

---

# 可编程控制器原理及应用

## 实验指导书

上海交通大学

电气工程与自动化实验室

2012 年 3 月

郑重声明：版权属于上海交通大学电气工程实验中心，仅供上海交通大学电气工程系课程：可编程控制器原理及应用使用。未经允许不得随意上传至百度文库、论坛等网络媒体。

---

|                                 |               |
|---------------------------------|---------------|
| <b>第一章 可编程控制器简介 .....</b>       | <b>- 1 -</b>  |
| 一、PLC 的结构及各部分作用 .....           | - 2 -         |
| 二、PLC 的工作原理 .....               | - 3 -         |
| 三、TWDLCAE40DRF 的主要功能 .....      | - 4 -         |
| <b>第二章 可编程控制器的软件介绍及使用 .....</b> | <b>- 5 -</b>  |
| 一、软件概述 .....                    | - 5 -         |
| 二、编程语言描述 .....                  | - 6 -         |
| 三、指令系统 .....                    | - 9 -         |
| <b>第三章 可编程控制器的实验装置介绍 .....</b>  | <b>- 18 -</b> |
| <b>第四章 PLC 应用实验与设计 .....</b>    | <b>- 19 -</b> |
| 实验一    基本指令的编程练习 .....          | - 19 -        |
| 实验二    四节传送带的模拟 .....           | - 21 -        |
| 实验三    LED 数码显示控制 .....         | - 22 -        |
| 实验四    十字路口交通灯控制的模拟 .....       | - 23 -        |
| 实验五    装配流水线控制的模拟 .....         | - 24 -        |
| 实验六    水塔水位控制的模拟 .....          | - 25 -        |
| 实验七    天塔之光 .....               | - 26 -        |
| 实验八    液体混合装置控制的模拟 .....        | - 27 -        |
| 实验九    机械手的模拟控制 .....           | - 28 -        |
| 实验十    组态监控实验 .....             | - 29 -        |

---

# 第一章 可编程控制器简介

PLC 是一种新型的电气控制装置，它将传统的继电控制技术和计算机控制技术融为一体，被广泛应用于各种生产机械和生产过程的自动控制。

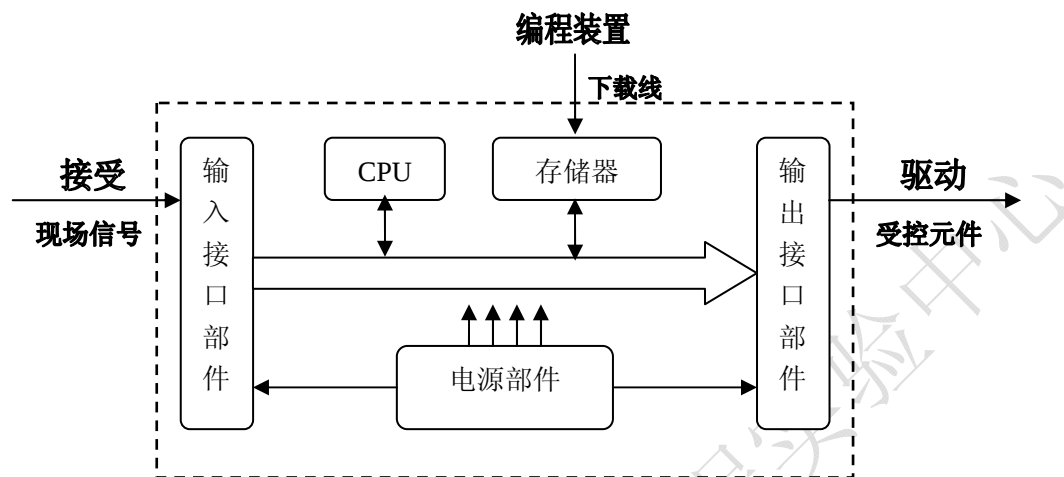
传统的继电接触控制具有结构简单、易于掌握、价格便宜等优点，在工业生产中广泛应用。但这类控制装置体积大，耗电较多，功能少，特别是靠硬件连接构成系统，接线复杂，通用性和灵活性差。

PLC 的起源可以追溯到 60 年代，美国通用汽车公司为了适应汽车型号不断翻新的需要，对生产线上的控制设备提出了新的要求，为此设想：把计算机的功能完善、通用灵活等优点和继电器接触控制简单易懂等优点结合起来，从而提供了继电器控制系统无法比拟的灵活性。并要求把计算机的编程方法和程序输入方法加以简化，使得不熟悉计算机的人也能方便使用。但这一时期它主要是代替继电器系统完成顺序控制，逻辑运算，计时、计数、故称为可编程逻辑控制器，简称 PLC (Programmable Logical Controller)。

进入 80 年代，随着微电子技术和计算机技术的发展，可编程控制器的功能已远远超出逻辑控制、顺序控制的范围，它具有数据运算、传送与处理功能和可以进行模拟量控制、位置控制，特别是远程通讯功能的实现，易于实现柔性加工和制造系统，因此将其称为可编程控制器 (Programmable Controller) 简称 PC，但为了与个人电脑 PC 相区别，仍将其称为 PLC。

目前 PLC 已广泛应用于冶金、矿业、机械、轻工等领域，加速了机电一体化的进程。

## 一、PLC 的结构及各部分作用



PLC 硬件系统结构图

### 1. 主机

主机部分包括中央处理器（CPU）、系统程序存储器和用户程序及数据存储器。

CPU 是 PLC 的核心，一切逻辑运算及判断都是由其完成的，并控制所有其它部件的操作。它就是我们常说的电脑芯片。

内部存储器有两类：一类是系统程序存储器，另一类是用户程序及数据存储器

系统程序存储器：主要存放系统管理和监控程序及对用户程序作编译处理的程序。系统程序已由厂家固定，用户不能更改。

用户程序及数据存储器：主要存放用户编制的应用程序及各种暂存数据、中间结果。

### 2. 输入/输出（I/O）接口

输入接口用于接收输入设备（如：按钮、行程开关、传感器等）的控制信号。

输出接口用于将经主机处理过的结果通过输出电路去驱动输出设备（如：接触器、电磁阀、指示灯等）。

### 3. 电源

电源指为 CPU、存储器、I/O 接口等内部电子电路工作所配备的直流开关稳压电源。

#### 4. 编程器

编程器是 PLC 很重要的外部设备，它主要由键盘、显示器组成。编程器分简易型和智能型两类。小型 PLC 常用简易编程器，大、中型 PLC 多用智能编程器。编程器的作用是编制用户程序并送入 PLC 程序存储器。利用编程器可检查、修改、调试用户程序和在线监视 PLC 工作状况。现在许多 PLC 采用和计算机联接，并利用专用的工具软件进行编程和监控。

## 二、PLC 的工作原理

PLC 采用“顺序扫描、不断循环”的工作方式，这个过程可分为输入采样、程序执行、输出刷新三个阶段，整个过程扫描并执行一次所需的时间称为扫描周期。

### 1、输入采样阶段

PLC 在输入采样阶段，以扫描方式顺序读入所有输入端的通/断状态或输入数据，并将此状态存入输入状态寄存器，即输入刷新。接着转入程序执行阶段。在程序执行期间，即使输入状态发生变化，输入状态寄存器的内容也不会改变，只有在下一个扫描周期的输入处理阶段才能被读入。

### 2、程序执行阶段

PLC 在执行阶段，按先左后右，先上后下的步序，执行程序指令。其过程如下：从输入状态寄存器和其它元件状态寄存器中读出有关元件的通/断状态，并根据用户程序进行逻辑运算，运算结果再存入有关的状态寄存器中。

### 3. 输出刷新阶段

在所有指令执行完毕后，将各物理继电器对应的输出状态寄存器的通/断状态，在输出刷新阶段转存到输出寄存器，去控制各物理继电器的通/断，这才是 PLC 的实际输出。

由 PLC 的工作过程可见，在 PLC 的程序执行阶段，即使输入发生了变化，输入状态寄存器的内容也不会立即改变，要等到下一个周期输入处理阶段才能改变。暂存在输出状态寄存器中的输出信号，等到一个循环周期结束，CPU 集中将这些输出信号全部输出给输出锁存器，这才成为实际的 CPU 输出。因此全部输入、输出状态的改变就需要一个扫描周期，换言之，输入、输出的状态保持一个扫描周期。

### 三、TWDLCAE40DRF 的主要功能

扫描周期： 常规（循环）或周期（常数）（2 到 150 ms）

执行时间： 0.14  $\mu$ s 到 0.9  $\mu$ s 对一条列表指令

数据存储器容量： 256 个内存位

程序存储器容量： 3000 列表指令

RAM 备份： 在锂电池充满电后，通过内部锂电池备份大约可持续 30 天（典型）在 25° C。电池从 0% 到 90% 的充电时间为 15 小时。在充电 9 小时使用 15 小时的情况下，电池寿命大约为 10 年。电池不可更换。

编程端口： EIA RS-485

I/O 点数及其他扩展模块：

一体型 40 I/O 控制器：

- 具有 24 个数字量输入，14 个继电器和 2 个晶体管输出
- 具有 2 个模拟电位器输入
- 具有 1 个集成的串行口
- 具有一个支持附加串行口的插槽
- 内置 RTC
- 具有一个用户可更换电池的电池盒
- 最多连接 7 个扩展 I/O 模块
- 最多连接 2 个 AS-Interface V2 总线接口模块
- 可连接 1 个 CANopen 现场总线接口模块
- 连接一个可选内存插卡(32 KB 或 64 KB)
- 可连接一个可选的操作器显示模块

图例：

TWDLCAE40DRF



---

## 第二章 可编程控制器的软件介绍及使用

### 一、软件概述

TwidoSoft 是一个为 Twido 可编程控制器建立、配置和维护应用程序的图形化开发环境。TwidoSoft 是一款 32 位的基于 Windows 的程序，可在操作系统是 Microsoft Windows 98 第二版和 Microsoft Windows 2000 Professional 的个人计算机运行。TwidoSoft 1.13 或更高版本也可以在 Microsoft Windows XP 上运行。

本实验所使用的软件版本是 TwidoSoft V3.5。

#### 1. TwidoSoft 的主要功能

- 应用程序浏览器

使用应用程序浏览器查看、配置、编制和维护应用程序。同样的命令也可以方便地从菜单中获得。可用图形化的方式来配置控制器，扩展 I/O 和选件。

- 存储器分配

通过状态栏中的存储器使用状态（在存储器编辑器中也可见）来查看程序已使用存储器的百分比。当可用存储器低到一定程度时，会发出警告。

- 在线和离线操作

离线操作中，TwidoSoft 未与控制器相连，所以只能改变 PC 存储器中的应用程序。使用离线操作来创建和配置应用程序。在线操作中，TwidoSoft 直接与控制器相连，所以能改变控制器存储器中的应用程序。使用在线操作来运行控制器和调试修改应用程序。

- 应用程序保护

应用程序可以被保护，以防止未授权的查看，修改和复制。

#### 2. PC 与 Twido 的通信

连接 PC 到 Twido 控制器需要通讯电缆。

可以用以下任意一种方式来连接：

- 使用特殊的多功能电缆将 PC 的 COM 或 USB 串行口与控制器的端口 1 相连：

- ✧ TSX PCX 1031 (2.5m) 电缆实现在 RS-485 和 RS-232 间的信号转换；
- ✧ TSX PCX 3030 (2.5m) 电缆实现在 RS-485 和 USB 间的信号转换。

- 使用电话线连接两个调制解调器，一个调制解调器接到控制器的端口 1，另一个接到 PC 的 COM 串口(或是内置调制解调器)。

所有 Twido 控制器的端口 1 都是内置的 RS-485 端口,使用此端口与 TwidoSoft 编程软件进行通讯。当把通讯电缆连至端口 1 时,自动进入与 TwidoSoft 通讯需要的通讯协议的工作模式。

## 二、编程语言描述

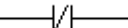
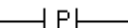
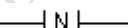
同其它电脑装置一样,PLC 的操作是依其程序操作进行的,而程序是用程序语言表达的,并且表达的方式多种多样。TwidoSoft 可使用梯形图、列表和 Grafcet 编程语言对 Twido 可编程控制器进行编程。

由梯形图语言编写的应用可以被转换为列表语言。如果指令遵循可逆性原则,由列表语言编写的程序可以被逆转为梯形图语言。本实验书详细介绍如何使用梯形图语言进行编程。

### 1. 梯形图图形单元

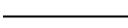
#### 触点

触点图形单元用于测试区编程且占据一个单元(一行一列)。

| 名字    | 图形单元                                                                                | 指令  | 功能                   |
|-------|-------------------------------------------------------------------------------------|-----|----------------------|
| 常开触点  |  | LD  | 当控制位对象为状态 1 时通过触点。   |
| 常闭触点  |  | LDN | 当控制位对象为状态 0 时通过触点。   |
| 上升沿触点 |  | LDR | 上升沿：检测控制位对象从 0 变为 1。 |
| 下降沿触点 |  | LDF | 下降沿：检测控制位对象从 1 变为 0。 |

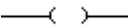
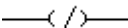
#### 连接元件

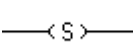
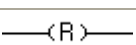
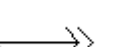
图形连接单元用于连接测试和动作图形单元。

| 名字   | 图形单元                                                                                | 功能                     |
|------|-------------------------------------------------------------------------------------|------------------------|
| 水平连接 |  | 两个电源栏之间测试和动作图形单元的连续连接。 |
| 垂直连接 |  | 平行测试和动作图形单元连接。         |

#### 线圈

图形单元用于动作区编程且占据一个单元(一行一列)。

| 名字   | 图形单元                                                                                | 指令  | 功能                |
|------|-------------------------------------------------------------------------------------|-----|-------------------|
| 直接线圈 |  | ST  | 相关位对象得到测试区结果值。    |
| 取反线圈 |  | STN | 相关位对象得到测试区结果的相反值。 |

|               |                                                                                   |           |                        |
|---------------|-----------------------------------------------------------------------------------|-----------|------------------------|
| 置位线圈          |  | S         | 当测试区结果为 1 时相关位对象被置为 1。 |
| 复位线圈          |  | R         | 当测试区结果为 1 时相关位对象被置为 0。 |
| 跳转或子程序调用      |  | JMP<br>SR | 连接到一个标注指令，向上跳转或向下跳转。   |
| 子程序返回（在扩展指令中） |  | RET       | 位于子程序末端返回到主程序。         |
| 结束程序（扩展指令）    |  | END       | 程序结束定义。                |

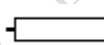
## 功能块

模块的图形单元在测试区被调用，需要四行两列单元（除了超高速计数器需要五行两列）。

| 名字              | 图形单元                                                                              | 功能                                                 |
|-----------------|-----------------------------------------------------------------------------------|----------------------------------------------------|
| 定时器，计数器，寄存器，等等。 |  | 每个功能模块使用输入和输出和其它图形单元连接。<br>注意：功能模块的输出不能互相连接（垂直短接）。 |

## 操作和比较块

比较模块在测试区被调用，操作模块在动作区被调用。

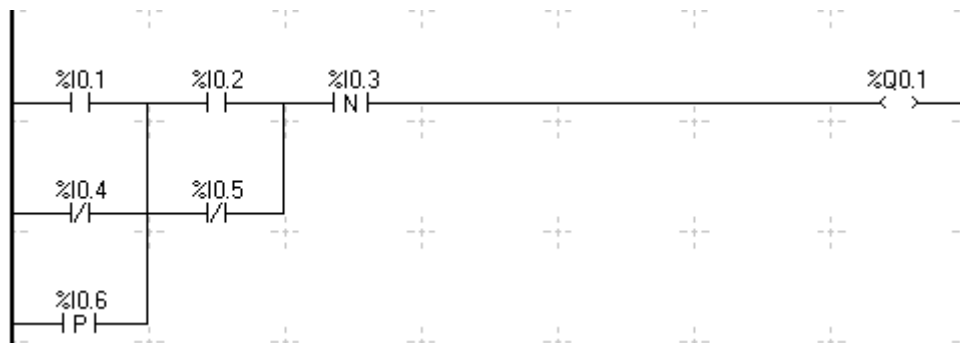
| 名字   | 图形单元                                                                                | 功能                               |
|------|-------------------------------------------------------------------------------------|----------------------------------|
| 比较模块 |  | 比较两个操作数，当结果为真时输出变为 1。<br>大小：一行两列 |
| 操作模块 |  | 完成算术和逻辑运算。<br>大小：一行四列            |

## 2. 梯形图编程原则

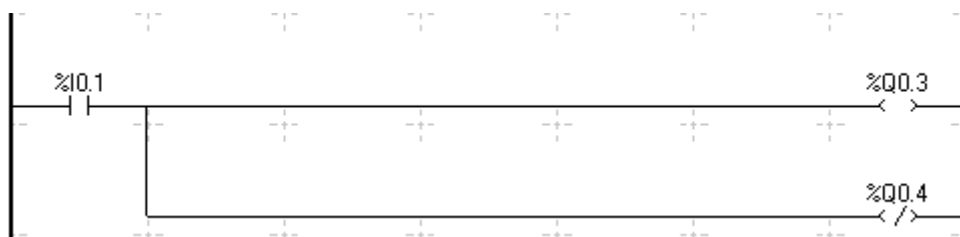
- (一) PLC 编程元件的触点在编程过程中可以无限次使用，每个继电器的线圈在梯形图中可以有多重涉及，但只有最后的结果才能更新硬件的输出。
- (二) 梯形图的每一逻辑行皆起始于左母线，终止于右母线。线圈总是处于最右边，且不能直接与左边母线相连。



- (三) 编制梯形图时，应尽量做到“上重下轻、左重右轻”。



(四) 两个或两个以上的线圈可以并联，但不可以串联。



(五) 同一梯级头中只能有一个操作模块，但可以并联多个线圈。

### 3. 编程方法

TwidoSoft 在开发应用程序时，提供了编辑器和浏览器窗口以执行相应的任务。一个 TwidoSoft 应用程序包含程序，配置数据，变量符号和文档管理。这些组成部分可以在构建应用程序时以任意顺序使用。例如，一个应用程序可以依照如下顺序开发：

- 首先，定义配置数据，
- 其次，定义变量符号，
- 然后，编写程序指令，
- 最后，添加注释和打印文档。

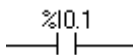
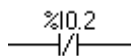
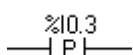
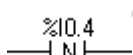
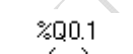
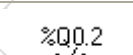
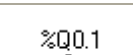
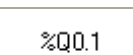
这只是组织这些任务的其中一种方法；如果以逆序执行也同样容易。使用不同的编辑器开发应用程序的每一个部分使得应用程序开发过程更加系统，使整个应用程序更加清晰。

### 三、指令系统

#### 1. 基本指令

##### 布尔指令

布尔指令可与梯形图语言元素相比较。这些指令归纳如下表所示：

| 条目   | 指令                       | 梯形图示例                                                                               | 描述                         |
|------|--------------------------|-------------------------------------------------------------------------------------|----------------------------|
| 测试元素 | LD 指令（装载）等价于常开触点。        |    | 当位%I0.1 为 1 时触点闭合。         |
|      | LND 指令等价于常闭触点。           |    | 当位%I0.2 为 0 时触点闭合。         |
|      | LDR 指令（装载上升沿）等价于上升沿检测触点。 |    | 当位%I0.3 从 0 到 1 的变化时触点闭合。  |
|      | LDF 指令（装载下降沿）等价于下降沿检测触点。 |    | 当位%I0.4 从 1 到 0 的变化时触点闭合。  |
| 动作元素 | 存储(ST)指令等价于线圈。           |   | 相关位对象取位累加器的逻辑值（前面逻辑运算的结果）。 |
|      | SNT 指令等价于取反线圈。           |  | 相关位对象为前面逻辑运算的取反。           |
|      | 置位指令（S）等价于置位线圈。          |  | 相关位对象置位为 1。                |
|      | 复位指令（R）等价于复位线圈。          |  | 相关位对象复位为 0。                |

用于布尔指令的允许操作数类型：

| 操作数    | 描述                               |
|--------|----------------------------------|
| 0/1    | 立即值 0 或 1                        |
| %I     | 控制器输入%Ii.j                       |
| %Q     | 控制器输出%Qi.j                       |
| %M     | 内部位%Mi                           |
| %S     | 系统位%Si                           |
| %X     | 步位%Xi                            |
| %BLK.x | 功能模块位（例如：时间模块%TMi.Q、计数模块%Ci.D 等） |
| %*:Xk  | 字位（例如，%MWi:Xk）                   |
| [ ]    | 比较表达式（例如，[%MWi<1000]）            |

## 程序指令

### END 指令

定义一个程序扫描执行的结束。可以设置在触点闭合时，则程序结束。  
无条件 END 指令示例：

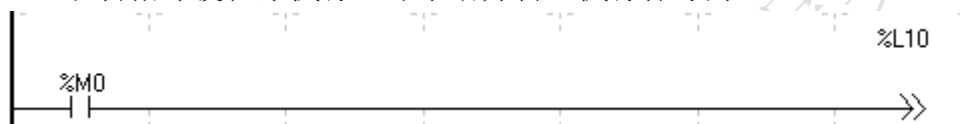


有条件 END 指令示例：

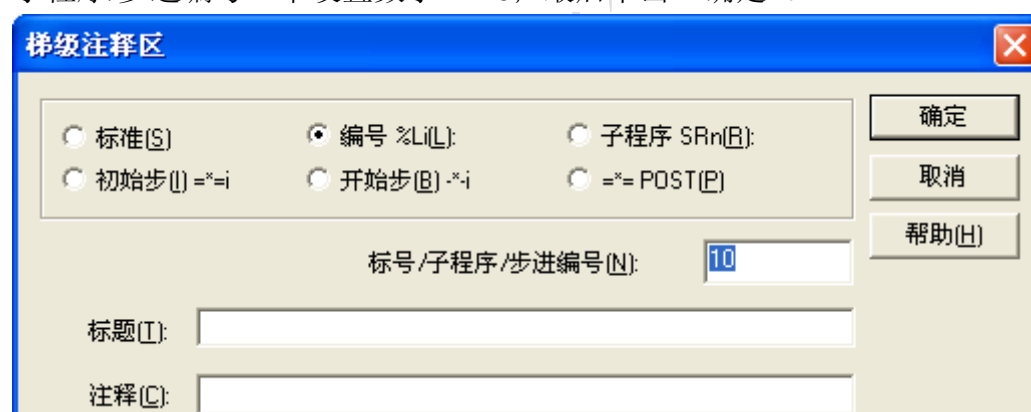


### 跳转指令

跳转指令使程序执行立即中断并转入执行标号为%Li (i = 1~16) 的程序行。



双击想要跳转到的梯级头；在梯级注释区内，选取“编号%Li(L):”，在“标号/子程序/步进编号”中设置数字 1~16；最后单击“确定”。



注意：

- 跳转指令不允许用于圆括号内，且不能位于指令 AND, OR 和右括号指令")"之间。
- 标号只能位于指令 LD, LDN, LDR, LDF 或 BLK 之前。
- 标号%Li 的编号在程序中只能被定义一次。
- 程序可以向下或向上跳转，当向上跳转时，必须注意程序扫描时间。
- 延长扫描时间可能导致看门狗的触发。

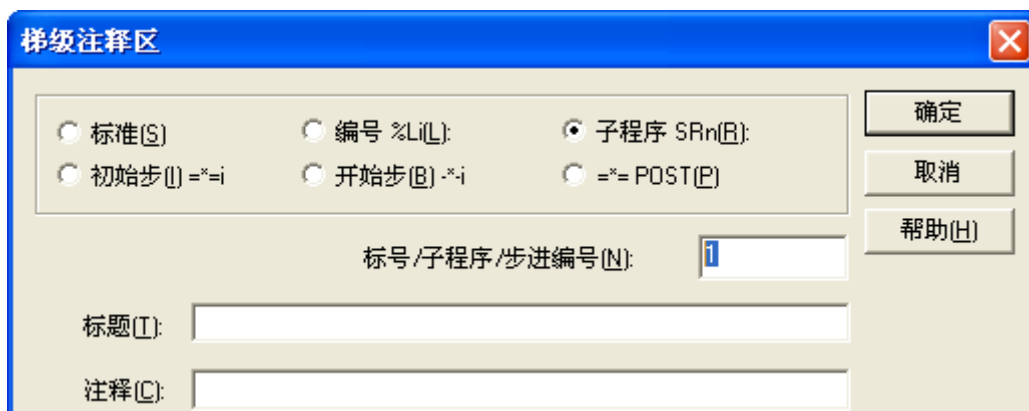
### 子程序指令

子程序由三部分组成：

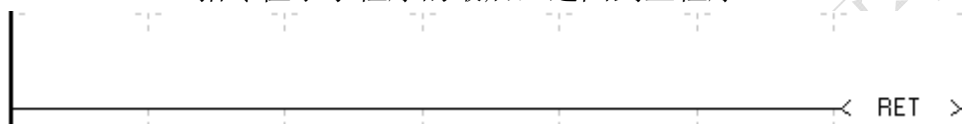
- (1) SRn 指令调用标号为 SRn 的子程序。



(2) 子程序用标号 SRn:表示, n=0 到 63 对于其它控制器。



(3) RET 指令位于子程序的最后, 返回到主程序。



注意:

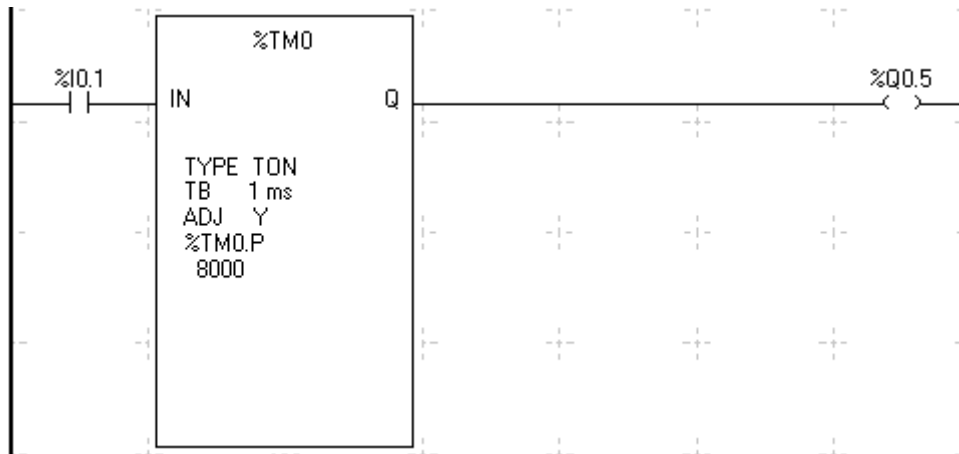
- 一个子程序不能调用另一个子程序。
- 子程序指令不允许用于圆括号内, 且不能位于指令 AND(, OR(和右括号指令")"之间。
- 标号只能位于指令 LD 或 BLK 之前, 用于标识一个布尔等式 (或梯级) 的开始。
- 赋值指令不能跟随在子程序调用之后。这是因为子程序可能改变布尔运算累加器的内容。这样返回时, 它的值可能与调用前不同。

### 3.基本功能模块

#### 定时器功能模块

在 PLC 内的定时器是根据时钟脉冲的累积形式, 当所计时间达到设定值时, 其输出触点动作, 时钟脉冲有 1ms, 10ms, 100ms, 1s, 1min。可以用用户程序存储器那的常数作为预置值。

定时器功能模块使用图例:

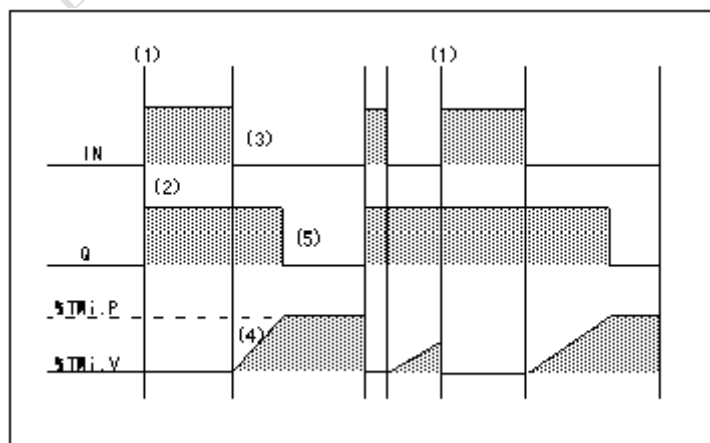


定时器具有如下参数：

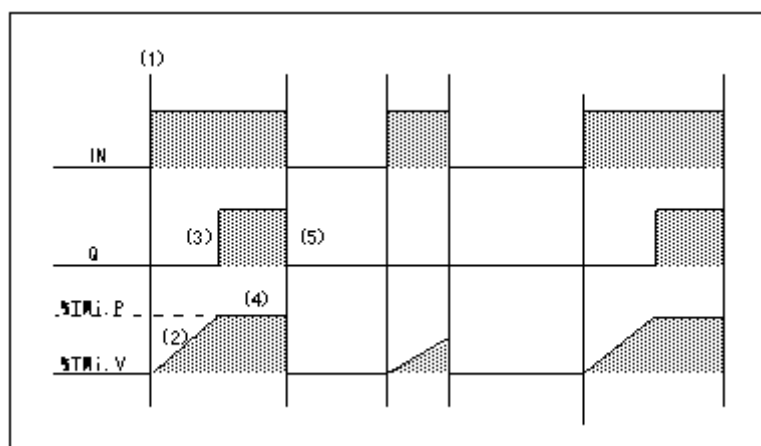
| 参数         | 标识     | 值                                                             |
|------------|--------|---------------------------------------------------------------|
| 定时器编号      | %Tmi   | 0 到 63:TWDLCAE10DRF 和 TWDLCAE16DRF<br>0 到 127 对所有其它控制器。       |
| 类型 (TYPE)  | TON    | • 定时器 +导通—延时 (默认)                                             |
|            | TOF    | • 定时器 关断—延时                                                   |
|            | TP     | • 脉冲 (单稳态)                                                    |
| 时基         | TB     | 1 分钟 (默认), 1 秒, 100 毫秒, 10 毫秒, 1 毫秒                           |
| 当前值        | %Tmi.V | 当定时器工作时, 该字从 0 增加到 %Tmi.P。可被程序读和测试, 但不可写。%Tmi.V 可以通过活动表编辑器修改。 |
| 预置值        | %Tmi.P | 0~9999, 该字可读, 测试和被修改, 默认值是 9999。周期或产生的延时为 %Tmi.P x TB。        |
| 动态监控表编辑器   | ADJ    | Y:Yes, 预置 %Tmi.P 值可以通过活动表编辑器修改。<br>N:No, 预置 %Tmi.P 值不能被修改。    |
| 输入使能 (或指令) | IN     | 上升沿 (TON 或 TP 类型) 或下降沿 (TOF 类型) 启动定时器。                        |
| 定时器输出      | Q      | 根据执行功能的类型, 相关位 %Tmi.Q 置为 1: TON, TOF, 或 TP                    |

**注意：**预置值越大，定时器的精度越高。

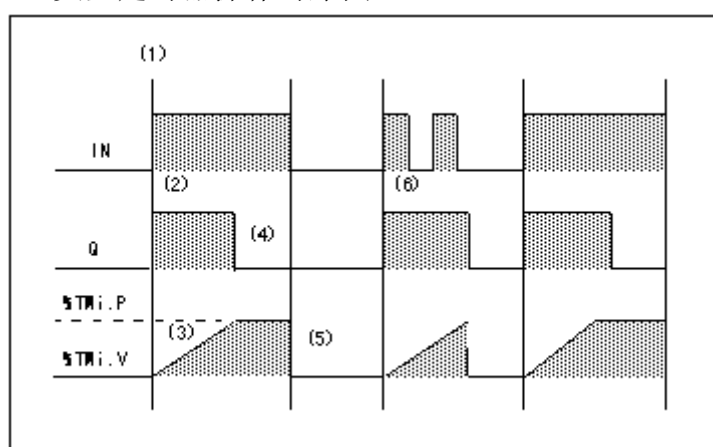
TOF 类型定时器操作时序图：



TON 类型定时器操作时序图：



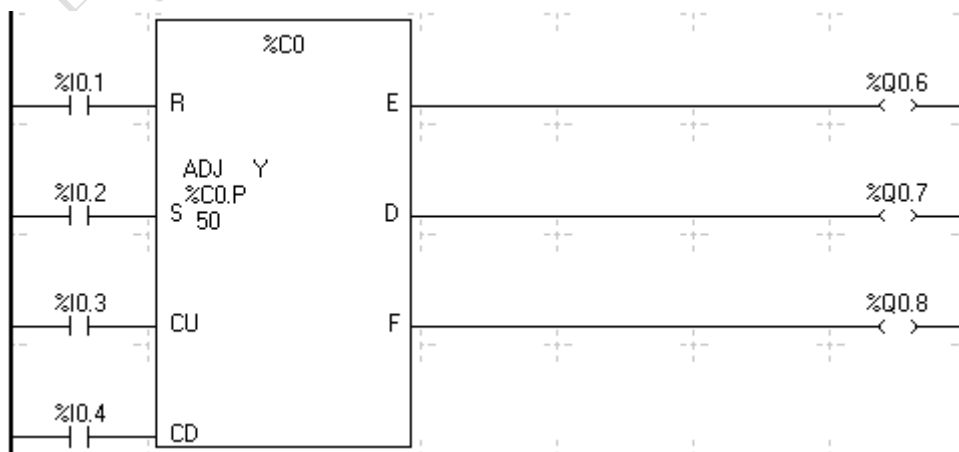
TP 类型定时器操作时序图：



加/减计数器功能模块

从动作按钮区选择一个计数器功能模块 ，该动作按钮位于梯形图模板工具栏，通过指向并点击鼠标左键，或按 **SHIFT** 和指定功能键，左击或按空格键插入模块。要选择一个单元，用户必须调用选择工具( 箭头图标) 然后用鼠标左击或利用箭头键移动到所选单元。最后，完成定时器编程。

计数器功能模块使用图例：



寄存器功能模块具有如下参数：

| 参数         | 标识        | 值                                                                            |
|------------|-----------|------------------------------------------------------------------------------|
| 计数器编号      | %Ci       | 0 到 127                                                                      |
| 当前值        | %Ci.V     | 字根据输入（或指令）CU 和 CD 被增加或减少。可被程序读和测试，但不可写。使用数据编辑器修改%Ci.V。                       |
| 预置值        | %Ci.P     | $0 \leq \%Ci.P \leq 9999$ 。能被读、测试和写(默认值: 9999)。                              |
| 用活动表编辑器编辑  | ADJ       | Y:Yes, 预置值可以通过活动表编辑器修改。<br>N:No, 预置值不能使用活动表编辑器修改。                            |
| 输入（或指令）复位  | R         | 状态为 1: %Ci.V = 0.                                                            |
| 输入（或指令）复位  | S         | 状态为 1: %Ci.V = %Ci.P.                                                        |
| 加运算输入（或指令） | CU        | 在上升沿增加%Ci.V。                                                                 |
| 减运算输入（或指令） | CD        | 在上升沿减少%Ci.V。                                                                 |
| 减运算溢出输出    | E (Empty) | 当减计数器%Ci.V 从 0 变到 9999 时, 相关位%Ci.E=1; 当%Ci.V 到达 9999 时置为 1, 如果计数器继续减少则复位为 0。 |
| 预置输出达到     | D (完成)    | 当%Ci.V=%Ci.P 时, 相关位%Ci.D=1。                                                  |
| 加运算溢出输出    | F (Full)  | 当%Ci.V 从 9999 变到 0 时, 相关位%Ci.F=1 (当%Ci.V 到达 0 时置为 1, 如果计数器继续增加则复位为 0)。       |

加/减计数器操作的主要过程：

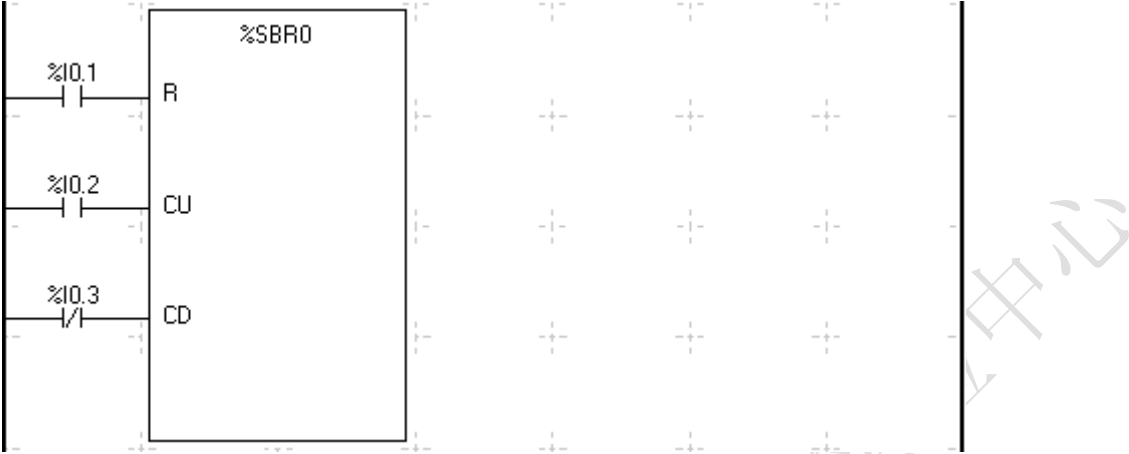
| 操作    | 动作                                                                                   | 结果                                                  |
|-------|--------------------------------------------------------------------------------------|-----------------------------------------------------|
| 计数    | 加计数输入 CU 出现上升沿（或指令 CU 被激活）。                                                          | 当前值%Ci.V 加 1。                                       |
|       | 当前值%Ci.V 等于预置值%Ci.P。                                                                 | “预置达到”输出位%Ci.D 变为 1。                                |
|       | 当前值%Ci.V 从 9999 变为 0。                                                                | 输出位%Ci.F（加计数溢出）变为 1。                                |
|       | 如果计数器继续增加。                                                                           | 输出位%Ci.F（加计数溢出）复位到 0。                               |
| 减计数   | 减计数输入 CD 出现上升沿（或指令 CD 被激活）。                                                          | 当前值%Ci.V 减 1。                                       |
|       | 当前值%Ci.V 从 0 变为 9999。                                                                | 输出位%Ci.E（减计数溢出）变为 1。                                |
|       | 如果计数器继续减少。                                                                           | 输出位%Ci.E（减计数溢出）复位到 0。                               |
| 加/减计数 | 要同时使用加计数和减计数功能（或同时激活指令 CD 和 CU），必须对同时控制两个对应的输入 CD 和 CU。这两个输入被连续扫描。如果它们都为 1，则当前值保持不变。 |                                                     |
| 复位    | 输入 R 置为状态 1（或者指令 R 被激活）。                                                             | 当前值%Ci.V 被强制为 0。输出%Ci.E, %Ci.D 和 %Ci.F 置为 0。复位输入优先。 |
| 置位    | 如果输入 S 置为 1（或指令 S 被激活）且复位输入为 0（或指令 R 未被激活）。                                          | 当前值%Ci.V 取%Ci.P 值且输出%Ci.D 置为 1。                     |

注：控制器停止后热启动对计数器的当前值（%Ci.V）没有影响。

移位寄存器功能模块

通过输入或指令可将寄存器中的位左移或右移，且热启动后，寄存器内的字仍保留。

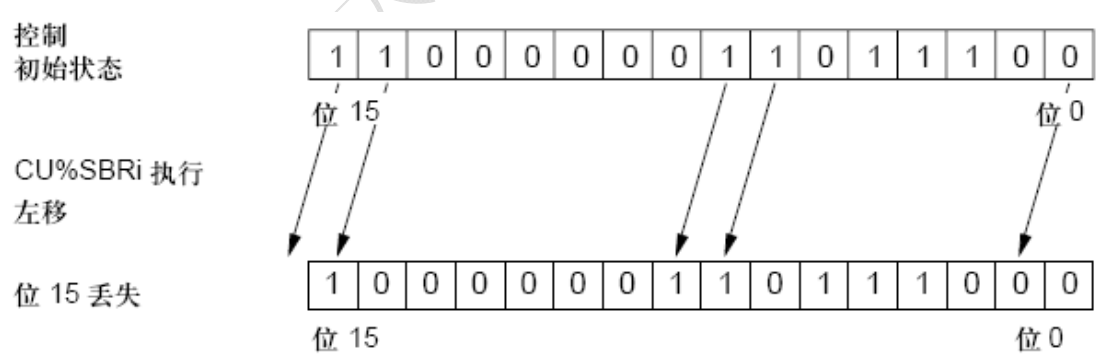
移位寄存器功能模块使用图例：



移位寄存器功能模块具有下列参数：

| 参数        | 标识       | 值                                           |
|-----------|----------|---------------------------------------------|
| 寄存器编号     | %SBRi    | 0 到 7                                       |
| 寄存器位      | %SBRi. j | 移位寄存器的位 0 到 15（j = 0 到 15）可被测试指令测试，且由赋值指令写。 |
| 输入（或指令）复位 | R        | 当功能块参数 R 为 1 时，将设置寄存器位 0 到 15 %SBRi. j 为 0. |
| 左移输入（或指令） | CU       | 其上升沿将寄存器位左移一位。                              |
| 右移输入（或指令） | CD       | 其上升沿将寄存器位右移一位。                              |

移位操作前后的位形式如图：

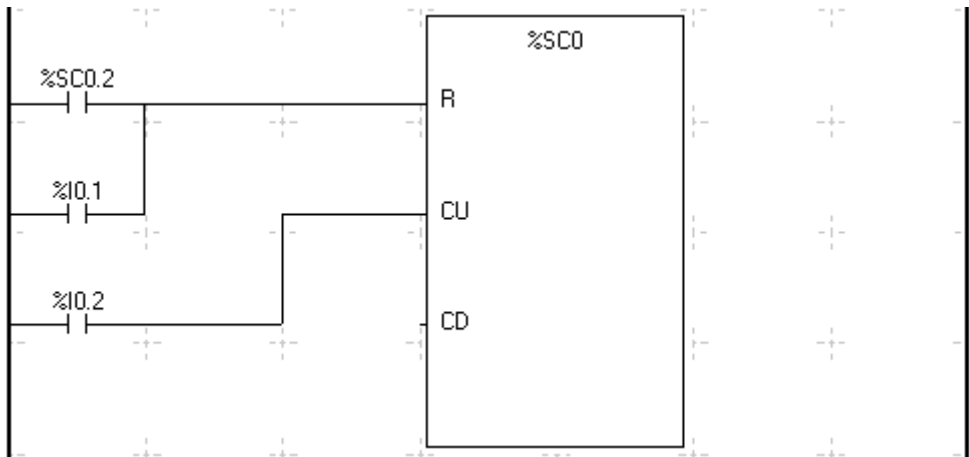


使用指令 CD 右移一位也是如此，位 0 被丢失。

步进计数器功能模块

步进计数器功能模块(%SCi)提供了一系列的步，这些步可赋值给动作。从一个步移动到另一个步取决于外部或内部事件。每当一个步处于激活状态时，相关位被置为 1。步进计数器在一个时刻只能有一个步被激活。

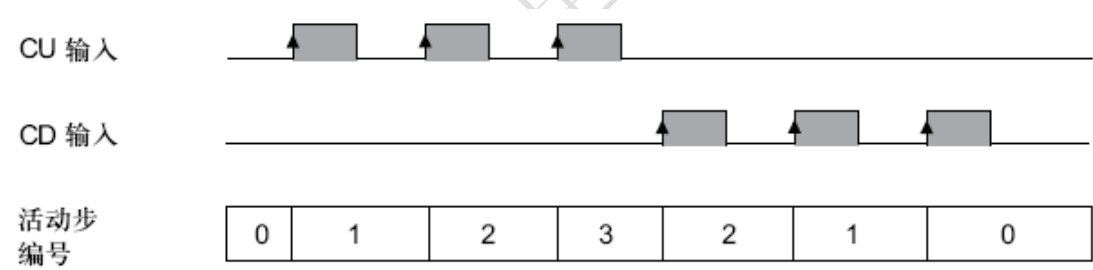
步进计数器功能模块使用图例：



步进计数器功能模块具有下列参数：

| 参数        | 标识     | 值                                               |
|-----------|--------|-------------------------------------------------|
| 步进计数器编号   | %SCi   | 0 - 7                                           |
| 步进计数器位    | %SCi.j | 步进计数器的位 0 到 225 (j = 0 到 225) 可被装载逻辑测试，且由赋值指令写。 |
| 输入（或指令）复位 | R      | 当功能块参数 R 为 1 时，将复位步进计数器。                        |
| 输入（或指令）增加 | CU     | 其上升沿将步进计数器增加一步。                                 |
| 输入（或指令）减少 | CD     | 其上升沿将步进计数器减少一步。                                 |

下面是步进计数器功能模块操作时序图。



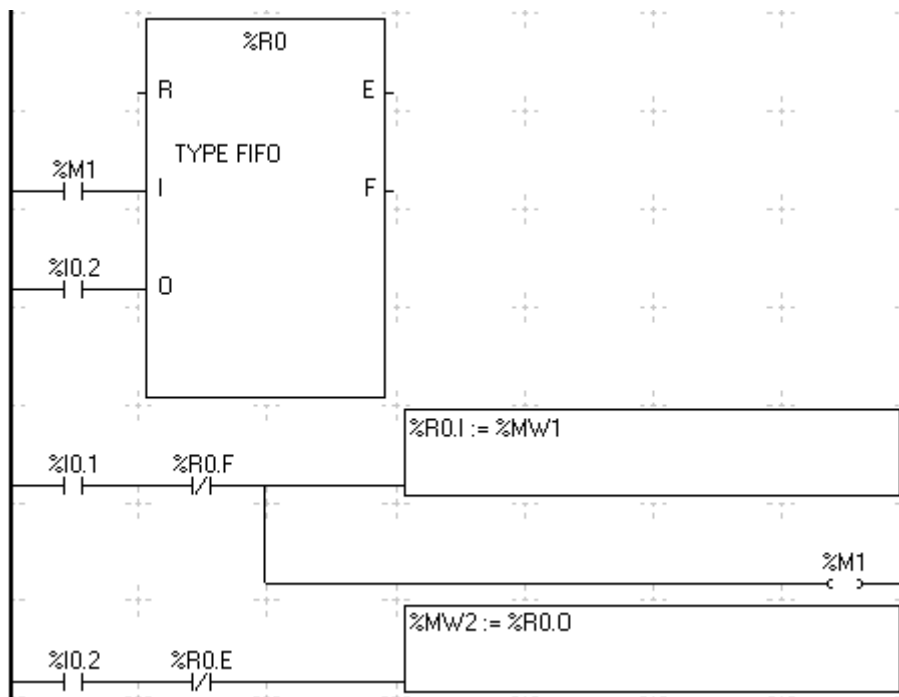
#### 4.高级功能模块

Twido 可编程控制器的高级模块包括 LIFO/FIFO 寄存器功能模块（%Ri）、脉宽调制功能模块(%PWM)、脉冲发生器输出功能模块（%PLS）、鼓控制器功能模块(%DR)、高速计数器功能模块(%FC)、超高速计数器功能模块(%VFC)、发送/接收消息-交换指令（EXCH）、交换控制模块（%MSGx）。本实验书将详细介绍寄存器功能模块和 PID 功能模块的使用，其他请参阅软件手册。

##### LIFO/FIFO 寄存器功能模块

一个寄存器是一个内存块，可以存储 16 个 16 位的字，它有以下两种方式：队列方式（先进先出）即 FIFO；堆栈方式（后进先出）即 LIFO。

寄存器功能模块使用图例：



上图的编程示例显示了寄存器%R0 未满足(%R0.F = 0)时，使用存储请求(%IO.1) 将一个存储字(%MW1)的内容装入寄存器(%R0.I)。寄存器的存储请求由%M1 实现。取出请求由输入%IO.2 来实现，且如果寄存器不为空（%R0.E = 0），则%R0.O 的内容被装入%MW2。

寄存器功能模块具有如下参数：

| 参数        | 标识         | 值                     |
|-----------|------------|-----------------------|
| 寄存器编号     | %Ri        | 0 到 3.                |
| 类型        | FIFO/ LIFO | 队列方式或堆栈方式。            |
| 输入字       | %Ri.I      | 寄存器输入字。可读取，测试，和写入。    |
| 输出字       | %Ri.O      | 寄存器输出字。可读取，测试，和写入。    |
| 存储输入（或指令） | I (In)     | 上升沿处将字%Ri.I 的内容存入寄存器。 |
| 取回输入（或指令） | O (Out)    | 上升沿处将一个数据字装入字%Ri.O 内。 |
| 输入（或指令）复位 | R (Reset)  | 状态为 1 时，初始化寄存器。       |
| 空输出       | E (Empty)  | 对应位%Ri.E 表示寄存器空。可调试。  |
| 满输出       | F (Full)   | 对应位表%Ri.F 示寄存器满。可调试。  |

注：控制器停止后热启动 (%S1=1)对寄存器的当前值和输出位的状态没有影响。

---

## 第三章 可编程控制器的实验装置介绍

本实验所选用的可编程控制器为施耐德公司提供的 TWIDO 系列的 TWDLCAE40DRF 一体型控制器。

实验箱中共分主控制模块和实验部分模块，可以做基本指令的编程练习、四节传送带的模拟、LED 数码显示控制、十字路口交通灯控制的模拟、装配流水线控制的模拟、水塔水位控制的模拟、天塔之光、液体混合装置控制的模拟、机械手的模拟控制的模拟等共九个实验，本课程原则上要求同学选做其中的六个实验。第十个为组态监控实验，在完成前面实验的基础上进行拓展。

## 第四章 PLC 应用实验与设计

### 实验一 基本指令的编程练习

#### (一) 与或非逻辑功能实验

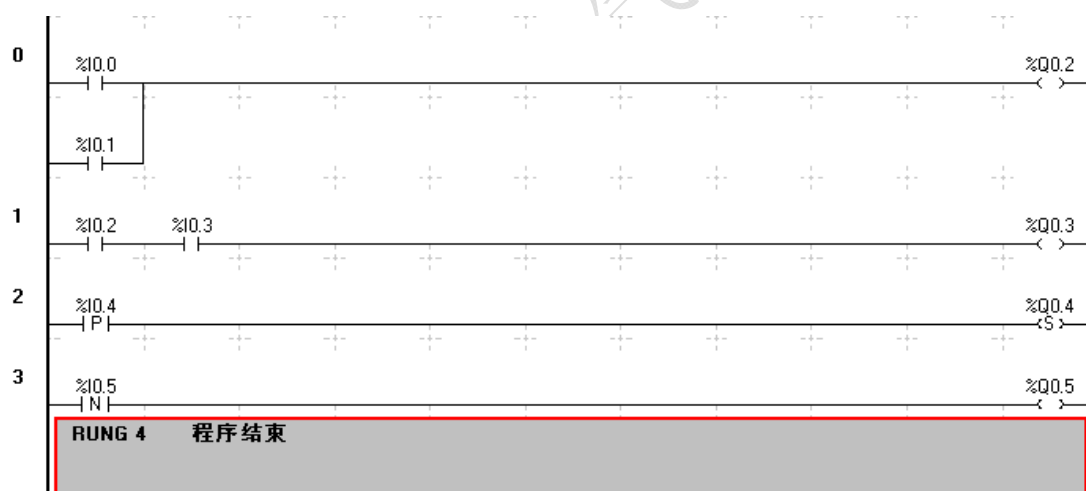
##### 一、实验目的

1. 熟悉 PLC 实验装置，TWIDO 系列编程控制器的外部接线方法
2. 了解编程软件 TWIDOSOFT 的编程环境，软件的使用方法。
3. 掌握与、或、非逻辑功能的编程方法。

##### 二、梯形图参考程序

通过程序判断 Q2~Q5 的输出状态，然后再输入并运行程序加以验证。

参考程序如下：



##### 三、实验步骤

1. 将输入部分的 COM0 连到电源+24V，输出部分的 COM2 连到 0V，再将“基本指令的编程练习”部分的+24V 和 COM 与对应的电源的+24V 和 0V 相连。
2. 将输入部分的 I1~I4 分别与拨码开关 SA1~SA4 相连，I5、I6 分别与按钮开关 SB5、SB6 相连。输出部分 Q2~Q5 与 L1~L4 相连。
3. 通过 USB 下载线连接计算机与 PLC 主机。打开编程软件 TWIDOSOFT，新建一个 twd 文件。在弹出的“功能级别管理”对话框中选择“自动”，“最高可能性”。

然后确定。

4. 点击“硬件”—>“更改控制器类型”，在对话框中选择控制器型号“TWDLCAE40DRF”。再选择“控制器”—>“选择连接”，在菜单栏中选择“控制器”——“USB”。（如果下载线为 TSXCUSB485，此处选择 COM7、COM8 或 COM9 等）。

注意：每次新建一个 twd 文件均需重新选择控制器类型和连接方式。

5. 输入梯形图，检查无误后，按“控制器”—>“连接”，将缩编程序下载到主机内，点运行按钮并确认后，程序即开始运行。有关的指示灯将显示运行结果。

6. 分别拨动输入开关 I1~I8，然后“控制器”—>“切换动态显示”观察实验结果，并观察实验装置输出指示灯是否符合逻辑。

## （二）定时器/计数器功能实验

### 一、实验目的

掌握定时器、计数器的正确编程方法，并学会定时器和计数器扩展方法，用编程软件对可编程控制器的运行进行监控。

### 二、实验要求

#### 1. 定时器的认识实验

建立一个 TON 定时器，实现：当按住 SB1 三秒钟后，灯 L1 亮。注意：PLC 的输出端口使用 Q2~Q15，相应的输出 COM 端须接 0V。

#### 2. 定时器扩展实验

通过使用两个定时器，实现：按一下 SB2，灯 L2 亮，三秒钟后，L2 熄灭，同时 L3 亮。提示：用 TOFF 和 TP 定时器实现。

#### 3. 计数器的认识实验

使用一个加/减计数器，实现当按下五次 SB3 时，灯 L4 亮；按下 SB4，则计数器复位。

## 实验二 四节传送带的模拟

### 一、实验目的

通过各基本指令，进一步熟练掌握 PLC 的基本编程和程序调试。

### 二、控制要求

对一个用四条皮带运输机的传送系统，分别用四台电动机带动，控制要求如下：启动时先启动最末一条皮带机，经过 5 秒延时，再依次启动其他皮带机。停止时应先停止最前一条皮带机，待料运送完毕后再依次停止其它皮带机。当某条皮带机发生故障时，该皮带机及其前面的皮带机应立即停止，而该皮带机以后的皮带机待运完后才停止。例如 M2 故障，M1、M2 立即停，经过 5 秒延时后，M3 停，再过 5 秒，M4 停。

### 三、传送带模拟实验面板图说明

M1~M4 用来模拟传送带的运行状态，灯亮时表示该传送带正在运行，熄灭表示停止。A,B,C,D 分别用于模拟传送带出故障时的信号，由按钮手动输入。SB2 平时为高电平，按下时为低。

### 四、输入/输出连线列表

| 面板  | M1   | M2   | M3   | M4   | A    | B    | C    | D    | SB1  | SB2  |
|-----|------|------|------|------|------|------|------|------|------|------|
| PLC | Q0.2 | Q0.3 | Q0.4 | Q0.5 | I0.1 | I0.2 | I0.3 | I0.4 | I0.0 | I0.5 |

## 实验三 LED 数码显示控制

### 一、实验目的

了解并掌握置位和复位指令 SET、RST 在控制中的应用及其编程方法。

### 二、控制要求

在 LED 数码显示控制实验区内完成本实验。

按照下面给出的控制要求编写梯形图程序，输入到可编程序控制器中运行，根据运行情况进行调试，修改程序，直到通过为止。

按下启动按钮后，有八组 LED 发光二极管模拟的八段数码管开始显示：显示一段段显示，显示次序是 A、B、C、D、E、F、G、H。随后显示数字及字符 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F、G，在返回初始显示，并循环不止。

### 三、输入输出列表

|     |      |      |      |      |      |      |      |      |      |
|-----|------|------|------|------|------|------|------|------|------|
| 面板  | A    | B    | C    | D    | E    | F    | G    | 启动   | 停止   |
| PLC | Q0.1 | Q0.2 | Q0.3 | Q0.4 | Q0.5 | Q0.6 | Q0.7 | I0.1 | I0.2 |

## 实验四 十字路口交通灯控制的模拟

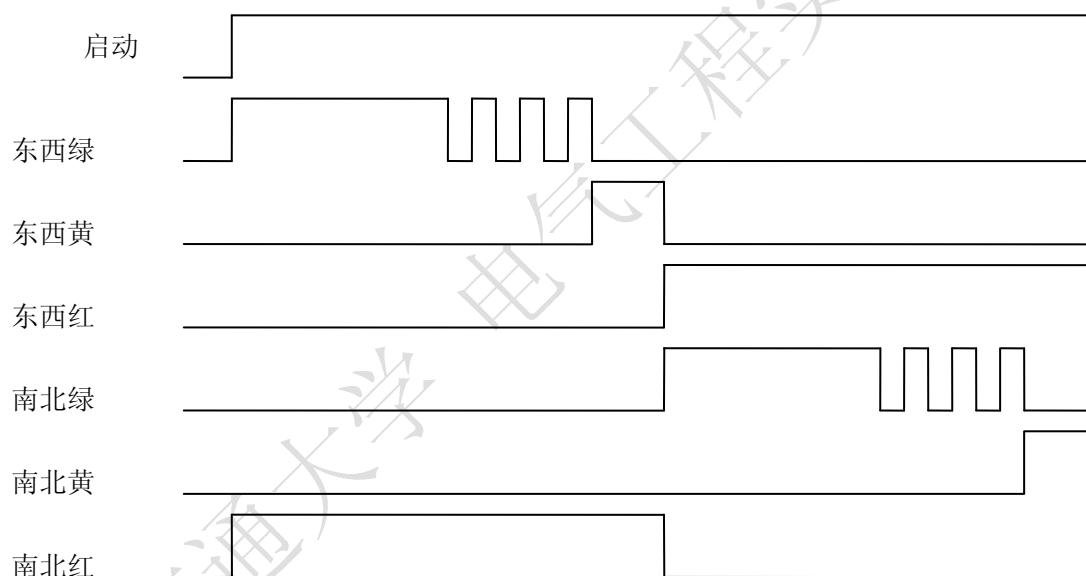
### 一、实验目的

熟练使用各基本指令，根据控制要求，掌握 PLC 的编程方法和程序调试方法，使学生了解用 PLC 解决一个实际问题的全过程。

### 二、控制要求

信号灯受一个启动开关控制，当启动开关接通时，信号灯系统开始工作，且先南北红灯亮，东西绿灯亮。当启动开关断开时，所有信号灯都熄灭。

时序图如下：



### 三、输入输出列表

|     |      |      |      |      |      |      |      |      |      |
|-----|------|------|------|------|------|------|------|------|------|
| 面板  | 启动   | 南北 G | 南北 Y | 南北 R | 东西 G | 东西 Y | 东西 R | 甲    | 乙    |
| PLC | I0.1 | Q0.2 | Q0.3 | Q0.4 | Q0.5 | Q0.6 | Q0.7 | Q0.8 | Q0.9 |

## 实验五 装配流水线控制的模拟

### 一、实验目的

了解以为寄存器指令（包括左移位，右移位指令）在控制系统中的应用及编程方法。

### 二、实验原理

使用移位寄存器指令，可以大大简化程序设计。移位寄存器指令所描述的操作过程如下：

若在输入端输入一串脉冲信号，在移位脉冲作用下，脉冲信号依次移到移位寄存器的各个继电器中，并将这些继电器的状态输出，每个继电器可以在不同的时间内得到由输入端输入的一串脉冲信号。

### 三、实验要求

传送带共有十六个工位，工件从 1 号位装入，分别在 A（操作 1）、B（操作 2）、C（操作 3）三个工位弯成三种装配操作，经最后一个工位后送入仓库；其它工位均用于传送工件。

### 四、输入/输出接线列表

|     |      |      |      |      |      |      |
|-----|------|------|------|------|------|------|
| 面板  | 启动   | 复位   | 移位   | A    | B    | C    |
| PLC | I1   | I2   | I3   | Q0.2 | Q0.3 | Q0.4 |
| 面板  | D    | E    | F    | G    | H    |      |
| PLC | Q0.5 | Q0.6 | Q0.7 | Q0.8 | Q0.9 |      |

---

## 实验六 水塔水位控制的模拟

### 一、实验目的

用 PLC 构成水塔水位自动控制系统。

### 二、实验内容

当水池水位低于水池低水位界限（S4 为 ON 表示），阀 Y 打开进水（Y 位 ON）定时器开始定时，4 秒后，如果 S4 还不为 OFF，那么阀 Y 指示灯闪烁，表示阀 Y 没有进水，出现故障，S3 为 ON 后，阀 Y 关闭（Y 为 OFF）。当 S4 为 OFF 时，且水塔水位低于水塔低水位界时 S2 为 ON，电机 M 运转抽水，当水塔水位高于水塔高水位界时点击 M 停止。

### 三、实验面板说明

面板中 S1 表示水塔的水位上限，S2 表示水塔水位下弦，S3 表示水池水位上限，S4 表示水池水位下弦，M1 为抽水电机，Y 为水阀。

### 四、输入/输出接线列表

|     |     |     |     |     |      |      |
|-----|-----|-----|-----|-----|------|------|
| 面板  | SB1 | SB2 | SB3 | SB4 | L1   | L2   |
| PLC | I1  | I2  | I3  | I4  | Q0.2 | Q0.3 |

## 实验七 天塔之光

### 一、实验目的

用 PLC 构成天塔之光模拟控制系统。

### 二、控制要求

合上启动按钮后，按以下规律显示：

L1 -> L1,L2 -> L1,L3 -> L1,L4 -> L1,L5 -> L1,L2,L4 ->  
L1,L3,L5 -> L1 -> L2,L3,L4,L5 -> L1,L6 -> L1,L7 -> L1,L8 -> L1,L9  
->L1 -> L6,L7,L8,L9 -> L10 -> L11 -> L12->L1……如此循环，周而复始。

### 三、输入/输出接线列表

|     |     |     |      |      |      |      |      |      |      |
|-----|-----|-----|------|------|------|------|------|------|------|
| 面板  | SB1 | SB2 | L1   | L2   | L3   | L4   | L5   | L6   | L7   |
| PLC | I1  | I2  | Q0.2 | Q0.3 | Q0.4 | Q0.5 | Q0.6 | Q0.7 | Q0.8 |

|     |      |      |       |       |       |  |  |  |  |
|-----|------|------|-------|-------|-------|--|--|--|--|
| 面板  | L8   | L9   | L10   | L11   | L12   |  |  |  |  |
| PLC | Q0.9 | Q.10 | Q0.11 | Q0.12 | Q0.13 |  |  |  |  |

## 实验八 液体混合装置控制的模拟

### 一、实验目的

熟练使用各条基本指令，通过对工程实例的模拟，熟练地掌握 PLC 的编程和程序调试。

### 二、控制要求

本装置为两种液体混合模拟装置，SL1、SL2、SL3 为液面传感器，液体 A、B 阀门与混合液阀门由电磁阀 YV1、YV2、YV3 控制，M 为搅匀机电机，控制要求如下：

初始状态：装置投入运行时，液体 A、B 阀门关闭，混合液阀门打开 8 秒将如期放空后关闭。

启动操作：按下启动按钮 SB1，装置就开始按下列约定的规律操作：

液体 A 阀门打开，液体 A 流入容器。当液面到达 SL2 时，SL2 接通，关闭液体 A 阀门，打开液体 B 阀门。液面到达 SL1 时，关闭液体 B 阀门，搅匀电机开始搅匀。搅匀电机工作 5 秒钟后停止搅动，混合液体阀门打开，开始放出混合液体。当液面下降到 SL3 时，SL3 由接通变为断开，再过 2 秒后，容器放空，混合液体阀门关闭，开始下一周期。

停止操作：按下停止按钮 SB2 后，在当前的混合液操作处理完毕后，才停止操作（停止在初始状态）。

### 三、输入/输出接线列表

|     |      |      |      |      |      |      |      |      |      |
|-----|------|------|------|------|------|------|------|------|------|
| 面板  | Y1   | Y2   | Y3   | M    | L1   | L2   | L3   | SB1  | SB2  |
| PLC | Q0.2 | Q0.3 | Q0.4 | Q0.5 | I0.0 | I0.1 | I0.2 | I0.3 | I0.4 |

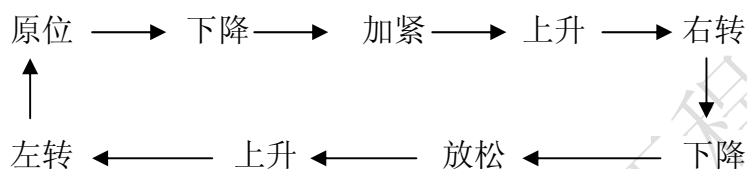
## 实验九 机械手的模拟控制

### 一、实验目的

用数据移位指令来实现机械手动作的模拟。

### 二、控制要求

图中为一个将物件从 A 处移向 B 处的机械手，上升/下降和左移/右移的执行用双线圈二位电磁阀推动气缸完成。而我们这个试验其实只要求看到灯的亮灭即可。当某个电线圈通电，就一直保持现有的机械动作。另外，夹紧/放松由单线圈二位电磁阀推动气缸完成，线圈通电执行夹紧动作，断电执行放松动作。工作过程如下：



### 三、机械手动作的模拟试验面板图及接线列表

此面板中的启动，停止用动断按钮来实现，限位开关用钮子开关来模拟，电磁阀和原位指示灯用发光二极管来模拟。

|     |      |      |      |      |      |      |      |      |      |
|-----|------|------|------|------|------|------|------|------|------|
| 面板  | YV1  | YV2  | YV3  | YV4  | YV5  | SQ1  | SQ2  | SQ3  | SQ4  |
| PLC | Q0.2 | Q0.3 | Q0.4 | Q0.5 | Q0.6 | I0.1 | I0.2 | I0.3 | I0.4 |

|     |      |      |
|-----|------|------|
| 面板  | SB1  | SB2  |
| PLC | I0.5 | I0.6 |

## 实验十 组态监控实验

### 一、实验目的

熟悉 Citect 组态软件的使用，通过以太网通信实现对 PLC 的监控。

### 二、实验要求

在选做的实验中，选择一个，将其 I/O 的输入修改为中间继电器输入%Mi，通过内部中间继电器来触发控制，然后在组态软件中组态监控界面、配置监控变量，设置通讯，最终实现对原基础程序的组态监控。

### 三、示例程序参考步骤

下面通过一个简单示例来介绍通过 Citect 组态软件对 PLC 进行监控，主要内容是在通过组态软件来配置界面控件、设置监控变量、配置通信参数，然后通过以太网来控制 PLC 的开关量输出。具体参考步骤如下：

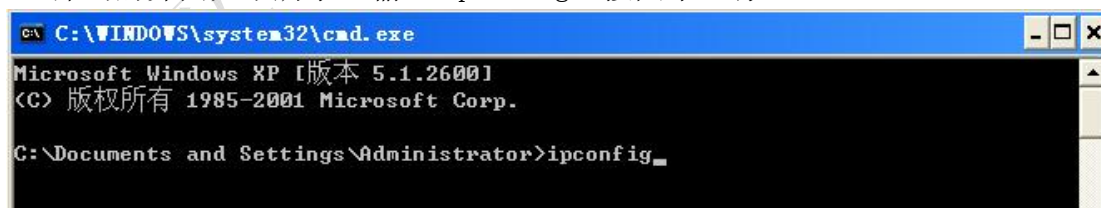
#### 1、查看计算机 IP

在建立计算机与 PLC 通讯时，需要填写 IP 地址及相关设置，以下步骤将演示如何查看本机的 IP 地址。

在 windows 左下角“开始”菜单中选择“运行”，输入“cmd”，按回车。



弹出的界面如下所示，输入 ipconfig，按回车运行。



如果计算机不执行 ipconfig 命令，是由于 XP 没有 DOS 模式，可能是系统的 Path 被更改了，输入 ping 等命令要输入完整路径才可以，比如

C:\windows\system32\ping.exe

此时，在我的电脑右键属性，“高级”选项卡，最下面的“环境变量”，进入系统变量，找到“path”一项，按“编辑”，，，变量值前面的不要动，在后面加上分号后输入系统安装的路径 C:\windows\system32\

再次进入 cmd 界面，并输入 ipconfig，这时，计算机的网络参数会显示如下。

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>ipconfig

Windows IP Configuration

Ethernet adapter 本地连接:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . . : 219.228.111.164
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 219.228.111.254

Ethernet adapter 无线网络连接:

    Media State . . . . . : Media disconnected

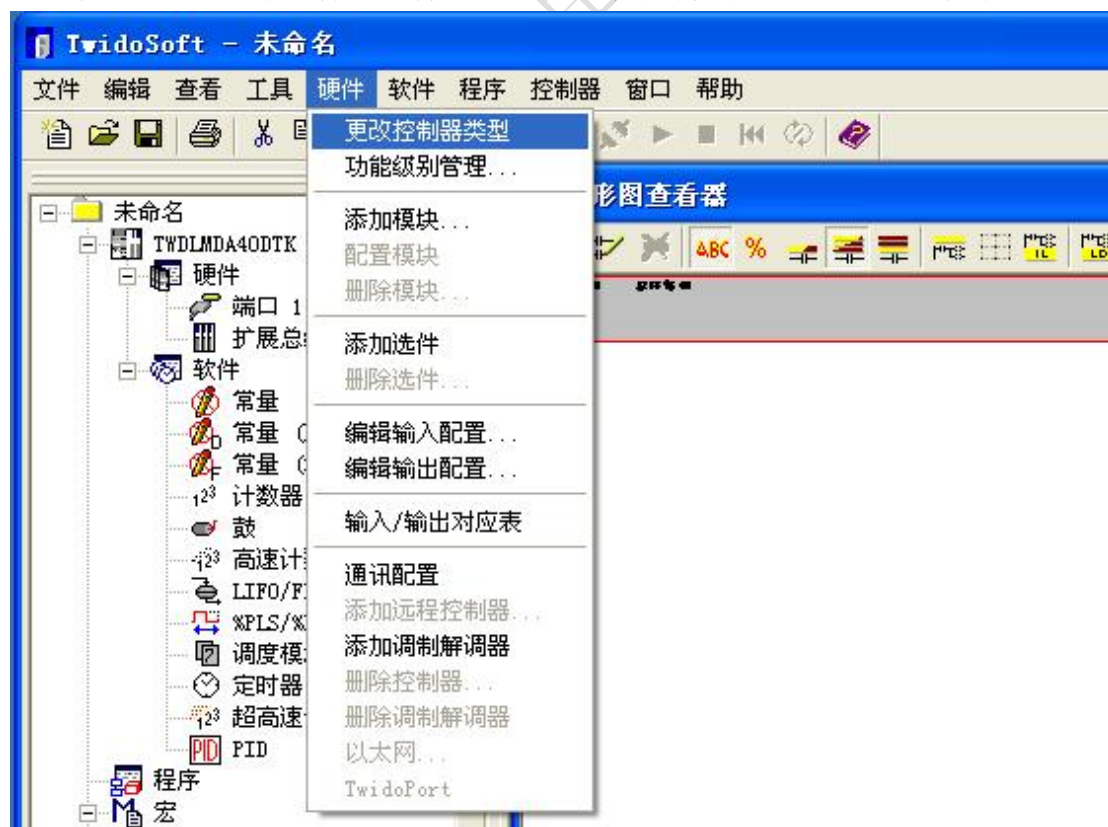
C:\Documents and Settings\Administrator>
```

IP Address 是本机的 IP 地址，在设置 PLC 的 IP 时，IP 地址前 3 个字段必须一致。上图中 subnet mask 是子网掩码，Gateway 是网关，在 PLC 以太网设置时都需要用到。

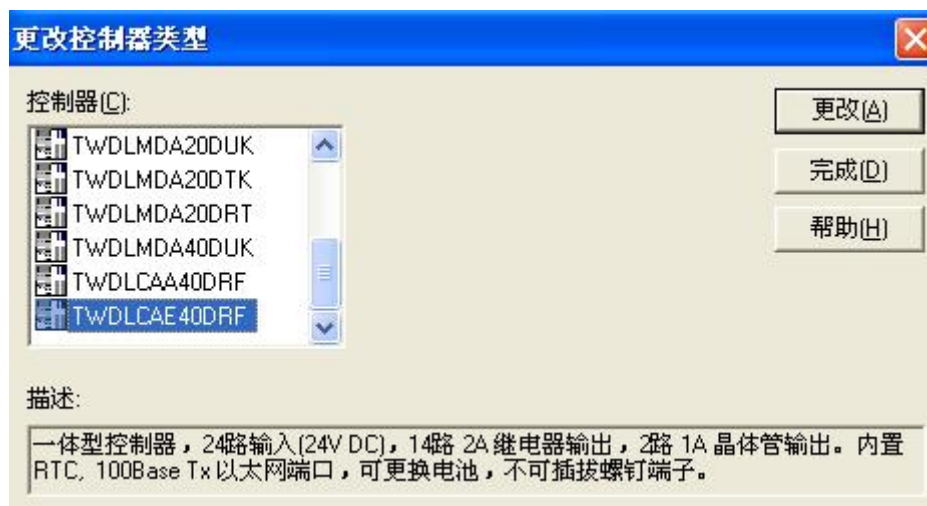
## 2、PLC 以太网通讯设置

### (1) 更改控制器类型

在 twido soft 中选择“硬件”——“更改控制器类型”，入下图所示。



选择“TWDLCAE40DRF”，按“更改”。



## (2) 以太网设置

当选择正确的控制器类型后，在左边的窗口中，“硬件”属性里出现“以太网端口”。



双击“以太网端口”，弹出的界面如下。选择第一选项卡“IP 地址配置”——“使用指定 IP 地址”。



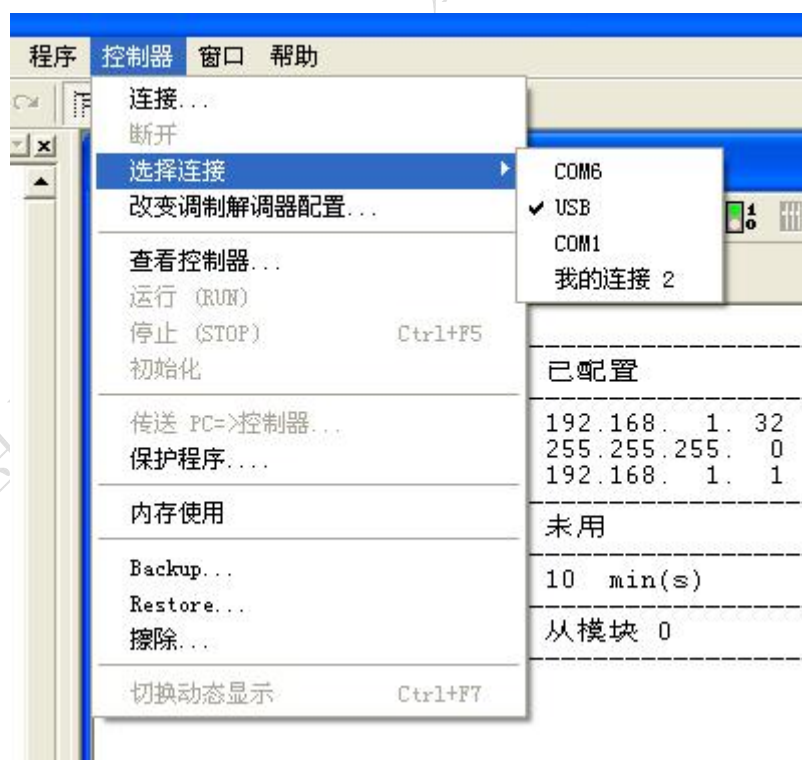
“子网掩码”和“网关”输入在 IPconfig 中查询到的地址。

“IP 地址”中自定义一个 IP，（前三个字段须与网关前三个字段相同）。

### (3) 下载程序

在完成上述设置，并编写好程序后，在菜单栏中选择“控制器”——“USB”。

（如果下载线为 TSXCUSB485，此处选择 COM7、COM8 或 COM9 等）。

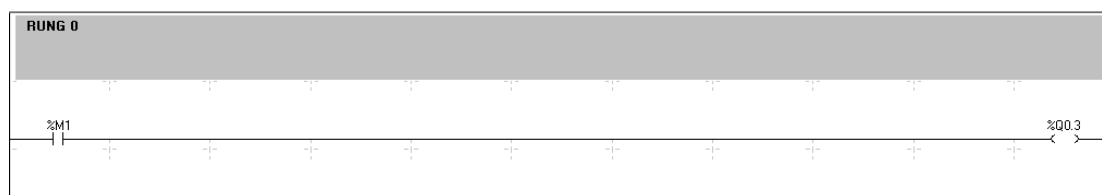


最后，点击“连接”，将程序下载到 PLC 中。

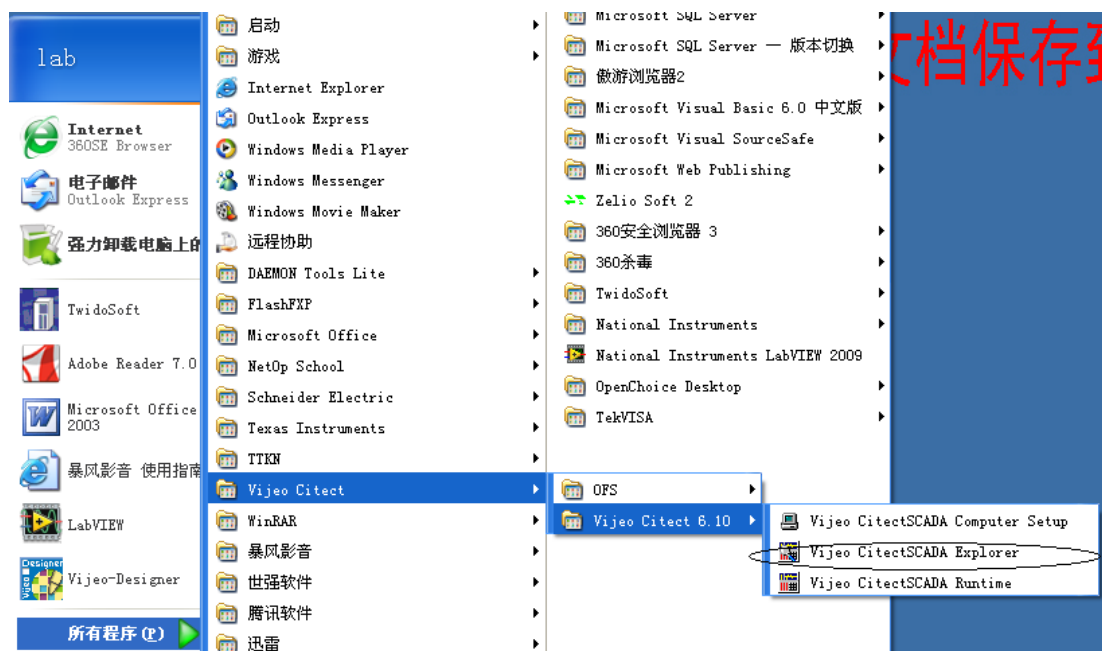
## 3、利用 citect 实现在电脑上对 PLC 的监控

我们通过 citect 软件在电脑上建立界面，以此来控制 twido 中的程序。我们

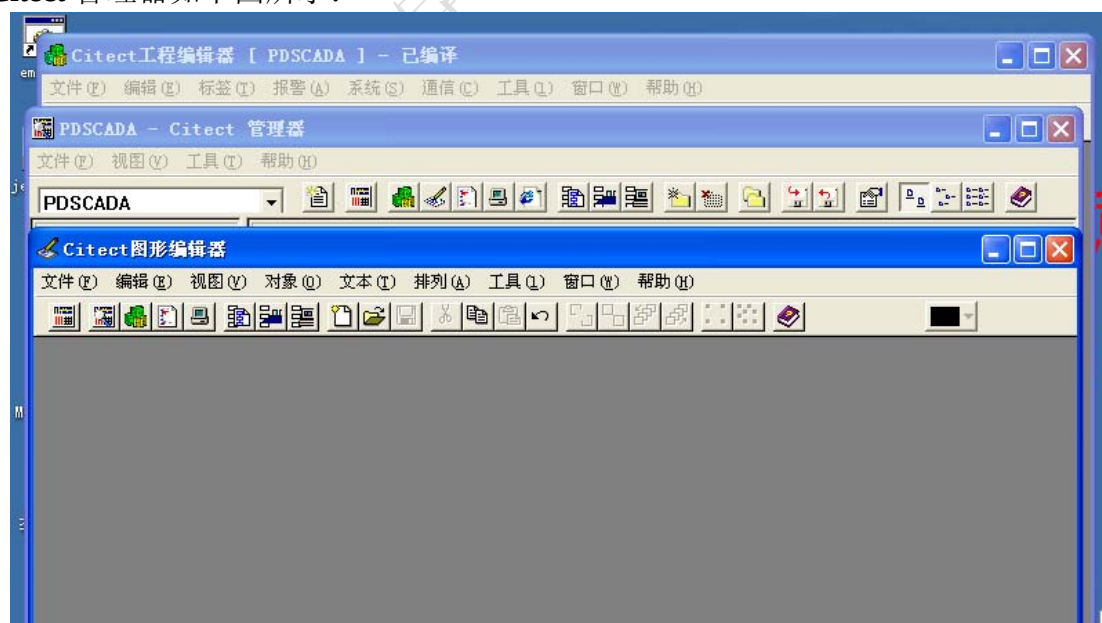
要实现的是下面这个简单的控制程序。



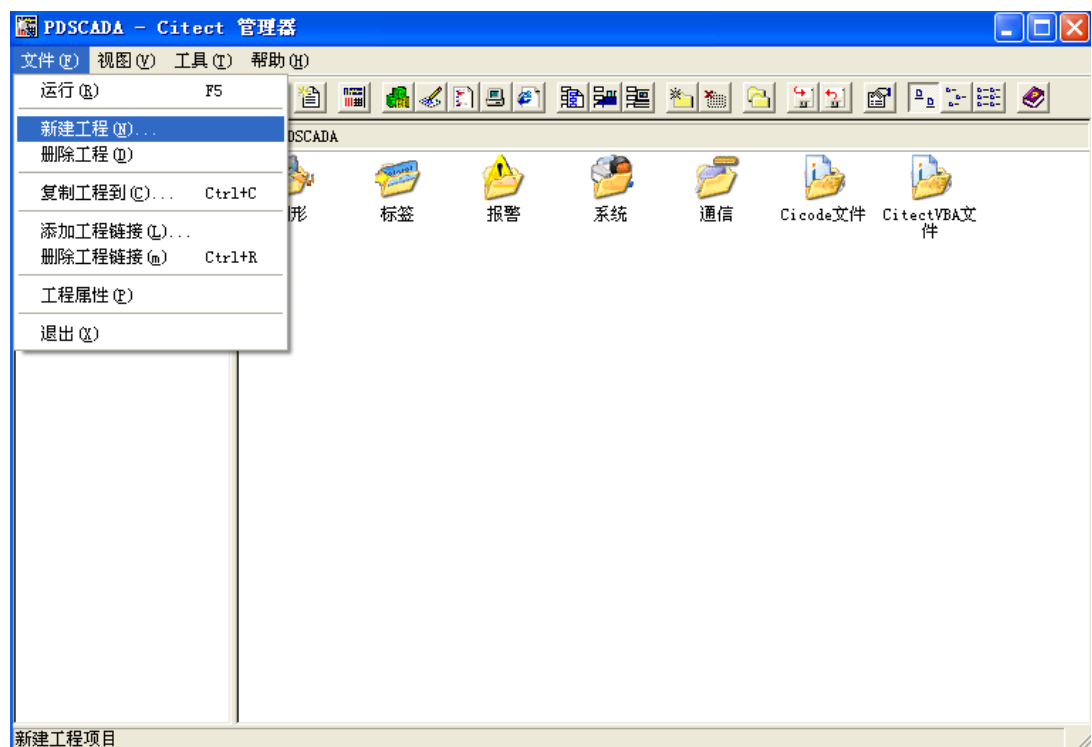
首先在程序中找到 Citect 程序，如下图所示。



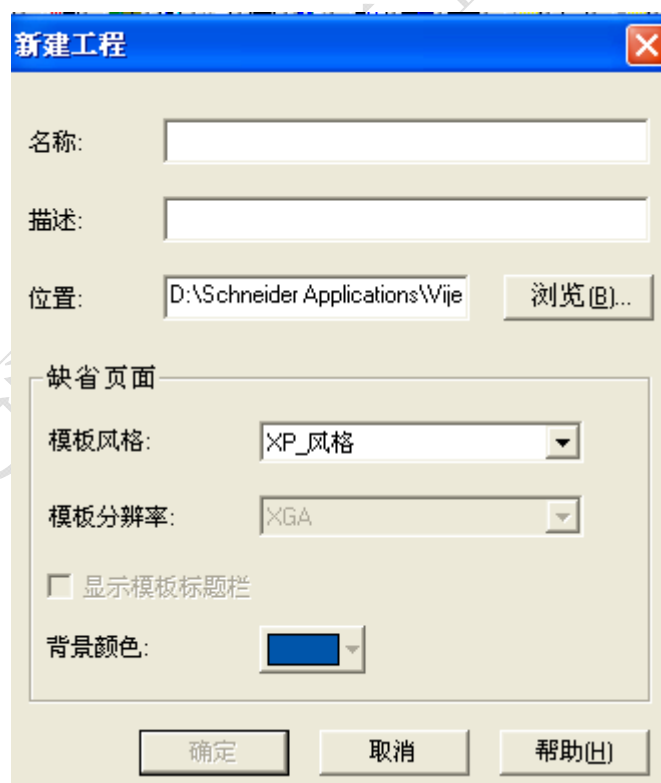
打开后会出现三个对话框：Citect 工程编辑器、Citect 图形编辑器、PDSCADA Citect 管理器如下图所示：



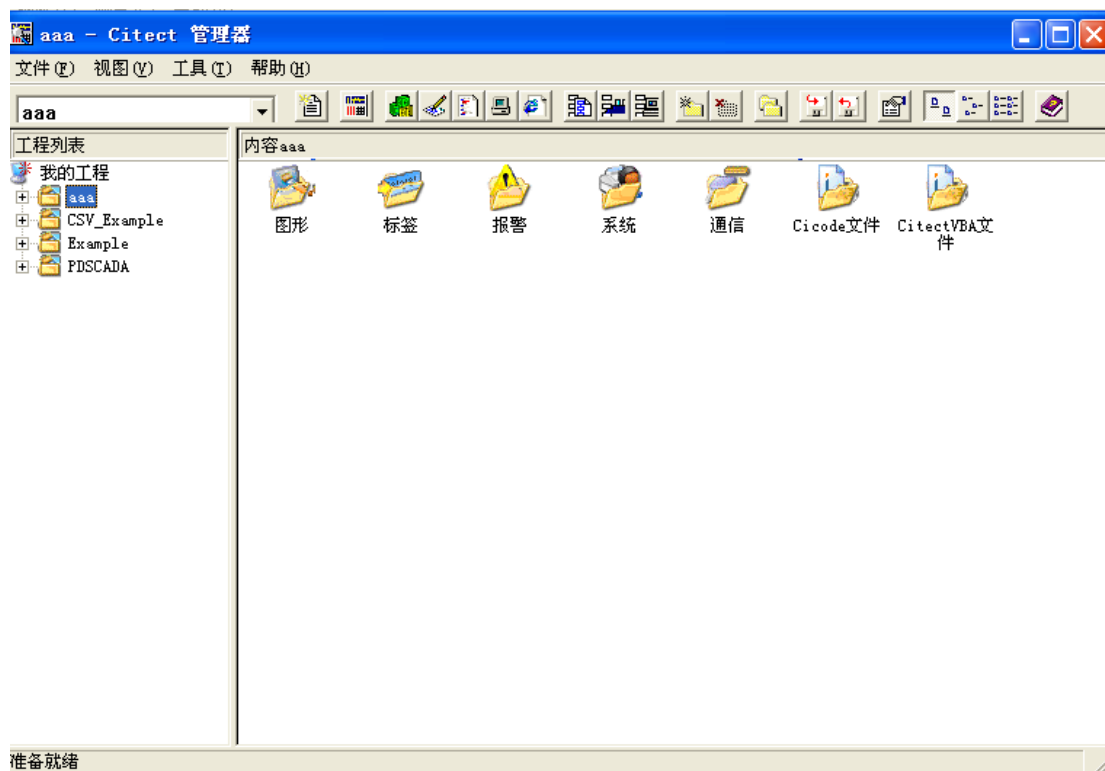
首先新建一个工程，在 Citect 管理器中新建一个工程，操作如下图所示：



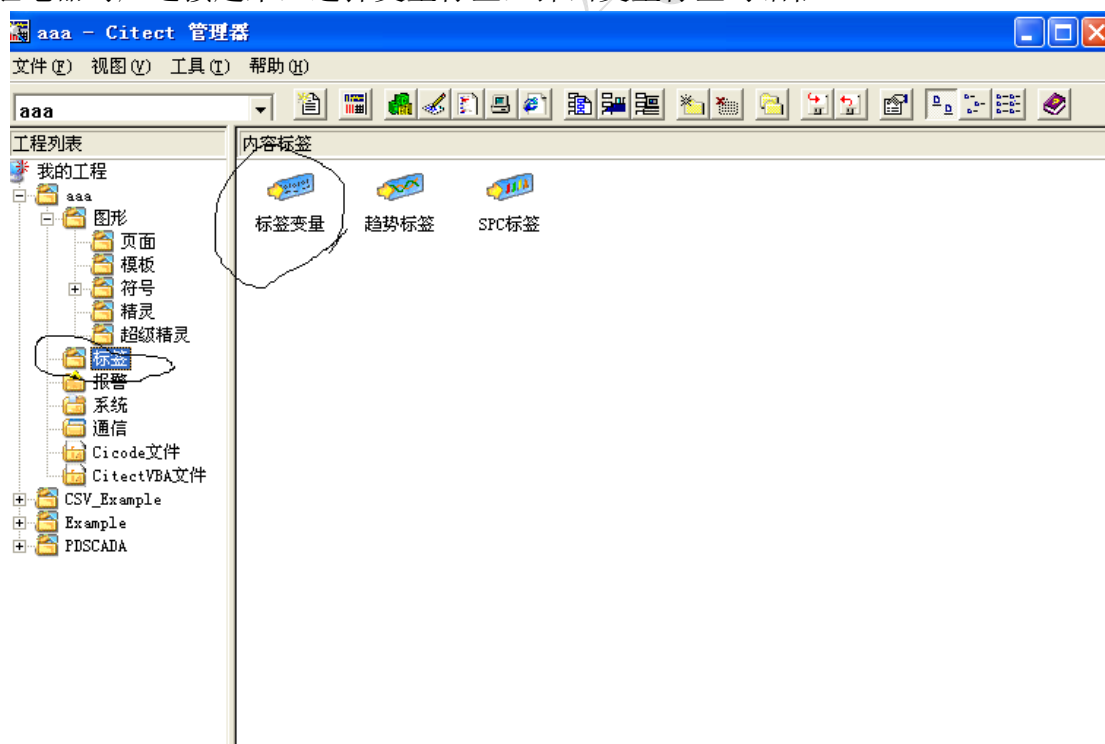
弹出如下对话框，填好即可。



这样在管理器中出现了我们新建的工程 aaa。



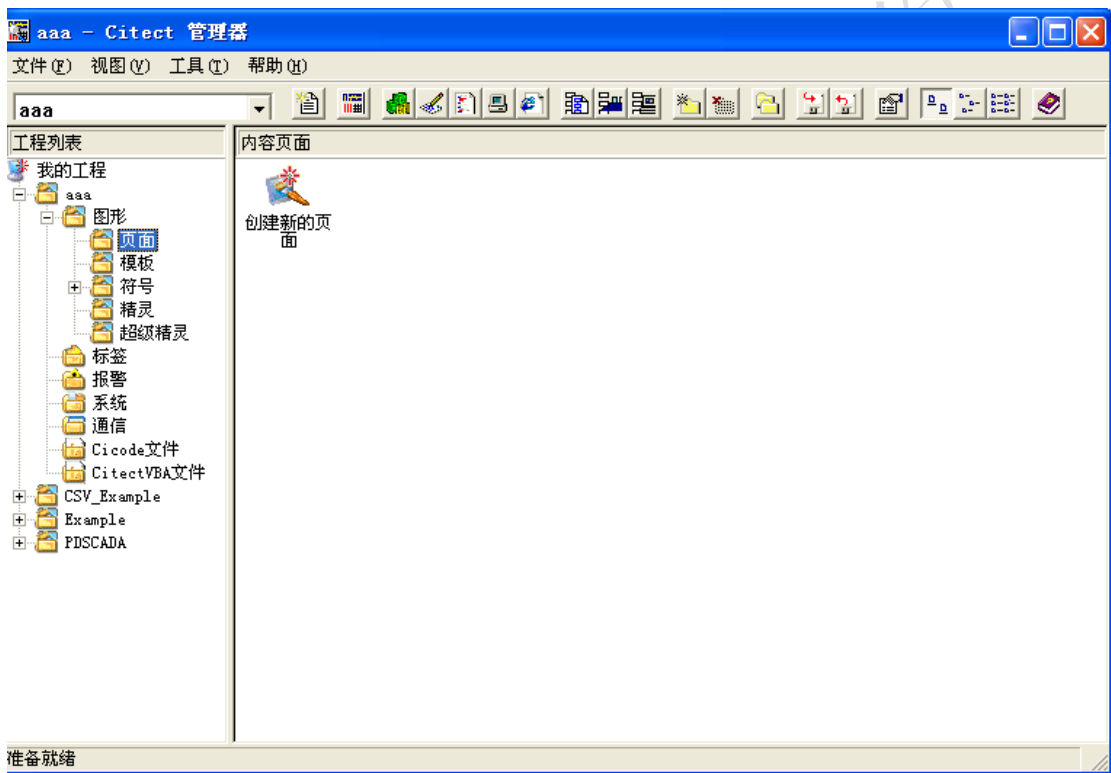
然后在工程中新建标签，标签的作用就是把我们的图形界面和 twido 中的各个继电器对应连接起来，选择变量标签，弹出变量标签对话框。



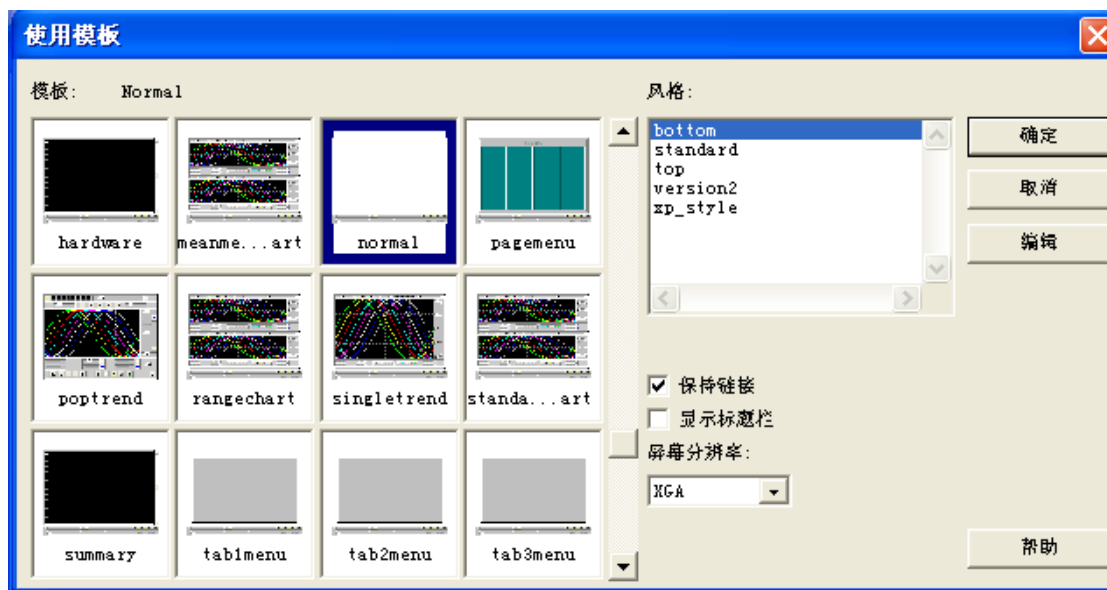
变量标签的名称与你 twido 中用的中间继电器的名称一致，地址就是你中间继电器所对应的物理地址，如果使用的是%M1 等字节型，则地址类型为 0000+1+M 后面的数字,数据类型为 digital。



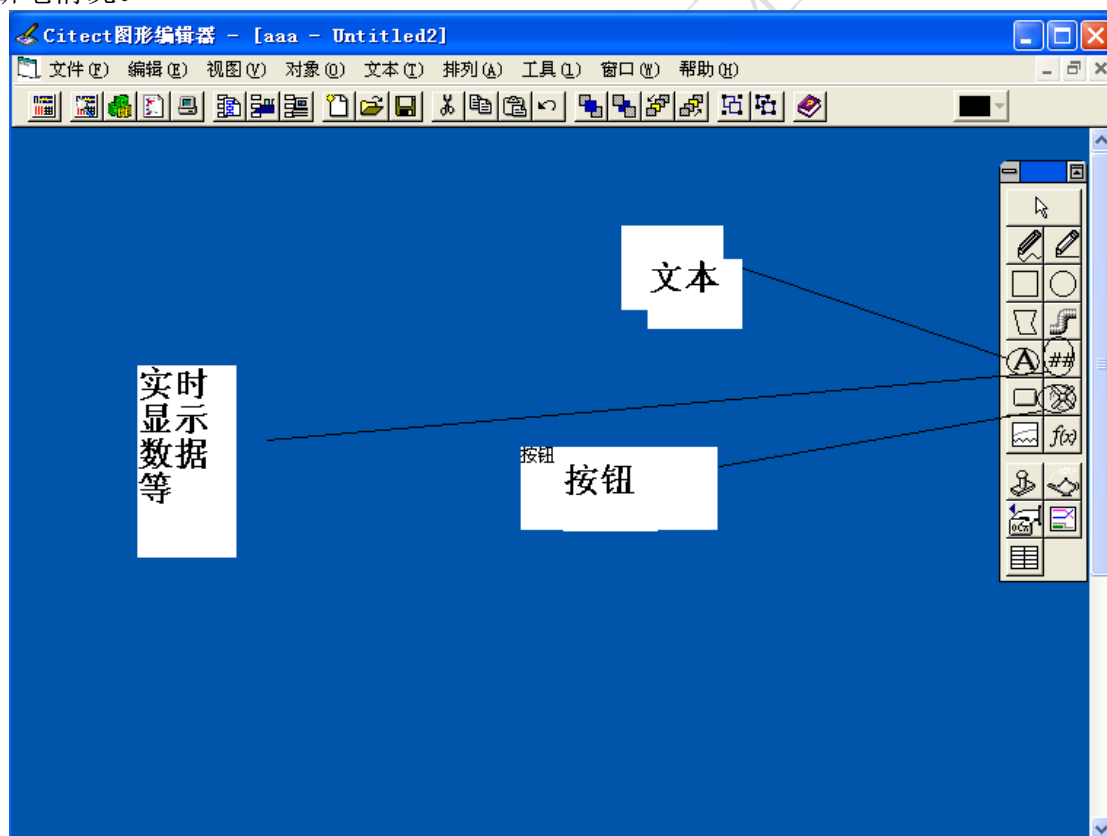
定义好变量标签后我们就可以根据需求来建立自己的页面，点击图形选相中的创建新的页面弹出如下对话框，



选择你需要的模板即可。



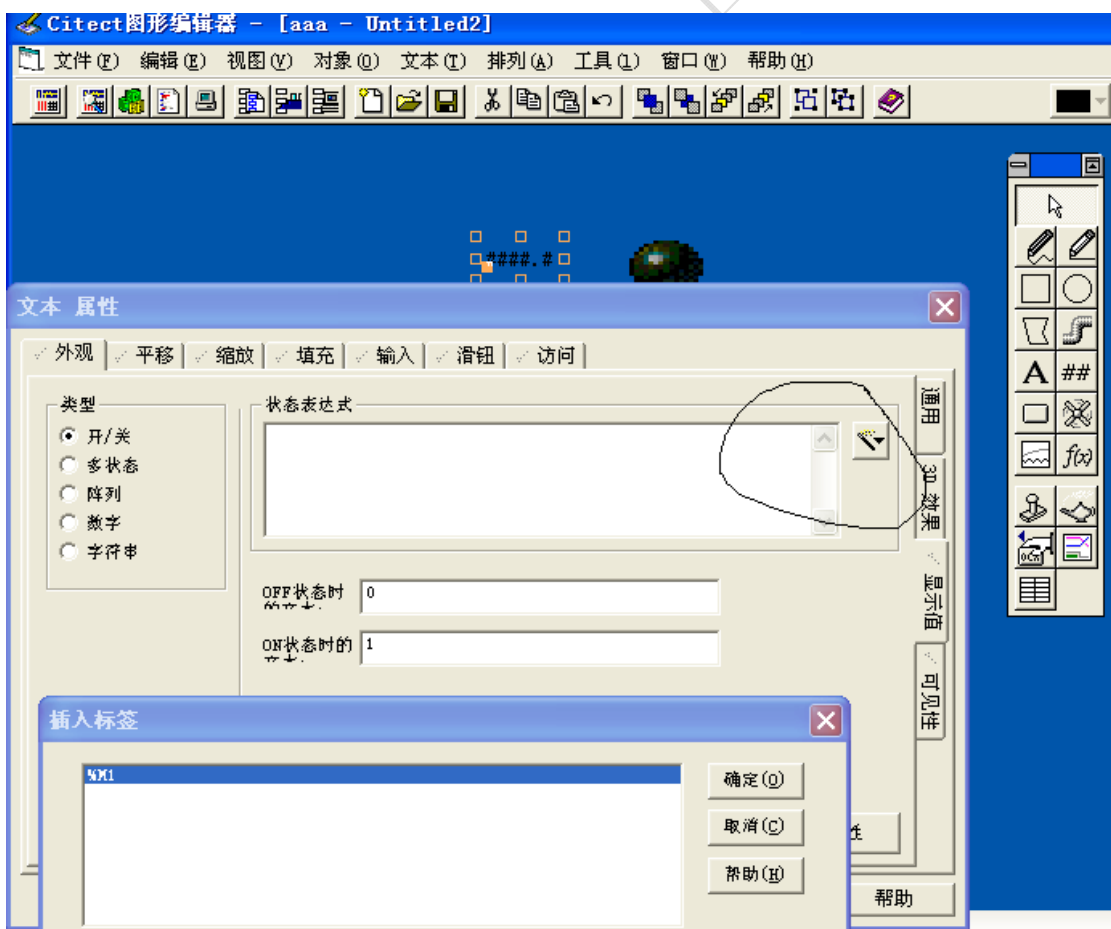
选中模板后图形编辑器会自动弹出，就可以建立自己的页面了。因为我们只要用到一个按钮，并通过直接对界面上按钮的开关作用，控制%M1 中间继电器的通断电情况。



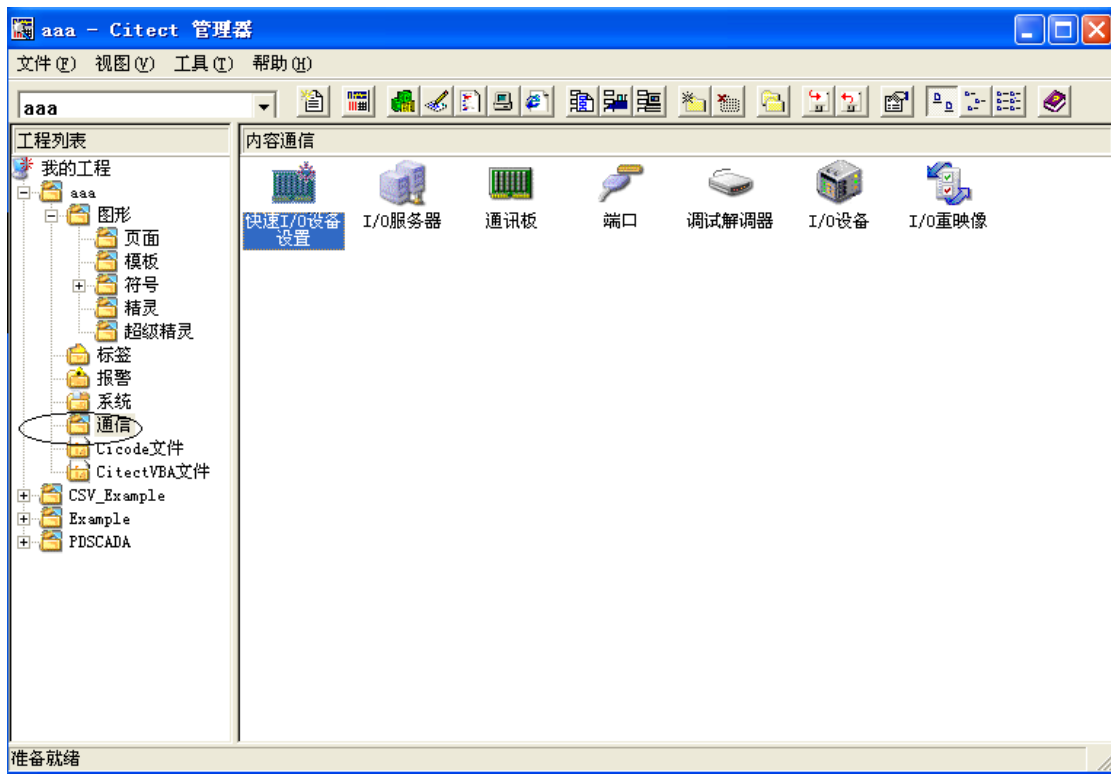
我们首先在界面上放置按钮和数据符号，双击按钮就会弹出其属性设置对话框，选择其类型为开关，并插入变量标签，就是我们一开始建立的变量标签，以此来建立按钮和 twido 中%M1 的对应关系。



其次，我们还可以在按钮旁边放置数据来显示它的通断情况，双击数据显示符号，就会弹出其属性设置对话框，如下图所示：开通状态显示的内容可用户自行定义，然后还要插入标签，同样使该数字和 `twido%M1` 建立连接，使其能够动态显示 `%M1` 的通断状况。



页面建好后，我们最后应该实现 `twido` 和 `citect` 之间的通信，使得能在电脑上通过 `citect` 中界面操作来控制 `plc` 运行情况，在通信中选择快速 I/O 设备设置。



弹出如下对话框



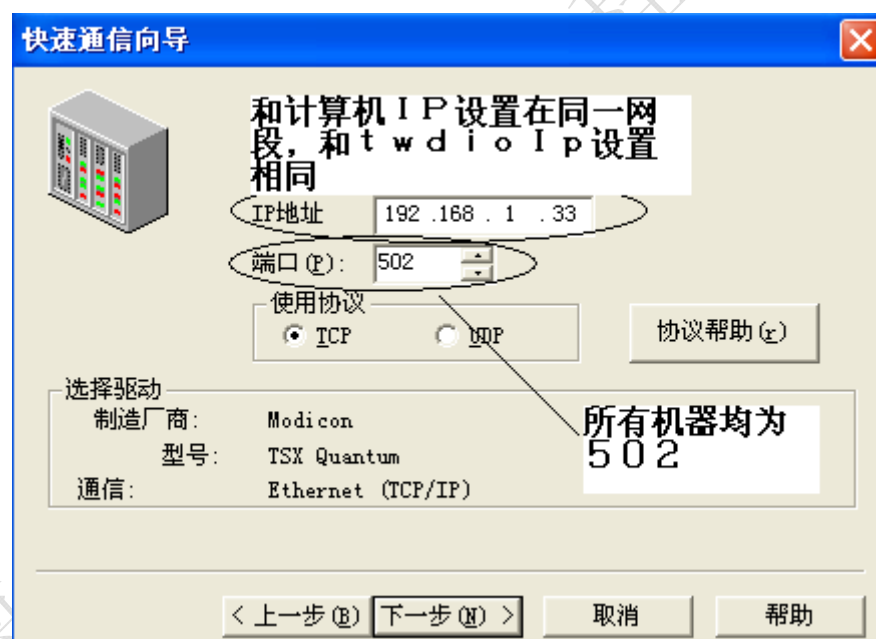




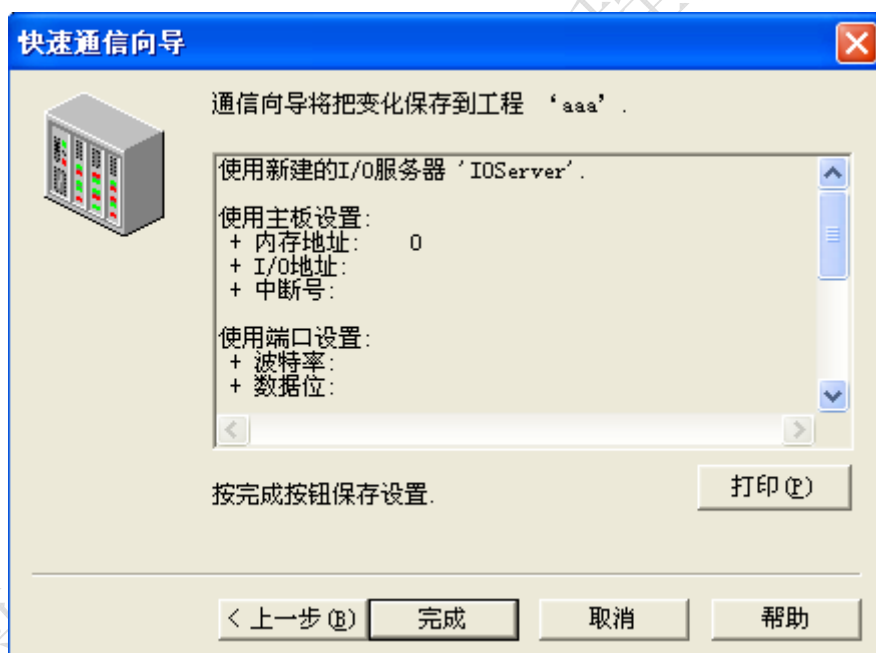


驱动程序需要注意，我们选择 modican 中的 TSX Quantum 如下图所示：

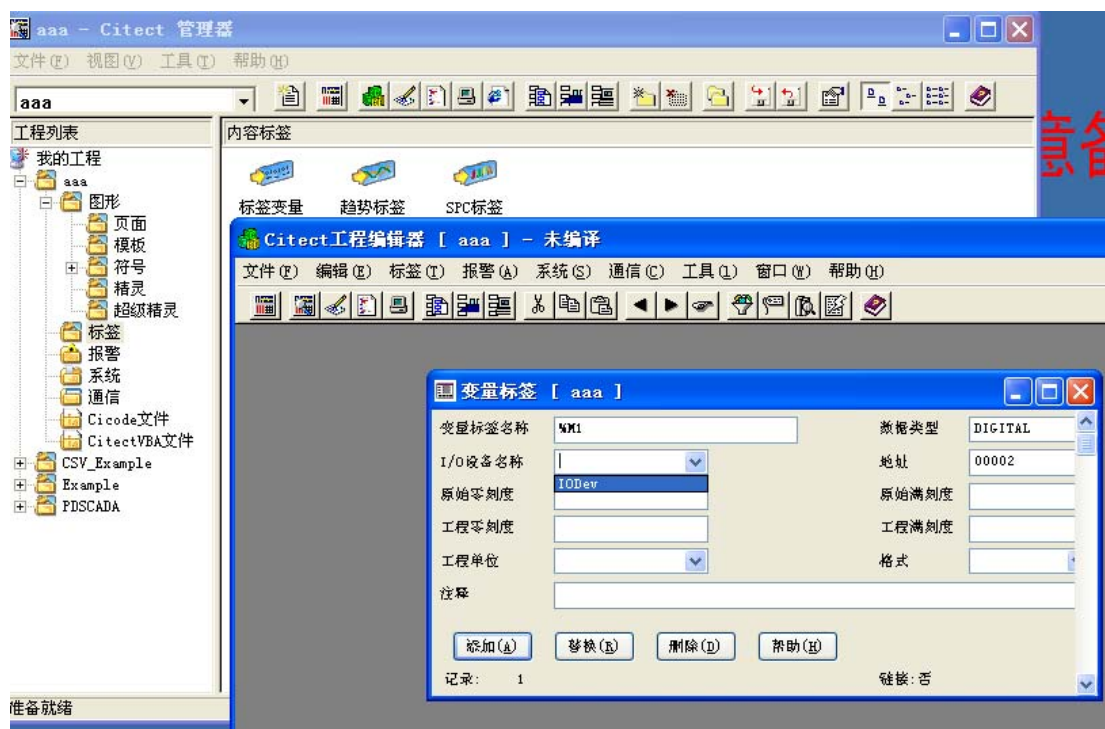




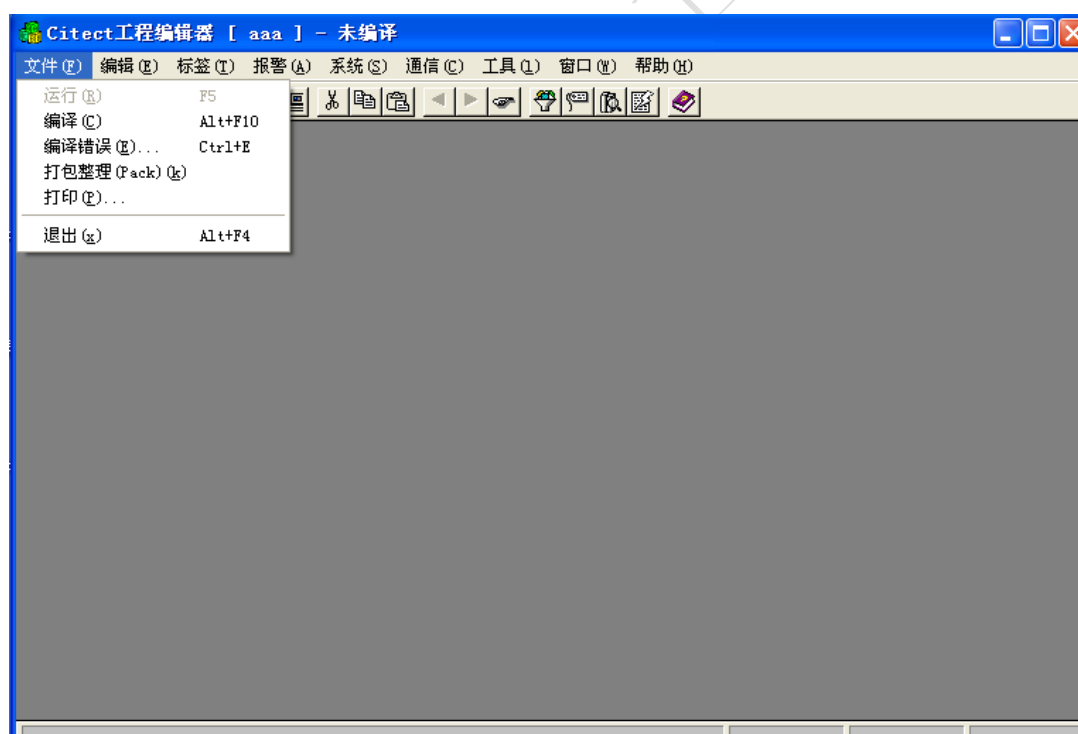
下面讲一下怎么获取计算机的 IP 地址，



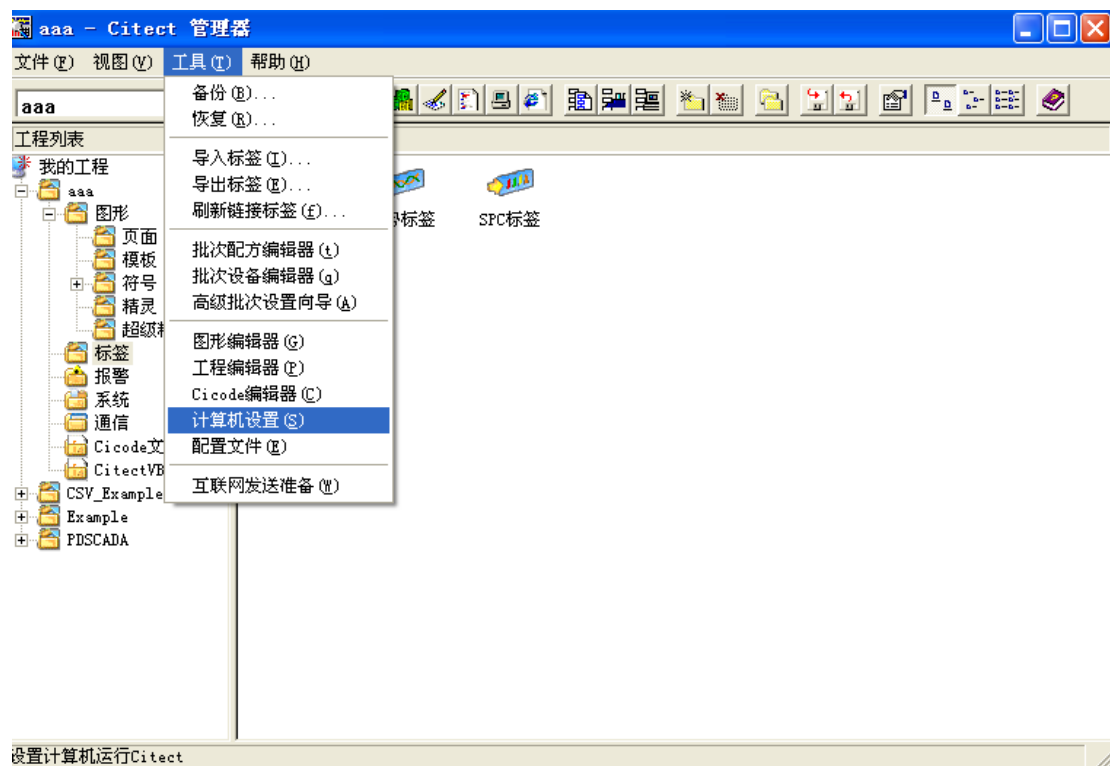
最后我们还需要改变的是我们的标签，在 citect 管理器中找到标签中的变量标签，弹出如下对话框，我们需要把 I/O 设备这一栏重新填好。然后选择替换即可。



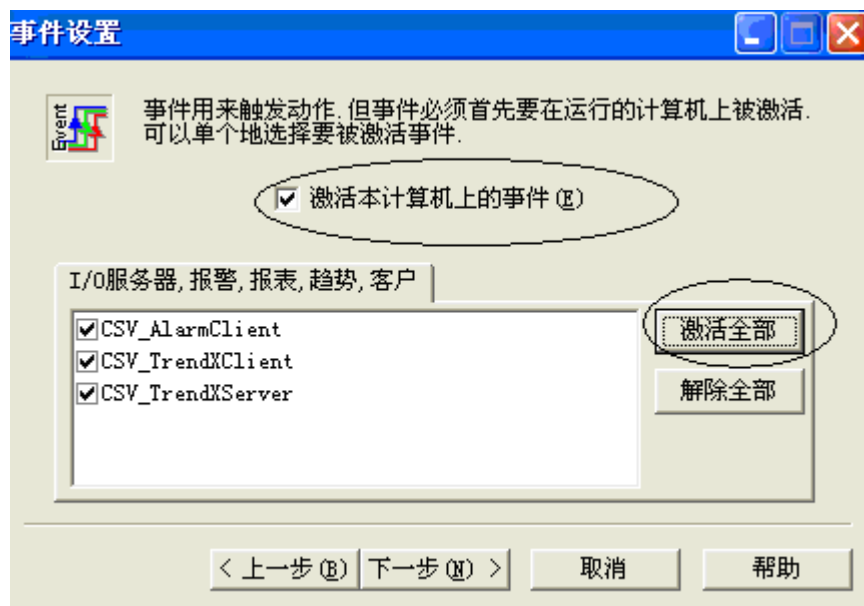
然后在工程编辑器中编译一下我们的工程。



编译完成后，在 citect 管理器中工具栏下面找到计算机设置，修改一下计算机的设置我们就可以运行了。



一直下一步直到出现下面的对话框。



如上图所示激活计算机上的全部事件。然后继续下一步直到下面的选项单，在启动页面上选择自己工程的首页。



完成后即可运行程序进行测试。