

实验一 键盘显示实验

一、 实验目的

- (1) 掌握MCS51 系统中键盘显示接口的方法;
- (2) 掌握中断处理程序的编程方法。

二、 硬件原理图

2.1 数码管显示原理介绍

实验箱上有8个共阴极数码管,图2-1是其外部特性图,图2-2是其内部原理图,由图2-2 可知,每个数码管由8个发光二极管组成,其中a~DP称为数码管的段控信号,K是8个发光二极管的公共端,称为位控信号。所以为了让数码管显示一个数字,必须将位控信号接低电平,段控信号接高电平。例如:如要显示数字“1”,b端和c 端应该接高电平,其余各端接低电平。

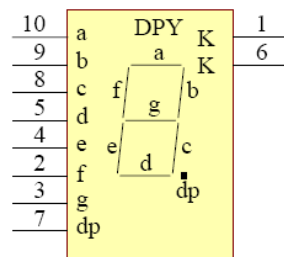


图 2-1 八段数码管外部特性图

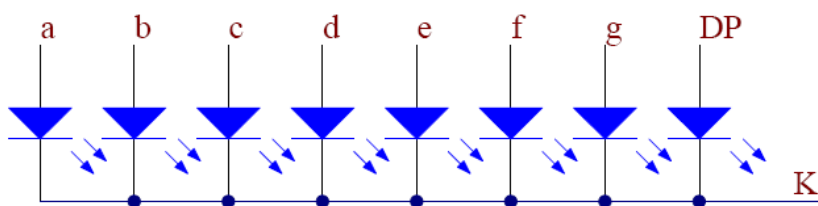
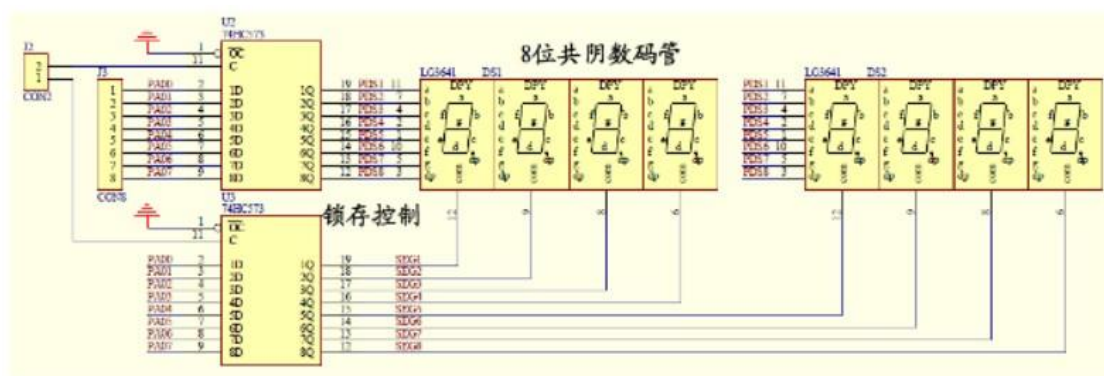


图 2-2 八段数码管内部原理图



模块连线方式:

杜邦线连接（用杜邦线连接<单片机 IO>和<模块接口>）			
单片机 IO 口	模块接口	杜邦线数量	功能
P0	J3	8	共阳数码管数据端
P2.2（段锁存）	J2（B）	1	段锁存
P2.3（位锁存）	J2（A）	1	位锁存

2.2 键盘扫描原理介绍

键盘扫描原理如图2-3 所示，“K1”和“K2”是按键，按键的两端都接有10k的上拉电阻。当行扫描端为低电平，同时“K1” 键被按下时，“K1”的列读取端就会出现低电平，由处理器对行扫描信号和列读取信号进行综合判断后，得出“K1”键被按下的结论。

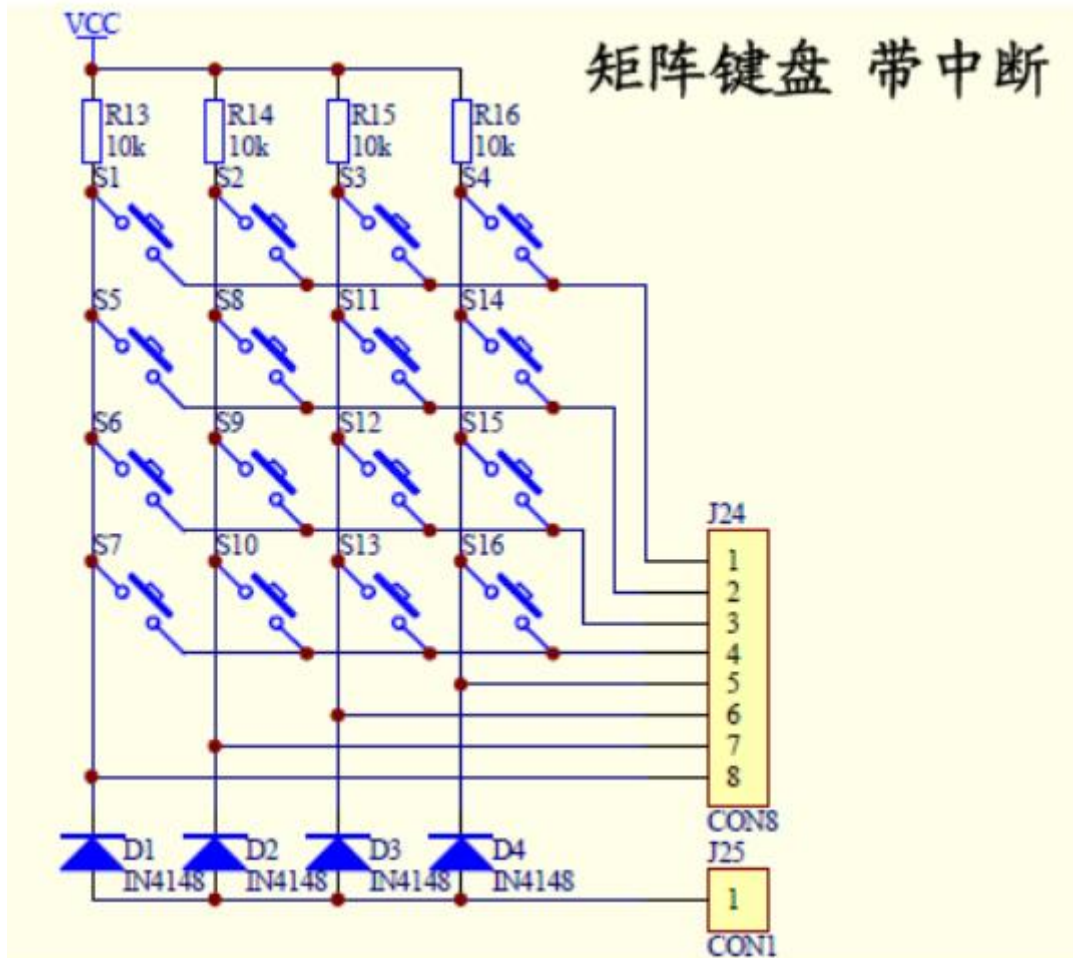


图 2-3 键盘显示原理图

模块连线方式：

单片机 IO 口	模块接口	杜邦线数量（根）
P1 或者 P3	J24	8
P3.2	J25（仅中断扫描用到）	1

三、 实验要求

（1）用两个学时完成显示部分程序。

- 1) 要求在 8 个数码管上显示小组每位同学的学号；
- 2) 要求分别用汇编和 C 语言实现。

（2）用两个学时完成键盘部分程序。

- 1) 求结合显示部分程序，当按下某个键时，在数码管上显示键值，如按下键“1”时，数码管上显示 1，需要显示的键值范围为“1~F”，键“B”用小写字符“b”显示，键“D”用小写字符“d”显示，其余字母用大些表示；

2) 可以选用汇编或 C 语言实现。

(3) 做完实验后，每位同学都需要给老师讲解实验的程序，验收合格后再整理实验装置离开。

四、 开发环境

4.1 编程调试软件

开发调试程序使用的软件是 KeilC，关于软件的操作步骤如下：

(1) 打开软件

在计算机桌面上有快捷方式“Keil uVision3”，双击快捷图表后，出现如图 4-1 所示的界面。

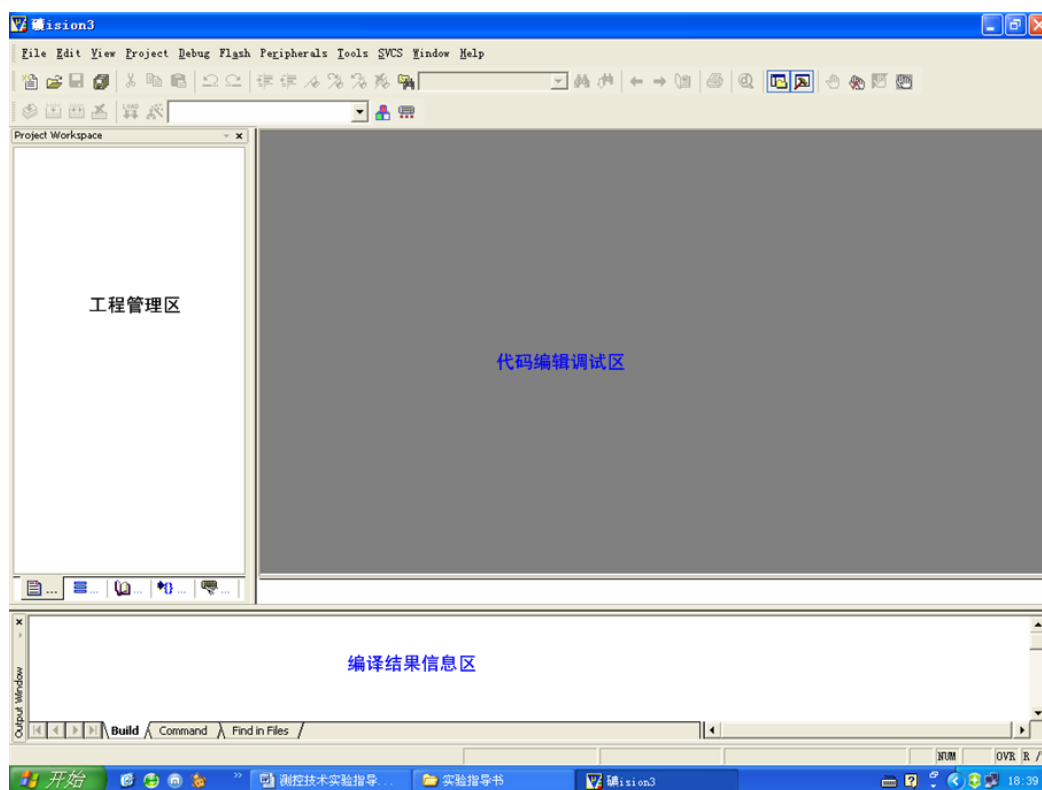


图 4-1 keilC 软件界面图

(2) 建立工程

点击软件菜单栏里的 Project，选择 new project，根据软件提示输入工程名，以及选择好工程的存储路径。注意计算机的 C 盘和 D 盘具有写保护功能，为防止由于计算机重启造成文件丢失的情况发生，请将自己的程序以自己姓名为文件夹名存储在 F 盘中。

(3) 选择 CPU

当工程建立完后，会出现选择 CPU 的对话框，请选择 ATMEL 公司的

AT89S52 单片机，如图 4-2 所示。

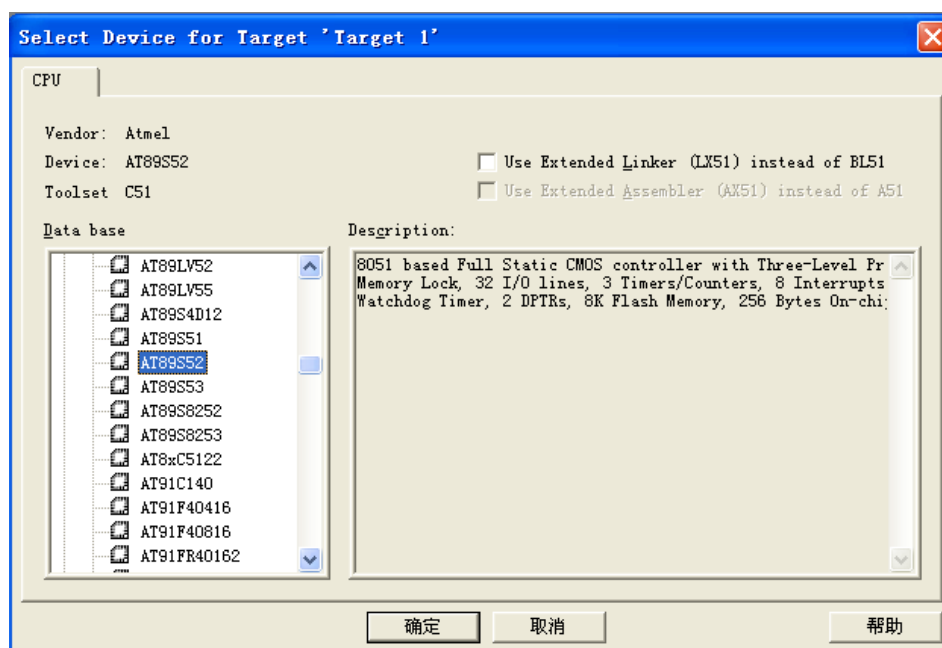


图 4-2 CPU 选择界面

选择好 CPU 后点击确定，会出现如图 4-3 所示的提示界面，请选择否。



图 4-3 是否添加标准启动代码界面

(4) 新建文件

如图 4-4 所示，建立一个新文件，在该文件中写入自己的程序代码，并保存，如果是用汇编语言编写的程序，请将文件的后缀改写为“`.ASM`”；如果是用 C 语言编写的程序，请将文件的后缀改写为“`.C`”。文件的保存位置与新建工程的保存位置要一致。

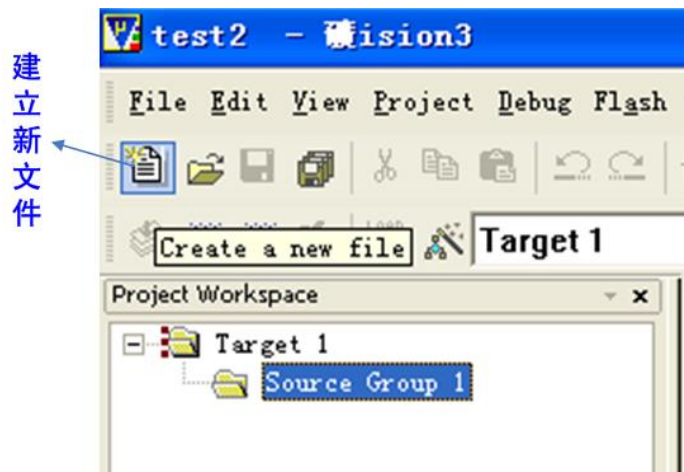


图 4-4 建立新文件界面

(5) 向当前工程添加文件

按照图 4-5 的方法可以向当前工程中添加文件，所添加的文件即为自己所编写的代码文件，可以是汇编语言代码文件，以“.ASM”为文件后缀，也可以是 C 语言代码文件，以“.C”为文件后缀。

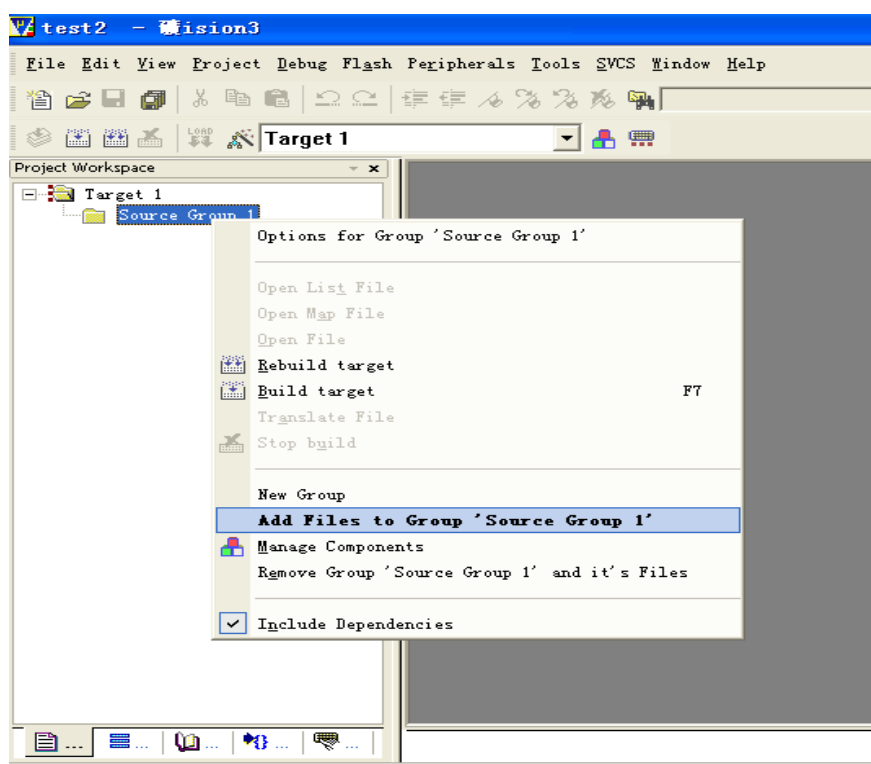


图 4-5 向工程添加文件界面

(6) 编译设置

为了生成可以下载到单片机中的文件，需要对工程的编译输出文件属性进行设置，操作过程如图 4-6 和图 4-7 所示，在图 4-7 中需要将“Creat HEX File”选

中。

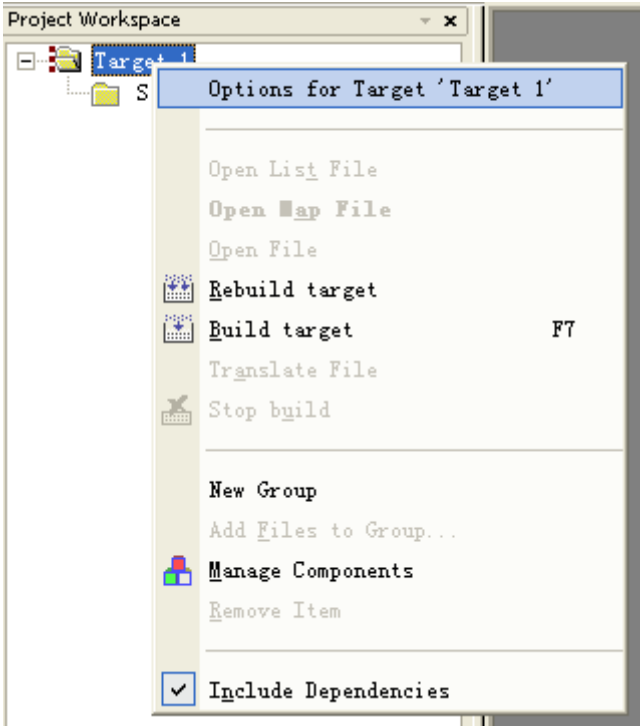


图 4-6 编译设置选择

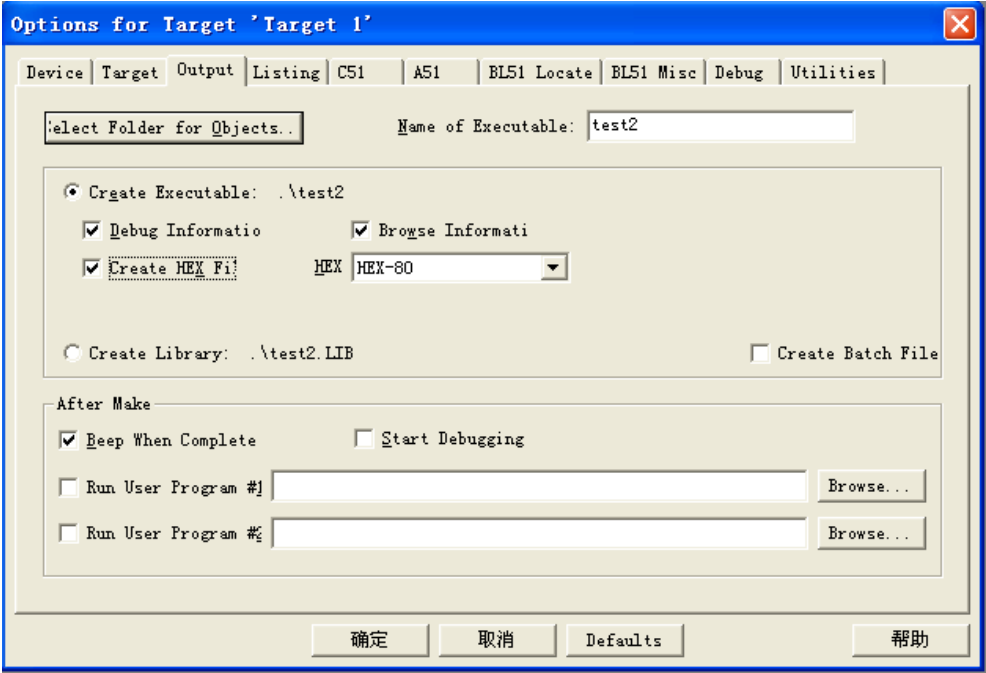


图 4-7 编译输出文件设置

(7) 对文件进行编译

当程序代码编写完后，按图 4-8 的方法可以对文件进行编译，生成可下载到单片机中的程序。

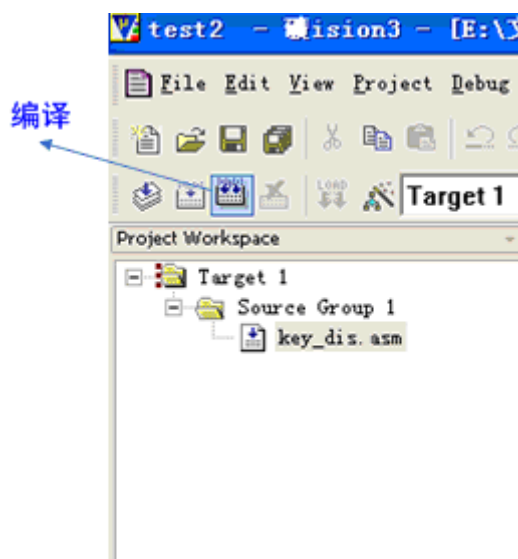


图 4-8 对程序文件进行编译界面

编译链接成功后，出现如图 4-9 所示窗口，表示成功；若不成功，还需要排除 C 源文件中的语法等错误后，再次编译，直到成功。

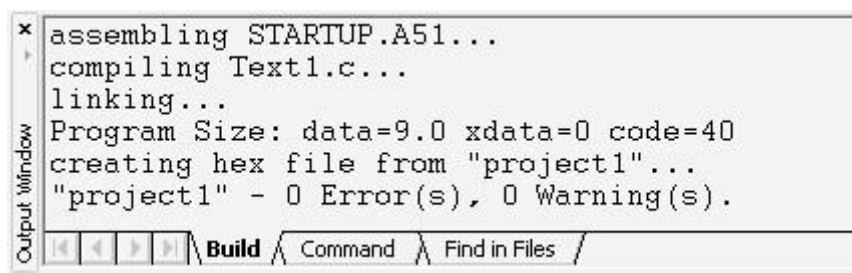


图 4-9 编译链接成功后界面

(8) 软件仿真调试

KeilC 软件有很强的软件仿真调试功能,通过软件仿真可以使程序单步运行,通过软件中提供的“Watch”窗口以及特殊寄存器观察窗口,可以发现程序执行过程中的问题,关于调试功能的详细使用请参考相关书籍。调试按钮如图 4-10 所示。



图 4-10 对程序进行调试

仿真时出现如图 4-11 所示的仿真窗口。其左上角是寄存器窗口，显示 CPU 中各个寄存器的内容，右下角是变量显示窗口，其中，Local 页面显示局部变量，

Watch 页面可以通过设置添加并显示用户所需要的各种变量。

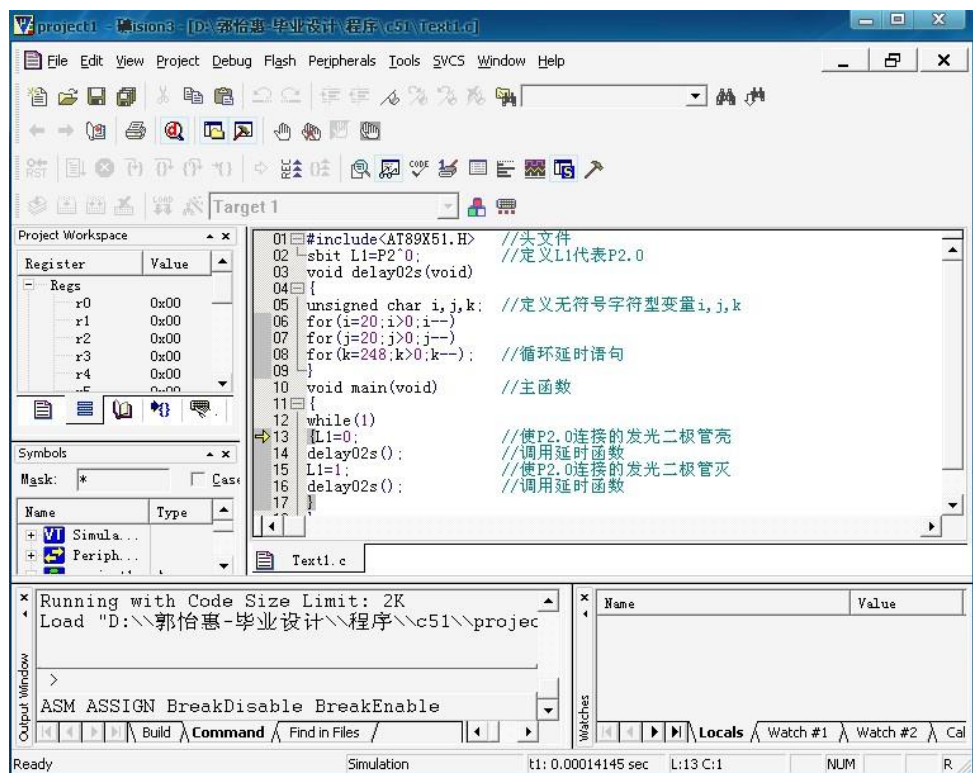


图 4-11 仿真调试窗口

在仿真窗口中，按照图 4-12 来选择菜单，则可以将单片的端口添加到仿真窗口中。

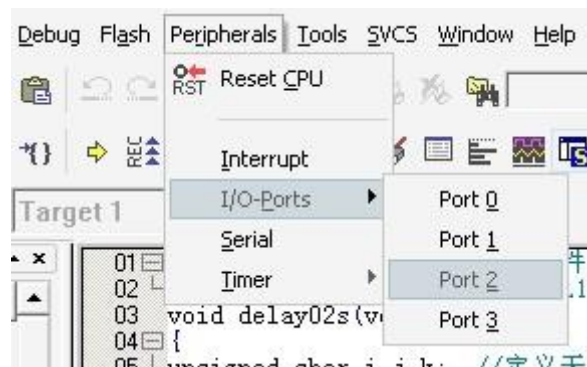


图 4-12 将单片机端口添加到仿真调试窗口

随后屏幕弹出显示单片机端口的各个引脚状态的窗口，如图 4-13 所示。

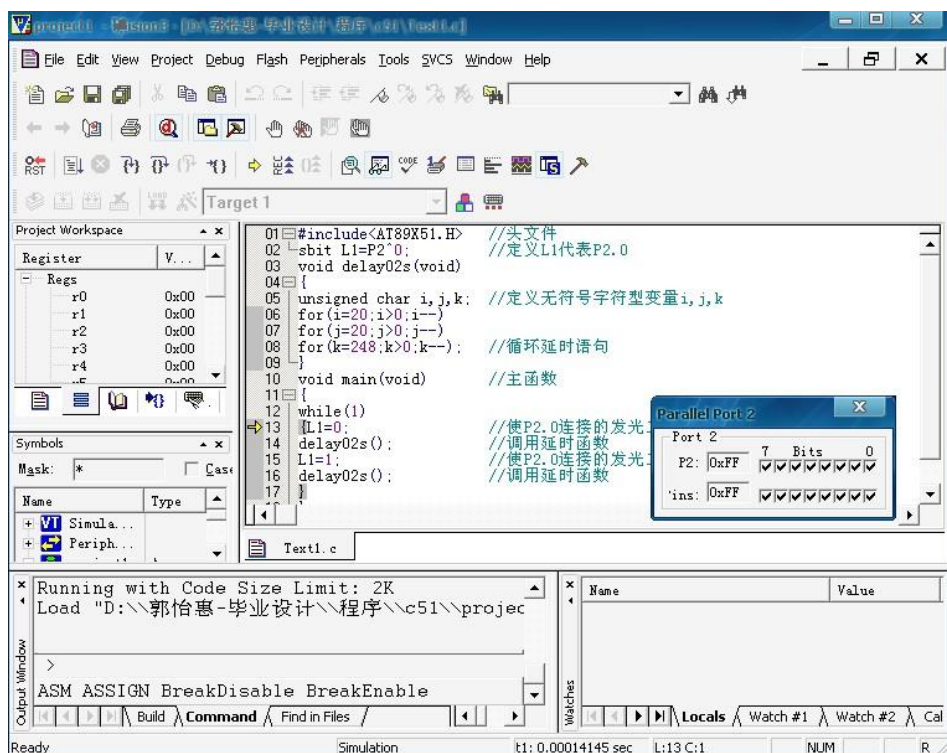


图 4-13 单片机端口状态显示界面

在图 13 中，单击 Debug 菜单，在其下拉菜单中单击 Run 选项。程序开始运行，可以看到 P2 口的 P2.0 引脚的状态在变化，表示实际电平变化。若在该引脚接发光二极管，则可以看到发光二极管闪烁。

到此为止，已经完成了一个项目实现的全过程，因为已经生成用于下载到单片机中的扩展名为 HEX 的文件（简称 HEX 文件），生成后的 HEX 文件就在项目文件夹中。

4.2 程序下载软件

下载程序需要用到另外一个软件 STC-ISP 下载软件，从桌面找到 STC-ISP 下载软件的快捷方式，双击打开软件。进入 STC-ISP 下载软件后的界面如图 4-14 所示。

分 4 步进行：

- 1) 界面左上角显示的为要下载程序单片机的型号，型号如果不对可以单击下拉选择正确的单片机型号，必须是 STC 系列。（注意：实验板芯片型号为 STC89C54RD+）。
- 2) 打开需要烧些的 HEX 文件。
- 3) 选择对应的串口（该串口可以在设备管理器里找到，如图 4-15 所示的串口号为 com3）。

4) 点击下载，关闭开发板电源，然后稍等片刻打开电源，等待下载完成。

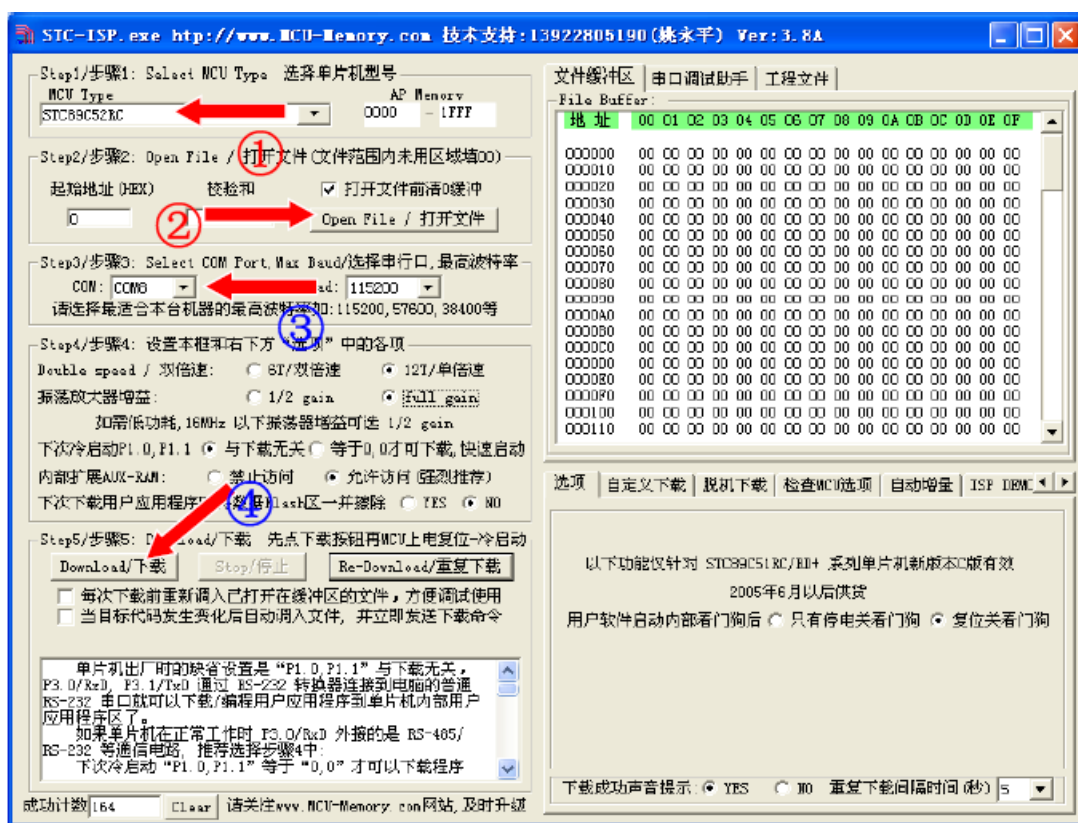


图 4-14 进入 STC-ISP 下载软件的界面

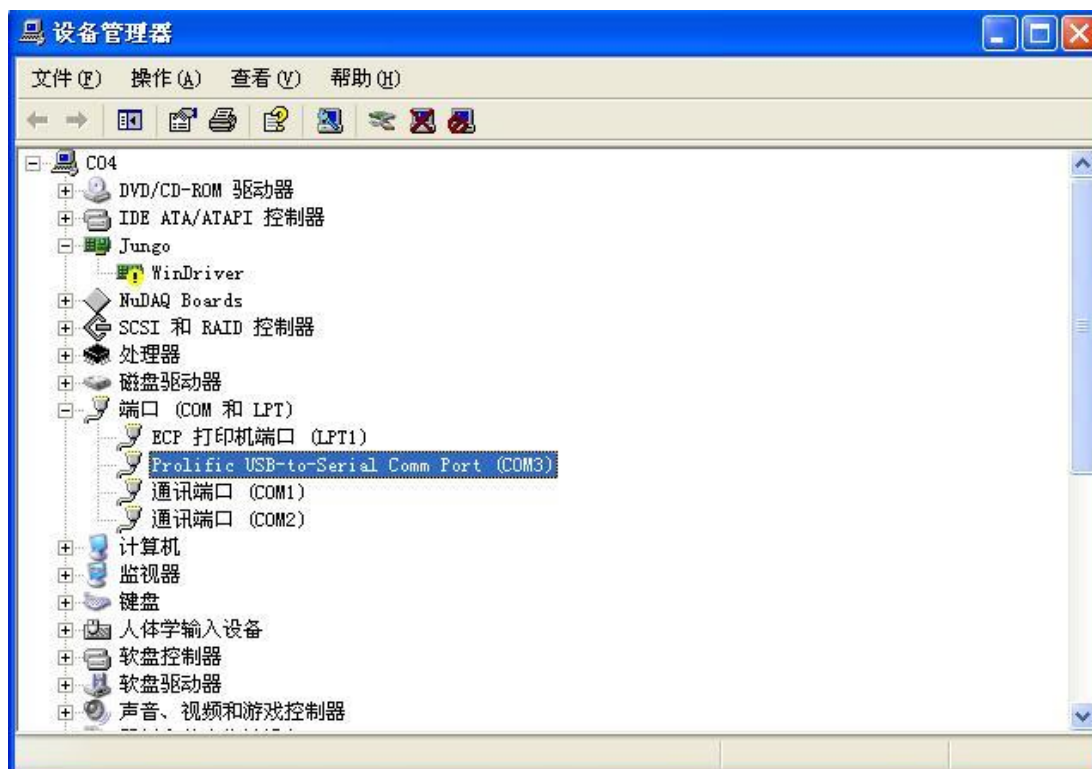


图 4-15 通过设备管理器查找串口号界面

若出现如图 4-16 所示的界面，请检查如下原因。

- 1) 第 4 步开关顺序不正确
- 2) 串口选择不正确
- 3) 串口线或者 usb 转串口线没有连接好
- 4) 晶振没有插紧
- 5) 芯片没有放置正确（不要放偏）
- 6) 软件不兼容，可以下载最新版本
- 7) 最高波特率调至 115200

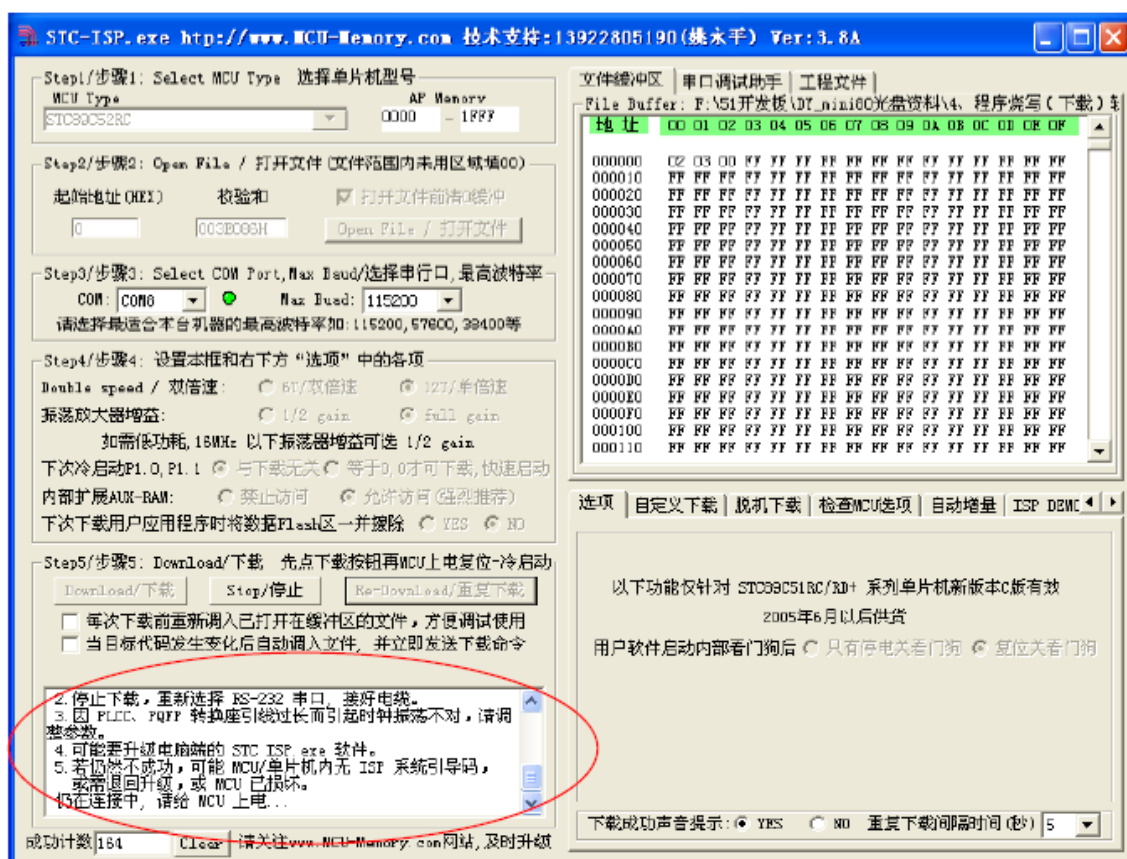


图 4-16 下载时出错界面

五、 例程参考

5.1 汇编语言程序结构框架

;程序中中断跳转设置

ORG 001BH ;定时器 1 中断

LJMP TIMER_1

```

        ORG      0023H      ;串口发送或接收中断
        LJMP     UART_RI_TI

        ORG      002BH      ;定时器 2 中断
        LJMP     TIMER_2

;主程序
        ORG      0030H

INT_0:
        .....

        RETI              ;中断返回指令
        .....           ;各应用程序模块

        END              ;程序结束

```

显示部分示例：

```

;动态扫描数码显示程序
;P0 口接 J3，数据端口
;P2.2 接 J2 B 端子，表示段码锁存
;P2.3 接 J2 A 端子，标志位码锁存

ORG 00H

MOV 20H, #3FH    ;0
MOV 21H, #06H    ;1
MOV 22H, #5BH    ;2
MOV 23H, #4FH    ;3
MOV 24H, #66H    ;4
MOV 25H, #6DH    ;5


LATCH1 BIT P2.2
LATCH2 BIT P2.3

```

```

START:CALL SCAN
      JMP START
SCAN:MOV A, #0FEH    ;扫描子程序
      MOV R0, #20H
      SETB C
      MOV R2, #06H
LOOP:RLC A

      MOV P0, A
      SETB LATCH2
      CLR LATCH2
      MOV P0, @R0
      SETB LATCH1
      CLR LATCH1
      INC R0
      CALL DELAY
      DJNZ R2, LOOP
      MOV R2, #07H
      RET
DELAY:MOV R3, #1    ;扫描延时
D1:MOV R4, #2
D2:MOV R5, #248
DJNZ R5, $
DJNZ R4, D2
DJNZ R3, D1
RET
END

```

5. 2C 语言程序结构框架

(1).8 位数码管显示:

```
DataPort= WeiMa[i]; //取位码
```

```
LATCH2=1;      //位锁存
```

```
LATCH2=0;
```

```
DataPort= DuanMa[i]; //取显示数据，段码
```

```
LATCH1=1;      //段锁存
```

```
LATCH1=0;
```

```
Delay(200); //扫描间隙延时，时间太长会闪烁，太短会造成重影
```

```
i++;
```

```
if(8==i)      //检测 8 位扫描完全结束？如扫描完成则从第一个开始再次  
扫描 8 位
```

```
i=0;
```

(2)键盘扫描示例程序:

```
#define KeyPort  P1
```

```
unsigned char KeyScan(void) //键盘扫描函数，使用行列逐级扫描法
```

```
{
```

```
    unsigned char Val;
```

```
    KeyPort=0xf0; //高四位位置高，低四位拉低
```

```
    if(KeyPort!=0xf0) //表示有按键按下
```

```
    {
```

```
        DelayMs(10); //去抖
```

```
        if(KeyPort!=0xf0)
```

```
        {          //表示有按键按下
```

```
            KeyPort=0xfe; //检测第一行
```

```
            if(KeyPort!=0xfe)
```

```
            {
```

```
                Val=KeyPort&0xf0;
```

```
                Val+=0x0e;
```

```

        while(KeyPort!=0xfe);
        DelayMs(10); //去抖
        while(KeyPort!=0xfe);
        return Val;
    }
    KeyPort=0xfd; //检测第二行
    if(KeyPort!=0xfd)
    {
        Val=KeyPort&0xf0;
        Val+=0x0d;
        while(KeyPort!=0xfd);
        DelayMs(10); //去抖
        while(KeyPort!=0xfd);
        return Val;
    }
    KeyPort=0xfb; //检测第三行
    if(KeyPort!=0xfb)
    {
        Val=KeyPort&0xf0;
        Val+=0x0b;
        while(KeyPort!=0xfb);
        DelayMs(10); //去抖
        while(KeyPort!=0xfb);
        return Val;
    }
    KeyPort=0xf7; //检测第四行
    if(KeyPort!=0xf7)
    {
        Val=KeyPort&0xf0;
        Val+=0x07;

```



```

        while(KeyPort!=0xf7);
        DelayMs(10); //去抖
        while(KeyPort!=0xf7);
        return Val;
    }

}

}

return 0xff;
}

/*-----
    按键值处理函数，返回扫描键值
-----*/

unsigned char KeyPro(void)
{
    switch(KeyScan())
    {
        case 0x7e:return 0;break;//0 按下相应的键显示相对应的码值
        case 0x7d:return 1;break;//1
        case 0x7b:return 2;break;//2
        case 0x77:return 3;break;//3
        case 0xbe:return 4;break;//4
        case 0xbd:return 5;break;//5
        case 0xbb:return 6;break;//6
        case 0xb7:return 7;break;//7
        case 0xde:return 8;break;//8
        case 0xdd:return 9;break;//9
        case 0xdb:return 10;break;//a
        case 0xd7:return 11;break;//b
        case 0xee:return 12;break;//c
        case 0xed:return 13;break;//d
    }
}

```

```
case 0xeb: return 14; break; //e
case 0xe7: return 15; break; //f
default: return 0xff; break;
}
}
```

六、 实验报告

- (1) 绘出硬件原理图
- (2) 给出软件流程图
- (3) 写出实验步骤
- (4) 附上带注释的软件源码，并对各模块进行说明
- (5) 总结实验心得

七、 注意事项

- (1) 在用汇编编程时，程序注释的分号是英文输入模式下的分号，如果用中文模式下的分号，程序在编译时会出错；
- (2) 实验成绩=考勤（10%）+实验过程及结果（50%）+实验报告（40%）；