

实验八 定时器精确延时

【相关知识】

1. 51 单片机有哪些中断源

51 单片机有 5 个中断源：

中断源	中断向量号	说明
外部中断引脚/INT0	0	通过外部引脚 P3. 2 口触发中断
定时器 0 溢出中断	1	内部定时器 T0 溢出触发中断
外部中断引脚/INT1	2	通过外部引脚 P3. 2 口触发中断
定时器 1 溢出中断	3	内部定时器 T1 溢出触发中断
串口收发中断	4	串口接收/发送完成数据触发中断

之前我们讲解了外部中断，现在我们讲一讲定时器中断。

2. 51 的时钟系统介绍

晶振(晶体振荡器)可以产生稳定的方波。先看看晶振是什么样的。



晶振在封装上都会标有晶振的频率，左图就是 12M 的晶振，右图是 32.768k 的晶振，晶振的单位是 Hz，代表一秒钟能产生多少个方波。

这里我们先说一下 51 的内部时钟系统，51 的时钟其实很简单，51 他没有很复杂的时钟结构。我们使用的外部晶振一般是 12Mhz 或者是 11.0596Mhz，但是对于 51 来说，他的机器频率是晶振频率的 1/12. 对于 12Mhz 外部晶振来说，机器频率就是 1Mhz。

也就是说我们之前所说的执行一条空语句，花费的时间其实一个机器周期(机器周期 = $1s / \text{机器频率}$)，一个机器频率，我们算一下(以 12M 晶振为例)， $1s/1Mhz = 1\mu s$ ；我们大概是通过执行 1000000(一百万)次空语句，来软件延时到 1s 钟的。

3. 定时/计数器介绍

包括我们的 51 单片机，其内部的定时器的全称是什么？是定时/计数器。也就是说，所谓的定时，是通过晶振确定的时间基准之后，再通过计数计算出来的。

51 内部有两个 16 位定时/计数器，分别是 T0 和 T1，他们简单来说有什么特点呢？就是一个机器频率自加 1，因为是 16 位的计数器嘛，最多的计数值无非就是从 0 一直计数到 $2^{16}-1 = 65535$ 次，最长计时时间是 $65536\mu s = 65.536ms = 0.065536s$ 。

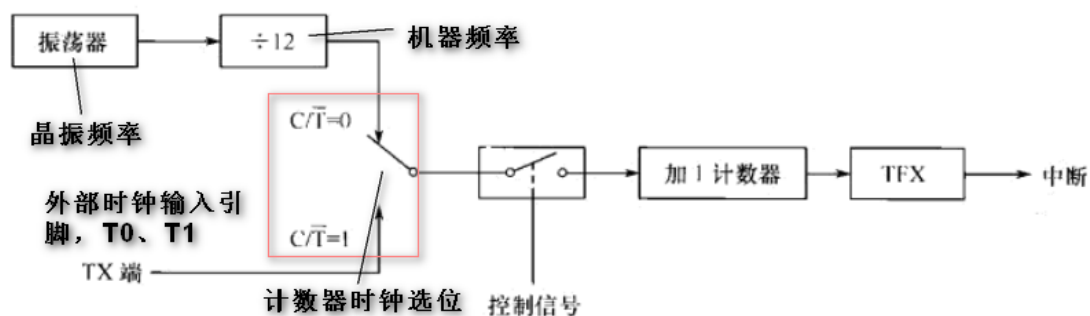
之前我们讲个外部中断的触发方式，他是通过外部的 /INT0 引脚触发的，对于定时计数器来说，他是在它溢出时会向 CPU 申请一个中断。

那么问题又来了，溢出时什么意思，我从 0000 0000 0000 0000 默默一直加呀加，加到最后变成了 1111 1111 1111 1111 到了最大值，我再加一，本来是变成了 1 0000 0000 0000 0000 的，但是我现在的最高位已经没有地方保存了，我们就说最高位溢出了。

溢出就是计数到了最大值，再加一，就溢出。此时，如果开启了定时器溢出中断，便会向 CPU 申请中断，进而执行定时器对应的中断服务函数。

4. 定时器的时钟源

之前我们讲解都是以 51 内部的机器频率 1Mz 来作为定时器的时钟输入的，其实定时器的时钟源也可以是外部的输入引脚 T0 (P3. 4)、T1 (P3. 5)。



5. 定时器的模式

上面我们讲的 16 位计数方式其实只是 51 的定时器的计数方式 1。

M1	M0	方式	功能说明
0	0	0	13 位的定时/计数器, 由 TL 中低 5 位和 TH 中 8 位组成
0	1	1	16 位定时/计数器
1	0	2	8 位自动重装定时/计数器, TL 为计数器, TH 为常数寄存器
1	1	3	定时器 0 分为两个 8 位定时/计数器, 定时器在该方式停止计数

对于 51 的定时器 T0 是有方式 0、1、2、3 的，对于定时器 T1 只有方式 0、1、2。

定时器的具体讲解，请看《单片机原理与应用：基于 Proteus 和 Keil》书中第六章。

6. 定时器相关配置寄存器

TCON 寄存器

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
8FH	8EH	8DH	8CH	8BH	8AH	89H	88H
位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0

位 7: 定时器 T1 溢出标志

T1 溢出时, 硬件自动使 TF1 置 1, 并向 CPU 申请中断。当进入中断服务程序时, 硬件自动将 TF1 清 0。TF1 也可以用软件清 0

位 6: 定时器 T1 运行控制位

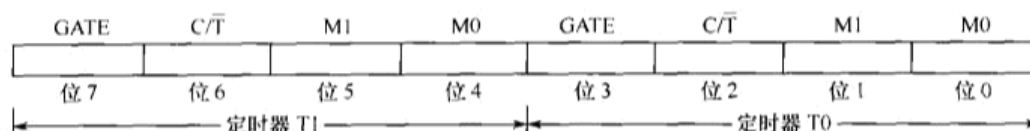
由软件置位和清零。GATE 为 0 时, T1 的计数仅由 TR1 控制, TR1 为 1 时允许 T1 计数, TR1 为 0 时禁止 T1 计数。GATE 为 1 时, 仅当 TR1 为 1 且 $\overline{\text{INT1}}$ 输入为高电平时才允许 T1 计数, TR1 为 0 或 $\overline{\text{INT1}}$ 输入低电平都将禁止 T1 计数

位 5: 定时器 T0 溢出标志。其功能和操作情况同位 7

位 4: 定时器 T0 运行控制位, 其功能和操作情况同位 6

位 3~0: 外部中断 $\overline{\text{INT1}}$ 和 $\overline{\text{INT0}}$ 请求及请求方式控制位, 其功能见第 5 章

TMOD 寄存器



TMOD 的低 4 位为 T0 的方式字, 高 4 位为 T1 的方式字

TMOD 不能位寻址, 必须整体赋值

M1 和 M0: 工作方式选择位

两位可形成 4 种编码, 对应 4 种工作方式, 见表 6.1

C/T: 定时和外部事件计数方式选择位 *

0: 定时器方式。定时器以振荡器输出时钟脉冲的 12 分频信号 (即机器周期) 作为计数信号

1: 外部事件计数器方式, 以外部引脚的输入脉冲作为计数信号

GATE: 门控位

0: 定时器计数不受外部引脚输入电平的控制, 只受定时器运行控制位 (TR0、TR1) 控制

1: 定时器的计数受定时器运行控制位和外部引脚输入电平的控制。其中 TR0 和 $\overline{\text{INT0}}$ 控制 T0 的运行, TR1 和 $\overline{\text{INT1}}$ 控制 T1 的运行

6. 开启总中断及定时器溢出中断

配置 IE 寄存器, 最高位位 IE[7]: EA 总开关, IE[1]: ET0 定时器 0 溢出中断,

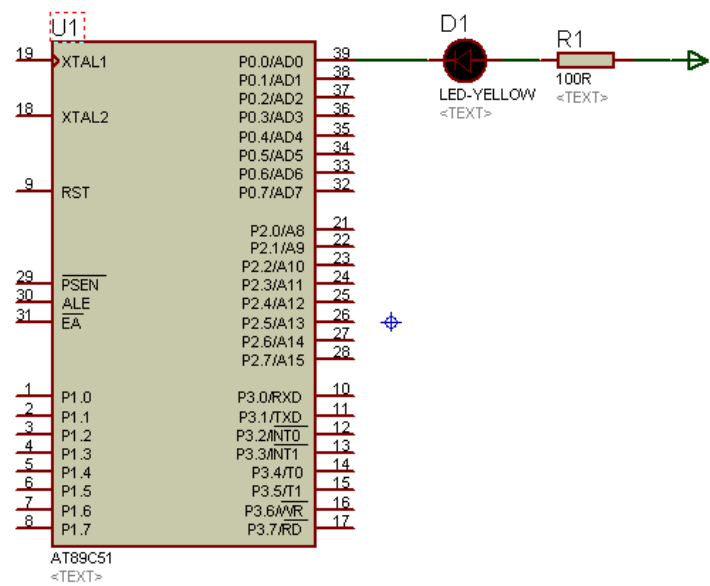
IE[1]: ET1 定时器 1 溢出中断。

7. 编写中断服务程序

```
void function(int t) interrupt 1
{
}

```

【实验电路】



【元件清单】

元器件名称		说明
AT89C51		主控芯片
CHIPRES100R		100 欧电阻
LED-YELLOW		黄色 LED 灯

【参考代码】

示例 1 LED 灯一秒闪烁一次

```
#include "reg51.h"

unsigned char InterruptCount = 0;

sbit led = P0^0;

int main()
{
    /* 定时器 0 工作在方式 1：16 位定时计数模式 */
    TMOD = 0x01;

    /* 这里计数初值为 62500，16 次就可以得到 1 000 000 次计数 */
    TH0 = 0x0b;
    TL0 = 0xdc;

    /* 开启总中断和定时器中断 */
}
```

```
IE = 0x82;

/* 开启定时器开始计时 */
TR0 = 1;

led = 0;

while(1);
}

/**
 * @Description 定时器 0 中断服务函数
 */
void TIM0_Handler() interrupt 1
{
    TR0 = 0;

    /* 手动的装载初值 */
    TH0 = 0x0b;
    TL0 = 0xdc;

    TR0 = 1;

    InterruptCount++;

    if(InterruptCount >= 16)
    {
        InterruptCount = 0;

        led = ~led;
    }
}
```

【代码分析】

设置定时器 0 的计数模式为方式 1, 16 位定时技术模式, 每次从 65535 - 62500 = 3036 开始计数到 65535, 共计数 62500 次, 反复计 16 次, 便计数了 1000000 次, 计时到了 1s。

3036 换算成 16 进制为 0xbdc。

【实验任务】

使用定时器, 实现电子表正常运行界面。