

NAIS

WELCOME TO NAIS

PLC基础培训

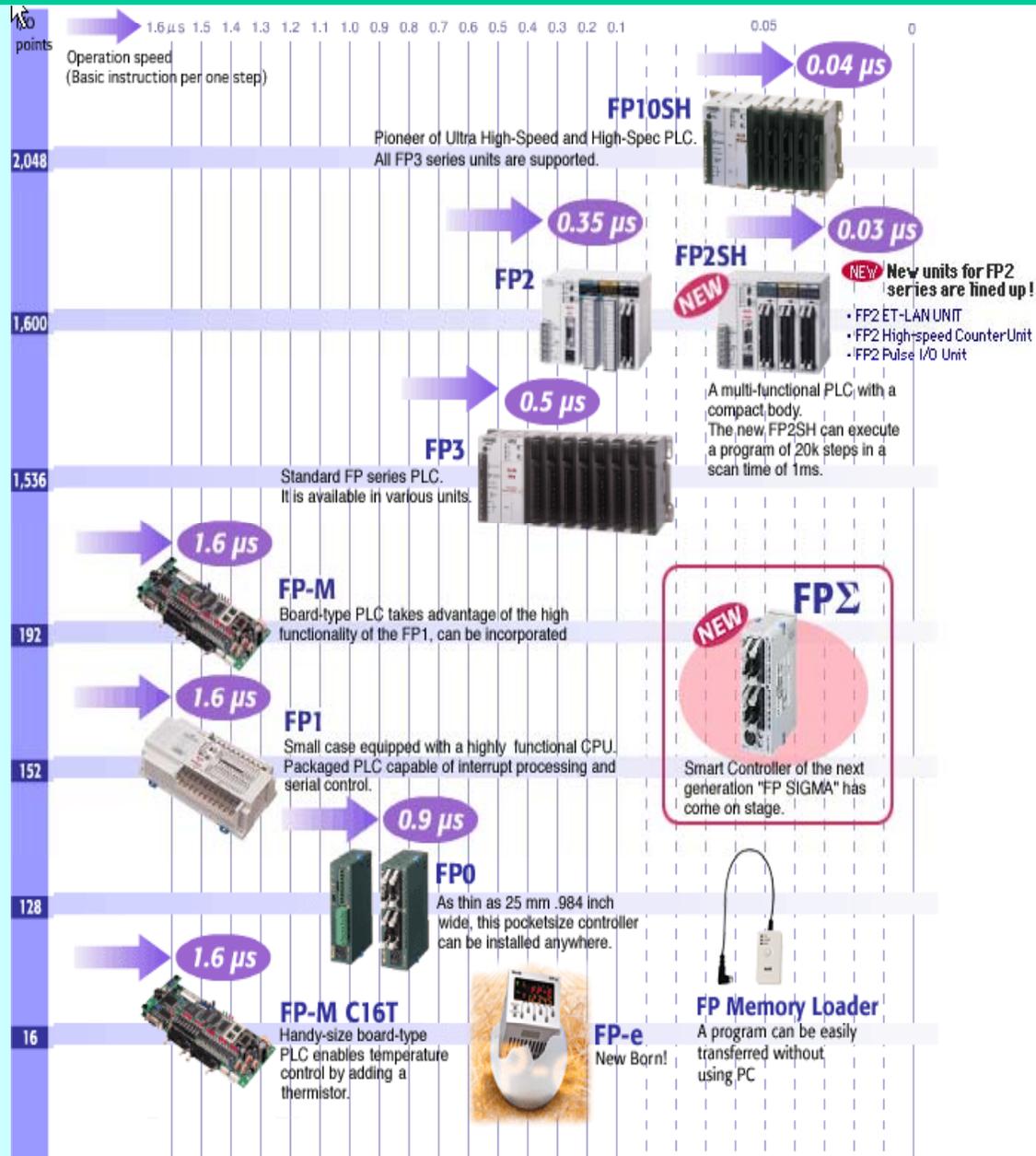
PLC SEMINAR

初学者也能得心应手
简便编程！

松下电工(中国)有限公司
制御EC

Smart Solutions by **NAIS**

松下PLC系列简介



【目录】

【第1章 PLC入门】

- 1-1 什么是可编程控制器 (PLC)
- 1-2 顺序控制器的种类及变迁
- 1-3 如何选择PLC的机型

【第2章 可编程控制器的构成】

- 2-1 PLC内部的构成要素
- 2-2 PLC的动作原理
- 2-3 各部名称及功能
- 2-4 PLC的输入输出部
- 2-5 PLC的内部继电器一览表
- 2-6 PLC的编程工具
- 2-7 编程工具的操作菜单
- 2-8 培训模型的输入输出分配

【第3章 编程的基础知识】

- 3-1 PLC的回路图
- 3-2 梯形图的阅读方法
- 3-3 基本指令
- 3-4 编程时的注意事项
- 3-5 编程错误一览表

【第4章 PLC的基本回路】

- 4-1 自保持回路
- 4-2 自保持回路的改进①
- 4-3 微分 (DF) 指令
- 4-4 自保持回路的改进②
- 4-5 **步进跟踪编程法**的自保持回路
- 4-6 定时器 (TM) 指令
- 4-7 定时器应用回路

【第5章 编程实践】

- 5-1 一般的输出控制
- 5-2 利用符号梯形图方式编写程序
- 5-3 利用**步进跟踪编程法**控制输出
- 5-4 实践**步进跟踪编程法** 绘制时序图
- 5-5 实践**步进跟踪编程法** 编写梯形图
- 5-6 挑战课题 - 第1工程、第2工程
- 5-7 挑战完成课题
- 5-8 自动 手动切换回路
- 5-9 编程建议

【第6章 便利指令介绍】

- 6-1 SET·RST指令

【第7章 PLC基础教程练习题】

【前言】

可编程控制器(Programmable Controller)于1968年在美国首次登场,是用于自动控制的控制器。与当时作为控制领域的主流利用继电器的自动控制方式相比,具有

- ◆ 易于编写、修改程序。
- ◆ 高度的控制性能。
- ◆ 无触点、长寿命。
- ◆ 高可靠性。

等多项良好特性,因此迅速得到推广普及。

最近,随着半导体技术的飞速发展,可编程控制器更加

- ◆ 小型化。
- ◆ 高性能化。
- ◆ 低价格化。

现在,PLC的应用领域早已不仅限于生产设备,在楼宇自动化、列车、汽车、自动售货机、停车场管理、水库控制等各种领域,PLC也都得到越来越广泛的应用。

本培训的目标是让任何人都能得心应手,简单方便地掌握这个承担着自动控制核心任务的PLC的基础知识。

祝愿大家活用本教材、迈入自动控制领域,在这个永无止境的、无限广阔的“最刺激的世界”里大显身手、尽展才华。



技成培训网
www.jichengpeixun.com

第1章 PLC入门

【1-1. 什么是可编程控制器？】

可编程控制器是顺序控制专用的控制器，正式英文名称是Programmable Controller (简称PC)。在中国为了与个人计算机相区别，简称为PLC(沿用旧称Programmable Logic Controller)。

可编程控制器(以下简称PLC)将来自输入设备的信号，按照给定的条件进行处理、运算、判断并将该结果输出到外部设备。

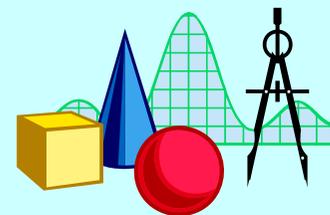
在PLC诞生之前，自动控制是利用继电器、定时器等组合实现的。

因此，伴随着控制内容的改变，必须花费很多时间进行配线施工，在实际应用中存在很多缺点。

与计算机控制相比，PLC最大的不同点是

- ◆ 配备有丰富的顺序控制专用指令
- ◆ 通过专用指令能够方便地编制程序
- ◆ 高速度重复循环执行程序(扫描)
- ◆ 充分考虑到恶劣的使用环境，抗干扰能力强

由于上述原因，PLC作为自动控制用控制器当前最为普及。

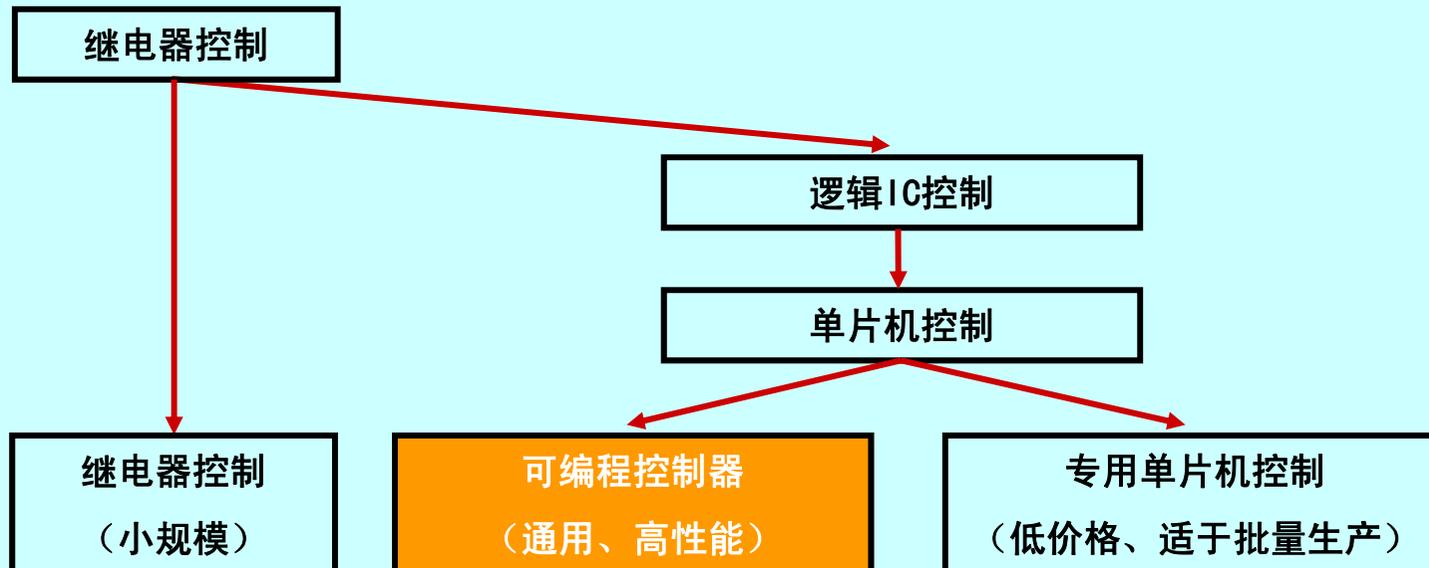


【1-2. 顺序控制器的种类及变迁】

【区分】

	优点	缺点
继电器控制	用于小规模、简单控制时价格低 抗干扰能力强	需要动作确认 控制内容修改困难
PLC 控制 可编程控制器	适于小~大规模、高性能、通用性好 程序修改简便 快速投入使用	大量生产时与专用控制器相比价格较高
专用电路 单片机等	大量生产时价格低	需要开发时间和技术 程序修改困难

【变迁】



【1-3. PLC的机型选择】

PC在引入PLC时，应根据控制对象选择PLC的型号，但是在选择时请注意以下几点：

【PLC机型选择要点】

最重要

◆ 控制规模(I/O点数)

根据不同的控制规模，必要的输入输出点数(I/O)不同。

但是也应考虑到将来的改造要求、保留有适当的余量，再计算必要的输入输出点数、选择最佳的机型。如果点数不足，则无法进行控制。

请注意千万不要导致点数不足。

FP0 32点型： 输入16点 + 输出16点 = 合计32点

： 最大点数128点

◆ 指令处理速度

指令的处理速度是决定程序处理时间(机械的速度)的重要因素。

当编写比较长、比较复杂的程序时请注意处理速度。

(但是近年的PLC已经被高速化，用于通常用途时基本不存在处理速度的问题。)

FP系列最新机型 FP2SH的基本指令的处理速度为30纳秒

30纳秒 = 0. 000 000 030 秒
 msec μ sec nsec

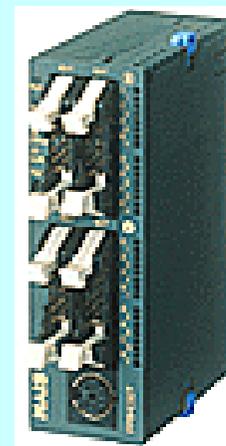
◆ 程序容量

PLC的程序容量以[步(step)]为单位表示。

程序是决定机械动作的重要因素。所必须的程序随控制内容的不同而不同，但至少需要输入输出点数100倍左右的程序容量。

FP0 32点型 = 5000步

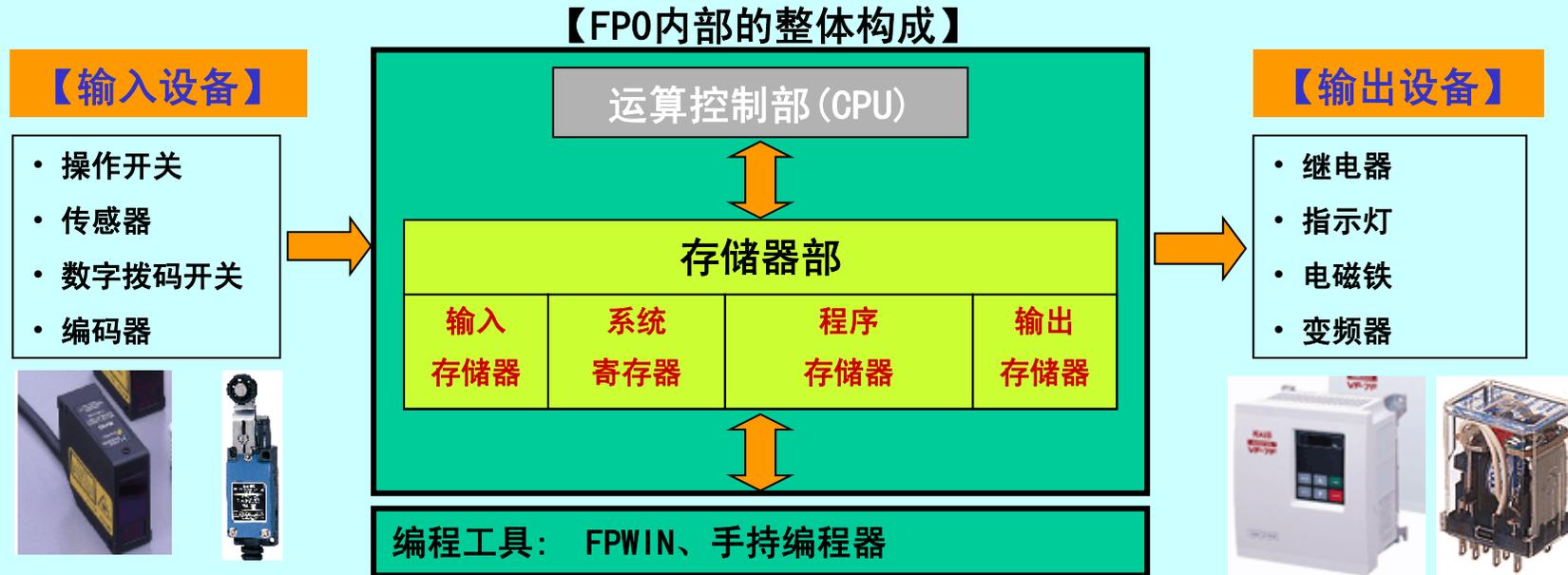
香烟盒大小的
超小型PLC



技成培训网
www.jichengpeixun.com

第2章 可编程控制器的构成

【2-1. PLC内部的构成要素】



【运算控制部 CPU】

按照程序对输入输出进行控制

【存储器部】

存储程序及运算所必需的信息

【输入存储器】

保存输入设备的ON、OFF状态

【输出存储器】

保存运算结果的输出状态

【程序存储器】

保存用户编写的程序

利用FPWIN或手持编程器等编程工具写入程序

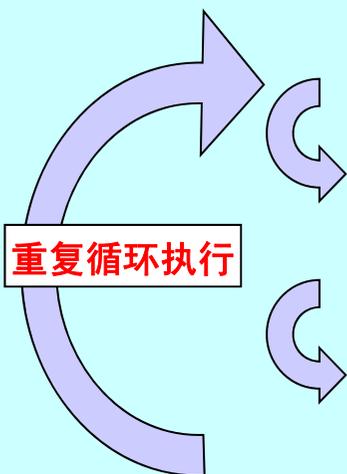
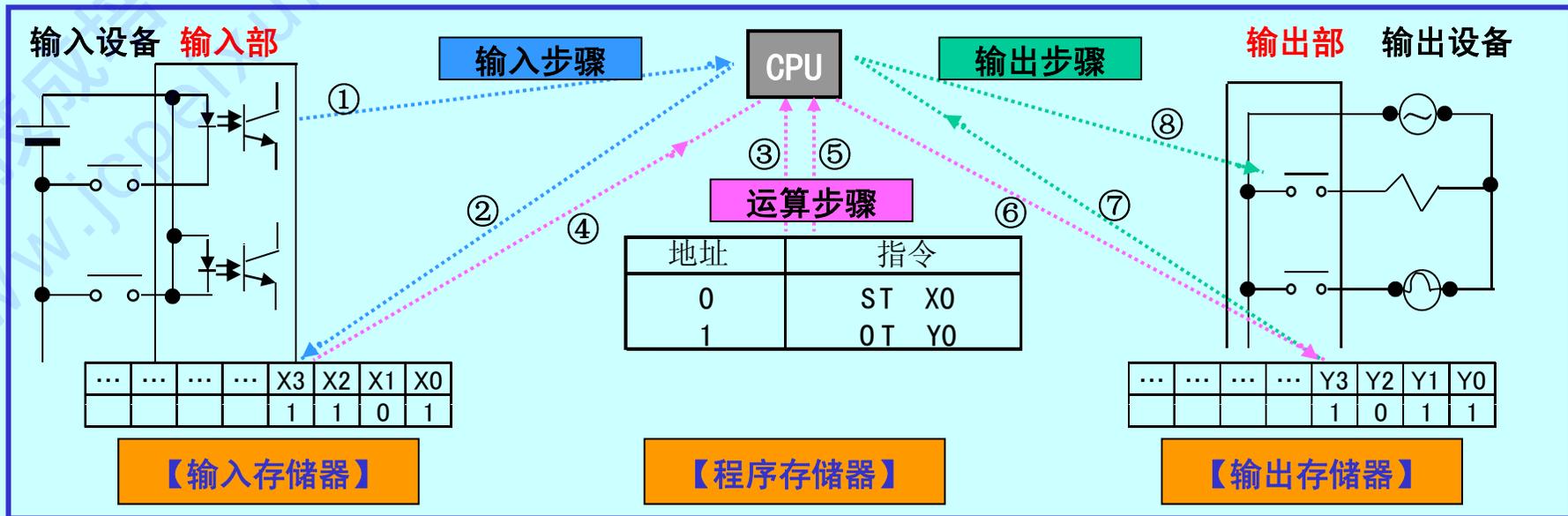
【系统寄存器】

决定PLC运行环境的部分

作为程序的一部分、在传输程序时随程序一起被写入PLC的存储器

【2-2. PLC的动作原理】

从PLC的输入开始到输出位置的处理流程如下所示：



【何谓扫描时间？】

PLC按照输入步骤、运算步骤、输出步骤不断循环反复执行程序。每1周期的处理时间被称为扫描时间。

【1 扫描时间】

设想在替换继电器电路的情况下，一般达到10msec以内即可认为是理想情况

【2-3. 各部名称及功能】

【FP0 C32 控制单元】

① 状态显示LED

表示PLC的运行/停止、错误/报警等动作状态

② 输入部

使用连接端子与输入设备相连

③ 模式切换开关

切换PLC的运行模式

④ 输出部

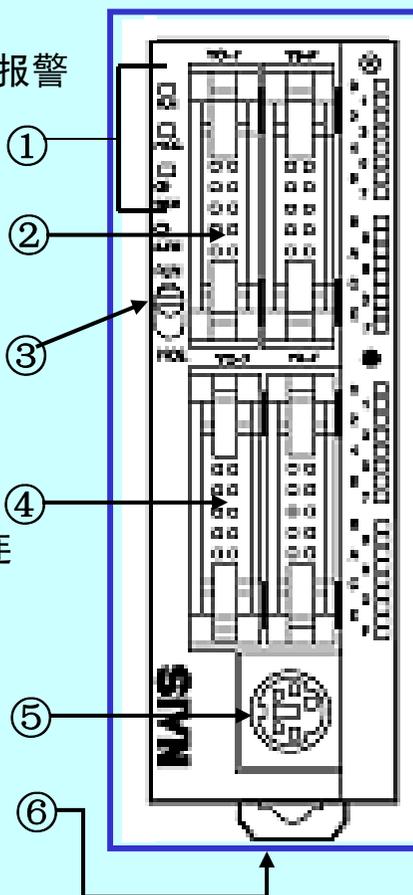
使用连接端子与输出设备相连

⑤ 编程口

用于与编程工具相连

⑥ 电源部

提供DC24V电源



【模式切换开关】

开关位置	动作模式
RUN (上)	处于RUN模式 开始执行程序、运行。
PROG (下)	处于PROG模式 停止运行。

【状态显示LED】

RUN (绿)	在RUN模式下、 执行强制输入输出时闪烁
PROG (绿)	在PROG模式下亮 表示运行停止状态
ERROR/ ALARM (红)	如果检测到自检错误则闪烁。发生硬件异常以及程序运算停滞或者看门狗动作时开始闪烁。

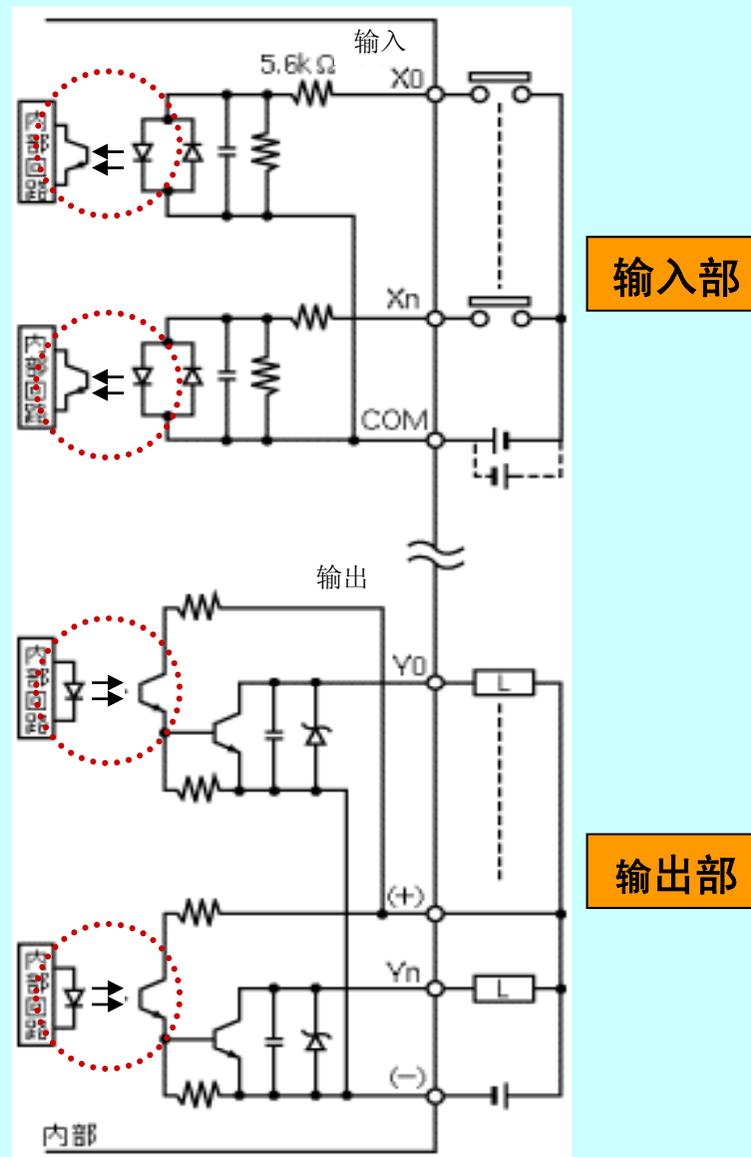
【2-4. PLC的输入输出部分】

◆ 输入部

由电子回路构成，用于连接操作开关、传感器等输入设备，把信号读入PLC内部。为防止外部干扰信号侵入输入部分，采用光电耦合器保护。

◆ 输出部

与输入设备相同、由电子回路构成，用于连接电机、变频器、显示器等，向外部输出信号。输出部也与输入部一样，为防止外部干扰信号侵入，采用光电耦合器保护。



【2-5. PLC的内部继电器一览表】

◆继电器分类

PLC中所使用的继电器，按功能和类型分为不同类型。

【例：FP-0 32点型控制单元】

名称		使用点数	功能
外部输入	X	208点 (X0~X12F)	按来自外部的输入进行ON/OFF
外部输出	Y	208点 (Y0~Y12F)	向外部输出ON/OFF状态
内部继电器	R	1008点 (R0~R62F)	仅在程序中ON/OFF
定时器	T	定时器、计数器合计为144点 (T0~T99/C100~C143)	到达定时器的设定时间后变为ON
计数器	C		计数器计数到时为ON
特殊内部继电器	R	64点 (R9000~)	特定条件下ON/OFF，即具有某种意义的特殊继电器

◆继电器序号的规定

X、Y、R的编号、用10进制和16进制的组和来表示。（因为经常是把16点作为一组来处理）

T、C接点时，仅用10进制来表示。

【外部输入 (X) 时】

X□□□ X0、X1……XF

【10进制】

【16进制】

【定时器 (T) 时】

T□□

【10进制】

【2-6. PLC的编程工具】

松下电工向用户提供以下两种编写、编辑、调试PLC程序的专用工具。

◆ 编程器

【手持编程器 II】



【特点】

1. 小型便携
2. 便于修改部分程序
3. 仅监控显示指令

松下电工PLC

【FP系列】

◆ 使用计算机的编程工具

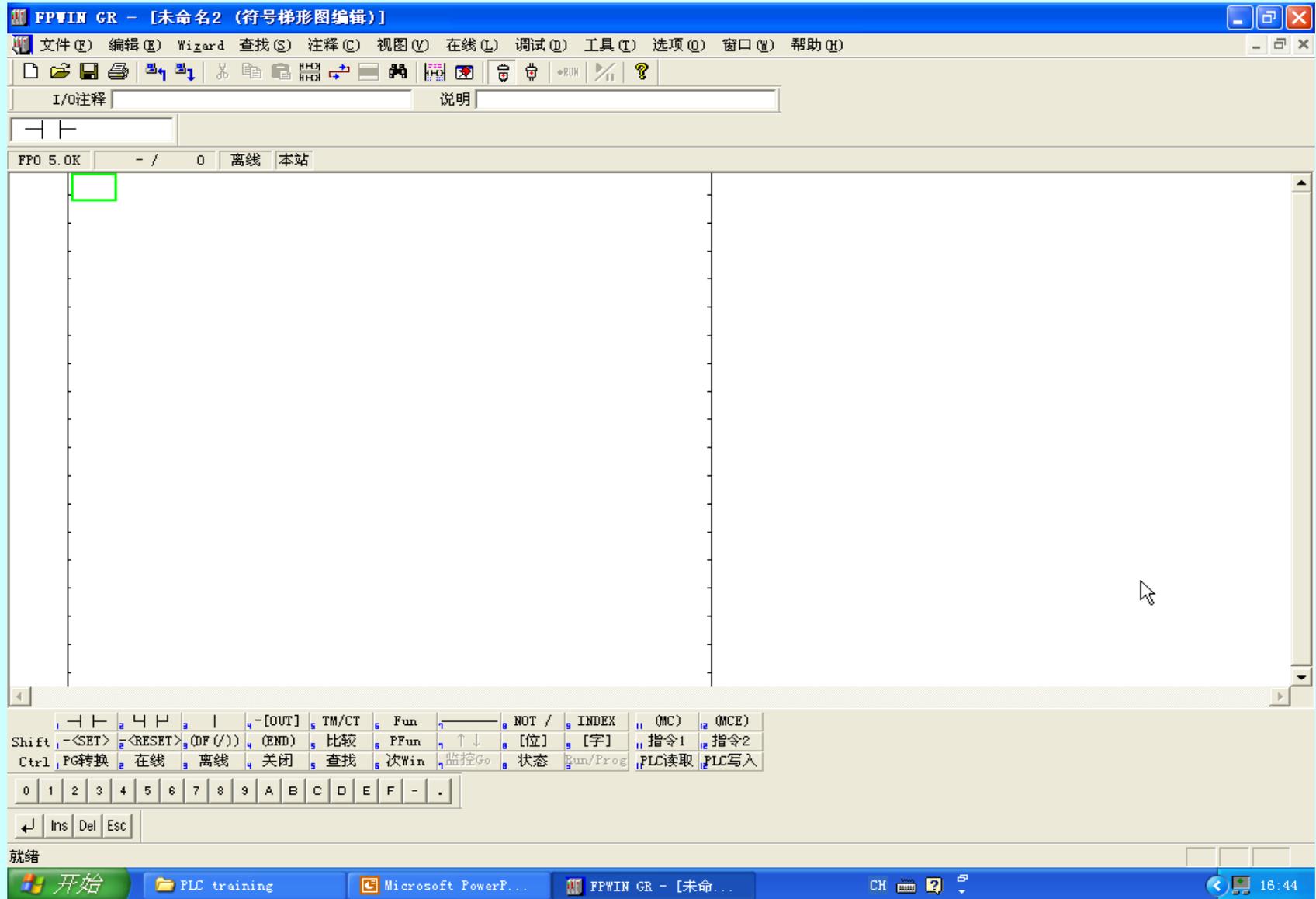
【FPWIN-GR】等



【特点】

1. 易于编辑调试
(计算机显示画面大)
2. 监控功能强
3. 易于编写梯形图

松下编程界面



【2-7. 编程工具的操作菜单】

◆ FPWIN的工具栏功能一览

将使用频率较高的指令作成了图标按钮（图例：布尔梯形图编辑方式）



新编写文件

打开文件

保存文件

打印输出

读出程序

写入程序

设备注释检索

注释显示切换

在线编辑方式

离线编辑方式

动作模式切换

监控开始/停止

参照帮助文件

【功能随模式不同变化而变化】

不可选择时以灰色显示

【从3种方式中选择】

菜单

视图(V)

布尔梯形图编辑 (BLD)

- ① 符号梯形图编辑方式
- ② 布尔梯形图编辑方式
- ③ 布尔形式编辑方式



用于输入的功能键的功能发生变化

	ST	OR	AND	OUT	TM/CT	Fun	STK	NOT /	INDEX	MC	MCE
Shift	SET	RESET	DF (/)	END	比较	PFun	↑↓	[位]	[字]	指令1	指令2
Ctrl	PG转换	在线	离线	关闭	查找	次Win	监控Go	状态	Run/Prog	PLC读取	PLC写入

【2-8. 培训模型的输入输出分配】

- ◆ 培训用模型与FP0 32点型的控制单元相连，其输入输出序号按下述方法分配

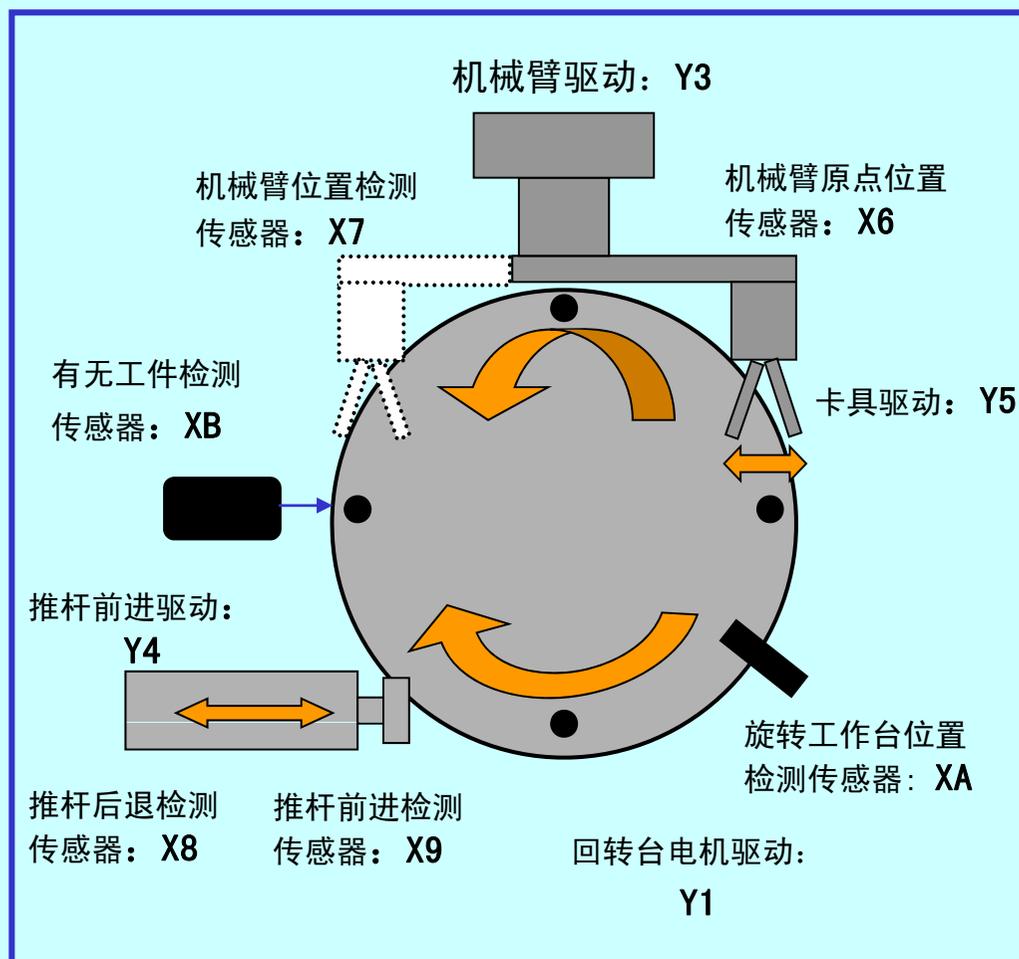
◆ 输入分配

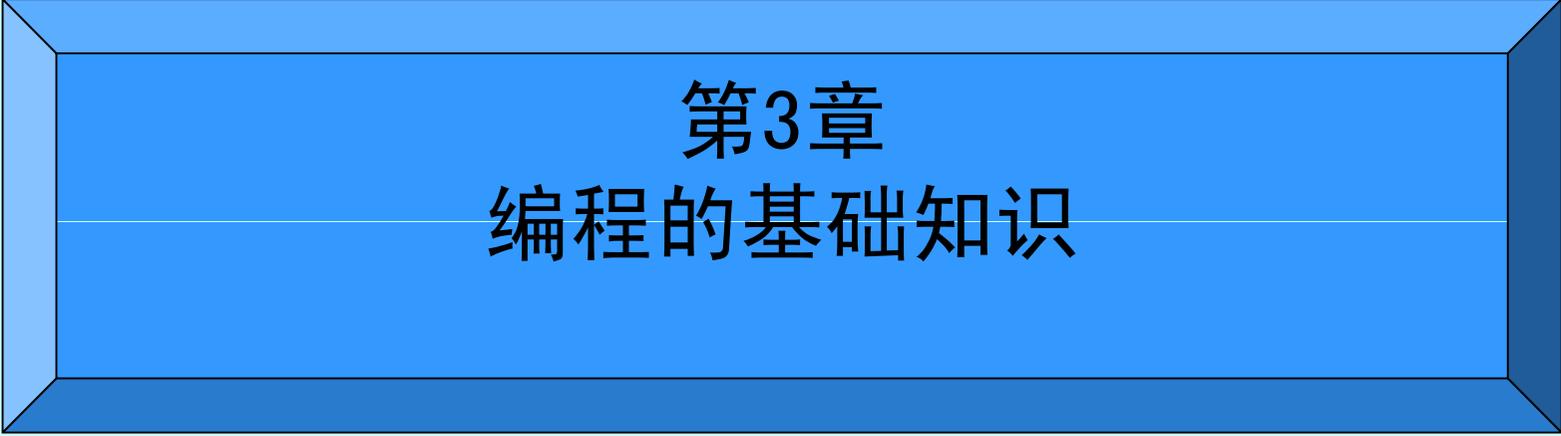
- X0~X5: 扳把开关(预留)
- X7: 机械臂位置传感器
- X6: 机械臂原点传感器
- X8: 推杆后退检测传感器
- X9: 推杆前进检测传感器
- XA: 旋转工作台位置检测传感器
- XB: 有无工件检测传感器

◆ 出力割付

- Y0: LED
- Y1: 回转台转动
- Y2: 回转台正反转切换
- Y3: 机械臂驱动
- Y4: 推杆前进驱动
- Y5: 夹具驱动

【培训用模型】





第3章 编程的基础知识

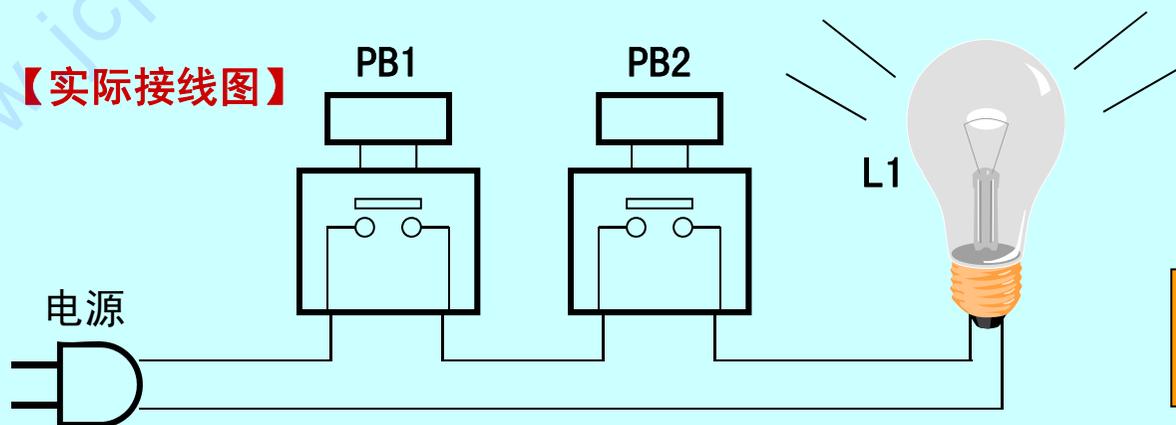
【3-1. PLC的回路图】

在PLC中使用的回路图被称为梯形图。

梯形图是使用触点符号、把自动控制作用电气回路来表示的“高级编程语言”。

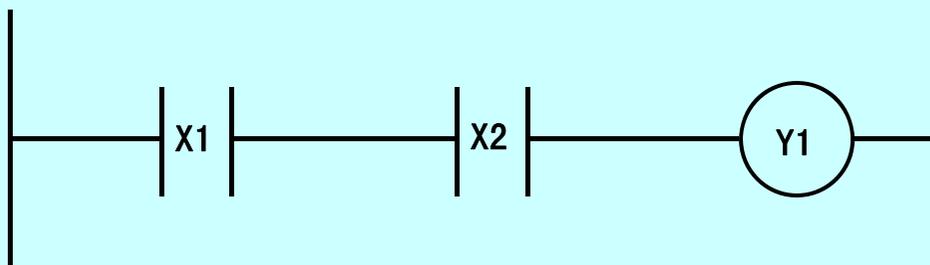
回路图举例：同时按下按钮SW（PB1、PB2），则灯（L1）亮。

【实际接线图】



不使用回路符号、而是直接表现机器的接续状态的图，称为实际接线图。

【梯形图】

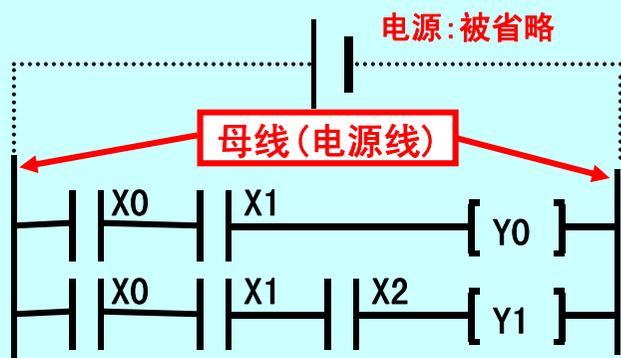


使用接点符号、把控制方法置换到回路图，这个回路图就称为梯形图。

【3-2. 梯形图的阅读方法】

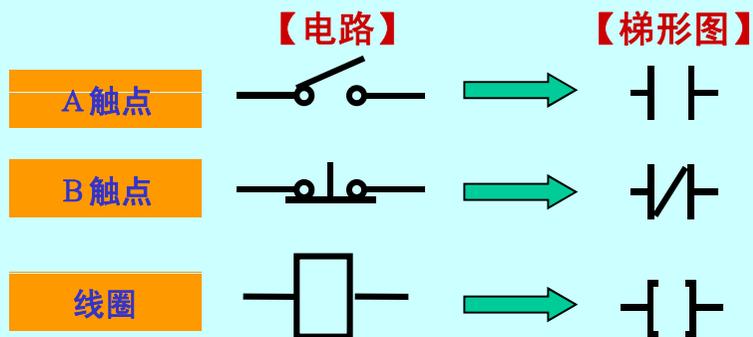
【梯形图】

一般在PLC的程序中，以梯形图形式表示电流方向。



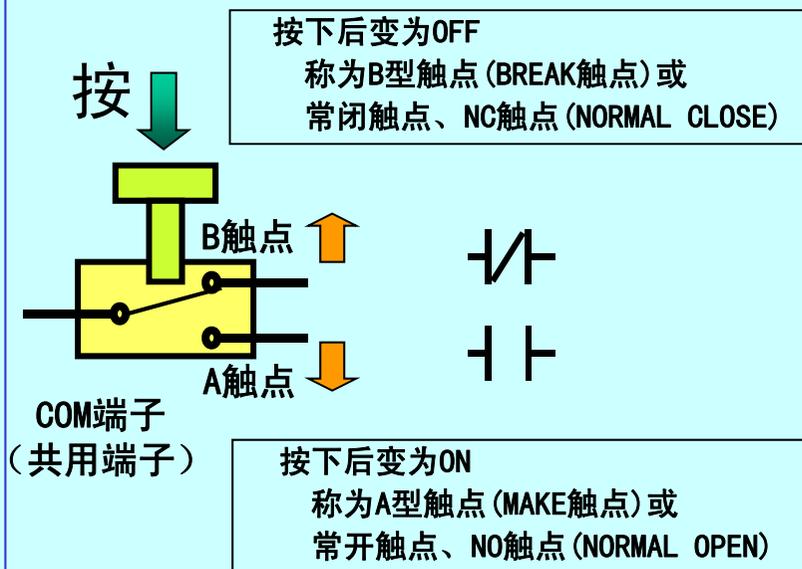
【梯形图的回路符号】

为了打印出以往在PLC中使用的各种电路触点符号，将这些内容文字符号化，统一成为A触点、B触点。



【什么叫A触点、B触点？】

例：按钮开关



【小结】

在PLC程序的多种方式中，作为具有代表性的梯形图方式，由于非常类似继电器顺序控制回路而被广泛使用。

【梯形图的绘制步骤】

- ①画出控制电源母线
 - ②在控制电源母线内连接各触点和输入输出继电器等要素
- 电路图中定时器、限位开关、继电器等触点的符号各不相同，而在PLC的梯形图中却不加以区别，仅使用打印机可以打印的文字符号。

【3-3. ST · ST / · OT指令】

◆ST（初始加载）· ST /（初始加载非）· OT（输出）

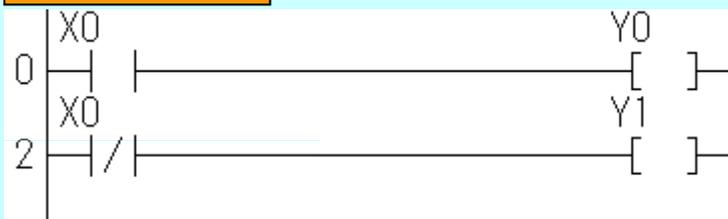
ST = 把A型触点连接到母线上的指令。

ST / = 把B型触点连接到母线上的指令。

OT = 向输出继电器线圈的输出指令。

ED = 表示程序结束。

【梯形图】

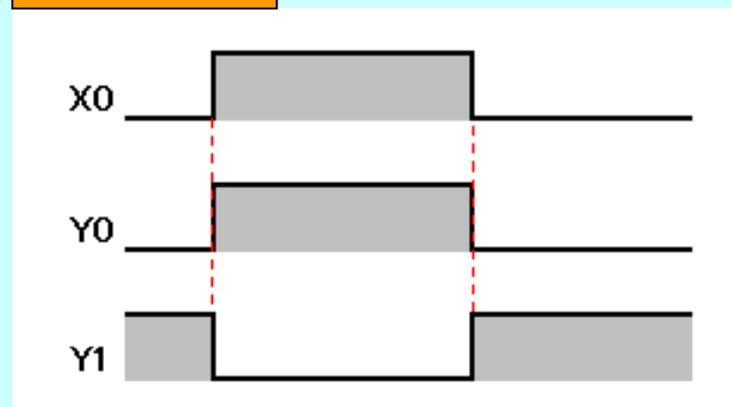


【程序动作说明】

X0为ON时、Y0为ON、Y1为OFF；

X0为OFF时、Y0为OFF、Y1为ON

【时序图】



【布尔助记符】

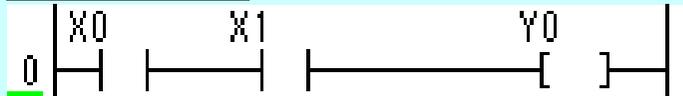
地址	指令
0	ST X0
1	OT Y0
2	ST/ X0
3	OT Y1

【3-3. AN（逻辑与）指令】

◆AN（AND 逻辑与）

AN = 把A型触点串联连接

【梯形图】

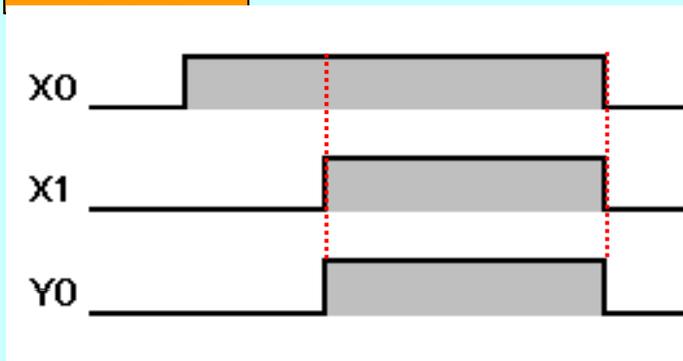


【程序动作说明】

X0为ON且、X1为ON时 Y0为ON

X0即使为ON, X1为OFF, 则Y0变为OFF

【时序图】



【布尔助记符】

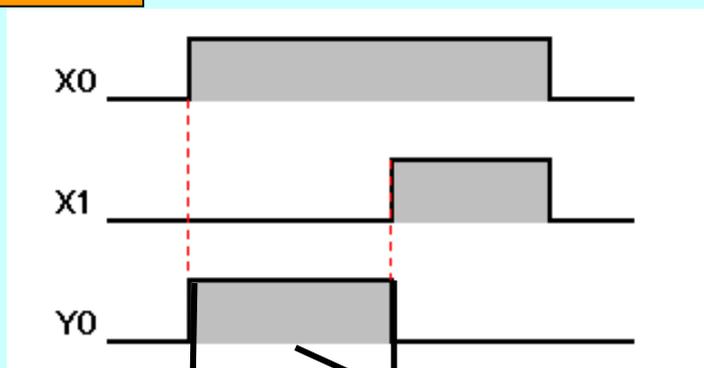
地址	指令
0	ST X0
1	AN X1
2	OT Y0

【3-3. AN/（逻辑与非）指令】

◆AN /（AND NOT逻辑与非）

AN / = 把B型接点串联连接

【时序图】

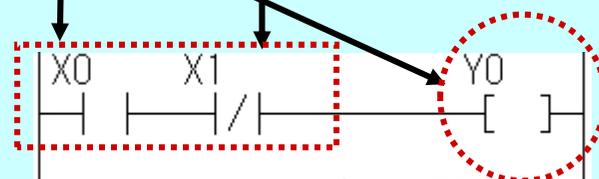


【程序动作说明】

X0为ON且、X1为OFF时 Y0为ON

X0即使为ON, X1为ON、则Y0变为OFF

【梯形图】



【布尔助记符】

地址	指令
0	ST X0
1	AN/ X1
2	OT Y0

X0为ON

直到X1变为
ON为止

Y0为ON

【重要回路：之一】

请一定记住这个程序模板，它是重要回路之一。

【3-3. OR · OR/指令】

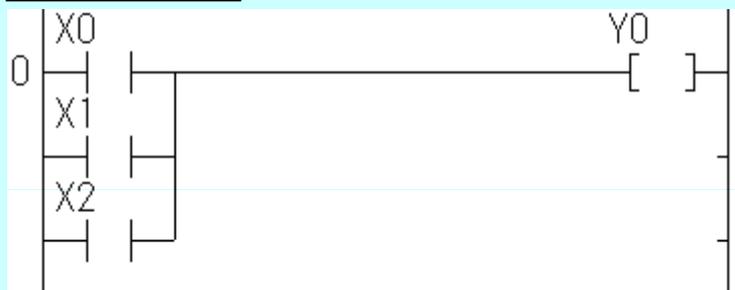
◆OR（逻辑或）·OR /（逻辑或非）

OR = 把A触点并联连接

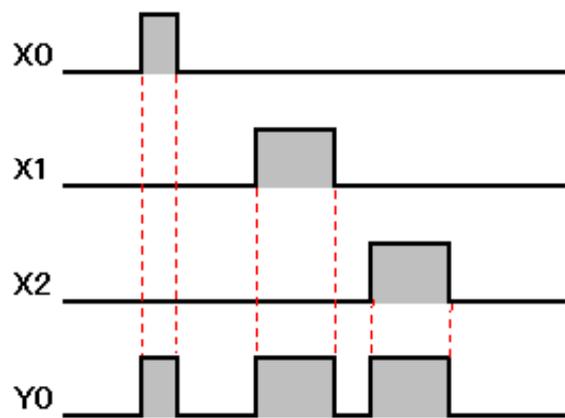
OR / = 把B触点并联连接

这是非常重要的基本回路之一

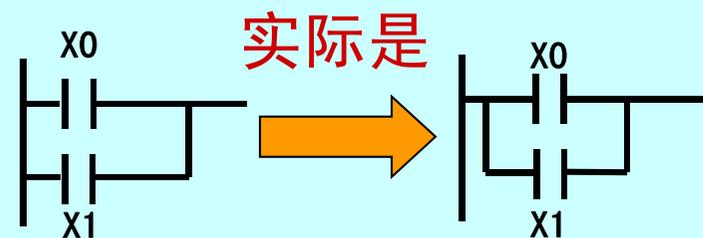
【梯形图】



【时序图】



【不使用ST X1，这是为什么？】



【程序动作说明】

即使X0、X1、X2之一为ON，Y0也为ON

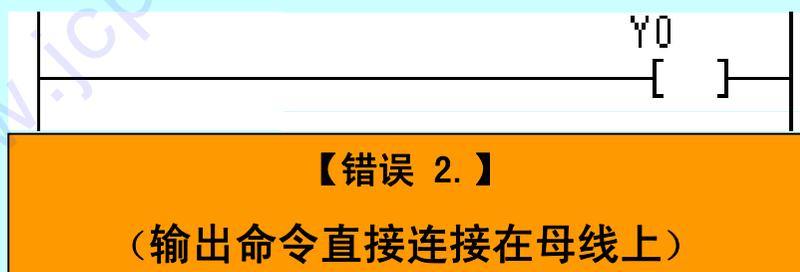
【布尔助记符】

地址	指令
0	ST X0
1	OR X1
2	OR X2
3	OT Y0

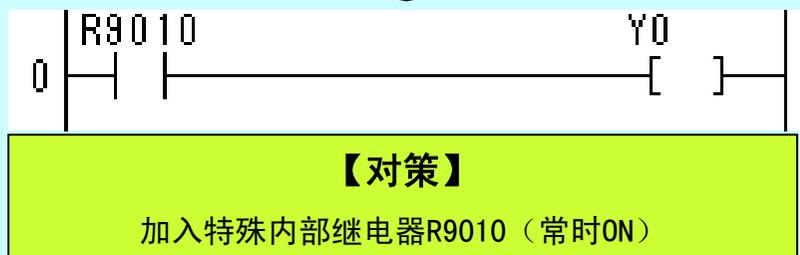
【3-4. 编程时的注意事项】

◆线圈的位置·双重输出

1. 在OT指令的线圈与右侧母线之间不能写有触点。
2. 不能把输出指令直接连接到母线上。
3. 相同序号的OT指令或定时器指令、计数器命令是错误的。



修改



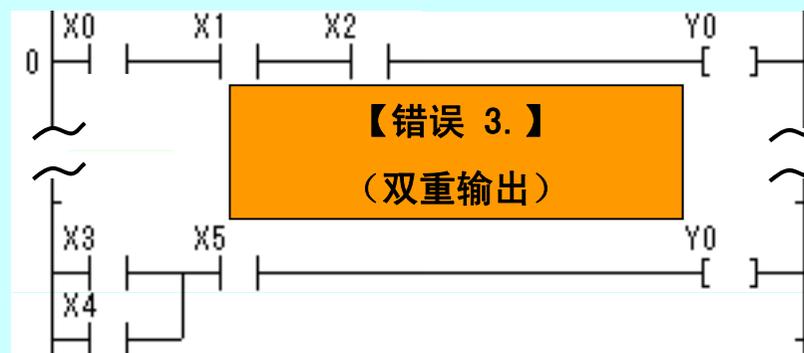
【程序的检查方法】

菜单

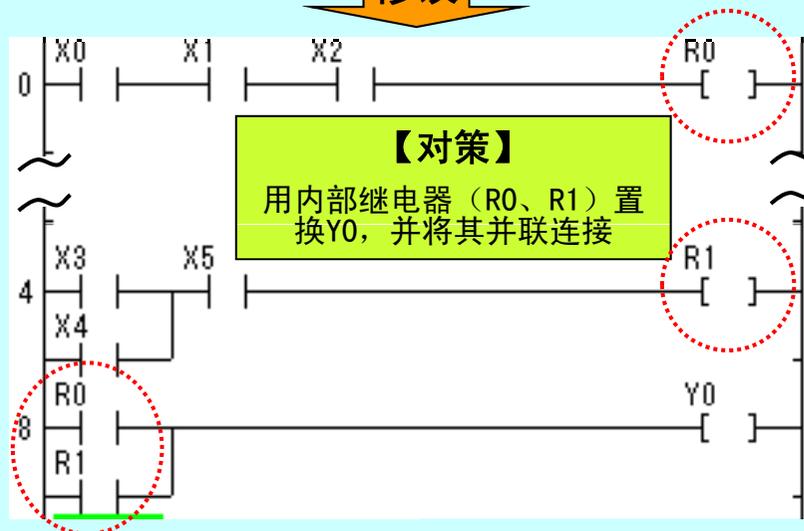
→ 调试(D)

→ 总体检查(C)

发现错误后, 显示出异常程序的地址



修改



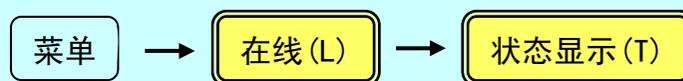
【3-5. 程序错误一览表】

◆CPU的「ERROR / ALARM」灯闪烁时，说明PLC上有某种错误发生，需要确认错误内容并加以处理

■确认错误方法 1.

【步骤】

PLC方式：在线

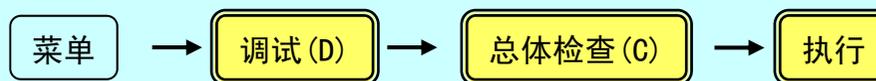


【显示错误代码及其内容】

■确认错误方法 2.

【步骤】

PLC方式：在线



【显示错误内容及其地址，修改程序】

【错误代码举例】

代码	名称	运行	错误内容及处理方法
E1	语法错误	停止	写入了含有语法错误的顺序控制程序。 请切换到PROG. 模式、修改错误部分。
E2	双重输出错误	停止	在输出指令或者保持指令中多次使用了相同的继电器。 请切换到PROG. 模式，将程序修改为在一个程序中某个继电器只使用一次。此外，也可以将系统寄存器设置为允许双重输出
E3	匹配指令不成立	停止	在例如跳转(JP和LBL)等匹配使用的指令中，由于缺少某一项或位置关系不正确等而无法执行。 请切换到PROG. 模式，将匹配使用的2个指令修改到正确位置。

技成培训网
www.jcpeixun.com

第4章 PLC的基本回路

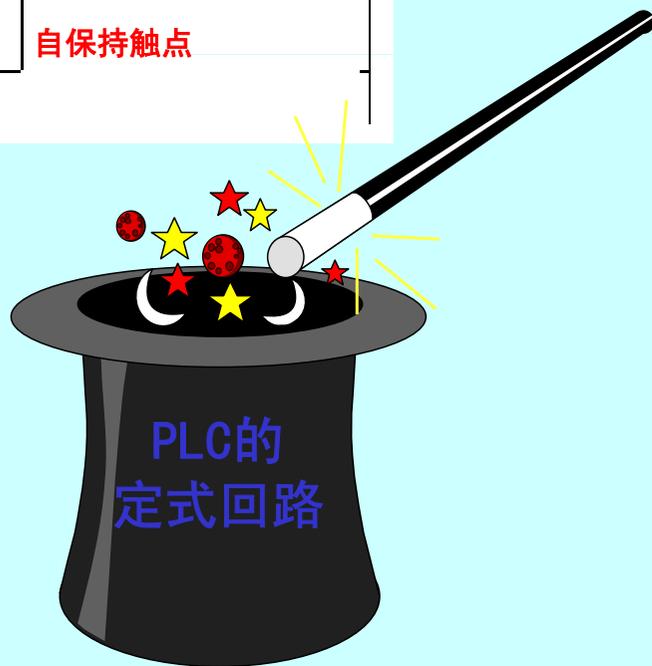
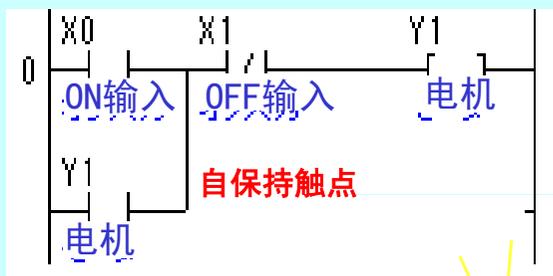
【4-1. 自保持回路】

【PLC的最重要回路】

自保持回路具有状态记忆功能

这是非常重要的回路

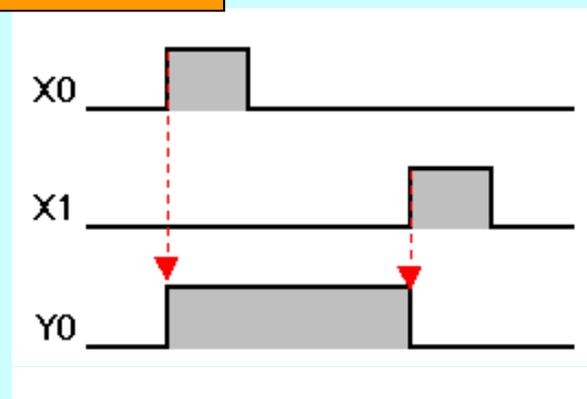
【梯形图】



【程序动作说明】

- ① 输入信号X0为ON，电机(Y1)变为ON
- ② 即使X0变为OFF状态、Y1仍能保持ON的状态
- ③ 输入为ON时，电机(Y1)变为OFF

【时序图】



【布尔助记符】

地址	指令
0	ST X0
1	OR Y0
2	AN/ X1
3	OT Y0

【4-2. 自保持回路的改进①】

【编制自保持程序】

请编写用于培训模型的自保持回路的程序。

【培训模型的动作】

- ①启动开关 (X0) ON
- ②回转台 (Y1) 转动
- ③到达传感器 (XA) 的位置后 (Y1) 停止转动

★ 请不要用手转动回转台！

程序动作确认的准备工作

清除程序的操作步骤

菜单

→ 编辑 (E)

→ 程序清除 (L)

强制输入输出的操作步骤

通过计算机强制进行ON OFF操作

菜单

→ 在线 (L)

→ 强制输入输出 (C)

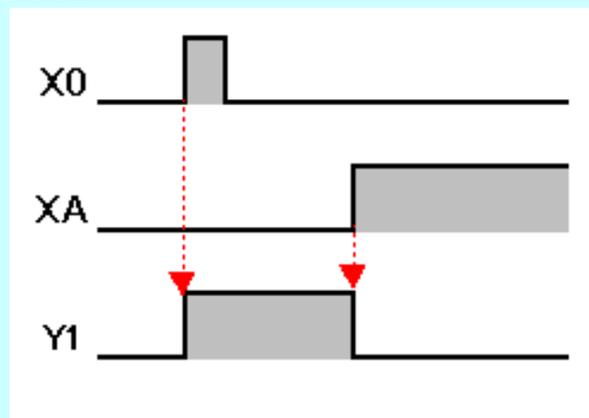
设备登录

→ 选择Y1后 点击ON按钮

在XA为OFF的位置点击OFF按钮

→ 解除

【时序图】



编制完程序之后，将程序下载到PLC
把动作模式切换到【RUN】
将 (X0) 置于ON，确认动作情况

【问题】

请再次闭合启动开关X0。
回转台会动作吗？

【答案：不会动作。】

【为什么不能再次启动？】

动作一次并停止后，位置检测传感器 (XA) 保持ON的状态。
因此，即使启动开关再次为ON，输出也不能变为ON。

使用以下的“DF”（微分）指令，
改进自保持回路的程序。

【4-3. 微分 (DF) 指令】

◆ DF (上升沿微分) · DF / (下降沿微分)

DF = 检测到输入信号有上升沿 (OFF→ON) 时, 将指定的线圈仅ON一个扫描周期。

输入从最初开始始终为ON的情况下, 不动作

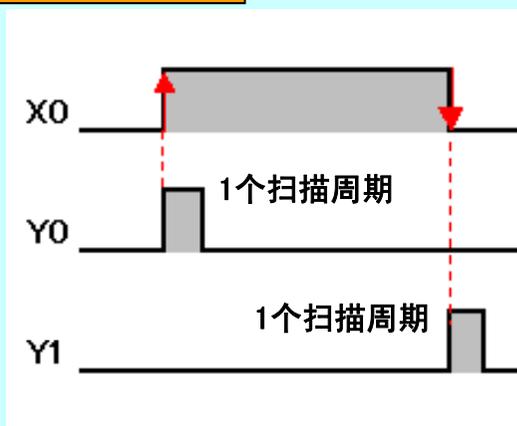
DF / = 检测到输入信号有下降沿时 (ON→OFF) 时, 将指定的线圈仅ON一个扫描周期。

输入从最初开始始终为OFF的情况下, 不动作

【梯形图】



【时序图】



【布尔助记符】

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

【程序动作说明】

- 1) X0从OFF变为ON, 仅一个扫描周期Y0为ON
- 2) X0从ON变为OFF, 仅一个扫描周期Y1为ON

因为仅ON一个扫描周期, 时间非常短暂,
所以几乎看不到Y0灯亮

【4-4. 自保持回路的改进②】

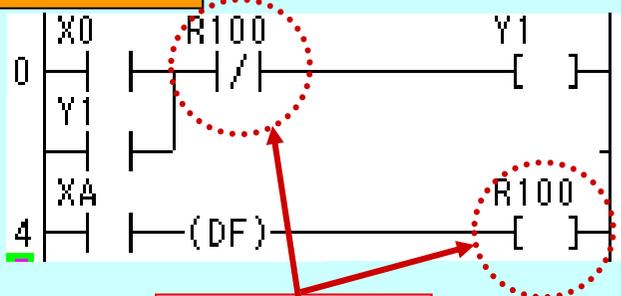
【对程序进行改进】

使用微分指令【DF】修改刚才所作的自保持回路，使之可以再次启动。

清除程序的操作步骤



【梯形图】



(仅一个扫描周期为ON)

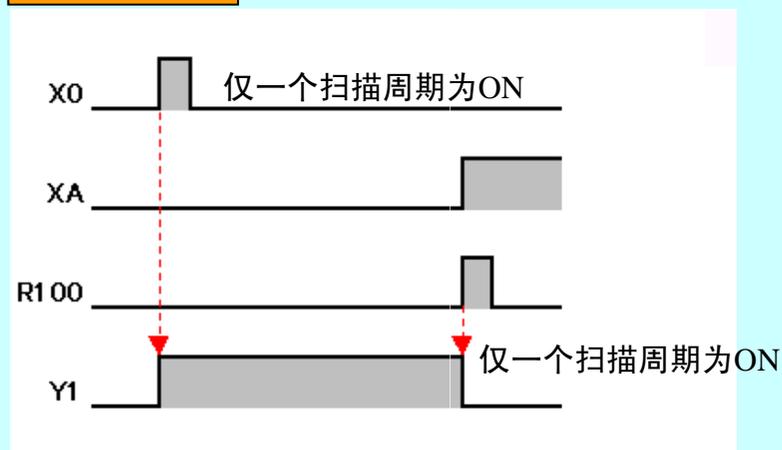
【功能解释】

- ①微分命令仅在其之前的触点发生ON→OFF或OFF→ON变化时，才使线圈在称为一个扫描周期的、非常短暂的时间内输出ON。
- ②即使位置检测传感器(XA)为ON状态保持不变，(R100)为ON的时间也仅是一瞬间。

【布尔助记符】

地址	指令
0	ST X0
1	OR Y1
2	AN/ R100
3	OT Y1
4	ST XA
5	DF
6	OT R100

【时序图】



【4-5. 步进跟踪法的自保持回路】

【问题】

在启动开关(X0)保持ON状态不变的情况下，
动作会怎样呢？

【答案：不停止】

在启动开关后加入微分指令试试看。
首先请清除以前的程序。

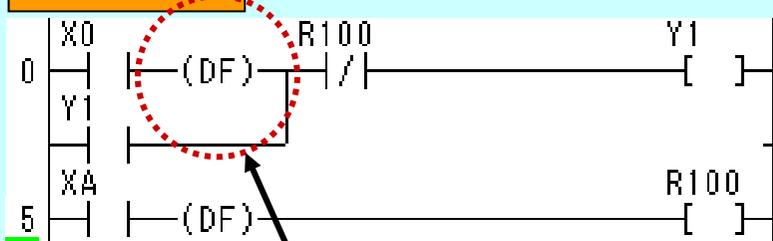
清除程序的操作步骤

菜单

编辑(E)

清除程序(L)

【梯形图】



通过加入微分指令(DF)，消除X0的ON的时间影响。
X0仅在OFF→ON变化时的一个扫描时间内流过电流。

【为什么不停止转动？】

通过对位置检测传感器(XA)使用微分指令使(R100)仅在一瞬间ON。

但由于(X0)为ON保持不变，即使通过(R100)在瞬间解除我保持，也会使(Y1)立即变成ON，回转台不停止转动。

【启动开关也需要微分指令】

如果对启动开关(X0)使用微分指令，则(X0)即使为ON保持不变，也会完全停止

【布尔助记符】

地址	指令
0	ST X0
1	DF
2	OR Y1
3	AN/ R100
4	OT Y1
5	ST XA
6	DF
7	OT R100

到这里，您已经掌握了
本次研修的80%了！



【重要】自保持回路用微分指令启动、用微分命令停止

【4-6. 定时器(TM)指令】

◆定时器(TM)

TM = 当输入信号为ON时, 在经过了设定的时间之后, 定时器触点变为ON(延时继电器)

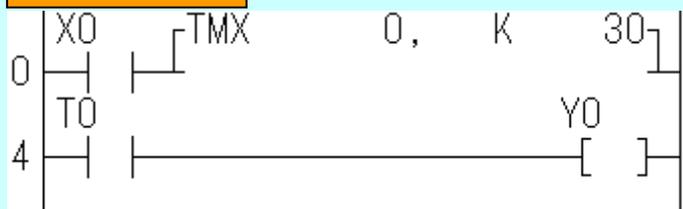
最大使用点数 = 定时器和计数器合计最大可以使用到144点(0~143)

定时器编号 = 初始值为可以使用100点(0~99), 当定时器的点数不足时可以增加

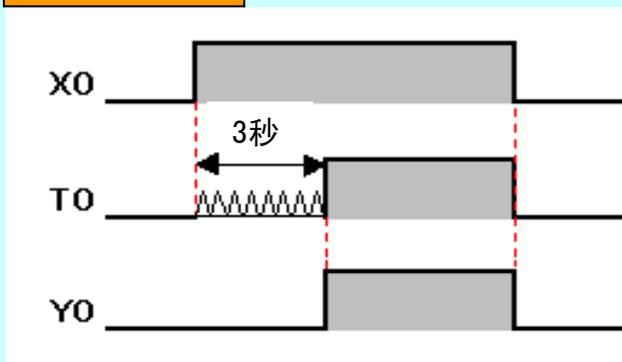
* 不能与计数器编号重复使用

* 定时器的设定方法 = $TMX(0.1\text{秒定时器}) \times 30 = 3\text{秒}$

【梯形图】



【时序图】



【程序动作说明】

当X0变为ON后, 对设定时间(3秒)进行减法运算, 经过3秒过后, 定时器触点T0变为ON, Y0也随之ON.

在到达定时器的定时之前, 如果X0变为OFF, 则经过值被复位, 回到原设定值, 而定时器的触点也不会为ON.

【定时器的分类】

定时器分为4种定时量程, 分别用字母来区别.

TM	L	=	0.001 秒单位
TM	R	=	0.01 秒单位
TM	X	=	0.1 秒单位
TM	Y	=	1 秒单位

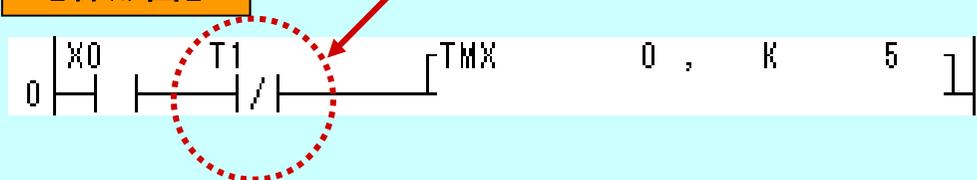
【4-7. 定时器应用回路】

【闪烁(往复)回路】

想使定时器反复动作时,请在程序的开头部分

※ 补充插入最后出现的定时器的b型触点

【梯形图】



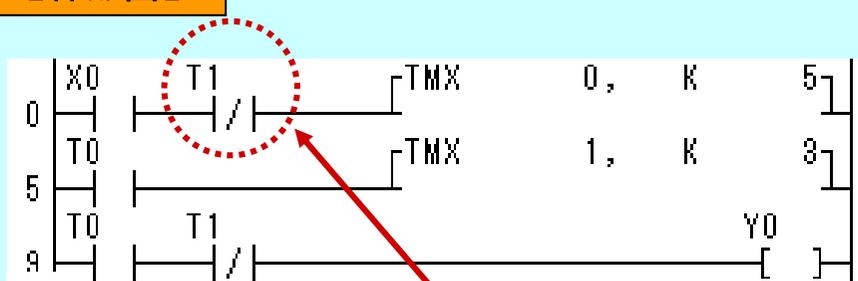
【程序动作说明】

(X0)为ON、直到(T1)为ON为止,在定时器0的线圈(TM X 0)中电流往复流过。

【练习题】

请在清除程序之后,编写下列程序,再次确认反复定时闪烁回路的动作。

【梯形图】



连接最后出现的
定时器的B型触点。

【PLC的定式回路】



用于身边的紧急状态时
闪烁的报警指示灯等。

作为PLC的定式回路,
请记住这种回路。

这些回路最多也就10种,
非常容易记住。

技成培训网
www.jcpeixun.com

第5章 编程实践

【5-1. 通常的输出控制】

【顺序动作程序 1.】

编写定时器应用回路，学习步进跟踪编程法的思路。

【程序动作说明】

- ① 启动开关 (X0) ON, 1秒后LED指示灯 (Y0) 亮
- ② 2秒后回转台 (Y1) 开始转动

首先清除以前的程序

清除程序的
操作步骤

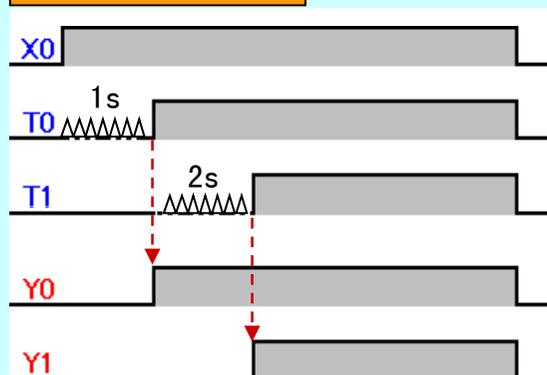
菜单

编辑 (E)

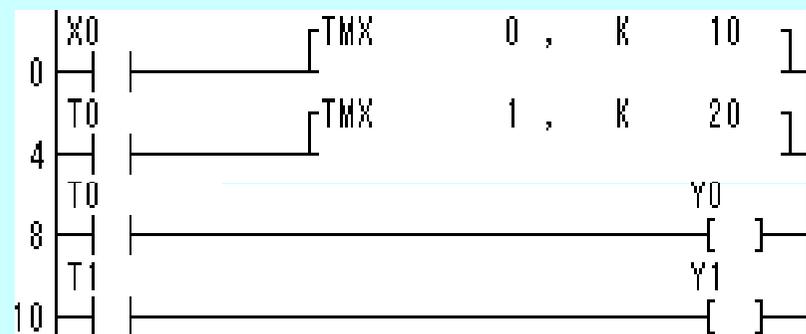
清除程序 (L)

然后，输入下述程序，下载到PLC中后确认动作。（PLC设为RUN模式）

【时序图】



程序展开



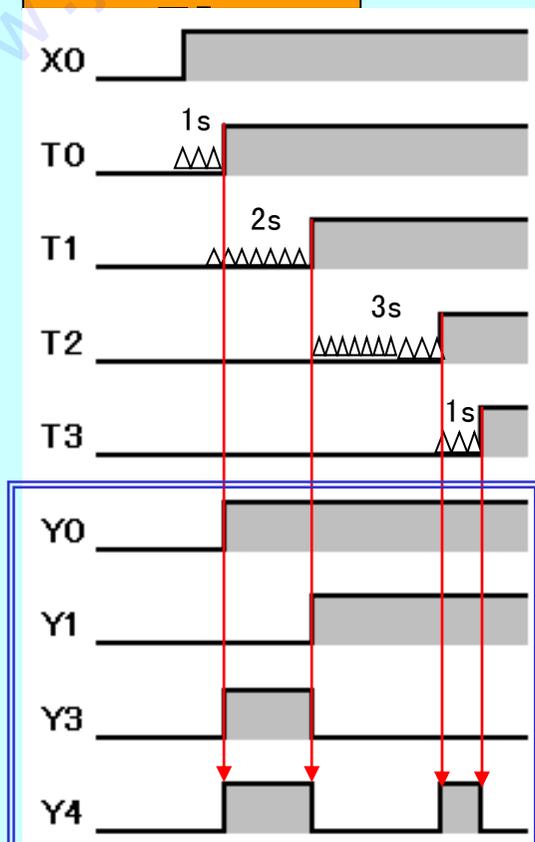
下面请继续编写程序2。

【5-3. 利用步进跟踪编程法的输出控制】

【顺序动作程序2.】

请编写顺序动作程序

【顺序动作时序】



编写程序

【程序动作说明】

- ① 启动开关 (X0) ON、1秒后LED (Y0) 和机械臂 (Y3)、推杆 (Y4) 3点输出为ON。
- ② 2秒后, 回转台 (Y1) 变为ON、机械臂 (Y3) 和推杆 (Y4) 同时变为OFF
- ③ 3秒后, 推杆再次变为ON
- ④ 1秒后, 推杆变为OFF, 一套动作结束

【完成程序】

请利用左侧的时序图编制4点输出 (Y0、Y1、Y3、Y4) 的程序。
因为 (Y4) 有二次为ON, 所以在编程时请加以注意.



【5-4. 实践步进跟踪编程法 绘制时序图】

如果使用步进跟踪法，**不论是谁都能简单地、机械地**编制程序。
利用步进跟踪编程法，按照以下步骤编写程序。

- **把机械动作的动作过程用时序图表示。**

请务必绘制时序图。如果能够绘制出时序图，
则可以认为任务已经完成一半。

- **按照上述时序图，编写梯形图程序。**



【时序图绘制方法】

- ① 根据用户的要求，将机械动作的输入和输出表述为时序图的形式。

(到此步骤为止，请与用户一边商讨一边绘制)

- ② 通过表达机械动作机械的时序图，绘制出利用自保持回路、按步分解的时序图(此为设计工作)
将输出发生变化的时刻点作为每一步的开始点，
这样可以使每一步的分解方法易于理解。

【5-5. 步进跟踪编程法实践 编写梯形图】

按照绘制好的时序图编写梯形图程序。

【编程方法】

※利用步进跟踪编程法的编程要点

- ① 利用自保持回路记忆各步序。
- ② 利用微分指令使自保持回路ON/OFF。
- ③ 务必在自保持回路的ON的部分中加入限制条件。
- ④ 自保持回路在最后之前不应置OFF，
而应该在最后一齐置OFF。
- ⑤ 对于结束信号，应利用微分信号在限制条件中
加入前一个的自保持输出。
- ⑥ 结束信号采用。
- ⑦ 输出控制部分集中放置到整个程序的最后。
将步序记忆回路程序块与输出回路程序块
分开编写，可以使程序检查变得非常轻松。

【培训模型的各工序动作】

通过影片了解培训模型是如何进行控制的。

将整个课题分解成3个部分。
由影片分别确认。

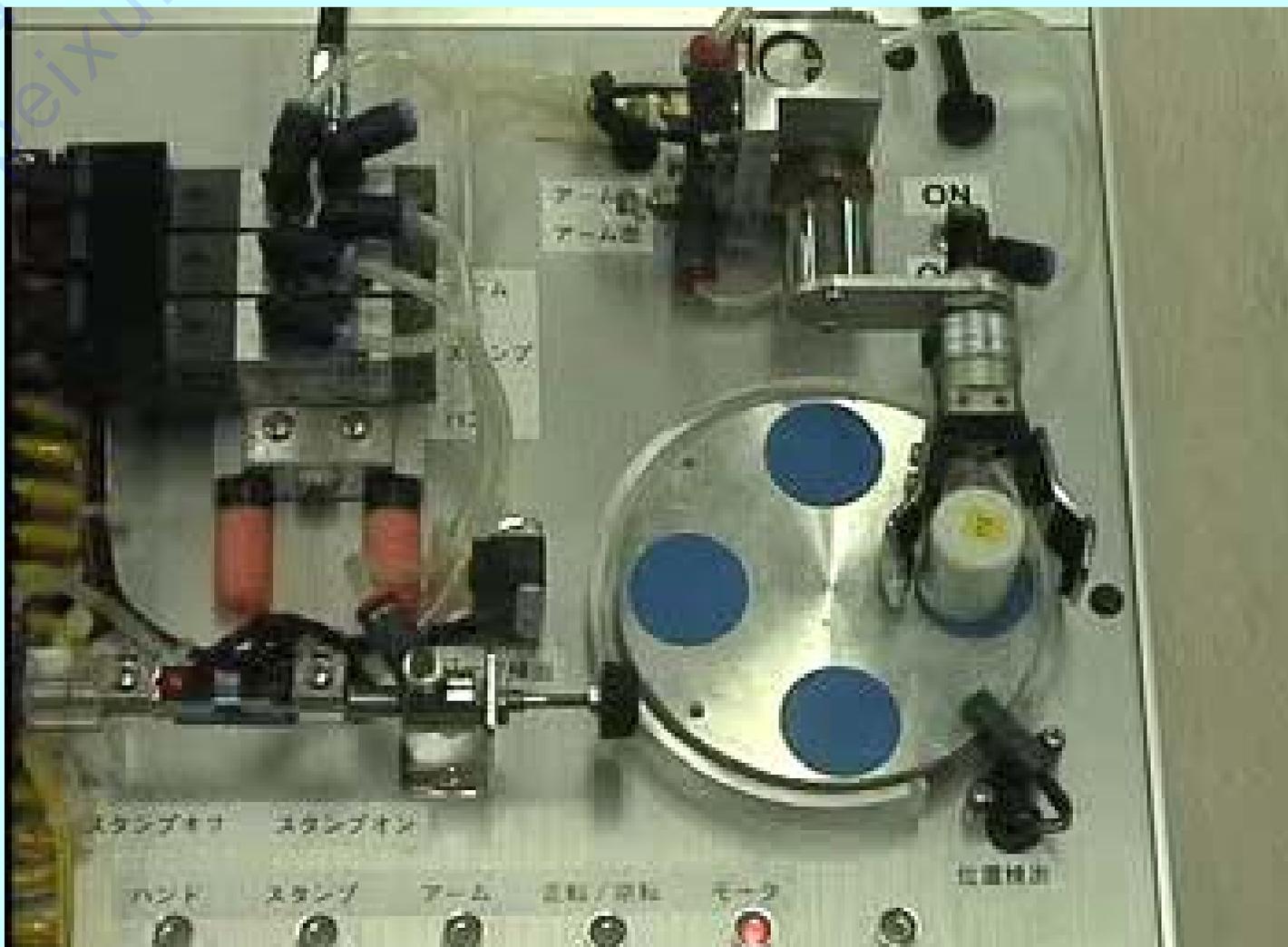
第 1 工程

第 2 工程

第 3 工程



Step 1



【5-6. 挑战课题 — 第 1 工程】

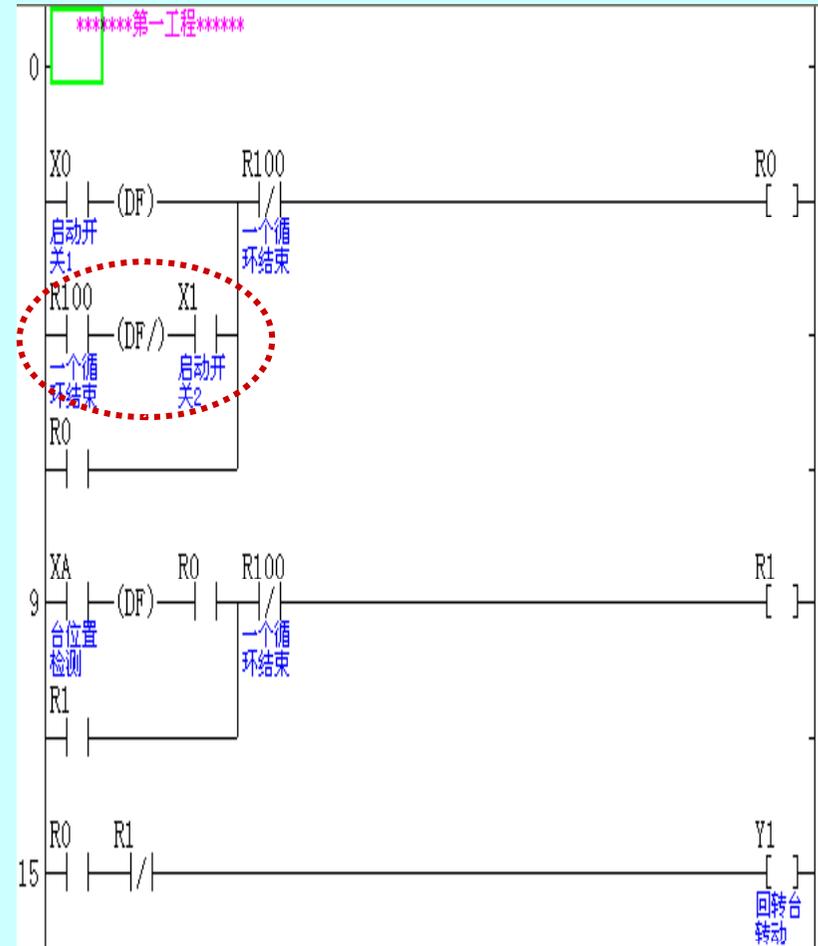
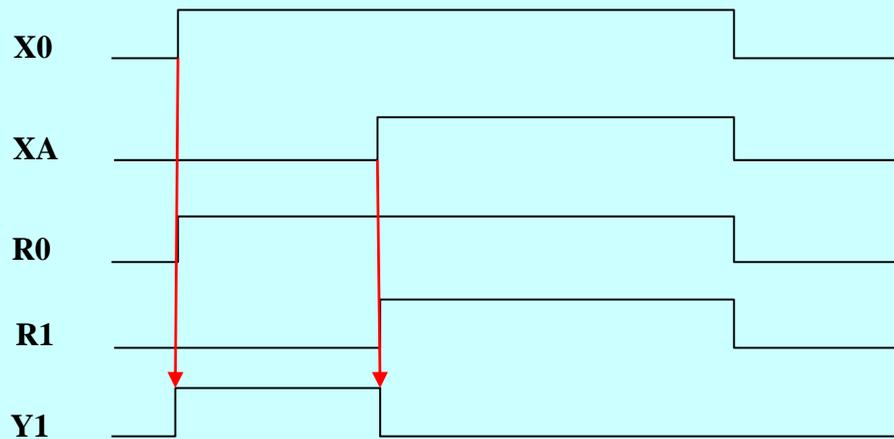
根据短片所播放的动作，按照以下的步骤，练习编写第1工程的程序。

1. 绘制时序图。
2. 以时序图为基础，编写梯形图程序。

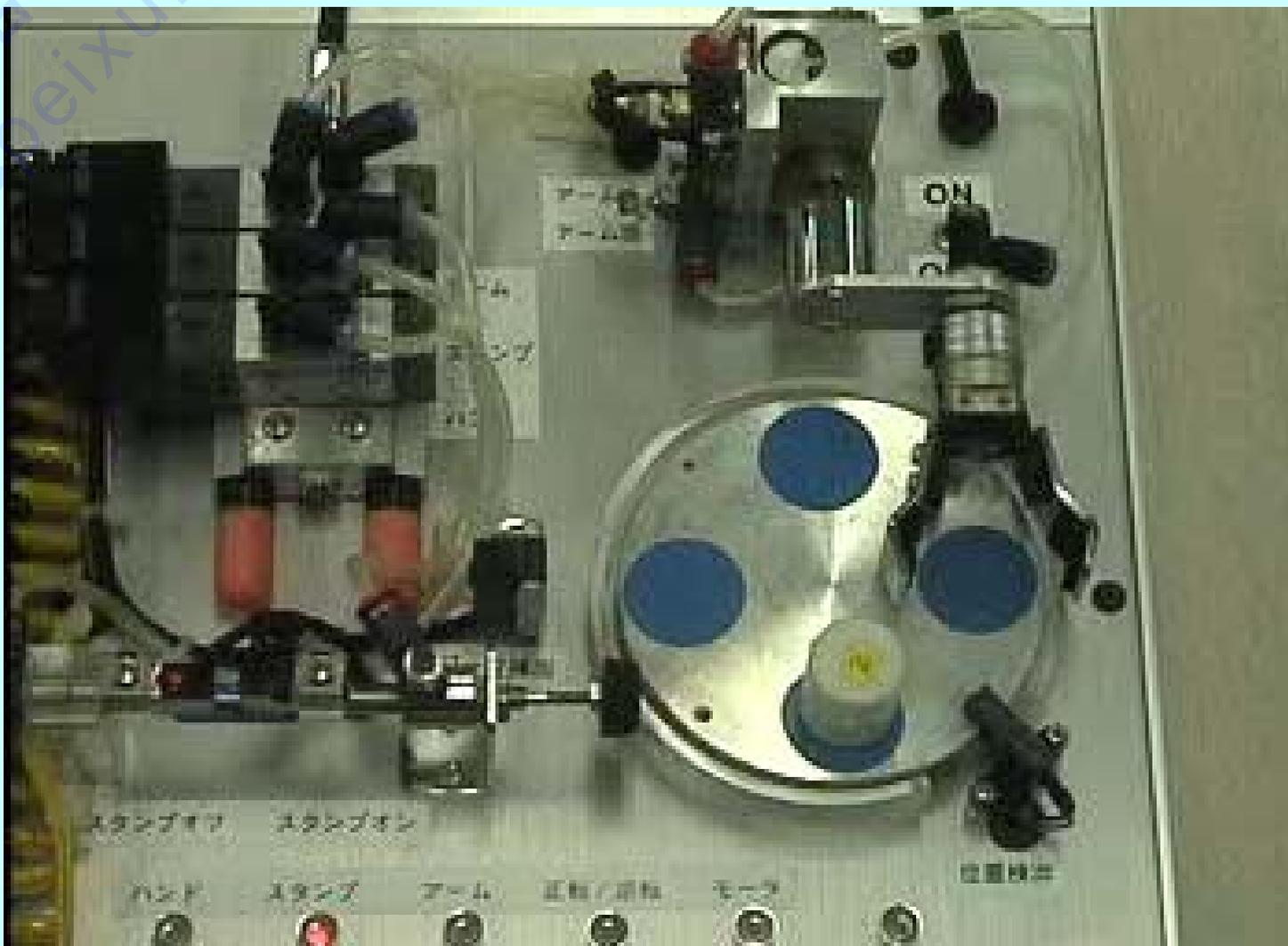
【第1工程的动作】

- ① 通过启动开关 (X0 ON) 使回转台转动 (Y1 ON)
- ② 通过位置检测传感器 (XA ON) 停止转动。

时序图



Step 2



【5-6. 挑战课题 — 第2工程】

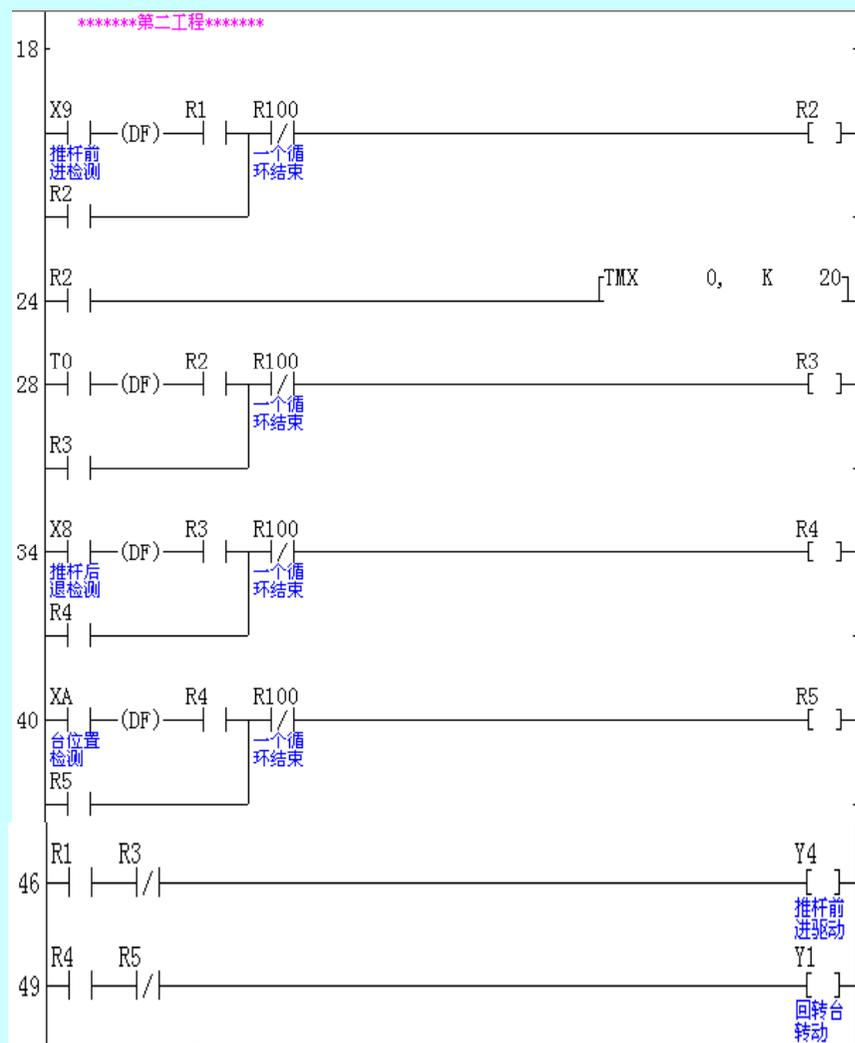
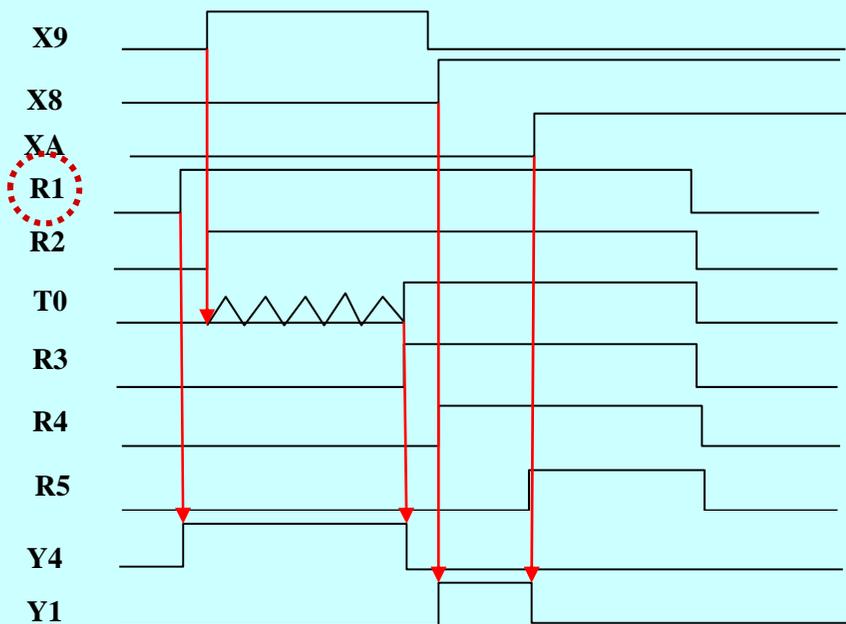
根据短片所播放的动作，按照以下的步骤，练习编写第2工程的程序。

1. 绘制时序图。
2. 以时序图为基础，编写梯形图程序。

【第2工程的动作】

- ③ 推杆前进。（Y4 ON）
- ④ 推杆前进以后，推杆前进端检测传感器（X9 ON）
- ⑤ 停机等待2秒钟。（使用0号定时器）
- ⑥ 推杆后退。（Y4 OFF）
- ⑦ 推杆后退之后，推杆后退端检测传感器（X8 ON）。
- ⑧ 使回转台（Y1）再次转动。
- ⑨ 通过位置检测传感器（XA ON）停止转动（Y1 OFF）

时序图



第2工程程序完成！
距成功只有一步之遥。

Step 3



【5-7. 挑战完成课题】

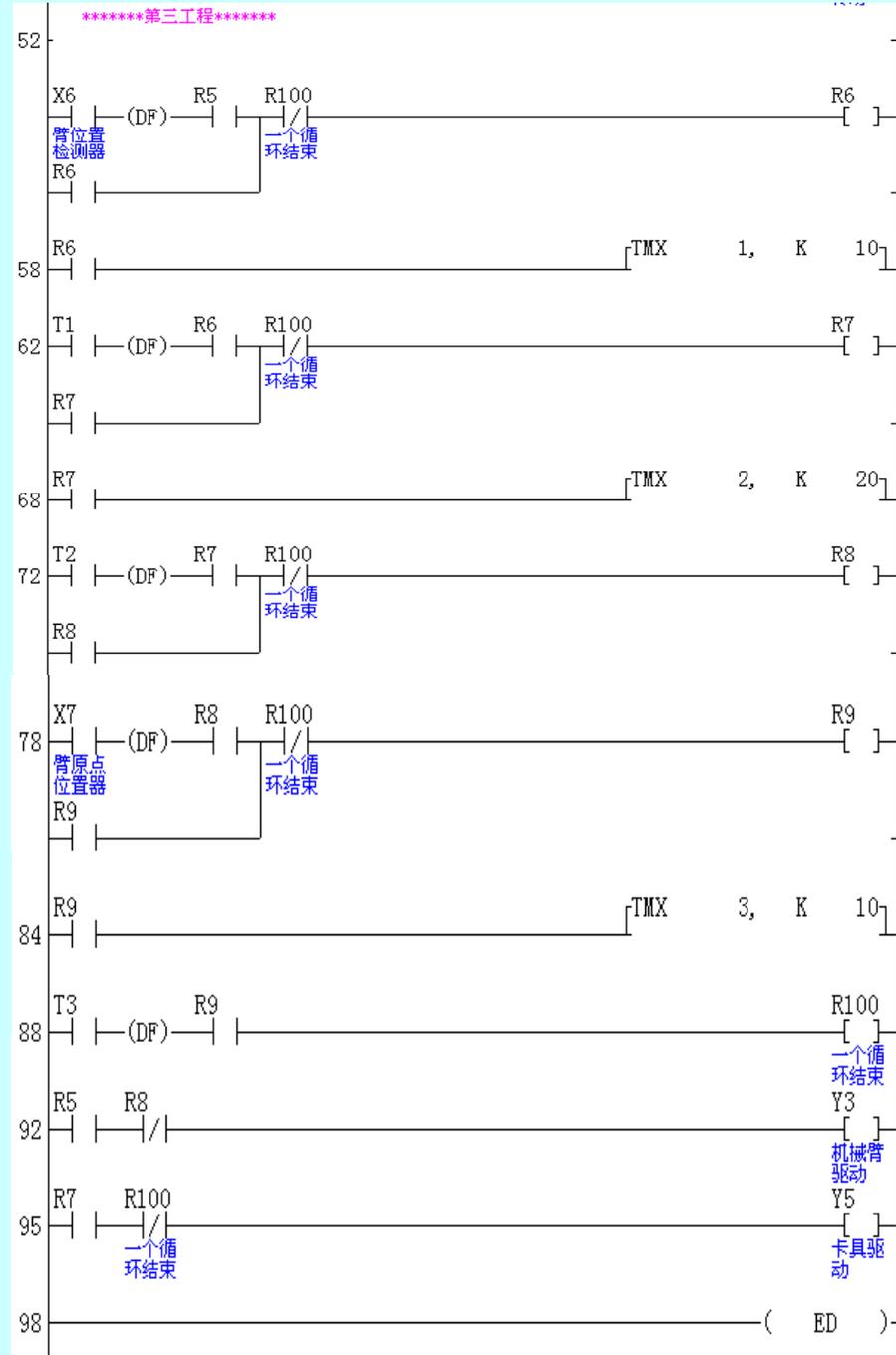
进一步添加搬移动作程序
完结自动运行回路(1个工作周期)。

【第3工程的动作】

- (10) 转动机械臂。(Y3 ON)
- (11) 机械臂转动位置检测传感器 (X6 ON)。
- (12) 停机等待1秒钟。(使用1号定时器)
- (13) 闭合夹具 (Y5 ON)、夹住工件。
- (14) 停机等待2秒钟。(使用2号定时器)
- (15) 机械臂返回原点。(Y3 OFF)
- (16) 机械臂原点位置检测传感器 (X7 ON)。
- (17) 停机等待1秒钟。(使用3号定时器)
- (18) 打开夹具 (Y5 OFF)、释放工件。

时序图

X6
X7
R5
R6
T1
R7
T2
R8
R9
T3
R100
Y3



【挑战完成课题】

【挑战课题 1】

如果开关(X1)为ON, 则连续运行。

请编写程序。

【挑战课题 2】

如果开关(X7)为ON, 则紧急停止。

请编写程序。



程序完成!

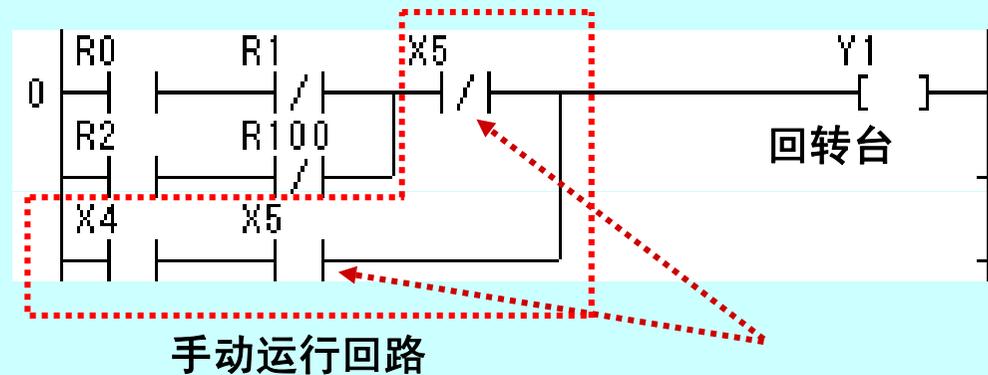


【5-8. 自动·手动切换回路】

◆ 补充自动运行·手动运行的切换回路

把输出部分的程序按下列回路修改，就可以实现自动运行·手动运行方式的切换。

【回转台输出回路】



【X5：置ON则切换到手动运行】

【程序动作说明】

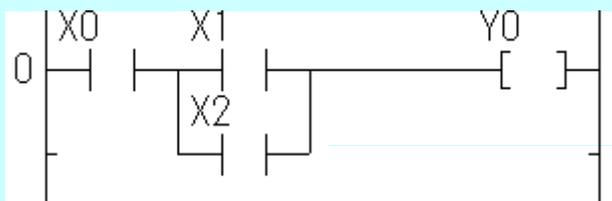
- ①X5 置ON则切换到手动运行
- ②X4 置ON则开始手动运行
- ③Y1(回转台)转动
- ④X5 置OFF则切换到自动运行

【5-9. 编程建议】

◆ 下面介绍简便、快速编程的要点。

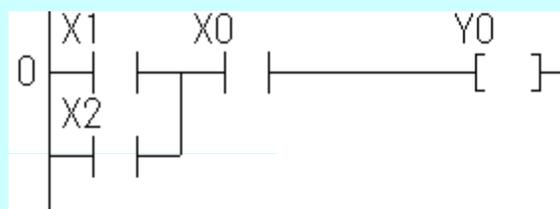
◆ 在串联连接回路中，要把连接触点较多的回路画在母线的左侧

需要ANS指令 合计5步



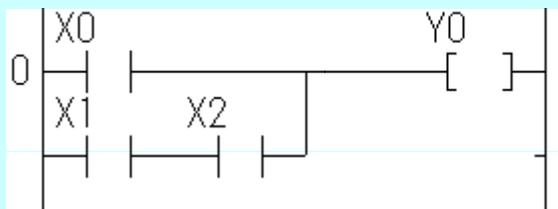
修改

不要ANS指令 合计4步



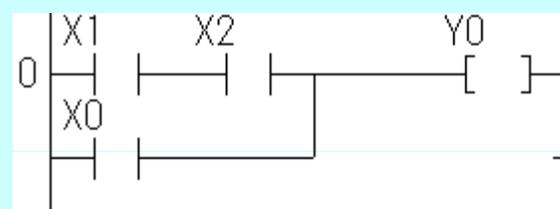
◆ 在并联连接回路中，要把连接触点较多的回路画在上侧

需要ORS指令 合计5步



修改

不要ORS指令 合计4步



第6章 便利命令介绍

【6-1. SET·RST 指令】

◆SET (置位)·RST (复位)

SET : 当执行条件成立时, 使输出变为ON并且保持ON的状态。

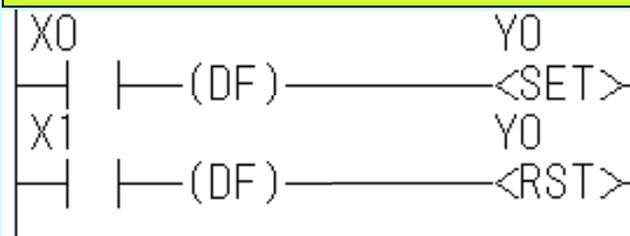
RST : 当执行条件成立时, 使输出变为OFF并且保持OFF的状态。

SET·RST可以特殊例外地使用“多重输出”。(不会产生语法错误)

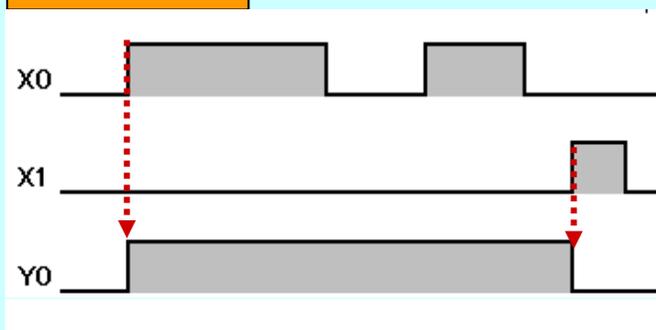
“必须”与微分指令组合使用。

程序的编写·调整都很简单。多次使用同一个输出时, 更加方便。

使用SET、RST指令后, 形成以下程序



【时序图】



【程序动作说明】

- ① 输入信号X0为ON。
- ② 输出Y0变为ON状态。
- ③ 即使X0变为OFF, Y0也仍然保持ON的状态。
- ④ 如果X1为ON, 则Y0变为OFF状态。

布尔助记符

地址、	指令
0	ST X0
1	DF
2	SET Y0
3	ST X1
4	DF
5	RST Y0

【6-1. 利用SET·RST指令的步进跟踪编程法】

【编程步骤】

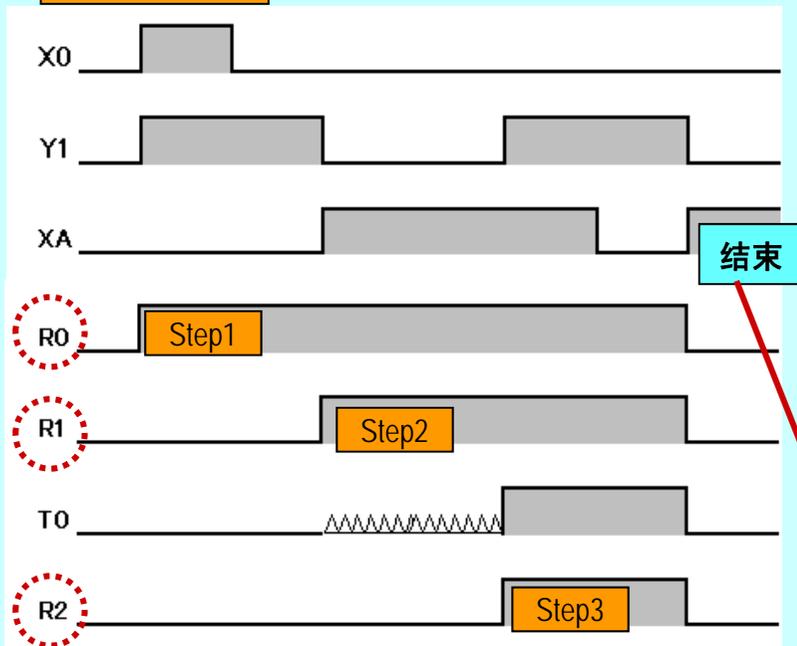
- ① 写出各输出类型的时序图。
- ② 在输出的变化点, 把用于记忆该步的内部继电器置ON。
- ③ 利用记忆各步的内部继电器、通过SET·RST指令, 将输出置位 (ON) 或复位 (OFF)。
- ④ 在各步的自保持回路中, 必须加入记忆前一步动作的内部继电器作为限制条件。
- ⑤ 利用结束信号, 将所有的记忆各步的内部继电器等用SET指令置ON的触点, 通过RST (复位) 指令置OFF。

使用SET·RST的优点:

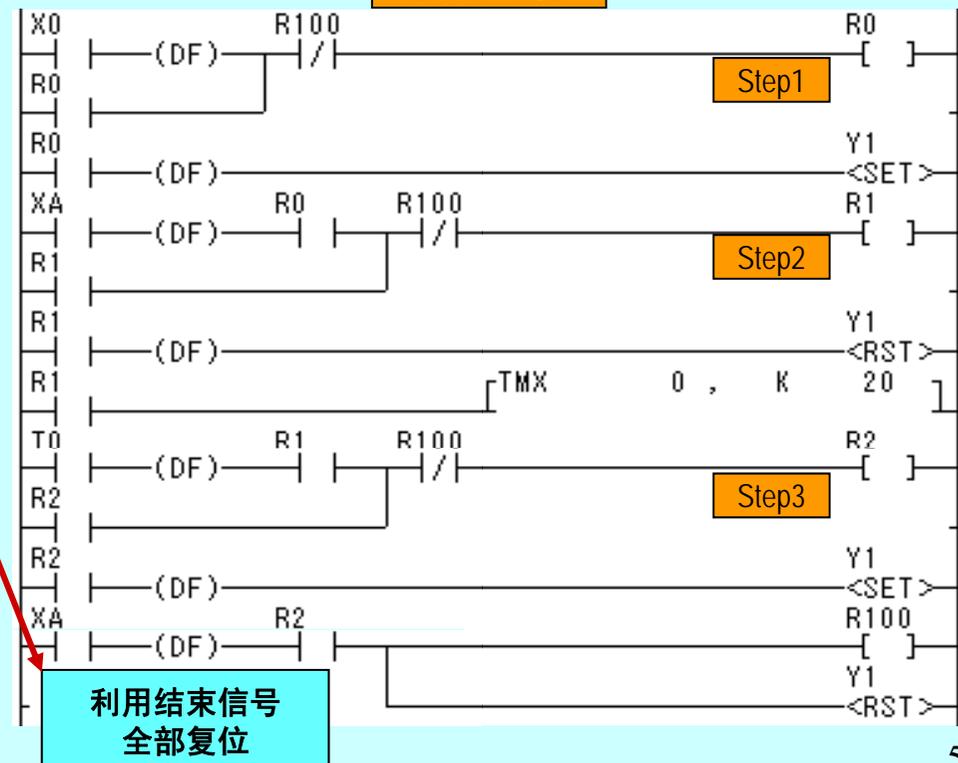
在各步能够直接对输出进行置位 (ON) 和复位 (OFF), 更利于理解程序动作. (因为可以双重输出)

重要 SET·RST命令必须和微分指令 (DF) 一起配套使用。

【时序图】



【梯形图】



利用结束信号
全部复位

技成培训网
www.jichengpeixun.com

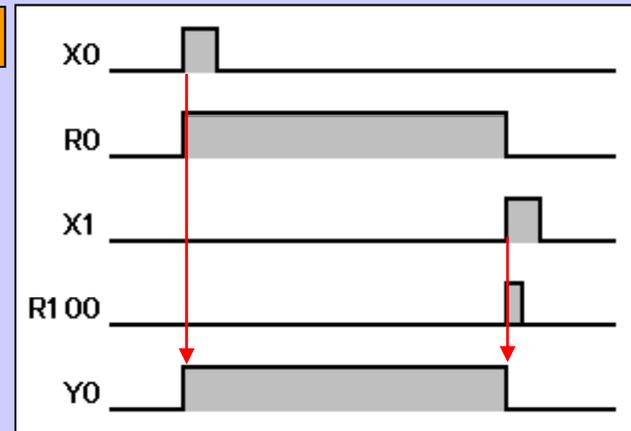
第7章
PLC基础教程
练习题

【7-1. PLC基础教程练习题】

【习题1】

按开关0 (X0) 则指示灯 (Y0) 亮
按开关1 (X1) 则指示灯 (Y0) 灭

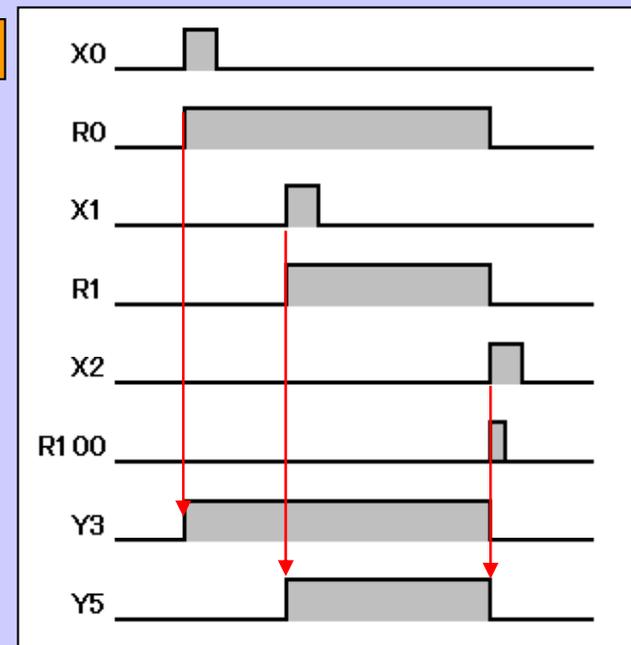
【时序图】



【习题2】

按开关 (X0) 则机械臂 (Y3) 动作, 然后按开关 (X1)
则机械手 (Y5) 动作. 再按开关 (X2) 则机械臂 (Y3)
和机械手 (Y5) 均OFF

【时序图】

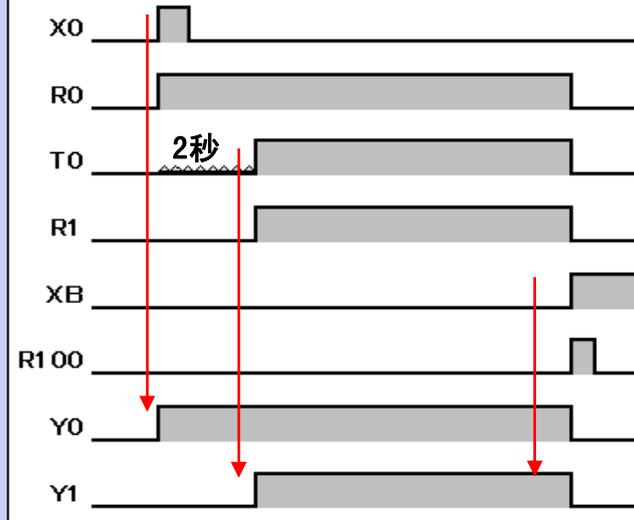


【7-1. PLC基础教程练习题】

【时序图】

【习题3】

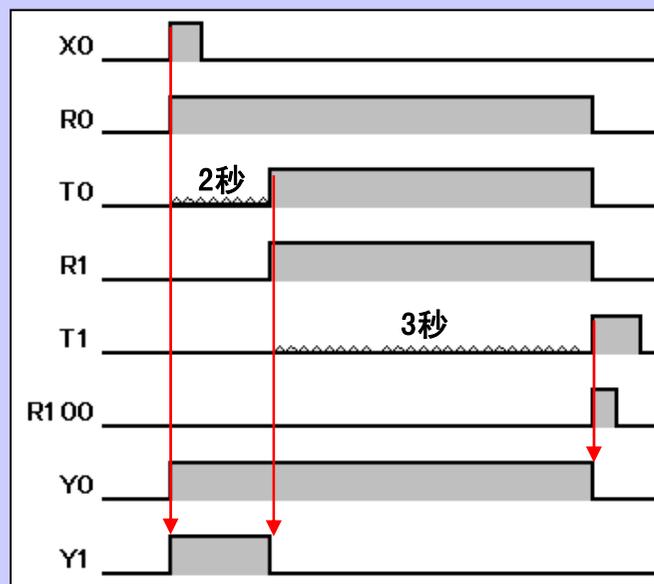
按开关 (X0) 则指示灯 (Y0) 亮
2秒钟后回转台 (Y1) 开始转动
用传感器 (XB) 检测到工件后,
指示灯 (Y0) 及回转台 (Y1) 均OFF



【时序图】

【习题4】

将开关 (X0) 置ON, 则指示灯 (Y0)、
指示灯 (Y2) 亮。
2秒钟后 (Y2) 指示灯熄灭,
3秒钟后 (Y0) 指示灯熄灭

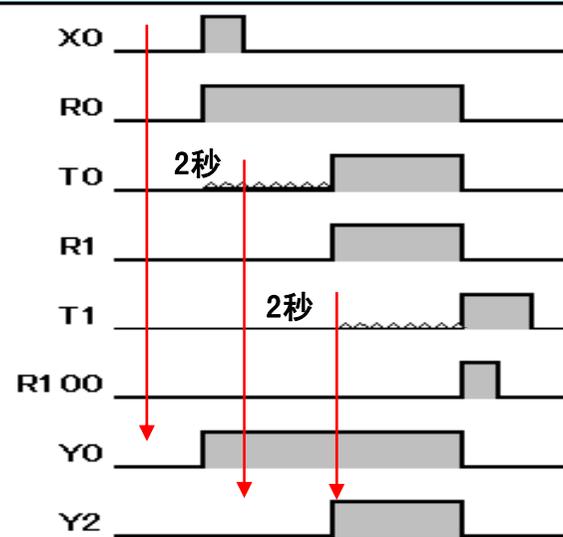


【7-1. PLC基础教程练习题】

【习题5】

将开关（X0）置ON，则（Y0）指示灯亮
2秒钟后，（Y2）指示灯也变亮
再经过2秒钟，（Y0）和（Y2）同时熄灭

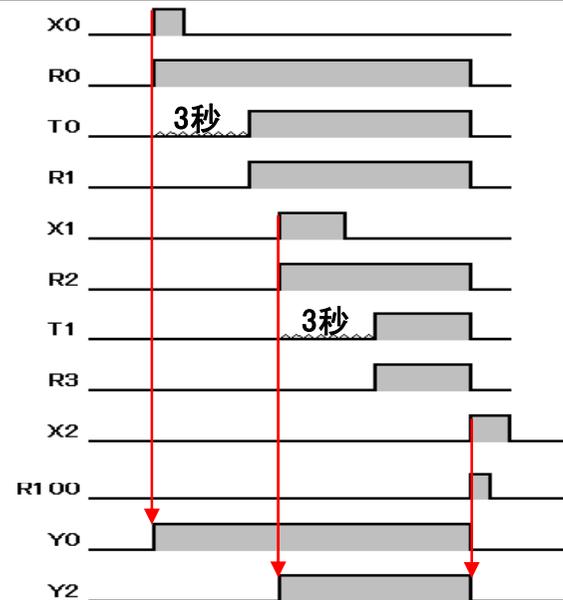
【时序图】



【习题6】

将开关（X0）置ON，则（Y0）指示灯亮
3秒钟以后，开关（X1）ON，则（Y2）指示灯亮
再经过3秒钟后，如果开关（X2）ON，
则（Y0）灯与（Y2）同时熄灭

【时序图】



【信息广场】

请访问以下网站，获取有用信息！

有关网页

松下电工株式会社

<http://www.mew.co.jp/>

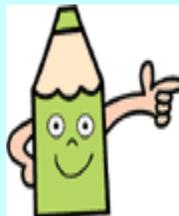
松下电工(中国)有限公司

<http://www.nais-c.com>

与PLC有关的最新信息时时更新！
www.naisplc.com 满载您有用的信息。

<http://www.naisplc.com>

其他



<http://www.mew.co.jp/e-acg>

<http://www.mew.co.jp/c-acg>

新 <http://www.nais-c.com/>

动手吧！补充您所需要的！



Smart Solutions by **NAIS**

预留

【4-3. ANS (堆栈与) · ORS (堆栈或) 指令】

◆ ANS (堆栈与) · ORS (堆栈或)

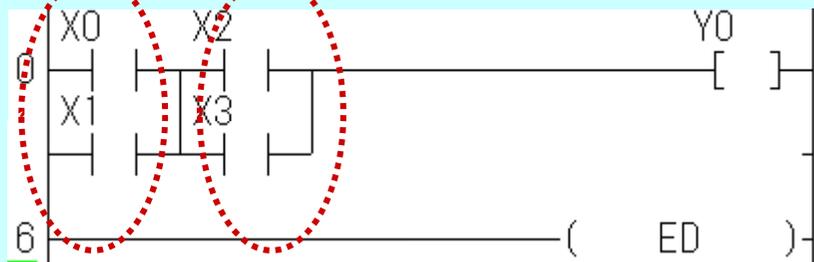
ANS = 把并联回路逻辑程序块串联连接

ORS = 把串联回路逻辑程序块并联连接

逻辑程序块包含了2个以上触点、**在逻辑程序块的起始处使用ST (初始加载)**

【梯形图】

逻辑块① + 逻辑块② → **【ANS】**

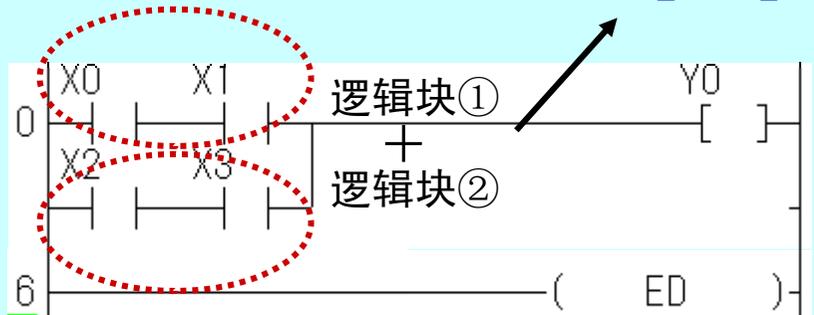


【布尔助记符】

地址	指令	
0	ST X0	①
1	OR X1	②
2	ST X2	
3	OR X3	
4	ANS	①+②
5	OT Y0	

【梯形图】

【ORS】



【布尔助记符】

地址	指令	
0	ST X0	①
1	AN X1	
2	ST X2	②
3	AN X3	
4	ORS	①+②
5	OT Y0	