

天津大学

定时-键盘-显示



专 业: 生物医学工程
年 级: 2022 级
班 级: 1 班
姓 名: 张台忍
学 号: 3022202299
邮 箱: ztr8526@gmail.com

2024 年 4 月 26 日

Contents

定时-键盘-显示.....	1
一 实验目的.....	3
二 实验设备.....	3
三 实验内容.....	3
1 建立 <i>Keil</i> 工程.....	3
2 需求分析.....	3
3 硬件电路搭建.....	4
4 代码编写.....	5
5 调试记录.....	8
6 结果与讨论.....	8

一 实验目的

1. 掌握C51编写定时中断函数的方法
2. 掌握矩阵键盘的使用方法
3. 掌握动态数码管的使用方法

二 实验设备

PC 微机一台, *Keil C51*集成开发环境一套, 51 单片机开发仪

三 实验内容

本实验流程图

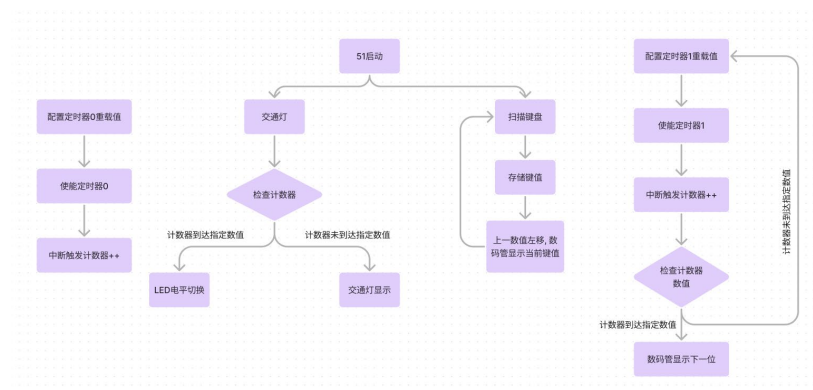


图 0 流程图

1 建立Keil工程

在*Keil*中, 参照实验0的方法, 建立C51工程, 然后新建C语言程序, 编译通过后进行接下来的操作.

2 需求分析

本次实验有两个需求, 描述如下:

1. 实现交通灯模块中4个LED灯分别以0.2s, 0.3s, 0.5s和1s周期闪烁(50%占空比)(时间要求精确, 用定时器中断实现).

2. 当16位矩阵键盘某键按下时，显示其键值于8个动态数码管(每个数码管显示5ms，总刷新率为25Hz)，每按一个键，数字向左移动一位，新的按键总是出现在最右侧。

对于需求1，可通过定义1ms的定时器和定义计数器，每当定时器中断跳出，计数器+1，当计数器数值分别为200,300,500,1000时，对控制交通灯的引脚电平取反，即可实现4个LED灯分别以0.2s,0.3s,0.5s和1s的周期闪烁。

对于需求2，先扫描按键状态，参照矩阵键盘原理图，可以通过先行后列或先列后行来扫描引脚电平状态，来判断是哪个按键按下。此处应注意物理按键的消抖处理。

对于动态数码管，先通过138译码器实现数码管的位选，选择显示数字的位置，再通过实验2提到的对单个数码管进行段选，实现不同数字的显示。由此可知，动态数码管其实每次只能有一个位置显示数字，因此要实现动态效果就要利用人眼视觉暂留，快速的刷新，让人眼看起来是动态显示的。此处要求每个数码管显示5ms，通过定时器来实现。每过5ms，位选移动一位。

由于只有8个数码管，因此可以定义一个长度为8的数组来存储每个数码管要显示的内容。当矩阵键盘按键按下时，将数组中最后一个元素删除并将前面的元素后移，首位赋值为按下的矩阵键盘按键对应的键值，这就可以实现新的按键出现在最右侧并且每按一个键数字都向左移动一位。

3 硬件电路搭建

参考原理图连接硬件电路，连接图如下。

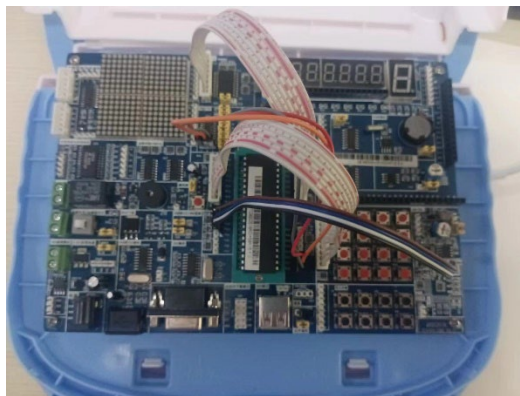


图 1 硬件电路连接

注意连接电路时，应把74HC165模块的JP165上的跳线帽拔下，这里的排针占用单片机引脚P1_7，若不拔下，后续扫描矩阵键盘引脚电平时，涉及到P1_7的4个按键将无效。其他电路正常连接即可。

4 代码编写

完成硬件电路搭建后，即可开始编写代码。

首先编写需求1的代码。使用定时器之前应该先初始化定时器，把定时器打开，并设置初值。代码如下：

```
1. void Timer1_Init(void)           //1 毫秒@12.000MHz
2. {
3.     ET1=1;
4.     EA=1;
5.     PT1=0;                       //打开定时器
6.     TMOD &= 0x0F;               //设置定时器模式
7.     TL1 = 0x20;                 //设置定时初始值
8.     TH1 = 0xD1;                 //设置定时初始值
9.     TF1 = 0;                    //清除 TF1 标志
10.    TR1 = 1;                     //定时器 1 开始计时
11. }
```

该代码可通过stc – isp直接生成，生成方式如图所示：

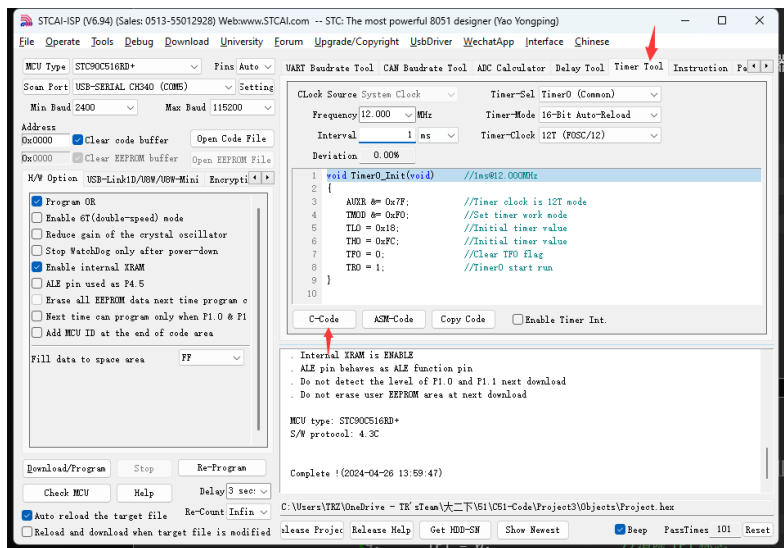


图 2 使用软件自动生成定时器初始化代码

该软件中Timer Tool选项中，可以选择晶振频率，定时器号，定时器模式等，并直接生成C语言代码或汇编代码。

初始化完成后，即可在main.c文件中编写中断函数，代码如下。

```
1. unsigned int count0=0;
2. void Timer0_LED() interrupt 1
```

《单片机原理与医学应用》课程作业

```
3. {
4.     TL0 = 0x18;           //设置定时初始值
5.     TH0 = 0xFC;          //设置定时初始值
6.     count0++;             //计数器+1
7.     if(count0%200==0)     //计数器 200 时，即 200ms 时，LED 电平反转
8.         P3_0=~P3_0;
9.     if(count0%300==0)     //计数器 300 时，即 300ms 时，LED 电平反转
10.        P3_1=~P3_1;
11.    if(count0%500==0)      //计数器 500 时，即 500ms 时，LED 电平反转
12.        P3_2=~P3_2;
13.    if(count0%1000==0)     //计数器 1000 时，即 1000ms 时，LED 电平反转
14.        P3_3=~P3_3;
15.    if(count0==3000)       //计数器 3000 时，计数器归零，防止后续益处风险
16.        count0=0;
17. }
```

这就实现了4个LED以不同的周期闪烁.

接下来是编写需求2的代码.

需求2要用到矩阵键盘和动态数码管，首先实现矩阵键盘的按键扫描. 观察原理图可知，可以通过先后扫描两个引脚的电平，来确定是哪个按键按下. 以下是代码实现.

```
1. int MatrixKey()
2. {
3.     int KeyNumber=0;
4.
5.     P1=0xFF; // 1111 1111 全部置高电平默认
6.     P1_7=0; // 矩阵按键第四列扫描
7.     Delay(5);
8.     if(P1_3==0){Delay(20);while(P1_3==0);Delay(20);KeyNumber=16;}
9.     if(P1_2==0){Delay(20);while(P1_2==0);Delay(20);KeyNumber=12;}
10.    if(P1_1==0){Delay(20);while(P1_1==0);Delay(20);KeyNumber=8;}
11.    if(P1_0==0){Delay(20);while(P1_0==0);Delay(20);KeyNumber=4;}
12.
13.    P1=0xFF;
14.    P1_6=0; // 矩阵按键第三列扫描
15.    Delay(5);
16.    if(P1_3==0){Delay(20);while(P1_3==0);Delay(20);KeyNumber=15;}
17.    if(P1_2==0){Delay(20);while(P1_2==0);Delay(20);KeyNumber=11;}
18.    if(P1_1==0){Delay(20);while(P1_1==0);Delay(20);KeyNumber=7;}
19.    if(P1_0==0){Delay(20);while(P1_0==0);Delay(20);KeyNumber=3;}
20.
21.    P1=0xFF;
22.    P1_5=0; // 矩阵按键第二列扫描
23.    Delay(20);
24.    if(P1_3==0){Delay(20);while(P1_3==0);Delay(20);KeyNumber=14;}
25.    if(P1_2==0){Delay(20);while(P1_2==0);Delay(20);KeyNumber=10;}
26.    if(P1_1==0){Delay(20);while(P1_1==0);Delay(20);KeyNumber=6;}
27.    if(P1_0==0){Delay(20);while(P1_0==0);Delay(20);KeyNumber=2;}
28.
29.    P1=0xFF;
30.    P1_4=0; // 矩阵按键第一列扫描
31.    Delay(20);
32.    if(P1_3==0){Delay(20);while(P1_3==0);Delay(20);KeyNumber=13;}
33.    if(P1_2==0){Delay(20);while(P1_2==0);Delay(20);KeyNumber=9;}
34.    if(P1_1==0){Delay(20);while(P1_1==0);Delay(20);KeyNumber=5;}
```

《单片机原理与医学应用》课程作业

```
35.     if(P1_0==0){Delay(20);while(P1_0==0);Delay(20);KeyNumber=1;}
36.
37.     return KeyNumber;
38. }
```

代码中先扫描每一列，在扫描每一行，即可确定按键位置。

另外，需求2还需要用到数码管，动态数码管的显示需要先位选，再段选。位选通过给138译码器赋值，转化为八位的数据，可以选出位置。要实现动态显示，就要快速的刷新，可通过定时器，到一定时间，显示下一个数码管，由于视觉暂留，看起来就是动态的。同时，数码管显示会有余晖，要消影处理。消影可以将段选电平置0实现。代码实现如下。

```
1. unsigned int count1=0;
2. void Timer1() interrupt 3
3. {
4.     TL1 = 0x18;           //设置定时初始值
5.     TH1 = 0xFC;           //设置定时初始值
6.     count1++;
7.     if(count1==5)         //5ms 位移一次
8.     {
9.         count1=0;         //把计数器归零
10.        P2--;              //数码管显示下一位
11.        P0=0;              //数码管消影
12.        P0=show[i];        //不同位置显示不同数字
13.        i++;
14.        if(i>7)
15.            i=0;           //防止溢出
16.        if(P2<0)
17.            P2=0x07;       //防止溢出
18.    }
19. }
```

实现动态显示后，就要实现按键检测并且更新显示内容。将按键检测代码放入主循环中。代码如下。

```
1. char dig[]={0x3f, 0x06, 0x5b, 0x4f,0x66, 0x6d, 0x7d, 0x07,0x7f, 0x6f, 0x7
  7, 0x7c,0x39, 0x5e, 0x79, 0x71}; //按键键值
2. char show[]={0,0,0,0,0,0,0,0}; //每个数码管要显示的内容
3. void main()
4. {
5.     //Timer0_Init();       //初始化定时器 0
6.     Timer1_Init();         //初始化定时器 1
7.     P2=0x00; //初始化译码器
8.     while(1)
9.     {
10.        char key=MatrixKey(); //扫描按键
11.        if(key)
12.        {
13.            int tmp=7;
14.            while(tmp>=0)
15.            {
16.                show[tmp]=show[tmp-1];
17.                tmp--;
18.            } //将数组内容全部后移
19.            show[0]=dig[key-1]; //将最新的按键更新到数组中
```

```
20.      }  
21.      }  
22. }
```

这样就实现了需求2.

5 调试记录

调试过程中，出现了含引脚`P1_7`的那一列按键失效的情况。首先检查了代码，发现代码不存在问题。于是去原理图中搜索引脚`P17`，发现在模块`74HC165`的`JP165`排线处，占用了`P1_7`，于是将硬件电路中此处的跳线帽拔下，重新装载程序，就解决了该问题。

在仿真过程中，出现了`CPU`过载的情况，这是由于检测引脚电平过快导致的，在代码中加入适当的延时函数即可解决。

6 结果与讨论

本次实验实现了定时器的使用，矩阵键盘控制动态数码管。进一步加深了对中断的理解，以及优化了单片机程序设计思路。