学院

单片机原理及应用 实验报告

|  |  |
| --- | --- |
| 成绩 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 专业班级 |  | 实验日期 |  |
| 姓 名 |  | 学 号 |  |
| 实验名称 | 定时器/计数器应用  ------用定时器实现电子秒表 | 指导教师 |  |

（报告内容包括实验目的、实验设备及器材、实验内容和要求、实验步骤、实验小结等）

1. **实验目的**
2. 熟悉单片机内部定时器的使用和编程方法；
3. 熟悉定时器初值及定时器的计算；
4. 掌握中断处理程序的编程方法。
5. **实验内容及原理**

1、定时器/计数器组成原理图

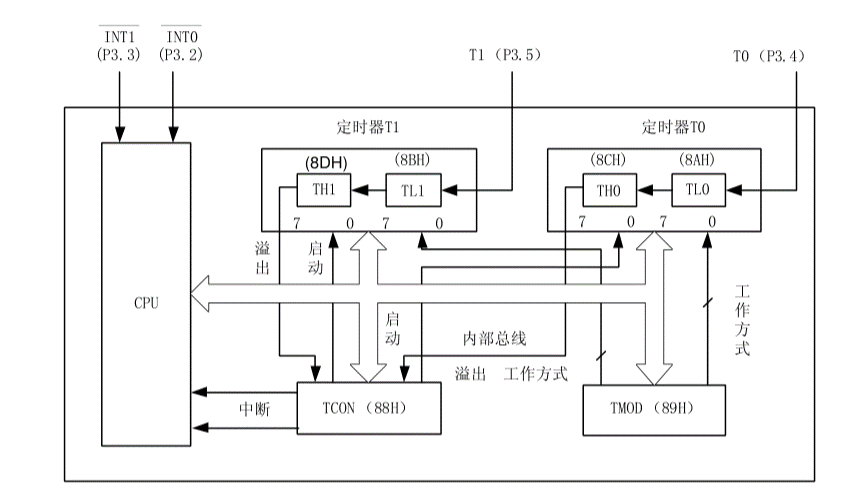


图1 定时器/计数器内部结构框图

51单片机内部有两个16位定时器/计数器，简称为T0和T1，可以独立使用。其中，TH1(高八位)、TL1（低八位）是T1的计数器，TH0(高八位)、TL0（低八位）是T0的计数器.他们的工作状态及工作方式由两个特殊功能寄存器TMOD和TCON的各位的状态或设置来决定。

2、定时器/计数器的控制

1）定时器/计数器工作模式寄存器TMOD（89H）

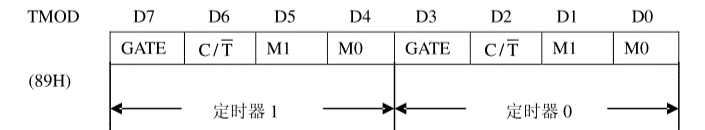


图2 TMOD的各位定义

2）定时器/计数器控制寄存器TCON（88H）

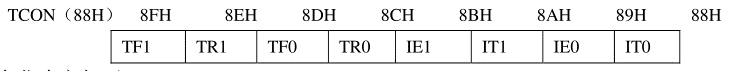


图3 TCON的各位定义

3、定时器/计数器的初始化

1）确定工作方式：对TMOD赋值。

2）预置定时/计数的初值。

4、定时器延时可以使用查询方式实现，也可以使用中断方式实现。与串口中断有关的中断允许控制寄存器IE：

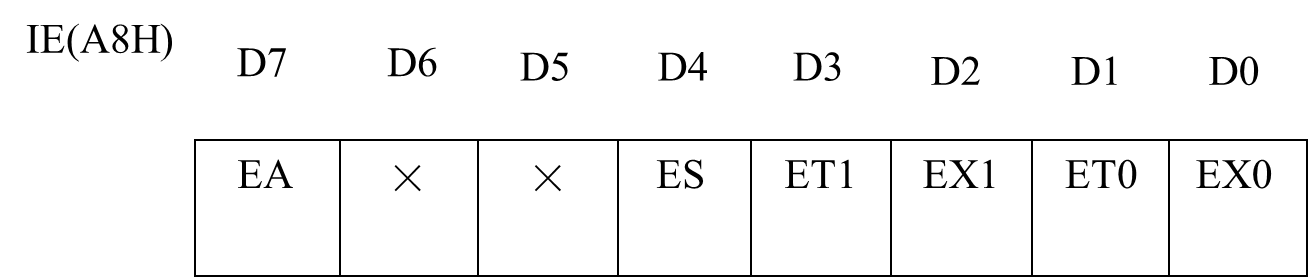


图4 中断允许寄存器IE的格式

1. **实验要求**

1、设计两位电子秒表，在00~99间循环；

2、用定时器T0产生周期为1s的方波，送P1.0，控制LED闪烁；

3、两位十进制计数每隔1s加1，计数结果送两位数码管显示；

4、按钮用作意外情况紧急报警。按钮接外部中断0，当按下按钮时，数码管闪烁显示“E0”，闪烁五次，然后退出报警，继续正常1s计数。

1. **实验源代码**

#include<reg52.h>  
  
sbit DULA = P3^0;  
sbit WELA = P3^1;  
sbit flag = P1^0;  
  
unsigned char flag\_int0 = 0;  
unsigned char flag\_time0 = 0;  
  
//7led显示字形码  
unsigned char wordlist[] = {0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71,0x00};  
//led灯位  
unsigned char ledlist[] = {0xfe,0xfd,0xfb,0xf7,0xef,0xdf,0xbf,0x7f};  
//行列式键盘键值  
unsigned char key\_buf[] = {0xee, 0xde, 0xbe, 0x7e,0xed, 0xdd, 0xbd, 0x7d,0xeb, 0xdb, 0xbb, 0x7b,0xe7, 0xd7, 0xb7, 0x77};  
//时间码  
unsigned char time\_index[] = {0,0};  
unsigned char time\_list[] = {0x00,0x00};  
  
void delay(unsigned int i) //延迟函数  
{  
 unsigned int j;  
 unsigned char k;  
 for(j=i;j>0;j--)  
 {  
 for(k=125;k>0;k--);  
 }  
}  
  
void display(unsigned char display\_list[],unsigned int time)  
{  
 unsigned char i;  
 //for(i=0;\*(display\_list+i);i++)  
 for(i=0;i<2;i++)//2对应led显示位数  
 {  
 //选择显示位  
 P0=\*(ledlist+i);  
 WELA=1;  
 WELA=0;  
 //输出字符码  
 P0=\*(display\_list+i);  
 DULA=1;  
 DULA=0;  
 delay(time);  
 }  
}  
  
void service\_int0(void) interrupt 0//外部中断0中断服务函数  
{  
 flag\_int0 = 1;  
}  
  
void service\_time0(void) interrupt 1//定时器中断0中断服务函数  
{  
 flag\_time0++;  
}  
  
void main()  
{  
 /\*中断服务初始化\*/  
 IE = 0x83; //10000011|中断服务  
 TCON = 0x11; //00010001|中断触发方式  
 TMOD = 0X1; //00000001|定时器工作方式  
 TH0=0Xb0;//定时器装载初值  
 TH1=0x3c;//50ms定时器  
 P1=0xff; //复位关闭全部LED  
 P2=0x0f; //键盘待触发状态  
 while (1)  
 {  
 if(flag\_int0)//紧急情况中断  
 {  
 unsigned char i = 0;  
 for(i=0;i<5;i++)  
 {  
 unsigned char j;  
 for (j = 0; j < 90; j++)  
 {  
 unsigned char display\_list[] = {0x79,0x3f};  
 display(display\_list,10);  
 }  
 for (j = 0; j < 90; j++)  
 {  
 unsigned char display\_list[] = {0x00,0x00};  
 display(display\_list,10);  
 }  
 }  
 flag\_int0 = 0;  
 }  
 if(flag\_time0>=13)//定时器中断服务  
 {  
 while(!flag);  
 time\_index[1] = time\_index[1]+1;  
 if(!(time\_index[1]%10))  
 {  
 time\_index[1] = time\_index[1]%10;  
 time\_index[0] = (time\_index[0]+1)%10;  
 }  
 flag\_time0=0;  
 time\_list[0] = wordlist[time\_index[0]];  
 time\_list[1] = wordlist[time\_index[1]];  
 }  
 display(time\_list,20);  
 }  
}

1. **实验总结**

程序使用中断方法实现，有效提高CPU的利用率。在使用中断方法时，要重点主要中断服务相关寄存器的初始化配置。对于使用定时器中断服务，我的总结出配置寄存器顺序（技巧）：

1. 设置IE寄存器，打开中断允许总开关、定时器中断允许
2. 设置TCON寄存器，打开定时器运行控制
3. 设置TMOD寄存器，设置定时器运行方式
4. 对定时器加载初值