

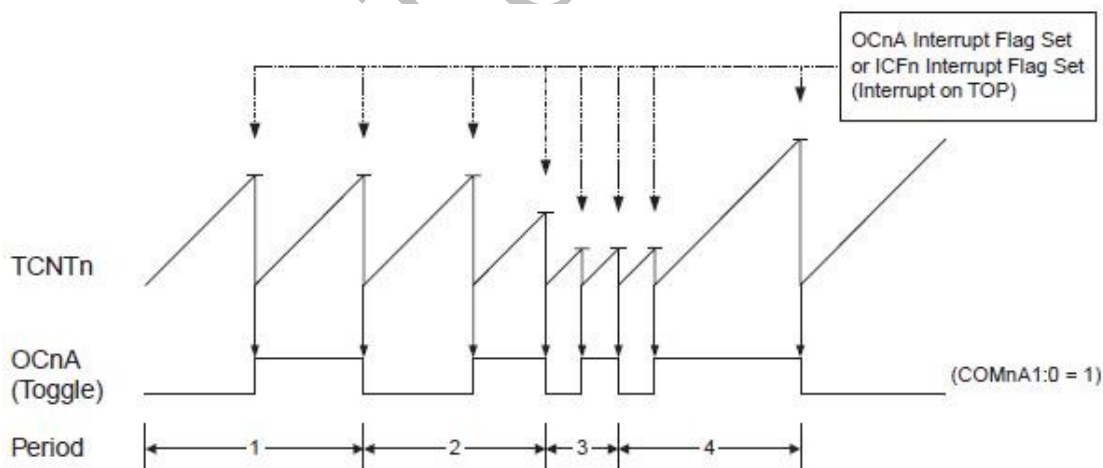
AVR ATmega16 16 位定时器/计数器

——CTC 模式《我和你》音乐源码

一、CTC(比较匹配时清零定时器)模式基本原理

在CTC 模式($WGM13:0 = 4$ 或 12) 里 $OCR1A$ 或 $ICR1$ 寄存器用于调节计数器的分辨率。当计数器的数值 $TCNT1$ 等于 $OCR1A$ ($WGM13:0 = 4$) 或等于 $ICR1$ ($WGM13:0 = 12$) 时计数器清零。 $OCR1A$ 或 $ICR1$ 定义了计数器的TOP 值, 亦即计数器的分辨率。这个模式使得用户可以很容易地控制比较匹配输出的频率, 也简化了外部事件计数的操作。CTC 模式的时序图为Figure 1。计数器数值 $TCNT1$ 一直累加到 $TCNT1$ 与 $OCR1A$ 或 $ICR1$ 匹配, 然后 $TCNT1$ 清零。

Figure 1 CTC 模式的时序图



利用 $OCF1A$ 或 $ICF1$ 标志可以在计数器数值达到TOP 时产生中断。在中断服务程序里可以更新TOP 的数值。由于CTC 模式没有双缓冲功能, 在计数器以无预分频器或很低的预分频器工作的时候将TOP 更改为接近BOTTOM 的数值时要小心。如果写入的 $OCR1A$ 或 $ICR1$ 的数值小

于当前TCNT1 的数值，计数器将丢失一次比较匹配。在下一次比较匹配发生之前，计数器不得不先计数到最大值0xFFFF，然后再从0x0000 开始计数到OCR1A或ICR1。在许多情况下，这一特性并非我们所希望的。替代的方法是使用快速PWM 模式，该模式使用OCR1A 定义TOP 值 (WGM13:0 = 15)，因为此时OCR1A 为双缓冲。为了在CTC模式下得到波形输出，可以设置OC1A在每次比较匹配发生时改变逻辑电平。这可以通过设置COM1A1:0 = 1 来完成。在期望获得OC1A 输出之前，首先要将其端口设置为输出 (DDR_OC1A = 1)。波形发生器能够产生的最大频率为

$$f_{OC2} = f_{clk_I/O}/2 \quad (OCR1A = 0x0000)。$$

频率由如下公式确定：

$$f_{oCnA} = \frac{f_{clk_I/O}}{2 \cdot N \cdot (1 + OCRnA)}$$

变量 N 代表预分频因子 (1、8、64、256 或 1024)。

二、相关寄存器

1、T/C1 控制寄存器 —— TCCR1A

Bit	7	6	5	4	3	2	1	0	
	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	TCCR1A
读/写	R/W	R/W	R/W	R/W	W	W	R/W	R/W	
初始值	0	0	0	0	0	0	0	0	

• Bit 7:6 - COM1A1:0: 通道A 的比较输出模式

• Bit 5:4 - COM1B1:0: 通道B 的比较输出模式

COM1A1:0与COM1B1:0分别控制OC1A 与OC1B状态。如果COM1A1:0 (COM1B1:0)的一位或两位被写入“1”，OC1A(OC1B) 输出功能将取代 I/O 端口功能。此时OC1A(OC1B)相应的输出引脚数据方向控制必须置位以使能输出驱动器。

OC1A(OC1B) 与物理引脚相连时, COM1x1:0 的功能由WGM13:0 的设置决定。Table 1给出当WGM13:0 设置为普通模式与CTC 模式(非PWM) 时COM1x1:0 的功能定义。

Table 1. 比较输出模式，非PWM

COM1A1/COM1B1	COM1A0/COM1B0	说明
0	0	普通端口操作，非OC1A/OC1B 功能
0	1	比较匹配时OC1A/OC1B 电平取反
1	0	比较匹配时清零OC1A/OC1B(输出低电平)
1	1	比较匹配时置位 OC1A/OC1B (输出高电平)

- Bit 3 - FOC1A: 通道A 强制输出比较
- Bit 2 - FOC1B: 通道B 强制输出比较

FOC1A/FOC1B只有当WGM13:0指定为非PWM模式时被激活。为与未来器件兼容,工作在PWM 模式下对TCCR1A 写入时,这两位必须清零。当FOC1A/FOC1B 位置1 , 立即强制波形产生单元进行比较匹配。COM1x1:0 的设置改变 OC1A/OC1B 的输出。注意FOC1A/FOC1B 位作为选通信号。COM1x1:0 位的值决定强制比较的效果。

在CTC 模式下使用OCR1A 作为TOP 值, FOC1A/FOC1B 选通即不会产生中断也不好清除定时器。

FOC1A/FOC1B 位总是读为0。

- Bit 1:0 - WGM11:0: 波形发生模式

这两位与位于TCCR1B 寄存器的WGM13:2 相结合，用于控制计数器的计数序列——计数器计数的上限值和确定波形发生器的工作模式。Table 2给出当WGM13:0设置为普通模式与CTC 模式(非PWM) 时的功能定义。

Table 2、CTC波形产生模式的位描述

模式	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	定时器/ 计数器工作模式	计数上限值TOP	OCR1x 更新时刻	TOV1 置位时刻
0	0	0	0	0	普通模式	0xFFFF	立即更新	MAX
4	0	1	0	0	CTC	OCR1A	立即更新	MAX
12	1	1	0	0	CTC	ICR1	立即更新	MAX

2、T/C1 控制寄存器 —— TCCR1B

Bit	7	6	5	4	3	2	1	0	
	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
读/写	R/W	R/W	R	R/W	W	W	R/W	R/W	
初始值	0	0	0	0	0	0	0	0	

- Bit 7 – ICNC1: 入捕捉噪声抑制器

置位ICNC1 将使能输入捕捉噪声抑制功能。此时外部引脚ICP1 的输入被滤波。其作用是从ICP1 引脚连续进行4 次采样。如果4 个采样值都相等，那么信号送入边沿检测器。因此使能该功能使得输入捕捉被延迟了4 个时钟周期。

- Bit 6 – ICES1: 输入捕捉触发沿选择

该位选择使用ICP1 上的哪个边沿触发捕获事件。ICES 为“0” 选择的是下降沿触发输入捕捉；ICES1 为“1” 选择的是逻辑电平的上升沿触发输入捕捉。

按照 ICES1 的设置捕获到一个事件后，计数器的数值被复制到 ICR1 寄存器。捕获事件还会置为 ICF1。如果此时中断使能，输入捕捉事件即被触发。

当 ICR1 用作 TOP 值（见 TCCR1A 与 TCCR1B 寄存器中 WGM13:0 位的描述）时，ICP1 与输入捕捉功能脱开，从而输入捕捉功能被禁用。

• Bit 5 – 保留位

该位保留。为保证与将来器件的兼容性，写 TCCR1B 时，该位必须写入“0”。

• Bit 4:3 – WGM13:2: 波形发生模式

见 TCCR1A 寄存器中的描述。

• Bit 2:0 – CS12:0: 时钟选择

这 3 位用于选择 T/C 的时钟源

Table 2. 时钟选择位描述。

CS12	CS11	CS10	说明
0	0	0	无时钟源（T/C 停止）
0	0	1	clkI/0/1（无预分频）
0	1	0	clkI/0/8（来自预分频器）
0	1	1	clkI/0/64（来自预分频器）
1	0	0	clkI/0/256（来自预分频器）
1	0	1	clkI/0/1024（来自预分频器）
1	1	0	外部 T1 引脚，下降沿驱动
1	1	1	外部 T1 引脚，上升沿驱动

选择使用外部时钟源后，即使 T1 引脚被定义为输出，其 1 引脚上的逻辑信号电平变化仍然会驱动 T/C1 计数，这个特性允许用户通过软件来控制计数。

3、T/C1 — TCNT1H 与 TCNT1L

Bit	7	6	5	4	3	2	1	0	
	TCNT1[15:8]								TCNT1H
	TCNT1[7:0]								TCNT1L
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始值	0	0	0	0	0	0	0	0	

TCNT1H与TCNT1L组成了T/C1的数据寄存器TCNT1。通过它们可以直接对定时器/计数器单元的16 位计数器进行读写访问。为保证CPU对高字节与低字节的同时读写，必须使用一个8 位临时高字节寄存器TEMP。TEMP 是所有的16 位寄存器共用的，

在计数器运行期间修改TCNT1的内容有可能丢失一次TCNT1与OCR1x的比较匹配操作。写TCNT1 寄存器将在下一个定时器周期阻塞比较匹配。

4. 1、输出比较寄存器1A — OCR1AH与 OCR1AL

Bit	7	6	5	4	3	2	1	0	
	OCR1A[15:8]								OCR1AH
	OCR1A[7:0]								OCR1AL
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始值	0	0	0	0	0	0	0	0	

4. 2、输出比较寄存器1B — OCR1BH与 OCR1BL

Bit	7	6	5	4	3	2	1	0	
	OCR1B[15:8]								OCR1BH
	OCR1B[7:0]								OCR1BL
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始值	0	0	0	0	0	0	0	0	

该寄存器中的16 位数据与TCNT1 寄存器中的计数值进行连续的

比较，一旦数据匹配，将产生一个输出比较中断，或改变OC1x 的输出逻辑电平。

输出比较寄存器长度为16 位。为保证CPU 对高字节与低字节的
同时读写，必须使用一个8 位临时高字节寄存器TEMP。TEMP 是所有的16 位寄存器共用的 。

5、输入捕捉寄存器1 — ICR1H 与ICR1L

Bit	7	6	5	4	3	2	1	0	
	OCR1[15:8]								OCR1H
	OCR1[7:0]								OCR1L
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始值	0	0	0	0	0	0	0	0	

当外部引脚ICP1(或T/C1的模拟比较器)有输入捕捉触发信号产生时，计数器TCNT1中的值写入ICR1 中。ICR1 的设定值可作为计数器的TOP 值。

输入捕捉寄存器长度为16 位。为保证CPU 对高字节与低字节的
同时读写，必须使用一个8 位临时高字节寄存器TEMP。TEMP 是所有的16 位寄存器共用的。

6、T/C1 中断屏蔽寄存器— TIMSK

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始值	0	0	0	0	0	0	0	0	

Note: 该寄存器包含几个 T/C 的中断控制位，只对 T1 位进行说明。

- **Bit 5 – TICIE1: T/C1 输入捕捉中断使能**

当该位被设为“1”，且状态寄存器中的I 位被设为“1” 时，T/C1 的输入捕捉中断使能。一旦TIFR 的ICF1 置位，CPU 即开始执行T/C1 输入捕捉中断服务程序([见ATmega 16中文说明书 P43 “ 中断”](#))。

- **Bit 4 – OCIE1A: 输出比较 A 匹配中断使能**

当该位被设为“1”，且状态寄存器中的I 位被设为“1” 时，T/C1 的输出比较A 匹配中断使能。一旦TIFR 上的OCF1A 置位，CPU 即开始执行T/C1 输出比较A 匹配中断服务程序([见ATmega 16中文说明书 P43 “ 中断”](#))。

- **Bit 3 – OCIE1B: T/C1 输出比较 B 匹配中断使能**

当该位被设为“1”，且状态寄存器中的I 位被设为“1” 时，使能T/C1 的输出比较B 匹配中断使能。一旦TIFR 上的OCF1B 置位，CPU 即开始执行T/C1 输出比较B 匹配中断服务程序([见ATmega 16中文说明书 P43 “ 中断”](#))。

- **Bit 2 – TOIE1: T/C1 溢出中断使能**

当该位被设为“1”，且状态寄存器中的I 位被设为“1” 时，T/C1 的溢出中断使能。一旦TIFR 上的TOV1 置位，CPU 即开始执行T/C1 溢出中断服务程序([见ATmega 16中文说明书 P43 “ 中断”](#))。

7、T/C 中断标志寄存器— TIFR

Bit	7	6	5	4	3	2	1	0	
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	TIFR
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始值	0	0	0	0	0	0	0	0	

Note: 该寄存器包含几个 T/C 的标志位，只对 T1 位进行说明。

• Bit 5 – ICF1: T/C1 输入捕捉标志位

外部引脚ICP1 出现捕捉事件时ICF1 置位。此外，当ICR1 作为计数器的TOP 值时，一旦计数器值达到TOP， ICF1 也置位。

执行输入捕捉中断服务程序时ICF1 自动清零。也可以对其写入逻辑“1” 来清除该标志位。

• Bit 4 – OCF1A: T/C1 输出比较 A 匹配标志位

当TCNT1 与OCR1A 匹配成功时，该位被设为“1”。

强制输出比较(FOC1A) 不会置位OCF1A。

执行强制输出比较匹配A 中断服务程序时OCF1A 自动清零。也可以对其写入逻辑“1” 来清除该标志位。

• Bit 3 – OCF1B: T/C1 输出比较 B 匹配标志位

当TCNT1 与OCR1B 匹配成功时，该位被设为“1”。

强制输出比较(FOC1B) 不会置位OCF1B。

执行强制输出比较匹配B 中断服务程序时OCF1B 自动清零。也可以对其写入逻辑“1” 来清除该标志位。

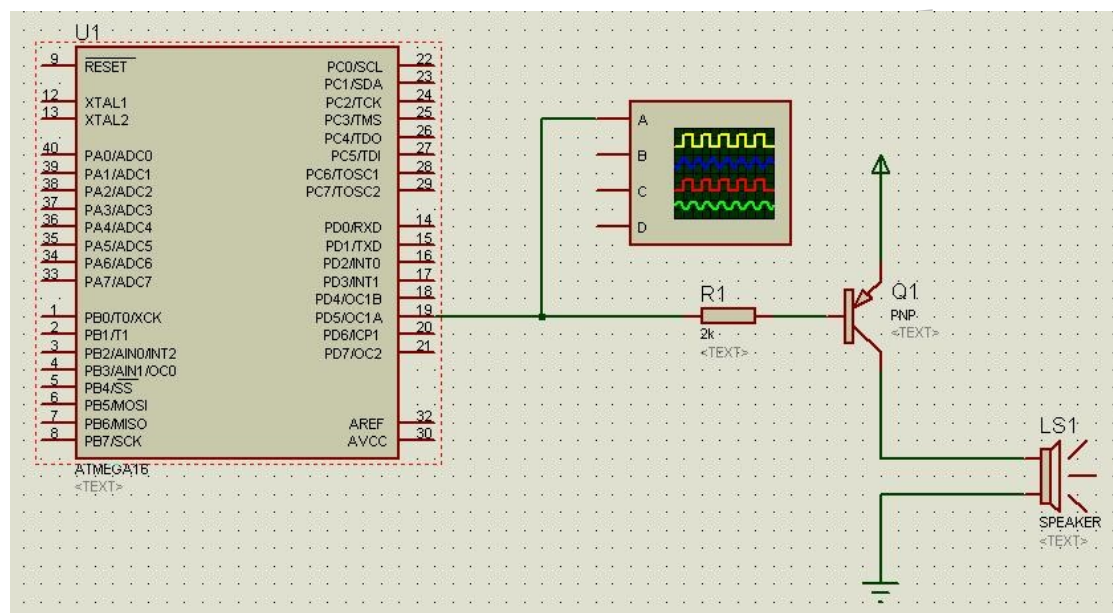
• Bit 2 – TOV1: T/C1 溢出标志

该位的设置与T/C1 的工作方式有关。工作于普通模式和CTC 模式时，T/C1 溢出时TOV1置位。对工作在其它模式下的TOV1 标志位置

位，见ATmega 16中文说明书 P104 Table 47。

执行溢出中断服务程序时OCF1A 自动清零。也可以对其写入逻辑“1” 来清除该标志位。

三、PROTEUS 仿真电路图



四、程序源码

```
/*  
实验名称：CTC波形应用实验  
日期：2011.02.17  
作者：Kukulee  
芯片类型：ATMEGA16 16PU  
时钟频率：1MHZ  
编译环境：ICCAVR Version 7.14C  
功能：用CTC波形演奏《我和你》音乐  
*/  
  
#include <iom16v.h>  
#include <macros.h>  
  
#define uint unsigned int  
#define uchar unsigned char
```

```

flash uint music_data_1[][2]=
{
    {587,800},{698,800},{466,1600},{523,800},{587,800},
    {349,1600},{466,800},{523,800},{587,800},{698,800},
    {523,3200},{587,800},{698,800},{466,1600},{523,800},
    {587,800},{392,1600},{523,800},{349,800},{523,800},
    {587,800},{466,1600},{784,1600},{698,1600},{784,1600},
    {466,1600},{587,800},{392,800},{587,1200},{698,400},
    {523,3200},{587,800},{698,800},{466,1600},{523,800},
    {587,800},{392,1600},{523,800},{349,800},{523,800},
    {587,800},{466,3200},{0,0}
};
//音乐数据，前面是频率，后面是持续时间

void delay_nms(uint k) //延时程序
{
    uchar a;
    for(;k;k--)
    {
        for (a = 220; a; a--) ;
    }
}

/*****
闹铃音乐相关定义
*****/

void music_1(uchar i) //CTC模式音乐演奏
{
    uint x;
    x=music_data_1[i][0]; //取音乐的频率数据
    if(x!=0)
    {
        x=(500000/x)-1; //1M时钟数据计算
        OCR1A=x; //比较值赋值
        //可根据公式算出CTC波频率
        TCCR1A = 0x50; //0B01010000 比较匹配时
        //OC1A/OC1B 电平取反
        TCCR1B = 0x09; //0B00001001 CTC模式
        //无预分频
    }
    else
    {
        TCCR1A = 0x00; //0B00000000 普通端口操作
        //非OC1A/OC1B功能
    }
}

```

```
void music()
{
    uchar z;
    TCCR1A = 0x40;           //0B01000000 比较匹配时
                             //OC1A电平取反
    TCCR1B = 0x09;           //0B00001001 CTC模式
                             //无预分频
    while(music_data_1[z][1]!=0) //等待最后一个音符到来
    {
        music_1(z);          //输出音符
        delay_nms((music_data_1[z][1])); //节拍延时
        TCCR1B = 0x00;        //奏完一个音符停止
        z++;                  //准备一下个音符
    }
    z = 0;
    delay_nms(1000);
}

void main()
{
    DDRD|=BIT(5);            //PD5设置为输出
    while(1)                  //设置一个死循环，不停演奏
    {
        music();
    }
}
```

参考资料

- 1、 AVR单片机ATmega16_16L_中文数据手册;
- 2、 天祥电子AVR视频教程;
- 3、 基于avr单片机- MEGA16 独立按键+数码管显示+DS18B20测温+LCD1602显示 —— 彩虹企鹅;