的十进制数 (BCD 码)

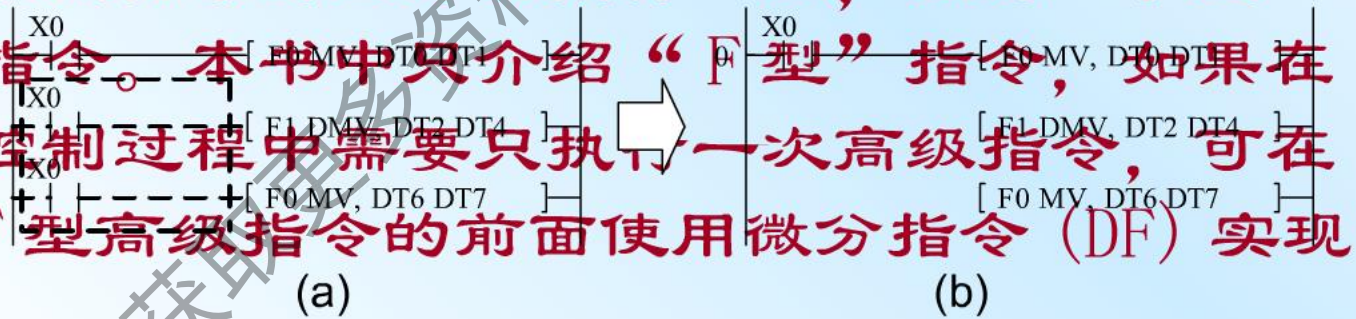
2、寄存器和常数

字继电器 (WX 、 WY 、 WR) 、定时器 / 计数器区 (T 、 C 、 SV 、 EV) 、数据寄存器 (DT) 、索引寄存器 (IX 、 IY) 和常数 (K 、 H) 均由 1 个字 (16-bit) 构成，且以字为单位进行处理。字继电器的内容按位对应其继电器元件的状态。

四、使用高级指令应注意的问题

在高级指令的前面必须加控制触点（触发信号），而在后面只能是右母线。

根据执行的过程，FP1 的指令有两种类型，即 F 型和 P 型。如果控制触点接通后，其后续的指令每个扫描周期都要执行一次，称为“F 型”指令；否则，如果后续的指令只在触发信号的上升沿执行一次，称为“P 型”指令。本书中只介绍“F 型”指令，如果在控制过程中需要只执行一次高级指令，可在 F 型高级指令的前面使用微分指令（DF）实现



第三章　FP1 的指令系统

第四节　FP1 的高级指令

一、数据传送指令

数据传送指令的功能是将源操作数中的数据，按照规定的要求，复制到目的操作数中去。可分为数据传送、位传送、数字传送、块传送及复制、寄存器交换等。

1、数据传送：F0(MV)、F1(DMV)、F2(MV/)、F3(DMV/)

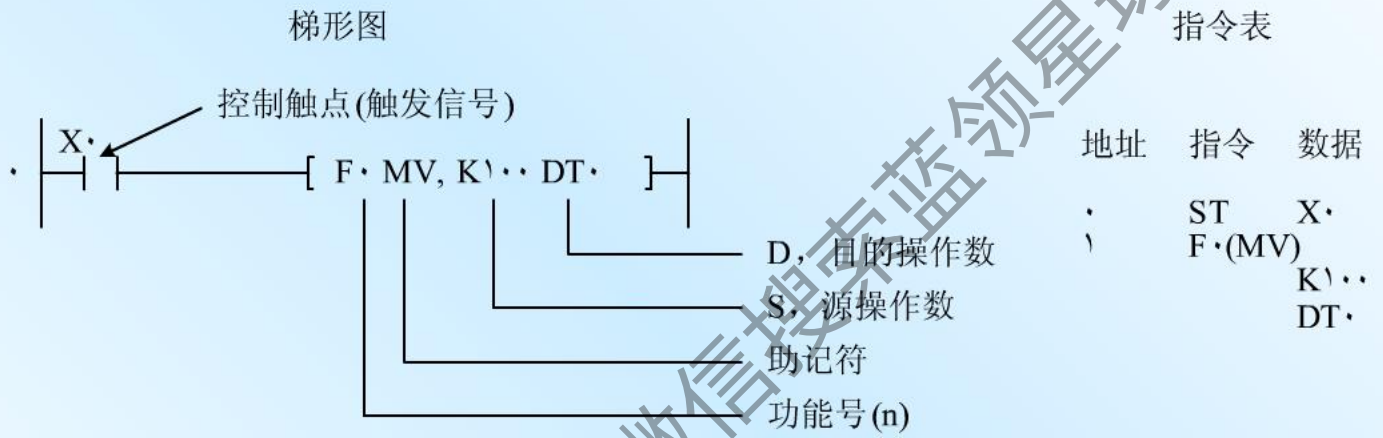
[F0 MV S, D]： 将一个 16 位的常数或寄存器中的数据传送到另一个寄存器中去。

[F1 DMV S, D]： 将一个 32 位的常数或寄存器区中的数据传送到另一个寄存器区中去。

[F2 MV/ S, D]： 将一个 16 位的常数或寄存器中的数据取反后传送到另一个寄存器中去。

[F3 DMV/ S, D]： 将一个 32 位的常数或寄存器区中的数据取反后传送到另一个寄存器区中去。

例 3-21

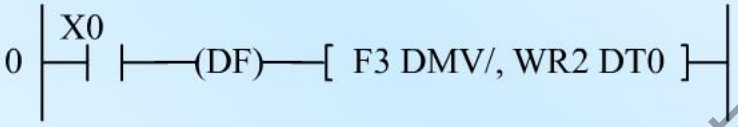


该程序的功能是： 当控制触点 X0 闭合时，每个扫描周期都要重复将十进制数 100 传送到内部字寄存器 DT0 中。

F0(MV) 指令对源操作数没有要求，而目的操作数不能是输入继电器 WX 和常数 K、H，原因很明显：目的操作数是用来保存结果的，自然不能用输入继电器和常数。后面介绍的其它指令也有类似情况。

例 3-22

梯形图



指令表

地址	指令	数据
0	ST	X0
1	F0(DMV/)	
		WR2
		DT0

与上例相比，该例有 5 点不同，下面加以详细说明。

1) 在控制触点后，增加了微分指令 DF，表示该指令仅在检测到控制触点 X0 闭合时执行一次；

2) F3(DMV/) 指令助记符的第一个字符为“ D”，表示该指令为双字操作，目的操作数为 DT0 寄存器，表示数据保存在寄存器 DT1 、 DT0 构成的 32 位单元中。在以后的双字操作指令中也遵循这一原则，即由相邻 2 个 16 位寄存器联合构成一个 32 位寄存器，默认指定的是低 16 位寄存器。如果低 16 位区已分别指定为 S 、 D，则高 16 位分别自动指定为 S+1 、 D+1，本例中：

S+1(高位) = WR3， S(低位) = WR2

D+1(高位) = DT1， D(低位) = DT0

3) F3(DMV/) 指令助记符的最后一个字符为“ /”，表示在进行传送时，要对被传送的数据先进行取反，然后将结果送往目的寄存器区。

4) 源操作数和目的操作数都用寄存器方式寻址，源操作数在执行指令后内容不变，目的操作数则被覆盖，相当于执行数据拷贝操作。数据的传递关系与结果参看下表。

5) 与 F0(MV) 指令不同的是， S 和 D 不能用 IY 寄存器。 IX 和 IY 除用作索引寄存器外，还可以用作通用寄存器。当用作通用 16 位寄存器时，二者可单独使用；当用作 32 位存储区时，二者联用， IX 存低 16 位， IY 存高 16 位，因此程序中只能引用 IX， IY 由系统自动引用，无论是 S 还是 D 均如此。这个规则对于所有的双字 (32-bit) 指令都适用。

2. 位传输：F5(BTM)、F6(DGT)

[F5 BTM S, n, D]： 16 位二进制数的位传送指令。将一个 16 位二进制数的任意指定位，拷贝到另一个 16 位二进制数据中的任意指定位中去。

[F6 DGT S, n, D]： 16 位十六进制数的位传送指令。将一个 16 位数据按十六进制，传送若干位 (digit) 到另一个 16 位寄存器区中去。

例 3-23

梯形图

指令表

地址	指令	数据
0	ST	X0
1	F5(BTM)	
		WX0
		H0B05
		DT0

例题说明：

当控制触点 X0 接通时， WX0 中第 05 位数据传送到 DT0 中的第 11 位去，如下图所示。 WX0 中的数据由前面的程序赋值， DT0 中的数据可能已经赋值，也可能没有赋值，但是执行完该指令后， DT0 的第 11 位被赋值为 1 。

源数据区

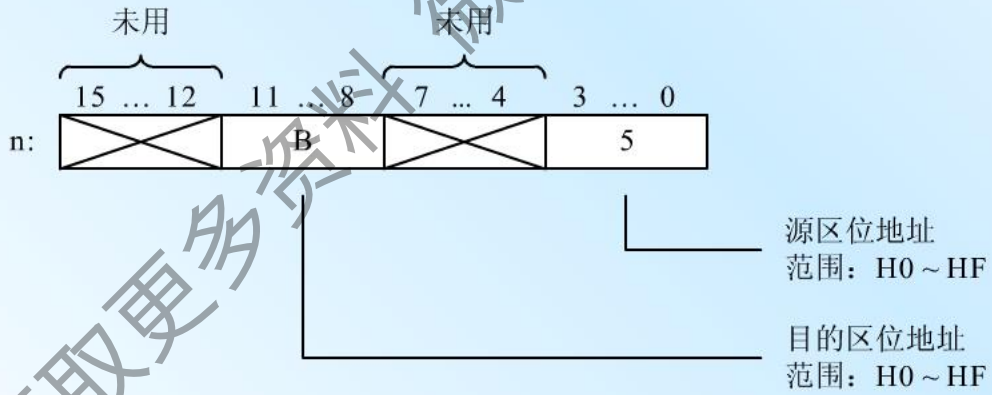
位地址	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WX0	1	1	0	0	1	0	0	0	1	0	1	1	0	1	1	1

1

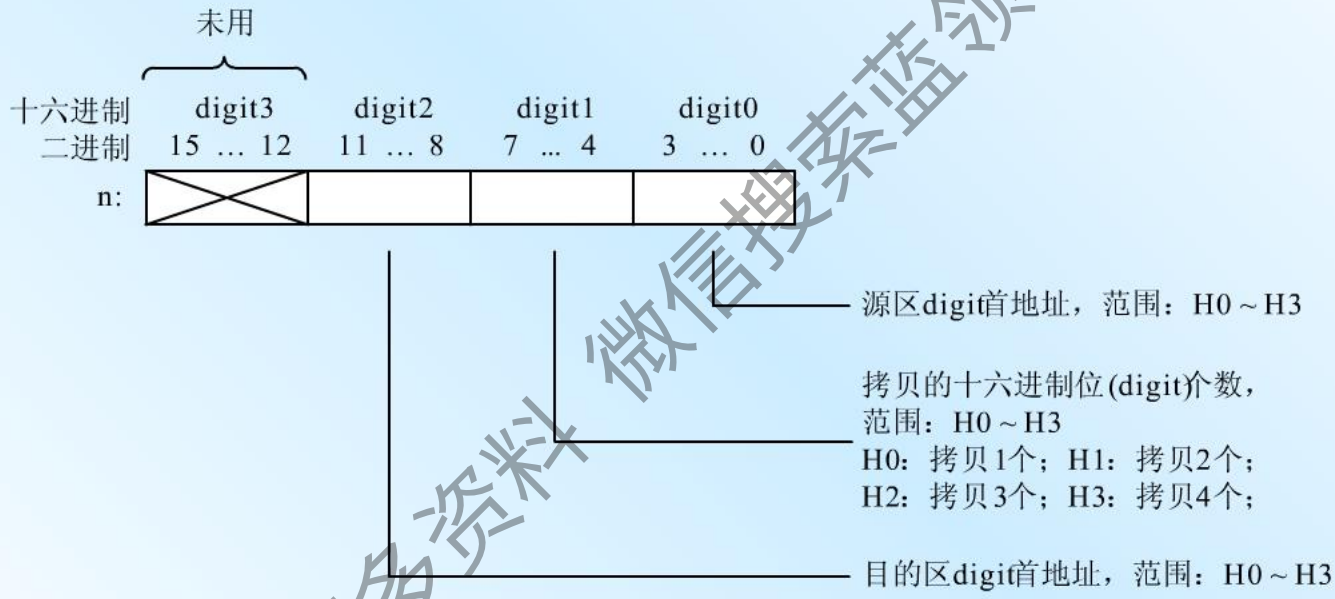
目的数据区

位地址	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DT0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

在 F5(BTM) 指令中， S 为源操作数，是被传送的 16 位常数或寄存器中的数据； D 为目的操作数，表示接收数据的 16 位目的寄存器； n 是 16 位的操作数，又称传输控制码，它指明了源操作数中哪一位数据将被传送以及传送到目的操作数中的哪一位置。在 n 中， bit0 ~ bit3 用以指定源操作数中哪一位将被传送， bit8 ~ bit11 用以指定被传送数据放在目的操作数的什么位置， bit4 ~ bit7 、 bit12 ~ bit15 这 8 位未用，可随便取值，不影响结果，为简便计，一般均取为 0 。因此，本例中源区位地址取为 H5 ，目的区位地址取为 HB 。 n 的设置参见下图。



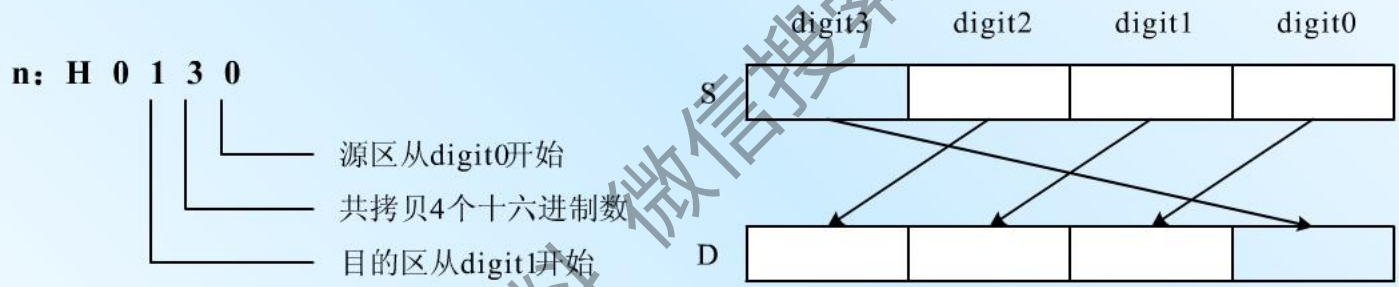
对于 F6(DGT)，在 n 的定义上有所不同，一是数据操作的最小单位为十六进制的 1 位，即 1digit，相当于二进制的 4bits；二是要拷贝的数据不像 F5 那样只有 1 位，而是有效范围内的任意位，因此还需要指定参与操作的位数。n 的设置可参考下图。



由图中可见，n 的 bit12 ~ bit15 未用，以十六进制表示，即 digit3 未用。

为了能够表示数据段，采用的是“首地址 + 段长度”的表示方式，即由 digit2 表示目的区首地址、 digit1 表示要拷贝的数据段位数、 digit0 表示源区首地址，这样进行操作的数据区地址就可唯一确定。

举例而言，若想将源区的 4 个十六进制位 (digit0 ~ digit3) 拷贝到目的区的 4 个十六进制位 (digit1 ~ digit3， digit0)，可将 n 取值为 H0130，其含义见左图，执行情况见右图。



值得注意的是，这里有个“循环”的概念，即如果目的区位数不够，自动回到最小位，再进行拷贝。如例中 S 的 digit3 应该送给 D 中的 digit4，但是 D 的最大位为 digit3，则该数据自动送往 D 的 digit0 。

3. 块传输指令：F10(BKMV)、F11(COPY)

1) F10(BKMV)：区块传输指令。

格式：**[F10 BKMV S1, S2, D]**

说明：数据段采用的是“首地址 + 尾地址”的表示方式，即将指定的以 S1 为起始地址、 S2 为终止地址的数据块拷贝到以 D 为起始地址的目的区中。要求 S1 和 S2 应为同一类型的寄存器，且 S2≥S1 。

2) F11(COPY)：块拷贝指令。

格式：**[F11 COPY S, D1, D2]**

说明：即将由 S 指定的 16-bit 常数或寄存器中的值重复拷贝到以 D1 为起始地址、 D2 为终止地址的目的区中。要求 D1 和 D2 应为同一类型的寄存器，且 D2≥D1 。

4. 数据交换指令：F15(XCH)、F16(DXCH)、F17(SWAP)

1) F15(XCH)： 16 位数据交换。

格式： **[F15 XCH D1 D2]**

说明：将 D1 和 D2 寄存器中的 16 位数据互相交换。

2) F16(DXCH)： 32 位数据交换。

格式： **[F16 DXCH D1, D2]**

说明：将 (D1+1，D1) 寄存器中的 32 位数据与 (D2+I， D2) 中的 32 位数据互换。

3) F17(SWAP)： 16 位数据的高低字节互换。

格式： **[F17 SWAP D]**

说明：将 D 寄存器中的 16 位数据高 8 位和低 8 位互换。

二、算术运算指令

算术运算指令共有 32 条，但是同前面介绍的比较指令类似，规律性很强。因此书中仅对其规律加以总结分析，掌握规律后，结合指令表，不难掌握这类指令。

1、指令分类

- 按照进位制可分为二进制 BIN 算术运算指令和 BCD 码算术运算指令，各为 16 条指令，后者在指令中增加大写字母“ B”以示区别。
- 按照参与运算的数据字长 (位数) 可以分为单字 (16-bit) 和双字 (32-bit) 指令，后者在助记符中以大写字母“ D”区别，在 FP1 的其它指令中也是采用这种方式。
- 按照运算规则可分为加、减、乘、除四则运算，以及加 1 、减 1 共六种基本运算。其中，加 1 和减 1 可以看作是加、减运算的特例。
- 按照参与运算的操作数的多少可分为一操作数、两操作数和三操作数。

2．操作数的数据范围

- 16 位二进制数：－32768 ~ 32767

  或 H8000 ~ H7FFF 。

- 32 位二进制数：－2147483648 ~ 2147483647

  或 H80000000 ~ H7FFFFFFF 。

- 4 位 BCD 码：　0 ~ 9999 。
- 8 位 BCD 码：　0 ~ 99999999 。

3．运算标志

算术运算要影响标志继电器，包括特殊内部继电器R9008、R9009和R900B。这里仅对影响情况做简单概括，详细情况需要结合具体的指令，参考手册学习掌握。

- R9008：错误标志。当有操作错误发生时，R9008接通一个扫描周期，并把发生错误的地址存入DT9018中。
- R9009：进位、借位或溢出标志。当运算结果溢出或由移位指令将其置1时，R9009接通一个扫描周期。
- R900B：0结果标志。当比较指令中比较结果相同，或是算术运算结果为0时，R900B接通一个扫描周期。

4．运算规则

1) 加法指令的算法

两操作数：**(D) + (S) → (D)**

三操作数：**(S1) + (S2) → (D)**

2) 减法指令的算法

两操作数：**(D) – (S) → (D)**

三操作数：**(S1) – (S2) → (D)**

3) 乘法指令的算法

(S1) × (S2) → (D)

乘法运算可能会导致 16 位数据升为 32 位，因此结果用 32 位存储；同理， 32 位乘法结果用 64 位存储。存储区自动取指定寄存器连续的高位寄存器，例如指定寄存器为 D，对于 64 位，结果自动存于 (D+3，D+2，D+1，D) 四个连续寄存器中。

4) 除法指令的算法

(S1) ÷ (S2) → (D)

除法运算在每次运算完后，商数保存于 D 中或 (D+1, D) 中。此外，还可能产生余数，如果是单字运算，可到 DT9015 中取余数；如果是双字运算，可到 (DT9016, DT9015) 中取余数。

5) 加 1 和减 1 指令算法

加 1 指令： **(D) + 1 → (D)**

减 1 指令： **(D) – 1 → (D)**

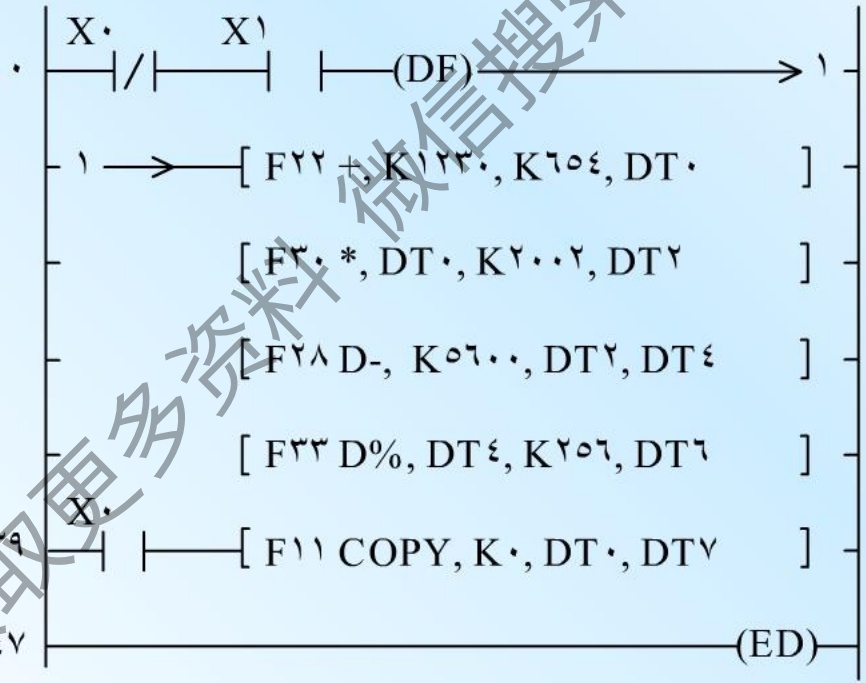
算术运算一般都是一次性的，而 PLC 采用的是扫描执行方式，因此该类指令常常和微分指令 (DF) 联合使用下面举例对算术指令加以说明。

5．举例

例 3-24：用算术运算指令完成算式，$\dfrac{5600-(1230+654)\times 2002}{256}$

这里包括了加、减、乘、除四种运算。要求 X1 闭合时开始运算，X0 闭合时各单元清零，且清零优先。

解：使用二进制 (BIN) 运算指令实现时，梯形图如下图。同样的功能也可采用 BCD 码运算指令实现。



三、数据比较指令

数据比较指令包括16位或32位数据比较指令、一个16位或32位数据与数据区间进行比较、数据块比较等5条指令……继电器R900……、R900A、R900B和R900C的状态来表示。

1．16位和32位数据比较指令：

F60（CMP）、F61（DCMP）

[F60 CMP，S1，S2]：16位数据比较指令。

[F61 DCMP，S1，S2]：32位数据比较指令。

该类指令的功能为：当控制触点闭合时，将S1指定数据与S2指定数据进行比较，比较的结果反映到标志位中。

表 3-14 16 位数据比较指令 F60(CMP) 对标志位影响

		标志位结果			
		R900A	R900B	R900C	R9009
		＞标志	＝标志	＜标志	进位标志
有符号数比较	S1 ＜ S2	OFF	OFF	ON	－
	S1 ＝ S2	OFF	ON	OFF	OFF
	S1 ＞ S2	ON	OFF	OFF	－
BCD 数据或无符号数比较	S1 ＜ S2	－	OFF	－	ON
	S1 ＝ S2	OFF	ON	OFF	OFF
	S1 ＞ S2	ON	OFF	OFF	OFF

如果程序中多次使用 F60(CMP) 指令，则标志继电器的状态总是取决于前面最临近的比较指令。为了保证使用中不出现混乱，一个办法是在比较指令和标志继电器前使用相同的控制触点来进行控制；另一个办法是在比较指令后立即使用相关的标志继电器。

2．16 位和 32 位数据区间比较指令

F62(WIN)、F63(DWIN)

[F62 WIN, S1, S2, S3]： 16 位数据区段比较指令。

[F63 DWIN, S1, S2, S3]： 32 位数据区段比较指令。

该类指令的功能为：当控制触点闭合时，将 S1 指定数据与 S2 指定下限、S3 指定上限的数据区间中的数据比较，比较的结果反映到标志位中。

表 3-15 16 位数据区间比较指令 F62(WIN) 对标志位影响

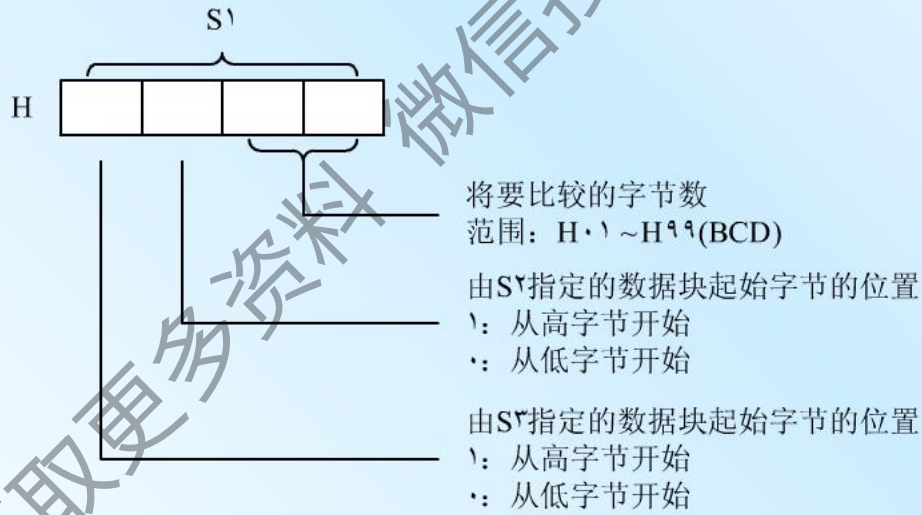
	标志位结果		
	R900A	R900B	R900C
	＞标志	＝标志	＜标志
S1 ＜ S2	OFF	OFF	ON
S2≤S1≤S3	OFF	ON	OFF
S1＞S3	ON	OFF	OFF

3．数据块比较指令：F64(BCMP)

[F64 BCMP, S1, S2, S3]：数据块比较指令。

该指令功能为：当控制触点闭合时，根据 S1 指定的比较参数，该参数包括数据块的起点和长度，比较由 S2 指定首地址的数据块和由 S3 指定首地址的数据块中的内容，当两个数据块完全相同时，特殊内部继电器 R900B 接通。

S1 指定的比较参数的定义见下图。



四、逻辑运算指令

该类指令很简单，包括与、或、异或和异或非4种。操作数均为16位，均有三操作数，将S1和S2分别进行上述4种运算，结果存于D中。

1. F65(WAN)：

格式：[F65 WAN S1, S2, D]
功能：16-bit data AND， 16 位数据“与”运算。

2. F66(WOR)：

格式：[F66 WOR S1, S2, D]
功能：16-bit data OR， 16 位数据“或”运算。

3. F67(XOR)：

格式：[F67 XOR S1, S2, D]
功能：16-bit data exclusive OR， 16 位数据“异或”运算。

4. F68(XNR)：

格式：[F68 XNR S1, S2, D]
功能：16-bit data exclusive NOR， 16 位数据“异或非”运算。

五、数据转换指令

数据转换指令包含各种数制、码制之间的相互转换，有二进制、十六进制及BCD码数据同ASCII码之间的相互转换，二进制数据与BCD码间的相互转换，指令较多。此外还有二进制数据的求反、求补、取绝对值；符号位的扩展等操作以及解码、编码、译码、数据分离、数据组合、数据查表等操作。通过这些指令，在程序中可以较好地解决PLC输入、输出的数据类型与内部运算数据类型不一致的问题。

1．区块检查码计算指令：F70 (BCC)

[F70 BCC S1，S2，S3，D]：这条指令常用于数据通信时检查数据传输是否正确。该指令是 FP1 指令系统中惟一的一条四操作数的指令。

2．码制变换指令：F71 ~ F83

1) F71 ~ F78：是 8 条三操作数的码制变换指令，分别实现十六进制数据、BCD 码、16 位二进制数据、32 位二进制数据与 ASCII 码间的互换，

2) F80 ~ F83：是 4 条双操作数的码制变换指令，分别实现 16 位和 32 位二进制数据与 BCD 码数据间的互换。

根据前面所学知识，不难推测出 S 和 D 可取用的寄存器范围。即目的寄存器不可取用 WX、K、H，当操作数是 32 位数据时，不可取用 IY 。

3．数据计算指令：F84 ~ F88

F84 ~ F88 这 5 条指令是将 D 指定的 16 位数据或 32 位二进制数据分别求反、求补、取绝对值，并将结果存储在 D 或 (D+1, D) 中。操作数 D 不可用寄存器 WX 、 K 、 H 。

4． 16 位数据符号位扩展指令：F89(EXT)

该指令的功能为：将 D 指定的 16 位数据的符号位全部拷贝到 D+1 寄存器的各个位中，保留 D 寄存器，扩展结果作为 32 位数据存储于 (D+1， D) 中。用该指令可将 16 位数据转变为 32 位数据。

5．编码／解码指令：F90 ~ F92

1) [F90 DECO S, n, D] ：解码指令。所谓解码，就是将若干位二进制数转换成具有特定意义的信息，即类似于数字电路中的 3-8 译码器的功能，将 S 指定的 16 位二进制数根据 n 规定的规则进行解码，解码的结果存于以 D 指定的 16 位寄存器作为首地址的连续区域。

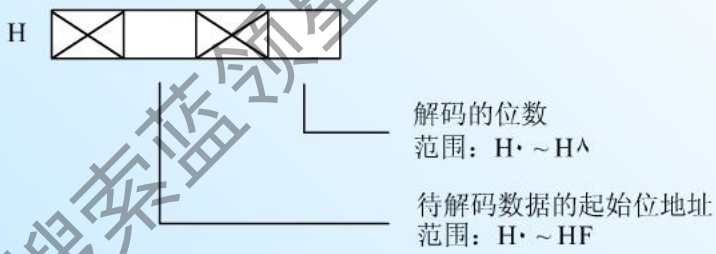


图 3-39 n 的格式示意图

2) [F91 SEGT S, D] ：位数据七段解码指令。是把一个 4 位二进制数译成七段显示码。即将 S 指定的 16 位数据转换为七段显示码，转换结果存储于以 D 为首地址的寄存器区域中。其中， S 为被译码的数据或寄存器。 D 为存放译码结果的寄存器首地址。

在执行该指令时，将每 4-bit 二进制码译成 7 位的七段显示码，数码的前面补 0 变成 8 位，因此，译码结果使数据位扩大了一倍。

3) [F92 ENCO S, n, D]： 编码指令。所谓编码，就是将具有特定意义的信息变成若干位二进制数。将 S 指定的 16 位二进制数据根据 n 的规定进行编码，编码结果存储于 D 指定的寄存器中。

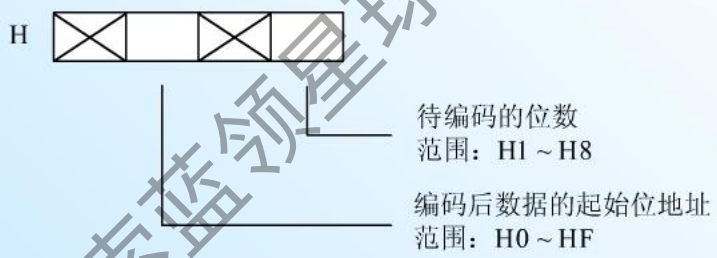


图 3-40 n 的格式示意图

其中， S 为被编码的数据或寄存器首地址， n 为编码控制字或存放控制字的寄存器。

nL 为 n 的 bit0 ~ bit3，用于设定编码数据的有效位长度， nL 的取值范围为 H1 ~ H8 。 S 的有效位长度＝ 2nL 。

nH 是 n 的 bit8 ~ bit11 ，用于设定 D 寄存器从何位开始存放结果， nH 取值范围为 H0 ~ HF 。

6．数据组和／分离指令：F93、F94

1) [F93 UNIT S, n, D]： 数据组合指令，其功能是将一组数据的低 4 位 (bit0 ~ bit3) 重新组成一个 16 位数据。

2) [F94 DIST S, n, D]： 数据分离指令，其功能和数据组合指令相反，是将一个 16 位数，每 4bits 为一组分成 4 组，按 n 规定的方式，存到结果寄存器 D 的低 4 位中去。

7．字符→ASCII 码转换指令：F95(ASC)

[F95 ASC S, D]： 将 S 指定的字符常数转换为 ASCII，转换后的结果存储于以 D 指定的 16 位寄存器开始的区域中。

8．表数据查找指令：F96(SRC)

[F96 SRC S1, S2, S3]： 在 S2(首地址) 和 S3(尾地址) 指定的数据区中查找与 S1 的内容相同的数据，并将查找到的数据的个数存储于特殊数据寄存器 DT9037 中，第一次发现该数据的位置存储于特殊数据寄存器 DT9038 中。

六、数据移位指令

FP1 高级指令系统中包含了位、字以及字段的左/右移位指令，共有16位数据的左/右移位、4位BCD码的左/右移位，字数据的左/右移位、16位数据的左/右循环移位等12条指令。其中位移位指令有进位标志位参与运算，并分为非循环移位指令（普通移位）和循环移位指令两种。这些移位指令比前文介绍过的SR指令的功能要强大得多，且不象SR那样每次只能移动1位，而是可以根据需要，在指令中设置一次移

1．16位数据的左/右移位指令

该类移位指令只是针对16位二进制数据，根据循环情况的不同又可分为普通（非循环）移位指令、循环移位指令和包含进位标志的循环移位指令三种情况。其区别主要在于移入位的数据处理上，简单地说，普通（非循环）移位指令不循环，移入位直接依次补0；循环移位指令移入位则由移出位补入；包含进位标志的循环移位指令移入位由进位标志依次补入。

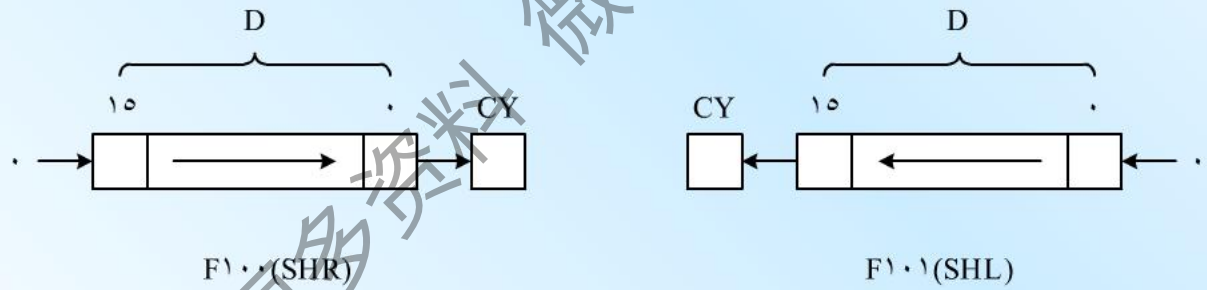
这里要注意的是，为了便于理解，也可将一次移动n位的过程理解成移动n次，每次移动1位，实际上指令是一次完成移位的。

1) 普通（非循环）移位指令

[F100 SHR, D, n]： 寄存器 D 中的 16 位数据右移 n 位，高位侧移入数据均为 0，低位侧向右移出 n 位，且第 n 位移入进位标志位 CY(R9009) 中。

[F101 SHL, D, n]： 寄存器 D 中的 16 位数据左移 n 位，高位侧向左移出 n 位，且第 n 位移入进位标志位 CY(R9009) 中，低位侧移入数据均为 0 。

其中， n 用于设定移位的位数，为常数或 16 位寄存器，取值范围为 K0 ~ K255 。

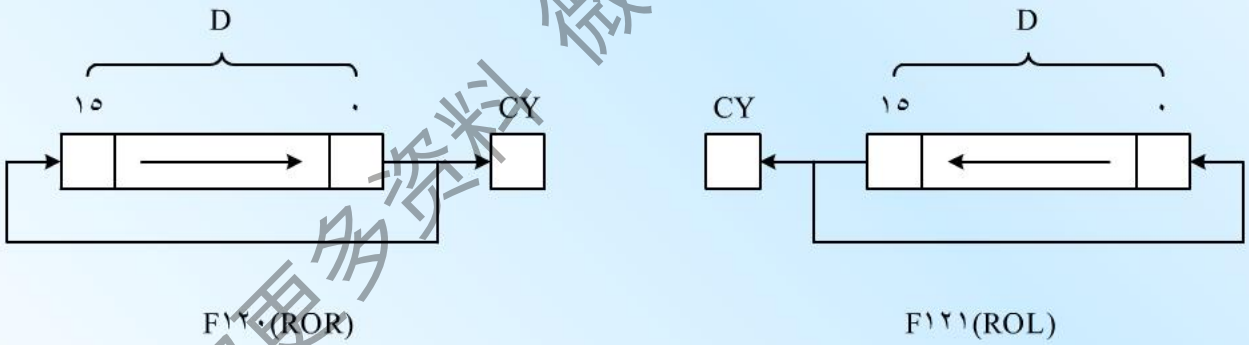


2) 循环移位指令

[F120 ROR, D, n]： 寄存器 D 中的 16 位数据右移 n 位，低位侧移出的 n 位依次移入高位侧，同时移出的第 n 位复制到进位标志位 CY(R9009) 中。

[F121 ROL, D, n]： 寄存器 D 中的 16 位数据左移 n 位，高位侧移出的 n 位依次移入低位侧，同时移出的第 n 位复制到进位标志位 CY(R9009) 中。

注意这两条指令与 F100 和 F101 的区别在于：这里是循环移位，而不是补 0 。

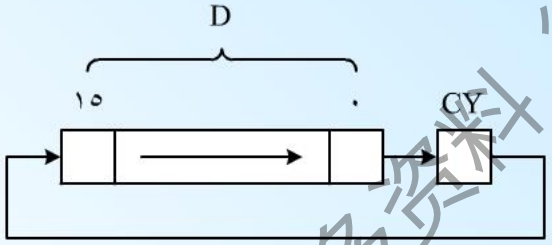


F١٢٠(ROR)　　F١٢١(ROL)

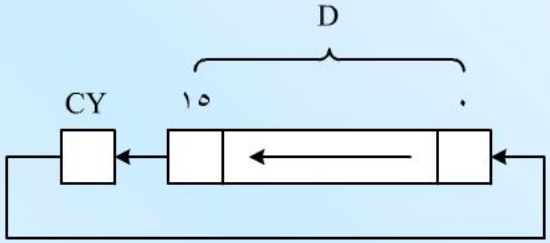
3) 包含进位标志的循环移位指令

[F122 RCR, D, n]： 寄存器 D 中的 16 位数据右移 n 位，移出的第 n 位移入进位标志位 CY，而进位标志位 CY 原来的数据则移入从最高位侧计的第 n 位。

[F123 RCL, D, n]： 寄存器 D 中的 16 位数据左移 n 位，移出的第 n 位移入进位标志位 CY，而进位标志位 CY 原来的数据则移入从最低位侧计的第 n 位。



F١٢٢(RCR)

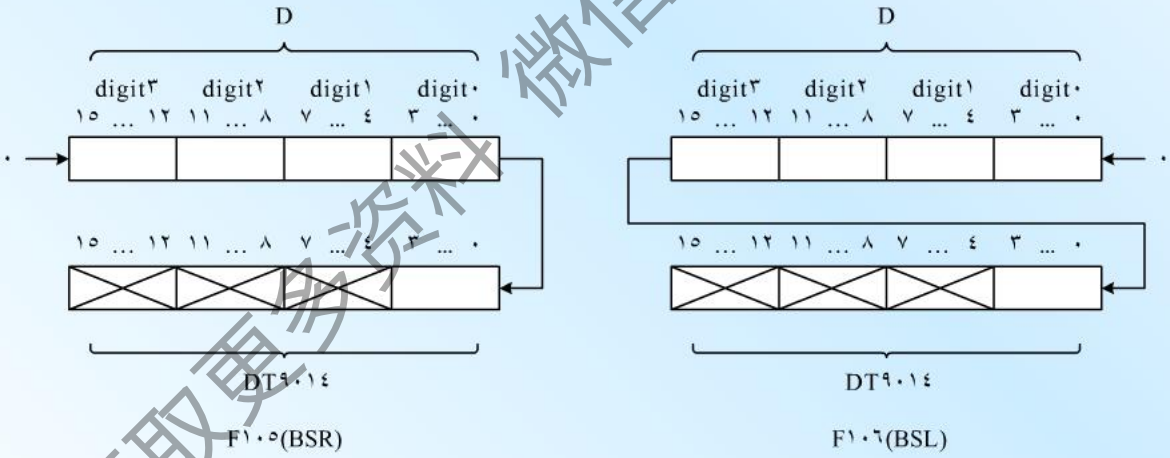


F١٢٣(RCL)

2．十六进制数的左／右移位指令

[F105 BSR, D]： 寄存器 D 中的 4 位十六进制数右移 1 位，相当于右移二进制的 4bits，移出的低 4bits 数据送到特殊数据寄存器 DT9014 的低 4bits，同时 D 的高 4bits 变为 0 。

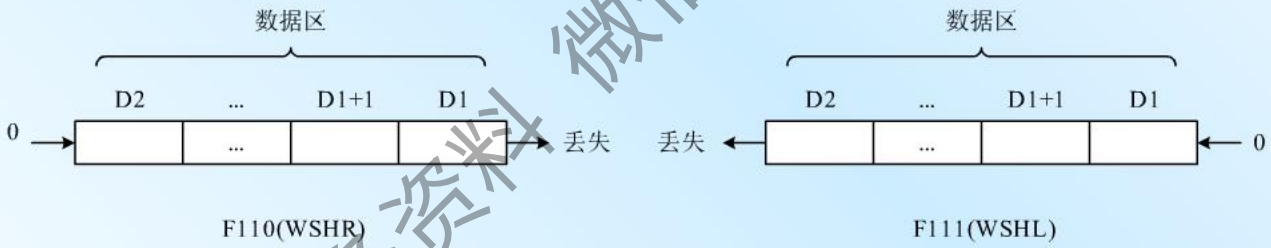
[F106 BSL, D]： 寄存器 D 中的 4 位十六进制数左移 1 位，相当于左移二进制的 4bits，移出的高 4bits 数据送到特殊数据寄存器 DT9014 的低 4bits，同时 D 的低 4bits 变为 0 。



3．数据区按字左／右移位指令

[F110 WSHR, D1, D2]：由 D1 为首地址， D2 为末地址定义的 16 位寄存器数据区，整体右移一个字，相当于二进制的 16-bit 。执行后，首地址寄存器的原数据丢失，末地址寄存器为 0 。

[F111 WSHL, D1, D2]：由 D1 为首地址， D2 为末地址定义的 16 位寄存器数据区，整体左移一个字，相当于二进制的 16-bit 。执行后，首地址寄存器为 0 ，末地址寄存器的原数据丢失。

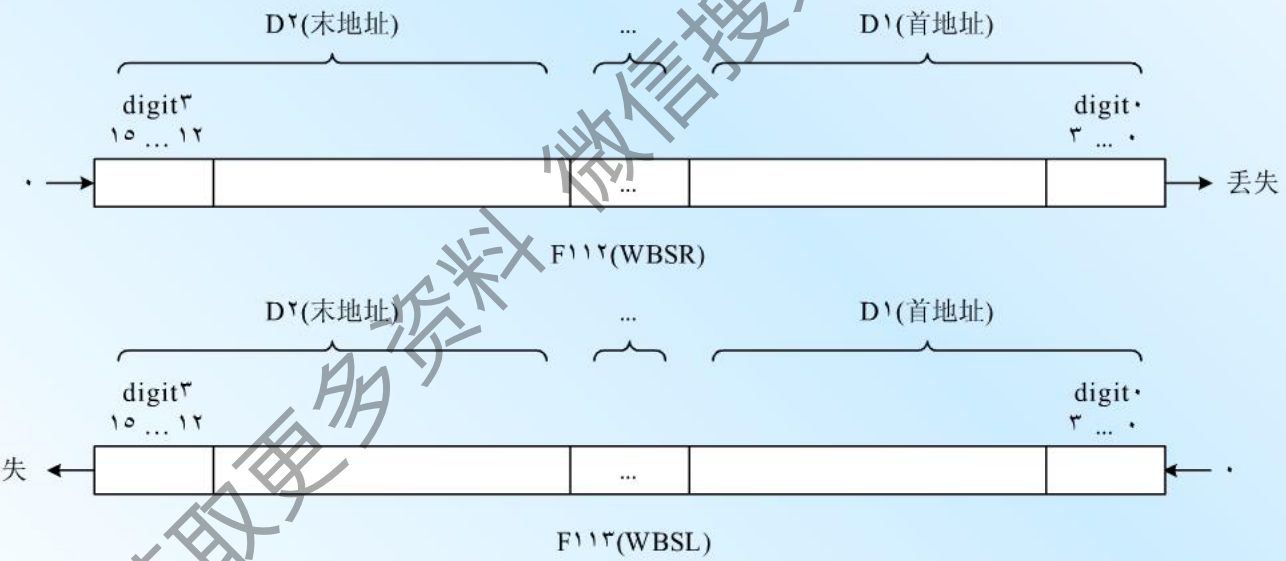


同前面针对数据区操作的高级指令一样， D1 和 D2 应是同一类型的寄存器，且末地址寄存器号应大于或等于首地址寄存器号，即 D2≥D1 。此外，还要注意的是首尾地址的编排顺序是左边为末地址、右边为首地址。

4．十六进制数据区的左／右移位指令

[F112 WBSR，D1，D2]： 由 D1 为首地址， D2 为末地址定义的 16 位寄存器数据区，整体右移一个十六进制数，相当于二进制的 4bits 。

[F113 WBSL，D1，D2]： 由 D1 为首地址， D2 为末地址定义的 16 位寄存器数据区，整体左移一个十六进制数，相当于二进制的 4bits 。



七、位操作指令

位操作就是指被操作的对象不是字，而是字中的某一位或几位。FP1系列PLC具有较强的位操作能力，可以进行16位数据的位置位（置1）、位复位（清0）、位求反以及位测试，还可计算16位或32位数据中，位值为“1”的位数。位操作指令共有6条，可分为位处理指令和位计算指令两类

1．位处理指令

- [F130 BTS, D, n]：位置 1 指令。
- [F131 BTR, D, n]：位清 0 指令。
- [F132 BTI, D, n]：位求反指令。
- [F133 BTT, D, n]：位测试指令。

前 3 条指令的功能是对位进行运算处理，分别对 D 寄存器中、位地址为 n 的数据位进行置位 (置 1) 、复位 (清 0) 、求反。其中，由于 n 用来表示 16 位数据的位地址，因此取值范围为 K0 ~ K15。

第 4 条指令用于测试 16 位数据 D 中任意位 n 的状态为“ 0” 还是为“ 1”。测试的结果存储在内部继电器 R900B 中，如果测试结果为 0，则 R900B=1；测试结果为 1， R900B=0 。

2．位计算指令

位计算指令就是计算寄存器的数据或常数中有多少位是“1”。

- **[F135 BCU，S，D]：** 16位位计算指令。
- **[F136 DBCU，S，D]：** 32位位计算指令。

F135(BCU)和F136(DBCU)的功能是分别统计S指定的16位和32位数据中位值为“1”的位的个数，并把统计的结果存储于D指定的存储区中。

八、特殊指令

1、时间变换指令：F138(HMSS)、F139(SHMS)

FP1-C24 以上机型均有日历及实时时钟功能。使用手持编程器或编程软件将年、月、日、时、分、秒、星期等的初值设置到特殊数据寄存器 DT9054 ~ DT9057 中，即可实现自动计时，即使断电后，计时也不会间断。校表时，采用舍入法，DT9058 是 30 秒校表寄存器，当 DT9058 置入“1”时，若秒位显示小于 30 秒则舍去，若大于 30 秒，则分位加“1”。

1) F138(HMSS)

格式：[F138 HMSS S, D]

功能：将以时 / 分 / 秒格式表示的时间数据，变换成以秒为单位的时间数据。将 (S+1, S) 中存放的时 / 分 / 秒数据转换为秒数据，结果存放于寄存器 (D+1, D) 中。在这里，S 和 D 中的数据均用 BCD 码表示。表示形式如下图所示。

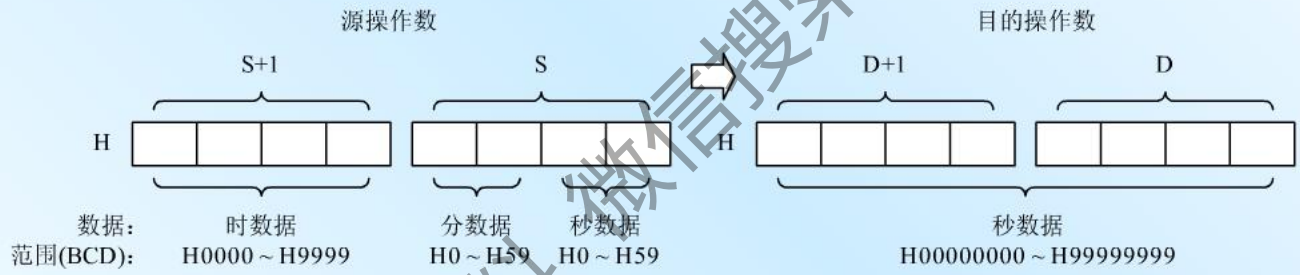


图 3-47 F138 指令中 S 和 D 的含义

2) F139(SHMS)

格式：[F139 SHMS S, D]

功能：将以秒为单位的时间数据，变换成以时 / 分 / 秒格式表示的时间数据。功能与 F138 完全相反。

2、进位位（CY）的置位和复位指令：F140(STC)、F141(CLC)

格式：[F140 STC]、[F141 CLC]

功能：F140(STC)和F141(CLC)指令是FP1高级指令中仅有的两条无操作数的指令，其功能是将特殊内部继电器R9009(进位标志位)置位和复位，即将R9009置为1或者清0。

3、刷新部分I/O指令：F143(IORF)

格式：[F143 IORF D1, D2]

功能：刷新指定的部分I/O点。

F143指令只要触发信号接通，即使在执行程序阶段，也能立即将输入(WX)或输出(WY)寄存器D1至D2的内容刷新，避免由扫描时间造成的延时。该指令要求D1和D2为同一类型的操作数，且D2≥D1。

4、串行数据通信指令：F144（TRNS）

格式：[F144 TRNS S, n]

功能：通过RS232串行口与外设通讯，以字节为单位，发送或接收数据。一般型号末端带“C”的PLC带有RS232串行口。

其中，S为发送或接收数据的寄存器区首地址，且S只能使用数据寄存器DT。寄存器S用作发送或接收监视之用，之后的寄存器S+1、S+2、……，存放着发送或接收的数据。也就是说，S+1为发送和接收数据的首地址，数据存放在S+1及以后的寄存器中。n则用来设定要发送的字节数。

1) **数据发送：**特殊内部继电器R9039是发送标志继电器，发送过程中R9039为OFF状态，结束后为ON状态。

2) **数据接收：**特殊内部继电器R9038是接收标志继电器，接收过程中R9038为OFF状态，结束后为ON状态。

5、并行打印输出指令：F147(PR)

格式：[F147 PR S, D]

功能：通过并行通讯口打印输出字符。每次执行打印指令可连续打印 12 个字符，并占用 37 个扫描周期，由 Y8 自动发出打印脉冲。C24 以上晶体管输出型的 PLC 具有并行打印输出功能。

其中，S 指定了要输出字符的首地址，S 和随后的 S+1 、S+2 等保存的必须是字符型 ASCII 数据。D 为打印机信号输出，只可用 WY 输出继电器，且 0 ~ 8 位与打印机对应，PLC 与打印机之间的连线如下表所示。

表 3-21 FP1 与打印机连接端点

晶体管输出型 FP1	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	COM	DC+5V
打印机	Data 1	Data2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8	Strobe	COM	DC+5V

6、自诊断错误设置指令：F148(ERR)

格式：[F148 ERR n]

功能：将某特殊状态设置为自诊断错误，或者将由自诊断错误 E45 、 E50 或 E200 ~ E299 引起的错误状态复位。 F148 指令的运行由 n 决定， n 为自诊断错误代码，范围为 0 和 100 ~ 299 。

- n = 0 ：清除由自诊断错误 E45 、 E50 或 E200 ~ E299 引起的错误状态；
- n = 100 ~ 299 ：将指令的触发信号设置为第 n 号自诊断错误。具体内容请参见手册的“错误代码表”。

7、信息显示指令： F149(MSG)

格式：[F149 MSG S]

功能：将 S 指定的字符常数 (以 M 开始的字符串) 显示在 FP 编程器 II 屏幕上。

8、时间运算指令：F157(CADD)、F158(CSUB)

1) F157(CADD)

格式：[F157 CADD S1, S2, D]

功能：在 (S1+2, S1+1, S1) 指定的日期 (年、月、日) 和时间 (时、分、秒) 数据中加上 (S2+1, S2) 指定的时间数据，所得的结果 (年、月、日、时、分、秒) 存放在 (D+2, D+1, D) 中，日期、时间数据均用 BCD 码表示。

2) F158(CSUB)

格式：[F158 CSUB S1, S2, D]

功能：同 F157(CADD) 类似，只是相加运算变为相减运算，详细用法请参考手册。

第三章结束

哈尔滨理工大学
电气与电子工程学院
电气技术教研室
2003 年 3 月 24 日

第一节 PLC 编程特点和原则

b. PLC 的编程特点

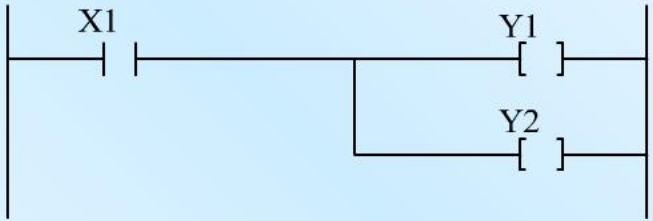
梯形图编程是 PLC 编程中最常用的方法。它源于传统的继电器电路图，但发展到今天两者之间有了较大的差别。

2. 程序的执行顺序

c. 继电器梯形图和 PLC 梯形图执行顺序的比较

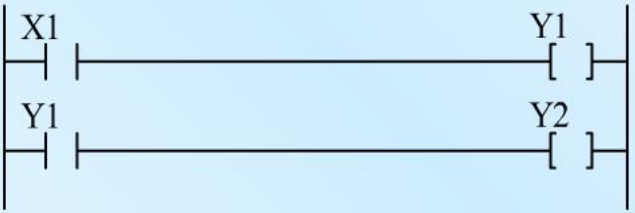
继电器梯形图

X1 闭合后，Y1 、Y2 同时得电

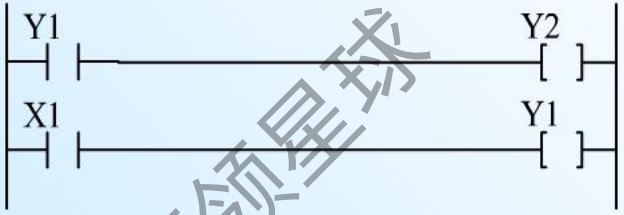


PLC 梯形图

X1 闭合后， Y1 先输出， Y2 后输出



X1 闭合后， Y1 、 Y2 在
同一扫描周期内动作



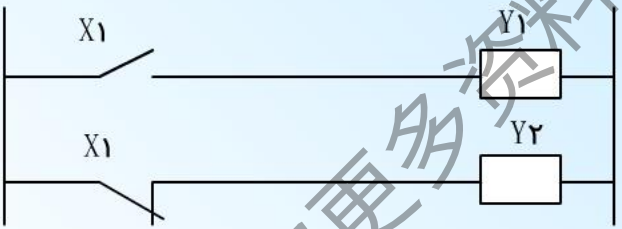
X1 闭合后， Y1 、 Y2 在
两个扫描周期内动作

扫描执行方式

优点： 可滤掉高频干扰，增强抗干扰能力。

缺点： 产生响应滞后，影响可靠性。

10. 继电器自身的延时效应



X1 动作时， Y1 、 Y2
不同时得电与断电

X1 动作时， Y0 、 Y1
同时得电与断电

1.PLC 中的软继电器

所谓软继电器是指 PLC 存储空间中的一个可以寻址的位。

在 PLC 中，软继电器种类多、数量大。

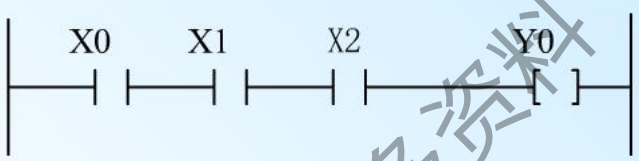
如 FP1-C24，共有 R 内部继电器 1008 个，特殊继电器 64 个，定时器 / 计数器 144 个。

寄存器中触发器的状态可以读取任意次，相当于每个继电器有无数个常开和常闭触点。

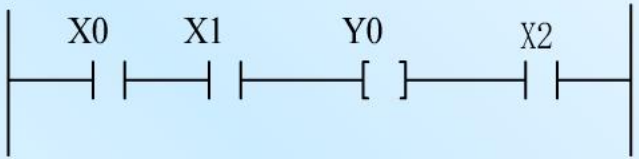
a. PLC 的编程原则

- 输入 / 输出继电器、内部辅助继电器、定时器、计数器等器件的**触点可以多次重复使用**，无需复杂的程序结构来减少触点的使用次数。
- **梯形图每一行都是从左母线开始，线圈终止于右母线。触点不能放在线圈的右边。**

接点和线圈的顺序：

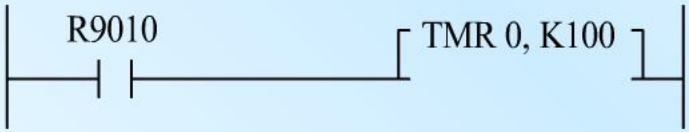


正确程序

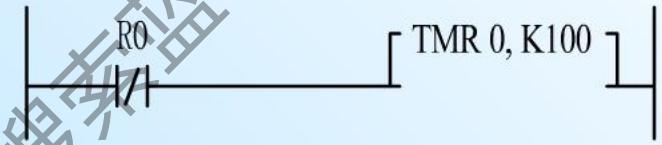


错误程序

1. **除步进程序外，任何线圈、定时器、计数器、高级指令等不能直接与左母线相连。**如果需要任何时候都被执行的程序段，可以通过特殊内部常闭继电器或某个内部继电器的常闭触点来连接。



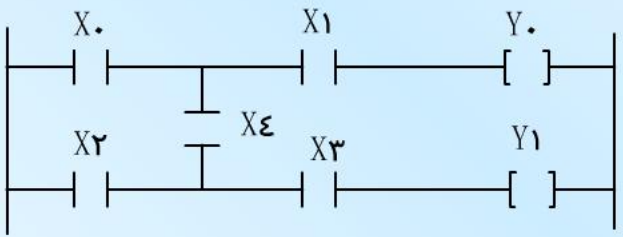
利用内部特殊继电器
实现常闭输出



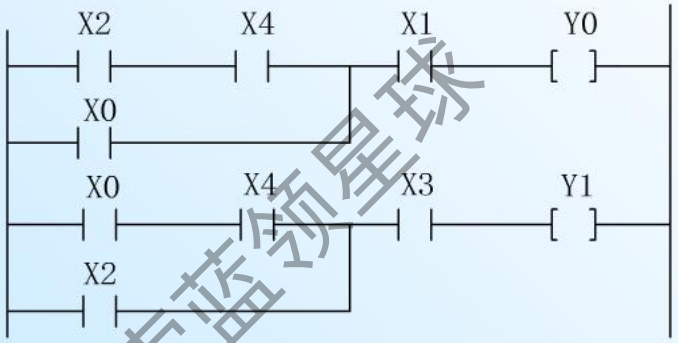
利用内部继电器常闭接点
实现常闭输出

4. 在程序中，**不允许同一编号的线圈两次输出。**下面的梯形图是不允许的。

1. 不允许出现桥式电路。



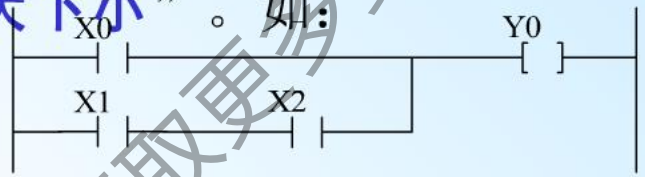
错误的桥式电路



桥式电路的替代电路

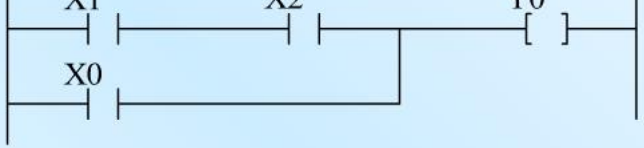
注意： 触点应画在水平线上，不能画在垂直分支上。

2. 程序的编写顺序应按**自上而下、从左至右**的方式编写。为了减少程序的执行步数，程序应为“**左大右小，上大下小**”。如：



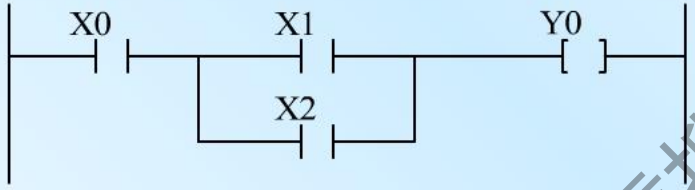
0	ST	X0
1	ST	X1
2	AN	X2
3	ORS	
4	OT	Y0

不符合上大下小的电路，共 5 步



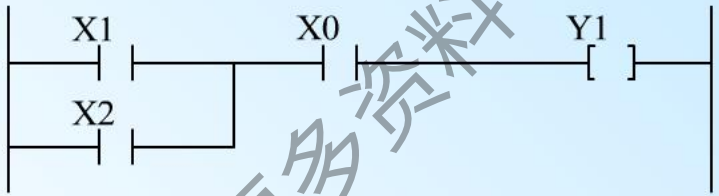
٠	ST	X٠
١	AN	X٢
٢	OR	X٠
٣	OT	Y٠

符合上大下小的电路，共 4 步



٠	ST	X٠
١	ST	X١
٢	OR	X٢
٣	ANS	
٤	OT	Y٠

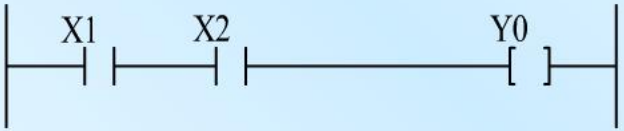
不符合左大右小的电路，共 5 步



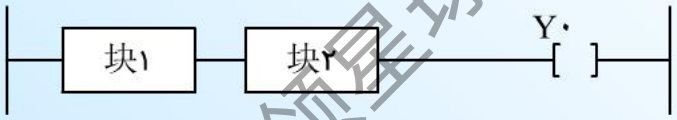
0	ST	X 1
1	OR	X 2
2	AN	X 0
3	OT	Y 1

符合左大右小的电路，共 4 步

b. AND 运算



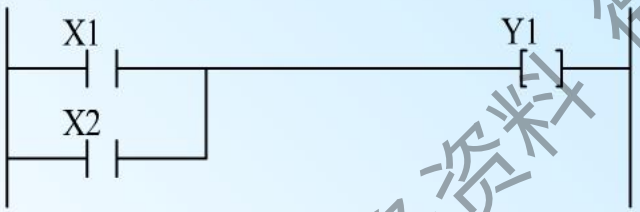
AND 电路，Y0 接受 X1 和 X2 的 AND 运算结果



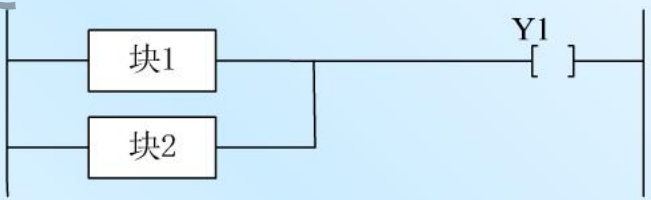
AND 扩展电路，Y0 接受块 1 和块 2 的 AND 运算结果

例如：只有当设备的状态为就绪状态，并且按下“开始”按扭，设备才能开始工作。

g. OR 运算



OR 电路，Y1 接受的是 X1 和 X2 的 OR 运算结果

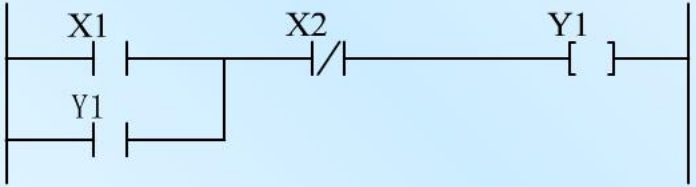


OR 扩展电路，Y1 接受的是块 1 和块 2 的 OR 运算结果

例如：在锅炉控制过程中，无论是水罐的压力过高，还是水温过高都要产生声光报警。

a. 自锁（自保持）电路

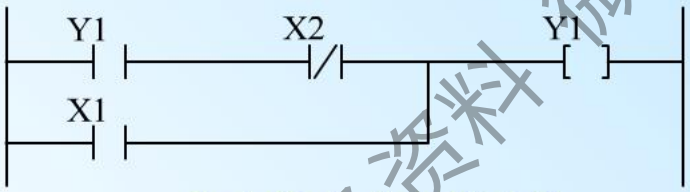
自锁电路分为：关断优先式和启动优先式



0	ST	X1
1	OR	Y1
2	AN/	X2
3	OT	Y1

关断优先式自锁电路

关断优先式自锁电路：当执行关断指令，X2 闭合时，无论 X1 的状态如何，线圈 Y1 均不得电。



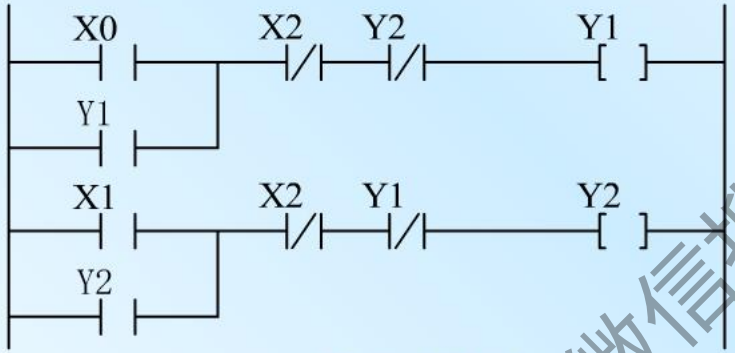
0	ST	Y1
1	AN/	X2
2	OR	X1
3	OT	Y1

启动优先式自锁电路

启动优先式自锁电路：当执行启动指令，X1 闭合时，无论 X2 的状态如何，线圈 Y1 都得电。

a. 互锁电路

互锁电路用于不允许同时动作的两个继电器的控制，如电机的正反转控制。



互锁控制电路

0	ST	X0
1	OR	Y1
2	AN/	X2
3	AN/	Y2
4	OT	Y1
5	ST	X1
6	OR	Y2
7	AN/	X2
8	AN/	Y1
9	OT	Y2

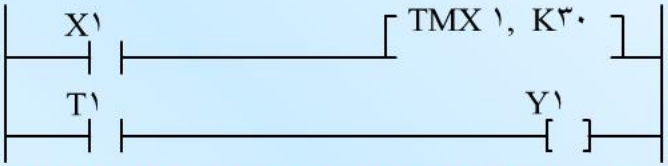
·时间电路

时间电路主要用于延时、定时和脉冲控制中。

时间控制电路既可以用定时器实现也可以用标准时钟脉冲实现。

在 FP1 型 PLC 内部有多达 100 个定时器和三种标准时钟脉冲（ 0.01s 、 0.1s 、 1s ）可用于时间控制。

– 延时电路

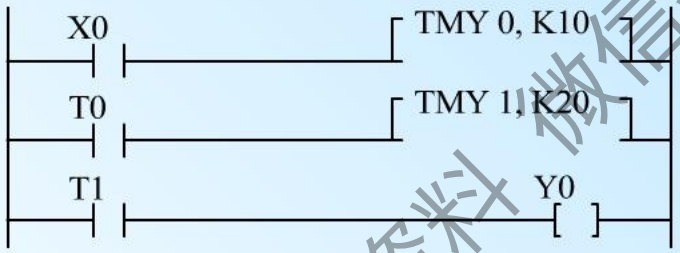


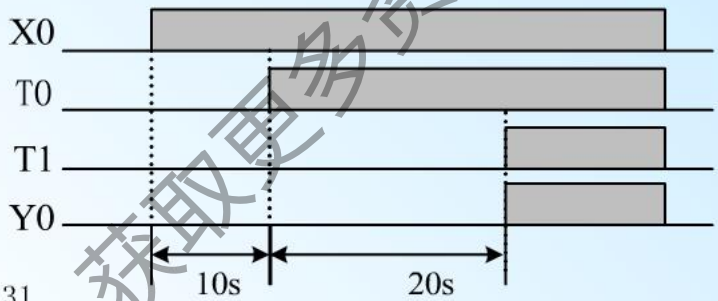

```
0   ST    X 1
1   TMX   1
      K  30
4   ST    T 1
5   OT    Y 1
```

时间继电器 TMX1 起到延时 30×0.1=3 秒的作用。

下图利用两个定时器组合以实现长延时。

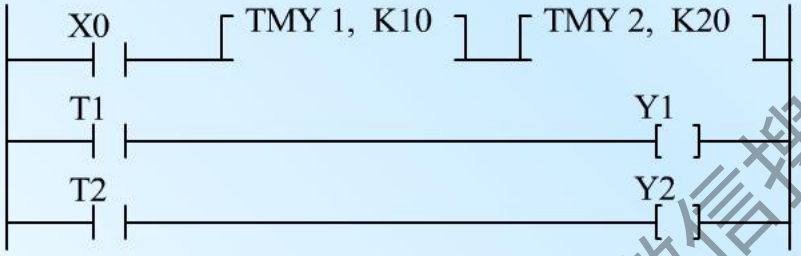
即 Y0 在 X0 闭合 30 秒之后得电。



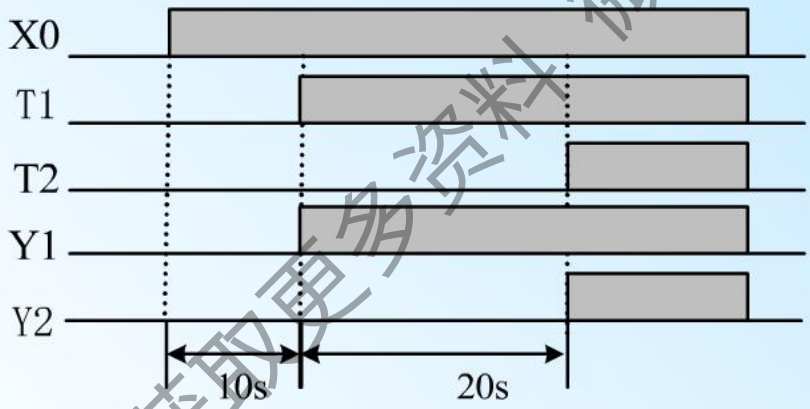


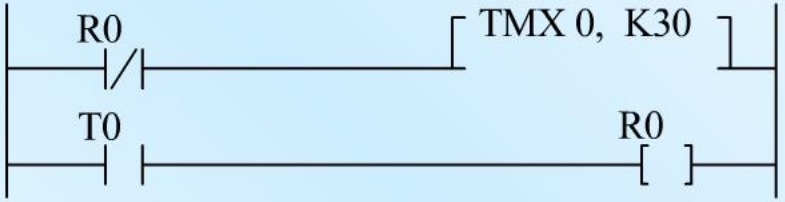

```
٠    ST    X٠
١    TM    Y٠
       K   ١٠
٥    ST    T٠
٦    TM    Y١
       K   ٢٠
١٠   ST    T١
١١   OT    Y٠
```

下图利用定时器串联实现长延时。即 Y2 在 X0 闭合 30 秒之后导通。



٠	ST	X٠
١	TM	Y١
	K	١٠
٥	TM	Y٢
	K	٢٠
٩	ST	T١
١٠	OT	Y١
١١	ST	T٢
١٢	OT	Y٢






```
0    ST/   R0
1    TM    X0
          K   30
4    ST    T0
5    OT    R0
```

利用定时器可以方便地产生脉冲序列。在上图程序的运行过程中，**R0 每隔 3 秒产生一次脉冲，其脉宽为一个扫描周期。**

在 FP1 的内部有七种标准的时钟脉冲继电器，分别为

R9018（0.01s），R9019（0.02s），R901A（0.1s），R901B（0.2s），R901C（1s），R901D（2s），R901E（1min）。若需要这几种时间的脉冲，可直接利用这几个时间脉冲发生器。

a. 分支电路

分支电路主要用于一个控制电路导致几个输出的情况。

例如，开动吊车的同时打开警示灯。

下图中，当 X0 闭合后，线圈 Y1 、 Y2 同时得电。

٠	ST	X٠
١	OT	Y١
٢	OT	Y٢

b. 电动机正反转控制

1. 系统结构

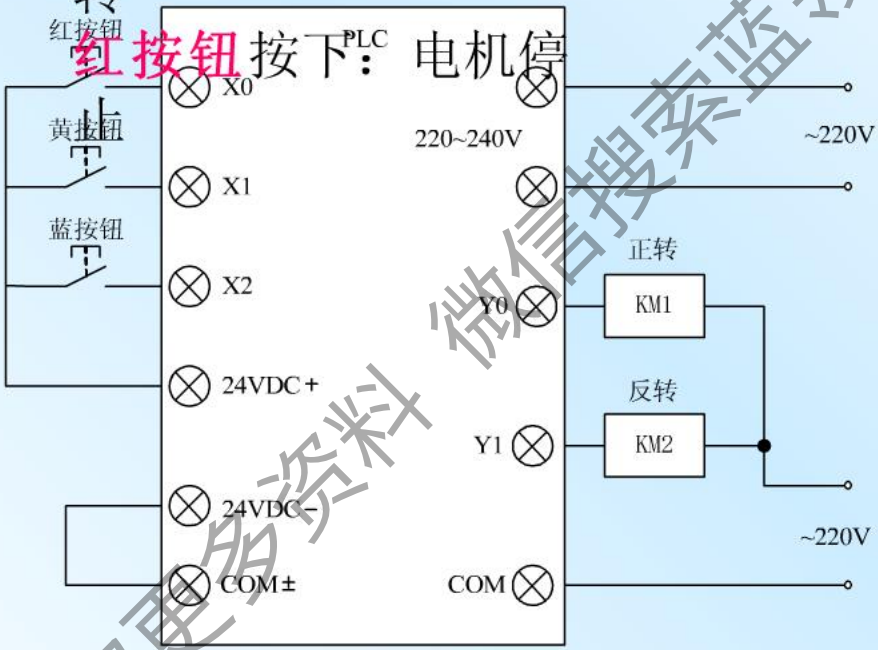
利用 PLC 控制一台异步电动机的正反转。

输入端直流电源 E 由 PLC 内部提供，可直接将 PLC 电源端子接在开关上。交流电源则是由外部供给。

黄按钮按下：电机正转

蓝按钮按下：电机反转

红按钮按下：电机停



PLC控制电动机正反转外部接线图

按动黄按钮时：

① 若在此之前电机没有工作，则电机正转启动，并保持电机正转；

② 若在此之前电机反转，则将电机切换到正转状态，并保持电机正转；

③ 若在此之前电机的已经是正转，则电机的转动状态不变。

电机正转状态一直保持到有篮按钮或红按钮按下为止。

按动蓝按钮时：

① 若在此之前电机没有工作，则电机反转启动，并保持电机反转；

② 若在此之前电机正转，则将电机切换到反转状态，并保持电机反转；

③ 若在此之前电机的已经是反转，则电机的转动状态不变。

电机反转状态一直保持到有黄按钮或红按钮按下为止。

按下红按钮时：停止电机的转动

注：电机不可以同时进行正转和反转，否则会损坏系统

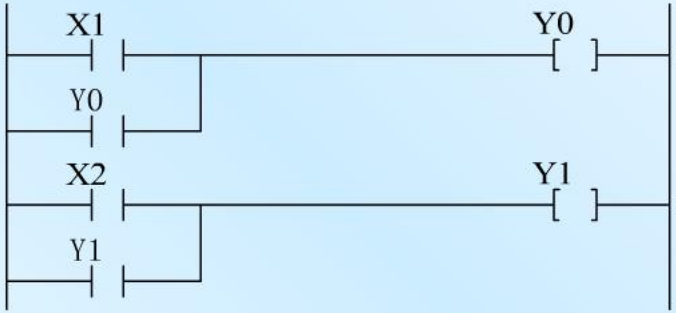
– PLC 的 I/O 点的确定与分配

电机正反转控制 PLC 的 I/O 点分配表

PLC 点名称	连接的外部设备	功能说明
X0	红按钮	停止命令
X1	黄按钮	电机正转命令
X2	蓝按钮	电机反转命令
Y0	正转继电器	控制电机正转
Y1	反转继电器	控制电机反转

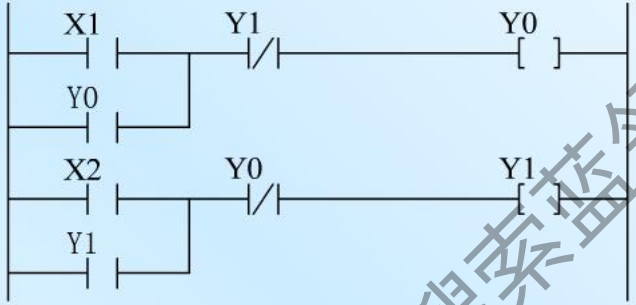
– 系统编程分析和实现

电机初步正转控制电路



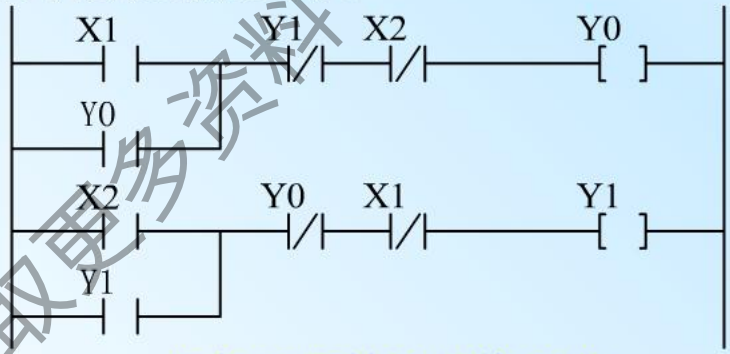
电机初步正反转控制电路

系统要求电机不可以同时进行正转和反转， 如下图所示利用互锁电路可以实现。



电机正反转的互锁电路

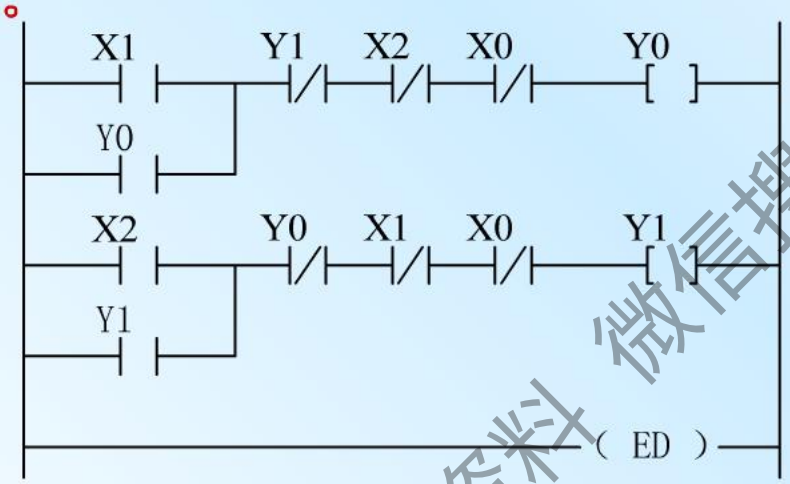
利用正转按钮来切断反转的控制通路；利用反转按钮来切断正转的控制通路。



电机正反转的切换电路

当按下红按钮时，无论在此之前电机的转动状态如何，都停止电机的转动。

利用红色按钮同时切断正转和反转的控制通路



0	ST	X1
1	OR	Y0
2	AN/	Y1
3	AN/	X2
4	AN/	X0
5	OT	Y0
6	ST	X2
7	OR	Y1
8	AN/	Y0
9	AN/	X1
10	AN/	X0
11	OT	Y1
12	ED	

电机正反转的最终控制程序

a. 锅炉点火和熄火控制

锅炉的点火和熄火过程是典型的定时器式顺序控制过程。

控制要求：

点火过程： 先启动引风，5 分钟后启动鼓风，2 分钟后点火燃烧。

熄火过程： 先熄灭火焰，2 分钟后停鼓风，5 分钟后停引风。

5.PLC 的 I/O 点的确定与分配

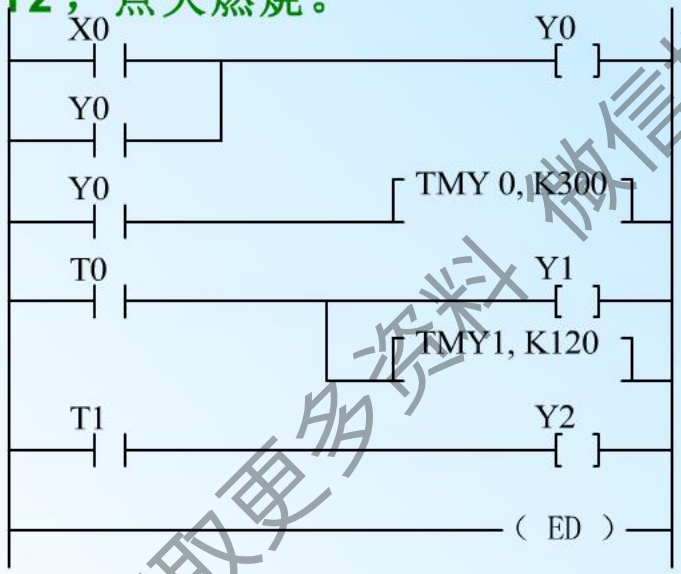
锅炉点火和熄火控制 PLC 的 I/O 点分配表

PLC 点名称	连接的外部设备	功能说明
X0	蓝按钮	点火命令
X1	红按钮	熄火命令
Y0	控制继电器 1	控制引风
Y1	控制继电器 2	控制鼓风
Y2	控制继电器 3	控制点火开关

(1) 点火过程

工作过程：

当蓝按钮按下（X0 接通）后，启动引风（Y0 输出。因 X0 选用的是非自锁按钮，故需要利用自锁电路锁住 Y0，同时利用 Y0 触发时间继电器 T0， T0 延时 300s （5 分钟）后，**输出继电器 Y1 动作，即启动鼓风。**同时利用 T0 触发定时继电器 T1， T1 延时 120s （2 分钟）后，**输出 Y2，点火燃烧。**



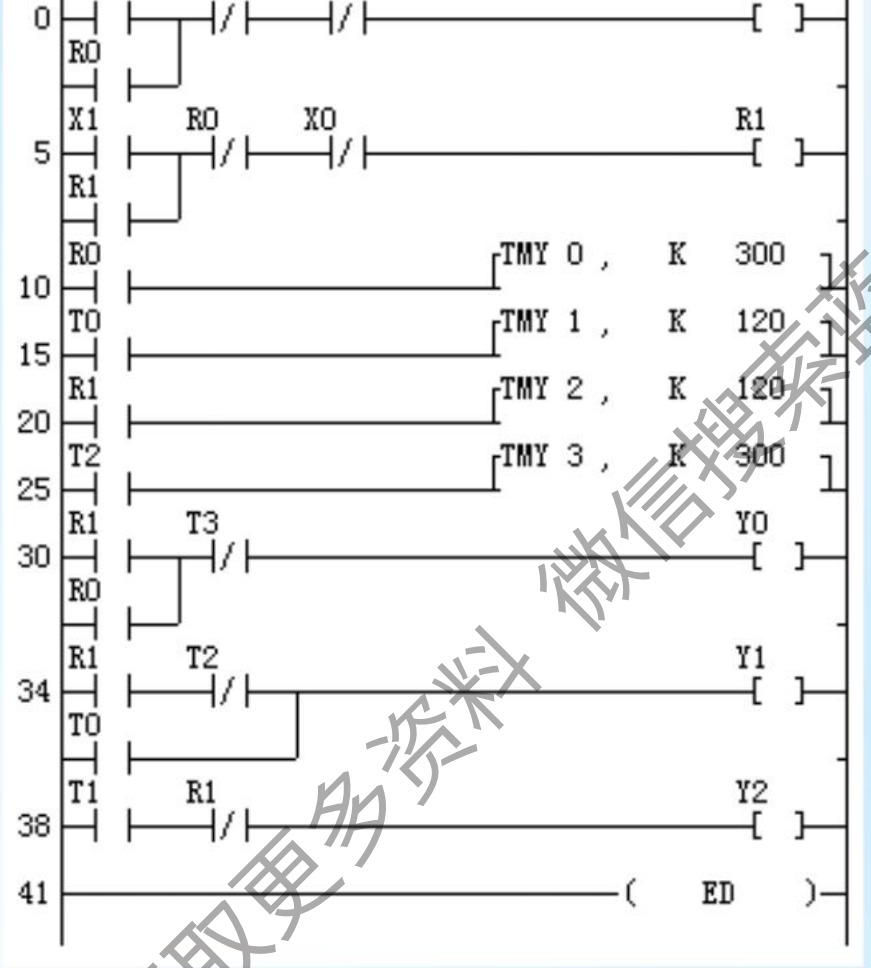
٠	ST	X٠
١	OR	Y٠
٢	OT	Y٠
٣	ST	Y٠
٤	TM	Y٠
	K	٣٠٠
٨	ST	T٠
٩	OT	Y١
١٠	TM	Y١
	K	١٢٠
١٤	ST	T١
١٥	OT	Y٢
١٦	ED	

锅炉点火过程控制程序

（2）系统的点火和熄火过程的综合程序

下面所示的两个程序都可以实现锅炉系统的点火和熄火过程控制，但实现的方式不同。

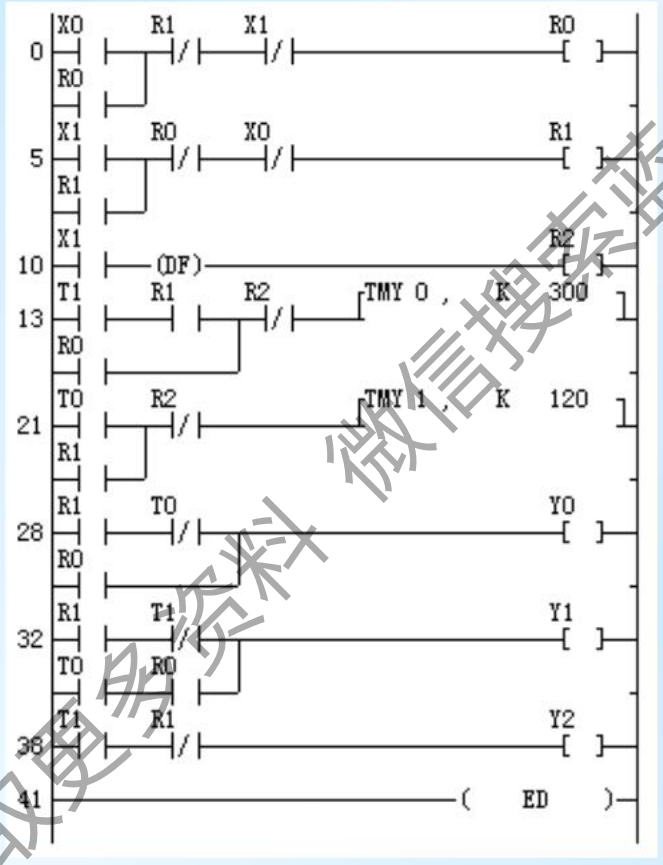
图 (a) 程序利用了 4 个时间继电器，但程序的逻辑关系比较简单易懂。




```
 1   OR      R0
 2   AN/     R1
 3   AN/     X1
 4   OT      R0
 5   ST      X1
 6   OR      R1
 7   AN/     R0
 8   AN/     X0
 9   OT      R1
10   ST      R0
11   TM      Y0
       K     300
15   ST      T0
16   TM      Y1
       K     120
20   ST      R1
21   TM      Y2
       K     120
25   ST      T2
26   TM      Y3
       K     300
30   ST      R1
31   OR      R0
32   AN/     T3
33   OT      Y0
34   ST      R1
35   AN/     T2
36   OR      T0
37   OT      Y1
38   ST      T1
39   AN/     R1
40   OT      Y2
41   ED
```


（a） 锅炉系统点火和熄火过程的综合程序（1）

图 (b) 程序利用了 2 个时间继电器，**节约了 2 个时间继电器，**但控制逻辑相对复杂些。



a. 房间灯的控制

现在一些宾馆和家庭客厅中的装饰灯，是利用一个开关来实现不同的控制组合。

例如，房间内有 1，2，3 号三个灯

按动一下开关，三个灯全亮；

再按一下，1，3 号灯亮，2 号灭；

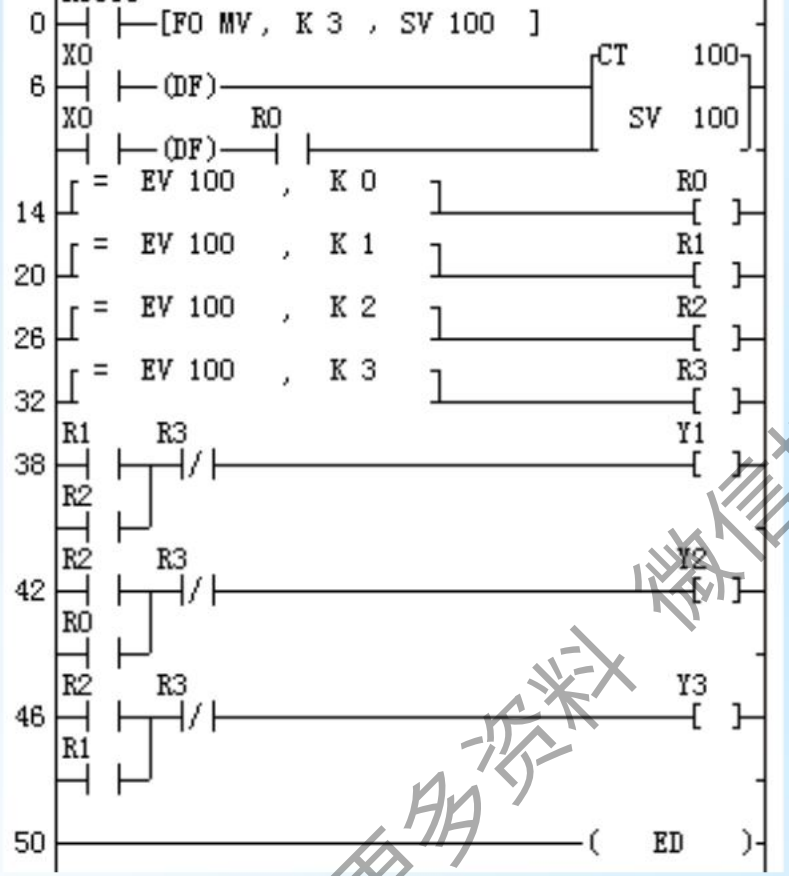
再按一下，2 号灯亮，1，3 号灭；

再按一下全部灭。

此控制是利用按动开关次数来控制各个灯的亮、灭，故可以用计数器来实现计数式顺序控制。

房间灯控制 PLC 的 I/O 点分配表

PLC 点名称	连接的外部设备	功能说明
X0	按钮	开关命令
Y1	控制继电器 1	控制 1 号灯亮灭
Y2	控制继电器 2	控制 2 号灯亮灭
Y3	控制继电器 3	控制 3 号灯亮灭




```
0   ST     R 9013
1   FO   (MV)
      K       3
     SV     100
6   ST      X0
7   DF
8   ST      X0
9   DF
10  AN      R0
11  CT     100
     SV    100
14  ST =
     EV    100
      K      0
19  OT      R0
20  ST =
     EV    100
      K      1
25  OT      R1
26  ST =
     EV    100
      K      2
31  OT      R2
32  ST =
     EV    100
      K      3
37  OT      R3
38  ST      R1
39  OR      R2
40  AN/     R3
41  OT      Y1
42  ST      R2
43  OR      R0
44  AN/     R3
45  OT      Y2
46  ST      R2
47  OR      R1
48  AN/     R3
49  OT      Y3
50  ED
```

房间灯计数式顺序控制程序

这里使用 R9013 是程序初始化的需要。一进入程序，就把十进制数 3 赋给 SV100 。从这以后 R9013 就不起作用了。

在程序中使用微分指令是使 X0 具有非自锁按钮的作用。

初始状态： EV100=3， R3 通 → Y1 、 Y2 、 Y3 不通， 3 个灯全灭；

第一次接通 X0 ： EV100=2 ， R2 通 → Y1 、 Y2 、 Y3 全通， 3 个灯全亮；

第二次接通 X0 ： EV100=1 ， R1 通 → Y1 和 Y3 通， Y2 断，故 2 号灭， 1 号和 3 号灯亮；

第三次接通 X0 ： EV100=0 ， R0 通 → Y2 通， Y1 和 Y3 断，故 2 号亮， 1 号和 3 号灯灭。

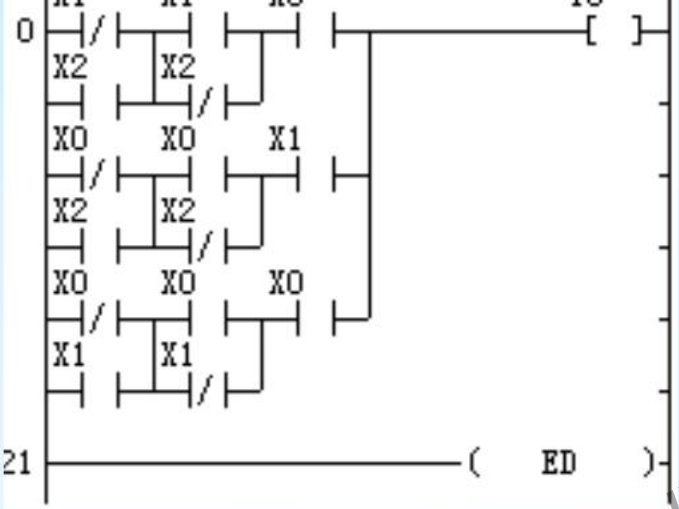
EV100=0 时，若再次闭合 X0 ，则计数器复位，灯全灭，程序从头开始重复以上过程。

四、多地点控制

要求：在三个不同的地方分别用三个开关控制一盏灯，任何一地的开关动作都可以使灯的状态发生改变，即不管开关是开还是关，只要有开关动作则灯的状态就发生改变。

三地控制一盏灯 I/O 分配表

PLC 点名称	连接的外部设备	功能说明
X0	A 地开关	在 A 地控制
X1	B 地开关	在 B 地控制
X2	C 地开关	在 C 地控制
Y0	灯	被控对象



0	ST/	X 1
1	OR	X 2
2	ST	X 1
3	OR/	X 2
4	ANS	
5	AN	X 0
6	ST/	X 0
7	OR	X 2
8	ST	X 0
9	OR/	X 2
10	ANS	

11	AN	X 1
12	ORS	
13	ST/	X 0
14	OR	X 1
15	ST	X 0
16	OR/	X 1
17	ANS	
18	AN	X 0
19	ORS	
20	OT	Y 0
21	ED	

三地控制一盏灯程序（1）

从这个程序中不难发现其编程规律，并能很容易地把它扩展到四地、五地甚至更多地点的控制。**但其设计方法完全靠设计者的经验，初学者不易掌握。**

者有章可循。

规定：输入量为逻辑变量，输出量为逻辑函数；常开触点为原变量，常闭触点为反变量。这样就可以把继电控制的逻辑关系变成数字逻辑关系。

三地控制一盏灯逻辑函数真值表

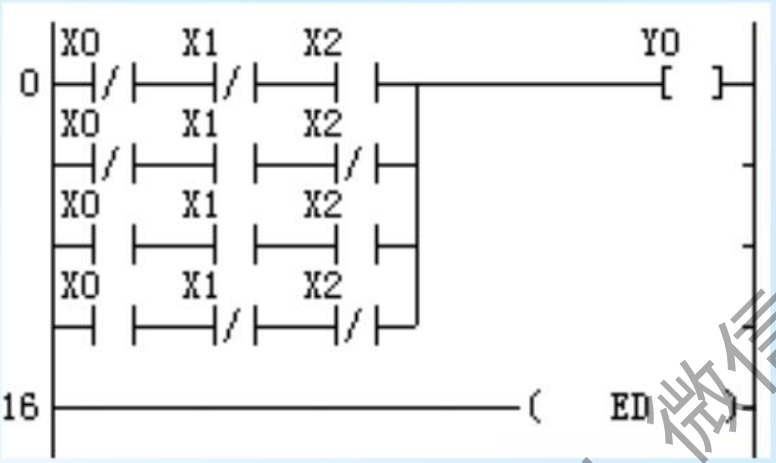
X0	X1	X2	Y0
0	0	0	0
0	0	1	1
0	1	1	0
0	1	0	1
1	1	0	0
1	1	1	1
1	0	1	0
1	0	0	1

真值表按照每**相邻两行只允许一个输入变量变化的规则排列。**即三个开关中的任意一个开关状态的变化，都会引起输出 Y0 由“1”变到“0”，或由“0”变到“1”。

由真值表写出输出与输入之间的逻辑函数关系式：

$$Y0=\overline{X0}\cdot X1\cdot X2+X0\cdot\overline{X1}\cdot X2+X0\cdot X1\cdot\overline{X2}+X0\cdot\overline{X1}\cdot\overline{X2}$$

可设计出梯形图程序如下图所示：

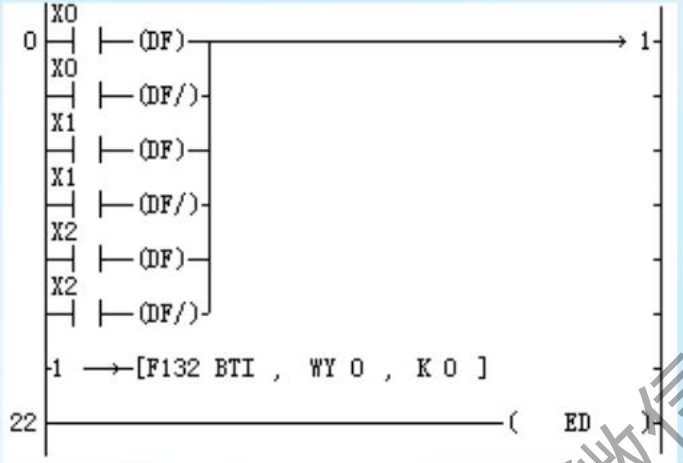



```
0   ST/    X0
1   AN/    X1
2   AN     X2
3   ST/    X0
4   AN     X1
5   AN/    X2
6   ORS
7   ST     X0
8   AN     X1
9   AN     X2
```

```
10  ORS
11  ST     X0
12  AN/    X1
13  AN/    X2
14  ORS
15  OT     Y0
16  ED
```

三地控制一盏灯程序（2）

使用高级指令可使程序更加简单。下图为应用高级指令 F132 编写的控制程序。



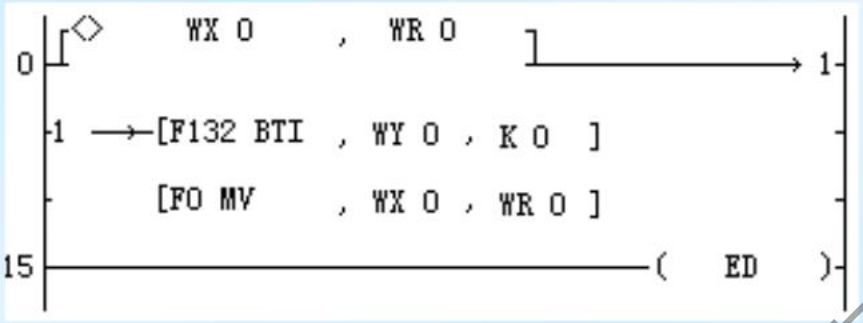

```
 0   ST     X 0
 1   DF
 2   ST     X 0
 3   DF/
 4   ORS
 5   ST     X 1
 6   DF
 7   ORS
 8   ST     X 1
 9   DF/
10   ORS
```

```
11   ST     X 2
12   DF
13   ORS
14   ST     X 2
15   DF/
16   ORS
17   F 132 (BTI)
       WY   0
       K    0
22   ED
```

三地控制一盏灯程序（3）

上面的程序只要开关动作（不管开关是接通还是断开），即将 Y0 求反。**程序中每一开关使用了两个微分指令，既可检测上升沿又可检测下降沿，**十分巧妙地实现了控制要求。

对于这种编程方式，无论多少个地方，只要在梯形图中多加几个输入触点和几条微分指令就可实现控制要求。




```
0  ST     ◇
       WX    0
       WR    0
5  F 132(BTI)
       WY    0
       K     0
10 F   0   (MV)
       WX    0
       WR    0
15 ED
```

三地控制一盏灯程序（4）

① 使用条件比较指令，只要（WXO）≠（WRO），就把 YO 求反。

② （WXO）→（WRO），使两个寄存器中内容完全一样。

③ **只要 WXO 中的内容改变，YO 的状态就立即变化。**

使用了字比较指令，故 WXO 中的 16 位都可以用来作为控制开关，使程序大大简化。

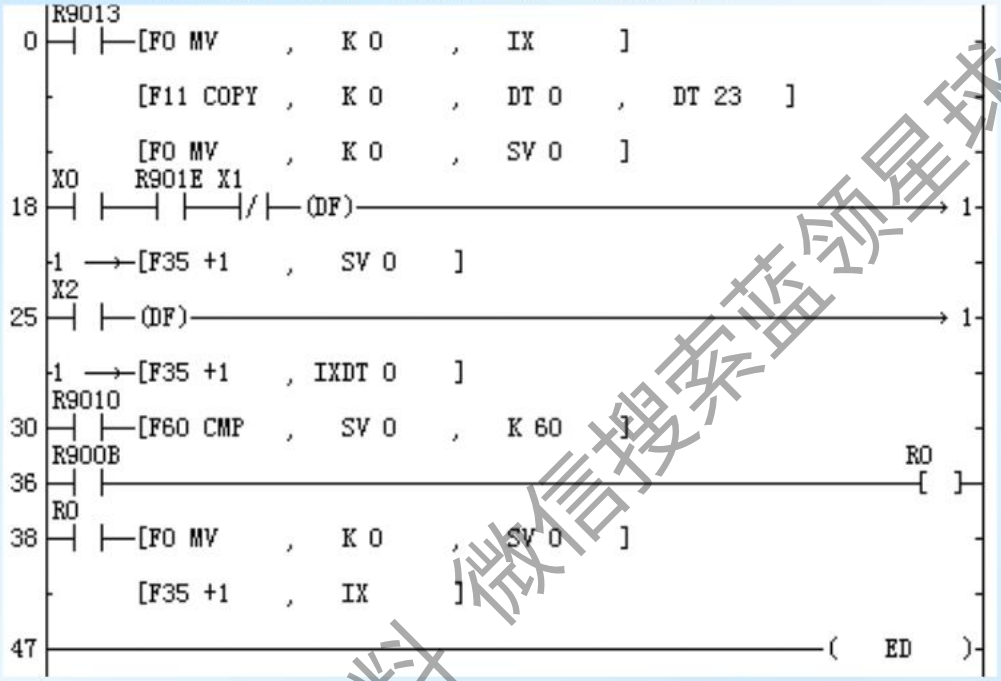
五、易拉罐自动生产线计数控制

在易拉罐自动生产线上，常常需要统计出每小时生产的易拉罐数量。罐装好的易拉罐饮料一个接一个不断地经过计数装置。假设计数装置上有一感应传感器，每当一听饮料经过时，就会产生一个脉冲。

要求：编制程序将一天24小时中每小时生产的数量统计出来。

易拉罐计数控制PLC的I/O点分配表

PLC点名称	连接的外部设备	功能说明
X0	蓝按钮（自锁）	启动命令
X1	红按钮（自锁）	停止命令
X2	传感器开关	计数脉冲



R9013： 对程序初始化。

DT0 ～ DT23：存放一天 24 小时每小时生产罐的数量；

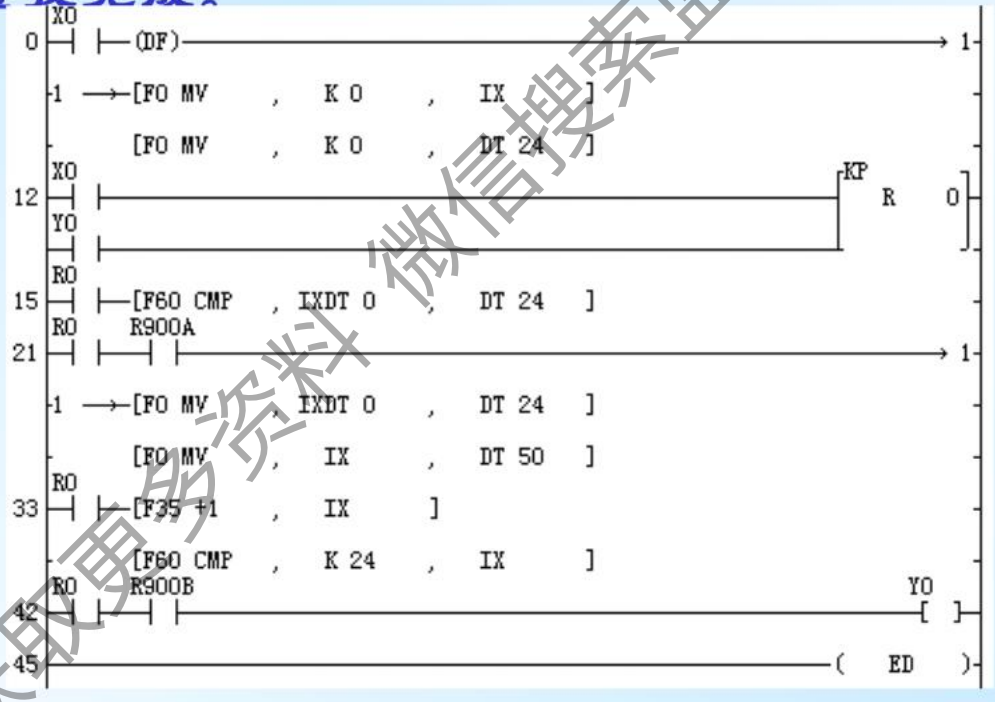
SV0： 记录每小时内的时间。

IX 作为地址修正值，当 F35 指令的操作数地址发生移动时，移动量为 IX 中的值。

如：当 IX=0 时， F35 指令将 DT0 的内容加 1；当 IX=10 时，则将 DT10 的内容加 1 。

上例中，一天 24 小时内每小时生产的易拉罐数已分别存储在数据寄存器 DT0 ～ DT23 中。编程找出其中最大的数，存入 DT24 中，并将最大数所在寄存器的编号存入 DT50 中。

要求：**X0 的上升沿开始查找，找到后，输出 Y0 表示查找完成。**



查找最大数据梯形图

查找数据中的最大数，只需将数据区中的数据进行两两比较即可。

索引寄存器 IX：用作地址修正；

R0：用来表示查找状态。

未查找完时，R0 一直接通，当查找结束时，R0 断开。

第五章 FP1的特殊功能及高级模块

第一节 FP1的特殊功能

一、脉冲输出

FP1 的输出端 Y7 可输出一路脉冲信号，最大频率范围为 45Hz ~ 5kHz 。这一功能只有晶体管输出方式的 PLC 才具有，且需配合脉冲输出控制指令 F164(SPD0) 使用。

图 5-1　脉冲输出进行位置控制示意图

二、高速计数功能（HSC）

在 FP1 内部有高速计数器，可同时输入两路脉冲。

- 最高计数频率： 10kHz；
- 计数范围： K-8388608～K8388607；
- 输入模式：加计数、减计数、可逆计数、两相输入；

此外，每种模式又分为有复位输入和无复位输入两种情况，输入计数不受扫描周期影响，处理过程中响应时间不延时。

1. 占用的输入端子

HSC 需占用 FP1 输入端子 X0、X1 和 X2。其中 X0 和 X1 作为脉冲输入端，X2 作为复位端，可由外部复位开关通过 X2 使 HSC 复位。

2. 输入模式及设置

HSC 的四种输入模式中，前三种为单相输入，最后一种为两相输入。如图 5-2 所示。

1) 加计数模式

2) 减计数模式

3) 加 / 减计数模式

4) 两相输入方式

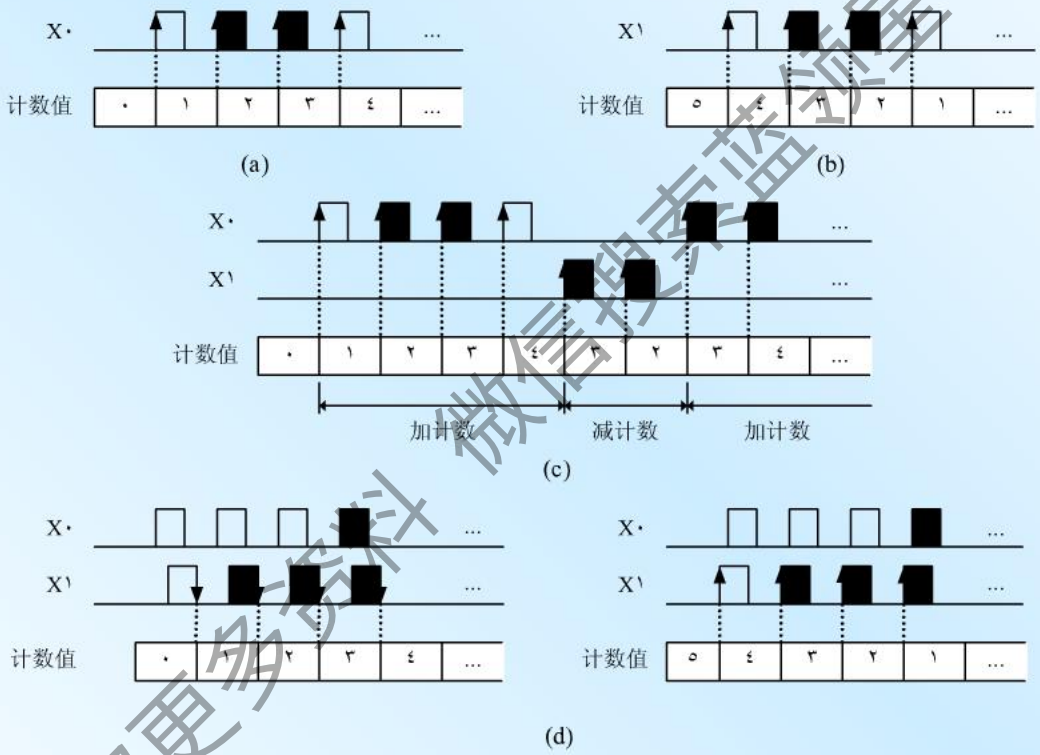


图 5-2　四种计数模式的脉冲波形示意图

3．与 HSC 相关的寄存器

表 5-1 系统寄存器 No.400 控制字说明

设定值	功能			输入模式
	X0	X1	X2	
H1	双相输入		—	双相输入方式
H2	双相输入		复位	
H3	加计数	—	—	加计数方式
H4	加计数	—	复位	
H5	—	减计数	—	减计数方式
H6	—	减计数	复位	
H7	加计数	减计数	—	加／减计数方式
H8	加计数	减计数	复位	
H0	HSC 功能未用			不工作（默认模式）

4．高速计数功能指令

1） 高速计数器的控制指令

[F0　MV,　S,　DT9052]：高速计数器控制指令

。

该指令功能是将S中的控制字数据写入DT9052中

，DT9052的低8

与其它高速计数指令有关的位
0：继续执行F162、F163、164、F165指令
1：清除F162、F163、164、F165指令

选择“复位输入端” X2的可用性控制位
0：复位输入端X2使能
1：复位输入端X2禁止

计数器输入控制位
0：接受计数输入
1：计数输入无效

软件复位控制位
0：不执行软件复位
1：高速计数器的经过值复位

2） 高速计数器经过值的读写指令

[F1 DMV, S, DT9044]：存储高速计数器经过值。将（S+1，S）中高速计数器的经过值写入DT9045、DT9044中。

[F1 DMV, DT9044, D]：调出高速计数器经过值。是将DT9045、DT9044中的经过值读出拷贝到（D+1，D）中。

3） 高速计数器输出置位复位指令

[F162 HCOS, S, Yn]：高速计数器的输出置位指令。

[F163 HCOR, S, Yn]：高速计数器的输出复位指令。

4） 速度和位置控制指令

[F164 SPDO, S]：速度及位置控制。该指令配合高速计数器和 Y7 的脉冲输出可以实现速度和位置控制。

a） 脉冲工作方式

b） 波形工作方式

5） 凸轮控制指令

[F165 CAMO, S]：凸轮控制。当高速计数器的经过值和参数表中设定的目标值相一致时，接通或断开参数表中指定的输出继电器。

三、可调输入延时滤波

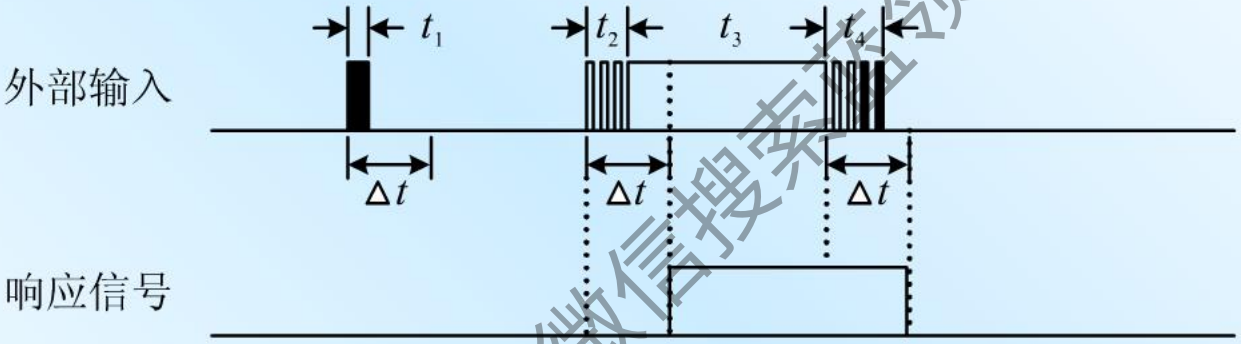


图 5-4　输入信号延时滤波示意图

图中，*t*1 为干扰脉冲，小于延时时间 Δ*t*，因此不响应；*t*2、*t*4 分别为机械开关接通和断开时的抖动时间，由图可见，经过延时，避开了输入信号的抖动部分，直接在稳定导通区间 *t*3 进行输入状态的采集和响应。

FP1 的延迟时间可以根据需要，在 1 ~ 128ms 之间进行调节。延时时间的设定是通过软件，在对应的系统寄存器中设置时间常数来实现，时间常数和延时时间的对应关系如下表：

表 5-8 时间常数与对应延时时间关系

时间常数 (BCD 码)	0	1	2	3	4	5	6	7
延时时间 (ms)	1	2	4	8	16	32	64	128

系统寄存器 No. 404 ~ 407 用于预先存放设置的时间常数，与输入端的对应关系为：

No.404： 设定 X0 ~ X1F 的时间常数。
No.405： 设定 X20 ~ X3F 的时间常数。
No.406： 设定 X40 ~ X5F 的时间常数。
No.407： 设定 X60 ~ X6F 的时间常数。

四、输入窄脉冲捕捉

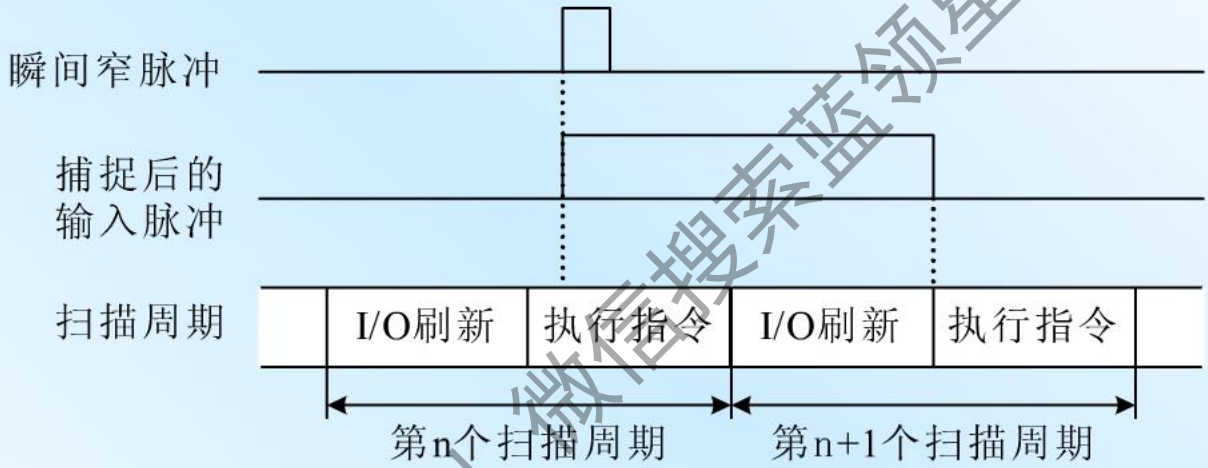
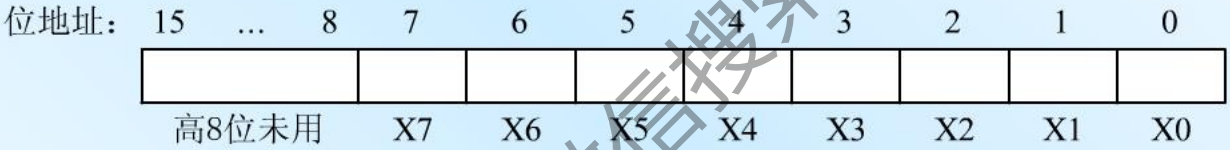


图 5-5　脉冲捕捉示意图

一个窄脉冲在第 n 个扫描周期的 I/O 刷新后到来，若无捕捉功能，此脉冲将会被漏掉；有了捕捉功能， PLC 内部电路将此脉冲一直延时到下一个 (第 n+1 个) 扫描周期的 I/O 刷新结束，这样 PLC 就能响应此脉冲。

只有输入端 X0 ~ X7 共 8 个输入端可以设成具有脉冲捕捉功能的输入端，这可以通过对系统寄存器 No. 402 的设置来实现。输入端子与系统寄存器 No. 402 的位对应关系如下所示：



输入端 X0 ~ X7 分别与 No. 402 的低 8 位对应，当某位设置为 1 时，则该位对应的输入端就具有脉冲捕捉功能；设置为 0 时，对应的输入端仍是普通的输入端。

五、特殊功能占用输入端优先权排队

大多数特殊功能均需占用 PLC 的 I/O 点，当多种功能同时使用时，对 I/O 的占用须按一定顺序进行优先权排队。

FP1 特殊功能优先权排队从高到低依次为：

高速计数器→脉冲捕捉→中断→输入延时滤波

六、其他功能

FP1 还有一些其它的特殊控制功能，如强制置位 / 复位控制功能、口令保护功能、固定扫描时间设定功能和时钟日历控制功能等

第五章 FP1的特殊功能及高级模块

第二节 FP1的高级模块

一、A/D 转换模块

1. 占用通道及编程方法

A/D 转换单元 4 个模拟输入通道占用输入端子分别为：

CH0： WX9(X90 ~ X9F)　　CH1： WX10(X100 ~ X10F)

CH2： WX11(X110 ~ X11F)　CH3： WX12(X120 ~ X12F)

PLC 每个扫描周期对各通道采样一次，并进行模数转换，转换的结果分别存放在输入通道 (WX9 ~ WX12) 中。

A/D 转换的编程可用指令 F0 实现，如 [F0 MV, WX9, DT0] 。执行这一指令后，CH0 输入的模拟信号经 A/D 转换变成数字信号后送入 WX9，并由 F0 指令读出保存到 DT0 中。其它通道也可仿照此格式进行编程。

注意：FP1 对 A/D 模块读取数据，每个扫描周期只进行一次。

2. A/D 的技术参数

项目	说明	
模拟输入点数	4 通道 / 单元 (CH0 ~ CH3)	
模拟输入范围	电压	0 ~ 5V 和 0 ~ 10V
	电流	0 ~ 20mA
分辨率	1/1000	
总精度	满量程的 ±1%	
响应时间	2.5ms/ 通道	
输入阻抗	电压	不小于 1 兆欧 (0 ~ 5V 和 0 ~ 10V 范围内)
	电流	250 欧姆 (0 ~ 20mA)
绝对输入范围	电压	+7.5V(0 ~ 5V) 、 +15V(0 ~ 10V)
	电流	+30mA(0 ~ 20mA)
数字输出范围	K0 ~ K1000(H0 ~ H03E8)	
绝缘方式	光耦合：端子与内部电路之间	
	无绝缘：通道间	
连接方式	端子板 (M3.5 螺丝)	

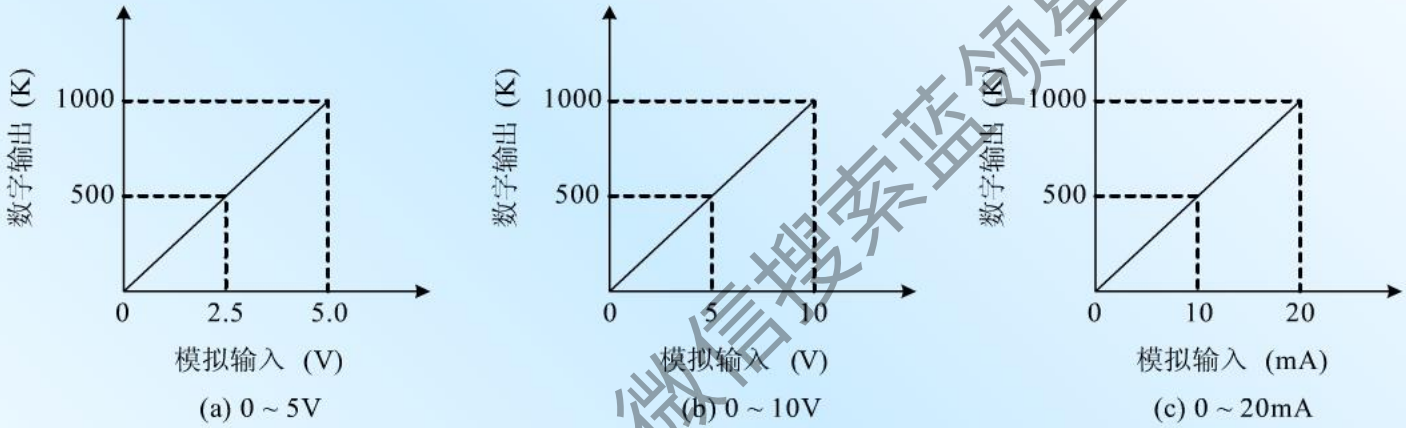


图 5-6 A/D 转换单元的输入输出特性

3. A/D 转换单元的面板布置及接线方法

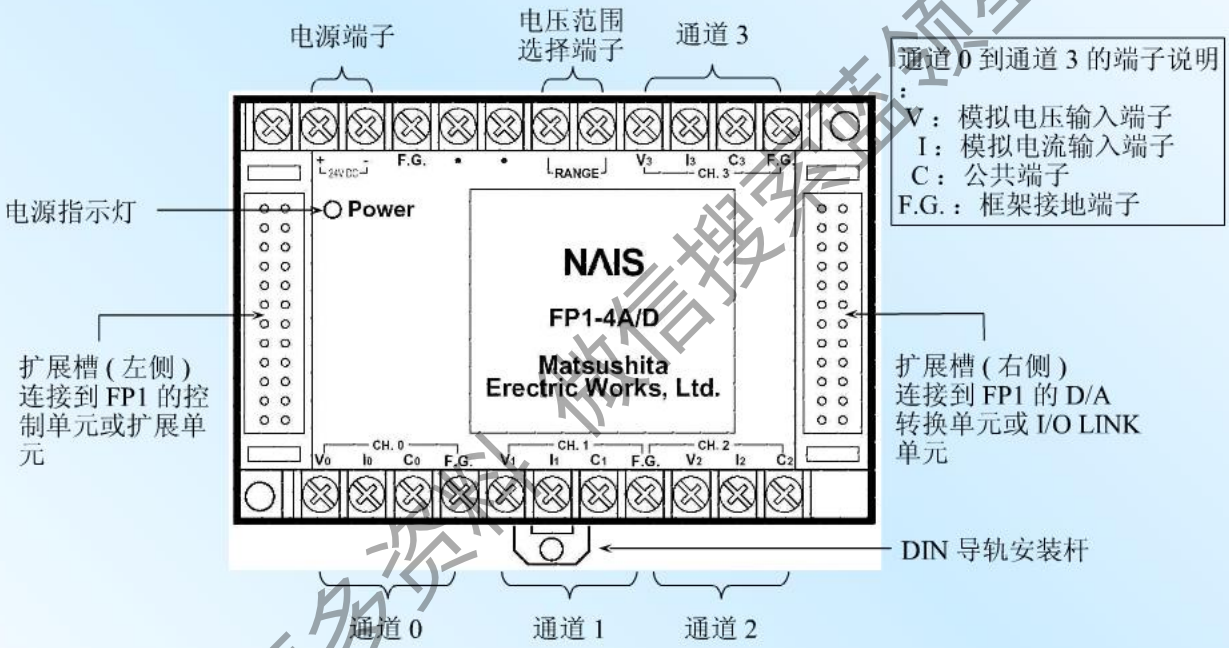


图 5-7　A/D 单元的面板布置图

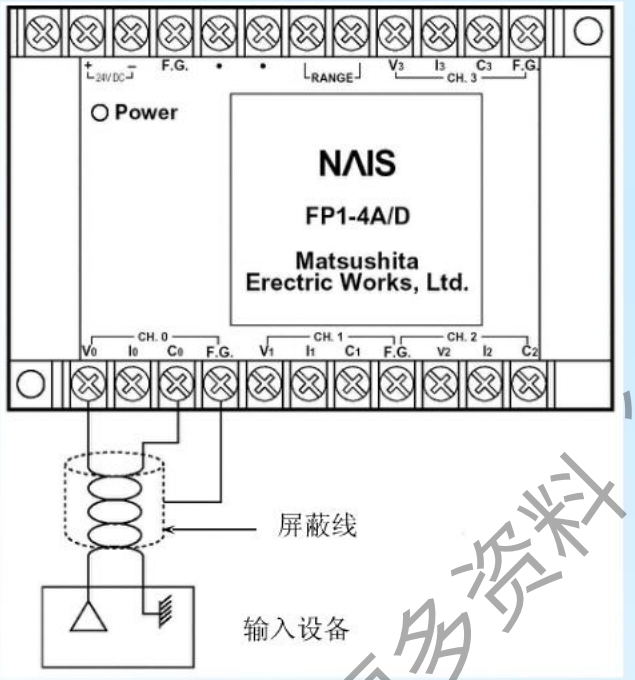


图 5-8　电压输入接线方式

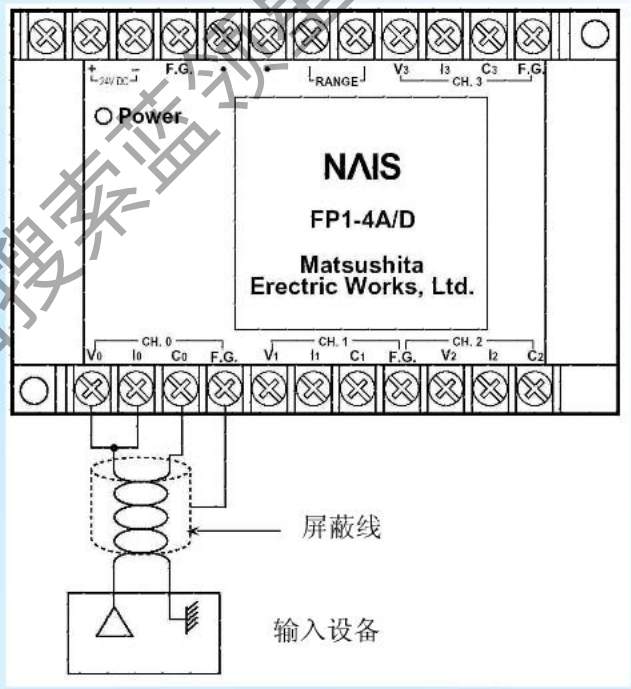


图 5-9　电流输入接线方式

4. 应用举例

当需对某信号进行监测，要求超限报警。这时可将该信号输入到 A/D，并用段比较指令将输入信号与上、下限进行比较。程序如图 5-10 所示。

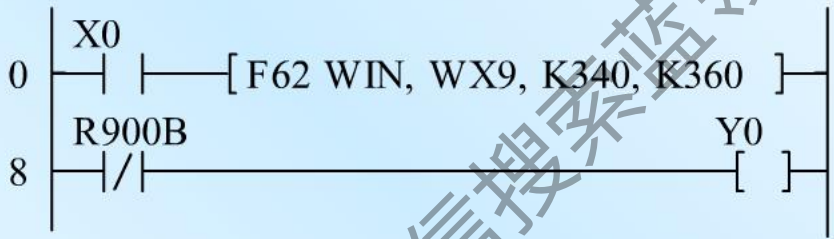


图 5-10　A/D 模块信号监控举例

若将 A/D 模块输入范围选在 0 ~ 10V(即将 RANGE 短接)，并将需监测的信号输入 CH0，则执行该程序后可实现下面功能：设输入信号上限为 3.6V，即对应 A/D 内部十进制数为 K360；输入信号下限为 3.4V，对应 A/D 内部十进制数为 K340。当输入信号在 3.4V ~ 3.6V 之间时则 R900B 常闭触点断开，故 Y0→OFF，报警灯不亮。若信号超出此范围则 R900B 常闭触点接通，故 Y0→ON，报警灯亮，从而实现对信号的监测。

二、D/A转换模块占用通道及编程方法

FP1 可扩展两个 D/A 模块，可用开关设定其单元号，即 No.0 和 No.1 ；每个 D/A 模块有两个输出通道，即 CH0 和 CH1 。

当开关置于左边时，该模块设为 No.0 ，其 I/O 通道分配如下：

CH0 ： WY9(Y90 ~ Y9F)　　CH1 ： WY10(Y100 ~ Y10F)

当开关置于右边时，该模块设为 No.1 ，其 I/O 通道分配如下：

CH0 ： WY11(Y110 ~ Y11F)　CH1 ： WY12(Y120 ~ Y12F)

D/A 转换的编程也可用指令 F0 实现，如 [F0 MV, DT0, WY9] 。执行这一指令后，将 DT0 的内容经 WY9 送往 D/A 转换器，并将转换好的模拟信号经 No.0 的 CH0 通道输出。其它通道也可仿照此格式进行编程。

注意：FP1 对 D/A 模块写入数据，每个扫描周期只进行一次。

2. D/A 的技术参数

项目	说明	
模拟输出点数	2 通道 / 单元 (CH0 ~ CH1)	
模拟输出范围	电压	0 ~ 5V 和 0 ~ 10V
	电流	0 ~ 20mA
分辨率	1/1000	
总精度	满量程的 ±1%	
响应时间	2.5ms/ 通道	
输出阻抗	不大于 0.5 欧姆 (电压输出端)	
最大输出电流	20mA(电压输出端)	
允许负载电阻	0 ~ 500 欧姆 (电流输出端)	
数字输出范围	K0 ~ K1000(H0 ~ H03E8)	
绝缘方式	光耦合：端子与内部电路之间	
	无绝缘：通道间	
连接方式	端子板 (M3.5 螺丝)	

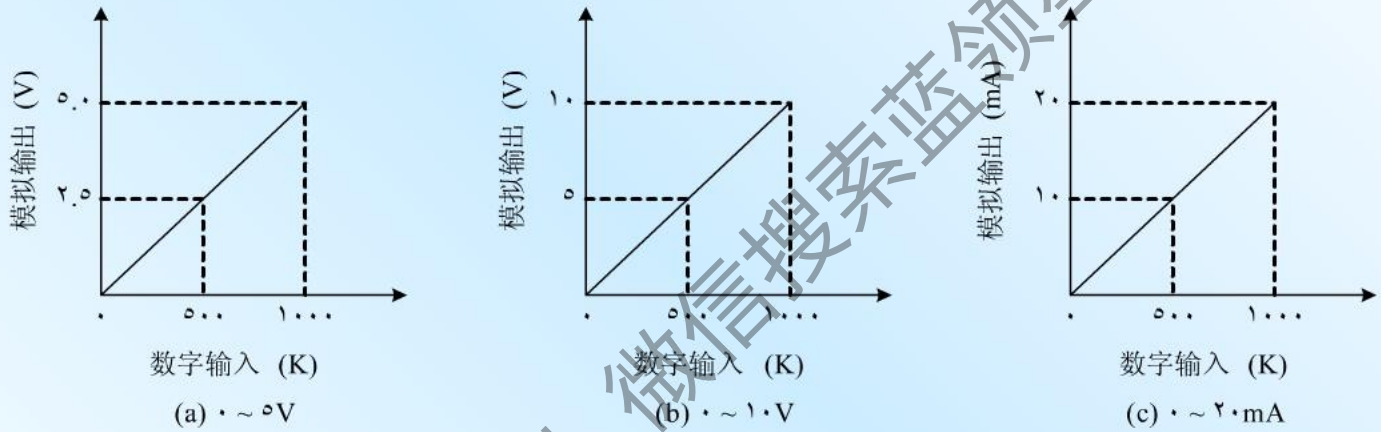


图 5-11 D/A 转换单元的输入输出特性

3. D/A 转换单元的面板布置及接线方法

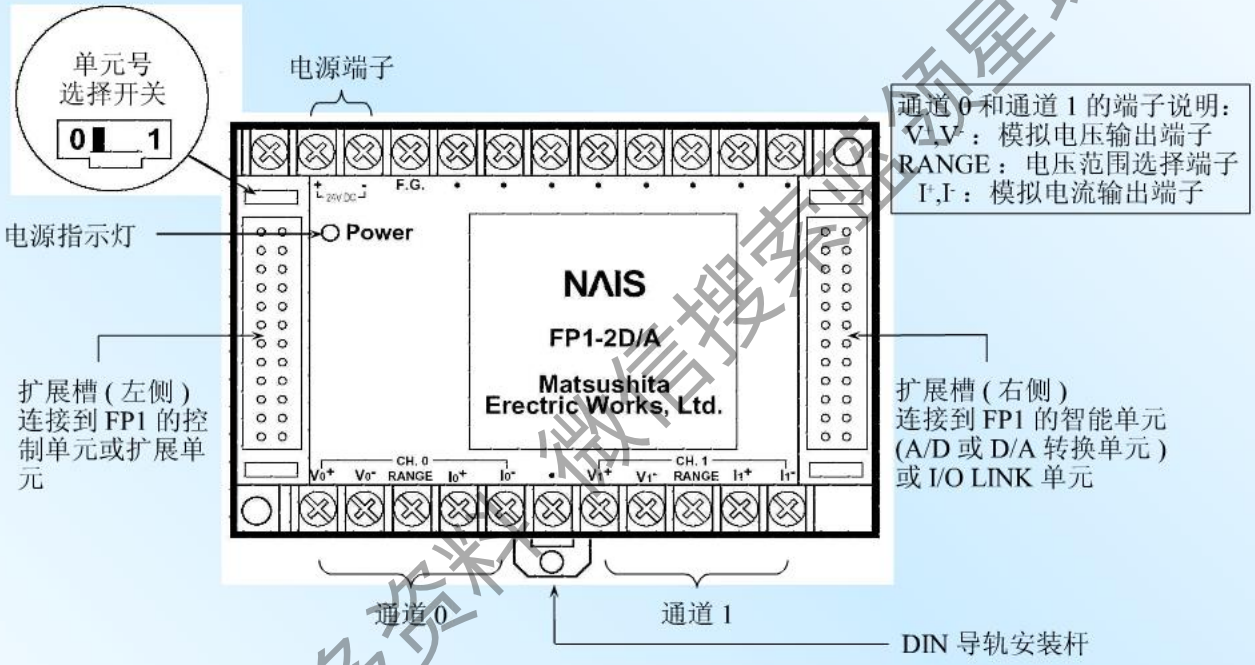


图 5-12　D/A 转换单元的面板布置图

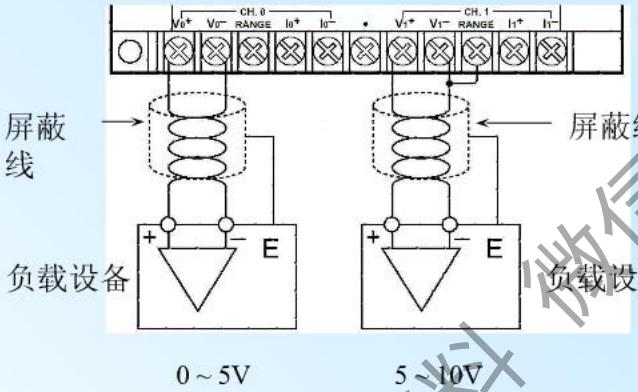


图 5-13　电压输出接线方式

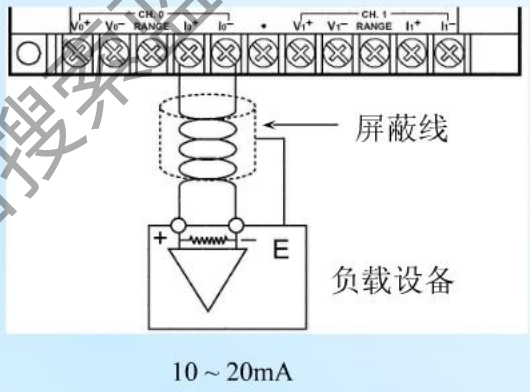


图 5-14　电流输出接线方式

4. 应用举例

三个模拟量信号分别从 A/D 模块的 CH0 ~ CH2 输入，求平均值，再由 D/A 模块 No.1 的 CH1 通道输出。梯形图如图 5-15 所示。

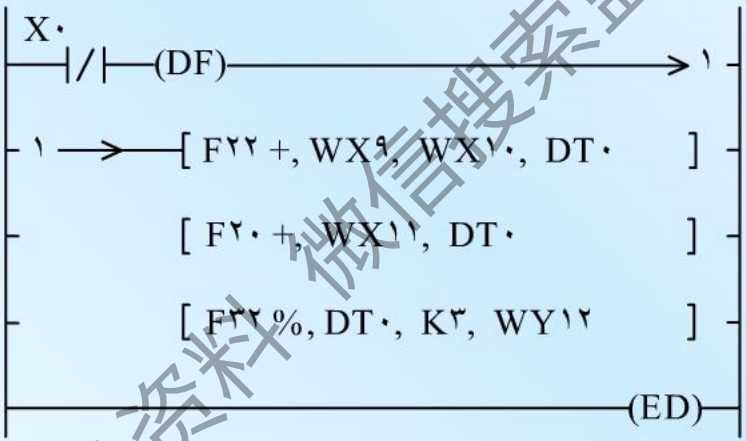


图 5-15　A/D 和 D/A 模块应用举例

第五章 FP1的特殊功能及高级模块

第三节 FP1的通讯功能

集散式控制系统的关键技术之一是系统的通信和互联。松下电工提供了6种（C-NET， F-Link，P-Link， H-Link， W-Link， FP以太网）功能强大的网络形式，同时提供了若干种与相应的网络连接方式有关的通信链接单元，适合于各种工业自动化网络的不同需要。

一、通讯的有关基本概念

1. 并行通信与串行通信
2. 同步通信与异步通信
3. 波特率
4. 单工、双工通信方式
   1) 单工通信
   2) 半双工通信 (Half Duplex)
   3) 全双工通信 (Full Duplex)

二、FP1 的通讯接口

FP1 系列 PLC 进行数据交换时常采用 RS232C 、 RS422 、 RS485 三种串行通信接口，相关的链接单元也有三种，均为串行通信方式。

- I/O LINK 单元是用于 FP1 和 FP3/FP5 等大中型 PLC 之间进行 I/O 信息交换的接口 (1 个 RS485 接口和 2 个扩展插座)；
- C-NET 适配器是 RS485 – RS422/RS232C 信号转换器 (1 个 RS485 、 1 个 RS422 、 1 个 RS232C 接口)，用于 PLC 与计算机之间的数据通讯；
- S1 型 C-NET 适配器是 RS485 – RS422 信号转换器 (1 个 RS485 、 1 个 RS422 接口)，用于 C-NET 适配器和 FP1 控制单元之间的通讯。

1．RS232C 通信接口

RS232C 所采用的电路是单端接收电路，数据传输速率最高为 20kbps，电缆最长为 15m。

2．RS422 通信接口

RS422 标准规定的电气接口是差分平衡式的，能在较长的距离内明显地提高传输速率，例如，1200m 的距离，速率可以达到 100kbps，而在 12m 等较短的距离内则可提高到 10Mbps。

3．RS485 通信接口

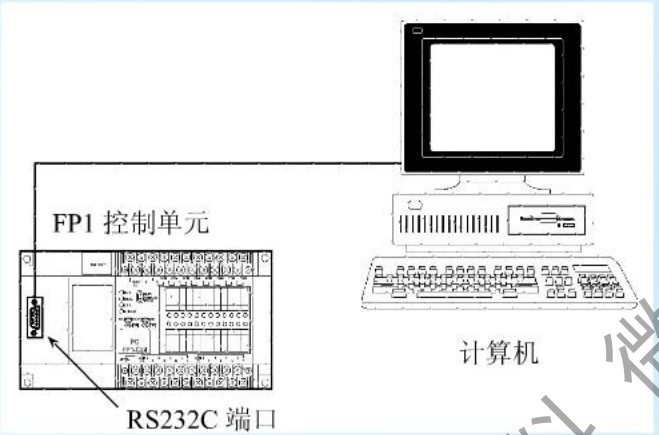
RS485 实际上是 RS422 的简化变型，RS485 分时使用一对信号线发送或接收。可以高速

三、通讯方式

1． FP1 与计算机 (PC) 之间的通讯

一般地，一台计算机与一台 FP1 之间的通讯称 1:1 方式，一台计算机与多台 FP1 之间的通讯称 1:N 方式。

有两种方法可以实现一台计算机与一台 FP1 之间的通讯。一种方法是直接通过 FP1 的 RS232 口与 PC 进行串行通讯。另一种方法可经 RS232/RS422 适配器用编程电缆同 PC 进行通讯。



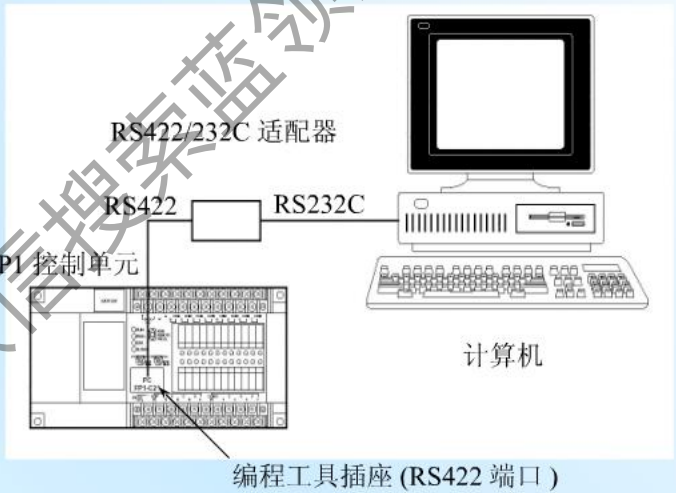


图 5-17　直接通过 RS232 口进行串行通讯　　图 5-18　　通过适配器进行通讯

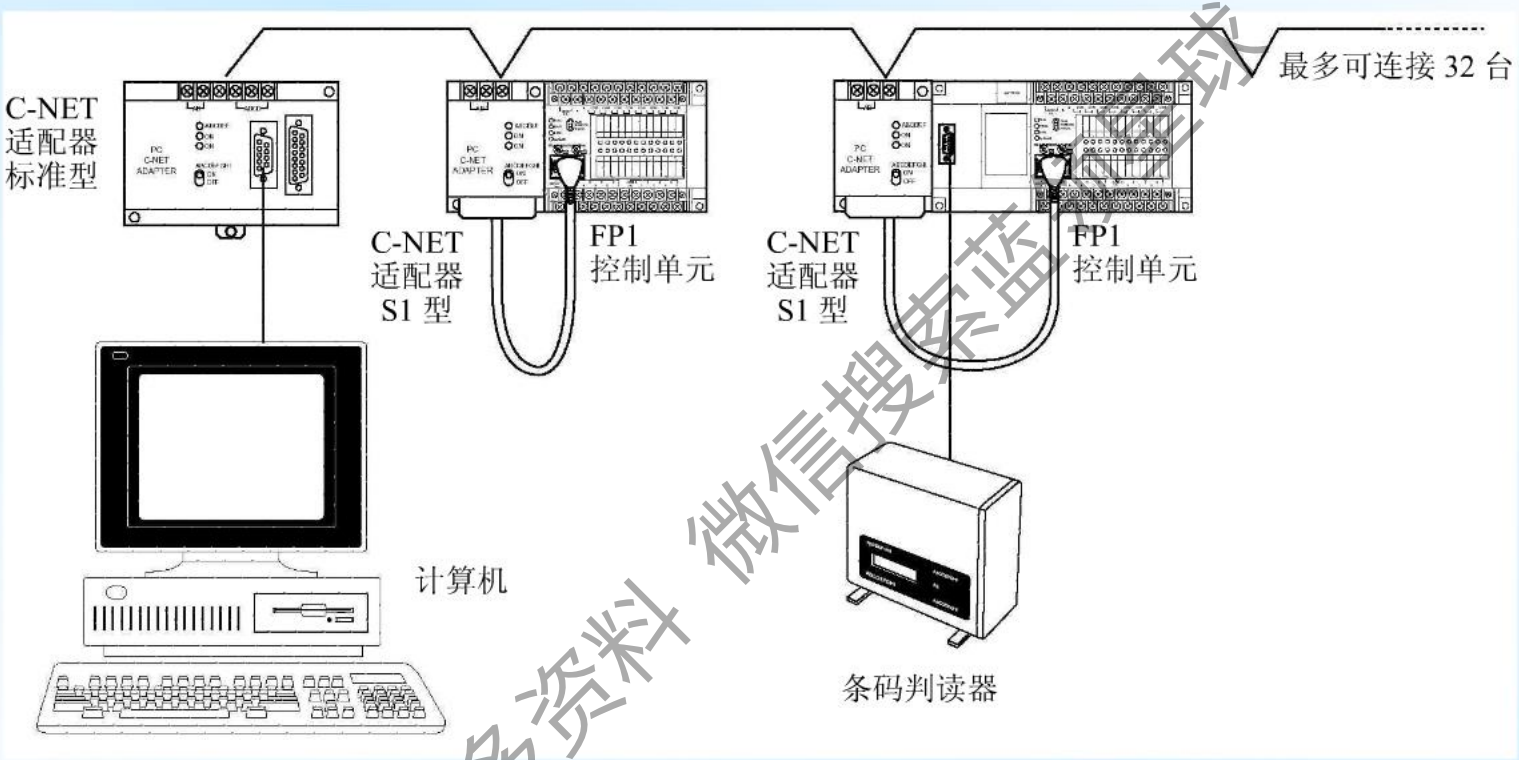


图 5-19　1:N 通讯方式

2． FP1 与 FP3/5 的通讯

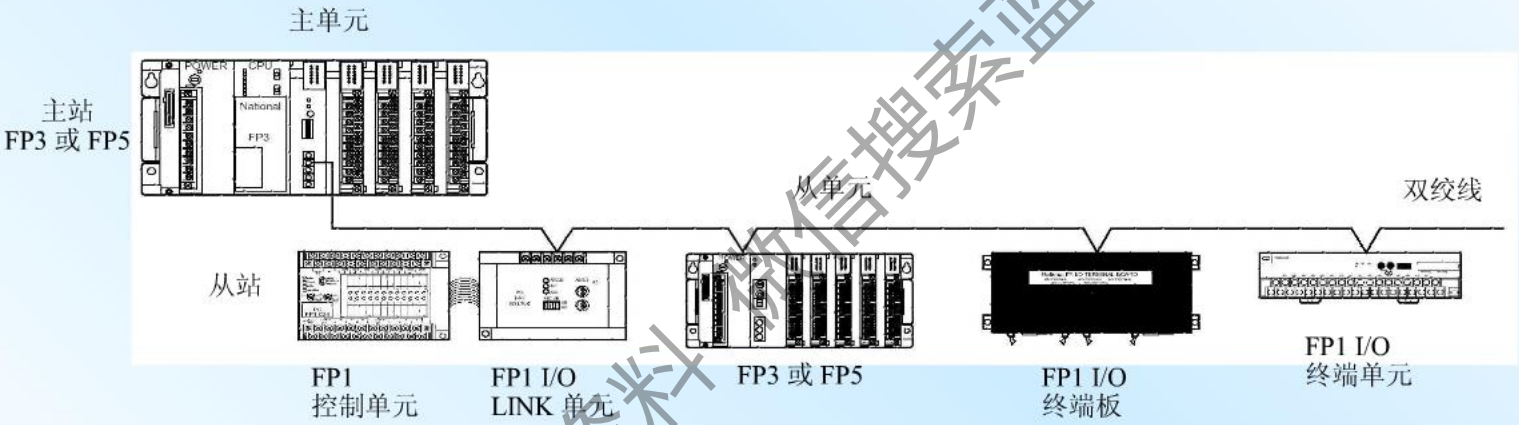


图 5-20　FP1 与 FP3/5 的通讯

3． FP1 和外围设备之间的通信

FP1 的相关外围设备有：智能终端 I.O.P.，条形码判读器、EPROM 写入器和打印机等。这些外围设备均设有 RS232 串行通信口，可以方便地实现与 FP1 的通信。

四、专用通信协议 MEWTOCOL

通信协议是通信双方就如何交换信息所建立的一些规定和过程。它是 FP 系列 PLC 网络设计的基础。

FP1 采用松下电工公司专用通信协议 – MEWTOCOL 。该协议共分为两个部分：

- **MEWTOCOL-COM**：计算机与 PLC 之间的命令通信协议；
- **MEWTOCOL-DATA**：PLC 与 PLC 之间及 PLC 与计算机之间的数据传输协议。

MEWTOCOL-DATA 协议用于分散型工业局域网 H-LINK 、 P-LINK 、 W-LINK 及 ETLAN 中 PLC 与 PLC 之间及 PLC 与计算机间的数据传输。

第五章结束

哈尔滨理工大学
电气与电子工程学院
电气技术教研室
2003 年 3 月 24 日

第六章 松下电工 PLC 编程工具及三维力控监控组态软件简介

第一节 松下电工 PLC 编程工具简介

FP1 系列 PLC 的编程手段有两种：

3. 利用相应配套编程软件在个人计算机上进行。

4. 使用 FP 手持编程器。

a. 松下电工 PLC 编程软件

1. 概述

日本松下电工公司开发的 PLC 编程软件有三种：

- DOS 环境下使用的 NPST-GR
- Windows 环境下 FPSOFT
- Windows 环境下 FPWIN-GR

●NPST-GR 采用的是典型的 DOS 界面。具有中、英文两种版本。因汉化后的 CNPST-GR 开发的比较早，对近几年生产的 FPO 、 FP2 等系列 PLC 不支持。

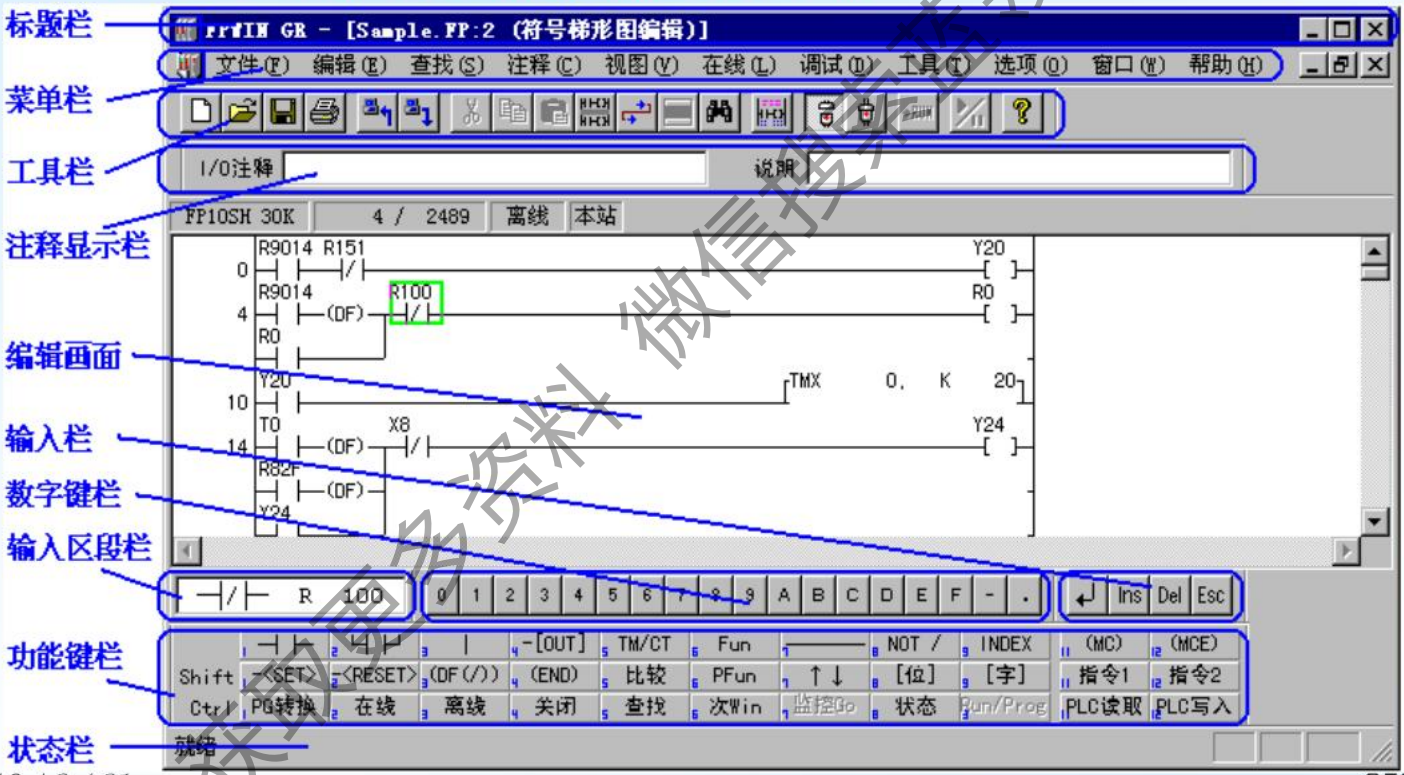
●FPWIN-GR 软件采用的是典型的 Windows 界面。具有中、英文两种版本。由于 FPWIN-GR 是新近开发出来的软件，其各项功能更趋合理、使用更加方便。

●FPSOFT 软件是早期开发的，它的出现开创了 Windows 环境的 PLC 编程软件的先河。但由于它开发得较早，虽大部分功能与上述 FPWIN-GR 相似，但有些功能不如 FPWIN-GR 那样完善。

1. FPWIN-GR 软件（汉化 1.1 版本）

(1) 认识 FPWIN － GR

FPWIN- GR 界面各部分名称及分布：



显示当前正在输入的回路。通过单击输入栏中的[Enter]或按键盘中的[Enter]键确认输入内容。

●功能键栏

在编写程序时：

i.用鼠标点击“功能键栏”实现指令输入。

j.用功能键“F1”～“F12”与“SHIFT”的组合实现指令输入。

k.用功能键“F1”～“F12”与“CTRL”的组合实现指令输入。

各个按钮左下角的数字表示所对应的功能键号。第1段、第2段中分布的是主要指令的快捷键。第1段的操作**只需按功能键**即为有效。第2段的操作需**同时按 Shift + 功能键**有效。第3段中分布的是功能的快捷键。第3段的操

(a) 在功能键栏中输入 [F1]、[F2]、[F4]、[F8] 或 [SHIFT]+[F1]（[F2]、[F8]）时，将显示触点线圈的基本指令如下图：

X：输入外部输入

Y：输入外部输出

R：输入内部继电器

L：链接继电器

P：脉冲继电器

T：定时器触点

C：输入计数器触点

E：输入错误警告继电器

比较：输入数据比较指令

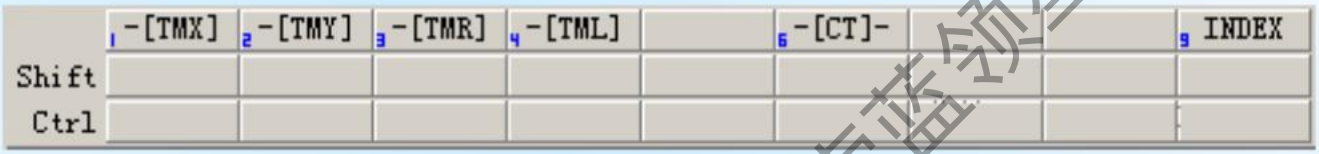
NOT/：将到光标位置为止的运算结果反转

INDEX：输入索引修饰

№ 清除：清除输入区段中的设备编号

↑ ↓：用于上升沿检出 / 下降沿检出的图形符号。能否使用本功能取决于所用 PLC 机型。

(b) 在功能键栏中输入 [F5] 时，将显示定时器 / 计数器指令 (TM/CT)。



TMX ： 输入 0.1 秒定时器

TMY ： 输入 1 秒定时器

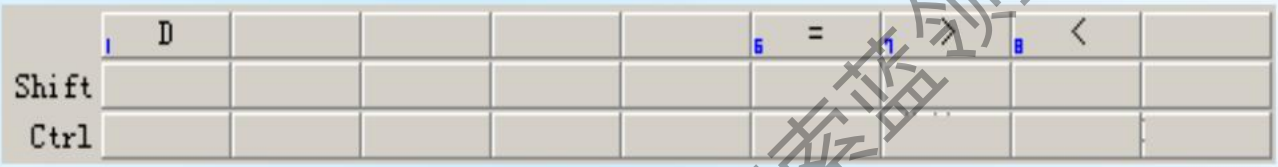
TMR ： 输入 0.01 秒定时器

TML：输入 0.001 秒定时器

CT ： 在输入区段中输入计数器

INDEX：在输入区段中输入索引修饰

(c) 在功能键栏中输入[SHIFT]+[F5]时，将显示比较指令（CMP）。



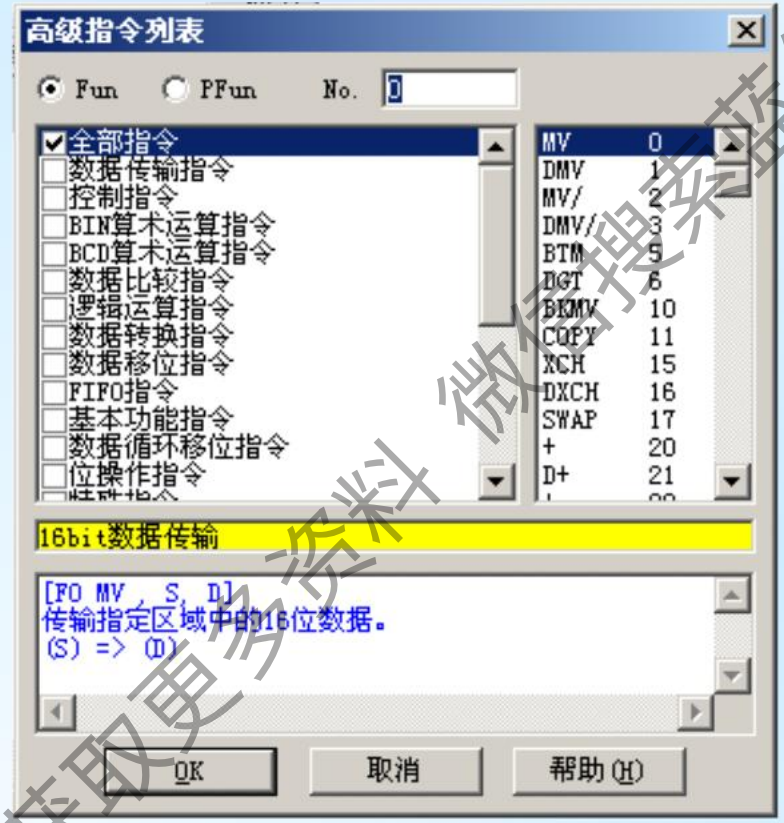
D： 在进行双字 (32 位) 比较指令时输入

=： 相等比较时输入

>： 大于比较时输入

<： 小于比较时输入

(d) 在功能键栏中输入[F6]、[SHIFT]+[F6]或[SHIFT]+[F9]时，将显示高级指令列表。



高级指令有以下两种类型：

FUN： **每次扫描执行型指令，** 按 [F6] 键。

PFUN：微分执行型指令，按 Shift + [F6] 键。

在高级指令列表中：

左侧选择指令的类型，右侧会显示出该类型中的相关指令。按指令的序号排列。下部显示相应指令的说明。 选择 [OK]，指令出现在编辑画面中。

一个高级指令出现在编辑画面后，需添加相应的指令参数，**将光标移到指令中的 [??????] 位置处进行添加。** 这时功能键栏变为下图。

	1	2	3	4	5	6	7		9	10
	WX	WY	WR	WL	DT	LD	FL		INDEX	No. 清除
Shift	SV	EV	K	H	M	f				
Ctrl										

WX：输入 WX（以字指定的外部输入）

WY：输入 WY（以字指定的外部输出）

WR：输入 WR（以字指定的内部继电器）

WL： WL（以字指定的链接继电器）

DT： 输入 DT（数据寄存器）

LD： 输入 LD（链接数据）

FL： 输入 FL（文件寄存器）

SV： 输入 SV（定时器·计数器的设定值）

EV： 输入 EV（定时器·计数器的目标值）

K： 在输入区段中输入 10 进制常数

允许输入范围：

16 位运算时： K-32,768 ～ K32,767

32 位运算时： K-2,147,483,648 ～ K2,147,483,647

H： **在输入区段中输入 16 进制常数**

允许输入范围：

16 位运算时： H0 ～ HFFFF

32 位运算时： H0 ～ HFFFFFFFF

M：输入字符串常数

f：输入实数常数

INDEX： 在输入区段中输入索引寄存器，或者在输入区段内输入设备索引修饰。

№ 清除： 清除输入区段中的设备编号

(e) 在功能键栏中输入 [F9] 时，将显示指定索引。

注意：

在符号梯形图编辑模式下编写了程序以后，为了确定由梯形图所编写的程序，必须进行“程序转换”处理。

在进行程序转换处理时，有以下几点限制：

1) 程序行数的限制

一次可转换的程序行数须在 33 行以内。

2) . 折回点的限制

在一个程序块中，所使用的折回点总数不能超过 32 个。

3) . OR 指令数的限制

连续输入的 OR 指令数量不能超过 33 个。

4)． OT 指令数的限制

连续输入的 OT 指令数量不能超过 33 个。

5)． PSHS 指令数的限制

可连续使用的 PSHS 指令的次数有一定限制。限制数量随 PLC 机型的不同而有所差别。

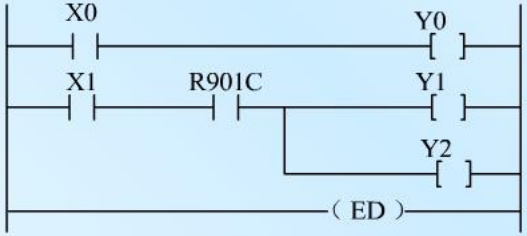
FP0 、 FP1 、 FP-M ： 最多 8 次

FP10 、 FP10S ： 最多 5 次

FP5 、 FP3 、 FP-C 、 FP2 、 FP2SH 、 FP10SH ： 最多 7 次

1. FPWIN-GR 应用举例

下面通过一个简单的例子说明如何利用 FPWIN-GR 输入程序。梯形图程序如图所示。




```
0    ST    X0
1    OT    Y0
2    ST    X1
3    AN    R901C
4    OT    Y1
5    OT    Y2
6    ED
```

启动 FPWIN-GR，选择“创建新程序”，在选择机型的对话框中选择“ FP1 C24,C40”，**此时在屏幕上显示的是“符号梯形图编辑”区**（若在标题栏显示的不是“符号梯形图编辑”，可以选择菜单“视图\符号梯形图编辑”），在屏幕的左上角显示一个**绿色的矩形光标**。

利用鼠标（也可用键盘）操作输入程序如下：

1) 输入 [F1]-[F1]-[0]-[Enter]，显示结点 X0。

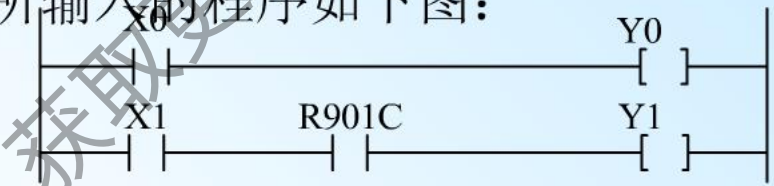
2) 输入 [F4]-[F2]-[0]-[Enter]，显示结点 Y0。

通过 1）、2）两步所输入的程序如下图所示：

3）输入 [F1]-[F1]-[1]-[Enter]，显示结点 X1。

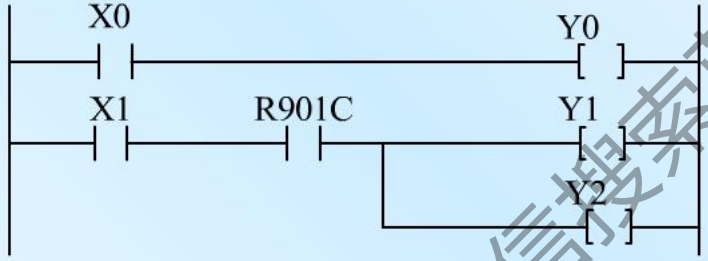
4）输入 [F1]-[F3]-[9]-[0]-[1]-[C]-[Enter]，显示结点 R901C。

5）输入 [F4]-[F2]-[1]-[Enter]，显示结点 Y1。通过以上几步所输入的程序如下图：



6）将光标放在结点 R901C 和 Y1 之间的下一行，输入 [F3]-[F4]-[F2]-[2]-[Enter] 显示结点 Y2 。

输入的程序如下图所示：



7）输入 [Shift]+[F4]-[Enter]，显示程序结束标志。此时完全输入例程。

8) [Ctrl]+[F1], 进行程序转换，然后保存文件即可。

选择菜单“视图 | 布而非梯形图编辑”，即可看到本程序的助记符程序如图 6-9 （ a) 所示。

a. FP 编程器Ⅱ

FP 编程器Ⅱ是一种手持编程工具。适用于 FP 系列的 PLC(FP1 、 FP3 、 FP5 、 FPI0S 、 FPI0 、 FP-C 和 FP-M 等) 。

手持编程器的功能如下：

① 程序编辑。

② FP 编程器Ⅱ具有“ OP”功能。用此功能，可监视或设置存储于 PLC 中的继电器通 / 断状态、寄存器内容以及系统寄存器参数等。

③ 程序双向传送到 FPWIN － GR 或 PLC 中等

1． FP 编程器 II 键盘功能介绍

FP 编程器 II 如下图所示。

(1) 插座

插座是 FP 编程器Ⅱ与 PLC 、 PC 机或调制解调器相连接的接口。当与 FP1 、 FP3 、 FP5 、FPI0S 或 FPI0 相连时，可用作为 RS422 接口；当与 FP-C 和 FP-M 连接时，可作为 RS232 接口。

(2) 液晶显示器 (LCD)

LCD 用于显示指令及信息。在显示窗口可同时**显示两行信息或数据**。若出现错误，**在显示窗口的上一行将显示出错信息。**

(3) 操作键

利用操作键，可通过 FP 编程器Ⅱ进行输入指令与设置系统寄存器值，以及监视继电器或寄存器等项操作。

FP 编程器Ⅱ上共有 35 个键。

继电器指令键

ST
X-WX

- 用于输入 ST（初始加载）指令。
- 用于输入 X 或 WX 触点。

AN
Y-WY

a. 用于输入 AN（与）指令。
b. 用于输入 Y 或 WY（输出继电器）。

OR
R-WR

① 用于输入 OR（或）指令。
② 用于输入 R 或 WR（内部继电器）。

OT
L-WL

① 用于输入 OT（输出）指令。
② 用于输入 L 或 WL（链接继电器）。

FN/P
FL

① FN 为“扫描执行方式”的高级指令；P 为“脉冲执行方式”的高级指令。每按一次该键，可交替输入 FN 和 P。
② 用于输入 FL（文件寄存器）。

NOT
DT/Ld

① 用于输入 NOT（非）指令。
② 用于输入 DT（数据寄存器）或 LD（链接数据寄存器）。

STK
IX/IY

① 当输入 ANS（组与）指令时，依次按 Y-WY 键和该键。

② 当输人 ORS（组或）指令时，依次按 OR R-WR 键和该键。

③ 用于输入IX或 IY（索引寄存器）。

TM
T-SV

① 用于输入 TM（定时器）指令。

② 用于输入 T（定时器触点）或 SV（定时器 / 计数器的预置值）。

CT
C-EV

① 用于输入 CT（计数器）指令。

② 用于输入 C（计数器触点）或 EV（定时器 / 计数器的经过值）。

字母数字键

= C	D D	< E	> F	8	9	A	B
4	5	6	7	0	1	2	3

a. 数字键用于输入数值和序号。

b. 当输入 ST、AN 或 OR 指令后，还可输入上挡键“ =”、“ >”、“ <”及“ D”，以组成字比较指令。

其它键

“帮助/清除”键

(HELP)
CLR

① 当显示指令时，用此键可清除 LCD 下面一行的指令名和操作数，而地址仍保留，以便输入新的指令。

② 当监视寄存器时，用此键可清除寄存器值，以便重新设置新的数值

③ 在初始状态下，若按 SHIFT SC 键后再按此键，可显示非键盘指令的代码表。

④ 在执行 OP 功能时，若按 SHIFT SC 键后再按此键，可列出 OP 功能表。

“全清”键

ACLR

① 清除当前显示的所有数据（清屏）。

② 若执行 OP 功能过程中按此键，将退出 OP 功能。

③ 按此键后，将显示出两个（* *）号，此状态称为“初始状态”。

“删除/插入”键

(DELT)
INST

① 在程序中插入刚输入的指令。

② 按 SHIFT SC 键后再按此键，可删除 LCD 中下面一行中的内容。

“数制转换/常数”键

(BIN) K/H ① 输入常数字符 K 或 H 时，每按一下此键，将交替显示 K 或 H。
② 按此键可以以十进制 (K) 或十六进制 (H) 显示寄存器值。依次按 SHIFT SC 键和此键，还可以以二进制显示寄存器值。

“指令切换”键

SHIFT SC a. 按此键进入 SC 方式，在 SC 方式下可输入一些键盘上没有的基本指令（非键盘指令），如 ED（结束）或 NOP（空操作）指令。再按一下此键，即可退出 SC 方式。
用此键激活一些键上用橙色表示的一些功能。例如先按此键，然后再按 (DELT) INST，可删除当前屏上的指令。

“操作/负号”键

(－) OP ① 用此键可进入 OP 功能，但需先按 ACLR 键后再按此键。
② 用此键可为常数或数值输入负号。

“查找/上箭头” 键

① 查找带有继电器、寄存器名的指令程序和地址。
② 按此键可使 LCD 上显示的指令按地址顺序向上滚动。

“读取/下箭头”键

① 从 PLC 中读取指令、继电器状态或寄存器值。
② 按此键可使 LCD 上显示的指令按地址顺序向下滚动。

“写入”键

为将指令、寄存器值或继电器状态写入 PLC，在输入指令或参数后，须按此键。

“输入”键

① 录入高级指令名和高级指令、CT、TM 指令的操作数。
② 录入所选择的 OP 功能。

说明：

(1) 有些键具有两种或多种功能，如 ST X-WX 键既可以输入 ST 指令，也可以输入继电器名称 X 、 WX 。而具体输入的内容将根据当时的操作而自动识别。当需输入指令码时，按此键输入的是 ST ；若是在输入基本指令后再按此键，则输入 X ；当输入高级指令 [例如 FO （ MV ）] 之后，按此键则输入为 WX 。

(2) 一些操作键上的斜线（ / ）表示每按一次键，被斜线隔开的字符将交替输入显示。

(3) 一些操作键上用橙色表示的字符或功能（如 BIN 、—、 DELT 、 HELP ），只有先按了 SHIFT SC 键后再按这些键才有效。

(4) 每按一次键，蜂鸣器会响一下。如果输入了错误的数值，蜂鸣器就会响两次。所以，当听到蜂鸣器响两声时，须重新输入正确的数值。若连续发声报警，说明操作或运行有误，错误信息随之显示于 LCD 的上面一行。

若要停止报警，可按 ACLR 键或 (HELP) CLR 键停止报警并删除错误指令或解除错误运行状态。

2．指令输入方式

指令按其输入方式可分为三类：键盘指令、非键盘指令和高级功能指令。

1）键盘指令

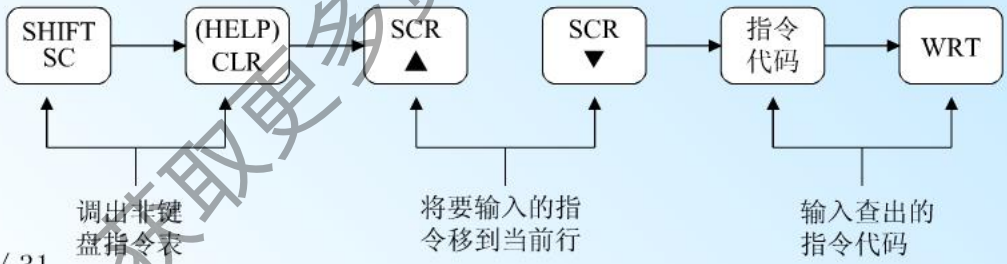
这类指令是指键盘上已标明的指令，只需直接按键即可输人。

2）非键盘指令

这类指令是指键盘上没有，需用指令代码方可输入的指令。输入步骤分为两种：

a．当已知指令代码，输入指令的步骤为：

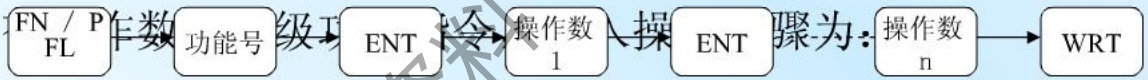
b．当不知道指令代码需借助 (HELP) CLR 键调出非键盘指令表。其输入指令的步骤为：



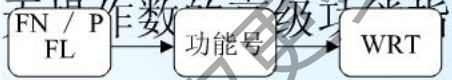
3）高级功能指令

这类指令是指键盘上没有，需借助于 FN/P FL 键方可输入的指令，一条完整的指令包括有“F”、功能号、助记符和若干操作数，其中除助记符可自动生成之外，其它都需一步一步地输入。**每输入完一个内容后，** ENT **键将其存入程序缓冲器中，只有到输入完最后一个操作数后，才用** WRT **键将其指令存入 PLC。**高级功能指令根据指令中是否有操作数而输入步骤不同。

a．无操作数的高级功能指令，输入操作步骤为：



b．无操作数的高级功能指令，输入操作步骤为：



3．清除命令

清除命令可以分为三类：

1）利用 CLR 键清除屏幕当前行显示（即 LCD 上的第二行），以便对该行指令做出修改。

2）利用 ACLR 键将当前屏幕显示全部清除，以便进行程序调试、监控等操作，但程序仍保留在内存中，即仍可重新调出。

3）利用 OP-0 功能将程序从内存中清除，即程序不能再被调出，这是在输入一个新程序之前必须进行的工作。

有关其他操作细节和 OP 功能请读者参阅 FP 编程器 II 操作手册。

第三节 监控组态软件简介

一、监控组态软件简介

1. 概念

组态软件指一些数据采集与过程控制的专用软件，它们是在自动控制系统监控层一级的软件平台和开发环境，能以灵活多样的组态方式（而不是编程方式）**提供良好的用户开发界面和简捷的使用方法，**其预设置的各种软件模块可以非常容易地实现和完成监控层的各项功能，并能同时**支持各种硬件厂家的计算机和 I/O 设备，与高可靠的工控计算机和网络系统结合，**可向控制层和管理层提供软、硬件的全部接口，进行系统集成。

2. 组态软件的发展和现状

世界上第一个把组态软件做为商品进行开发、销售的专业软件公司是美国的 Wonderware 公司，它于 80 年代末率先推出第一个商品化**监控组态软件** Intouch。此后组态软件得到了迅猛的发展。目前世界上的组态软件有几十种之多，国际上较知名的监控组态软件有：

Fix， Intouch， Wincc， LabView， Citech 等。

3. 组态软件的特点

a. 使用简单，用户只需编写少量自己所需的控制算法代码，甚至可以不写代码。

b. 运行可靠。

c. 提供数据采集设备的驱动程序。

d. 提供自动化应用系统所需的组件。

e. **强大的图形设计工具。**

二、力控监控组态软件简介

力控监控组态软件（ForceControl）是一个面向方案的HMI/SCADA（human machine interface/ supervisory control and data acquisition）平台软件。分布式实时多数据库系统，可提供访问工厂和企业系统数据的一个公共入口。**内置TCP/IP协议的网络服务程序使用户可以充分利用Intranet或Internet的网络资源。**

力控可用于开发石油、化工、半导体、汽车、电力等多个行业和领域的工业自动化、过程控制、管理监测、工业现场监视、远程监视/远程诊断等系统。

1. ForceControl 集成环境：

开发系统（Draw）： 是一个集成环境，可以创建工程画面，配置各种系统参数，启动力控其它程序组件等。

界面运行系统（View）： 界面运行系统用来运行由开发系统 Draw 创建的画面。

实时数据库（DB）： 是数据处理的核心，构建分布式应用系统的基础。它负责实时数据处理、历史数据存储、统计数据处理、报警处理、数据服务请求处理等。

I/O 驱动程序： I/O 驱动程序负责力控与 I/O 设备的通信。它将 I/O 设备寄存器中的数据读出后，传送到力控的数据库，然后在界面运行系统的画面上动态显示。

网络通信程序（NetClient/NetServer）： 网络通信程序采用 TCP/IP 通信协议，可利用 Intranet/Internet 实现不同网络结点上力控之间的数据通信。

2. ForceControl 2.6 中其它的可选程序组件：

串行通信程序（ SCOMClient/SCOMServer ）：

两台计算机之间，使用 RS232C/422/485 接口，可实现一对一的通信；如果使用 RS485 总线，还可实现一对多台计算机的通信。

拨号通信程序（ TelClient/TelServer ）：

任何地方与工业现场之间，只要能拨打电话，就可以实现对远程现场生产过程的实时监控，唯一需要的是 Modem 和电话线。

Web 服务器程序（ Web Server ）：

Web 服务器程序可为处在世界各地的远程用户实现在台式机或便携机上用标准浏览器实时监控现场生产过程。

控制策略生成器（ StrategyBuilder ）：

是面向控制的新一代软件**逻辑自动化控制软件**。提供包括：变量、数学运算、逻辑功能和程序控制处理等在内的十几类基本运算块，内置常规 PID 、比值控制、开关控制、斜坡控制等丰富的控制算法。同时提供开放的算法接口，可以嵌入用户自己的控制程序。

三、力控实例入门

1. 建立工程

打开应用管理器，选择“增加新应用”，在应用名称对话框中输入一个应用程序的名称“MonitorPLC”，按“确定”按钮。在工程列表中会出现新建的工程，单击该工程并进入组态，打开Draw，开始组态工作。

3. 创建点

a. Draw 导航器中双击“实时数据库”项使其展开，在展开项目中双击“数据库组态”启动组态程序 DbManger，如图所示。

– 启动 DbManger 后出现 DbManger 主窗口，如图所示。

a. 选择菜单命令“点 / 新建”或在右侧的点表上双击任一空白行，出现“指定区域和点类型”对话框，如图所示。

a. 选择“区域… 00”及“数字 I/O 点”点类型，然后单击“继续》”按钮，进入点定义对话框，如图所示。

供的缺省值。单击 “确定”按钮，在点表中增加了一个点“ MX0”，如图所示。

a. 重复以上步骤，创建 MX1 、 MY0 、 MY1 和 MY2 点。

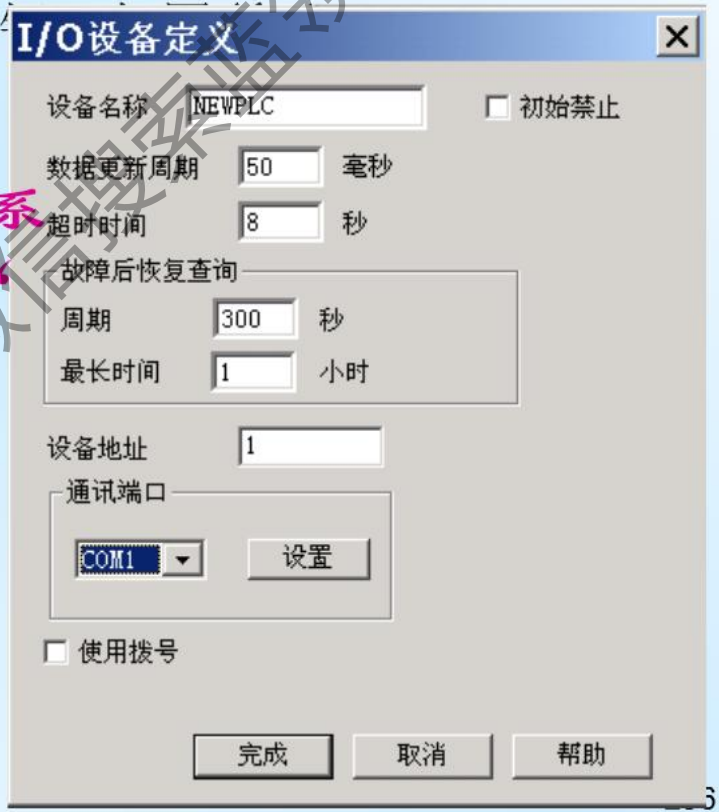
最后单击“存盘”按钮保存组态内容，然后单击“退出”按钮，返回到主窗口。

1. 定义 I/O 设备

在数据库中定义了上述 5 个点后，下面将建立一个 I/O 设备— PLC，上述定义好的 5 个点的值将取自 PLC。

a. 在 Draw 导航器中**双击“实时数据库”**项使其展开，选择“I/O**设备驱动”项**使其展开，在展开项目中选择“PLC”项并双击使其展开，然后继续选择厂商名**“NaiS（松下电工）”**并双击使其展开后，选择项目**“FP 系列”**，如图所示。

a. 双击项目“FP系列”出现“I/O设备定义”对话框，在“设备名称”输入框内键入一个人为定义的名称“NEWPLC”（大小写不限）。在通信端口下拉条中选择“COM1”，“设备地址”输入框内键入1。其余保持默认值。点击“完成”按钮。



此时在导航器的“FP系列”下面增加了一项“NEWPLC”。

a. 数据连接

现在将已经创建的 5 个数据库点与 NEWPLC 联系起来，以使这 5 个点的 PV 参数值能与 I/O 设备 NEWPLC 进行实时数据交换。这个过程就是建立数据连接的过程。由于数据库可以与多个 I/O 设备进行数据交换，所以我们必须指定哪些点与哪个 I/O 设备建立数据连接。

a. 启动数据库组态程序 DbManager，双击点“ MX0”，切换到“数据连接”页，出现如图所示对话框。

b. 点击参数“PV”，在“连接I/O设备”的“设备”下拉框中选择设备“NEWPLC”。点击“增加”按钮，出现如图所示的“设备连接项”对话框。

在“寄存器/继电器”选择框中选择“X/WX（外部输入继电器）”，在“地址”输入框中输入0，“位偏移”输入框中输入0，点击“确定”返回。

重复上述步骤，可连接所有定义过的点。在重复上述步骤时，**对于同一个继电器，位偏移依次加 1 。**对话框中填写的值如表所示。

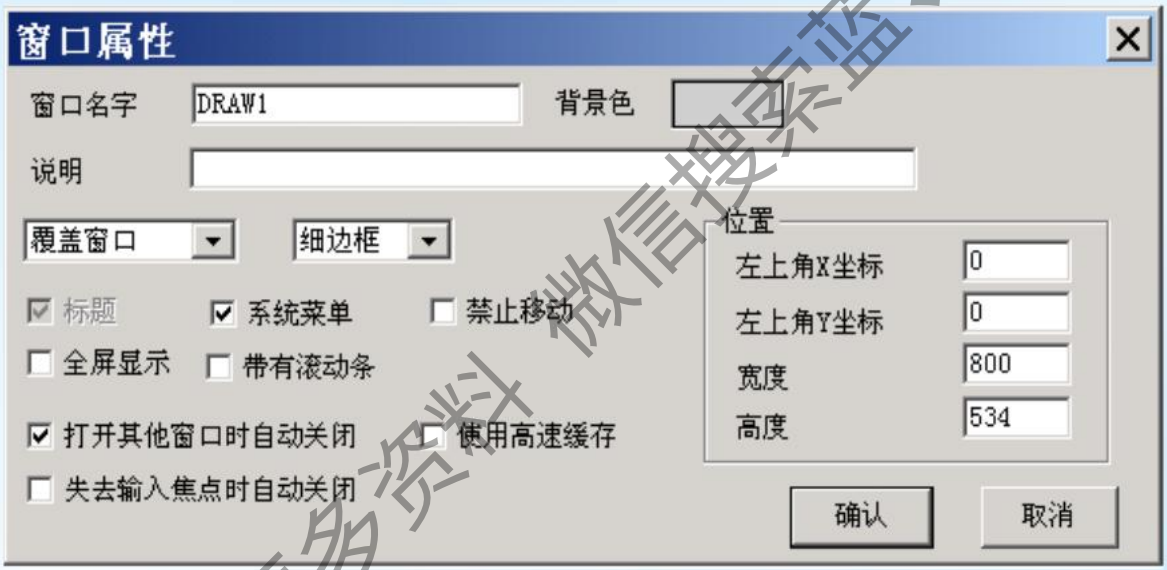
	MX0	MX1	MY0	MY1	MY2
寄存器 / 继电器	X（按位）	X（按位）	Y（按位）	Y（按位）	Y（按位）
数据格式	bit	bit	bit	bit	bit
地址	0	1	0	1	2

最终结果如图所示。单击“退出”按钮，返回 DRAW 主窗口。

	NAME [点名]	DESC [说明]	%IOLINK [I/O连接]
1	MX0		PV=NEWPLC:序号,寄存器,格式,,地址,偏移:1000000-1-0-0-0
2	MX1		PV=NEWPLC:序号,寄存器,格式,,地址,偏移:1000001-1-0-0-1
3	MY0		PV=NEWPLC:序号,寄存器,格式,,地址,偏移:2000000-2-0-0-0
4	MY1		PV=NEWPLC:序号,寄存器,格式,,地址,偏移:2000001-2-0-0-1
5	MY2		PV=NEWPLC:序号,寄存器,格式,,地址,偏移:2000002-2-0-0-2
6			
7			
8			
9			
10			
11			
12			

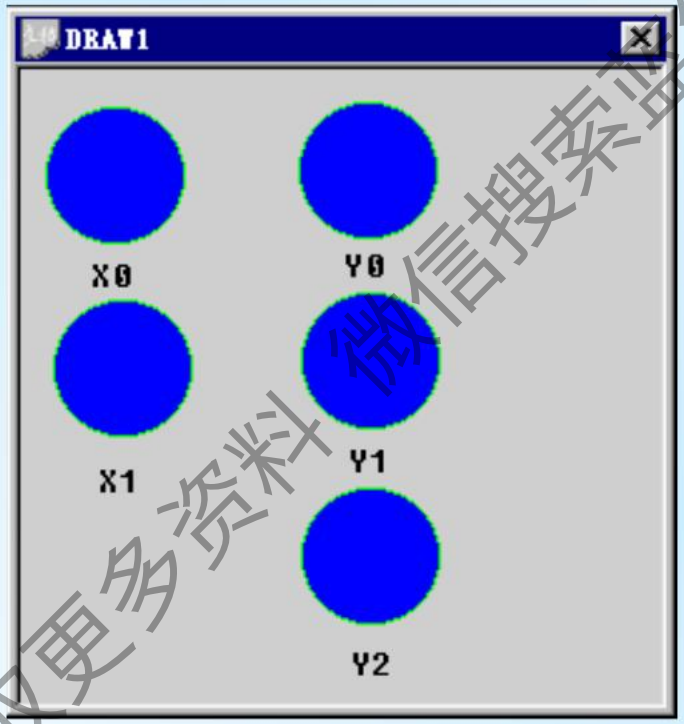
1. 创建窗口

选择“文件[F]/新建”命令出现“窗口属性”对话框，如图所示。



全部保持默认值，点击“确定”按扭，建立了一个新的窗口。

按图所示绘制窗口图形。



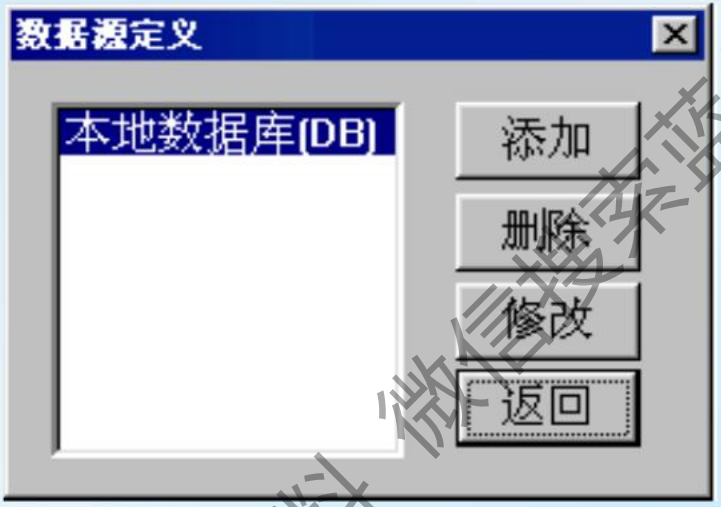
1. 制作动画连接

前面已经做了很多事情，包括：制作显示画面、创建数据库点，并通过一个自己定义的I/O设备“NEWPLC”把数据库点的过程值与设备NEWPLC连接起来。现在再回到开发环境Draw中，通过制作动画链接使显示画面活动起来。

（1）定义数据源

界面系统除了可以访问本地数据库（即与界面系统运行在同一台PC机上的数据库）外，还可以通过网络访问安装在其它计算机上的ForceControl数据库中的数据。因此，当在界面系统Draw中创建变量时，如果变量引用的是外部数据源（包括：ForceControl数据库，DDE服务器或其它第三方数据提供方），首先对要引用的外部数据源进行定义。

激活 Draw 菜单“特殊功能[S]/数据源定义”，出现“数据源定义”列表框，如图所示。



列表框中已经存在了一个数据源：“本地数据库（DB）”。这是**系统缺省定义的数据源，它指向本机上的 DB 数据库**。

（2）单击“取消”和“返回”按钮，退出“数据源定义”对话框。

（3）动画连接

有了变量之后就可以制作动画连接。一旦创建了一个图形对象，给它加上动画连接就相当于赋予它“生命”使其“活动”起来。双击“X0”上面的图形，弹出如图所示的“动画连接”对话框。

单击“颜色相关动作”一列中的“条件”按扭，弹出“颜色变化”对话框如图所示。

单击“变量选择”按扭，弹出“变量选择”对话框，如图所示。

在上图中，选择“MX0”和“PV”，点击“选择”按扭。然后“确对话框，则第一个圆的动画连接就制作完成。同理，按上述定义其余图形的动画连接。注意变量选择与相应的标注相同，0 为监视 PLC 中的 X0 的接点，依次类推。保存制作结果。

1. 配置系统

在导航器中选择“配置”、“初始启动设置”，弹出“初始启动设置”对话框，如图所示。点击“增加”按扭，选择“DRAW1”，“确定”该对话框。

到现在为止，上位机的组态程序已经制作完成。连接 PLC 和计算机，启动 FPWIN-GR，编一小段 PLC 程序下载到 PLC 中并让其运行，再切换到“离线”状态。然后在 ForceControl 工程管理器中选择应用程序“MonitorPLC”，进入“运行系统”。接通 PLC 的 X0， X1 点可以看到组态画面上的图形颜色随 PLC 上接点的变化而变化。

第七章 监控组态软件与PLC应用总体设计

第一节 自动售货机PLC控制与监控组态设计

一、仿真系统组成

本仿真系统由上位机和下位机两部分组成。上位机利用PC机，下位机利用松下的FP1系列可编程控制器FP1-C24。

上位机内装北京力控组态软件FORCECONTROL 2.6和松下编程软件FPWIN-GR。组态软件FORCECONTROL用以制作仿真画面、编写仿真程序并与下位机进行通信。FPWIN-GR是松下可编程序控制器与PC机联机的编程支持工具，利用它可以实现程序输入、程序注释、程序修改、程序编译、状态监控和测试以及设置系统寄存器和PLC各种参数等。

二、 自动售货机功能分析

1．自动售货机的基本功能

售货机基本功能：对投入的货币进行运算，并根据货币数值判断是否能购买某种商品，并做出相应的反应。

2．仿真实验系统中售货机的分析

售货机的全部功能是在上位机上模拟的，其部分硬件由计算机软件模拟代替。

如钱币识别系统可以用按压某个“仿真对象”输出一个脉冲直接给 PLC 发布命令。

1）实验状态假设

a 自动售货机只售 8 种商品；

b. 自动售货机可识别 10 元、 5 元、 1 元、 5 角、 1 角硬币；

c. 自动售货机可退币 10 元、 5 元、 1 元、 5 角、 1 角硬币；

d. 自动售货机有液晶显示功能；

f. 实验中售货机忽略了各种故障以及缺货等因素。

2）一次交易过程分析

a. **初始状态：** 由电子标签显示各商品价格，显示屏显示友好界面，此时不能购买任何商品。

b. **投币状态：** 按下投币按钮，显示投币框，按下所投币值，显示屏显示投入、消费、余额数值，当所投币值超过某商品价格时，相应商品选择按钮发生变化，提示可以购买。

c. **购买状态：** 按下可以购买的“选择”按钮，所选的商品出现在出货框中，同时显示屏上的金额数字根据消费情况相应变化。取走商品后出货框消失。

d. **退币状态：** 按下退币按钮，显示退币框，同时显示出应退币值及数量。按下确认钮，则恢复初始状态。

三、设计任务的确定

上位机与下位机之间的任务分工：

上位机主要用来完成仿真界面的制作工作；下位机则主要用来完成 PLC 程序的编写。

在进行 PLC 程序的编写时需要先分配 PLC 的 I/O 点，确定上、下位机的接口。然后，对上位机和下位机分别进行设计工作。最后，进行上位机设计结果与下位机设计结果的配合工作，经调试后完成整个系统的设计。

一方面，仿真的自动售货机接受 PLC 的控制指令并完成相应的动作；另一方面，仿真界面中的仿真自动售货机的运行，都是由组态界面所提供的命令语言来完成的。

四、程序设计部分

1．程序设计说明

仿真程序的编写利用了力控组态软件FORCECONTROL2.6。下位机程序的编制则是利用松下PLC专用编程软件FPWIN-GR完成的。

2． PLC程序设计

把一次交易过程分为几个程序块：

运行初期电子标签价格的内部传递；投币过程；价格比较过程；选择商品过程；退币过程。

1）运行初期电子标签价格的内部传递程序的设计

仿真系统运行初期的任务：

① 要由 PLC 向仿真画面相应对象传递已经存储好的价格；

② 给投入显示、消费显示及余额显示寄存器清零；

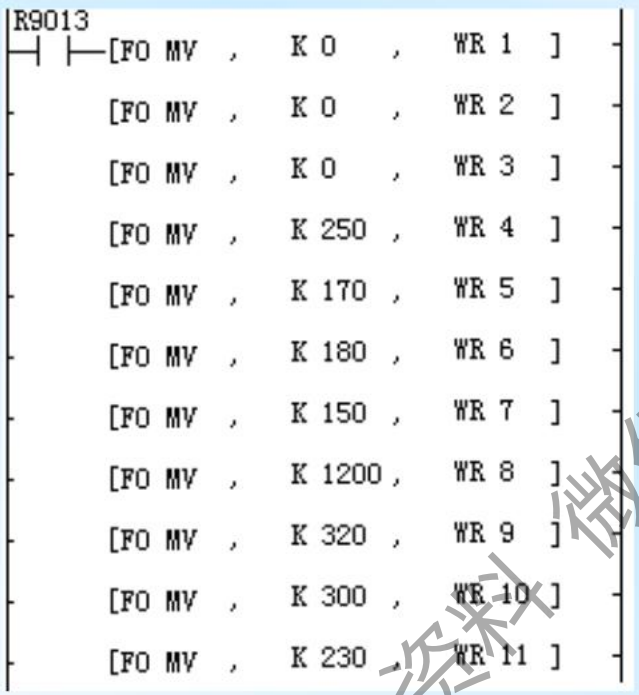
③ 给存储退币币值的存储器清零。

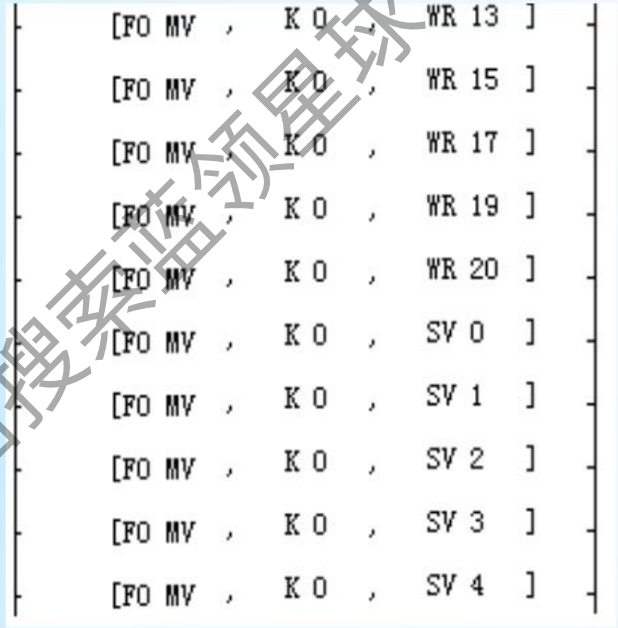
程序编制过程中，要用到运行初期闭合继电器 R9013 、 16 位数据传送指令 F0 ，**同时在上位机 FORCECONTROL 中，必须定义相应的变量，来实现与 PLC 程序的对接。**

电子标签价格内部传递变量表：

说明	上位机 FORCECONTROL 变量	对应 PLC 地址
投入显示	POITR001.PV	WR1
消费显示	POIXF002.PV	WR2
余额显示	POIYE003.PV	WR3
01 商品价格	JG01.PV	WR4
02 商品价格	JG02.PV	WR5
03 商品价格	JG03.PV	WR6
04 商品价格	JG04.PV	WR7
05 商品价格	JG05.PV	WR8
06 商品价格	JG06.PV	WR9
07 商品价格	JG07.PV	WR10
08 商品价格	JG08.PV	WR11
退币 10 元	TB$100.PV	SV0
退币 5 元	TB$50.PV	SV1
退币 1 元	TB$10.PV	SV2
退币 5 角	TB$5.PV	SV3
退币 1 角	TB$1.PV	SV4

运行初期电子标签价格的内部传递程序：





系统初始化时，通过运行初期闭合继电器R9013在第一次扫描时将数值传递给上位机。通过指令F0给WR1～WR11及SV0～SV4赋初值。

2） 投币过程

每投下一枚硬币，**投入显示将增加相应的币值，余额也增加同样的币值。**

投币过程变量表：

说明	上位机 FORCECONTROL 变量	对应 PLC 地址
投入一角	TR$1.PV	R200
投入五角	TR$5.PV	R201
投入一元	TR$10.PV	R202
投入五元	TR$50.PV	R203
投入十元	TR$100.PV	R204

投币过程梯形图：

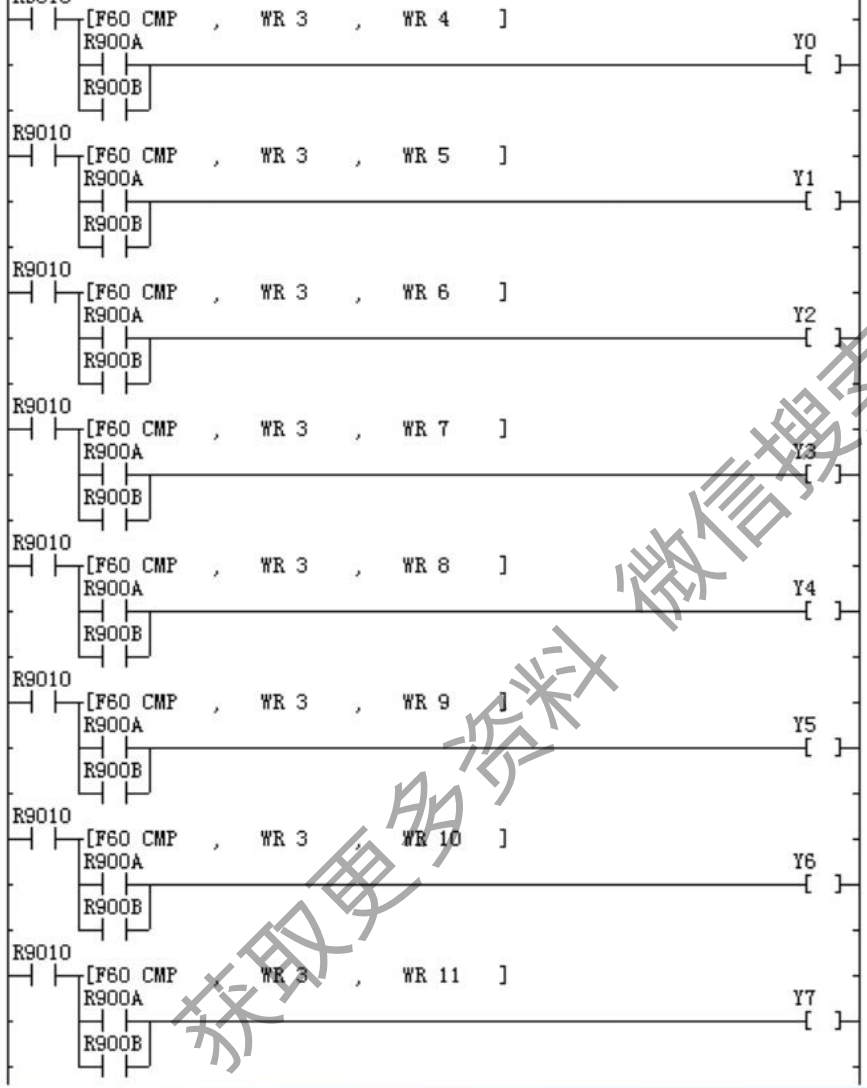
在上图中，当按下投入一角时，相当于让 R200 接通，**之所以用一个微分指令，就是要只在接通时检测一次，不能永远加下去**。投入一角要使投入显示、余额显示都相应增加相同数值，加法由 16 位加法指令 F20 实现的。投入五角、一元、五元、十元，原理同上。

3） 价格比较过程

价格的比较要贯穿实验过程的始终，只要余额大于某种商品价格时，就需要输出一个信号，提示可以购买。这里用选择灯来代表此信号。

价格比较过程变量表

说明	上位机 FORCECONTROL 变量	对应 PLC 地址
01 商品灯亮	D01.PV	Y0
02 商品灯亮	D02.PV	Y1
03 商品灯亮	D03.PV	Y2
04 商品灯亮	D04.PV	Y3
05 商品灯亮	D05.PV	Y4
06 商品灯亮	D06.PV	Y5
07 商品灯亮	D07.PV	Y6
08 商品灯亮	D08.PV	Y7



本图中，在程序执行过程中，R9010 始终保持闭合，是 16 位数据比较指令， F60 用来比较余额和商品的价格， R900A 是大于标志，R900B 是等于标志。**当余额大于等于某种商品价格时，程序使相应的指示灯闪烁表示可以购买该种商品。**

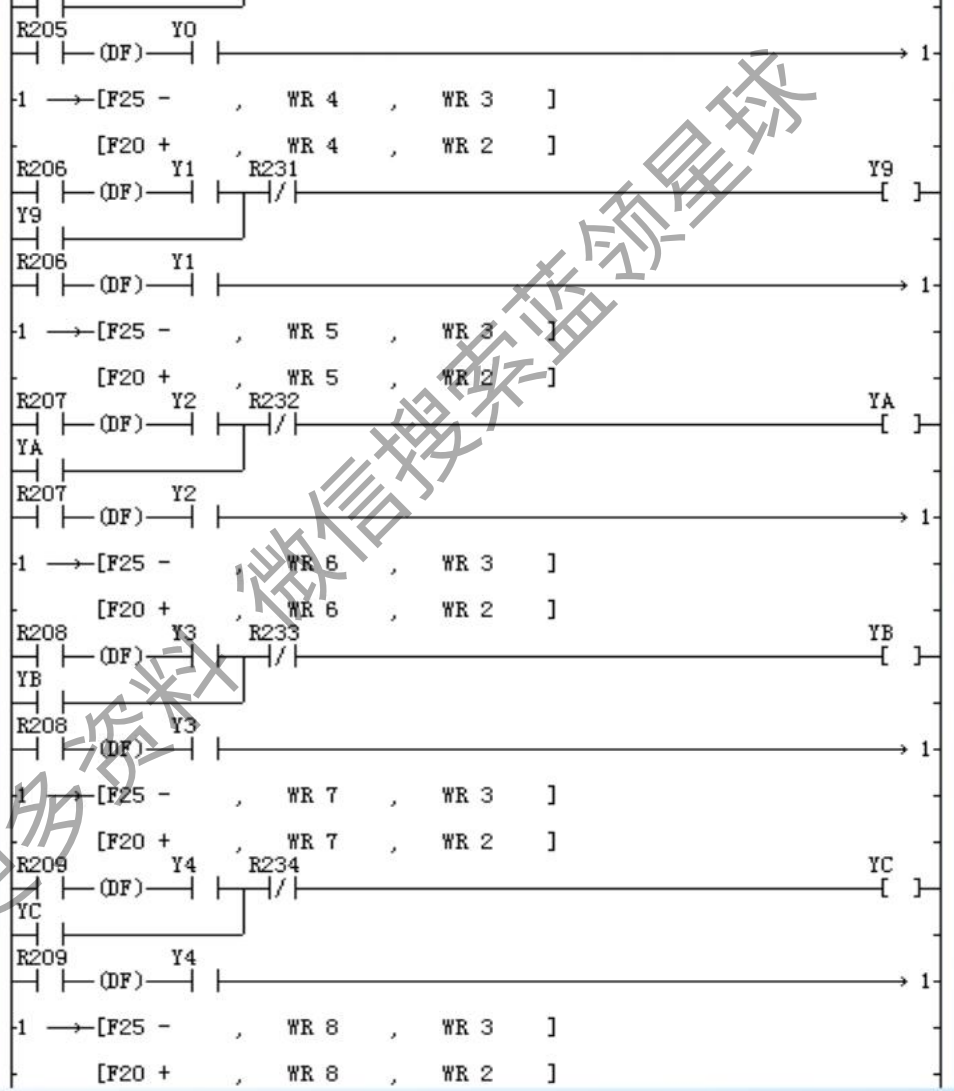
4）选择商品过程

当投入的币值可以购买某种商品时，按下相应的“选择”按钮即可在出货框中出现该种商品，同时消费显示栏中显示出已经消费掉的金额，余额也将扣除已消费的币值，接着余额继续与价格比较，判断是否能继续购买。出现在出货口的商品在没有取走前，一直保持显示状态，用鼠标点击该商品代表已经取走，出货口中的商品隐藏。

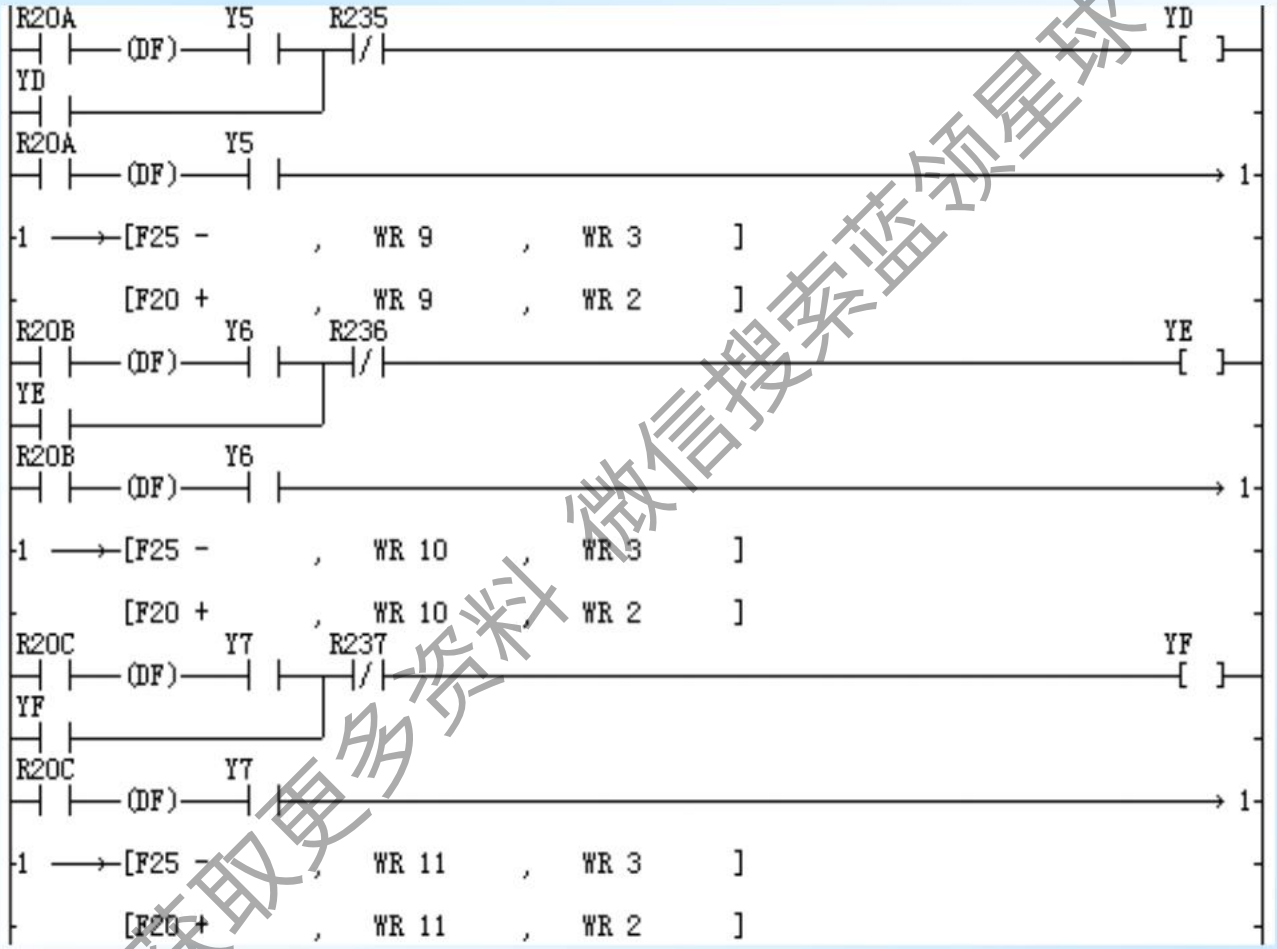
选择商品过程变量表

说明	上位机 FORCECONTROL 变量	对应 PLC 地址
选择 01 商品	XZ01.PV	R205
选择 02 商品	XZ02.PV	R206
选择 03 商品	XZ03.PV	R207
选择 04 商品	XZ04.PV	R208
选择 05 商品	XZ05.PV	R209
选择 06 商品	XZ06.PV	R20A
选择 07 商品	XZ07.PV	R20B
选择 08 商品	XZ08.PV	R20C
01 商品出现	CX01.PV	Y8
02 商品出现	CX02.PV	Y9
03 商品出现	CX03.PV	YA
04 商品出现	CX04.PV	YB
05 商品出现	CX05.PV	YC
06 商品出现	CX06.PV	YD
07 商品出现	CX07.PV	YE
08 商品出现	CX08.PV	YF

取 01 商品	Q01.PV	R230
取 02 商品	Q02.PV	R231
取 03 商品	Q03.PV	R232
取 04 商品	Q04.PV	R233
取 05 商品	Q05.PV	R234
取 06 商品	Q06.PV	R235
取 07 商品	Q07.PV	R236
取 08 商品	Q08.PV	R237



选择商品梯形图（续上图）



在选择商品的过程中：

一是要使商品出现在出货框中，二是要实现内部货币的运算。

如：按下选择 01 商品键，相当于给 R205 加一个信号（只接受一次脉冲，所以用 DF 微分指令），当 Y0 接通（01 商品灯亮）时，则系统显示可以购买 01 商品。由于取 01 商品 R230 是常闭触点，故 Y8 输出，代表在出货框中出现 01 商品，购买成功。当按下取 01 商品按钮时，R230 断开，不能输出 Y8，代表 01 商品被取走。

内部币值的计算和是否取走商品无关，只要按下选择按钮，并且可以购买此商品就要从余额中扣除相应的金额，显示消费的币值。加法由 F20 指令实现，减法由 F25 指令实现。

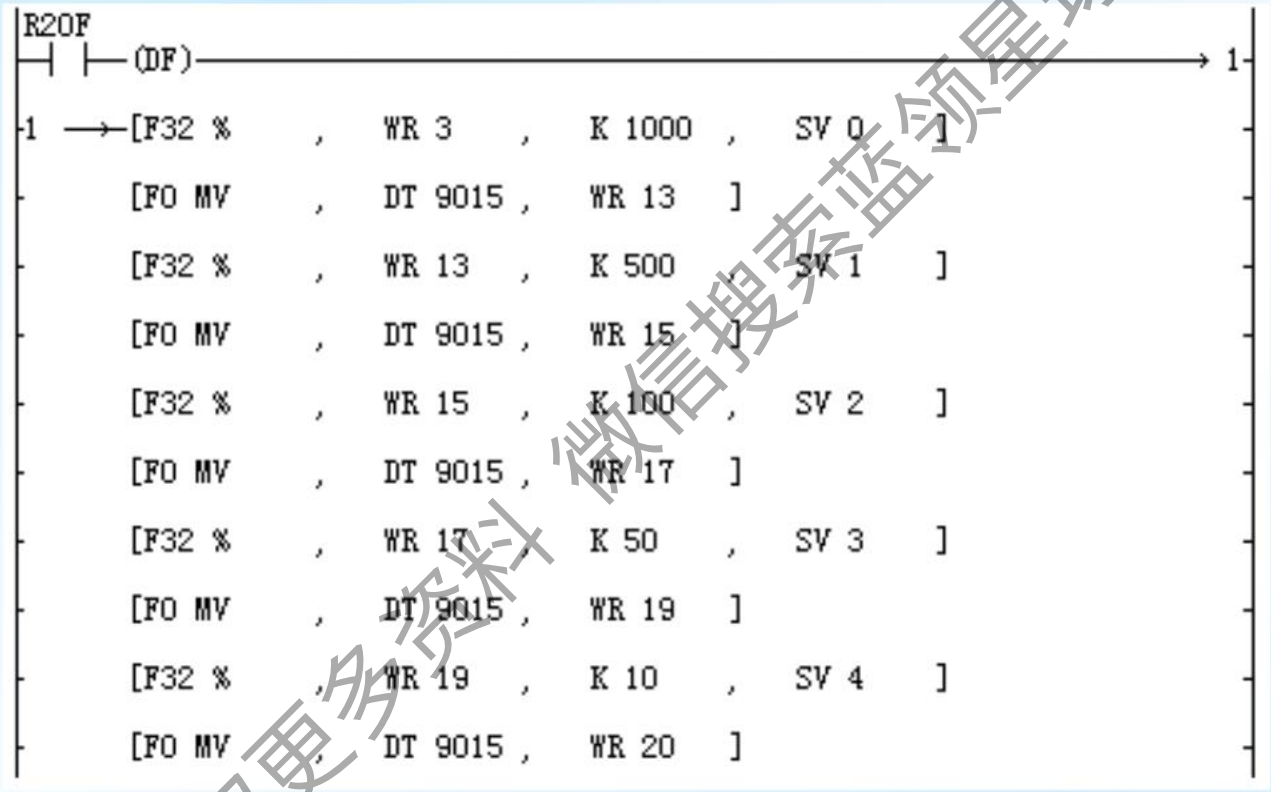
5）退币过程

在退币过程中，最主要的是要完成退币的运算过程，根据结果输出相应的钱币，**退币结束时还要给程序中使用到的某些寄存器重新赋零。**

退币过程变量表：

说明	上位机 FORCECONTROL 变量	对应 PLC 地址
退币按钮	TENTER	R20F
退币 0.1 元	TB$1.PV	SV4
退币 0.5 元	TB$5.PV	SV3
退币 1 元	TB$10.PV	SV2
退币 5 元	TB$50.PV	SV1
退币 10 元	TB$100.PV	SV0
退币确任按钮	TUIBIOK.PV	R0

退币过程梯形图：



退币过程：在按下退币按钮（即 R20F 接通）时执行，同样也用到一个微分指令，在接收到信号时产生一次开关脉冲，进而执行一次其下面的指令。

F32 是除法指令，第一次将余额的币值除以 1000，商存储于 SV0 中，作为退币 10 元的输出值。余数则存储于特殊数据寄存器 DT9015 中，下次将不能被 1000（10 元）整除的余数除以 500（5 元），商存储于 SV1 中，余数继续下传，直至被 1 角除过，由于所投币值最小是 1 角，并且商品价格也确定在整角，所以最终能被 1 角整除。

在程序的初始化时曾给 WR13、WR15、WR17、WR19 和 WR20 赋零，WR13、WR15、WR17、WR19 和 WR20 是程序的中间量。

退币过程结束后，PLC 要将寄存器中的数值置回原定的初值 0，完成一次交易，防止下一次交易时出错。

数据初始化梯形图：

程序中分别将投入显示、消费显示、余额显示、**10**元存储、**5**元存储、**1**元存储、**5**角存储和**1**角存储清零，还将中间量**WR13**、**WR15**、**WR17**、**WR19**和**WR20**清零。

五、仿真界面的设计

1. 售货机背景的设计

售货机背景是一个不动的画面，可以利用图片处理的方法按照制定样式的功能画出售货机的整体。

2. 显示屏部分的设计

图中的欢迎字符是可以闪烁变化的，‘aaaa’字符可以用来显示系统的时间。

交易过程中的币值显示画面如下图：

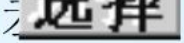
3．电子标签的设计

电子标签用来显示程序中传递上来的价格，其中要有可以定义的字符，设计出的标签为 

。

其中的字符‘J.03’表示可以显示03商品价格的变量。

4．按钮的设计

“选择”按钮的设计要反映出可以购买和不可购买时的差 

所以其中也要有可以变化的字符。设计如右图，**其中字符‘选择’在满足条件以后可以闪烁变色。按钮均可以动作**。

5．投退币提示框的设计

投、退币提示框中要有可以投入的硬币、确认按钮以及框架，其中硬币、确认按钮和字符‘a’均是可以定义的变量。

投币提示框

退币提示框

6．出货框的设计出货框

出货框中要有01至08商品的示意图以及框架。其中的商品在满足条件后可以出现，鼠标点击后可以消失，是可定义的变量。

出货框

三、仿真界面中各变量的定义

仿真程序上的各部分若实现仿真功能，就必须定义成相应的变量，再与 PLC 程序中的软继电器相匹配，这样才能实现 PLC 的控制功能。

1．中间变量

中间变量的作用域为整个应用程序，不限于单个窗口。中间变量适于作为整个应用程序动作控制的全局性变量、全局引用的计算变量或用于保存临时结果。

该仿真实验系统中有 3 个中间变量：

(1)poiwindows：该变量是控制显示屏的。

poiwindows=1：显示屏进入投币交易状态；

poiwindows=0：显示屏返回初始欢迎状态。

(2)poiwinJB：该变量是控制投币框的。

poiwinJB=1：显示投币框；

poiwinJB=0：投币框消失。

(3)poiwinTB：该变量是控制退币框的。

poiwinTB=1：显示退币框，

poiwinTB=0：退币框消失。

2. 数据库变量

当要在界面上显示处理数据库中的数据时，需要使用数据库变量。一个数据库变量对应数据库中的一个点参数。数据库变量的作用域为整个应用程序。

数据库变量有三种：实型数据库变量、整型数据库变量和字符数据库变量。

仿真系统中有 56 个整型数据库变量，分别对应 PLC 程序中的 56 个软继电器。

3. 仿真界面与 PLC 程序的配合定义

(1) 初始状态

poiwindows=0：显示屏显示初始欢迎状态。

以 01 商品为例，电子标签中的字符‘J.01’对应的变量 JG01.PV 与 PLC 程序中的地址 WR4 相匹配，WR4 中存储的数据为 250，如何让字符显示 2.50 元呢？

方法：在开发系统（Draw）中，双击字符‘J.01’，来到“动画连接”画面，选择“数值输出”中的“模拟”项，键入‘JG01.PV/100’即可，由250到2.50实际是计算机来完成的。其它的价格也是如此显示的。

动画连接画面

(2)投币状态

当投币时，按下“投币”提示字，出现投币框。如何定义“投币”呢？双击汉字“投币”，来到“动画连接”画面，选择“触敏动作”中的“左键动作”，在“动作描述”框中作如下定义：按下鼠标时， poiwinJB=1，poiwinJB 这个变量是控制投币框的，当 poiwinJB=1 时，出现钱币和提示框； poiwinJB=0，钱币和提示框隐藏。

下面分别定义提示框和钱币以及“确认”按钮。

双击提示框，来到“动画连接”画面，选择“显示／隐藏”项，定义 poiwinJB==1 时显示，各硬币也用同样的方法定义，“确认”按钮也同样定义，这样就使在按下汉字“投币”时，变量 poiwinJB=1，从而出现投币框，以及硬币等。

定义了投币框的显示状态，用鼠标点击代替了实际过程中的钱币投入动作，最重要的任务是投币运算，下面介绍钱币的定义方法。

以下元为例：双击下元按钮，来到“动画连接画面，选择“触敏动作”中的“左键动作”，在动作描述中如下定义：按下鼠标时， poiwindows=1; TR$100.PV=1; 释放鼠标时 ,TR$100.PV=0; 动作描述画面下图所示。

其中 poiwindows=1，是让显示屏不再显示友好界面，来到交易界面；

TR$100.PV=1 时给 PLC 发出一个接通信号，由于 TR$100.PV 对应的 PLC 地址是 R204，使得 R204 继电器导通，转而执行相应的加十元程序。

同样定义其它钱币，注意其对应的 PLC 软继电器。

最后还要定义“确认”按钮。要实现的功能是按下“确认”按钮时，所有的钱币以及投币提示框均消失。

这里作如下定义：双击“确认”按钮，来到“动画连接”画面，选择“触敏动作”中的“左键动作”，在动作描述中作如下定义：按下鼠标时， poiwinJB=0；

poiwinJB=0 时，所有的钱币以及投币提示框均消失，这是由计算机控制的内部变量。

投币以后，显示屏要及时反映出投币情况，同时“选择”指示也要相应变化（闪烁、变色）。下面来定义显示屏和“选择”按钮。

显示屏要显示3种数据，分别为：投入显示、消费显示、余额显示。

三种显示均用力控软件自带的附件—数码管来显示。

先在工具箱中点击“选择子图项”，在子图库中找到仪表中的数码管，放在显示屏中，作为投入显示，再复制两个，分别作为消费显示、余额显示。双击数码管来到数码管属性设置画面，在表达式中作如下定义：poiTR001.PV/100，poiTR001.PV 连接的是 PLC 程序中的 WR1 软继电器，是用来存储投入显示数据的，除以 100 同样是为了 PLC 数据计算的方便。这样就可用数码管来显示投入的币值。同样定义消费显示，余额显示。

数码管的属性设置画面如图所示：

“选择”按钮要根据余额的数值发生闪烁和变色。

定义过程如下：双击“选择”按钮，来到“动画连接”画面，在“颜色相关动作”中选择“闪烁”项，分别定义属性和频率，在变量选择项中选择相应的指示灯变量。

以 01 商品的选择指示灯为例，在变量选择项中选择 D01.PV==1，**满足条件时指示灯变色。**

这样就定义好了投币状态的上位机仿真变量，配合 PLC 程序可以实现投币功能。

下图是一幅投币时的画面，投入 6.60 元，还未买商品，注意看显示屏的显示以及选择按钮的变化，此时还不能购买 05 号商品（价格 12 元）。

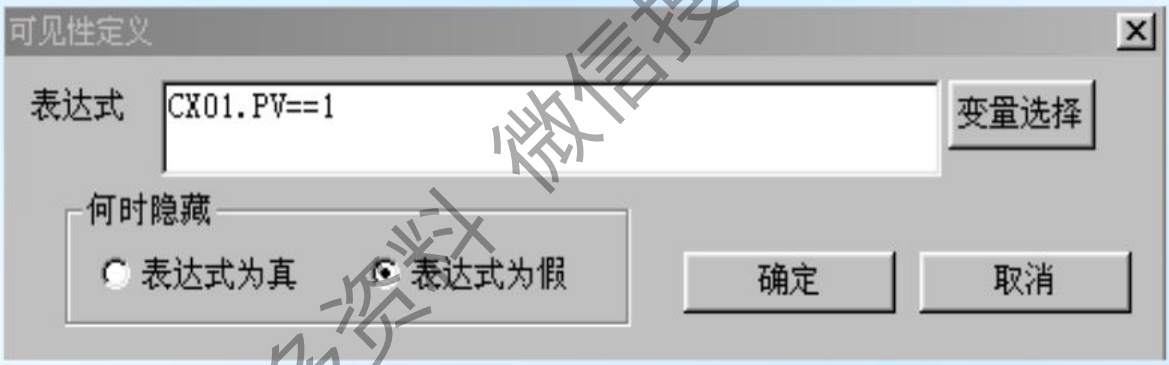
(3) 购买状态

定义了投币状态，就可以购买商品了。**当选择指示灯变色以后，按下它，将会在出货口处出现我们要买的商品。**

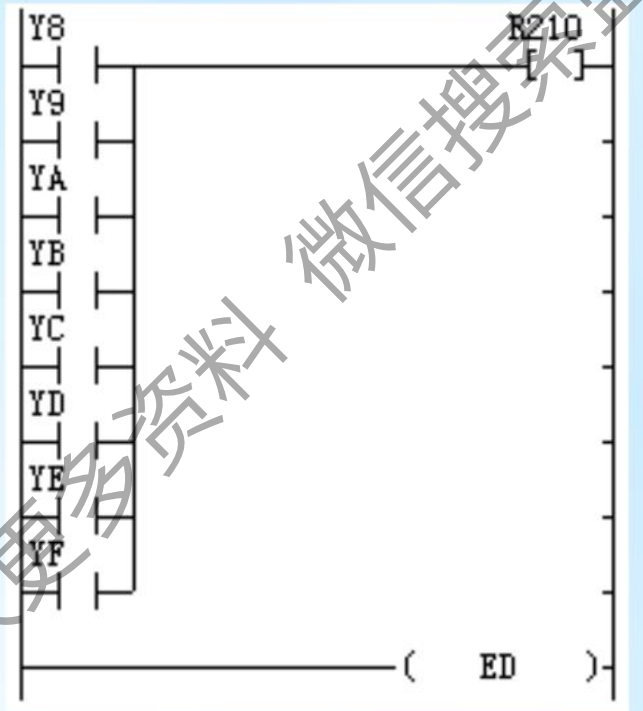
以 01 商品为例：定义“选择”按钮：双击“选择”按钮，来到“动画连接”画面，选择“触敏动作”中的“左键动作”，在动作描述中如下定义：按下鼠标时 XZ01.PV=1；释放鼠标时，XZ01.PV=0 。XZ01.PV 与 PLC 程序中的 R205 相对应，**按下可以购买商品的选择键，转而执行相应的 PLC 程序同时消费显示增加相应的币值，余额显示减少相应的币值，此时还要在出货口处出现相应的商品。**

用“显示/隐藏”功能来定义在出货口中出现的商品。

双击出货口处的小商品，来到“动画连接”画面，选择“显示/隐藏”项，定义 **CX01.PV==1** 时显示。定义画面下图所示。



出货口框架的隐藏/显现是用程序来控制的。当有一种商品出现在出货口，就会显示框架；当全部商品均消失后框架隐藏。程序如下图所示。图中 **R210** 是控制出货口框架是否出现的继电器。



(4) 退币状态

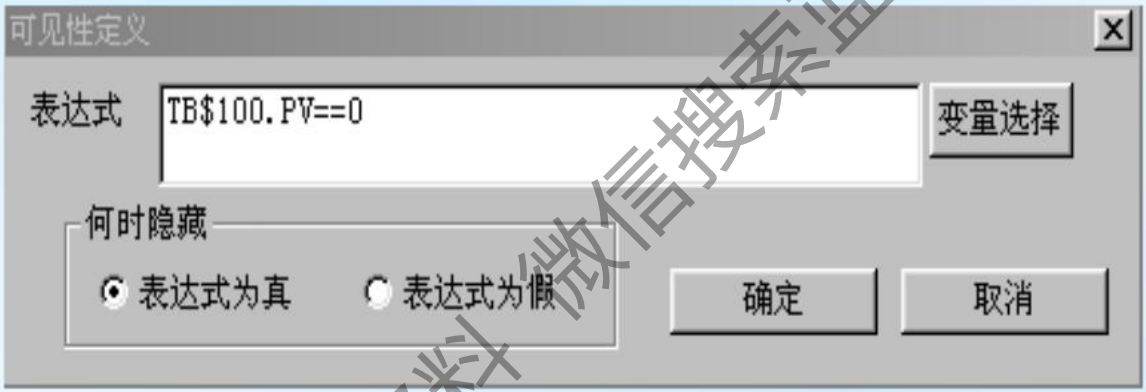
当按下“退币”按钮时，PLC 要进行退币运算。所以**按下“退币“按钮就要与 PLC 通讯，执行退币计算。**

下面来定义退币按钮。

双击“退币”按钮，出现“动画连接”画面，选择“触敏动作”中的“左键动作”，动作描述为：按下鼠标，poiwinJB=0；poiwinTB=1；Tenter.PV=1。释放鼠标，Tenter.PV=0；内部变量 poiwinJB=0 是让投币框消失，poiwinTB=1 是让退币框出现，Tenter.PV 与 PLC 程序中的 R20F 对应。

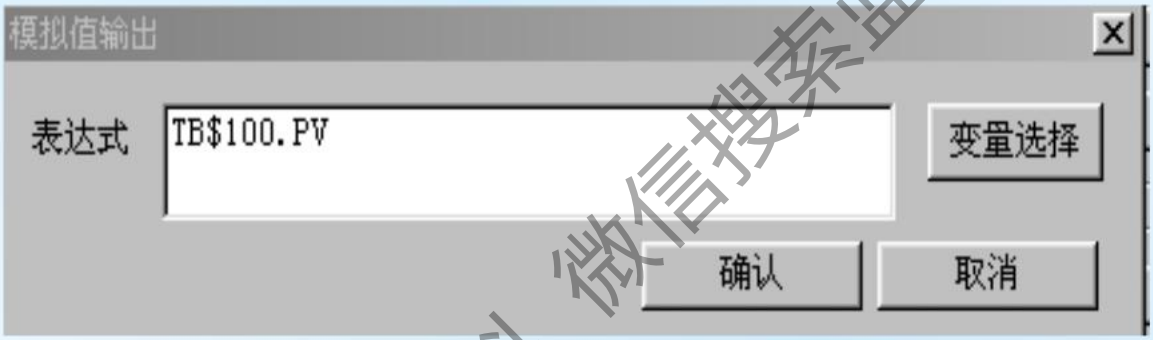
退币框中要有五种硬币，还要有表示硬币个数的数字。由于计算中采用的算法使得退币时按照币值大小顺序退币，例如退五元，只退一个五元，而不退五个一元。定义表示硬币个数的变量只用一位数即可。**在退币时，要退出的硬币及个数显示，而不退的硬币隐藏。**

以十元为例，定义钱币时，双击十元硬币，出现动画连接画面，选择“显现/隐藏”项，在“可见性定义”对话框中作如下图的定义。



其它硬币定义方法同上。

定义钱币个数：双击十元硬币个数字符“a”，出现“动画连接”画面，选择“数值输出”中的“模拟”项，作下图的定义。



同时钱币个数也要定义是否隐藏，定义方法和定义钱币相同。

定义“确认”键时，**按下”确认”键，代表取走了所有硬币，完成此次交易，因此退币“确认”键的定义很重要。**双击“确认”键，出现“动画连接”画面，选择“触敏动作”中的“左键动作”，在动作描述栏中定义如下：按下鼠标

poiwinTB=0；　　功能：退币框消失；

TuiBiok.PV=1；　功能：给 PLC 信号，闭合 R0，

完成数据的初始化；

poiwindows=0；　功能：显示屏显示友好界面

为了防止在未取走商品时退币，按下“确认”键后又不能返回原始状态。在描述中加入以下一段程序，**强行抛掉已经购买的商品。**

```
CX01.PV=0；
CX02.PV=0；
CX03.PV=0；
CX04.PV=0；
CX05.PV=0；
CX06.PV=0；
CX07.PV=0；
CX08.PV=0；
```

释放鼠标时，TuiBiok.PV=0。只是给PLC一个微分信号，不能将R0永远置为1。还有一点要注意，“确认”键也要有隐藏的时候，定义方法同钱币。

定义退币框架：

按下“退币”按钮后就会出现退币框架。可以这样定义：双击退币框架，来到“动画连接”画面，选择“显现/隐藏”项，在“可见性定义”表达式中定义poiwinTB==1时显现即可。

七、数据连接

1．定义 I/O 设备

数据库是从 I/O 驱动程序中获取过程数据的，而数据库同时可以与多个 I/O 驱动程序进行通信，一个 I/O 驱动程序也可以连接一个或多个设备。

下面创建 I/O 设备。

1）在 Draw 导航器中双击“实时数据库”项使其展开，选择“I/O 设备驱动”项使其展开，在展开项目中选择“PLC”项双击使其展开，然后继续选择厂商名“松下电工”并双击使其展开后，选择项目双击并按下图定义。

2) 单击“完成”按钮返回，在“松下电工”项目下面增加了一项“PLC001”。

如果要对 I/O 设备“ PLC001”的配置进行修改，双击项目“ PLC001”，会再次出现 PLC001 的“ I/O 设备定义”对话框。若要删除 I/O 设备“PLC001”，用鼠标右键单击项目“ PLC001”，在弹出的右键菜单中选择“删除”。

2．数据连接

刚刚创建了一个名为“ PLC001”的 I/O 设备，而且它连接的正是假想的 PLC 设备。

现在的问题是如何将已经创建的多个数据库点与 PLC 联系起来，以使这些点的 PV 参数值能与 I/O 设备 PLC 进行实时数据交换，这个过程就是建立数据连接的过程。

点与哪个 I/O 设备建立数据连接。为方便起见，将数据列表整理成如下表所示。

数字 I/O 表

	NAME [点名]	DESC [说明]	%IOLINK [I/O连接]
1	CX01	01商品出现	PV=PLC001:序号,寄存器,格式,地址,偏移:2000008-2-0-0-8
2	CX02	02商品出现	PV=PLC001:序号,寄存器,格式,地址,偏移:2000009-2-0-0-9
3	CX03	03商品出现	PV=PLC001:序号,寄存器,格式,地址,偏移:2000010-2-0-0-10
4	CX04	04商品出现	PV=PLC001:序号,寄存器,格式,地址,偏移:2000011-2-0-0-11
5	CX05	05商品出现	PV=PLC001:序号,寄存器,格式,地址,偏移:2000012-2-0-0-12
6	CX06	06商品出现	PV=PLC001:序号,寄存器,格式,地址,偏移:2000013-2-0-0-13
7	CX07	07商品出现	PV=PLC001:序号,寄存器,格式,地址,偏移:2000014-2-0-0-14
8	CX08	08商品出现	PV=PLC001:序号,寄存器,格式,地址,偏移:2000015-2-0-0-15
9	D01	01商品灯亮	PV=PLC001:序号,寄存器,格式,地址,偏移:2000000-2-0-0-0
10	D02	02商品灯亮	PV=PLC001:序号,寄存器,格式,地址,偏移:2000001-2-0-0-1
11	D03	03商品灯亮	PV=PLC001:序号,寄存器,格式,地址,偏移:2000002-2-0-0-2
12	D04	04商品灯亮	PV=PLC001:序号,寄存器,格式,地址,偏移:2000003-2-0-0-3
13	D05	05商品灯亮	PV=PLC001:序号,寄存器,格式,地址,偏移:2000004-2-0-0-4
14	D06	06商品灯亮	PV=PLC001:序号,寄存器,格式,地址,偏移:2000005-2-0-0-5
15	D07	07商品灯亮	PV=PLC001:序号,寄存器,格式,地址,偏移:2000006-2-0-0-6
16	D08	08商品灯亮	PV=PLC001:序号,寄存器,格式,地址,偏移:2000007-2-0-0-7
17	Q01	取01商品	PV=PLC001:序号,寄存器,格式,地址,偏移:3002300-3-0-23-0
18	Q02	取02商品	PV=PLC001:序号,寄存器,格式,地址,偏移:3002301-3-0-23-1
19	Q03	取03商品	PV=PLC001:序号,寄存器,格式,地址,偏移:3002302-3-0-23-2
20	Q04	取04商品	PV=PLC001:序号,寄存器,格式,地址,偏移:3002303-3-0-23-3
21	Q05	取05商品	PV=PLC001:序号,寄存器,格式,地址,偏移:3002304-3-0-23-4
22	Q06	取06商品	PV=PLC001:序号,寄存器,格式,地址,偏移:3002305-3-0-23-5
23	Q07	取07商品	PV=PLC001:序号,寄存器,格式,地址,偏移:3002306-3-0-23-6

（续上表）

24	Q08	取08商品	PV=PLC001:序号,寄存器,格式,地址,偏移:3002307-3-0-23-7
25	TENTER	退币按钮	PV=PLC001:序号,寄存器,格式,地址,偏移:3002015-3-0-20-15
26	TR$1	投入1角	PV=PLC001:序号,寄存器,格式,地址,偏移:3002000-3-0-20-0
27	TR$10	投入10角	PV=PLC001:序号,寄存器,格式,地址,偏移:3002002-3-0-20-2
28	TR$100	投入100角	PV=PLC001:序号,寄存器,格式,地址,偏移:3002004-3-0-20-4
29	TR$5	投入5角	PV=PLC001:序号,寄存器,格式,地址,偏移:3002001-3-0-20-1
30	TR$50	投入50角	PV=PLC001:序号,寄存器,格式,地址,偏移:3002003-3-0-20-3
31	TUIBIOK	退币ok	PV=PLC001:序号,寄存器,格式,地址,偏移:3000000-3-0-0-0
32	XZ01	选择01商品	PV=PLC001:序号,寄存器,格式,地址,偏移:3002005-3-0-20-5
33	XZ02	选择02商品	PV=PLC001:序号,寄存器,格式,地址,偏移:3002006-3-0-20-6
34	XZ03	选择03商品	PV=PLC001:序号,寄存器,格式,地址,偏移:3002007-3-0-20-7
35	XZ04	选择04商品	PV=PLC001:序号,寄存器,格式,地址,偏移:3002008-3-0-20-8
36	XZ05	选择05商品	PV=PLC001:序号,寄存器,格式,地址,偏移:3002009-3-0-20-9
37	XZ06	选择06商品	PV=PLC001:序号,寄存器,格式,地址,偏移:3002010-3-0-20-10
38	XZ07	选择07商品	PV=PLC001:序号,寄存器,格式,地址,偏移:3002011-3-0-20-11
39	XZ08	选择08商品	PV=PLC001:序号,寄存器,格式,地址,偏移:3002012-3-0-20-12
40	KUANG	购物框	PV=PLC001:序号,寄存器,格式,地址,偏移:3002100-3-0-21-0

模拟 I/O 表

	NAME [点名]	DESC [说明]	%IOLINK [I/O连接]
1	JG01	01商品价格	PV=PLC001:序号,寄存器,格式,地址,偏移:3000400-3-3-4-0
2	JG02	02商品价格	PV=PLC001:序号,寄存器,格式,地址,偏移:3000500-3-3-5-0
3	JG03	03商品价格	PV=PLC001:序号,寄存器,格式,地址,偏移:3000600-3-3-6-0
4	JG04	04商品价格	PV=PLC001:序号,寄存器,格式,地址,偏移:3000700-3-3-7-0
5	JG05	05商品价格	PV=PLC001:序号,寄存器,格式,地址,偏移:3000800-3-3-8-0
6	JG06	06商品价格	PV=PLC001:序号,寄存器,格式,地址,偏移:3000900-3-3-9-0
7	JG07	07商品价格	PV=PLC001:序号,寄存器,格式,地址,偏移:3001000-3-3-10-0
8	JG08	08商品价格	PV=PLC001:序号,寄存器,格式,地址,偏移:3001100-3-3-11-0
9	POITR001	投入显示	PV=PLC001:序号,寄存器,格式,地址,偏移:3000100-3-3-1-0
10	POIXF002	消费显示	PV=PLC001:序号,寄存器,格式,地址,偏移:3000200-3-3-2-0
11	POIYE003	余额显示	PV=PLC001:序号,寄存器,格式,地址,偏移:3000300-3-3-3-0
12	TB$1	退币0.1元	PV=PLC001:序号,寄存器,格式,地址,偏移:9000400-9-3-4-0
13	TB$10	退币1元	PV=PLC001:序号,寄存器,格式,地址,偏移:9000200-9-3-2-0
14	TB$100	退币10元	PV=PLC001:序号,寄存器,格式,地址,偏移:9000000-9-3-0-0
15	TB$5	退币0.5元	PV=PLC001:序号,寄存器,格式,地址,偏移:9000300-9-3-3-0
16	TB$50	退币5元	PV=PLC001:序号,寄存器,格式,地址,偏移:9000100-9-3-1-0

3．运行

保存所有组态内容，然后关闭所有力控程序，包括： Draw 、 DbManager 等。

将自动售货机的 PLC 程序下传到 PLC 装置中并让其执行，切换到离线状态，然后再次启动力控工程管理器，选择本工程，并单击**“进入运行”**按钮启动整个运行系统。在运行中，可以按照实际自动售货机的功能来操作，以检验所编程序的正确与否。

第二节 五层楼电梯PLC控制与监控组态设计

一、电梯的基本功能

在进行上位机程序以及下位机程序编写之前，首先要做的工作是确定电梯本身所具有的功能和电梯在乘客进行某种操作后应具有的状态。

1．电梯内部部件功能简介

在电梯内部，应该有五个楼层（1-5层）按钮、开门和关门按钮以及楼层显示器、上升和下行显示器。当乘客进入电梯后，电梯内应该有能让乘客按下的代表其要去目的地的楼层按钮，称为内呼叫按钮。

电梯停下时，应具有开门、关门的功能，即电梯门可以自动打开，经过一定的延时后，又可自动关闭。而且，在电梯内部也应有控制电梯开门、关门的按钮，使乘客可以在电梯停下时随时地控制电梯的开门与关门。

电梯内部还应配有指示灯，用来显示电梯现在所处的状态，即电梯是上升还是下降以及电梯处在楼层的第几层，这样可以使电梯里的乘客清楚地知道自己所处的位置，离自己要到的楼层还有多远，电梯是上升还是下降等。

2. 电梯的外部部件功能简介

电梯的外部共分五层，每层都应该有呼叫按钮、呼叫指示灯、上升和下降指示灯，以及楼层显示器。

呼叫按钮是乘客用来发出呼叫的工具，呼叫指示灯在完成相应的呼叫请求之前应一直保持为亮，它和上升指示灯、下降指示灯、楼层显示器一样，都是用来显示电梯所处的状态的。

五层楼电梯中，一层只有上呼叫按钮，五层只有下呼叫按钮，其余三层都同时具有上呼叫和下呼叫按钮。而上升、下降指示灯以及楼层显示器应相同。

3．电梯的初始状态、运行中状态和运行后状态分析

1）电梯的初始状态：设电梯位于一层待命，各层显示器都被初始化，电梯处于以下状态：

a. 各层呼叫灯均不亮；

b. 电梯内部及外部各楼层显示器显示均为“1”；

c. 电梯内部及外部各层电梯门均关。

2）电梯在运行过程中：

a. 按下某层呼叫按钮（1-5层）后，该层呼叫灯亮，电梯响应该层呼叫；

b. 电梯上行或下行直至该层；

c. 各楼层显示随电梯移动而改变，各层指示灯也随之而变；
d. 运行中电梯门始终关闭，到达指定层时，门才打开；
e. 在电梯运行过程中，支持其它呼叫。
3) 电梯运行后状态：在到达指定楼层后，电梯会继续待命，直至新命令产生。
a. 电梯在到达指定楼层后，电梯门会自动打开，经一段延时自动关闭，在此过程中，支持手动开门或关门；
b. 各楼层显示值为该层所在位置，且上行与下行指示灯均灭 。

二、 实际运行中的情况分析

1. 分类分析

1) 电梯上行分析：

若电梯在上行过程中，某楼层有呼叫产生时，可分以下两种情况：

a. 若呼叫层处于电梯当前运行层之上目标运行层之下，则电梯在完成前一指令之前先上行至该层，完成该层呼叫后再由近至远的完成其它各个呼叫操作；

b. 呼叫层处于电梯当前运行层之下，则电梯在完成前一指令之前不响应该指令，直至电梯重新进入待命状态为止。

2）电梯下行分析：

若电梯在下行过程中，楼层有呼叫产生时，分以下两种情况：

a. 若呼叫层处于电梯当前运行层之下目标运行层之上，则电梯应在完成前一指令之前先下行至该层，完成该层呼叫后再由近至远地完成其它各个呼叫动作；

b. 若呼叫层处于电梯运行层之上，则电梯在完成前一指令之前不响应该指令，直至电梯重新处于待命状态为止。

2. 总结规律

由以上各种分析可以看出，电梯在接受指令后，总是由近至远地完成各个呼叫任务。电梯机制只要依此原则进行设计动作，就不会在运行时出现电梯上下乱跑的情况了。

界面

左半部分是电梯的内视图，其中包含一个楼层显示灯、开门按钮、关门按钮、一到五层的呼叫按钮以及电梯的上升和下降状态指示灯等。两扇电梯门打开后可以看到楼道的景象。

右半部分是五层楼宇电梯的外视图，表示五层楼宇和一个电梯的轿箱。在电梯的外视图中，一层有一个上呼叫按钮五层有一个下呼叫按钮，二、三和四层有上、下呼叫按钮各一个，每个呼叫按钮内都有一个相应的指示灯，用来表示该呼叫是否得到响应。轿箱的电梯门和每层的楼层门都可以打开。

3．仿真电梯的控制要求

1）接受每个呼叫按钮（包括内部和外部的呼叫）的呼叫命令，并作出相应的响应。

2）电梯停在某一层（例如3层）时，此时按动该层（3层）的呼叫按钮（上呼叫或下呼叫），则相当于发出打开电梯门命令，进行开门的动作过程；若此时电梯的轿箱不在该层（在1、2、4、5层），则等到电梯关门后，按照不换向原则控制电梯向上或向下运行。

3）电梯运行的不换向原则是指电梯优先响应不改变现在电梯运行方向的呼叫，直到这些命令全部响应完毕后才响应使电梯反方向运行的呼叫。例如现在电梯的位置在一层和二层之间上行，此时出现了一层上呼叫、二层下呼叫和三层上呼叫，则电梯首先响应三层上呼叫，然后再依次响应二层下呼叫和一层上呼叫。

4）电梯在每一层都有一个行程开关，当电梯碰到某层的行程开关时，表示电梯已经到达该层。

5）当按动某个呼叫按钮后，相应的呼叫指示灯亮并保持，直到电梯响应该呼叫为止。

6）当电梯停在某层时，在电梯内部按动开门按钮，则电梯门打开，按动电梯内部的关门按钮，则电梯门关闭。**但在电梯行进期间电梯门是不能被打开的。**

7）当电梯运行到某层后，相应的楼层指示灯亮，当电梯运行到前方一层时楼层指示灯灭。

三、设计部分

首先，应该做上位机与下位机之间的任务分工：

上位机主要用来完成仿真界面的制作及动画连接工作。

下位机则主要用来完成 PLC 程序的运行。

其实，上位机与下位机的设计工作是密切配合的。它们无论在通讯中使用的变量，还是在进行界面仿真时控制的对象都应该是一致的。总体上讲，仿真界面是被控对象，PLC 是存储运行程序的装置，而控制指令则由仿真界面中的仿真控制器件发出。另一方面，仿真界面中仿真电梯的运动，门的运动等，都是由力控组态软件所提供的命令语言来完成的。

1. PLC 程序中 I/O 点的定义

在编程过程中，所用到的 I/O 地址分配如下表所示。编程过程可分为电梯内部和电梯外部两部分进行。

I/O 分配表

说明	对应 PLC 地址	说明	对应 PLC 地址
外部一层上呼叫按钮	R101	外部一层上呼叫灯	Y1
外部二层上呼叫按钮	R102	外部二层上呼叫灯	Y2
外部二层下呼叫按钮	R103	外部二层下呼叫灯	Y3
外部三层上呼叫按钮	R104	外部三层上呼叫灯	Y4
外部三层下呼叫按钮	R105	外部三层下呼叫灯	Y5
外部四层上呼叫按钮	R106	外部四层上呼叫灯	Y6
外部四层下呼叫按钮	R107	外部四层下呼叫灯	Y7
外部五层下呼叫按钮	R108	外部五层下呼叫灯	Y8
一层行程开关	R109	一层位灯	Y9
二层行程开关	R10A	二层位灯	YA
三层行程开关	R10B	三层位灯	YB
四层行程开关	R10C	四层位灯	YC
五层行程开关	R10D	五层位灯	YD

（续上表）

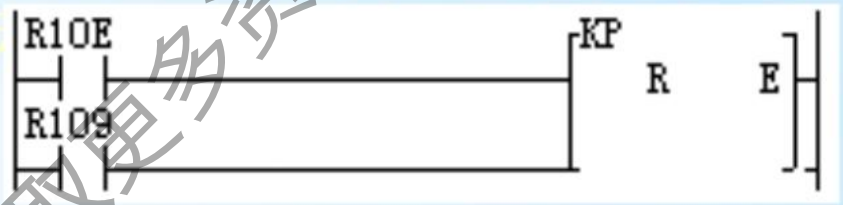
内部一层呼叫按钮	R10E	电梯上升	YE
内部二层呼叫按钮	R10F	电梯下降	YF
内部三层呼叫按钮	R110	上升指示灯	Y10
内部四层呼叫按钮	R111	下降指示灯	Y11
内部五层呼叫按钮	R112	电梯开门	Y12
开门呼叫按钮	R113	电梯关门	Y13
关门呼叫按钮	R114		
开门行程开关	R115		
关门行程开关	R116		
内部一层呼叫灯	RE		
内部二层呼叫灯	RF		
内部三层呼叫灯	R10		
内部四层呼叫灯	R11		
内部五层呼叫灯	R12		

2. 电梯内部的PLC编程

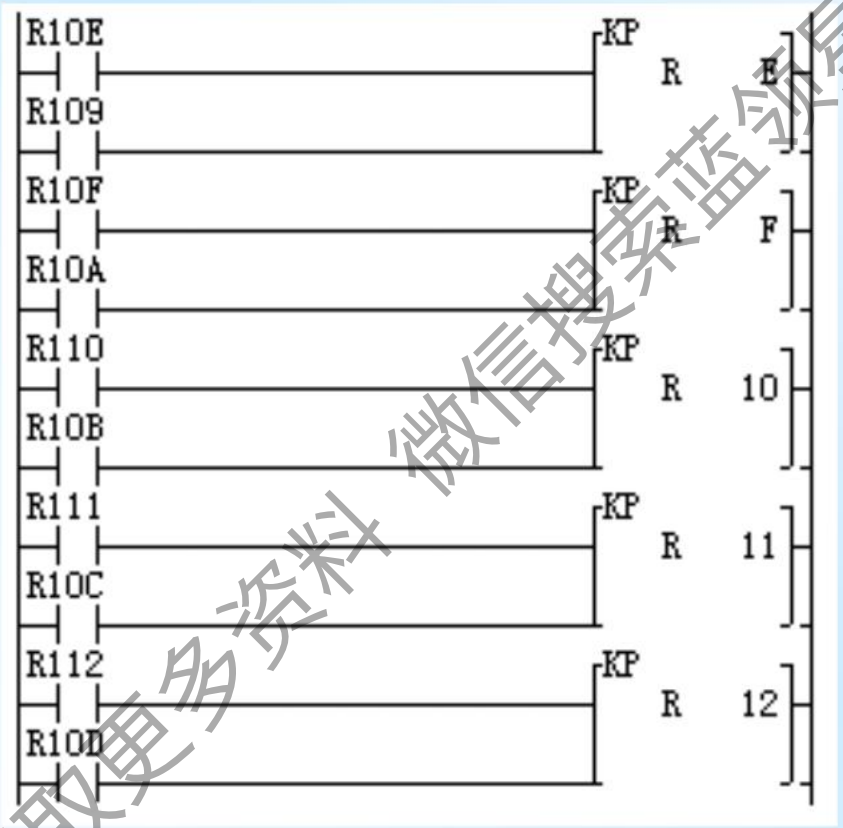
1）五层楼的内呼叫灯 PLC 程序

电梯内部的五个呼叫按钮，指定的是电梯的运行目标。因此在电梯未运行到指定目标时，该层呼叫灯应一直有显示（为绿），因此输出时就应该使用保持继电器。另外，当电梯达到指定楼层时，呼叫灯应该灭掉，即保持继电器断开。

先以一层的呼叫灯为例，所得的程序如下图所示：



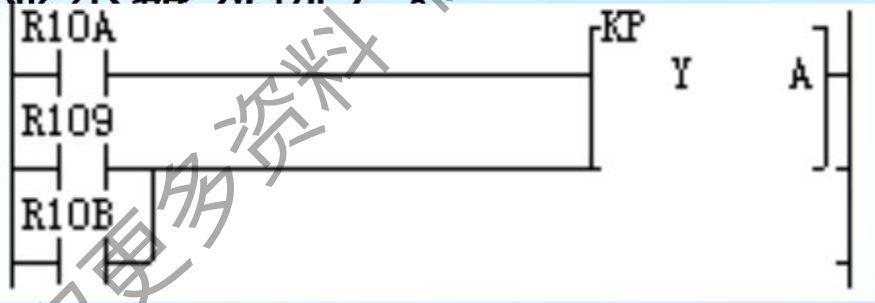
五层楼内呼叫灯的 PLC 梯形图程序如下图：



2）电梯内的楼层显示器

楼层显示器是以电梯是否碰到行程开关来决定的。显示器同样有保持特性。另外要替换某一显示器的值，需要电梯接触到其上层或下层的行程开关。

综合以上因素可得程序如下图所示（以第二层显示器为例）。

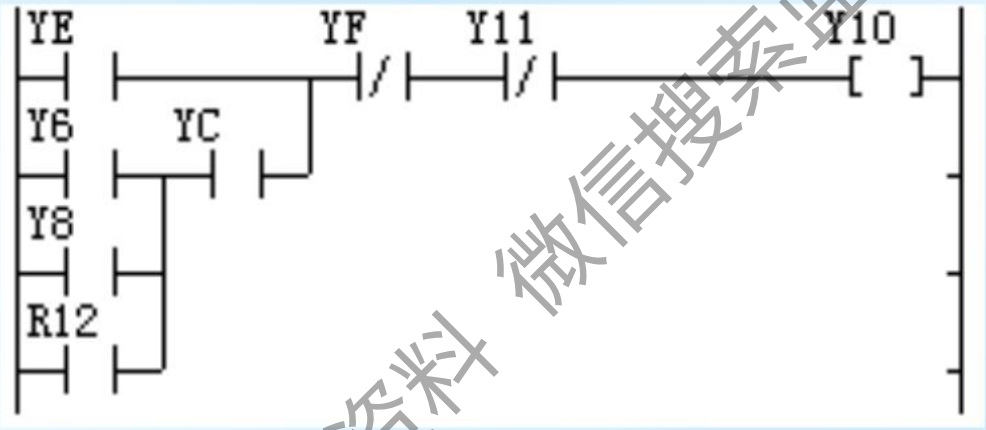


3）电梯内的电梯升降显示器

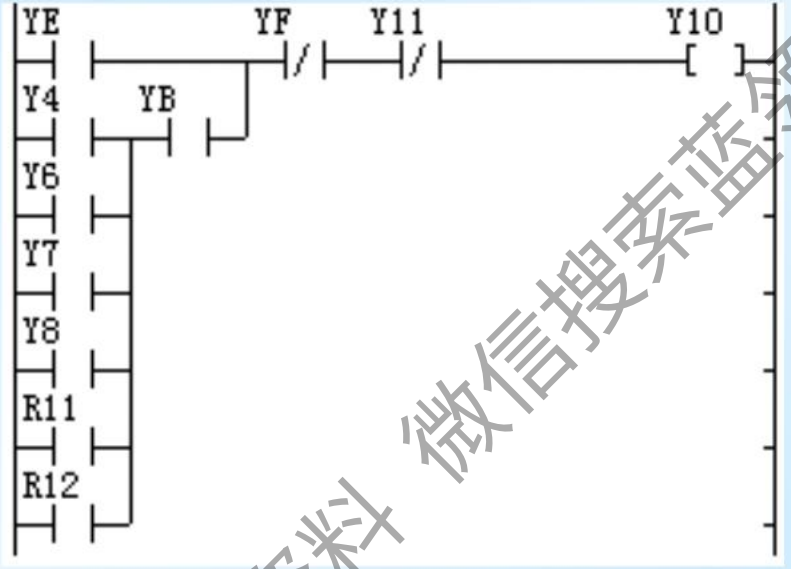
升降显示器的状态共三种：显示上升、显示下降、或都不显示。另外，无论上升还是下降，都与电梯的呼叫有密切关系。上升包括了从第一层到第五层的上升运动，下降也同样如此。因此程序应从最基本、简单的过程入手。现以电梯从第四层到第五层的上升为例。

若五层有呼叫，包括两种情况：电梯内呼叫、电梯外呼叫。若电梯由第四层上行至第五层，此时 Y10 亮，Y11 灭，下降触点 YF 断开，上升触点 YE 闭合。并列的条件还有四层外部上呼叫闭合、五层下呼叫闭合、五层呼叫按钮的闭合，这些条件同样使得四层显示器改变，因此在编程时都要考虑。

四层升至五层的升降显示器 PLC 程序如下图：



由三层升至五层的升降显示器 PLC 程序如下图：



由上面的分析可以看出，整个电梯上升显示程序即是对各层的上升程序取程序块并联逻辑操作。

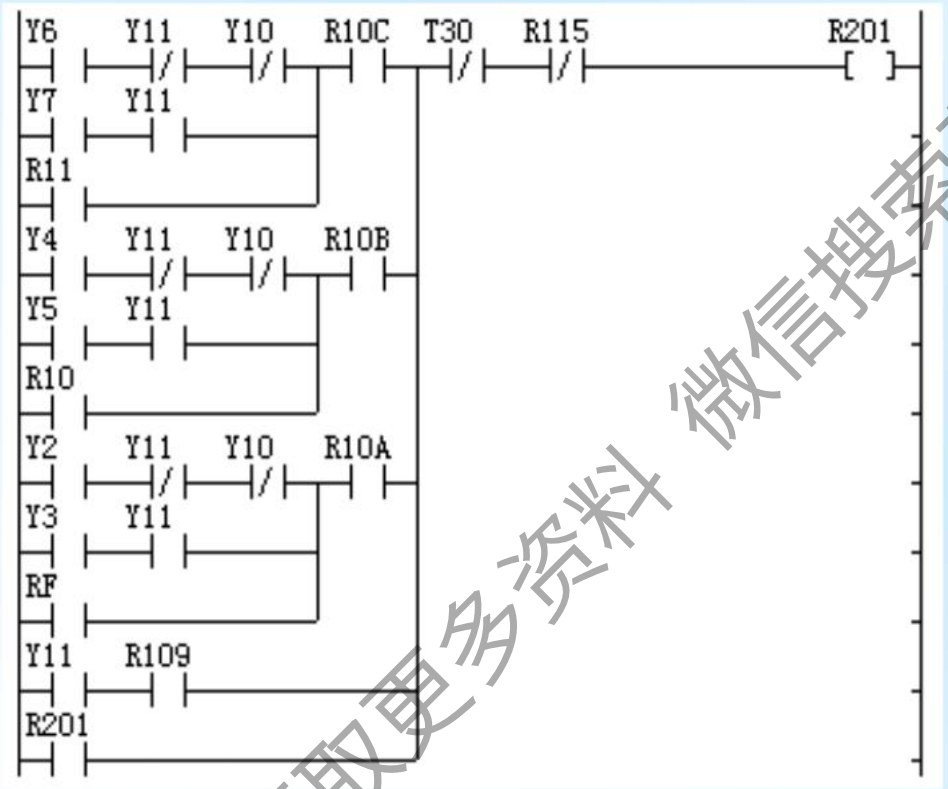
电梯下降指示灯的程序编写方法与上升指示灯的编写方法是一样的。

3. 电梯到达楼层后的停止

由于在电梯外部有上升呼叫和下降呼叫，所以**当呼叫方向与电梯运行方向相同时，电梯才能停止。**

下面以向下呼叫停止 **R201** 为例说明。而上升呼叫停止 **R200** 的编程思路与下降呼叫停止 **R201** 相似。

电梯到达呼叫楼层后停止的 PLC 程序如下图所示。



当电梯一直不向下运行，即 Y11 接通时，只有当每层楼的向下呼叫即 Y3、Y5、Y7 和该层的行程开关接通时，电梯到达各层后 R201 才接通，否则并不接通。

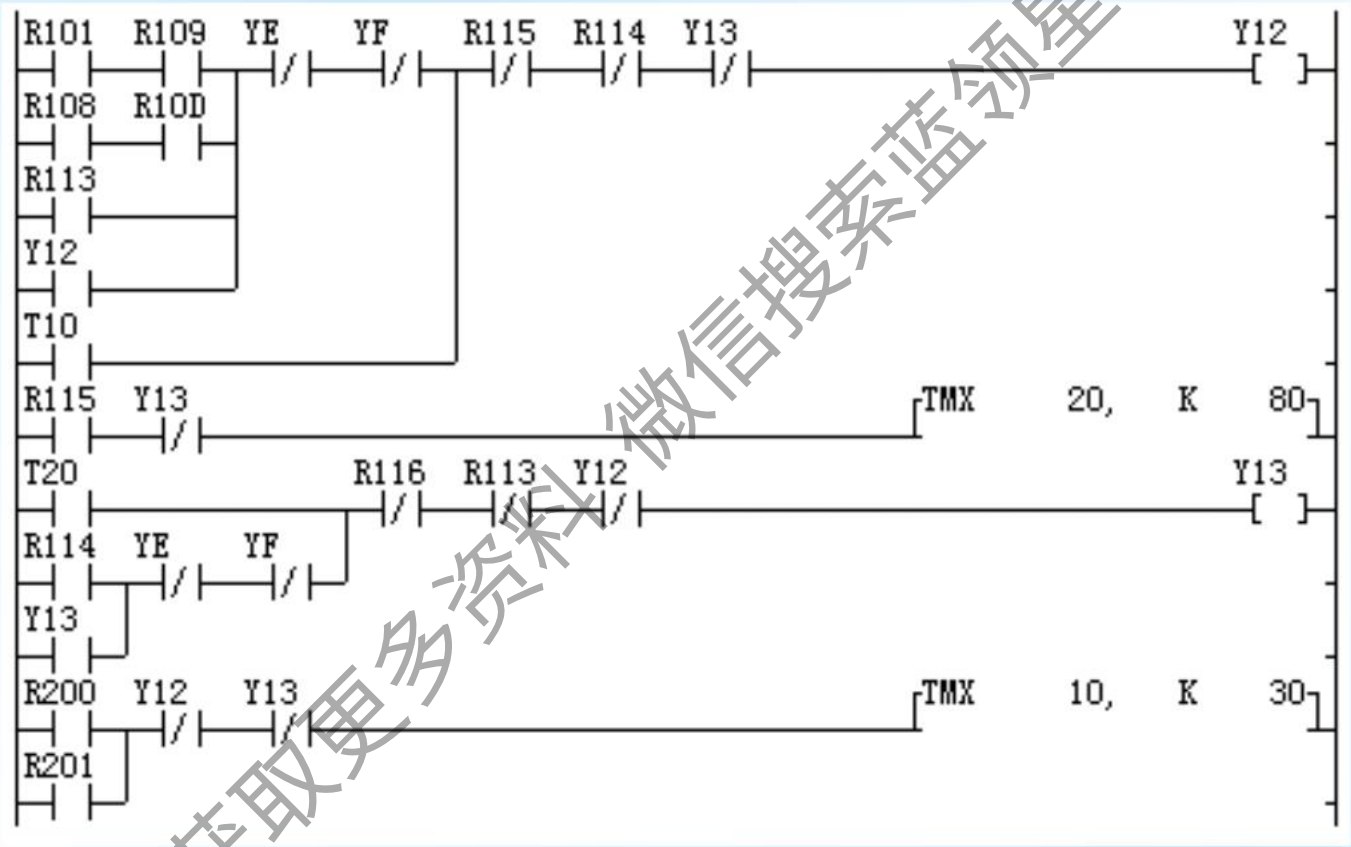
另外，如果电梯一直停在某一层，不显示上升与下降，该层的上呼叫接通时 R201 也接通。

4. 电梯的开、关门程序

以电梯的开门 Y12 程序为例说明。

首先，只有当电梯既不上升也不下降时才能进行开门，即 Y12 才能输出。无论电梯的上升停止或下降停止，只要 R200 或 R201 有一个输出，电梯经过延时后都会自动开门。如果电梯停在某一层，则该层的外部呼叫也会输出 R200 或 R201，电梯门也会打开。当开门 Y12 输出时，关门 Y13 断开。如果关门 Y13 接通时，Y12 也应立即断开。

电梯开门的 PLC 梯形图程序如下图：

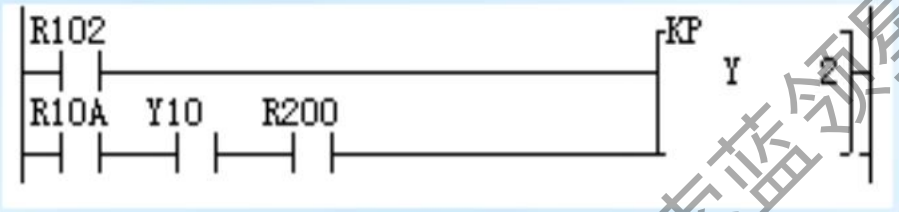


5．电梯外部操作与显示的PLC程序

1）外部呼叫

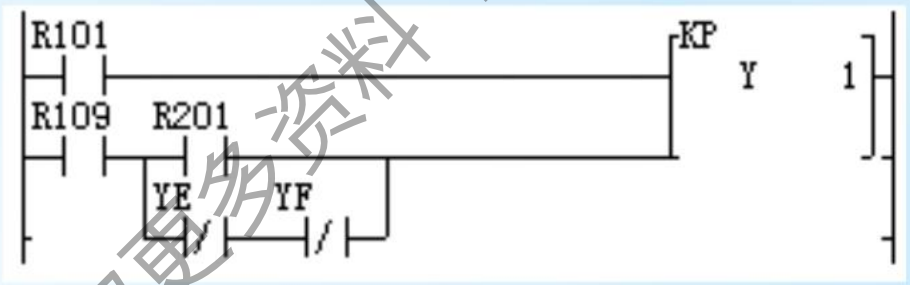
电梯外部呼叫与内部呼叫类似的。一个呼叫灯的接通对应于相应的外部呼叫接通，其关闭条件为相应行程开关闭合，相应升降指示灯闭合。另外，外部呼叫同样有保持特性，故也应使用保持继电器作为输出。需要指出的是第一、第五层外呼叫是单向的，故其关闭条件也相应变为行程开关闭合“AND”电梯升降断开。

a. 2层－4层以2层为例：



2层电梯外部上呼叫梯形图

b. 1层、5层以1层为例：



1层电梯外部呼叫梯形图

下呼叫与上呼叫类似，不同之处在于关闭条件中的 Y10（上升指示）应换为 Y11（下降指示）。

2）楼层的电梯位置指示灯

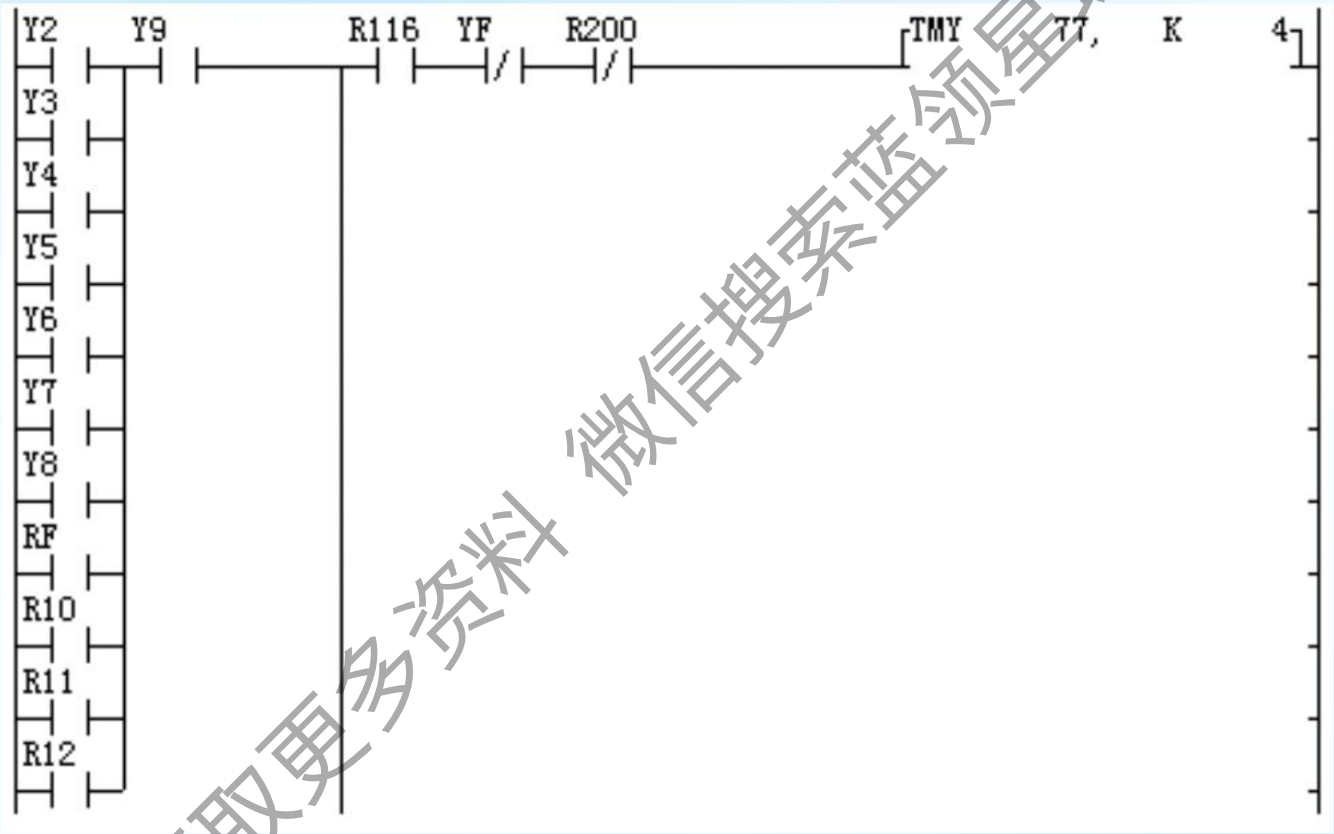
外部楼层的电梯位置指示灯与电梯内的位置指示灯相同。

3）电梯的上升与下降

设电梯从一层到五层上升。

由于电梯在第一层，此时构成电梯上升的因素包括：二层上 / 下呼叫、三层上 / 下呼叫、四层上 / 下呼叫、五层下呼叫、内二层呼叫灯、内三层呼叫灯、内四层呼叫灯、内五层呼叫灯，这 11 种条件对于电梯的上升是一种逻辑“OR”的关系，而这些条件的产生的前提条件则是电梯此时在一层，即一层位灯 Y9 有输出。可见，Y9 与前 11 个“ OR”逻辑是“ AND”的关系。

电梯从一层到五层上升的梯形图如下图：



电梯由二层上升到五层、三层上升到五层、四层上升到五层的编程思路与一层上升到五层相似，不同之处为上升条件同某层位灯逻辑“AND”运算之后还应排除电梯下降指示的情况，这里将电梯下降指示 Y11 的常闭触点与之串联。另外由于电梯上升，电梯门应关门，电梯下降线圈 YF 应断电，这又是和以上程序块取逻辑“AND”与的过程。

电梯下降程序的编写方法与上升程序的编写方法相似。

需要指出的是，**电梯上升与下降都是建立在开门和关门继电器线圈不接通的情况下。**因此，在电梯上升与下降的过程中要将这两个因素考虑在内。

四、五层楼电梯PLC控制参考程序

五层楼电梯仿真系统的PLC控制参考程序如教材图7-32所示。由于篇幅的限制，有关五层楼电梯仿真系统界面的制作、脚本程序的编写、仿真系统的运行过程以及系统编程时易出现的问题和解决办法在这儿就不详细地叙述了。读者可把本书配套光盘中的应用程序“五层楼电梯”拷贝到自己的计算机中运行，通过实际操作来了解电梯的基本功能，仔细分析该仿真系统的设计过程，从中学习利用监控组态软件进行PLC系统设计的方法和技巧。