

# 程序总框架及 7279 部分程序介绍

---

## 主要内容：

- 主程序框架
- 显示部分程序
- 键盘部分程序

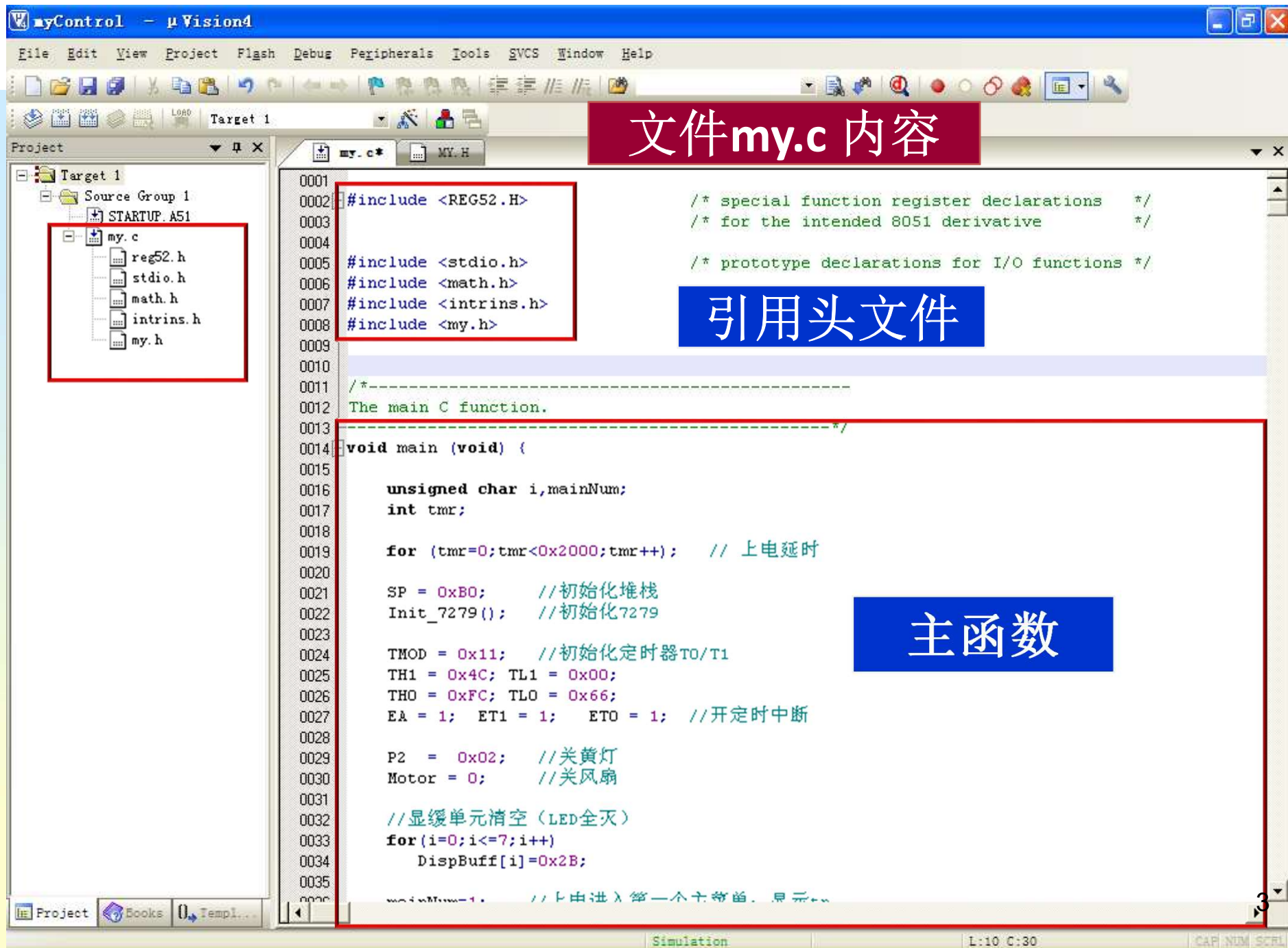
# 一、主程序框架

## (1) 头文件

主程序**My.C** 中，包含了几个头文件：

```
#include <REG52.H>
#include <stdio.h>
#include <math.h>
#include <intrins.h>
#include <my.h>
```

- 1)、**my.h**中有全局变量定义、函数声明、特殊位定义等；
- 2)、先编辑好**my.h**，存在项目文件夹里；
- 3)、在**my.c**中写**#include <my.h>**，然后Rebuilde程序**my.c**，左边的**my.c**项目分支中自然就会挂上**my.h**



## 头文件my.h 内容

```
001 #define U8 unsigned char
002 //延时
003 #define Somenop();      _nop();_nop();_nop();_nop();_nop();
004 #define Somenop10();    Somenop();Somenop();
005 #define Somenop25();    Somenop();Somenop();Somenop();Somenop();Somenop();
006 #define Somenop50();    Somenop25();Somenop25();
007 #define Somenop100();   Somenop25();Somenop25();Somenop25();Somenop25();
008
009 #define      CMD_RESET      0xA4      // 复位
010 #define PIDA 12
011 #define PIDB 10
```

### 宏定义

```
012
013 sbit CS      = P1^4;      // 7279 CS(片选) 接到P1.4口
014 sbit CLK     = P1^5;      // 7279 CLK 接到P1.5口
015 sbit DATA   = P1^7;      // 7279 DATA 接到P1.7口
016 sbit DS1820_DQ = P1^3;    // DS18B20 接到P1.3口
017 sbit Motor   = P1^2;      // 电机
018
019 sbit ECLK    = P1^1;      // E2PROM时钟
020 sbit EDTA    = P1^0;      // E2PROM起始数据
```

### 特殊位定义

```
021
022 unsigned char KeyNum,KeyValue;      //键号、键值
023 unsigned char TempCount,RunCount,MotorCount,MotorNow,PACount;
024 unsigned char rNum,bFNum,NowEPVal,LastVal;
025 int NowEPVal1,LastVal1;
026 unsigned int KeyCount1,KeyCount2;
027 unsigned char PAStep;
```

### 全局变量

```
028
029 void Init_7279(void);      // HD7279初始化
030 void send_byte(unsigned char); // 发送一个字节
031 void delay(unsigned char); // 延时
032 void write_7279(unsigned char, unsigned char); // 写数据
033 void display(unsigned char buff[]); // 写数据
034 unsigned char receive_byte(void);
035 unsigned char ReadKey(void);
```

### 函数声明

## (2) 主函数框架

```
void main (void)
{
    // 变量定义                    // 上电延时
    // 初始化堆栈                  // 初始化7279
    // 初始化定时器T0/T1          // 定时中断设置
    // 外设初始状态                // 关绿灯、关风扇
    // 显缓单元清空 (LED全灭)     // 变量初始化

    while(1)
    {
        display(DispBuff); //显示 (按显缓单元的内容显示)
        Key(); //键盘扫描, 返回键号KeyNum或0xff (无键按下)
        if (KeyNum!=0xff)
        {
            // 有键按下, 按照键号执行菜单显示或进入子菜单
            if (KeyNum==0) .....
            else if (KeyNum==1) .....
                .....
        }
    }
}
```

## 二、显示部分程序

### (1) 7279初始化

```
void Init_7279(void)
{
    CS = 0;                // 片选使能置0
    Somenop50();Somenop50(); // 延时
    send_byte(CMD_RESET);  // 7279复位命令
                           // CMD_RESET=A4H
    Somenop50();Somenop50(); // 再延时
    CS = 1;                // 片选使能置1，完成初始化
}
```



## (2) 7279显示程序

显缓单元的设置：

为8个LED设置8个连续的显缓单元 —— 显缓数组。

```
unsigned char DispBuff[8];
```

目标：让任意一个LED显示任意一个字符，不论是数字还是自定义字符。

工作：(1) 无需查硬件连线，**编程**获得上下排8个LED与显缓单元的对应关系；  
(2) **编程**获得LED各段与A~G, dp的对应关系。

## 编写显示测试程序:

### 3、下载数据但不译码

```
main()
```

```
{
```

```
// 各种初始化(必须含7279初始化)
```

```
while(1)
```

```
{
```

```
    write_7279(0x90, 0x01);
```

```
}
```

```
}
```

D7	D6	D5	D4	D3	D2	D1	D0		D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	0	a2	a1	a0		DP	A	B	C	D	E	F	G

0x90

0x91

0x92

0x93

0x94

0x95

0x96

0x97

0x01

0x02

0x04

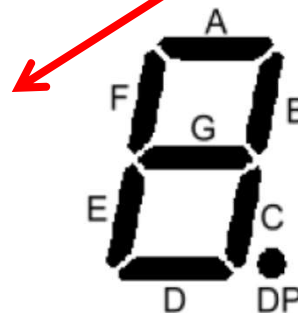
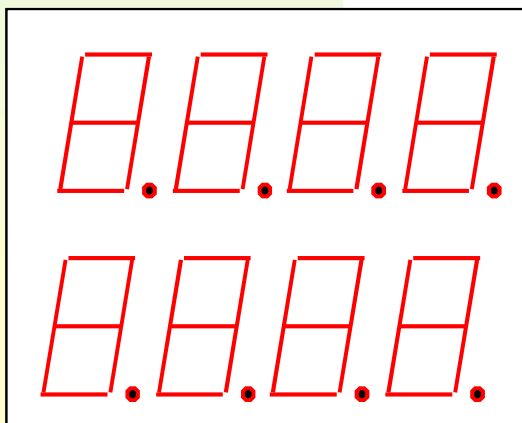
0x08

0x10

0x20

0x40

0x80



D7	D6	D5	D4	D3	D2	D1	D0
DP	A	B	C	D	E	F	G
F	B	E	A	D	C	DP	G



# 构造字形表:



序号	字符	D7	D6	D5	D4	D3	D2	D1	D0	字形码
		F	B	E	A	D	C	DP	G	
00 <sub>H</sub>	0	1	1	1	1	1	1	0	0	FC <sub>H</sub>
01 <sub>H</sub>	1	0	1	0	0	0	1	0	0	44 <sub>H</sub>
.....	.....									.....
09 <sub>H</sub>	9									
0A <sub>H</sub>	A									
0B <sub>H</sub>	B									
.....	.....									.....
0F <sub>H</sub>	F									.....
10 <sub>H</sub>	0.	1	1	1	1	1	1	1	0	FE <sub>H</sub>
11 <sub>H</sub>	1.	0	1	0	0	0	1	1	0	46 <sub>H</sub>
.....	.....									.....
19 <sub>H</sub>	9.									
1A <sub>H</sub>	A.									
1B <sub>H</sub>	B.									
.....	.....									.....
1F <sub>H</sub>	F.									.....
20 <sub>H</sub>	P	1	1	1	1	0	0	0	1	F1 <sub>H</sub>
21 <sub>H</sub>	-	0	0	0	0	0	0	0	1	01 <sub>H</sub>
.....	.....									.....

显示字符  
0 ~ F

显示字符  
0. ~ F.

显示特殊字符

字形表数组:

```
unsigned char code LEDValue[50] =  
    { 0xFC, 0x44, ....., 0xFE, 0x46, ....., 0xF1, 0x01, ....., } ;
```

例: 下排最后一位显示数字字符a, a不确定为0~F中任意值

```
write_7279(0x93 , LEDValue[a]) ;
```

显示位置  
从90<sub>H</sub>~97<sub>H</sub>循环

从显缓数组  
dispBuff[0]~dispBuff[7]  
中取待显示的字符

## 显示子函数:

```
//*****显示子函数*****  
//一次更新8个LED,  需要显示的字符序号在buff中  
void display(unsigned char buff[])  
{  
    unsigned char i;  
    for( i=0 ; i<=7 ; i++ )  
        write_7279(0x90+i,LEDValue[buff[i]]);  
}
```

在主程序中调用显示子函数：

```
main()  
{  
    // 各种初始化  
    // 初始显缓送值, DispBuff中送初始8个显缓值  
    while(1)  
    {  
        display(DispBuff) ;  
        Key() ;  
        if (KeyNum!=0xff)  
            .....  
    }  
}
```

### 三、键盘部分程序

#### (1) 读键值

```
/**读7279键值***/
unsigned char ReadKey(void)
{
    unsigned char readkey;
    CS = 0;
    Somenop50();Somenop50();
    send_byte(0x15);           //读键值命令15H
    Somenop25();Somenop25();
    readkey = receive_byte();  //接收键值(按时序接收)
    CS = 1;
    return(readkey);          //返回键值
}
```

## (2) 键值测试程序 — 在下排的第1、2位显示键值

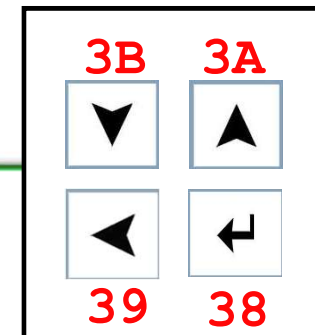
```
main()
{
    unsigned char temp,i;
    // 各种初始化(必须含7279初始化)
    for(i=0;i<8;i++)    DispBuff[i]=0x2B; //初始全灭

    while(1)
    {
        display(DispBuff);
        temp = ReadKey(); //读键值
        DispBuff[0] = (temp>>4) & 0x0f;
        DispBuff[1] = temp & 0x0f;
    }
}
```

显示键值高4位和低4位

构造键值表:

```
unsigned char code KeyTabel[4] =
    {0x3B,0x3A,0x39,0x38};
```





### (3) 键盘扫描函数 Key()

```
void Key(void)
```

```
{
```

```
    unsigned char temp,i;
```

```
    temp = ReadKey(); //读键值
```

```
    if (temp==0xff)
```

```
    {
```

```
        KeyNum = 0xff;
```

```
        KeyValue = 0xff;
```

```
    }
```

```
    else
```

```
    {
```

```
        if (KeyValue!=0xff)
```

```
            KeyNum=0xff;
```

```
        else
```

```
        {
```

```
            KeyValue = temp;
```

```
            for(i=0;i<=3;i++)
```

```
                if (KeyValue == KeyTabel[i])
```

```
                    { KeyNum = i; break; }
```

```
            }
```

```
        }
```

```
    }
```

```
unsigned char code KeyTabel[4] =  
    {0x3B,0x3A,0x39,0x38};
```

KeyNum = 0xff;  
KeyValue = 0xff;

若当前无键按下，  
则键值=ff，键号=ff

if (KeyValue!=0xff)  
 KeyNum=0xff;

防止键盘连续响应：  
若当前有键按下，但前一次键值有值  
说明是键盘没有抬起，此时键号=ff

KeyValue = temp;  
for(i=0;i<=3;i++)  
 if (KeyValue == KeyTabel[i])  
 { KeyNum = i; break; }

根据键值查键号

