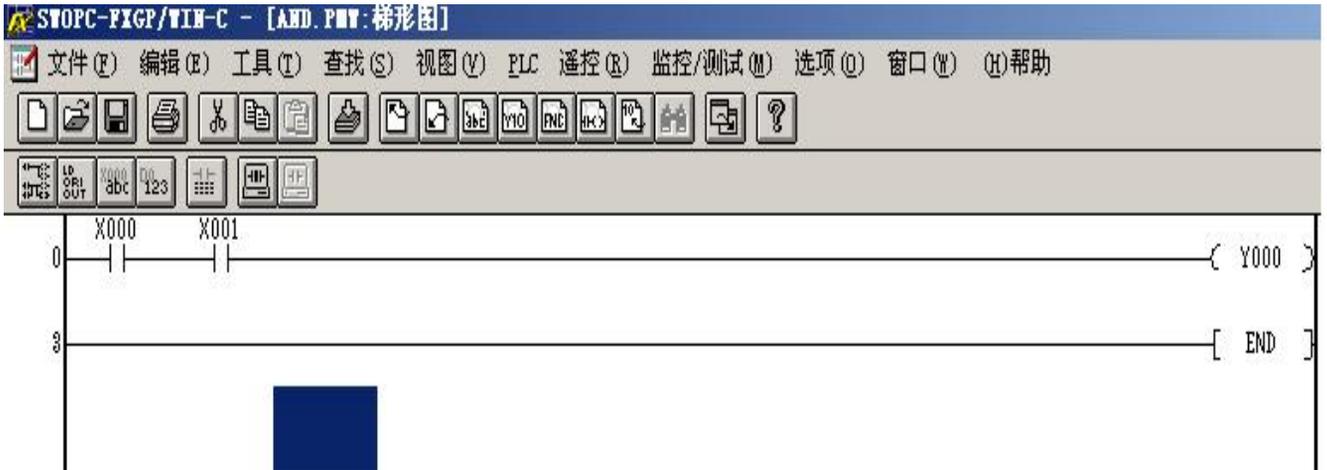


C语言实现梯形图写法

| | |
|-------|------|
| 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

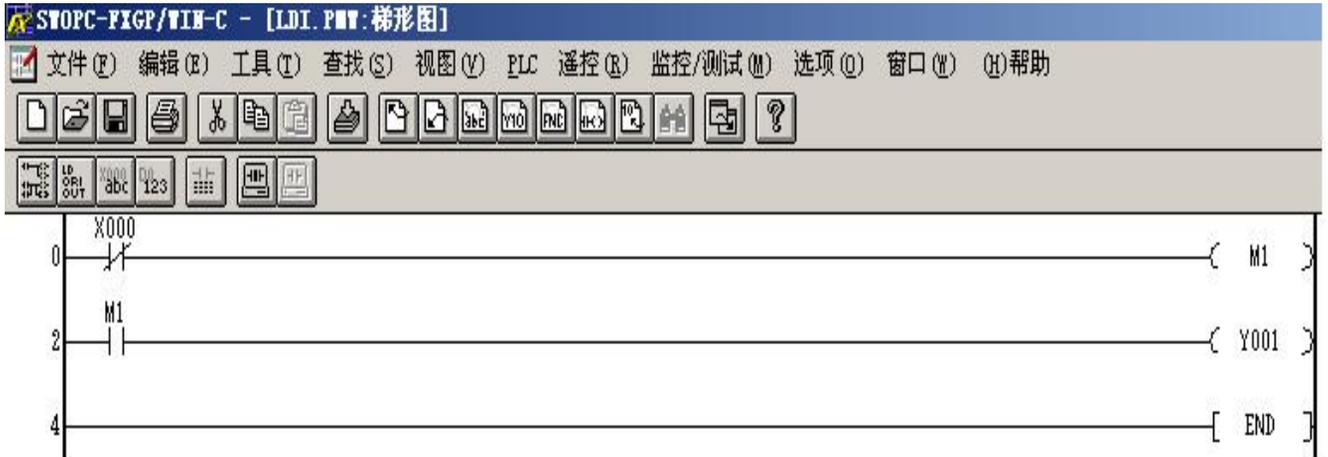


C语言写法:

```
/*  
这里添入你的C逻辑程序。  
*/  
void PLC_Task(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

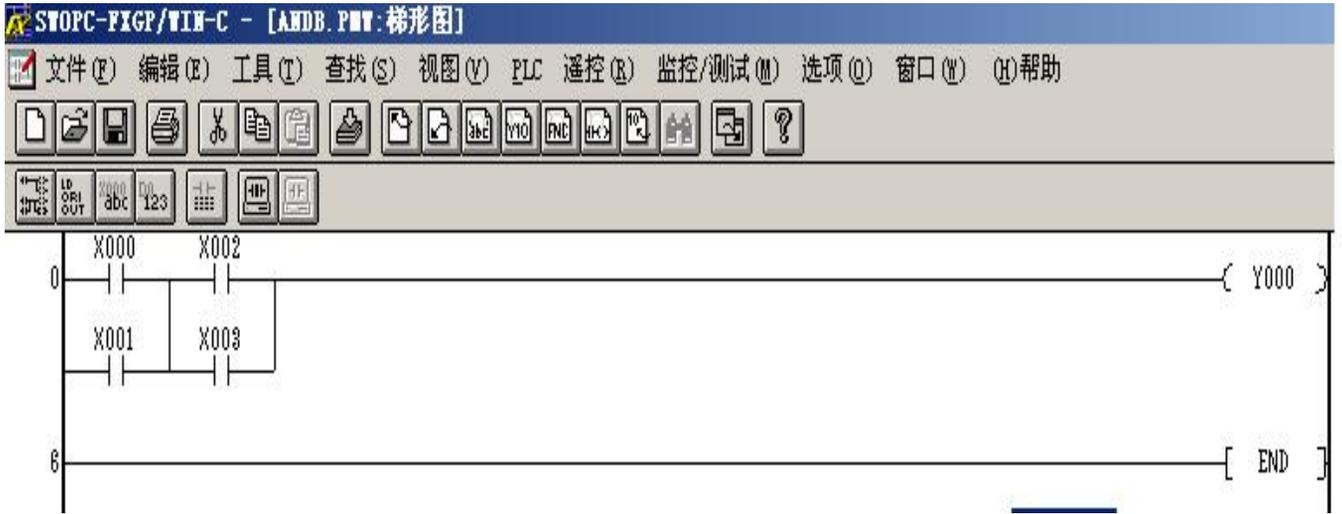


C语言写法:

```
/*  
这里添加你的C逻辑程序。  
*/  
  
void PLC_Task(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:

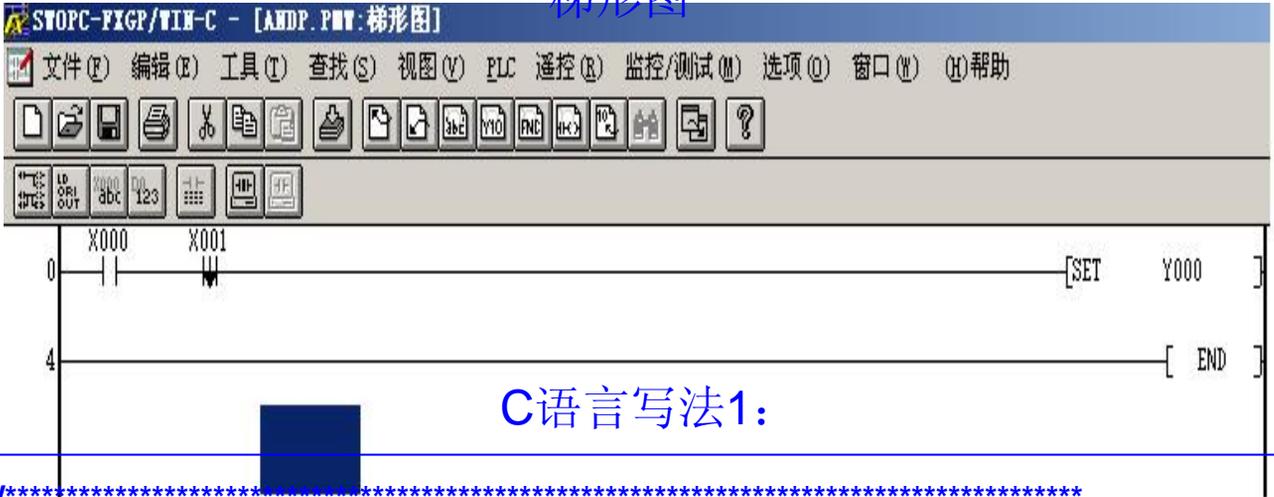
```
/******  
这里添入你的C逻辑程序。  
******/
```

void PLC_Task(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, YO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:

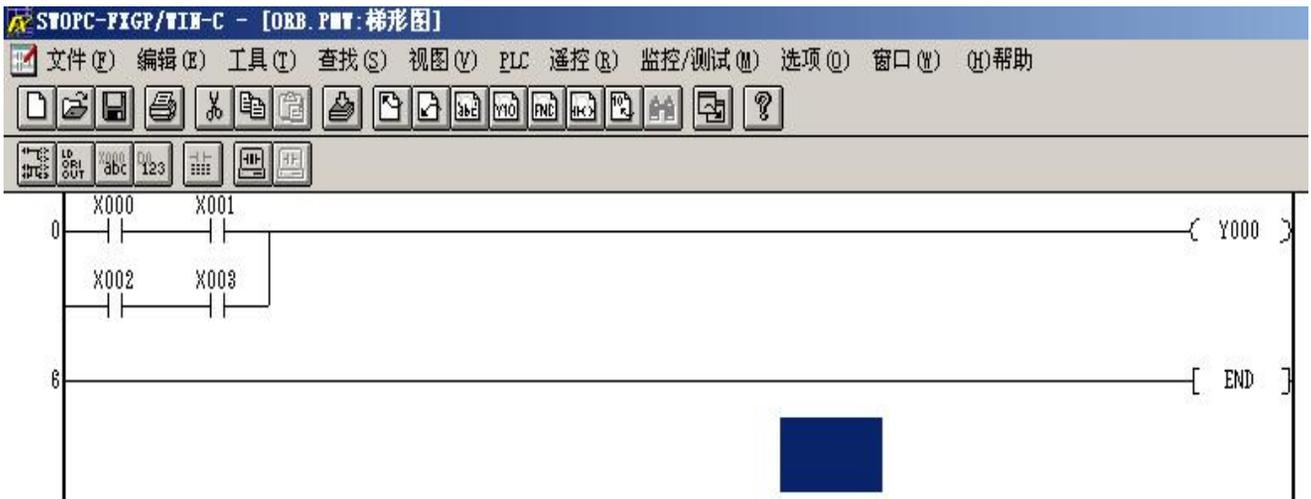
```
/*  
这里添入你的C逻辑程序。  
*/  
  
void PLC_Task(void)  
{  
    if(X1==0) M0=0;           //如果X1接通，辅助继电器M0通电。表示X1接通过。  
  
    if((X0==0)&&(X1==1)&&(M0==0)) //如果X0接通，X1松开了，辅助继电器动作了。  
    {  
        M0=1;Y0=0;           //辅助继电器复位，Y0通电保持。  
    }  
}
```

C语言写法2:

```
/*  
这里添入你的C逻辑程序。  
*/  
  
void PLC_Task(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

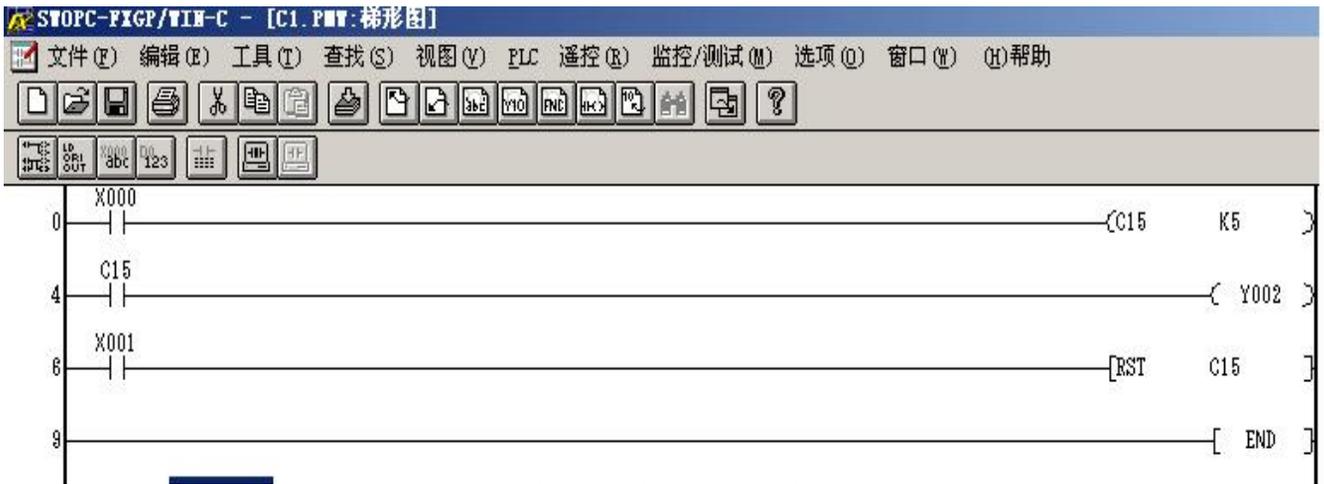


C语言写法:

```
/*  
这里添入你的C逻辑程序。  
*/  
  
void PLC_Task(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)：

```
/*  
*****  
这里添入你的C逻辑程序。  
*****  
*/
```

```
void PLC_Task(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。
```

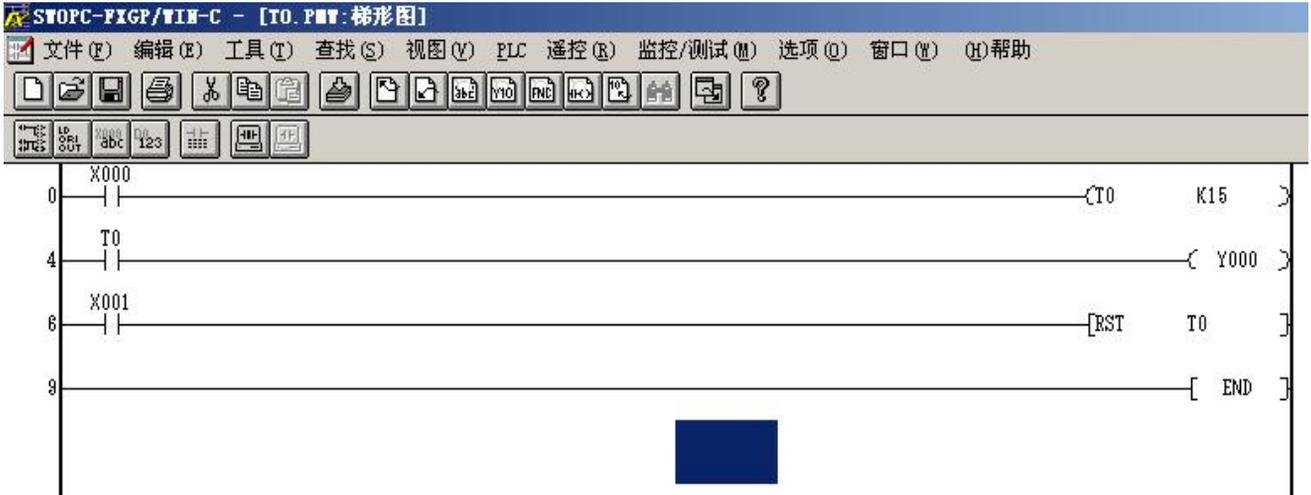
```
        else Y2=1;                  //如果计数器没有动作够，Y2停止。
```

```
    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

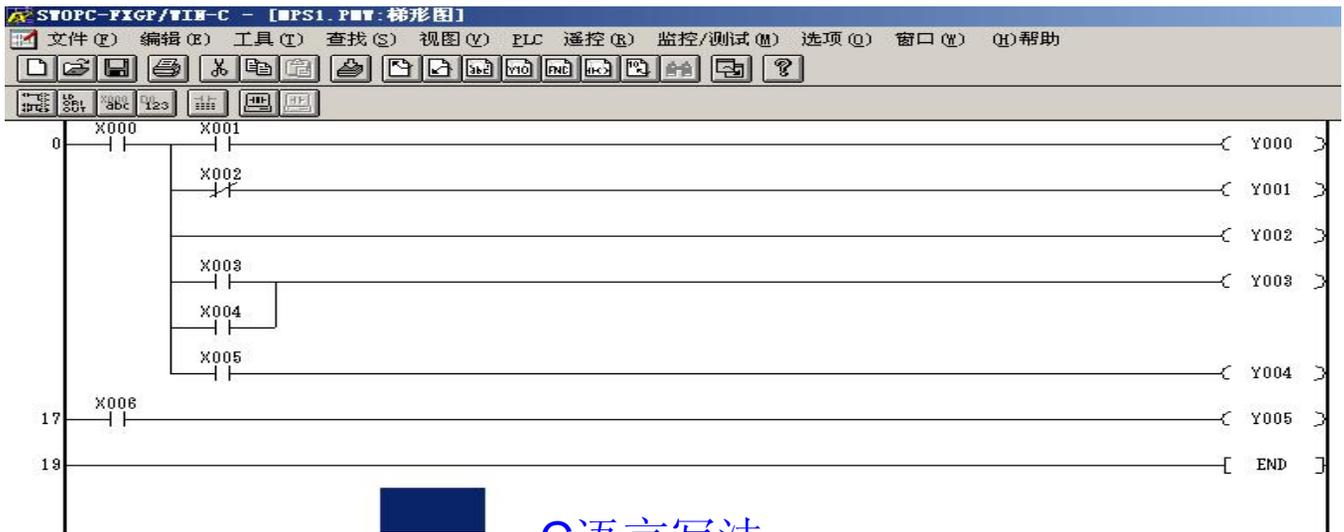


C语言写法:

```
/*
这里添加你的C逻辑程序。
*/
void PLC_Task(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



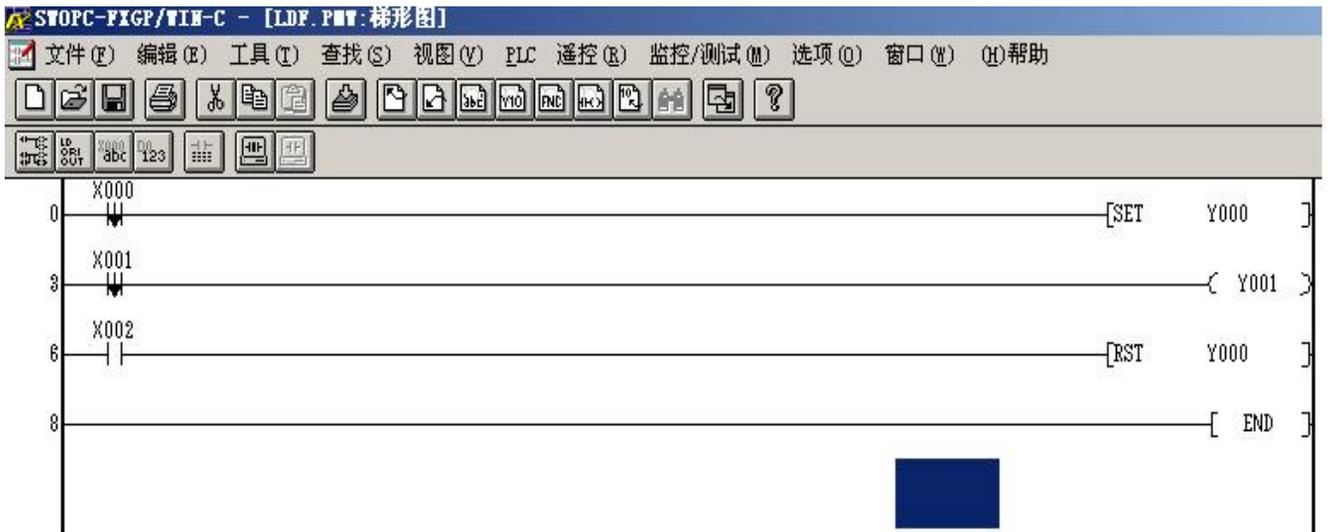
C语言写法:

```
/*  
这里添入你的C逻辑程序。 */  
void PLC_Task(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_Task(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:

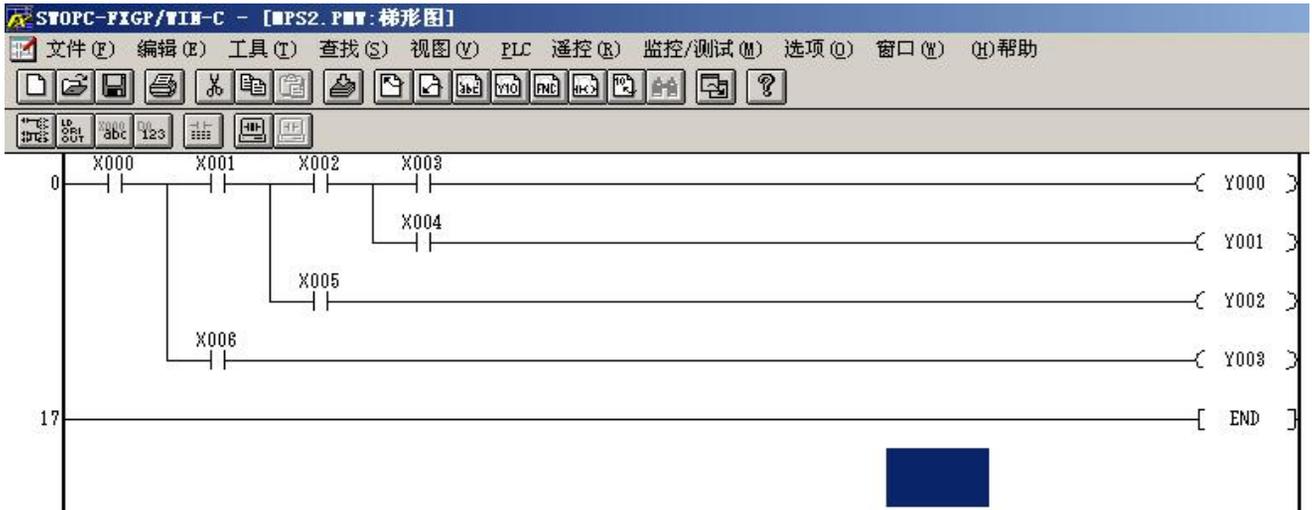
```
/*  
这里添入你的C逻辑程序。  
*/
```

void PLC_Task(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:

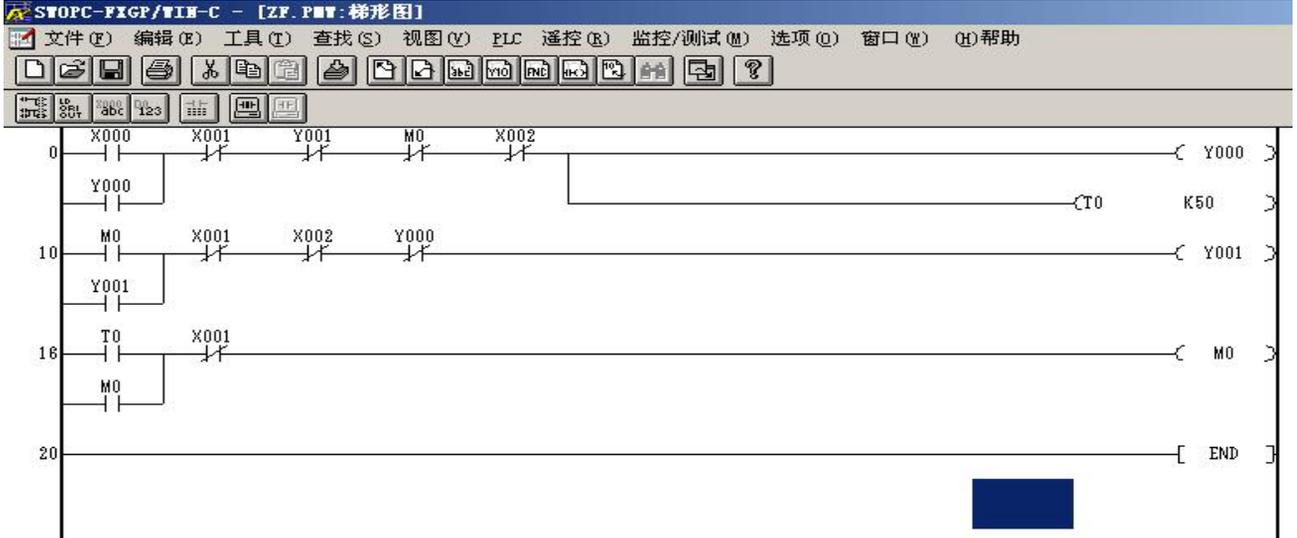
```
/*  
这里添加你的C逻辑程序。  
*/
```

void PLC_Task(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



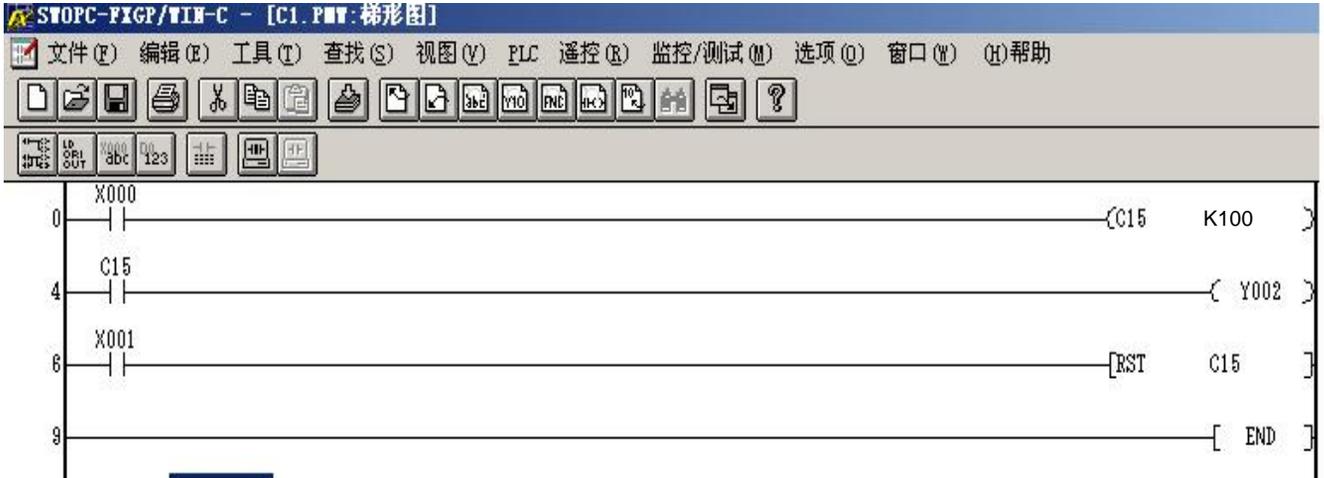
C语言写法:

```
/*  
*****  
这里添入你的C逻辑程序。  
*****  
*/  
void PLC_Task(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。  
}
```

控制板简单使用例子。

1: 带显示的倒计数器。

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）：

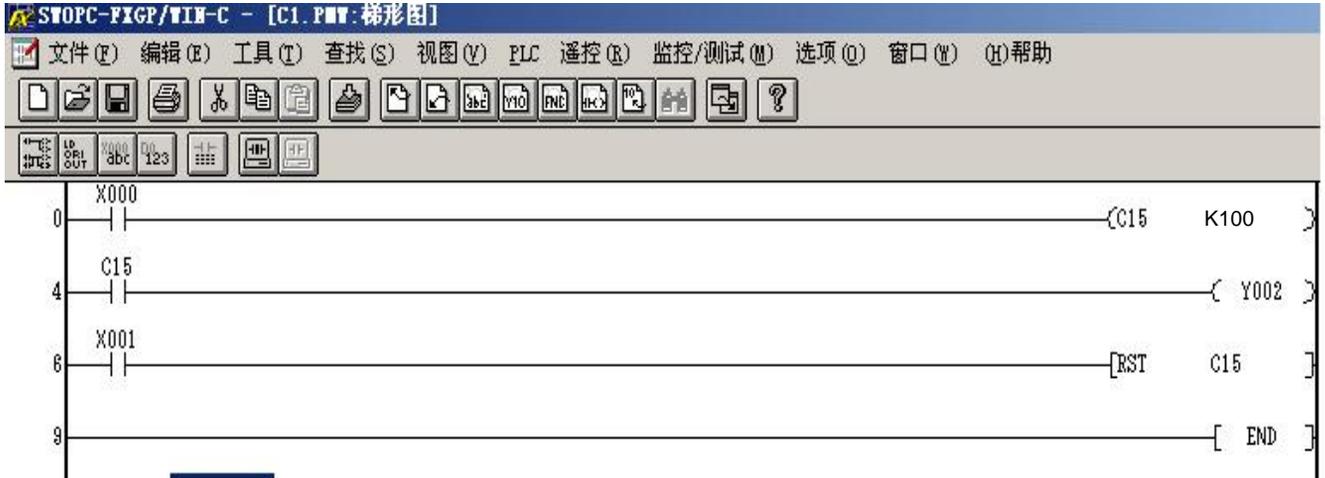
X0为加数开关，每动作一次,C0加一次。加到设定值，Y2动作。X1为计数器复位开关。LED数码管显示C的值。

void PLC_Task(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  
                                       //如果没有计数5次，继续计数。  
        else C0K--;                 //把C0K值送到LED显示。  
        ABA=C0K;  
    }  
    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）：

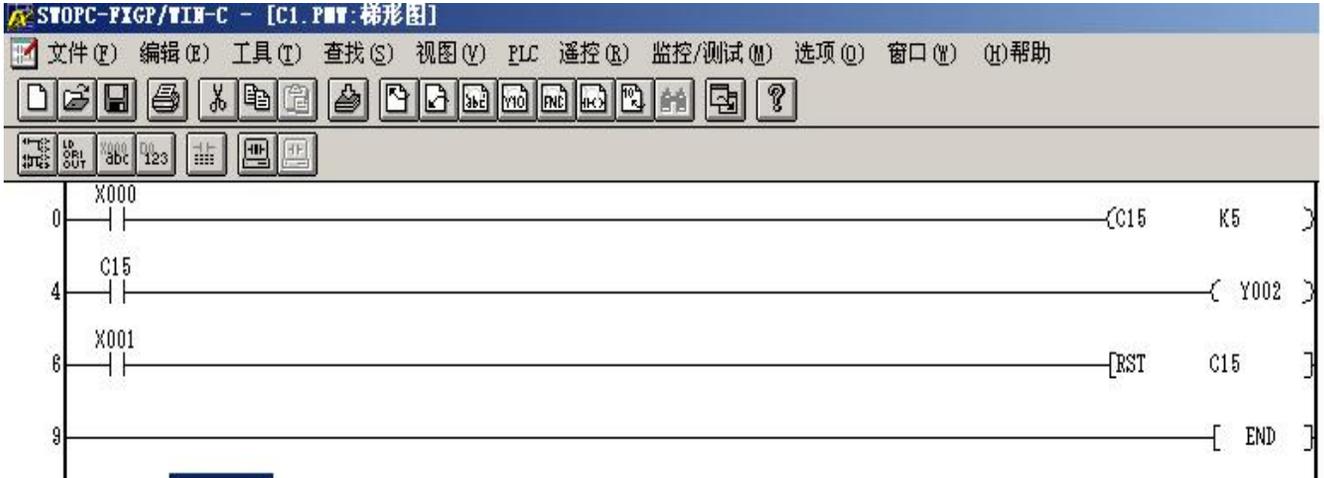
X0为加数开关，每动作一次,C0加一次。加到设定值，Y2动作。X1为计数器复位开关。LED数码管显示增长的值。

void PLC_Task(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）：

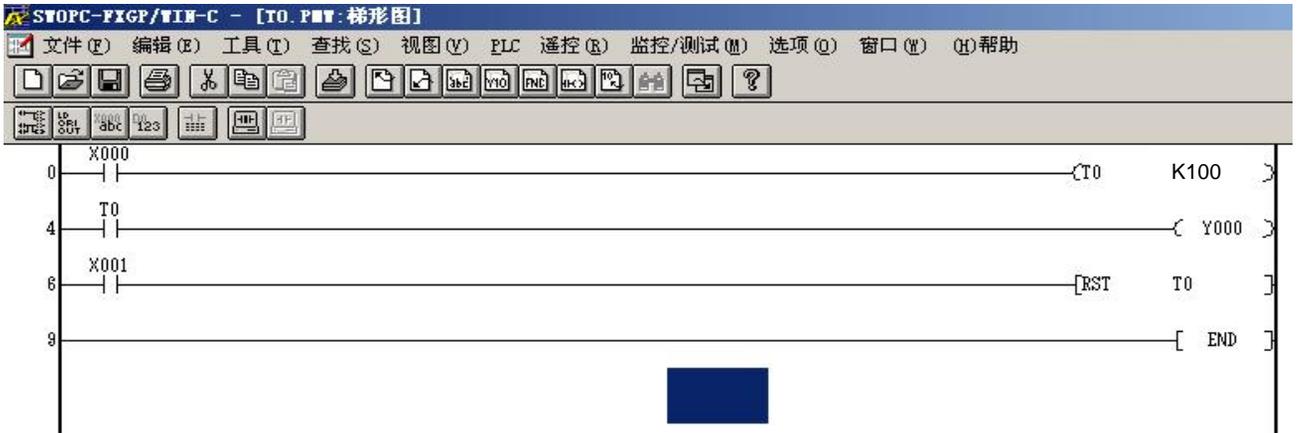
X0为加数开关，每动作一次，C0加一次。加到设定值，Y2动作。X1为计数器复位开关。LED数码管显示增长的值，每计数一次，存储计数值，断电数据不丢失。

void PLC_Task(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_Task(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  
}
```