

第 6 章

习题

6-4

答：采样周期定的太小，会使转换器的低频信号的测量范围慢小。因此，V—F 转换器这类问题要结合应用实际，选择最合适的采样周期值。

6-5

解 1：方式 0

确定装入 T0 的常数。计数器每机器周期计数值加 1，机器周期为：

$$\frac{12(\text{分频})}{12\text{MHz}} = 1\mu\text{s}$$

设所需初值为 x，则 $(2^{13} - x) \times 1 \times 10^{-6} = 1 \times 10^{-3}$

即： $x = 2^{13} - 10^3 = 7192 = 1\text{C18H} = 1\ 1100\ 0001\ 1000\text{B}$

根据 13 位定时器的特性，TH0 放 x 的高 8 位，TL0 放低 5 位，我们将 x 值重排如下：**x = 1110 0000 1 1000**，即得本任务的时间常数为：TH0 = E0H，TL0 = 18H。

```
ORG      0000H
AJMP     MAIN
ORG      000BH
LJMP     TIME0INT
ORG      0050H
MAIN:    MOV     SP, #60H
MOV      TMOD, #00H
MOV      TL0, #18H
MOV      TH0, #0E0H
SETB     ET0                      ; 允许 T0 中断
SETB     EA                      ; 开放总中断
SETB     TR0                      ; 启动 T1 定时
SJMP     $
TIME0INT: MOV     TL0, #18H
MOV      TH0, #0E0H
RETI                                ; 中断返回
END
```

方式 1

确定装入 T0 的常数。计数器每机器周期计数值加 1，机器周期为：

$$\frac{12(\text{分频})}{12\text{MHz}} = 1\mu\text{s}$$

设所需初值为 x，则 $(2^{16} - x) \times 1 \times 10^{-6} = 1 \times 10^{-3}$

即： $x = 2^{16} - 10^3 = 64536 = \text{FC18H}$

即得本任务的时间常数为：TH0 = FCH，TL0 = 18H。

```
ORG      0000H
```

```

        AJMP      MAIN
        ORG       000BH
        LJMP      TIME0INT
        ORG       0050H
MAIN:    MOV       SP, #60H
        MOV       TMOD, #01H
        MOV       TL0, #18H
        MOV       TH0, #0F0H
        SETB      ET0                ; 允许 T0 中断
        SETB      EA                ; 开放总中断
        SETB      TR0                ; 启动 T1 定时
        SJMP      $
TIME0INT: CLR      TR0                ; 禁止 T0
        CLR      ET0
        RETI                     ; 中断返回
        END

```

方式 2

确定装入 T0 的常数。计数器每机器周期计数值加 1，机器周期为：

$$\frac{12(\text{分频})}{12\text{MHz}} = 1\mu\text{s}$$

方式 2 为 8 位常数自动重装，每次溢出最大值为 256 μs，为编程方便，又减少中断次数，采用 4 次 250 μs 的方法实现本任务。其时间常数为：TH0 = 06H，TL0 = 06H。

```

        ORG       0000H
        AJMP      MAIN
        ORG       000BH
        LJMP      TIME0INT
        ORG       0050H
MAIN:    MOV       SP, #60H
        MOV       R7, #4
        MOV       TMOD, #02H
        MOV       TL0, #06H
        MOV       TH0, #06H
        SETB      ET0                ; 允许 T0 中断
        SETB      EA                ; 开放总中断
        SETB      TR0                ; 启动 T1 定时
        SJMP      $
TIME0INT: DJNZ     R7, RETURN
        CLR      TR0                ; 禁止 T0
        CLR      ET0
RETURN:  RETI                     ; 中断返回
        END

```

解 2：方式 0

```
#include<reg52a.h>
```

```
void timer0() interrupt 1
```

```
//T0 中断周期 1ms
```

```
{ TL0=0x18;TH0=0xe0;}
```

```

void main()
{
    TMOD = 0x00;           // T0 工作于方式 0 定时
    TL0=0x18;TH0=0xe0;
    ET0 =1; EA = 1;  TR0 = 1;
    while (1);             //等待中断
}

    方式 1
#include<reg52a.h>
void timer0() interrupt 1  //T0 中断周期 1ms
{ ET0 =1;TR0 = 1; }
void main()
{
    TMOD = 0x01;           // T0 工作于方式 1 定时
    TL0=0x18;TH0=0xf0;     //时间常数
    ET0 =1;
    EA = 1;
    TR0 = 1;               //开中断并启动定时
    while (1);             //等待中断
}

    方式 2
#include<reg52a.h>
#define uchar unsigned char
uchar data intcount=4;
void timer0() interrupt 1  //T0 中断周期 1ms
{
    intcount -- ;
    if (intcount == 0)
    {
        ET0 =0;
        TR0 = 0;
    }
}
void main()
{
    TMOD = 0x02;           // T0 工作于方式 2 定时
    TL0=0x06;TH0=0x06;     //时间常数
    ET0 =1;
    EA = 1;
    TR0 = 1;               //开中断并启动定时
    while (1);             //等待中断
}

```

6-6

解 1: 方式 0

13 位计数初值 (8192-计数值) 高 8 位为 NH, 低 8 位为 NL (高 5 位补 0)。

```

        ORG      0000H
        AJMP     MAIN
        ORG      001BH
        LJMP     TIME1INT
        ORG      0050H
MAIN:    MOV      SP, #60H
        MOV      TMOD, #40H
        MOV      TL1, #NHH
        MOV      TH1, #NLH
        SETB     ET1           ; 允许 T1 中断
        SETB     EA           ; 开放总中断
        SETB     TR1          ; 启动 T1 定时
        SJMP     $
TIME1INT: CLR      TR1         ; 禁止 T1
        CLR      ET1
        RETI          ; 中断返回
        END

```

方式 1

16 位计数初值（65536-计数值）高 8 位为 NH，低 8 位为 NL。

```

        ORG      0000H
        AJMP     MAIN
        ORG      001BH
        LJMP     TIME1INT
        ORG      0050H
MAIN:    MOV      SP, #60H
        SETB     F0           ; 主程序下达停止命令标志：1 计数，2 停止
        MOV      TMOD, #50H
        MOV      TL1, #NLH
        MOV      TH1, #NHH
        SETB     ET1           ; 允许 T0 中断
        SETB     EA           ; 开放总中断
        SETB     TR1          ; 启动 T1 定时
        SJMP     $
TIME1INT: JNB     F0, STOPT1
        MOV      TL1, #NHH
        MOV      TH1, #NLH
        SJMP     RETURN
STOPT1:  CLR      F0
        CLR      TR1         ; 禁止 T1
        CLR      ET1
RETURN:  RETI          ; 中断返回
        END

```

方式 2

方式 2 为 8 位计数（少用），计数值在 TL1 中设计数初值为 NL。

```

                ORG      0000H
                AJMP     MAIN
                ORG      001BH
                LJMP     TIME1INT
                ORG      0050H
MAIN:           MOV      SP, #60H
                MOV      TMOD, #60H
                MOV      TL1, #NLH
                MOV      TH1, #NLH
                SETB     ET1                ; 允许 T0 中断
                SETB     EA                ; 开放总中断
                SETB     TR1                ; 启动 T1 定时
                SJMP     $
TIME1INT:       CLR      TR1                ; 禁止 T1
                CLR      ET1
                RETI                ; 中断返回
                END
```

解 2: 方式 0

```
#include<reg52a.h>
```

```
void timer1() interrupt 3                //T1 中断服务程序
```

```
{ ET1 = 0;TR1 = 0; }
```

```
void main()
```

```
{
```

```
    TMOD = 0x40;                // T1 工作于方式 0 计数
```

```
    TL1 = 0xNL;TH1 = 0xNH;      //计数常数
```

```
    ET1 = 1;
```

```
    EA = 1;
```

```
    TR1 = 1;                    //开中断并启动定时
```

```
    while (1);                 //等待中断
```

```
}
```

解 2: 方式 1

```
#include<reg52a.h>
```

```
bit end_of_count                //主程序下达停止命令标志：1 计数，2 停止
```

```
void timer1() interrupt 3        //T1 中断服务程序
```

```
{
```

```
    if (end_of_count ==0)
```

```
    {
```

```
        ET1 = 0;
```

```
        TR1 = 0;
```

```
    }
```

```
    else
```

```

    {
        TL1 = 0xNL;
        TH1 = 0xNH;                //重装计数常数
    }
}
void main()
{
    TMOD = 0x50;                    // T1 工作于方式 1 计数
    TL1 = 0xNL;TH1 = 0xNH;         //计数常数
    ET1 = 1;
    EA = 1;
    TR1 = 1;                        //开中断并启动定时
    end_of_count 成协议= 1;
    while (1);                      //等待中断
}
方式 2
#include<reg52a.h>
void timer1() interrupt 3          //T1 中断服务程序
{ ET1 = 0;TR1 = 0; }
void main()
{
    TMOD = 0x60;                    // T1 工作于方式 6 计数
    TL1 = 0xNL;TH1 = 0xNH;         //计数常数
    ET1 = 1;
    EA = 1;
    TR1 = 1;                        //开中断并启动定时
    while (1);                      //等待中断
}

```

方式 0、1、2 的最大计数值分别为 8192、65536 和 256。

6-7

解：产生方波仍由定时器 T1 承担，而外部中断信号直接输入 $\overline{INT0}$ 。参考程序如下

```

ORG      0000H
AJMP     MAIN
ORG      0003H                ; 外部中断 0
LJMP     INT0P
ORG      001BH                ; 定时器 1
LJMP     IT1P
ORG      0050H
MAIN:    MOV     SP, #60H
         LCALL   PTM
STOP:    SJMP     STOP
PTM:     MOV     TMOD, #20H
         MOV     TL1, #9CH
         MOV     TH1, #9CH
         SETB    EX0            ; 外部中断 0 中断开放
         SETB    ET1

```

```

        SETB     EA
        RET
INT0P:  CPL      TR1          ; 核心指令
        RETI
IT1P:   CPL      P1.0
        RETI
        END

```

6-1 训练题

解：参考程序如下

```

        ORG      0000H
        AJMP     MAIN
        ORG      0050H
MAIN:    MOV      SP, #60H
        ACALL    PTM
TESTTF1: JNB      TF1, TESTTF1
        CLR      TF1          ; 软件清除 T0 溢出标志
        CPL      P1.0         ; P1.0 求反，以输出方波
        MOV      TL1, #0CH    ; 用指令重装时间常数
        MOV      TH1, #1EH
        SJMP     TESTTF0      ; 输出方波循环，将子程序隔开
PTM:     MOV      TMOD, #10H
        MOV      TL1, #0CH
        MOV      TH1, #1EH
        SETB     TR1
        RET
        END

```

6-2 训练题

解：输出 100 个周期的方波，定时器溢出 200 次，以此为条件控制输出停止。参考程序清单如下：

```

        ORG      0000H
        AJMP     MAIN
        ORG      000BH
        LJMP     T0INT
        ORG      0050H
MAIN:    MOV      SP, #60H
        ACALL    PTM
TESTR6: CJNE     R6, #0, TESTR6
        CLR      TR0
        CLR      ET0
        SJMP     $
PTM:     MOV      TMOD, #02H
        MOV      TL0, #9CH
        MOV      TH0, #9CH

```

	MOV	R6, #0C8H	; 200 次
	SETB	ET0	
	SETB	EA	
	SETB	TR0	
	RET		
T0INT:	CPL	P1.0	; P1.0 求反, 以输出方波
	DEC	R6	; 计数值减 1
	RETI		
	END		

6-3 训练题

解: 参考程序如下

```
#include<reg51.h>
#define uchar unsigned char
void timer0() interrupt 1          /*1ms*/
{TR1 =~ TR1; }
void timer1() interrupt 3          /*产生方波*/
{ P1_0 =~ P1_0; }
void main()
{
    TMOD = 0x26; TL0=0xff;TH0=0xff;
    TL1 = 0x9c;TH0 = 0x9c;
    IE = 0x8a;                      /*等价于 ET0 =1; ET1 = 1;EA = 1;*/
    TR0 = 1;
    while (1);                      /*方波受控输出*/
}
```

刘焕成 2010 年 8 月 16 日修改

2011 年 2 月 20 日修改

2011 年 11 月 02 日修改