

PLC基本结构和原理概述

- **教学目的：** 让学生了解PLC的基本结构和组成，了解其基本工作方式；了解PLC的主要性能指标和参数，学会按实际控制要求选用适当的PLC。
- **教学安排：**
 - (1) 旧课复习 (5分钟)
 - (2) 新课讲解 (80分钟)
 - (3) 新课小结 (5分钟)
- **作业：** 课本习题

第十章 可编程序控制器

§ 10.1 概述

§ 10.2 基本概念和编程语言简介

§ 10.3 PLC指令及编程方法

§ 10.4 应用举例

§ 10.5 实验

§ 10.1 概述

10.1.1 什么是PLC ？

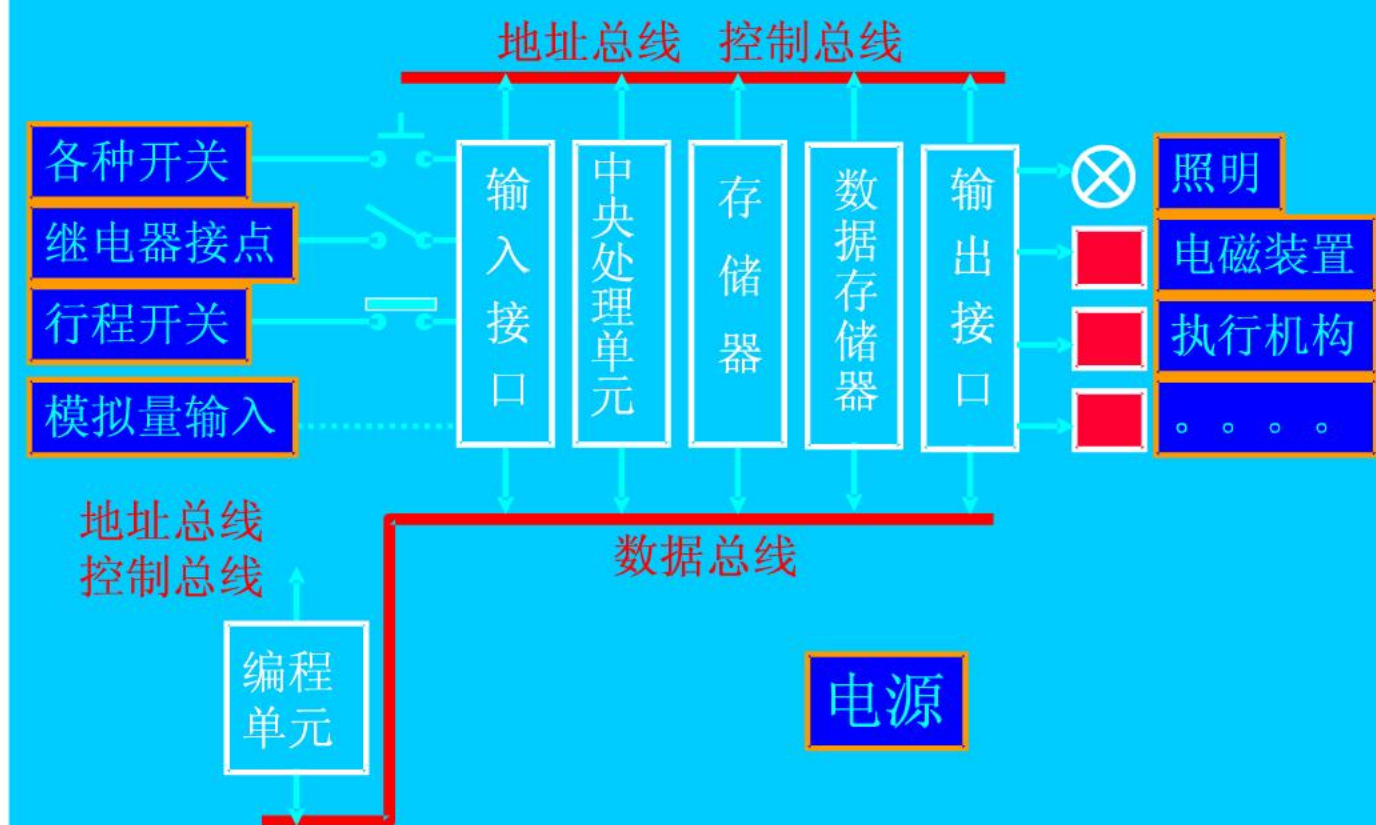
PLC 是一种专门用于工业控制的计算机。

- 早期的**PLC**是用来替代继电器、接触器控制的。它主要 用于顺序控制，只能实现逻辑运算。因此，被称为可编程逻辑控制器 (**Programmable logic controller**, 略写 **PLC**)

- 随着电子技术、计算机技术的迅速发展，可编程控制器的功能已远远超出了顺序控制的范围。被称为可编程控制器 (**Programmable controller**, 略写 **PC**)。为区别于**Personal Computer (PC)**，故沿用**PLC** 这个略写。

10.1.2 结构及工作原理

1. PLC结构示意图



2. 各组成部分的作用

- **CPU:** (1) 将各种输入信号取入存储器。
(2) 编译、执行指令。
(3) 把结果送到输出端。
(4) 响应各种外部设备的请求。
- **存储器:**
 - RAM:** 存储各种暂存数据、中间结果、用户正调试的程序。
 - ROM:** 存放监控程序 and 用户已调试好的程序。

- **输入、输出接口：**采用光电隔离，减小了电磁干扰。

输出三种形式：继电器 -- 低速大功率

可控硅 -- 高速大功率

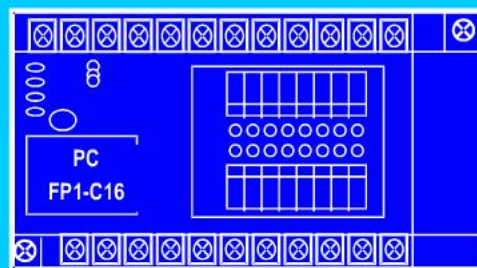
晶体管 -- 高速小功率

- **各种接口、高性能模块：**便于扩展。

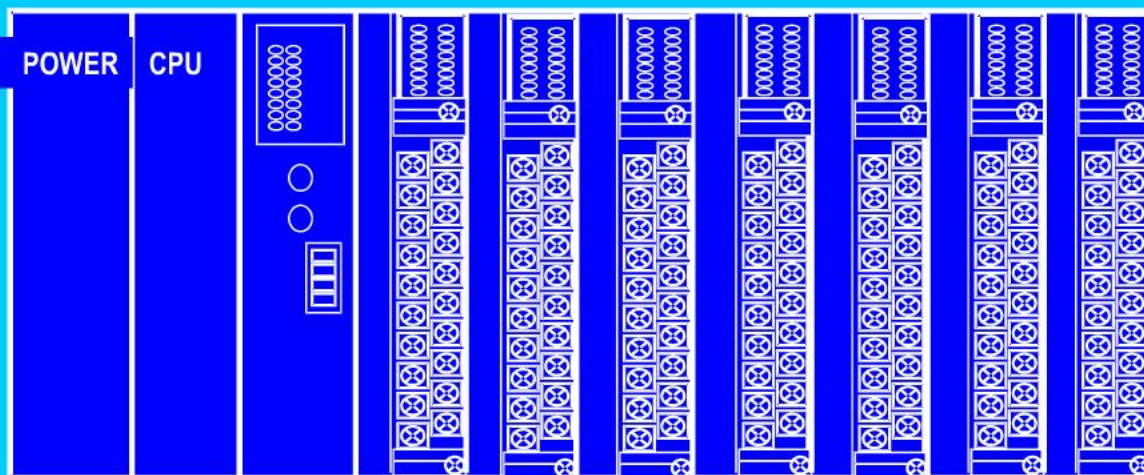
小型机：一体机。有接口可扩展。

中、大型机：模块式。可根据需要在主板上随意组合。

小型机：



中、大型机：



10.1.3 工作方式

微机：等待命令。

PLC：循环扫描。

CPU从第一条指令开始扫描，遇到结束符又返回第一条，不断循环。

这种工作方式
有什么好处？

答：对慢速响应系统，
增强了抗干扰能力。



10.1.4 主要技术性能

1. 输入/输出点数 (I/O点数)。
2. 扫描速度。 单位： ms /1000步 或 μs /步
3. 内存容量。
4. 指令条数。
5. 内部寄存器数目。
6. 高性能模块。

10.1.5 优点

1. 抗干扰、可靠性高。
2. 模块化组合式结构，使用灵活方便。
3. 编程简单，便于普及。
4. 可进行在线修改。
5. 网络通讯功能，便于实现分散式测控系统。
6. 与传统的控制方式比较，线路简单。

10.1.6 应用

1. 用于开关逻辑控制。
2. 用于机加工数字控制。
3. 用于闭环过程控制。
- 4 用于组成多级控制系统。

PLC的基本概念和编程语言

- **教学目的：** 让学生掌握PLC的基本硬件组成，内部各寄存器的作用，介绍PLC的梯形图和语句表的编程方式，要求学生自己能够编写功能框图。
- **教学安排：**
 - (1) 旧课复习 (5分钟)
 - (2) 新课讲解 (80分钟)
 - (3) 新课小结 (5分钟)
- **作业：** 课本习题

§ 10.2 基本概念和编程语言简介

10.2.1 寄存器和接点的概念

PLC的内存除存放用户和系统的程序外，还有四个区：

I/O区：可直接与外部输入、输出端子传递信息

内部辅助寄存器区：存放中间变量

数据区：存放中间结果

专用寄存器区：定时时钟、标志、系统内部的命令

用户在对这四个区进行操作时，可以以**寄存器和/或接点**的方式进行。

以I/O区为例：寄存器是一个16位二进制单元，16位中的每一位是一个接点，对应外部的一个输入/输出端子。



输入寄存器 WX_m

输出寄存器 WY_m

输入端子 X_{mn}

输出端子 Y_{mn}

m : 十进制数, 寄存器编号

n : 16进制数(0~F), 寄存器的第 n 位

例：若X3 为“ON”，则 WX0 的第三位为“1”

若 WY1=7，则表明Y10、Y11、Y12三个接点
“ON”

PLC中有两类接点：常开接点和常闭接点。

符号分别为：



接点通断情况与接点的赋值有关：（以 X0为例：
若 X0的逻辑赋值为“1”，则



I/O区：可以以接点和寄存器的方式对其进行操作。

内部辅助寄存器区：可以以接点和寄存器的方式对其操作。（**Rmn**、**WRm**）

数据区：只能以寄存器的方式进行操作。（**DTm**）

不同型号的**PLC**，其内存分配有所不同。

如：松下电工的**FP1-24**

I区：**X0~X12F** (**WX0~WX12**)

O区：**Y0~Y12F** (**WY0~WY12**)

实际输入端子：**X0~XF**

实际输出端子：**Y0~Y7**

其他的**I/O**区可作为辅助寄存器用。

专用寄存器(FP1机: **WR900~WR903**)。常用的如下:

R900A: “>”标志

R900B: “=”标志

R900C: “>”标志

R9010: 常ON继电器

R9011: 常OFF继电器

R9013: 仅在第一个扫秒周期ON, 其他时候均为OFF

R9014: 仅在第一个扫秒周期OFF, 其他时候均为ON

R9018~R901D: 周期分别为 0.01s, 0.02s, 0.1s, 0.2, 1s, 2s的时钟
脉冲继电器

专用数据寄存器(FP1机: **DT9000~DT9069**)

设置日期时间、高速计数器、步进等指令有关。

10.2.2 编程语言

一. PLC 的编程语言有：

指令表（助记符）语言

梯形图语言

流程图语言

布尔代数语言

} 常用

助记符语言：类似于微机中的汇编语言。

梯形图语言：沿袭了传统的控制图。直观明了，易于掌握。

二. 梯形图的规则:

(1) 梯形图的左边为起始母线，右边为结束母线。

梯形图按从左到右、从上到下的顺序书写。

(2) 梯形图中的接点（对应触头）有两种：

常开（) 和 常闭（)

(3) 输出用 [] 表示，如--[R0]、--[Y0]。一个

输出变量只能输出一次。输出前面必须有接点。

(4) 梯形图中，接点可串可并，但输出只能并不能串。

(5) 程序结束时有结束符 --- (ED)。

§ 10.3 PLC指令及编程方法

各个厂家生产的 **PLC** 产品的指令系统大同小异。编程方法也类似。

以下介绍以松下电工的**PLC**产品为例。

10.3.1、指令的分类

一．按编程器输入指令的方式分类

键盘指令、非键盘指令、高级指令

PLC的基本指令语法

- **教学目的：**让学生熟悉PLC的各种基本指令语法，能够识别各种语法错误，学生能自己编写功能单一的小程序。
- **教学安排：**
 - (1) 旧课复习 (5分钟)
 - (2) 新课讲解 (80分钟)
 - (3) 新课小结 (5分钟)
- **作业：**课本习题

PC FP PROGRAMMER



键盘指令: 可从键盘上直接键入的指令

扩展功能指令: 用F键加功能号方可键入的指令。

非键盘指令: 用指令代码方可输入的指令。



二. 按指令的功能分类

按指令的功能可分为：

- 基本指令
- 数据传送指令
- 算术运算指令
- 位移指令
- 位操作指令
- 数据变换指令
- 转移控制指令
- 特殊控制指令

10.3.2、基本指令

ST: (Start)

从母线开始一个新逻辑行时，或开始一个逻辑块时，输入的第一条指令。

ST: 以常开接点开始 **ST/:** 以常闭接点开始

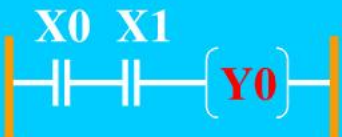
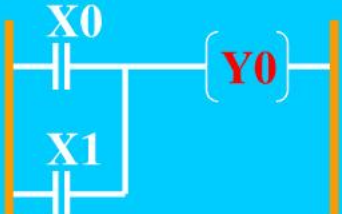
OT: (Output)

表示输出一个变量。

ED: (End) 表示程序无条件结束。

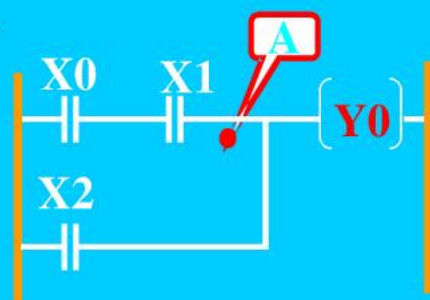
CNED: (Condition end) 程序有条件结束。

NOP: (No-operation) 空操作指令。

逻辑关系	梯形图	助记符
与 AND 当 X0 与 X1 都 “ON” 时， 则输出 Y0 “ON”。		ST X0 AN X1 OT Y0
或 OR 当 X0 或 X1 “ON” 时， 则输出 Y0 “ON”。		ST X0 OR X1 OT Y0
非 NOT 当 X0 “OFF” 时， 则输出 Y0 “ON”。		ST / X0 OT Y0

注意： 与、或、非运算均是对从该指令前面的ST指令到该指令的前一个指令处的结果进行运算。

例：

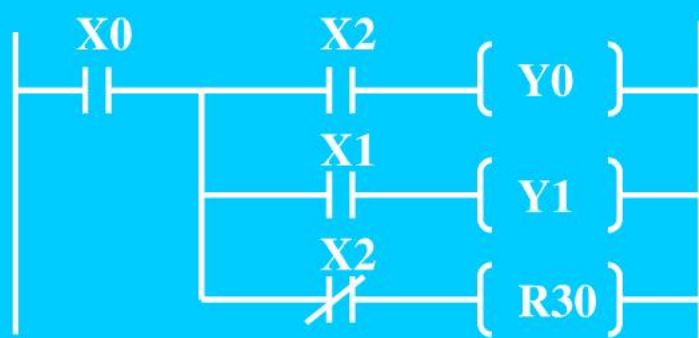


ST	X0
AN	X1
OR	X2
OT	Y0

X2是与图中A点处的结果（即**X0**与**X1**的结果）相或，而不是与**X1**相或。

逻辑关系	梯形图	助记符
区块与 AND STACK 当 “X0 或 X1”与 “X2 或X3” 都 “ON” 时， 则输出 Y0 “ON”。		ST X0 OR X1 ST X2 OR X3 ANS OT Y0
区块或 OR STACK 当 “X0 与 X1”或 “X2 与 X3非” “ON” 时，则输出 Y0 “ON”。		ST X0 AN X1 ST X2 AN / X3 ORS OT Y0

PSHS, RDS, POPS (栈指令)



ST X0

PSHS

AN X2

OT Y0

RDS

AN X1

OT Y1

POPS

AN/ X2

OT R30

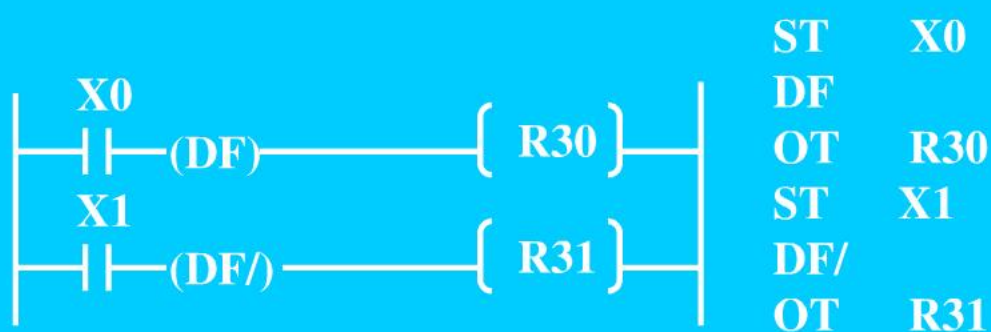
功能解释

PSHS (Push Stack): 将结果存入堆栈

RDS (Read Stack): 从堆栈读数

POPS (Pop Stack): 从堆栈读数并清空堆栈

DF, DF/: 微分指令



功能解释

X0接通瞬间（上升沿），**R30**接点接通一个扫描周期。

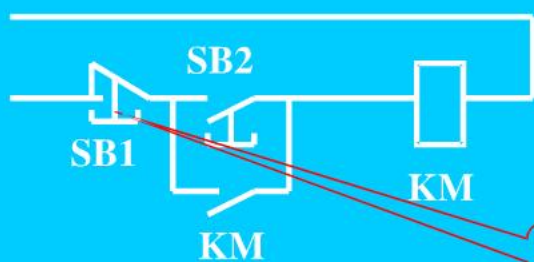
X1断开瞬间（下降沿），**R31**接点接通一个扫描周期。

指令使用

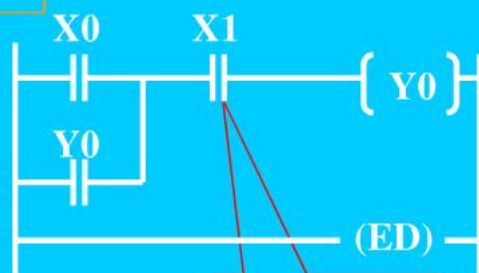
当只需要信号的上升或下降沿时使用。例：启动或停车按钮信号。

例1：直接启动停车控制

程序：



控制电路图



常闭按钮

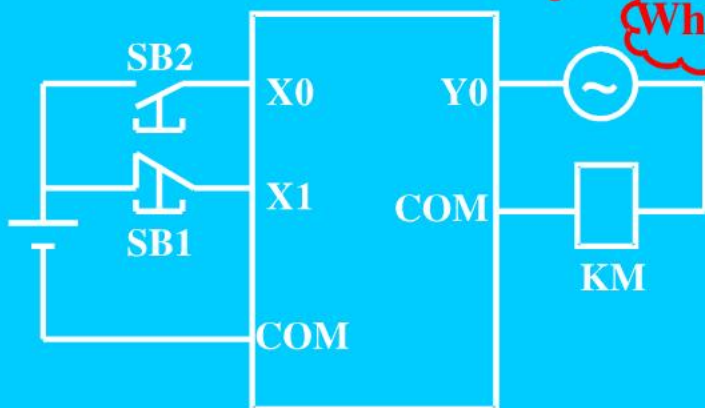
常开接点

I/O分配：

X0：启动

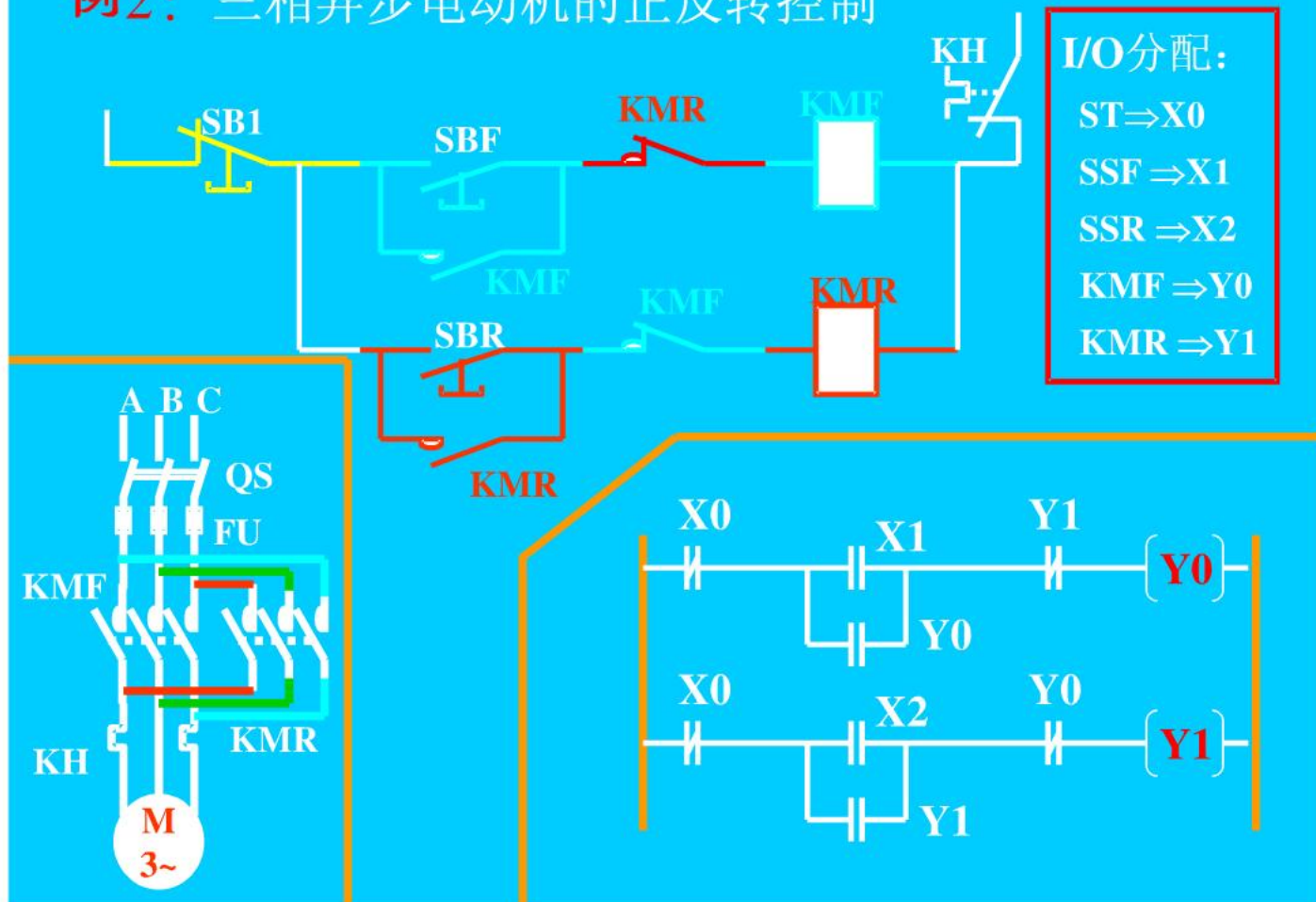
X1：停车

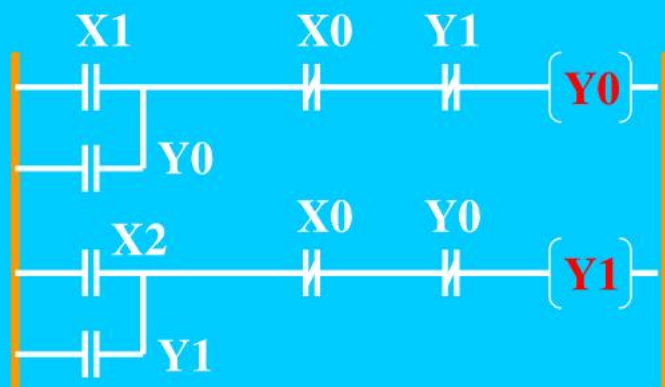
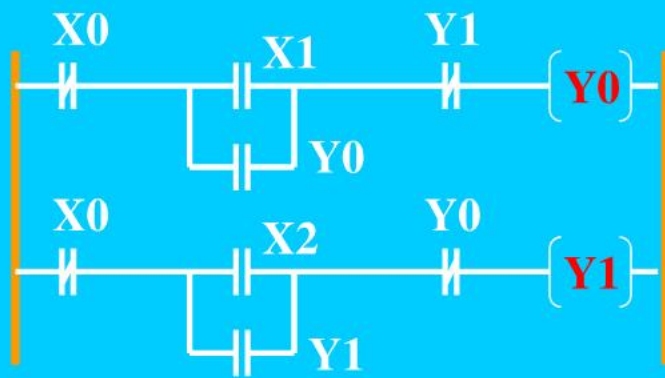
Y0：KM



PLC外部接线图

例2：三相异步电动机的正反转控制





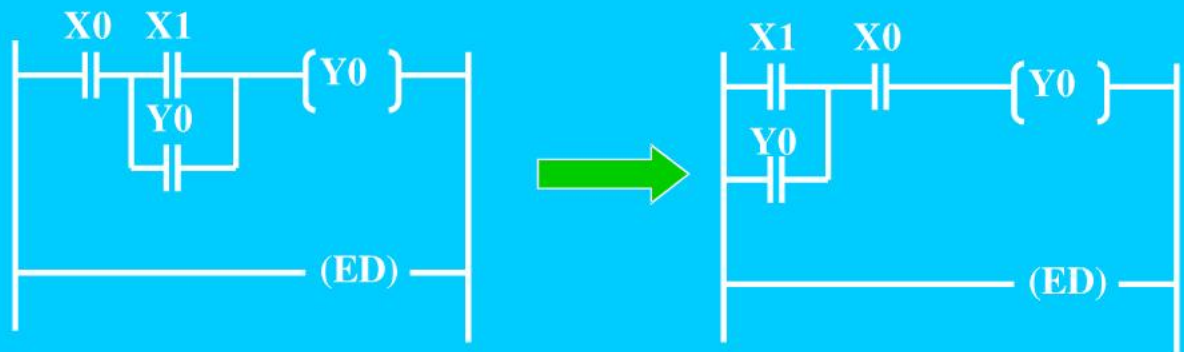
```

ST    X1
OR    Y0
AN /  X0
AN /  Y1
OT    Y1
ST    X2
OR    Y1
AN /  X0
AN /  Y0
OT    Y0
ED

```

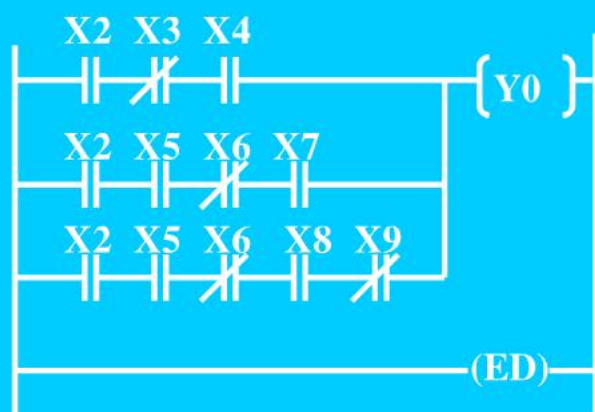
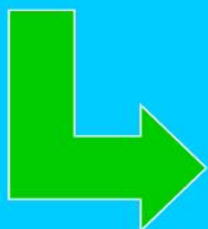
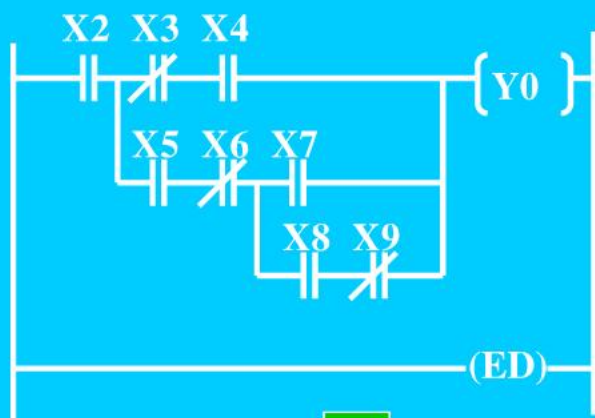
编程中应注意的几个问题

1.用电路变换简化程序(减少指令的条数)



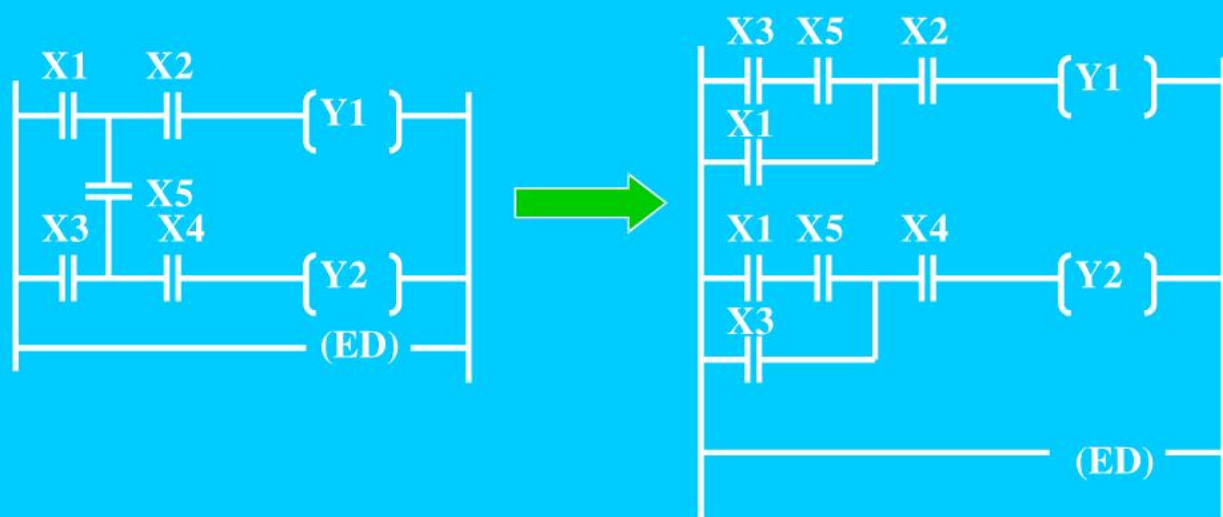
编程中应注意的几个问题

2. 逻辑关系应尽量清楚(避免左轻右重)



编程中应注意的几个问题

3.避免出现无法编程的梯形图



KP: (Keep)

置位信号: 例X0



ST X0

ST X1

KP R0

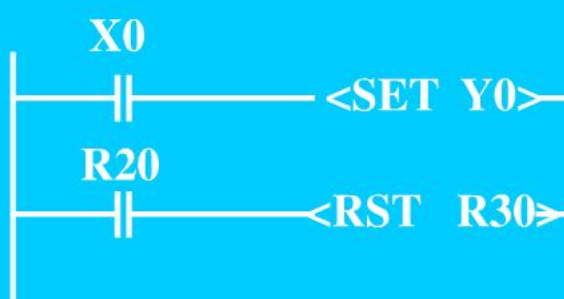
说明:

- (1) 在置位信号接通的瞬间, **R0**置**1**。以后无论置位信号状态如何, 只要复位信号断开, **R0**的状态均为**1**。
- (2) 在复位信号接通的瞬间, **R0**置**0**。
- (3) 在复位信号和置位信号同时接通时, 复位优先。

SET、RST（Reset）

这两条指令的功能类似于**KP**指令，但使用比**KP**指令灵活。

例：



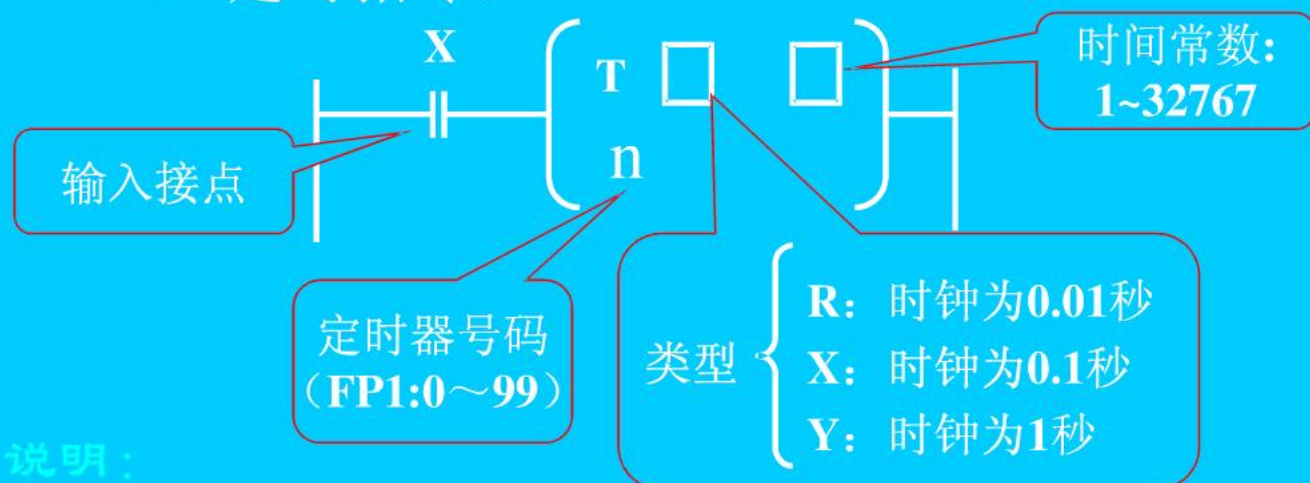
ST X0

SET Y0

ST R20

RST R30

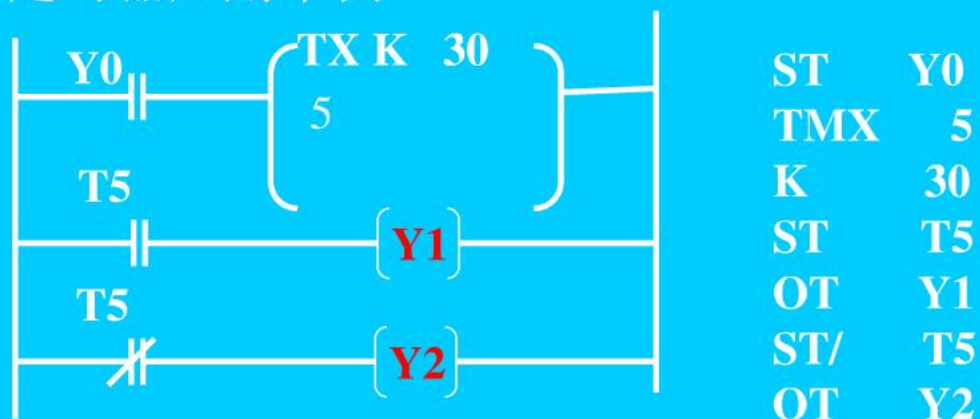
TM: (定时指令)



说明:

- (1) 时间常数与类型一起确定了定时的时间。
- (2) 定时器为减计数。当输入接点接通时，每来一个时钟脉冲减1，直到减为0。这时，定时器的常开接点闭合，常闭接点断开。当输入接点断开时，定时器复位。

例：定时器应用举例



动作说明：

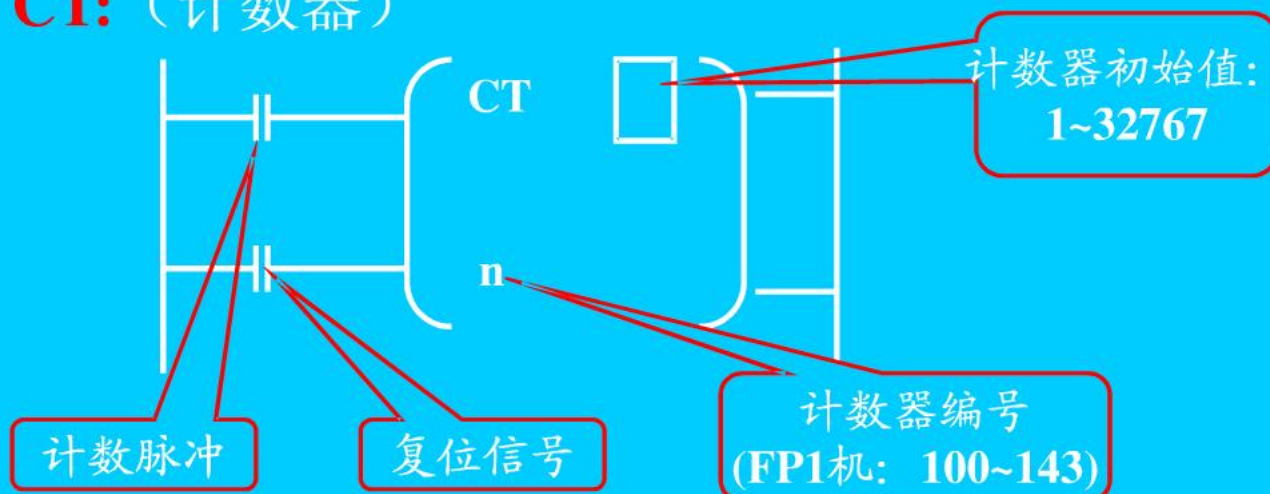
当Y0闭合后，定时器TM5开始计时。经过 $30 \times 0.1 = 3s$ 后，Y1闭合，Y2断开。

与定时器有关的两个寄存器：**EVn** 和**SVn**
(n为寄存器编号, 该编号与定时器编号对应)

EVn: 存储定时器**TMn**的过程值。

SVn: 存储定时器**TMn**的设置值。

CT: (计数器)



说明:

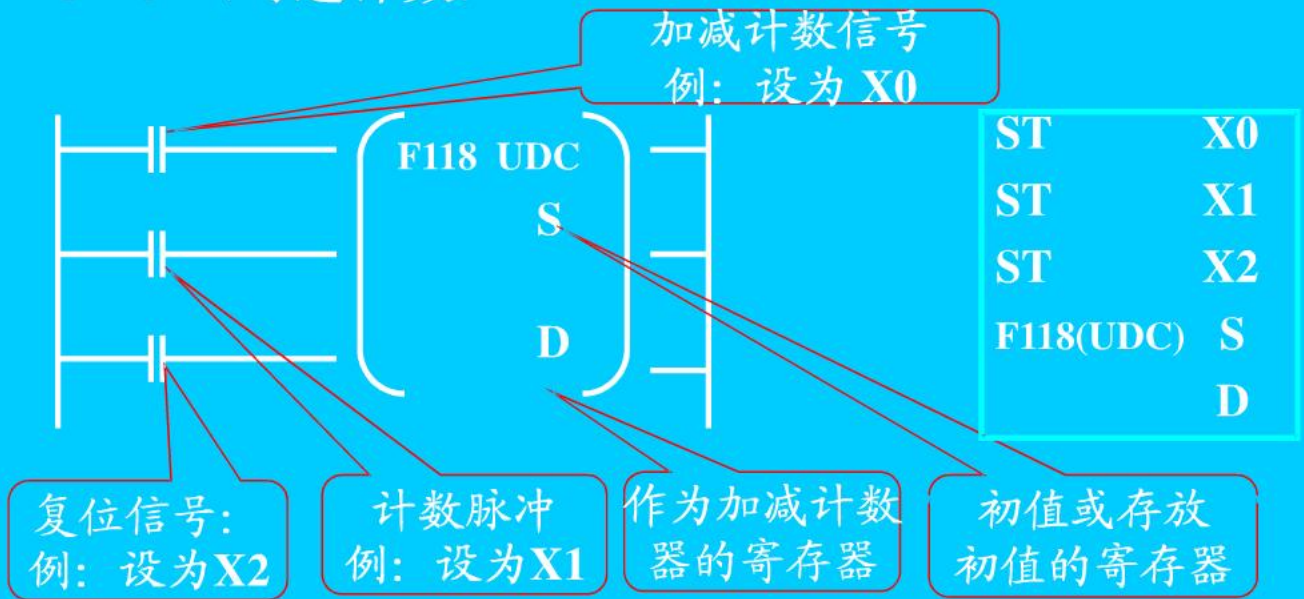
- (1) 复位信号接通时，计数器复位，装入初始值。
- (2) 复位信号断开时，每来一个计数脉冲减1，直到减为0，计数器的常开接点接通，常闭接点断开。

与计数器有关的两个寄存器：**EVn** 和**SVn**
(n为寄存器编号, 该编号与计数器编号对应)

EVn: 存储计数器 **CTn** 的过程值。

SVn: 存储计数器 **CTn** 的设置值。

UDC: (可逆计数)



	WX	WY	WR	SV	EV	DT	IX	IY	常数
S	√	√	√	√	√	√	√	√	√
D		√	√	√	√	√	√	√	

10.3.3、其他常用指令简介

一. 数据传送指令:

F0(MV) 、 F1(MV/)

 (F0 MV S D)

功能说明: **S**为常数或寄存器, **D** 为寄存器。当条件满足时, **S**或**S**的内容传送到**D**。

F0(MV): 把 **S** 或 **S** 的内容传送到**D**。

F1(MV/): 把 **S** 或 **S** 的内容求反后传送到**D**。

例：某控制控制系统，若出现严重故障信号，则所有的动作全停，故障指示显示。

I/O分配：

X0：故障信号

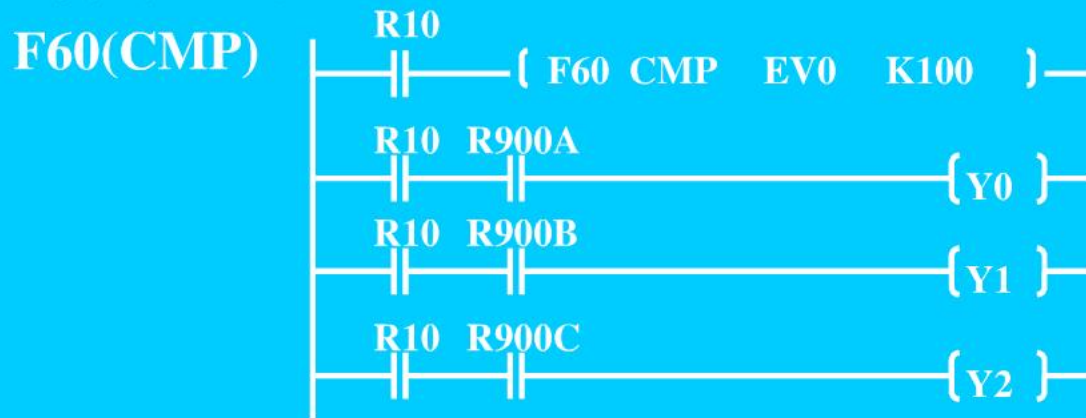
Y0：故障指示

其他输出端：**Y1~YF**



其他传送指令常用的有：**F10(BKMV)**，**F11(COPY)**等等。

二. 算术指令:



功能说明：当**R10**接通时，比较**EV0**和**100**，若**EV0>100**，则**Y0**接通；若**EV0=100**，则**Y1**接通；若**EV0<100**，则**Y2**接通。

其它的算术运算指令，如算术运算、**BCD**码运算、比较指令等等。

例：设计一个顺序控制电路：

启动时，电机**D1**先启动，
3s后电机**D2**启动，再过
5s后电机**D3**启动。

方案一

I/O分配：

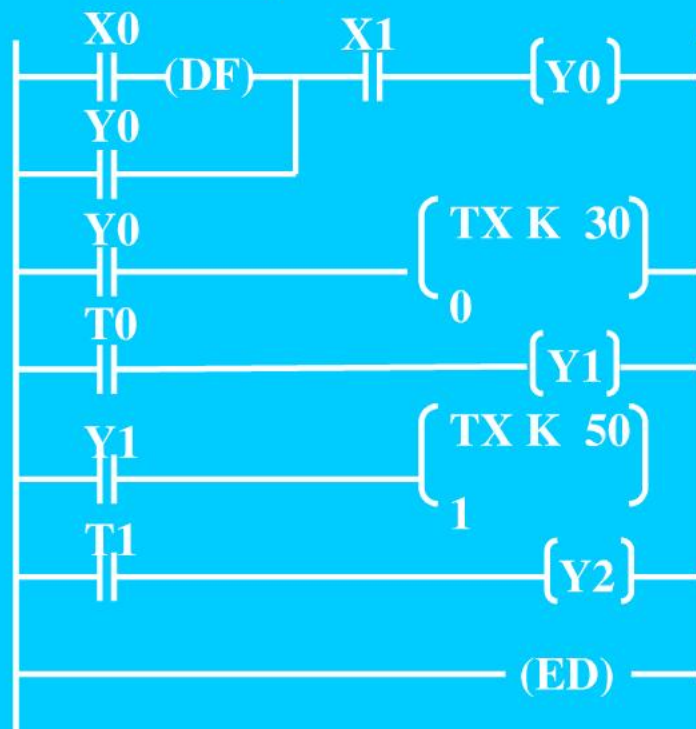
X0： 启动按钮

X1： 停车按钮

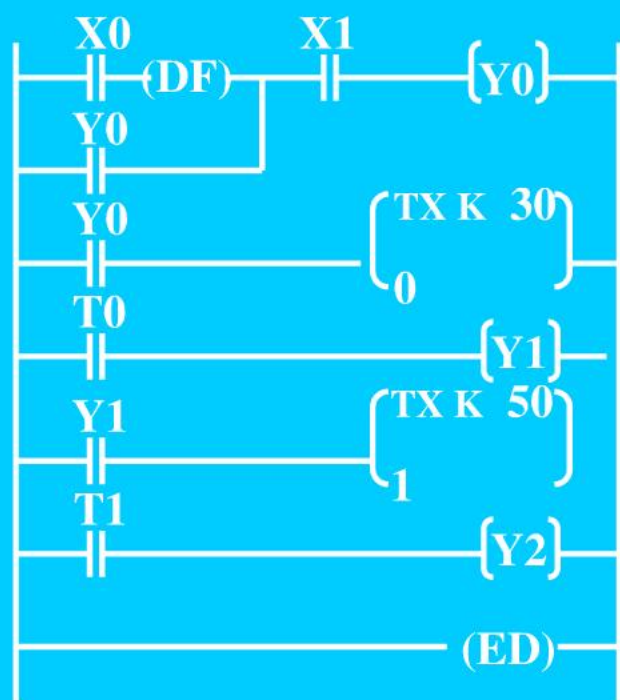
Y0： 电机**D1**

Y1： 电机**D2**

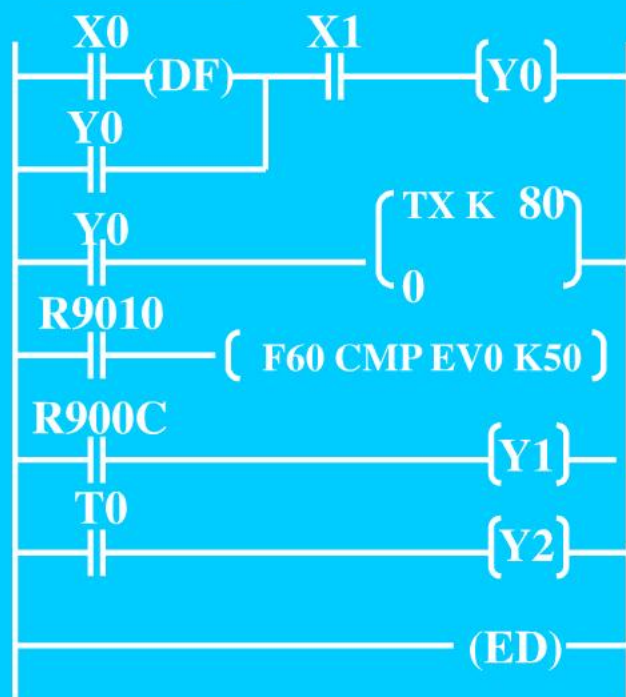
Y2： 电机**D3**



方案一

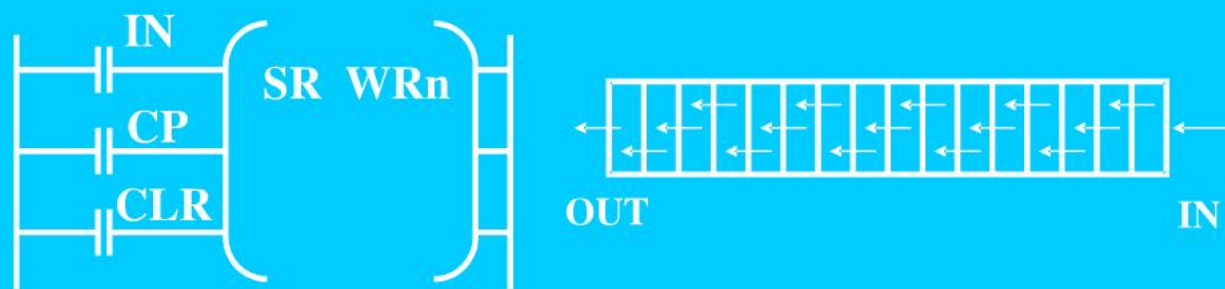


方案二



三. 移位指令：

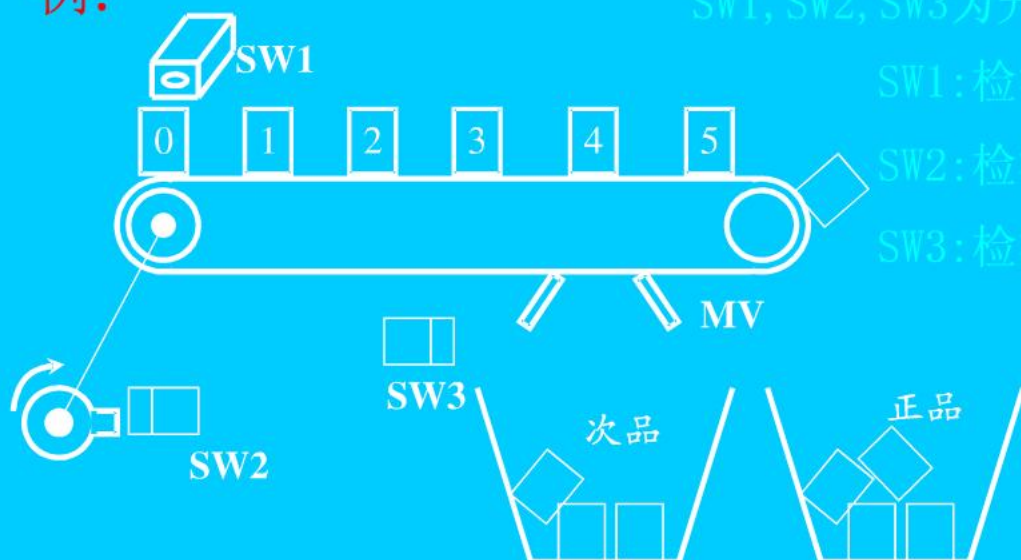
(1) 左移指令 SR



说明：

- (1) **SR**指令只能对**WR**型寄存器进行移位。
- (2) **IN**：串行输入端（补位），接点通，补**1**，接点断，补**0**。
- (3) **CP**：移位脉冲
- (4) **CLR**：复位清零。

例：



I/O分配：

X0: SW1

X1: SW2

X2: SW3

X3: SB

Y0: MV

凸轮每转一圈发出一个脉冲，且每个脉冲过一个物品。当次品移到4号位时，电磁阀MV打开，次品落入次品箱内。无次品则自动掉入正品箱内。有一复位按钮SB实现手动复位。

I/O分配:

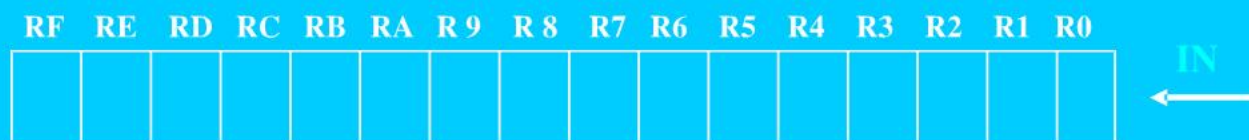
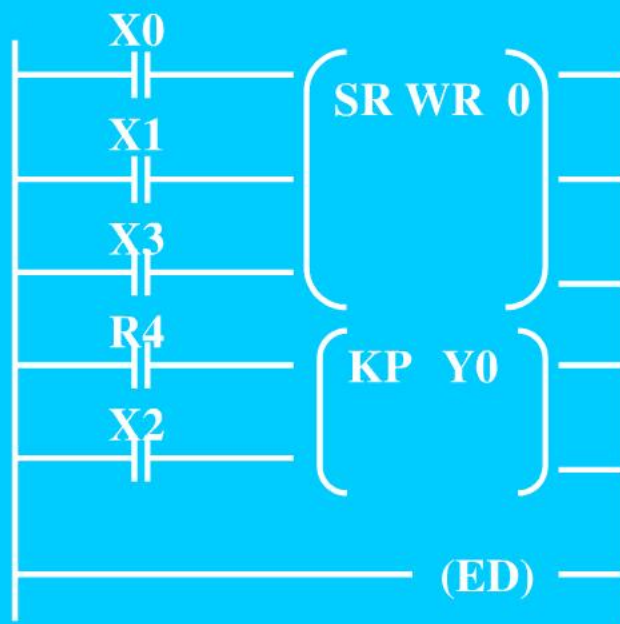
X0: SW1 检测有无次品

X1: SW2
检测凸轮的突起

X2: SW3
检测有无次品落下

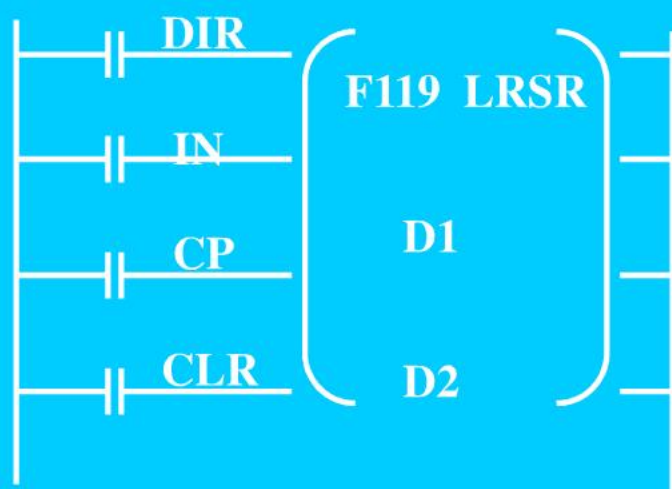
X3: SB 复位

Y0: MV



无次品时该位
总为0

(2) 双向移位指令 LRSR



DIR: 移位方向。

ON:左移 **OFF:**右移

IN: 串行输入端

CP: 移位脉冲

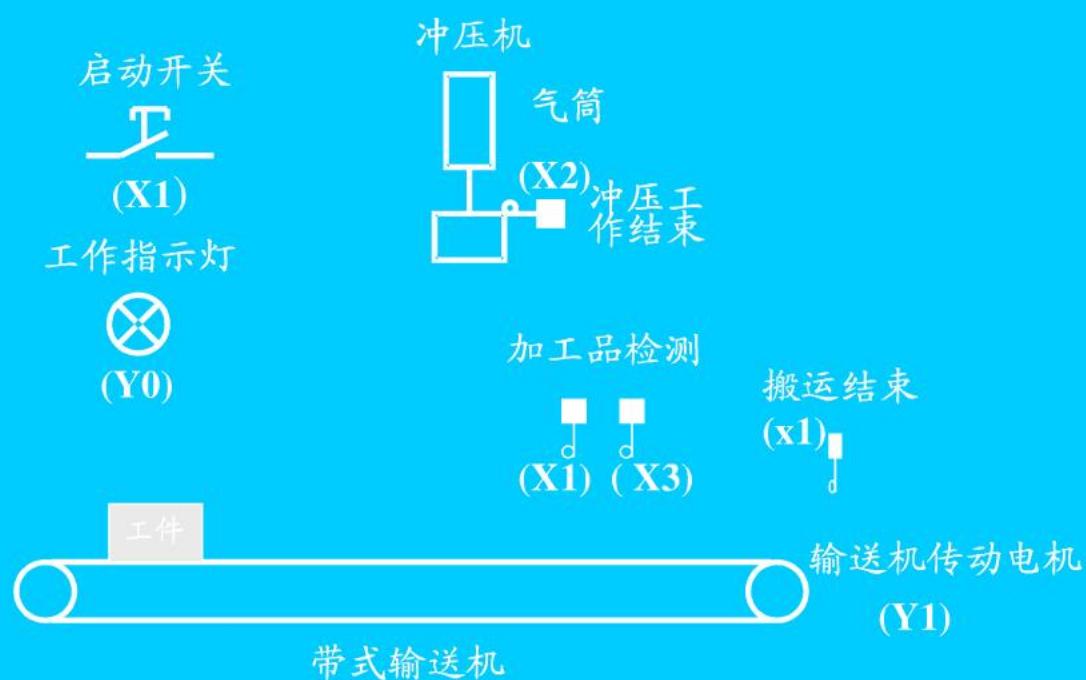
CLR: 复位清零

说明：**D1**，**D2**为移位区的首址和末址。可使用的

寄存器有：**WY, WR, SV, EV, DT**

§ 10.4 应用举例

例一：冲压机控制程序



例二：楼宇电梯自动控制



顺序控制

例三：电机变速及精密位置控制



其他信号有：

启动、 停车、 紧急停车、运动方向控制等等

例四：在智能建筑技术中的应用

智能建筑的七个层次

- (7) 楼宇专用信息
- (6) 楼宇共用信息处理
- (5) 楼宇运营管理
- (4) 楼宇设备自动控制
- (3) 楼宇通讯
- (2) 智能建筑环境
- (1) 一般建筑

楼宇自动化：

- (1) 保安
- (2) 设备管理

PLC在其中的应用：

- (1) 防盗、防火系统
- (2) 供热、供气、空调、照明系统

(一) 办公自动化

(二) 楼宇自动化网络

例五：在大型车库管理中的应用

管理分为三级：

- (1) 中央监控级：一台**PC**及其外围设备。
- (2) 管理级：由总出入口的**PLC**和各分库的 **PLC** 构成。可监控全库车辆的进出情况，同时显示各分库是否有空车位。
- (3) 现场监控级：每个分库的**PLC**用远程 **I/O**单元检测车辆的出入情况。

例六：养鱼场管理中的应用

对鱼池中的水质、水温、流量等进行监测及控制，
以实现科学养殖

例七：24小时环境监测

系统分三级：中央监控级、数据采集及管理级、现场检测级。

PLC进行现场检测：传感器检测的各种信号经**A/D**转换后，送至上位**PLC**及数据采集用的**PC**机。经数据处理后送中央监控级，再由中央机发出命令，送到管理用的**PC**机进行处理。

§ 10.5 实验

10.5.1 PLC的三种工作方式：program, remote, run

编程时，把开关拨到 **PROG.** 处。

运行时，把开关拨到 **RUN** 处。

与PC机通讯时，开关拨至 **REMOTE** 处。

若开关在 **RUN** 处就试图修改程序，则蜂鸣器响。这时，按 **ACLR** 键 或 **CLR** 键，然后把开关拨到正确位置。

10.5.2 指令输入:

1. 每输入一条指令按一下  键存入。
2. I/O功能键有多种功能, 如: , 输入的是什么由编程器自动判断。
3. 操作键有多种功能, 如 , 用  键切换。

PC FP PROGRAMMER



I/O指令

键盘指令: 可从键盘上直接键入

非键盘指令: 用指令代码输入



若不知道指令代码则用



调出非键盘指令表, 用▲或▼找出代码。

扩展功能指令: 用F键加功能号键入

若指令后有操作数, 则在最后一个操作数之前, 每输入一个按一下ENT键, 最后一个操作数输入后按WRT键。

操作键

编程步骤:

(1) 清除老程序: ACLR (-) 0 END SC (DELT) INST

(2) 开始编程: 0 READ 显示: 0 NOP

(3) 输入程序:
(例)

ST X•WX	ST X•WX	1	WRT	ST X 1
OR R•WR	AN Y•WY	0	WRT	OR Y 0
TM T•SV	ST X•WX	5	ENT	TM X5
BIN K/H	3	0	WRT	K 30

(4) 输入结束行: SC 1 0 SC WRT END

实验预习要求:

1. 指令的输入方法

- 键盘指令
- 非键盘指令
- 扩展功能指令

2. 如何查找相关内容

- 查看某一地址的指令内容
- 查找某接点或寄存器
- 查找某条指令的地址

3. 如何修改程序

- 检查程序
- 删除指令
- 修改指令
- 插入指令

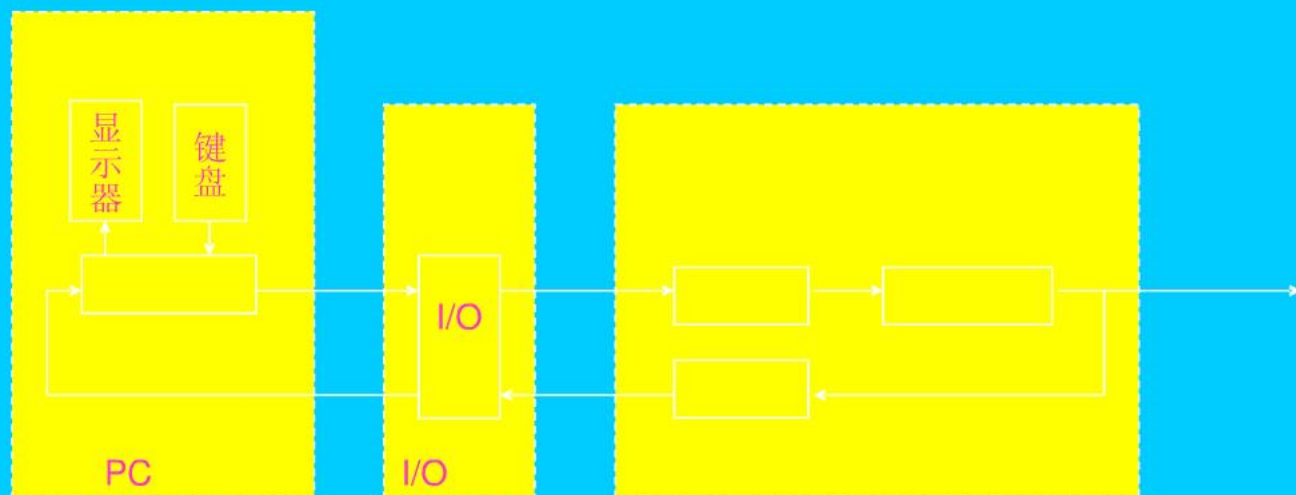
4. 如何调试程序

- 查看接点的通断情况
- 查看和/或更改寄存器的内容

§ 10.6 组态技术简介

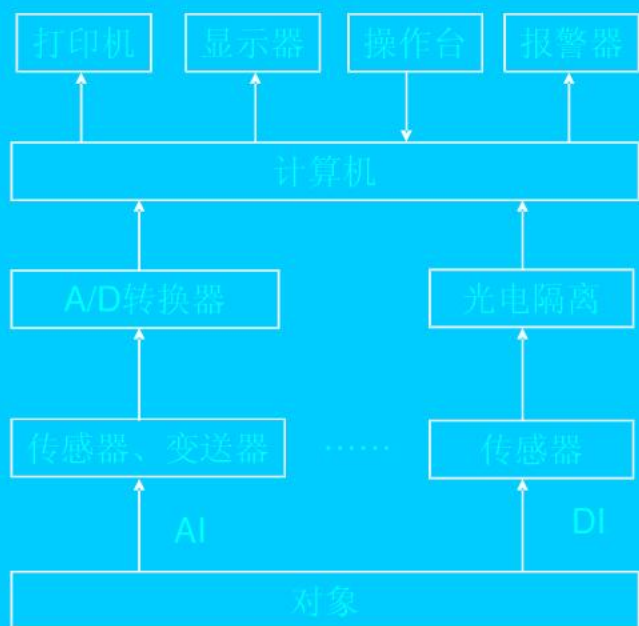
1.1 概述

1.1.1 组成

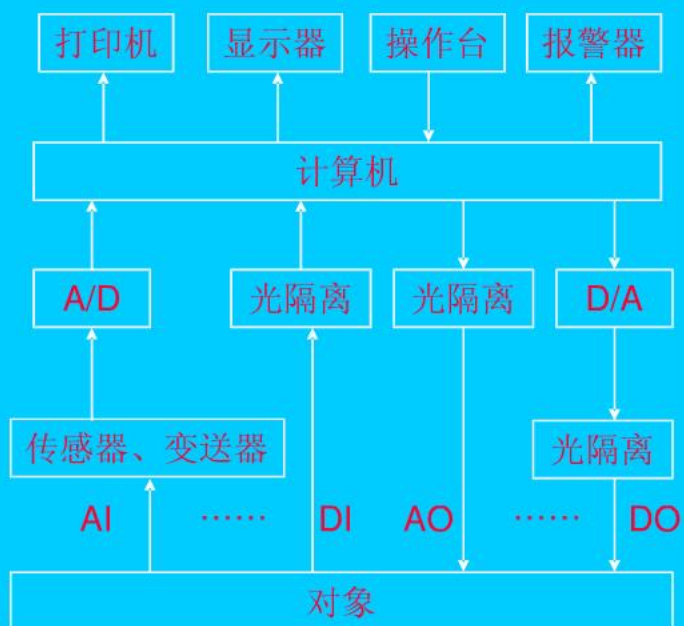


1.1.2 分类

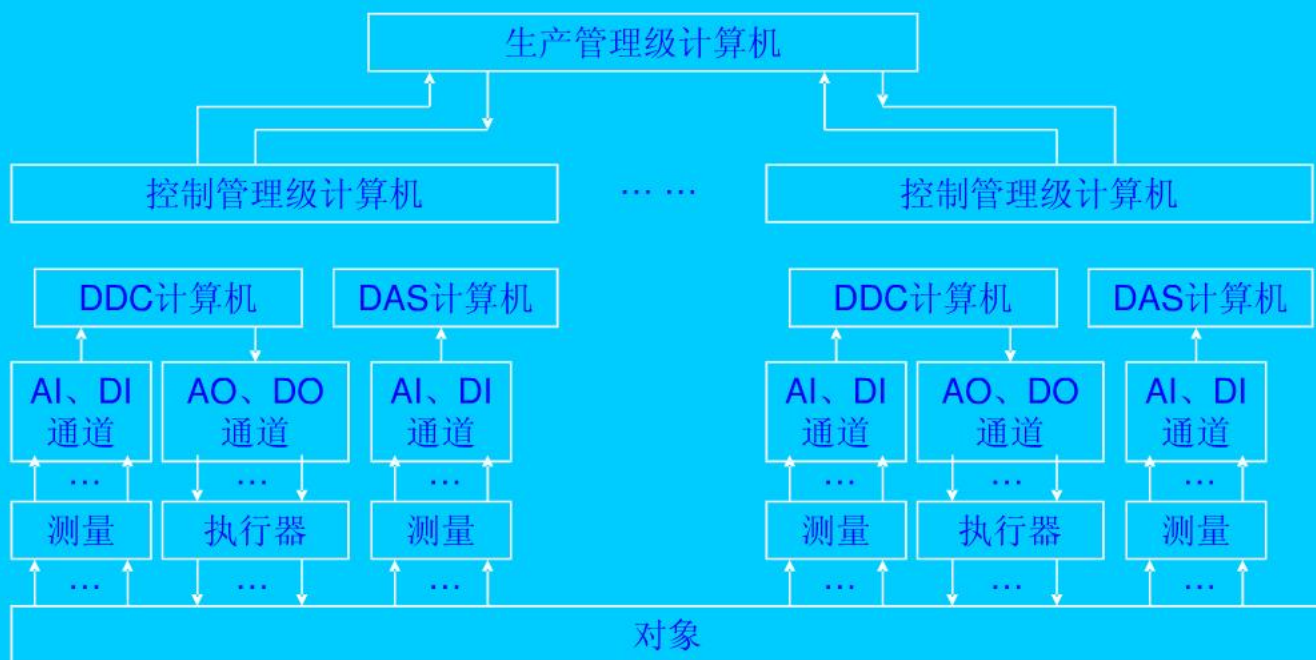
1、按功能分（三类）：



数据采集系统（DAS）



直接数字控制系统（DDC）



分布式控制系统DCS系统

2、按设计方法分：

以MCU为核心、以PCL为核心、以IPC为核心

1.1.3 优点

组态技术计算机技术综合发展的结果，是技术成熟化的标志。由于组态技术的介入，计算机控制系统的应用速度大大加快。

1.2 常用软件

专用DCS：德国西门子公司、日本三菱、台湾研华、中国时利和

通用组态软件：Intouch、WinCC、KingView、MCGS

1.3 功能

为用户提供数据采集与处理、画面设计、动画显示、报表输出、报警处理、流程控制等功能。

1.4 组态软件工程一般组建过程

1、工程项目系统分析

了解整个工程系统构成和工艺流程，明确主要的监控要求和技术要求。在此基础上，拟定组建工程的总体规划和设想。如：功能的实现、用户界面、动画效果、数据库与各界面之间的关系。做好工程的整体规划，在项目的组态过程中能够尽量避免一些无谓的劳动，快速有效地完成工程项目。

2、设计用户操作菜单

根据用户需要，随时对菜单的内容进行增加和删减，最终确定菜单。

3、制作动态临控画面

制作动态临控画面是组态软件的最终目的，界面必须友好，所有的动画效果要与数据库一一对应，实现内外结合的效果。

4、编写控制流程程序

动态画面制作过程中，除一些简单的动画由图形语言定义外，大多数较复杂的动画效果和数据之间的链接，都是通过一些应用程序命令或脚本程序来实现的。

5、完善菜单按钮功能

对一些功能比较强大、关联比较多的控制系统，还要通过制定一些按钮或链接其他画面以及执行某些命令。这样大大增加其数据传输的简捷性。

6、编写程序调试工程

工程中的用户程序编写好后，要进行在线的调试。在进行现场调试过程中，可以先借助一些模拟手段进行初调。以便增加程序的可靠性，减少综合测试的难度。

7、连接设备驱动程序

最后，要实现和外部设备的连接，在进行连通时，必须装入正确的驱动和定义通信协议是非常重要的。程序不能正常运作，很大程度上是通信通信协议的设置问题而造成的。

另外，合理地指定内部变量和外部变量也系非常重要的。

8、工程完工综合测试

经过上述的分步调试后，就可以对系统进行整体的连续调试了，一个好的工程必须要能够经得起考验，验收合格后就可以进行效验。为了保障工程技术人员开发劳动成果，各组态软件为用户提供相应的完善的保护措施，充分保护了知识产权的合法权益。

2.1 KingView工程浏览器

1、画面制作与管理

构建工业控制现场的画面，也是应用系统的主框架，它展现了工程的总体外观。

2、命令语言

通过一些应用程序命令或类C语言脚本程序来实现各种复杂的动画效果。

3、数据库

实时数据库是以数据对象为核心，也是应用系统的数据处理中心，是实现各个部分协调动作的源泉依据。

4、设备

连接和驱动外部设备工作环境的重要窗口，从而达到实时在线通信的目的。

5、配置

对不同应用场合进行界面、数据记录、权限、报警等多方面的操作

6、配方管理

描述生产产品所用的不同配料之间的比例关系，使用户提高生产效率

2.2 图形编辑工具

画面系用户监视和操作系统的人机界面。一个设计良好的画面必须能够反映整个系统的组成、结构，能让监控操作人员方便地掌握系统状态，并进行正确的操作。

2.3 I/O设备管理

组态王与 I / O 设备之间的数据交换采用以下五种方式：串行通信方式、板卡方式、网络模块、人机接口卡方式、DDE方式。

开发人员只需要按照设备向导提示完成各种I/O设备配置工作。如：添加新设备、删除设备、引用设备。

2.4 变量

1、变量类型及定义

基本变量类型

定义了内存变量、I/O变量，其取值分四类：离散型、实型、字符串行型、长整型。

结构变量，可以包含多个基本变量类型。

特殊变量对象

报警窗口变量、历史趋势曲线变量、系统预设变量。

2、变量属性及其配置

3、数据库的维护

2.5 动画连接

1、类型

属性变化连接、位置与大小变化连接、值输出连接、用户输入连接、特殊动画连接、滑动输入连接、命令语言连接。

2、步骤

- (1) 创建或选择连接对象
- (2) 双击图形对象，弹出“动画连接”对话框
- (3) 选择对象想要进行的连接
- (4) 为连接定义详细资料

2.6 趋势曲线与报警窗口

1、实时趋势曲线

实时趋势用于实时显示数据的变化情况。在画面运行时实时趋势曲线对象由系统自动更新。数据将从右向左移动。

2、历史趋势曲线

历史趋势用于对过去生产情况显示数据的变化情况。在画面运行时，数据不会自动更新，只能通过命令语言结合按钮对其进行控制。

3、报警窗口

运行报警和事件记录是控制软件必不可少的功能，当变量的数据或数据变化的变化异常时将产生报警，以便操作者采取必要的措施。

2.7 控制与控件函数

1、概述

控件是可重用的对象，用来执行专门的任务，其实质第一个窗口，但不是一个独立的窗口，通过控件属性，完成各种特定的功能。

2、棒图控件控件

棒图控件包括：二维棒图、三维棒图或二维饼形图，用于数据变量的显示。

3、温控曲线控件

温控曲线反映出实际测量值按设定曲线的变化情况。此控件主要适用于温度控制、流量控制等。

4、X-Y轴曲线控件

与温控曲线用途相似。

5、窗口类控件

提供的窗口类控件有：列表框控件、组合框控件、复选框控件、编辑框控件、单选按钮控件。这些控件的作用和操作方法与Windows操作系统中相应的标准窗口类控件相同。

6、超级文本显示控件

用于显示RTF格式或TXT格式的文件，而且也可在超级文本显示控件中输入文本字符串，然后将其存入到指定的文件中。

7、多媒体控件

提供AVI动画和视频输出等多个多媒体控件，用于播放图形动画和实现视频监控。

8、Active X 控件

支持Windows 标准的Active X控件，包括用户自制的Active X控件。为用户可以灵活地编制符合自身的控件或调用一个已有的标准来完成一项复杂的任务，而无须组态王做大量的复杂的工作。

2.7 命令语言

1、概述

能让用户定义命令语言驱动应用程序。包括：应用程序命令语言、热键命令语言、事件命令语言、数据改变命令语言、自定义函数语言和画面命令语言等。其具有完备的词法、语法查错功能和丰富的运算符、字符串函数、控件函数、**SQL**函数和系统函数。

2、句法

可以进行赋值、比较、数学运算，还提供了可执行的判断及循环表达式的逻辑操作能力。

组态王支持使用内建复杂的函数。关于各个函数的详细的解释，参阅“组态王用户手册”及“组态王函数手册”。

2.7 安全机制

1、系统安全管理

对画面上的图形对象设置访问权限，同时给操作者分配访问优先级和安全区。当操作者小于该对象访问权限时，该对象不能访问。

2、记录操作过程

记录程序运行中操作员的所有操作。只要用户定义了记录报警和事件文件，在运行时，用户的登录、注销和对变量的操作等事件都记录在报警事件文件中，以便设备或操作不当所造成的后果有迹可寻。

2.7 报表处理

数据报表反映生产过程中的数据、状态等，是对数据进行记录的一种重要形式。是生产过程必不可少的一部分。它既能反映系统实时的生产情况，也能对长期的生产过程进行统计、分析，使管理人员能够实时掌握和分析生产情况。

2.8 网络功能

完全基于网络的概念，是一种真正的C/S模式，支持分布式历史数据库和分布式报警系统，可运行在基于TCP/IP协议的网络上，使用户能够实现上、下位机以及更高层次的厂级连网。

用户可根据系统需要设立专门的I/O服务器、历史数据服务器、报警服务器、登录服务器和Web服务器等。