深 圳 大 学 实 验 报 告

课程名称：微型计算机技术

实验项目名称：定时器的使用

学院：医学院

专业：生物医学工程

指导教师：徐海华、刘昕宇

报告人：陈焕鑫 学号：2016222042 班级：生工 2 班

实验时间：2018-11-6

实验报告提交时间：2018-11-20

教务处制

一、实验目的

1．定时器是单片机中最重要的功能之一，了解溢出式传统定时器的使用方法。

二、实验仪器

微机原理实验板

三、实验内容

1、编写给予定时器的精确延时函数

延时函数使用定时器 0 编写，延时值使用 10ms，而具体延时多少个 10ms，则可由参数传递给延时函数。

2、使用定时器延时函数重写跑马灯程序

将以前使用 for 循环进行延时的跑马灯程序,改为使用定时器的延时方法进行运行。

3、编写基于定时器中断的数码管控制程序

使用定时器 0 中断，每进一次中断，就让数码管显示的数字加 1。通过调整定时器中断的时间，即对数码管显示的速度进行控制。

四、实验原理

1、首先是 51 单片机计数器的脉冲输入引脚。主要的脉冲输入脚有 Px,y， 也指对应 T0 的 P3.4 和对应 T1 的 P3.5，主要用来检测片外来的脉冲。而引脚 18 和 19 则对应着晶振的输入脉冲，脉冲的频率和周期为

F = f/12 = 11.0592M/12 = 0.9216MHZ　　T = 1/F = 1.085us

2、定时器有两种工作模式，分别为计数模式和定时模式。对 Px,y 的输入脉冲进行计数为计数模式。定时模式，则是对 MCU 的主时钟经过 12 分频后计数。因为主时钟是相对稳定的，所以可以通过计数值推算出计数所经过的时间。

3、51 计数器的计数值存放于特殊功能寄存器中。T0(TL0-0x8A, TH0-0x8C), T1(TL1-0x8B, TH1-0x8D)

4、TLx 与 THx 之间的搭配关系

1)、TLx 与 THx 之间 32 进制。即当 TLx 计到 32 个脉冲时，TLx 归 0 同时 THx 进 1。

这也称为方式 0。

2)、TLx 与 THx 之间 256 进制。即当 TLx 计到 256 个脉冲时，TLx 归 0 同时 THx 进 1。这也称为方式 1。在方式 1 时，最多计 65536 个脉冲产生溢出。在主频为 11.0592M 时，每计一个脉冲为 1.085us，所以溢出一次的时间为 1.085usx65536＝71.1ms。

3)、THx 用于存放 TLx 溢出后，TLx 下次计数的起点。这也称为方式 2。

4)、THx 与 TLx 分别独立对自己的输入脉冲计数。这也称为方式 3。

5、定时器初始化

1)、确定定时器的计数模式。

2)、确定 TLx 与 THx 之间的搭配关系。

3)、确定计数起点值。即 TLx 与 THx 的初值。

4)、是否开始计数。TRx

（1）和（2）可以由工作方式寄存器 TMOD 来设定，TMOD 用于设置定时/计数器的工作方式，低四位用于 T0，高四位用于 T1。其格式如下：

GATE：门控位，用于设置计数器计数与否，是否受 P3.2 或 P3.3 电压状态的影响。GATE＝0 时，表示计数器计数与否与两端口电压状态无关；GATA＝1 时，计数器是否计数要参考引脚的状态，即 P3.2 为高时 T0 才计数，P3.3 为高时 T1 才计数。

C/T:定时/计数模式选择位。　　＝0 为定时模式；　　=1 为计数模式。

M1M0：工作方式设置位。定时/计数器有四种工作方式，由 M1M0 进行设置。

6、计数器的溢出

计数器溢出后，THx 与 TLx 都归 0。并将特殊功能区中对应的溢出标志位 TFx 写为 1

7、计数的起点和终点

计数器溢出的终点是固定为 65535 这个值的，因此，只能通过修改计数的起点来控制溢出的时长。

五、实验方法及步骤

定时器计数周期为 1.085us，因此，要延时 10ms，需要计数(10000/1.085=)9216 次。定时器的溢出终点固定为 65535，因此需要设置定时的起点为(65535+1-9216=)56320，十六进制为 0xDC00。

首先，打开 Keil 软件，新建工程名为 Lab7Prj，在工程中添加空的 main.c 文件。

然后，对 main.c 文件进行编辑。结合原理图，我们对按键的引脚进行定义。

```
LED1 BIT P2.4   ;设置 LED1 的引脚为 P2.4
```

```
LED2 BIT P2.5   ;设置 LED2 的引脚为 P2.5
LED3 BIT P2.6   ;设置 LED3 的引脚为 P2.6
LED4 BIT P2.7   ;设置 LED4 的引脚为 P2.7
BEEP BIT P1.0   ;设置蜂鸣器的引脚为 P1.0
```

程序代码如下：

第二题：将以前使用 for 循环进行延时的跑马灯程序，改为使用定时器的延时方法进行运行。

```
#include "STC12C5A60S2.h"
sbit LED1 = P2^4;     //定义 LED1 的引脚 P2.4
sbit LED2 = P2^5;   //定义 LED2 的引脚 P2.5
sbit LED3 = P2^6;   //定义 LED3 的引脚 P2.6
sbit LED4 = P2^7;   //定义 LED4 的引脚 P2.7

void INIT_TIMER()   //初始化定时器
{
  TMOD = 0x01;      //设置定时器模式为 16 位定时器
}

void TIM_Delay10Ms(unsigned char cnt) //延时 10ms 函数
{
  TH0 = 0xDC;    //定时器 0 的高八位
  TL0 = 0x00;    //定时器 0 的低八位
  TR0 = 1;       //开始计时

  while(cnt != 0)       //当倒计时还未到 0 时
  {
    if(TF0)          //如果定时器 0 溢出
    {
      TF0 = 0;      //清楚标志
      TH0 = 0xDC;   //定时器 0 的高八位
      TL0 = 0x00;     //定时器 0 的低八位
      TR0 = 1;      //开始计时
      cnt--;        //倒计时递减
    }
  }
}

void LED_RUN()
```

```
{
  static unsigned char status = 0;  //函数内部静态变量用来存放状态

  switch(status)
  {
    case 0:
      LED1 = ~LED1;  //LED1 取反
      break;
    case 1:
      LED2 = ~LED2;  //LED2 取反
      break;
    case 2:
      LED3 = ~LED3;  //LED3 取反
      break;
    case 3:
      LED4 = ~LED4;    //LED4 取反
      break;
  }

  status++;     //状态加 1

  if(status > 3)  //如果状态溢出
  {
    status = 0; //重置状态
  }
}

void main()
{
  INIT_TIMER();  //初始化定时器

  while(1)       //死循环
  {
    TIM_Delay10Ms(100); //延时 1s
    LED_RUN();         //调用流水灯函数
  }
}
```

第三题：

使用定时器 0 中断，每进一次中断，就让数码管显示的数字加 1。通过调整定时器中断的时间，即对数码管显示的速度进行控制。

```
#include "STC12C5A60S2.h"
```

```
sbit SM_G4 = P2^3;

unsigned char Seg7Disp[10] = {0x03, 0x9F, 0x25, 0x0d, 0x99, 0x49, 0x41,
0x1F, 0x01, 0x09};

void INIT_SEG7()
{
  SM_G4 = 0;        //打开最后一个数码管显示
  P0 = 0xFF;      //数码管熄灭
}
void INIT_TIMER()   //初始化定时器
{



  EA = 1;     //打开中断允许总控制
  ET0 = 1;      //打开定时器 0 中断
  TMOD = 0x01;    //设置定时器模式为 16 位定时器
  TF0 = 0;    //清楚标志
  TH0 = 0xDC;   //定时器 0 的高八位
  TL0 = 0x00;     //定时器 0 的低八位
  TR0 = 1;      //开始计时
}

void Timer0() interrupt 1         //定时器 0 中断
{
  static unsigned char counter = 0;   //静态变量 counter
  static unsigned char i = 0;       //静态变量 i

  TF0 = 0;               //清楚标志
  TH0 = 0xDC;             //定时器 0 的高八位
  TL0 = 0x00;               //定时器 0 的低八位
  TR0 = 1;             //开始计时

  counter++;               //计时器加 1
  if(counter >= 100)            //计数器加到 100
  {
    P0 = Seg7Disp[i];         //点亮七段数码管
    i++;               //i 递增
    if(i >= 10)             //当 i 大于或等于 10 的时候
    {
      i = 0;             //将 i 清零
    }
    counter = 0;             //计数器 counter 清零
```

```
  }
}

void main()       //主函数
{
  INIT_SEG7();    //初始化七段数码管
  INIT_TIMER();   //初始化定时器

  while(1)        //死循环
  {

  }
}
```

检查代码无误之后，分别编译、链接、生成 Hex 文件，将 Hex 文件通过串口烧进实验平台中，观察实验现象。

六、实验现象

（1） 四个 LED 灯以跑马灯的形式循环点亮

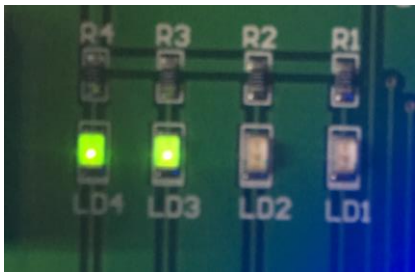

图 6-1

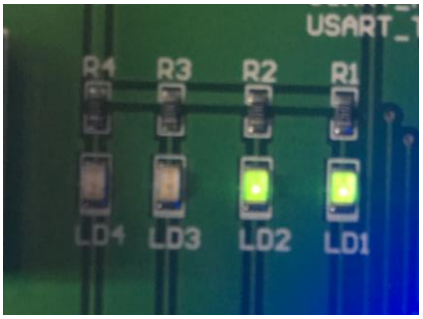

图 6-2

（2）第四个数码管以 1s 为间隔不断改变显示的数字，可以从 0 一直显示到 9 不断循环。

图 6-3 数码管显示‘0’

图 6-4 数码管显示‘9’

七、实验结论

通过这次实验的学习，我了解了定时器的使用方法，能够使用定时器的标志位写延时函数，也能够通过定时器中断来完成数码管显示的转变。

指导教师批阅意见: 成绩评定: 指导教师签字: 年　　月　　日
备注:

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。
2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。