

# 简单使用C语言写梯形图控制

V1.0

2008.09.02

开发环境: **KEIL C**

免责声明:

- 1: 因本人知识有限, 只写了常用功能, 程序的写法很粗陋, 不入高手眼界, 请海涵。
- 2: 此程序写法供懂**PLC**而不懂单片机又想用单片机的菜鸟参考。有误人子弟之处请谅解。
- 3: 此程序控制模块针对我控制板写的, 使用其他控制板出现问题, 本人概不解答。
- 4: **C**程序仅留**main.C**源程序, 编写者在**PLC**函数内编写逻辑程序。其他驱动封装为库文件, 并说明了调用函数功能。

# 具备知识

## C51基本知识:

买**C51**与**keil C**软件使用的书，掌握下面知识。

### 1: 基础知识:

标识符与关键字

数据类型

常量与变量

函数

运算符

### 2: 常用的C语言控制命令:

**if**语句

嵌套**if**语句

**Switch**语句

**for**循环

**While**循环

**do/while**循环

标识和**goto**语句

### 3: 知道使用**keil C**编程方法与编译程序以及烧录程序生成。

## 具备软硬件:

### 1: 软件,

(1): 安装了**keil C**软件。

(2): 安装了程序下载软件，如**AT89S5X**的**ISP**下载软件，  
或者有**STC**单片机的串口下载软件。

### 2: 单片机控制板，带程序下载口的。

### 3: 下载工具: 用**AT**单片机，买个下载器，25元。

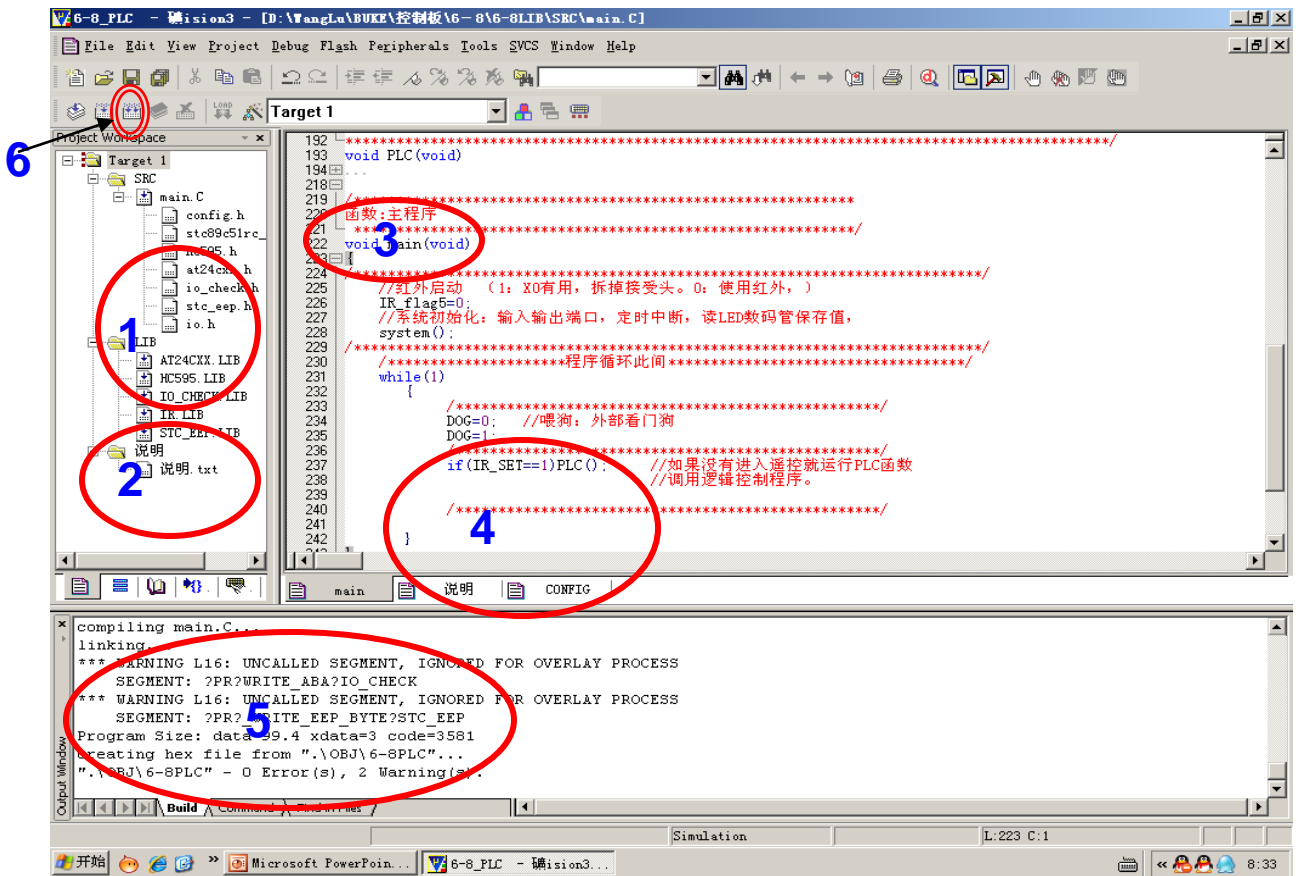
用**STC**单片机，买根串口下载器，8元。

# 目 录

- 一： 模块程序结构
- 二： 模块程序内变量说明
- 三： 模块程序函数接口说明
- 四： 把例子添加到**PLC**函数编程
- 五： 把编译完成文件烧录到单片机
- 六： **C**语言实现梯形图写法
- 七： **6—8**控制板资源。
- 八： **6—8**控制板简单使用例子。

# 一：模块程序结构

## 1：软件界面



- 1: 主程序包含的.H文件。
- 2: PLC函数，你在这里写你的逻辑控制功能函数。
- 3: 主程序的while循环，程序循环运行执行PLC函数在这里。
- 4: 这是AT24CO2,数码管，红外，STC内部存储数据的驱动程序，
- 5: 软件编译信息。
- 6: 编译按键，软件修改或者写完，按次按键，在软件编译信息栏显示编译信息，如果有错误与警告，上面会显示。

## 2: 主程序界面

Main程序看函数后面注释理解，while循环里不要修改。

```

/*****
函数:主程序
*****/
void main(void)
{
/*****
//红外启动 (1: X0有用, 拆掉接受头。0: 使用红外, )
IR_flag5=0;
//系统初始化: 输入输出端口, 定时中断, 读LED数码管保存值,
system();
*****/
/*****程序循环此间 *****/
while(1)
{
/*****
DOG=0; //喂狗: 外部看门狗
DOG=1;
*****/
if (IR_SET==1) PLC(); //如果没有进入遥控就运行PLC函数
//调用逻辑控制程序。

/*****
}
}

```

## 2: PLC函数

PLC函数中添加你的逻辑程序就可以了。

```

/*****
void PLC(void)
{
/*-----这里添加你的逻辑程序-----*/
if (X0==0) Y0=0;
else Y0=1;

if (X1==0) Y1=0;
else Y1=1;

if (X2==0) Y2=0;
else Y2=1;

if (X3==0) Y3=0;
else Y3=1;

if (X4==0) Y4=Y5=0;
else Y4=Y5=1;

if (X5==0) Y6=Y7=0;
else Y6=Y7=1;

/*-----*/
if (ABA>=ABC) Y0=0; //如果实际数量大于或者等于计划数量。Y0等量。
}

```

变量与输入输出定义根据我的板子写的，供参考。

```

/*-----*/
                输入输出接口
输入：          输出：
X0: P3_2       Y0: P2_4
X1: P3_3       Y1: P2_5
X2: P3_4       Y2: P2_6
X3: P3_5       Y3: P2_7
X4: P3_6       Y4: P2_0
X5: P3_7       Y5: P2_1
                Y6: P2_2 (继电器1)
                Y7: P2_3 (继电器2)
                Y10:P0_4 (指示灯)
-----
串口：          P3_0 P3_1
24C02：         P0_6 P0_7
看门狗：        P1_4
LED显示：
HC595-DAT :P0_1
HC595-RCK :P0_2
HC595-CLK :P0_3
/*-----*/
extern uint16 ABA;          // LED数码管显示值
extern uint16 ABC;          // 计划数量显示值
/*-----*/

```

```

/*P0*/
sbit Y10 = 0x84; //指示灯
/* P2 */
/*****P2*****/
sbit Y4 = 0xA0; //输出Y4
sbit Y5 = 0xA1; //输出Y5
sbit Y6 = 0xA2; //输出Y6
sbit Y7 = 0xA3; //输出Y7
sbit Y0 = 0xA4; //输出Y0
sbit Y1 = 0xA5; //输出Y1
sbit Y2 = 0xA6; //输出Y2
sbit Y3 = 0xA7; //输出Y3
/*****P1*****/

sbit DOG = 0x94; //外部看门狗
/*****P3*****/
sbit XX0 = 0xB2; //输入红外或者X0
sbit XX1 = 0xB3; //输入X1
sbit XX2 = 0xB4; //输入X2
sbit XX3 = 0xB5; //输入X3
sbit XX4 = 0xB6; //输入X4
sbit XX5 = 0xB7; //输入X5
/*****P3*****/

```

```

sfr WDT      = 0xA6;
sfr WDTSTC   = 0xE1;

```

```
extern bit M0, M1, M2, M3, M4, M5, M6, M7;
extern bit M10, M11, M12, M13, M14, M15, M16, M17;
extern bit M20, M21, M22, M23, M24, M25, M26, M27;
extern bit M30, M31, M32, M33, M34, M35, M36, M37;
```

内部软继电器，初始值为1。

```

//*****定时器动作开关
extern bit    T0_bit,T1_bit,T2_bit,T3_bit,T4_bit,T5_bit,
extern bit    T10_bit,T11_bit;
//*****定时器定时时间****

```

```
extern uint16 D0,D1,D2,D3,D4,D5,D6,D7;
/*****计数器*****/
```

```
extern bit CO_K=1,C1_K=1,C2_K=1,C3_K=1,C4_K=1,C5_K=1,C6_K=1,C7_K=1; //初始值为1，为0打开
```

```

/*****以下是红外遥控外部数据 *****/

```

```
extern bit IR_flag1; //设定保存数据
extern bit IR_flag2; //设定保存数据
extern bit IR_flag5; //IP驱动
extern bit IR_flag7;
extern uint8 CAO;
extern uint8 IR_SET; //1: 正常 2: 设定计划数量 3: 设定IP.
```

### 三：模块程序函数接口说明

变量与输入输出定义根据我的板子写的，供参考。

```
/*-----  
HC595函数  
extern void display_595(void);      //显示ABA的4位LED值  
/*-----  
AT24CXX函数  
/*-----  
功能：写数据到目标地址  
aDD: 地址  
DData: 数据（8位数据）  
例说：write(0x00,ABA); 写ABA的数到0x00地址。  
-----*/  
extern void write(unsigned char aDD,unsigned char DData);  
/*-----  
功能：读地址数据  
aDD: 地址  
例说：ABA=read(0x00); 读0x00地址数到ABA。  
-----*/  
extern unsigned char read(unsigned char aDD);  
/*-----  
输入输出函数  
-----*/  
extern void system(void); //系统初始化  
extern void write_ABA (void); //写ABA数码管值到AT24C02  
extern void write_ABC (void); //写ABC数码管值到AT24C02  
/*-----  
STC12C54AD函数  
/*****读addx地址起的num哥数值到地址*p*****/  
extern void Read_eep_all(uint16 addx,uint8 num,uint8 *p);  
  
/*****读一个地址的数值*****/  
extern uint8 Read_eep_Byte(uint16 X);  
  
/*****写addx地址起的num哥数值到地址*p*****/  
extern void write_eep_all(uint16 addx,uint8 num,uint8 *p);  
  
/*****写一个地址的数值*****/  
extern void write_eep_Byte(uint16 X,uint8 p);  
/*-----
```

这里不要用了。

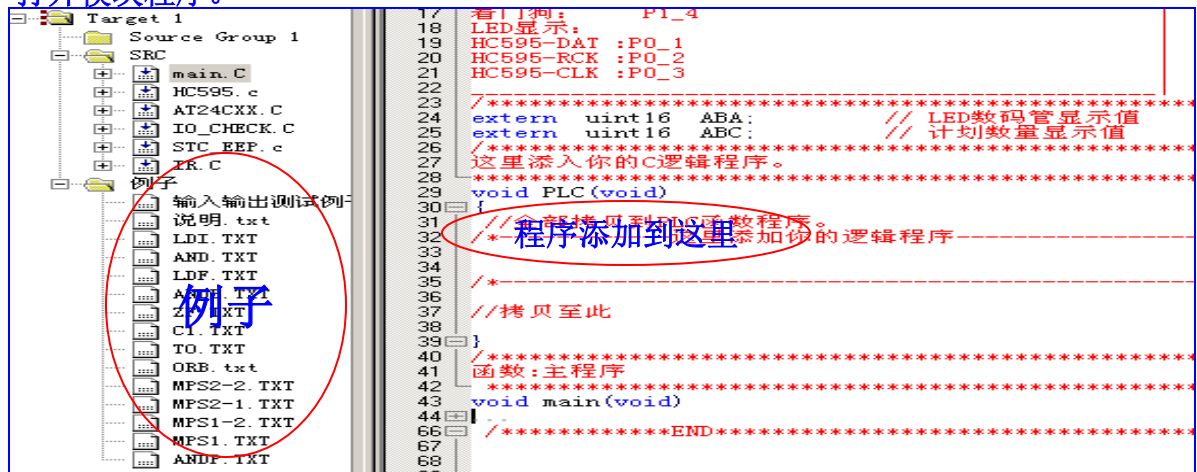
0—0F地址已经用了，勿用。

初始化端口，初始化定时器0.1,外部中断0，读存储值ABA,ABC,

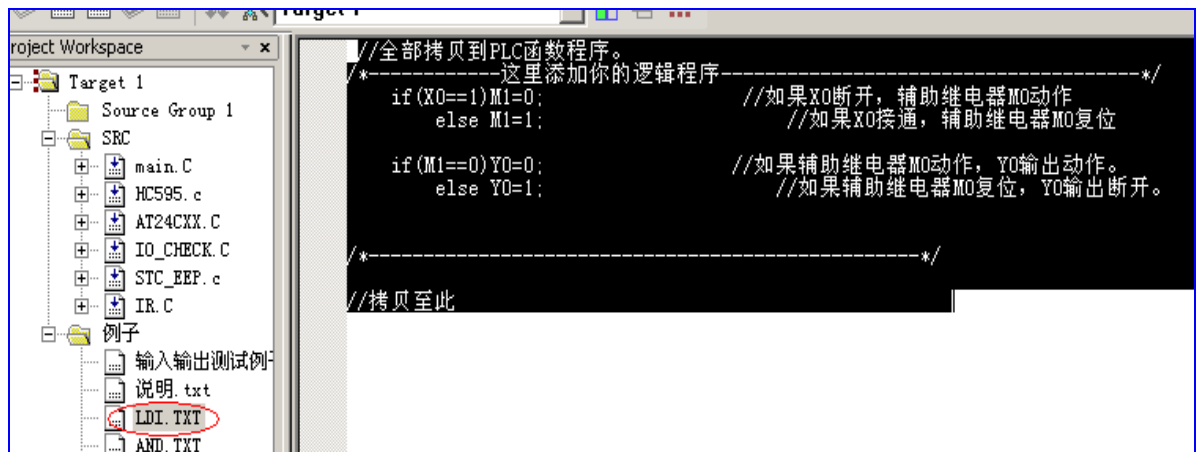
## 四：把例子添加到PLC函数编程

变量与输入输出定义根据我的板子写的，供参考。

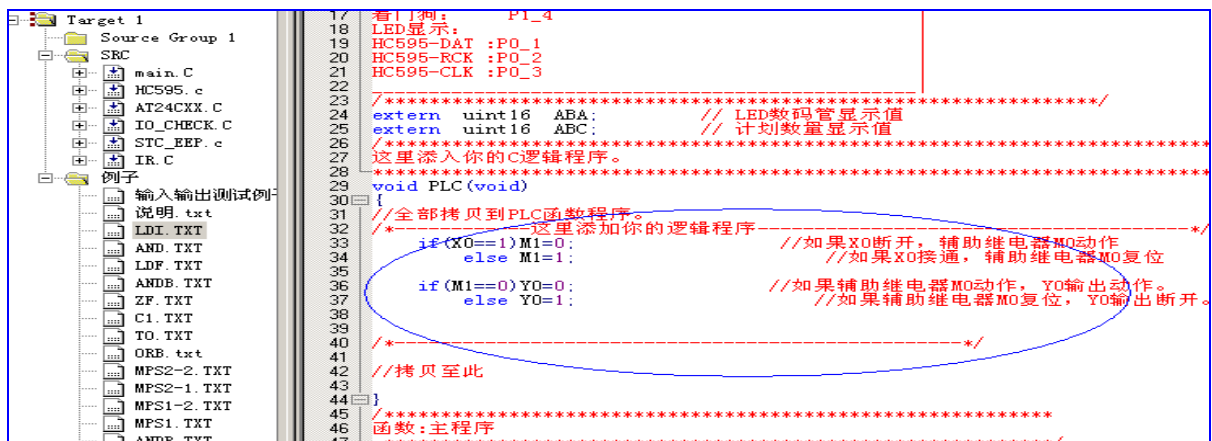
### 1：打开模块程序。



### 2：打开左边例子的 LDI.TXT,复制程序内容。

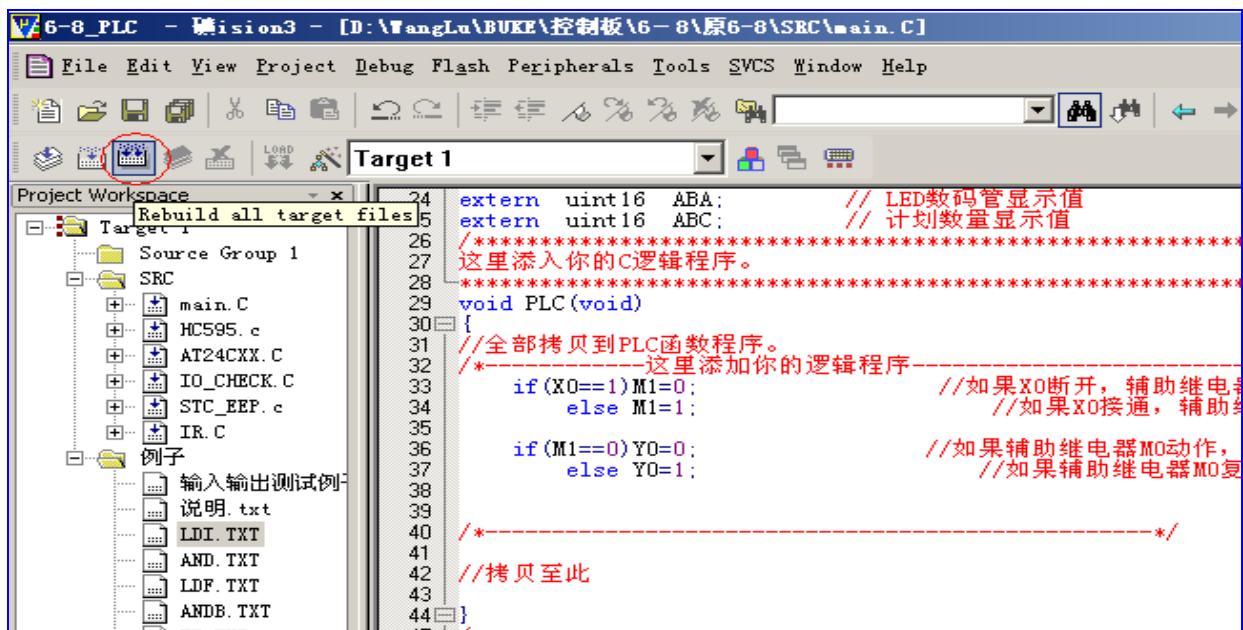


### 3：打开main.C文件，把复制程序的内容粘贴到PLC函数中。





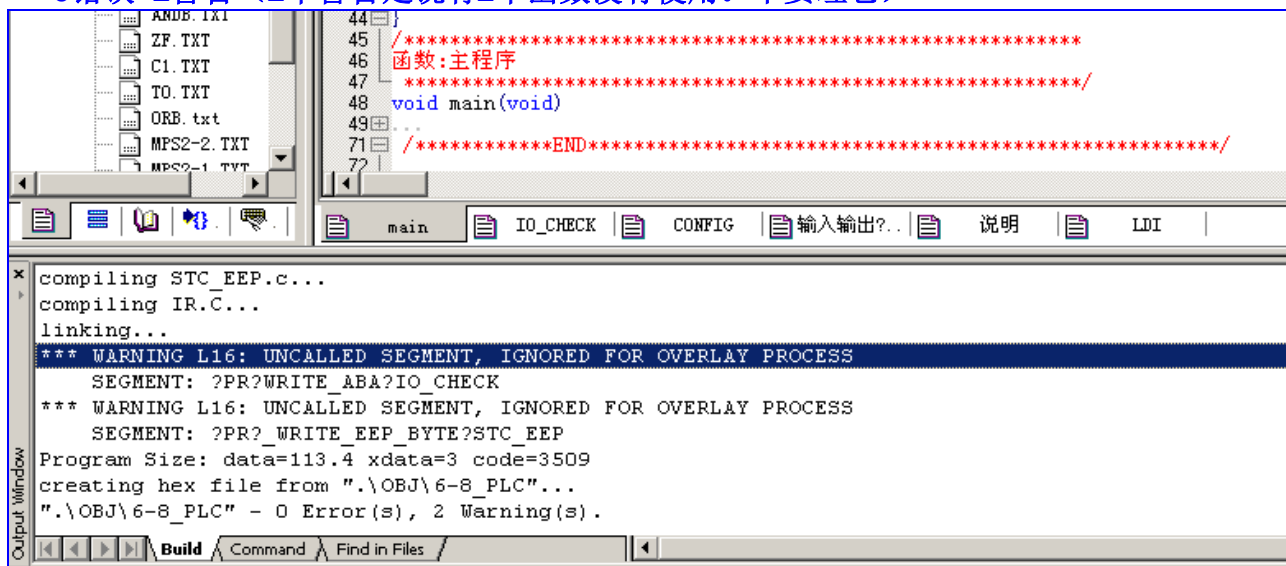
#### 4: 编译程序。点击编辑按钮。



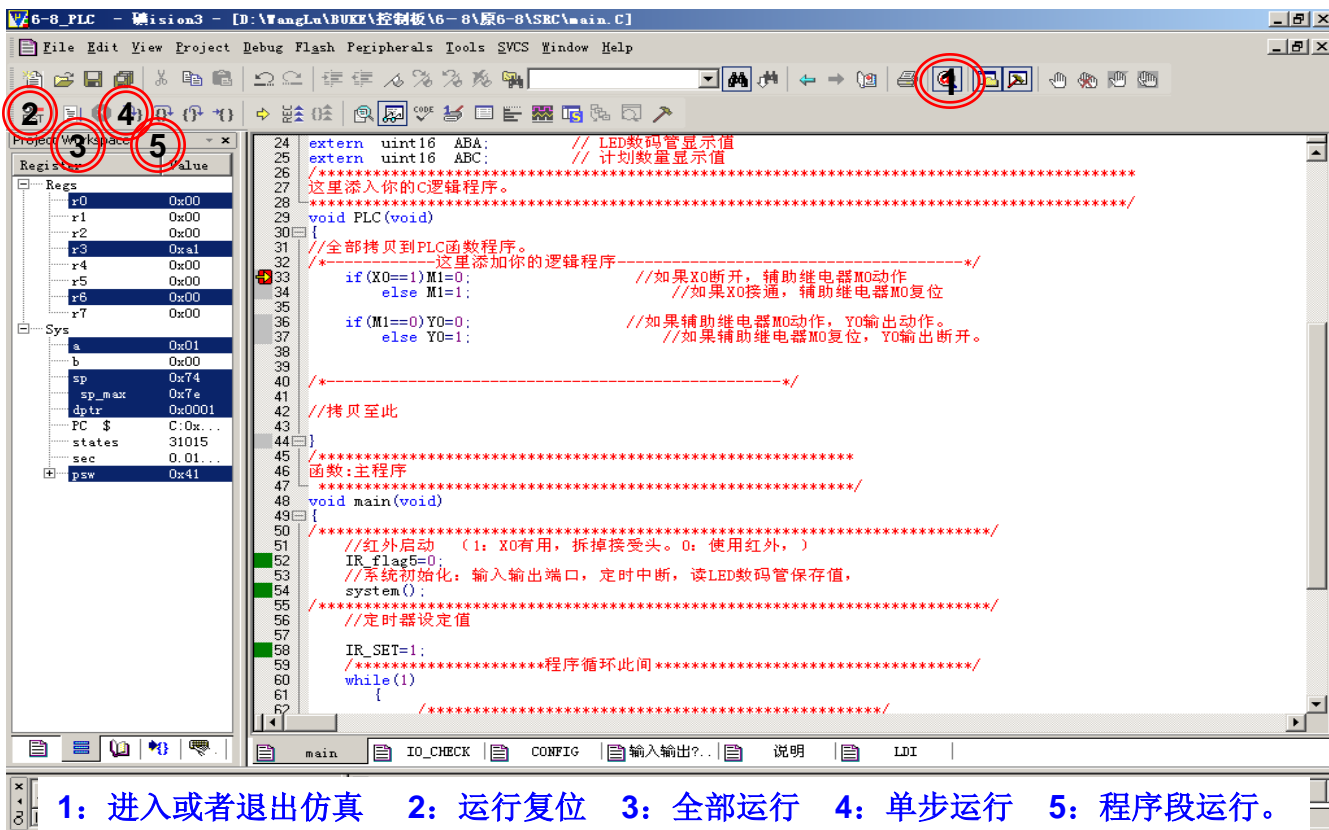
#### 5: 检查有没有错误。

消息栏显示：在工程目录的OBJ文件下生成了“6-8\_PLC.HEX”文件，（这个是要烧录到单片机的文件。）

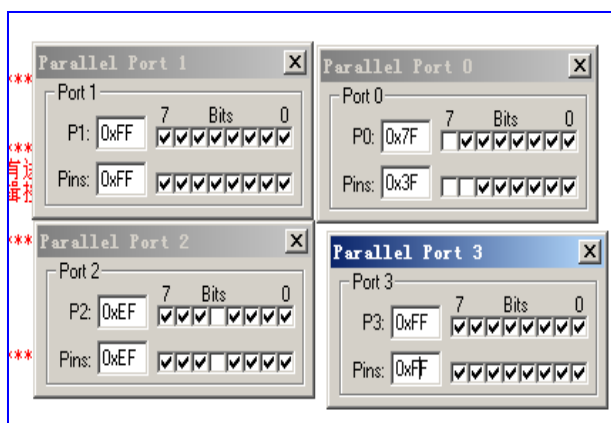
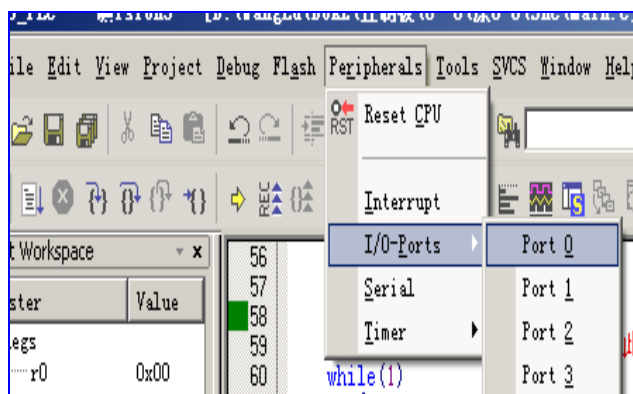
0错误 2警告（2个警告是说有2个函数没有使用。不要理它）



6: 至此，编译完成了，你也可以用KEIL C的仿真功能，模拟仿真，这个仿真没有到控制棒好用。



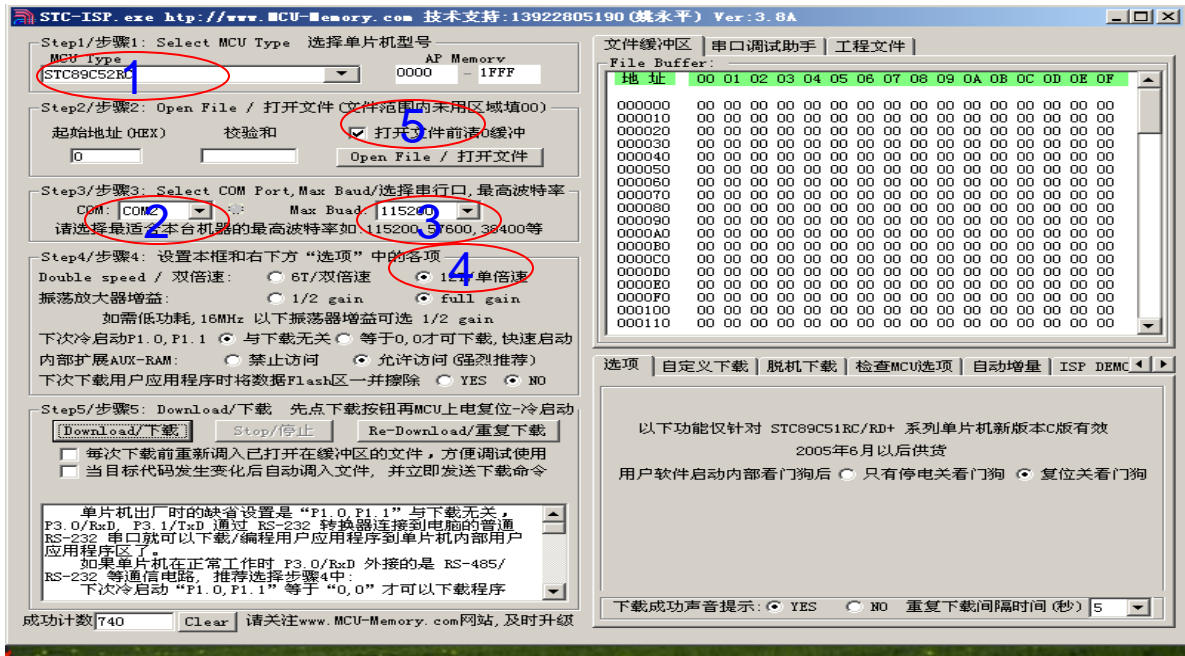
6: 你可以打开输入输出接口看端口状态。



## 五：把编译完成文件烧录到单片机

此以STC89C52单片机烧录为例

### 1：打开烧录软件软件，



1 选择单片机型号：STC89C52

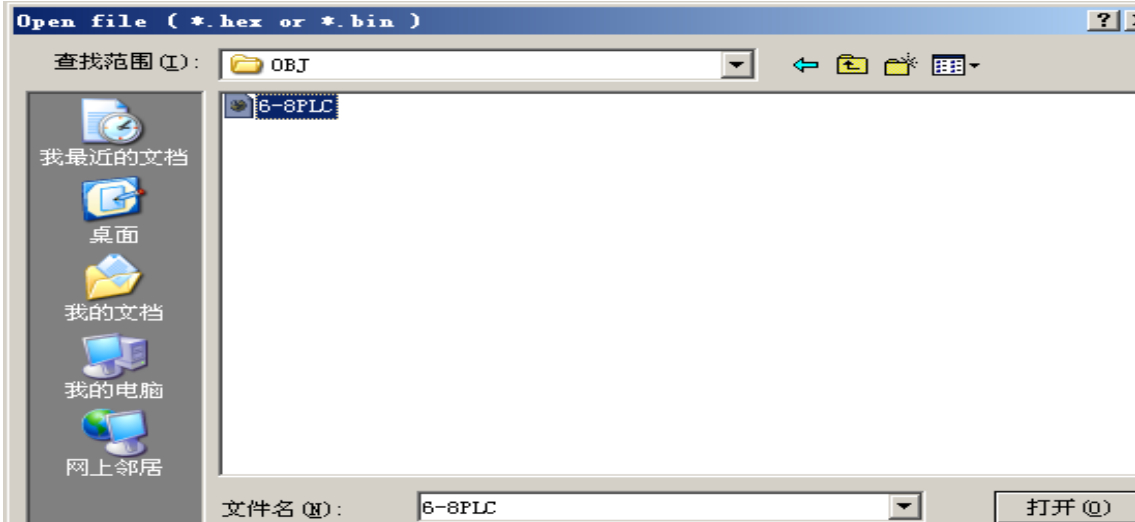
2 设定你串口号（比如COM1）

3 设定波特率：115200

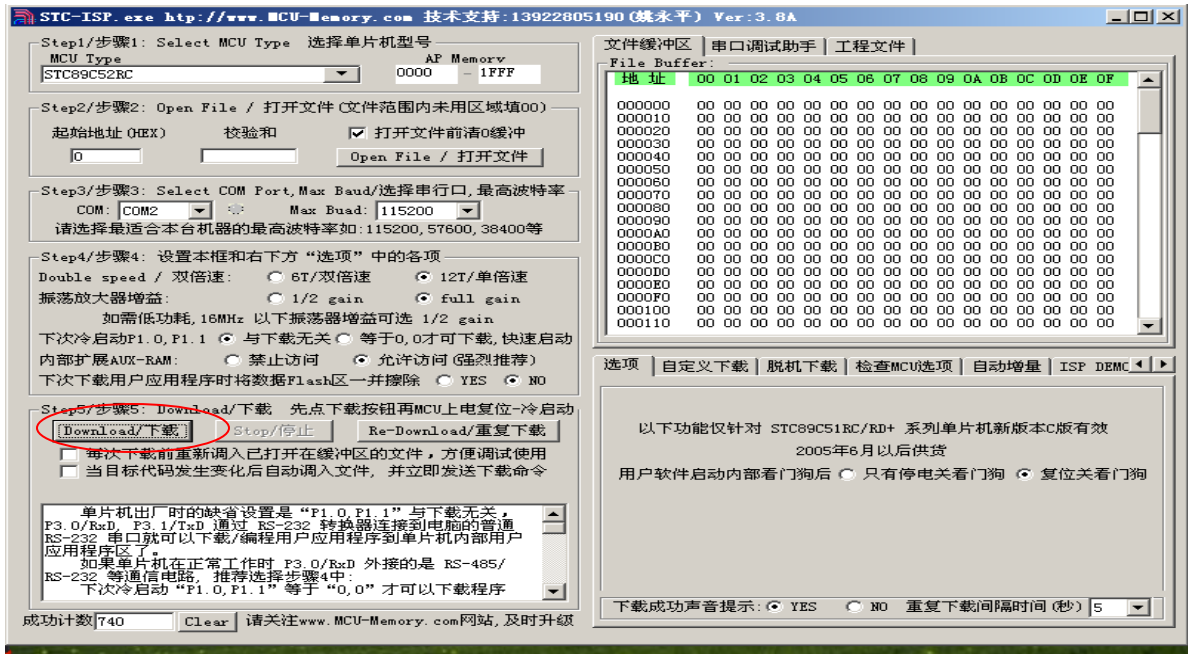
4 选择倍频：12T/单倍速

### 2：打开要烧录文件。

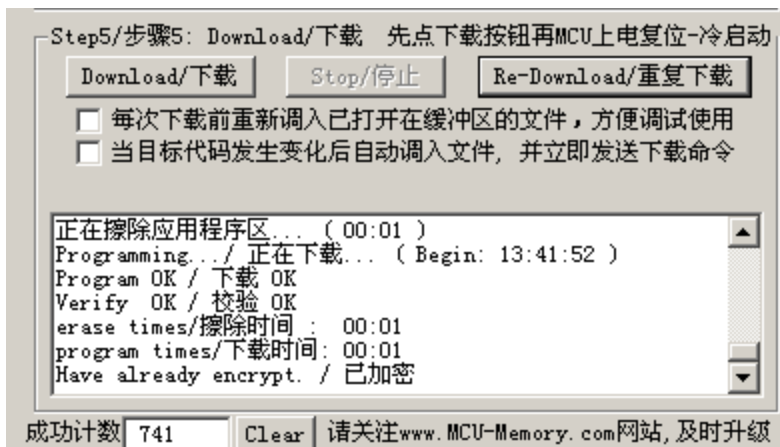
打开上图（5）按键。找到工程目录下OBJ文件里的“6-8PLC.HEX”文件并打开。



3: 把控制板的串口线接上, 先控制板别通电, 点击“下载”后, 再通控制板电。



4: 把控制板的串口线接上, 先控制板别通电, 点击“下载”后, 再通控制板电, 呵呵, 乖乖。程序下载完成了, 还加密了, 加密就是别人读不出你的程序, 放心的是程序还可以再使用。

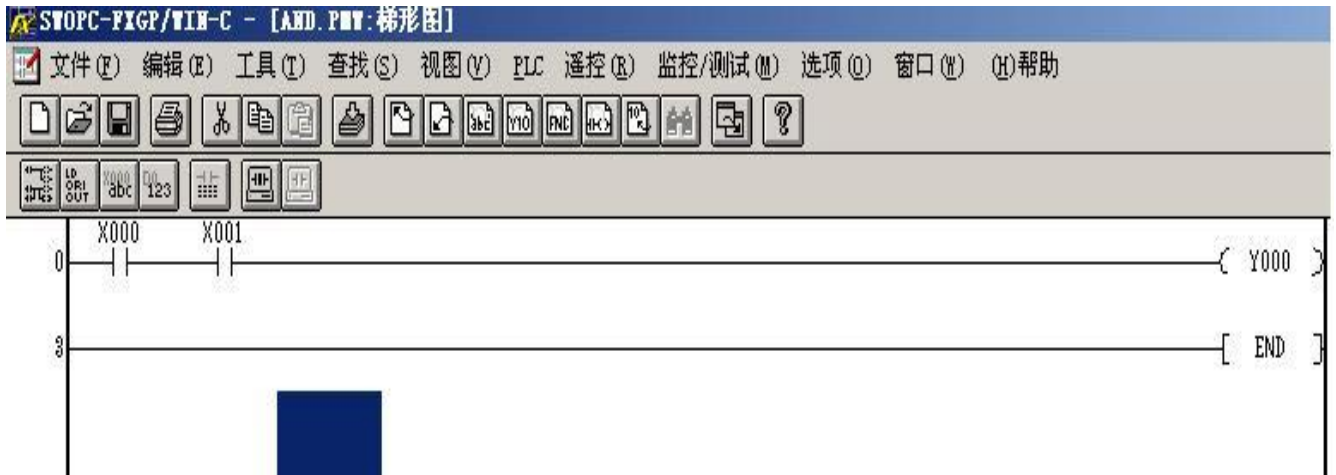


## 六：C语言实现梯形图写法

1:	AND
2:	LDI
3:	ANDB
4:	ANDP
5:	ORB
6:	C1
7:	T0
8:	MPS1
9:	LDF
10-1:	MPS2
10-2:	MPS2
11:	ZF

# 1:AND

## 梯形图

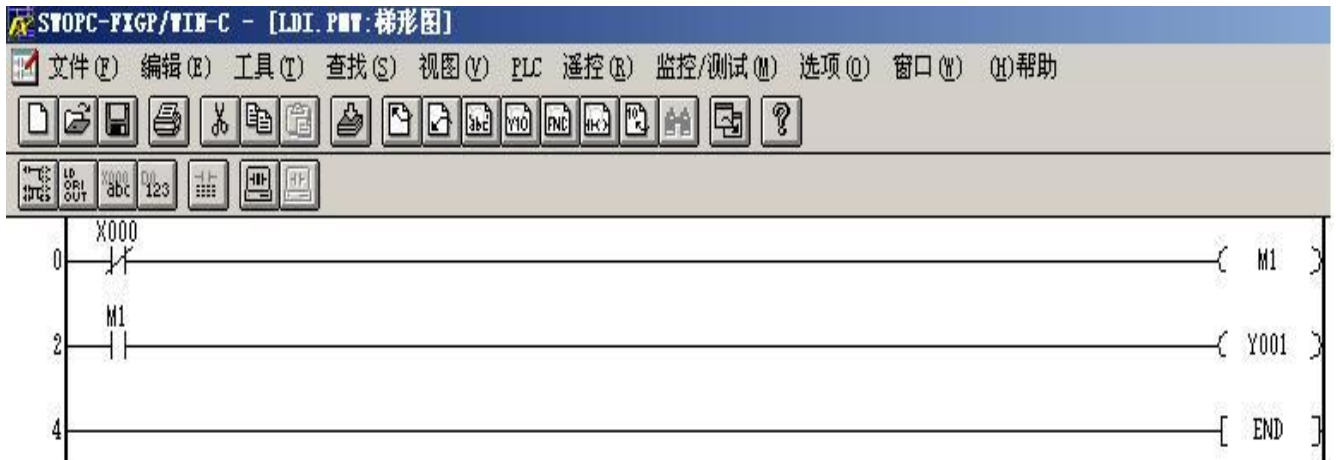


## C语言写法:

```
/******  
这里添加你的C逻辑程序。  
*****/  
void PLC(void)  
{  
    if((X0==0)&&(X1==0)) Y0=0;    //如果X0与X1同时按下条件成立，Y0动作  
    else Y0=1;                    //if条件不成立 Y0停止。  
}
```

## 2:LDI

### 梯形图

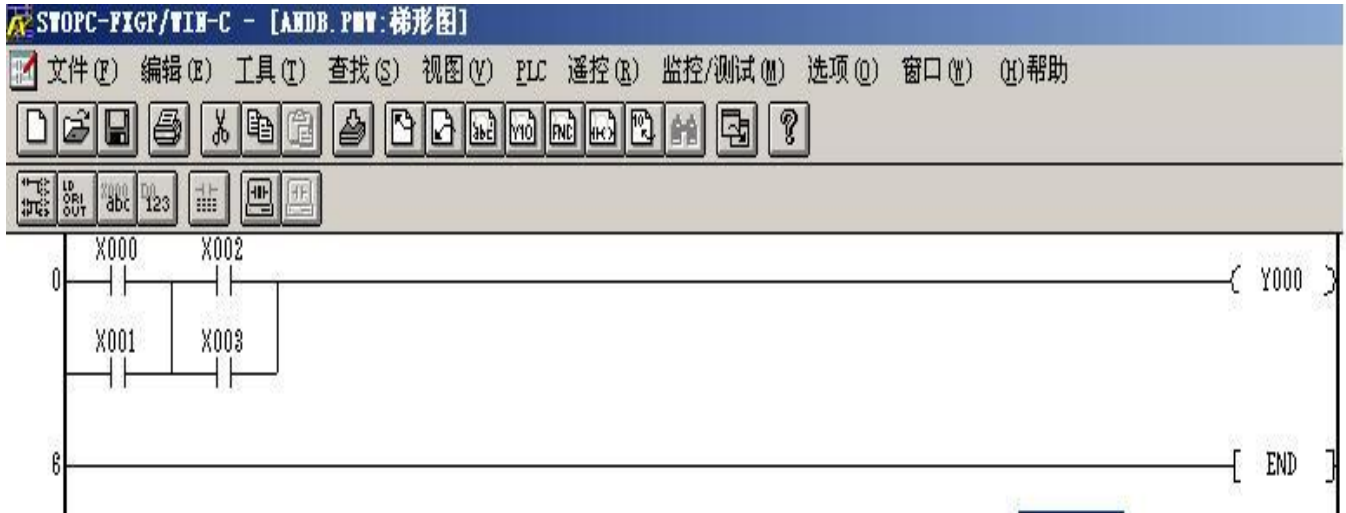


### C语言写法:

```
/*  
这里添加你的C逻辑程序。  
*/  
  
void PLC(void)  
{  
    if(X0==1)M1=0;           //如果X0断开，辅助继电器M0动作  
    else M1=1;               //如果X0接通，辅助继电器M0复位  
  
    if(M1==0)Y0=0;           //如果辅助继电器M0动作，Y0输出动作。  
    else Y0=1;               //如果辅助继电器M0复位，Y0输出断开。  
  
}
```

### 3:ANDB

#### 梯形图



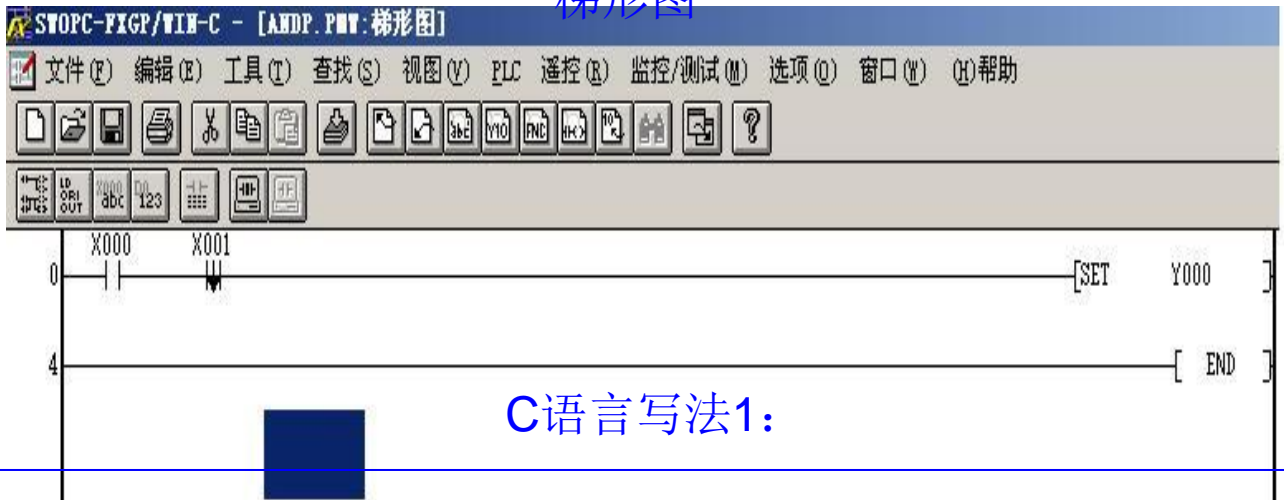
#### C语言写法:

```
/*  
这里添加你的C逻辑程序。  
*/  
void PLC(void)  
{  
    if(((X0==0)||X1==0)&&((X2==0)||X3==0))Y0=0; //如果X0或者X1任意一个接通  
                                                    //与X2与X3任意一个接通，Y0动作。  
    else Y0=1; //上面条件不成立，Y0断开。  
}
```



## 4:ANDP

### 梯形图



### C语言写法1:

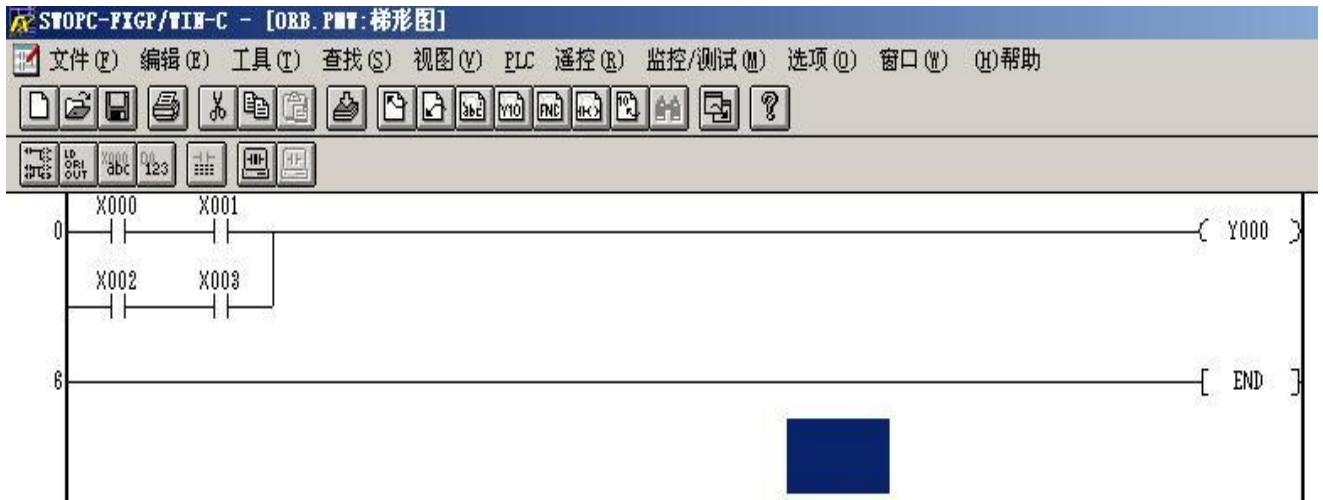
```
/*  
这里添加你的C逻辑程序。  
*/  
void PLC(void)  
{  
    if(X1==0) M0=0;           //如果X1接通，辅助继电器M0通电。表示X1接通过。  
  
    if((X0==0)&&(X1==1)&&(M0==0)) //如果X0接通，X1松开了，辅助继电器动作了。  
    {  
        M0=1;Y0=0;           //辅助继电器复位，Y0通电保持。  
    }  
}
```

### C语言写法2:

```
/*  
这里添加你的C逻辑程序。  
*/  
void PLC(void)  
{  
    if((X1==0) M0=0;           //如果X1接通，辅助继电器M0通电。表示X1接通过。  
    if((X1==1)&&( M0==0))M1=0; //如果X1断开并且M0通电过，辅助继电器M1动作  
  
    if((X0==0)&&(M1==0)){ M1=1;Y0=0;}//如果X0接通并且M1也接通，M1复位，Y0动作。  
}
```

## 5:ORB

### 梯形图

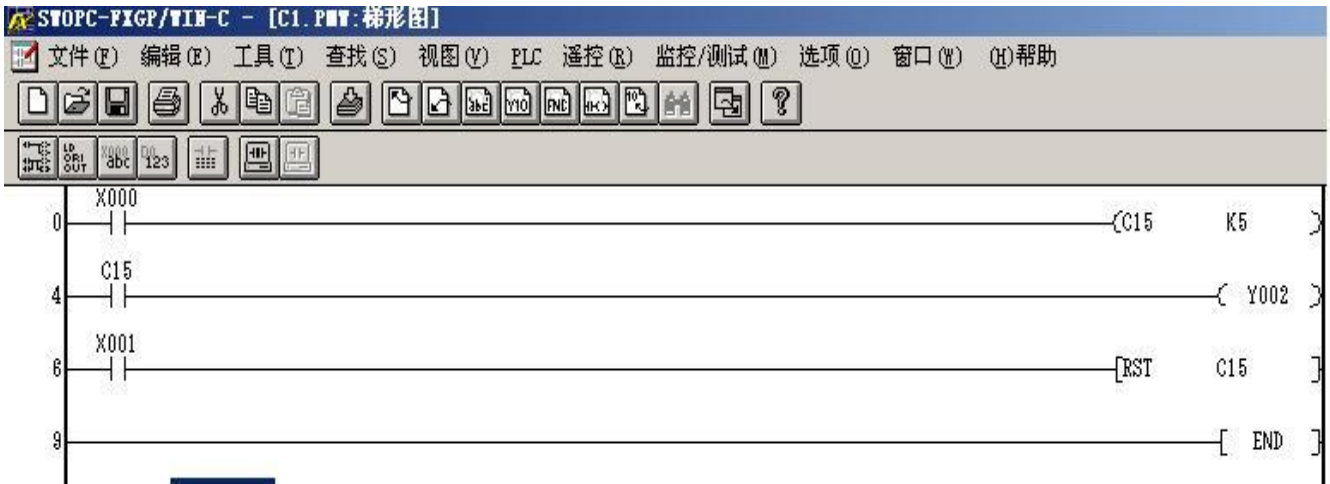


### C语言写法:

```
/*  
这里添加你的C逻辑程序。  
*/  
void PLC(void)  
{  
    if(((X0==0)&&(X1==0))||((X2==0)&&(X3==0))) Y0=0; //X0与X1同时接通，或者X2与X3  
                                                    //同时接通，Y0输出动作。  
    else Y0=1; //如果上面条件不成立，Y0断开。  
}
```

## 6:C1

### 梯形图



C语言写法(用T0代替)：

```
/******
```

这里添加你的C逻辑程序。

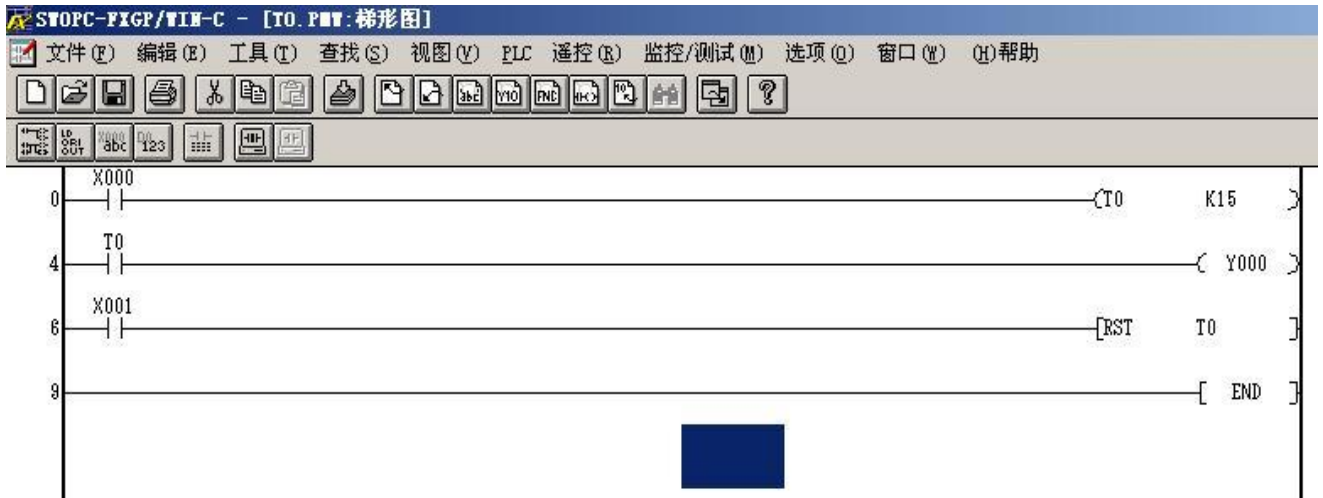
```
*****/
```

```
void PLC(void)
```

```
{  
    if((X0==0)&&(M0==1))                //如果X0接通，M0没有动作。条件成立。  
    {  
        M0=0;                            //M0动作：防止一直动作，下一动作准备。  
        if(C0_K==1){C0_K=0;C0K=5;}//如果计数器没有打开过，打开计数器，  
                                   //赋值给计数器  
        if(C0K==0)C0_bit=0;              //X0动作一次C15K减一次，如果减到5此为0，  
                                   //计数器C15_bit动作  
        else C0K--;                      //如果没有计数5次，继续计数。  
    }  
    if((X0==1)&&(M0==0))M0=1;            //如果X0松开，辅助继电器复位，下一动作准备。  
  
    if(C0_bit==0){Y2=0;}                  //如果计数器计数到了5次，Y2输出动作。  
    else Y2=1;                            //如果计数器没有动作够，Y2停止。  
    if(X1==0){C0_K=1;C0_bit=1;}          //X1接通，计数开关与标志为复位。  
}
```

## 7:T0

### 梯形图

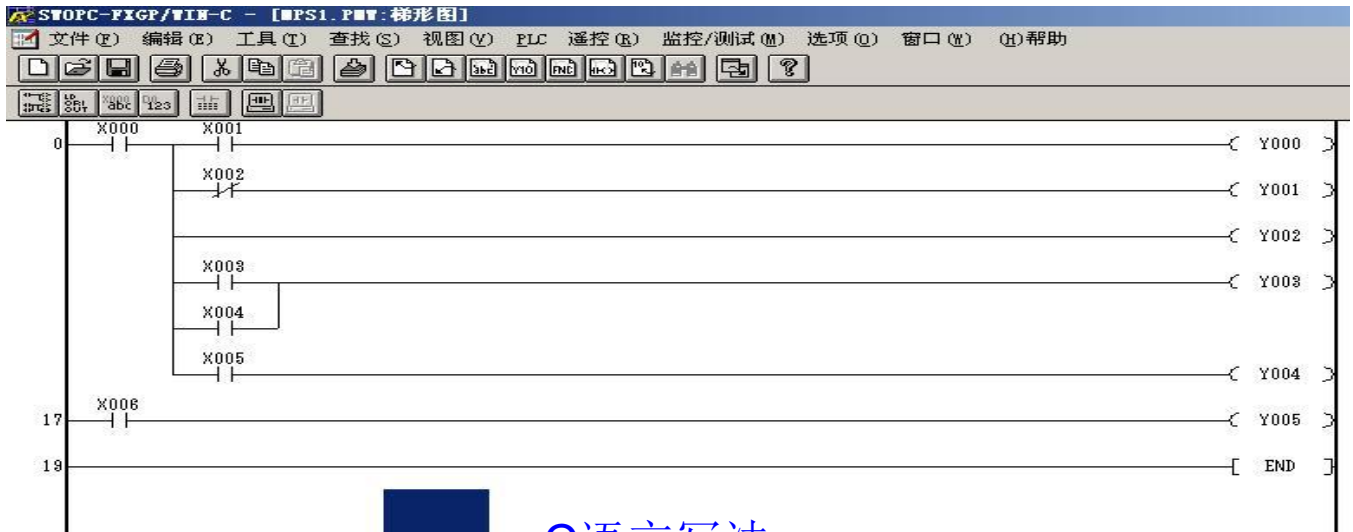


### C语言写法:

```
/*
这里添加你的C逻辑程序。
*/
void PLC(void)
{
    if(X0==0)                //如果X0接通
    {
        if(T0_K==1)         //如果定时没有打开。
        {
            T0_K =0;        //开定时器
            T0K=15;          //定时器值1.5秒
        }
    }
    if(T0_bit==0)Y0=0;        //1.5秒定时到了, Y0=0;
    else Y0=1;
    if(X1==0) T0_bit=1;       //复位动作开关
}
```

## 8:MPS1

### 梯形图



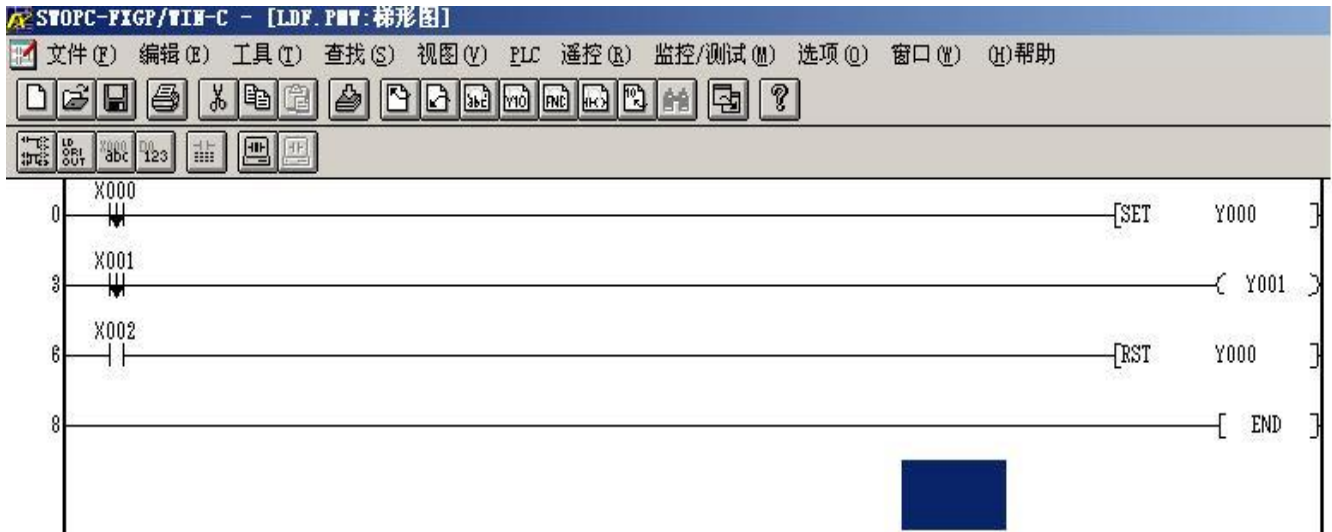
### C语言写法:

```
/*  
这里添入你的C逻辑程序。 */  
void PLC(void)  
{  
  
    if((X0==0)&&(X1==0)) Y0=0;  
    else Y0=1;  
    if((X0==0)&&(X2==1)) Y1=0;  
    else Y1=1;  
    if(X0==0)Y2=0;  
    else Y2=1;  
    if((X0==0)&&((X3==0)||((X4==0)))) Y3=0;  
    else Y3=1;  
    if((X0==0)&&(X5==0)) Y4=0;  
    else Y4=1;  
    if(X6==0)Y5=0;  
    else Y5=1;  
}
```

```
/*  
这里添入你的C逻辑程序。 */  
void PLC(void)  
{  
    if(X0==0)  
    {  
        if(X1==0) Y0=0;  
        else Y0=1;  
        if(X2==1) Y1=0;  
        else Y1=1;  
        Y2=0 ;  
        if((X3==0)||((X4==0))) Y3=0;  
        else Y3=1;  
        if(X5==0) Y4=0;  
        else Y4=1;  
    }  
    else {Y0=1;Y1=1;Y2=1;Y3=1;Y4=1;}  
    if(X6==0)Y5=0;  
    else Y5=1;  
}
```

## 9:LDF

### 梯形图

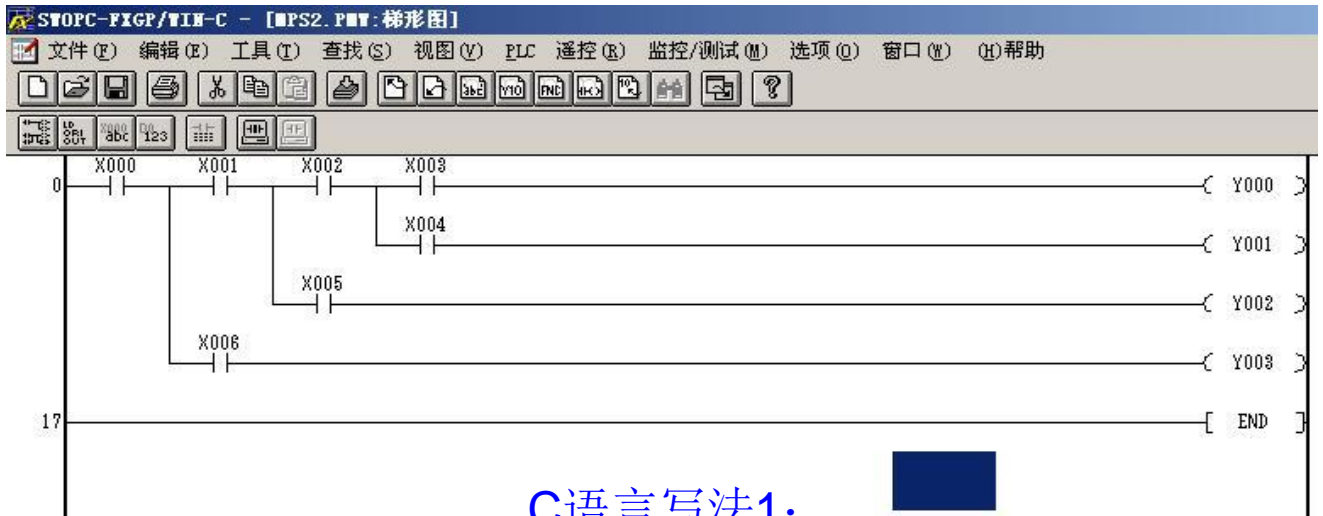


### C语言写法:

```
/*  
*****  
这里添加你的C逻辑程序。  */  
void PLC(void)  
{  
    if(X0==0) M0=0;                //X0接通，M0接通保持。  
    if((X0==1)&&(M0==0)) {M0=1;Y0=0;} //X0松开并且M0表示X0动作过。M0复位，Y0动作  
保持  
  
    if(X1==0) M1=0;                //X1接通，M1接通保持  
    if((X1==1)&&(M1==0)) {M1=1;Y1=0;} //X1松开了，M1是接通过，M1复位，Y1输出动作。  
    else Y1=1;                    //如果上面不成立，Y1断开。  
  
    if(X2==0)Y0=1;                //如果X2接通，Y0复位断开。  
}
```

## 10-1:MPS2

### 梯形图

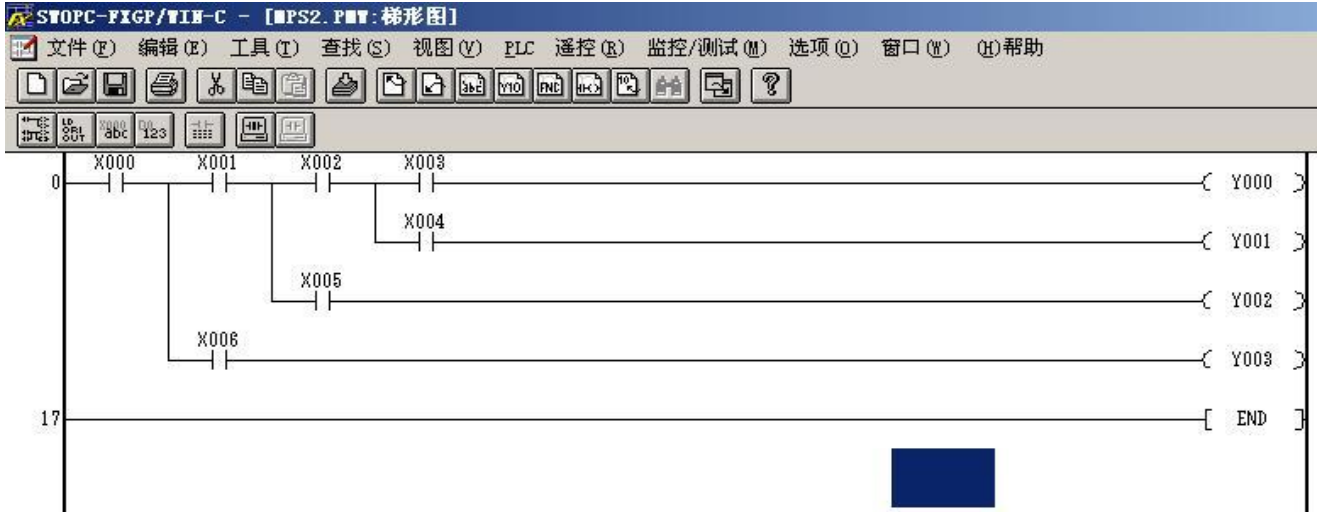


### C语言写法1:

```
/*  
这里插入你的C逻辑程序。  
*/  
void PLC(void)  
{  
    if(X0==0)                //X0接通了就执行下面程序。  
    {  
        if(X6==0) Y3=0;      //X6接通, Y3动作  
        else Y3=1;          //X6断开, Y3断开  
        if(X1==0)            //X0接通就执行下面动作。  
        {  
            if(X5==0)Y2=0;    //X5接通Y2动作  
            else Y2=1;        //X5断开Y2断开  
            if(X2==0)         //X2接通执行下面动作  
            {  
                if(X4==0)Y1=0; //X4接通,Y1动作。  
                else Y1=1;     //X4断开, Y1断开  
                if(X3==0)Y0=0; //X3接通, Y0动作  
                else Y0=1;     //X3断开, Y0断开  
            }else {Y0=1;Y1=1;} //---X2断开, Y0,Y1断开  
        }else {Y0=1;Y1=1;Y2=1;} //---X1断开, Y0,Y1,Y2断开  
    }else {Y0=1;Y1=1;Y2=1;Y3=1;} //---X0断开, Y0,Y1,Y2 Y3断开  
}
```

## 10-2:MPS2

### 梯形图



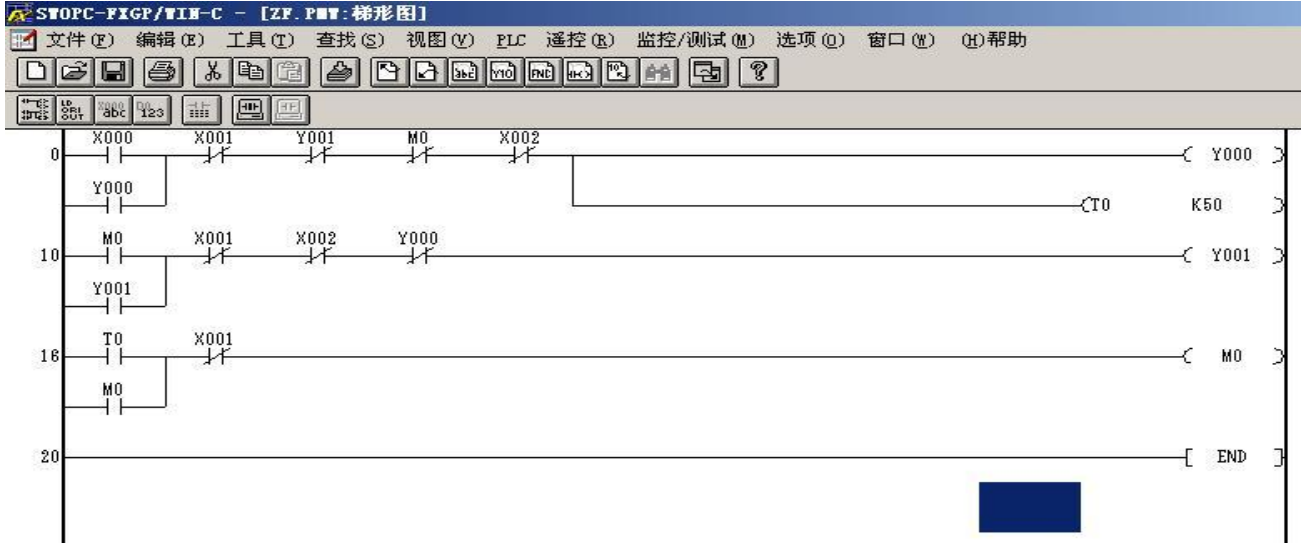
### C语言写法2:

```
/*  
这里添加你的C逻辑程序。  
*/  
void PLC(void)  
{  
    if((X0==0)&&(X1==0)&&(X2==0)&&(X3==0)) Y0=0;  
    else Y0=1;  
    if((X0==0)&&(X1==0)&&(X2==0)&&(X4==0)) Y1=0;  
    else Y1=1;  
    if((X0==0)&&(X1==0)&&(X5==0)) Y2=0;  
    else Y2=1;  
    if((X0==0)&&(X6==0)) Y3=0;  
    else Y3=1;  
}
```



# 11:ZF

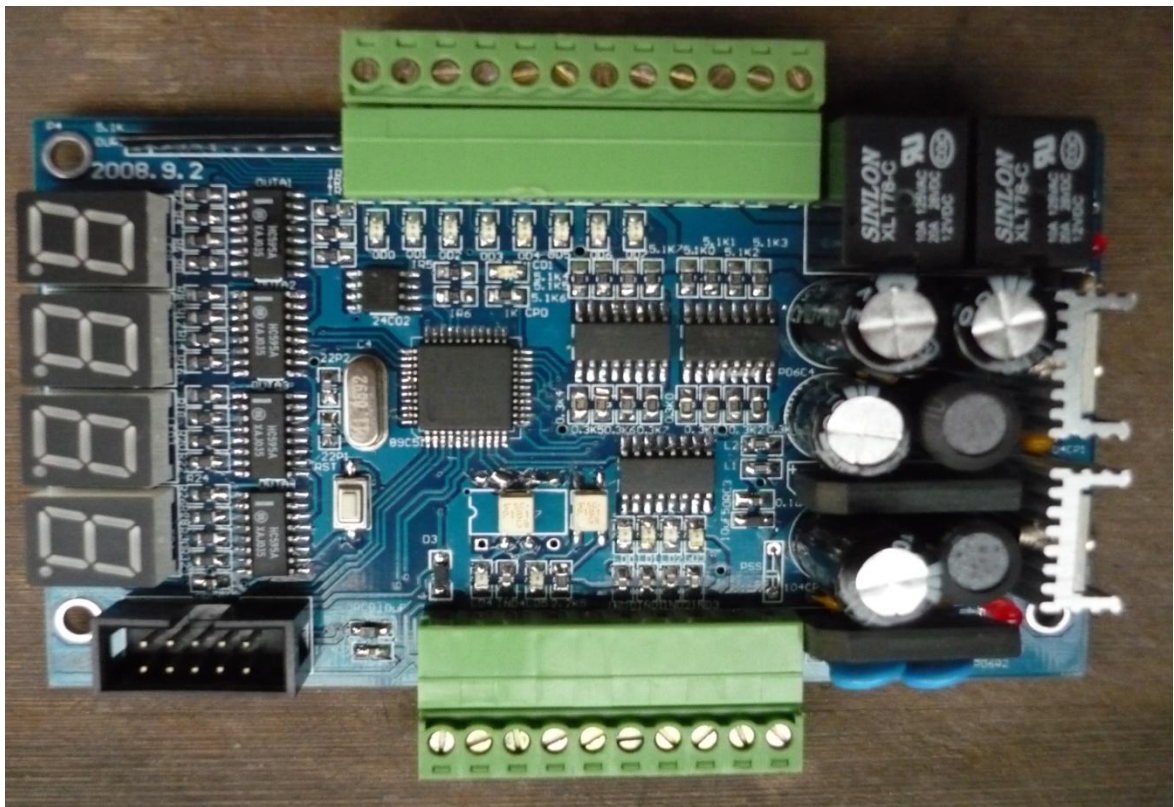
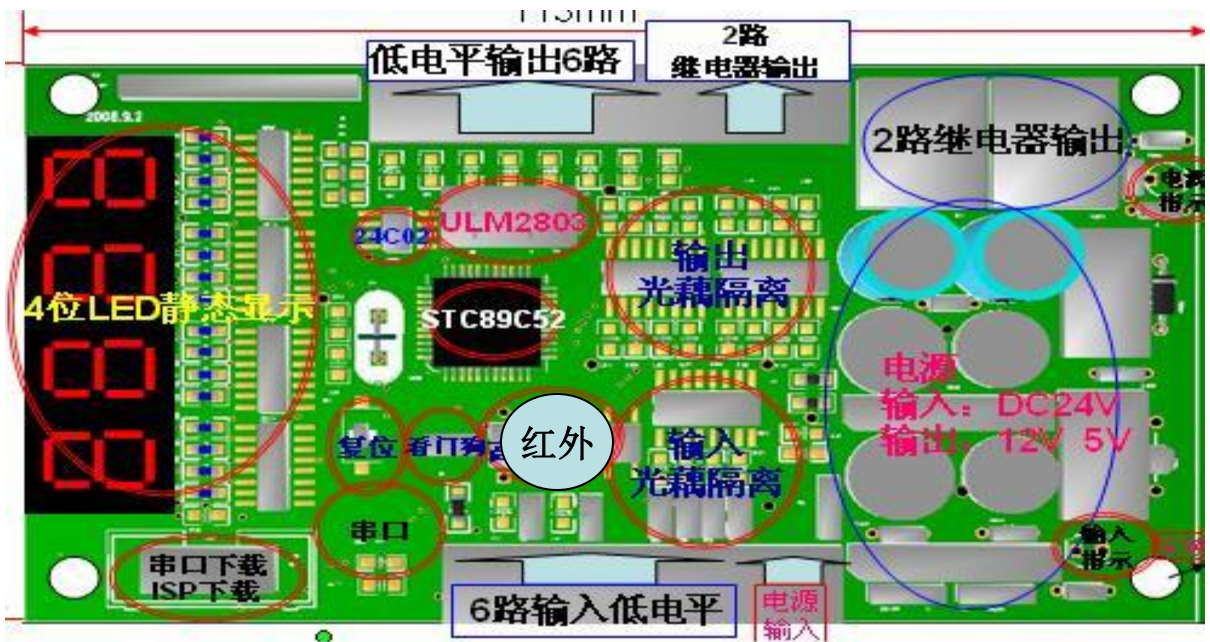
## 梯形图



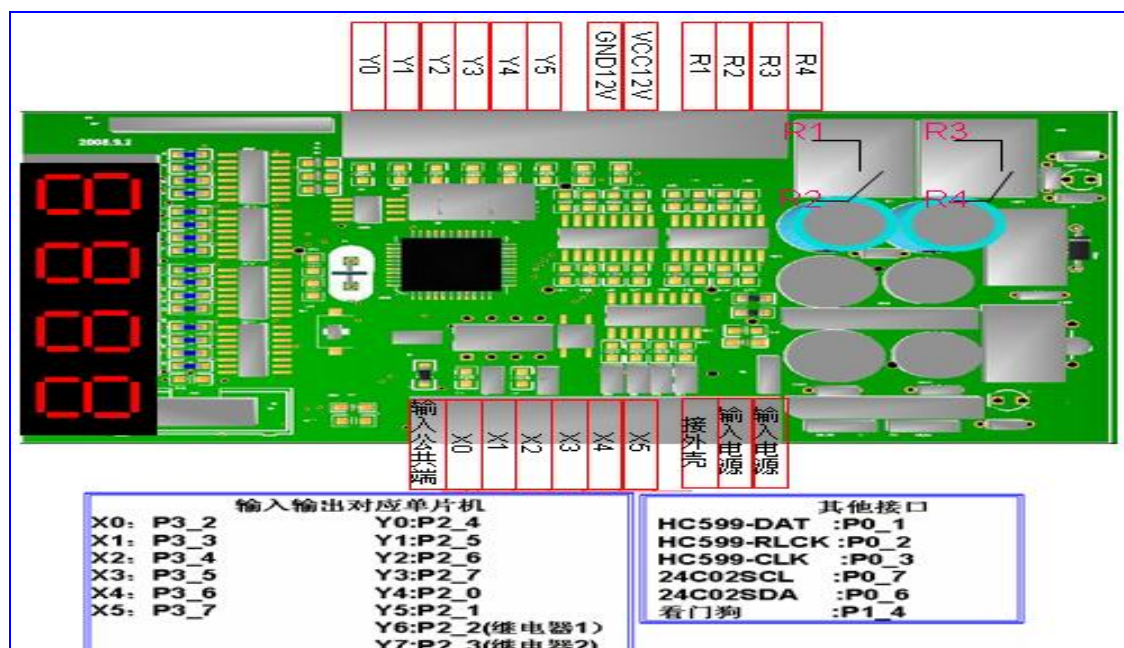
## C语言写法:

```
/*  
这里添加你的C逻辑程序。  
*/  
void PLC(void)  
{  
    if(((X0==0)||Y0==0)&&(X1==1)&&(Y1==1)&&(M0==1)&&(X2==1))  
        //X0 Y1任意一个接通， +X1断开 + Y0断开 + M0断开+ X2断开条件成立，执行  
        {  
            Y0=0;                //Y0动作  
            if(T0_K==1)          //如果定时器没有打开多，执行  
            {  
                T0_K=0;          //定时器打开  
                T0K=50;          //定时器值=50×0.1=5秒  
            }  
        }  
    else {Y0=1; T0_K=1;T0_bit=1;}//复位  
    if(((M0==0)||Y1==0)&&(X1==1)&&(X2==1)&&(Y0==1))Y1=0;  
    //如果M0或者Y1任意一个动作+X1断开+X2断开+Y0断开，条件成立.Y0动作。  
    else Y0=1;                //条件不成立，Y0断开。  
    if(((T0_bit==0)||M0==0)&&(X1==1)){M0=0;T0_bit=1;}//定时器或者M0动作，  
    //并且X1断开,M0动作。定时器复位  
    else M0=1; // if条件不成立M0复位。  
}
```

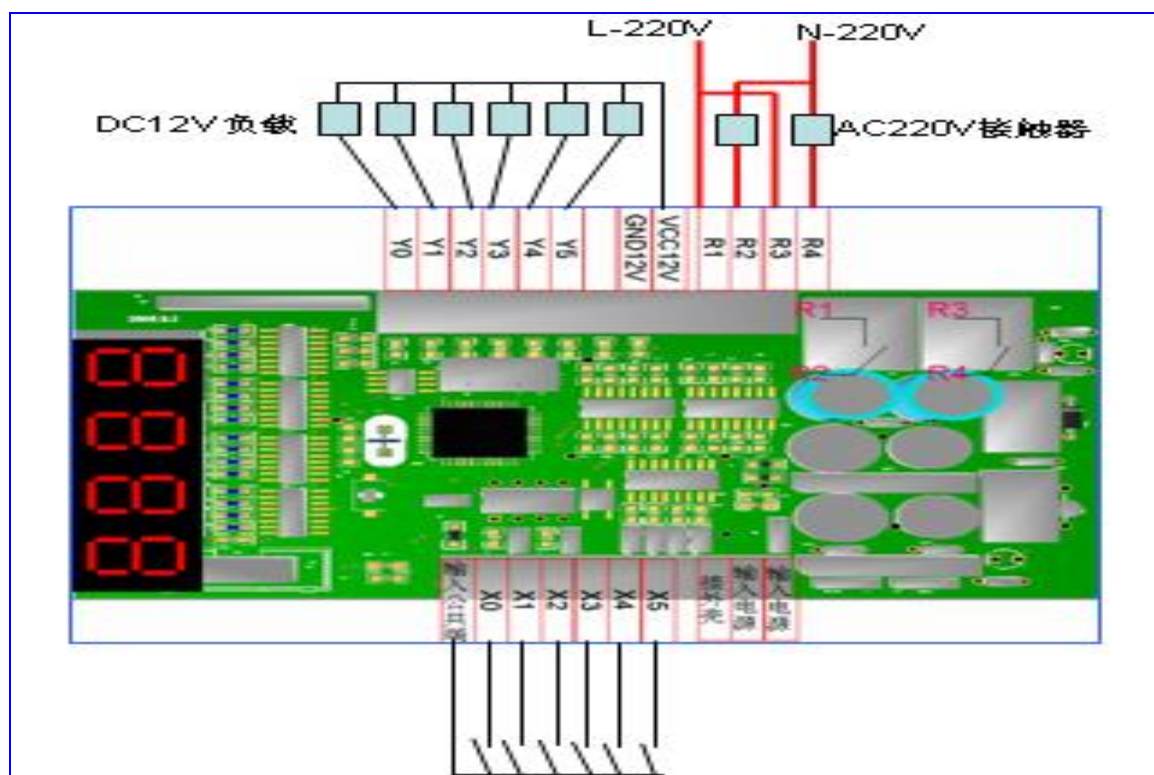
## 七：6—8控制板资源。



## 输入输出。



## 参考接线





硬件资源:

**1:CPU: STC89C52I或者AT89S52**

**2:存储(AT24C02),串口(MAX232), 外部看门狗(MAX813L).**

**3:显示: 4个74HC595驱动4位共阳数码管静态显示。**

**4:输入: 1路高速光藕隔离输入, 可当普通输入, 5路普通光藕隔离输入。**

**5:输出: 8路光藕隔离后驱动ULM2803,其中6路直接输出, 2路驱动继电器输出。**

**6:输入输出LED显示。**

**7:下载口: 保留AT89S单片机的ISP下载与STC89C单片机的串口下载。**

其他:

**1:板子尺寸: 113mm\*65mm.**

**2:安装方式: 螺丝孔固定4— $\phi$  3.5。**

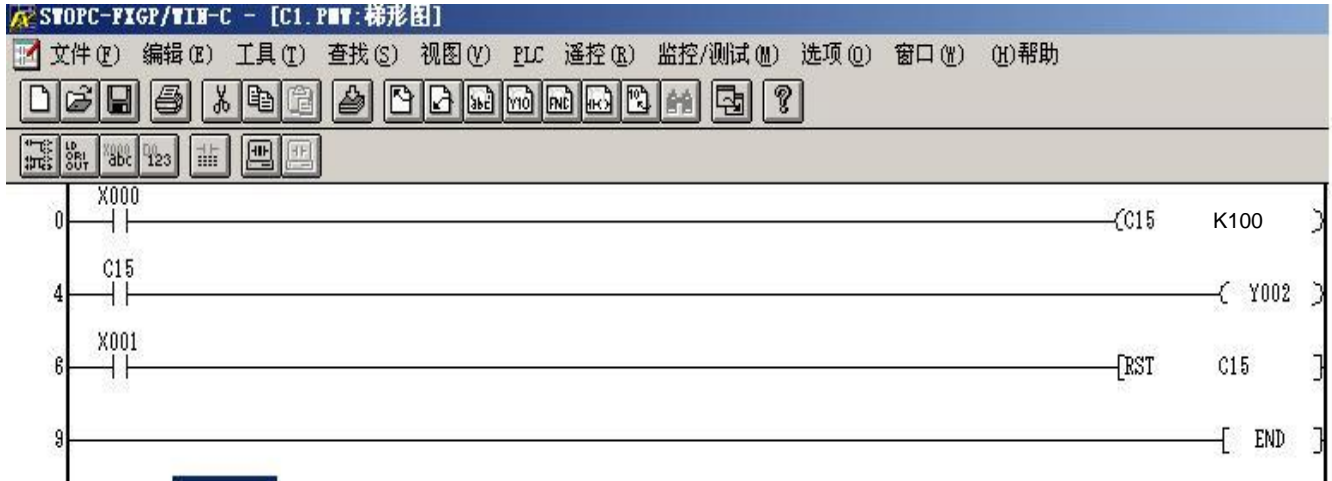
**3:输入输出用3.81间距的快速拔插端子。**

**4:输入工作电压: AC16V或者DC24V.**

## 八：6—8控制板简单使用例子。

### 1：带显示的倒计时器。

程序参考（6：C0指令）的写法。



控制要求（6-8控制模块只有C0-C7计数器，用C0代替）：

X0为加数开关，每动作一次，C0加一次。加到设定值，Y2动作。X1为计数器复位开关。LED数码管显示C的值。

Void PLC(void)

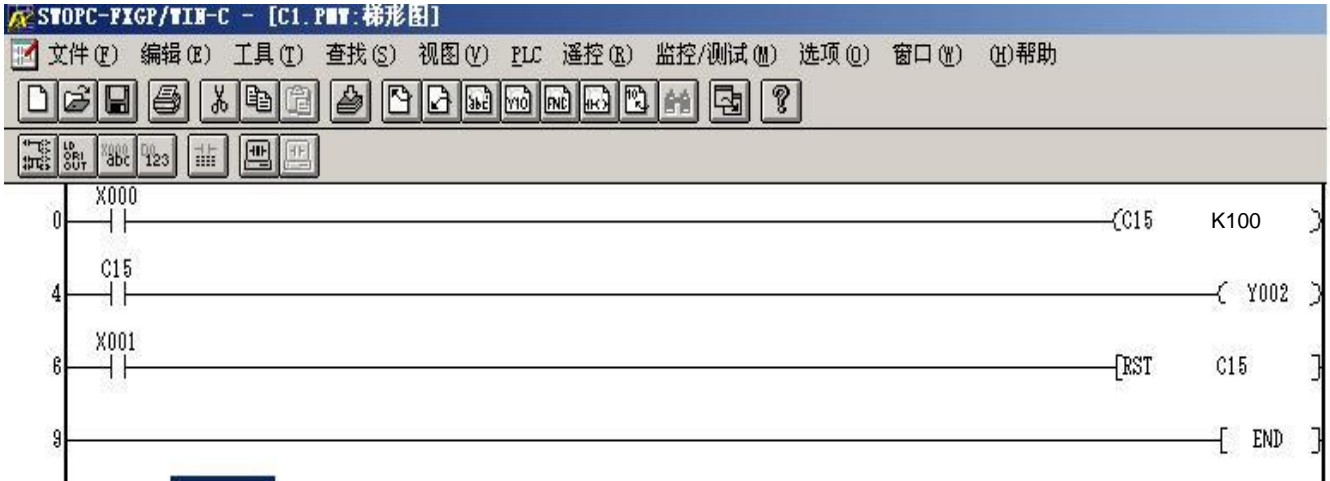
{

```
    if((X0==0)&&(M0==1))                //如果X0接通，M0没有动作。条件成立。
    {
        M0=0;                            //M0动作：防止一直动作，下一动作准备。
        if(C0_K==1){C0_K=0;C0K=100;}    //如果计数器没有打开过，打开计数器，
                                           //赋值给计数器
        if(C0K==0)C0_bit=0;              //X0动作一次C15K减一次，如果减到5此为0，
                                           //计数器C15_bit动作
        else C0K--;                       //如果没有计数5次，继续计数。
        ABA=C0K;                          //把C0K值送到LED显示。
    }
    if((X0==1)&&(M0==0))M0=1;            //如果X0松开，辅助继电器复位，下一动作准备。

    if(C0_bit==0){Y2=0;}                  //如果计数器计数到了100次，Y2输出动作。
    else Y2=1;                            //如果计数器没有动作够，Y2停止。
    if(X1==0){C0_K=1;C0_bit=1;}          //X1接通，计数开关与标志为复位。
```

## 2: 带显示的顺计数器。

程序参考（6: C0指令）的写法。



控制要求（6-8控制模块只有C0-C7计数器，用C0代替）：

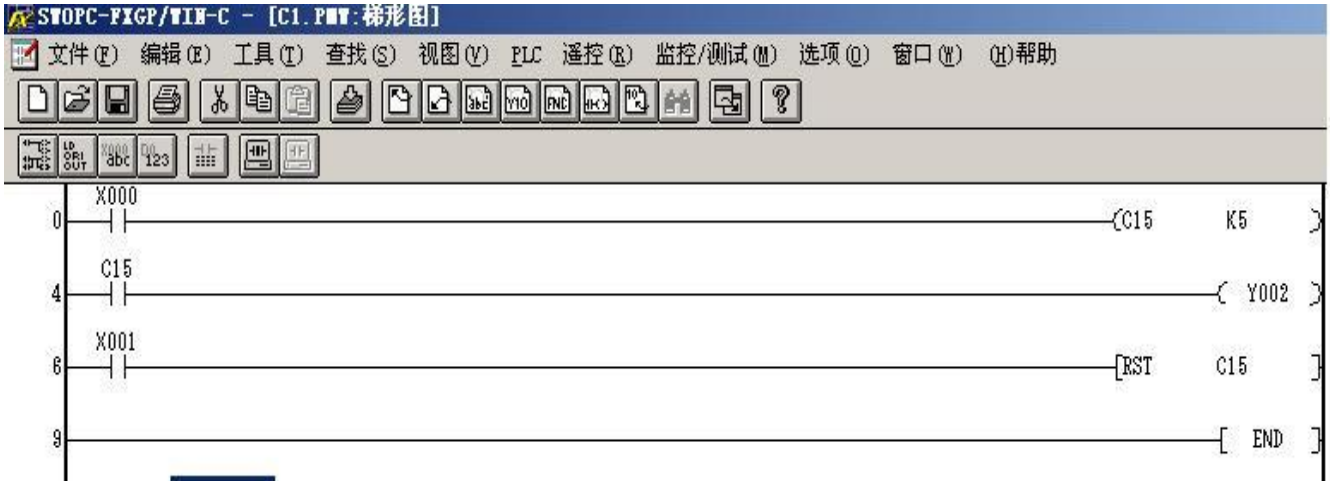
X0为加数开关，每动作一次,C0加一次。加到设定值，Y2动作。X1为计数器复位开关。LED数码管显示增长的值。

Void PLC(void)

```
{  
  
    if((X0==0)&&(M0==1))                //如果X0接通，M0没有动作。条件成立。  
    {  
        M0=0;                            //M0动作：防止一直动作，下一动作准备。  
        if(C0_K==1){C0_K=0;C0K=ABA=100;} //如果计数器没有打开过，打开计数器，  
                                           //赋值给计数器  
        if(ABA==C0K)C0_bit=0;            //X0动作一次C15K减一次，如果减到5此为0，  
                                           //计数器C15_bit动作  
        else ABA++;                      //如果没有计数100次，继续计数。  
    }  
    if((X0==1)&&(M0==0))M0=1;            //如果X0松开，辅助继电器复位，下一动作准备。  
  
    if(C0_bit==0){Y2=0;}                 //如果计数器计数到了100次，Y2输出动作。  
    else Y2=1;                           //如果计数器没有动作够，Y2停止。  
    if(X1==0){C0_K=0;ABA=0;C0_bit=1;}    //X1接通，计数开关与标志为复位。  
}
```

### 3: 带显示可以断电保存的顺计数器。

程序参考（6: C0指令）的写法。



控制要求（6-8控制模块只有C0-C7计数器，用C0代替）：

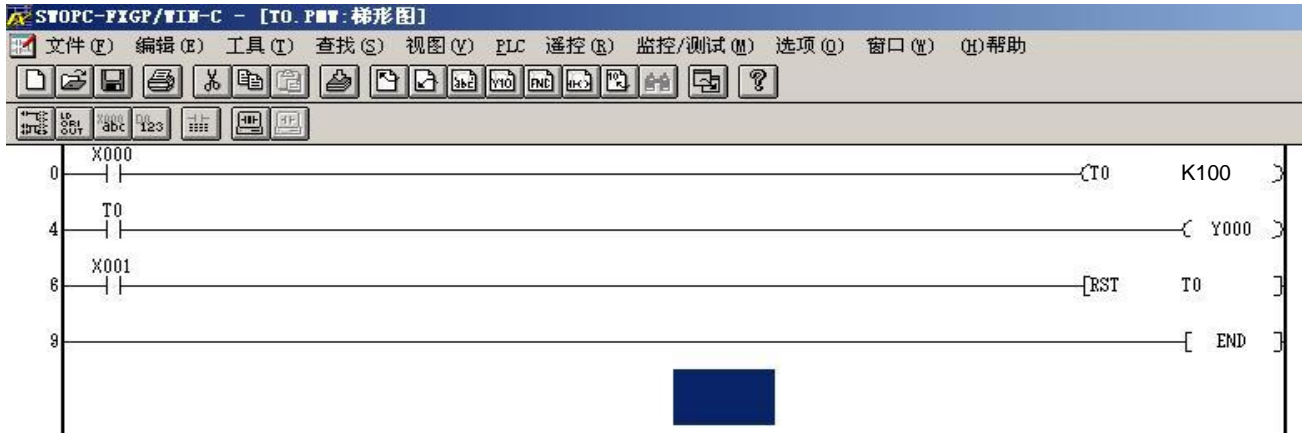
X0为加数开关，每动作一次，C0加一次。加到设定值，Y2动作。X1为计数器复位开关。LED数码管显示增长的值，每计数一次，存储计数值，断电数据不丢失。

Void PLC(void)

```
{  
  
    if((X0==0)&&(M0==1))                //如果X0接通，M0没有动作。条件成立。  
    {  
        M0=0;                            //M0动作：防止一直动作，下一动作准备。  
        if(C0_K==1){C0_K=0;C0K=ABA=100;} //如果计数器没有打开过，打开计数器，  
                                           //赋值给计数器  
        if(ABA==C0K)C0_bit=0;            //X0动作一次C15K减一次，如果减到5此为0，  
                                           //计数器C15_bit动作  
        else ABA++;                      //如果没有计数100次，继续计数。  
        write_ABA ();                    //计数值保存  
    }  
    if((X0==1)&&(M0==0))M0=1;            //如果X0松开，辅助继电器复位，下一动作准备。  
  
    if(C0_bit==0){Y2=0;}                  //如果计数器计数到了100次，Y2输出动作。  
    else Y2=1;                            //如果计数器没有动作够，Y2停止。  
    if(X1==0){C0_K=0;ABA=0;C0_bit=1;}    //X1接通，计数开关与标志为复位。  
}
```

## 4: 倒计时器

程序参考（7：T0指令）的写法。



控制要求：

X0为定时器开启信号，启动定时器后延时。10秒钟后Y0动作。LED数码管显示倒计时值，

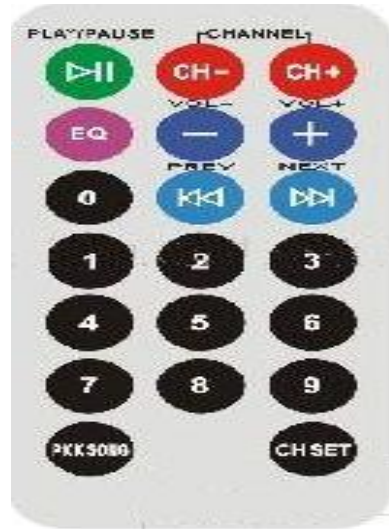
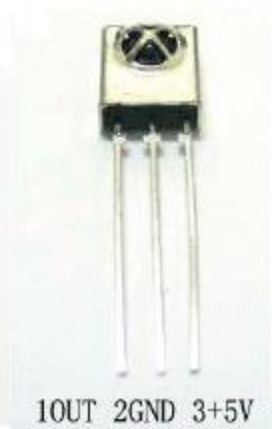
```
Void PLC(void)
{
    if(X0==0)                //如果X0接通
    {
        if(T0_K==1)         //如果定时没有打开。
        {
            T0_K =0;        //开定时器
            T0K=100;        //定时器值1.5秒
        }
        ABA=T0K;
    }
    if(T0_bit==0)Y0=0;        //1.5秒定时到了，Y0=0;
    else Y0=1;
    if(X1==0) {T0_K=1;T0_bit=1;} //复位动作开关
}
```



## 5: 红外遥控功能

接受头

遥控器



设定/退出



设定计划值ABC



设定控制板IP



输出操作



设定位右移



设定位左移





















设定值-



设定值+

**操作使用**（输入正确IP后，才能操作设定功能）：

- 1: 按  键，进入设定操作状态，这时显示控制板的IP.如0015，先按  键，再按  键就进入设定状态了，默认设定ABC。
- 2: 这时显示ABC值，万位闪动，闪动位就是设定位，按  键，闪动左移，按  键，闪动右移.按  键与  键闪动位值增或减，按  键—  键，闪动值等于按下的数字键。
- 3: 按  键是设定控制板IP，LED显示2位.操作同上。
- 4: 按  键是操作输出功能。这时显示的是ABA值，4位0.5秒闪动。这时按下  键—  键应输出取反，比如这时Y0灯亮，按   灯亮，再按  键—  灯灭，以此类推。
- 5: 设定完按  键退出设定。

如果需要红外功能main函数IR\_flag5=0;，X0的光藕要更换红外接受头。且X0不要使用了。

更多功能，后续。。。。