

HT48 MCU 定时器/计数器的使用

文件编码：HA0020s

简介：

本文以 HT48R10A-1 为例，分别介绍 HT48RXA-1 系列中定时器/计数器的内部计时模式、脉宽测量模式、事件计数模式这三种模式的使用方法及其注意事项。

使用说明

HT48R10A-1 是一个八位高性能精简指令单片机。有一个八位可编程的递增定时/计数器，可采用外部或内部时钟。内部时钟源来自系统时钟。外部的时钟输入允许用户测量外部事件、测量时间间隔、脉冲宽度或者用以产生精确的定时。该系列的单片机定时器功能和 HT48R10A-1 大致相同，只是在定时器个数和定时时钟源上略有不同。可见该系列的单片机比较文档。

HT48R10A-1 有两个和定时/计数器相关的寄存器 (TMR[0DH]、TMRC[0EH])。其中，TMR 为计数初值，该初值会硬件自动重新装入。TMRC 为控制寄存器。

HT48RXA-1 把系统时钟经过预分频后作为定时/计数器的内部时钟来源，预分频的阶数可由 TMRC 的前三位设定。这和 HT48 系列的其他微控制器有所区别。

定时/计数器控制寄存器 TMRC

符号	位	功能
PSC0~PSC2	0-2	定义预分频器级数，PSC2、PSC1、PSC0= 000： $f_{INT} = f_{SYS}/2$ 或 $f_{RTC}/2$ 001： $f_{INT} = f_{SYS}/4$ 或 $f_{RTC}/4$ 010： $f_{INT} = f_{SYS}/8$ 或 $f_{RTC}/8$ 011： $f_{INT} = f_{SYS}/16$ 或 $f_{RTC}/16$ 100： $f_{INT} = f_{SYS}/32$ 或 $f_{RTC}/32$ 101： $f_{INT} = f_{SYS}/64$ 或 $f_{RTC}/64$ 110： $f_{INT} = f_{SYS}/128$ 或 $f_{RTC}/128$ 111： $f_{INT} = f_{SYS}/256$ 或 $f_{RTC}/256$
TE	3	定义定时/计数器 TMR 的触发方式 (0=上升沿作用，1=下降沿作用)
TON	4	打开/关闭定时/计数器(1=打开，0=关闭)
—	5	未用，读数为“0”
TM0 TM1	6 7	定义工作模式： 01=事件计数模式(外部时钟) 10=定时模式(内部时钟) 11=脉冲宽度测量模式 00=未用

使用时，注意 f_{INT} CLOCK 时钟源不是在 tools/mask option 中选择，而是由软件在 tmrc 中选择。而采用 f_{RTC} 时钟源的条件是，掩膜选择应选择为：intRC+RTC 模式。

如果定时/计数器的记数值为 FFH，则定时/计数器会从预置寄存器中重载记数值。这时向 TMR 中写入数据会立即被定时/计数器载入。如果定时/计数器被关闭，则当定时/计数器重新开始记数时，定时/计数器也会从预置寄存器中载入数据，在这之前写入 TMR 的数据也会被载入定时/计数器。而当定时/计数器在运行状态下且他的值没有到 FFH 时，向 TMR 写入的数据只会被存放在预置寄存器中。直到定时/计数器发生溢出时定时/计数器才会从预置寄存器中载入数据。在读定时/寄存器时，

为了避免错误，时钟会被停止。时钟禁止可能会导致计数错误，程序员必须注意到这一点。
无论哪种模式，写“0”到 ETI 可以禁止中断服务。

例程：

<例 1>

；程序名：基本的内部定时模式 (wave.asm)

；目的：该例将用定时器在 pa.0 产生

；占空比 1:1 的 100ms 方波

；掩膜选择：先选择 mask option/osc /32krtc+bulitinrc,这时，在 timer clock 选项中，

；选取 32k rtc 选项以确定时基。

；程序清单：

```
include ht48R10A-1.inc
```

```
code .section at 0h 'code'
```

```
;;-----
```

```
data .section 'data'
```

```
pavalue db?
```

```
temp db ?
```

```
temp1 db ?
```

```
;;-----
```

```
main .section 'code'
```

```
org 00h
```

```
jmp start
```

```
org 08h
```

```
jmp intstart
```

```
start:
```

```
mov a,0feh
```

```
mov pac,a ;pa.0 为输出
```

```
main:
```

```
set intc.0 ;开总中断
```

```
set intc.2 ;开定时器中断
```

```
mov a,84h ;采用 fsys=32768Hz rtc,fint=fsys/32
```

```
mov tmrc,a
```

```
mov a,0ceh ;初值,fint 约等于 1ms,50*1ms=50ms
```

```
mov tmr,a ;开始计数
```

```
set tmrc.4
```

```
jmp $
```

```
intstart:
```

```
mov temp,a ;保存 acc
```

```
mov a,status ;保存 status
```

```
mov temp1,a
```

```
mov a,01h
```

```
xorm a,pa ;对 pa.0 取反
```

```
mov a,temp1
```

```
mov status,a ;恢复 status
```

```
mov a,temp ;恢复 acc
```

```
reti
```

```
end
```

程序说明：(1) 定时模式下，时钟源来自于 f_{INT} 时钟信号。运行后，pa.0 将产生占空比为 1:1 的方波。本例还表明计数初值的改变是在前一次计数结束后进行的。

<例2>

```

; 程序名：脉宽测量模式(wavetest.asm)
; 作者：黄山云
; 目的：本例将演示对一方波脉宽的测量。
; 程序清单：
; 掩膜选择：选石英振荡器，系统时钟频率 4MHz，分频系数为最小 2，定时器 fINT = 2MHz
;
include ht48R10A-1.inc
data .section 'data'
    count db ?
    over db ?
code .section at 0h 'code'
org 00h
jmp start
    org 08h
jmp timeint
start:
    clr over
    clr acc
    clr count ;程序初始化
    clr intc
    mov a,05h
    mov intc,a ; 允许总中断及定时器中断
    mov a,11000000B
    mov tmrc,a ; 设置模式为脉冲宽度测量模式，测量下降沿
    mov a,00h
    mov tmr,a ; 置计数初值
    set tmrc.4

loop:
    snz tmrc.4 ;查询计数允许位，看测量是否结束
    jmp count1 ; 结束（计数控制位为0）则跳到count1
    jmp loop ;未结束（计数控制位为1）则继续循环等待测量结束
count1:
    mov a,tmr
    jmp $
timeint:
    inc count ; 若脉宽超过0FFH，则定时器溢出，计数单元加1
    sz count ;count 若溢出则置位标志
    jmp $1 ; (1) 增加单元数可扩大测量范围
    set over.0
$1:
    reti

```

end

程序说明：

HT48RXA-1单片机的PC1引脚和TMR的输入引脚是复用的,这里用作TMR。只有这种模式TON会自动清零,其它模式TON位只可以用指令清除。

在脉冲宽度测量模式时,其 TON 和 TE 位皆为 1 时,如果引脚 TMR 接收到一个上升沿信号(如 TE 值为 0,则为下降沿信号)时,计数器会开始计数直至 TMR 引脚回到原来的电平为止,并且会将 TON 清零,只有这种模式 TON 会自动清零,其它模式 TON 位只可以用指令清除。计数器停止计数,测量的结果则仍然保留在定时/计数器之中。换句话说,脉冲宽度测量模式只能测量一个脉冲。只要 TON 位又被置位,则当引脚 TMR 接到跳变脉冲,测量周期会再次执行下去。在这个脉冲宽度测量模式中,定时/计数器并不是根据逻辑电平来计数,而是根据信号的跳变沿。一旦发生计数器溢出,计数器会从定时/计数器加载寄存器重新装入,同时还会发出中断请求,这个情况和事件计数模式和定时器模式一样。

在该模式下,脉宽的测量精度显然将由定时器的时基精度决定。测量前,估算频率的大致范围,可以提高测量的效率。在此基础上,选用合适的时基和足够的内存单元,可以实现所需的精度要求。

本例中,将测量1K频率的方波。程序采用内存单元作为计数扩展见(1),以提高测量范围和精度。

测试结果:该程序脉冲测量范围为16Hz~600KHz

<例3>

```

; 程序名:外部计数模式(COUNT.ASM)
; 作者:黄山云
; 目的:计算 1 0 0 ms时间内的外部输入的脉冲个数
; 程序清单;
; 掩膜选择:系统选 4 M的系统时钟源,指令周期为 1 u s
include ht48R10A-1.inc
;
data .section 'data'
    R1      db  ?
    R2      db  ?
    count  db  ?
    over   db  ?
code .section at 0h 'code'
org 00h
    jmp start
org 08h
    jmp timeint
start:
    clr over
    clr acc
    clr count ;程序初始化
    clr intc
    mov a,05h
    mov intc,a ;允许总中断及定时器中断

```

```

mov    a,01000000B
mov    tmrc,a           ;设置模式为计数测量模式，测量上升沿
mov    a,00h
mov    tmr,a           ;置计数初值
set    tmrc.4

MS:
                                ; ( 1 ) 由指令实现100ms定时
                                ;时间误差为2 μs

mov    a,0c8h           ;1 μ
mov    r1, a           ;1 μ
LP2:   mov    a,0a5h     ;1μ
mov    r2,a           ;1 μ
LP1:   sdz    r2         ;1μ
        jmp    LP1      ;2μ
        sdz    R1       ;1μ
        jmp    LP2      ;2μ
        clr    TMRC.4   ; ( 2 ) 停止计数
        mov    A,TMR    ;获取计数值
        jmp    $

timeint:
inc    count           ;定时器溢出，计数单元加1
sz    count            ;count 若溢出则置位标志
jmp    $1              ;增加单元数可扩计数范围
set    over.0
$1:
reti

end

```

程序说明：

使用时，应注意 TON 需指令清零见 (2)，只有在脉冲宽度测量模式中，系统会在一个脉宽周期完成后自动清除 TON 位。但在其它三种模式中，TON 位只可以用指令清除。否则，会一直计数到定时器溢出，之后会重载初值寄存器的初值。(1) 软件指令实现 100ms 的定时，

计算方法如下：

$$2 + [2 + 3 \times R2(165) + 3] \times R1(200) = 100002 \mu S$$

计数结果：输入 1 KHZ 方波，TMR 为 64H

输入 100 HZ 方波，TMR 为 0AH

输入 150 HZ 方波，TMR 为 0FH

均正确