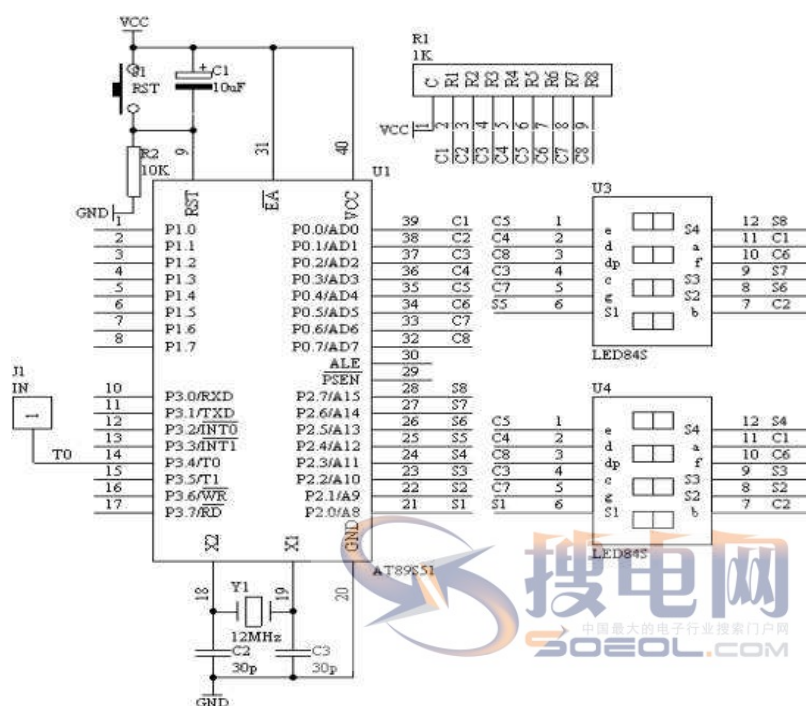


51 单片机频率计数器课程设计

1. 实验任务

利用 AT89S51 单片机的 T0、T1 的定时计数器功能，来完成对输入的信号进行频率计数，计数的频率结果通过 8 位动态数码管显示出来。要求能够对 0—250KHZ 的信号频率进行准确计数，计数误差不超过±1HZ。

2. 电路原理图



3. 系统板上硬件连线

(1). 把“单片机系统”区域中的 P0.0—P0.7 与“动态数码显示”区域中的 ABCDEFGH 端口用 8 芯排线连接。

(2). 把“单片机系统”区域中的 P2.0—P2.7 与“动态数码显示”区域中的 S1S2S3S4S5S6S7S8 端口用 8 芯排线连接。

(3). 把“单片机系统”区域中的 P3.4 (T0) 端子用导线连接到“频率产生器”区域中的 WAVE 端子上。

4. 程序设计内容

(1). 定时/计数器 T0 和 T1 的工作方式设置, 由图可知, T0 是工作在计数状态下, 对输入的频率信号进行计数, 但对工作在计数状态下的 T0, 最大计数值为 $f_{OSC}/24$, 由于 $f_{OSC}=12\text{MHz}$, 因此: T0 的最大计数频率为 250KHz 。对于频率的概念就是在一秒只数脉冲的个数, 即为频率值。所以 T1 工作在定时状态下, 每定时 1 秒中到, 就停止 T0 的计数, 而从 T0 的计数单元中读取计数的数值, 然后进行数据处理。送到数码管显示出来。

(2). T1 工作在定时状态下, 最大定时时间为 65ms , 达不到 1 秒的定时, 所以采用定时 50ms , 共定时 20 次, 即可完成 1 秒的定时功能。

5. C 语言源程序

```
#include <AT89X52.H>
```

```
unsigned char code dispbit[]={0xfe,0xfd,0xfb,0xf7,0xef,0xdf,0xbf,0x7f};
```

```
unsigned char code dispcode[]={0x3f,0x06,0x5b,0x4f,0x66,
```

```
0x6d,0x7d,0x07,0x7f,0x6f,0x00,0x40};
```

```
unsigned char dispbuf[8]={0,0,0,0,0,10,10};
```

```
unsigned char temp[8];
```

```
unsigned char dispcount;
```

```
unsigned char T0count;
```

```
unsigned char timecount;
```

```
bit flag;
```

```
unsigned long x;
```

```
void main(void)
```

```
{
```

```
    unsigned char i;
```

```
    TMOD=0x15;
```

```
    TH0=0;
```

```
    TL0=0;
```

```
    TH1=(65536-4000)/256;
```

```
    TL1=(65536-4000)%256;
```

```
    TR1=1;
```

```
    TR0=1;
```

```
    ET0=1;
```

```
    ET1=1;
```



```
EA=1;
```

```
while(1)
```

```
{
```

```
    if(flag==1)
```

```
    {
```

```
        flag=0;
```

```
        x=T0count*65536+TH0*256+TL0;
```

```
        for(i=0;i<8;i++)
```

```
        {
```

```
            temp[i]=0;
```

```
        }
```

```
        i=0;
```

```
        while(x/10)
```

```
        {
```



```
temp[i]=x%10;

x=x/10;

i++;

}

temp[i]=x;

for(i=0;i<6;i++)

{

    dispbuf[i]=temp[i];

}

timecount=0;

T0count=0;

TH0=0;

TL0=0;

TR0=1;

}
```



```
    }  
  
}  
  
void t0(void) interrupt 1 using 0
```

```
{  
  
    T0count++;  
  
}
```

```
void t1(void) interrupt 3 using 0
```

```
{  
  
    TH1=(65536-4000)/256;  
    TL1=(65536-4000)%256;
```

```
    timecount++;
```

```
    if(timecount==250)
```

```
    {
```

```
        TR0=0;
```

```
        timecount=0;
```



```
        flag=1;

    }

    P0=dispcode[dispbuf[dispcount]];

    P2=dispbit[dispcount];

    dispcount++;

    if(dispcount==8)

    {

        dispcount=0;

    }

}
```

