



● 产品名称

➤ MK7A21P

● 标题

如何使用 MK7A21P 的 TMR1 捕捉功能

● 简介

MK7A21P 的 TMR1 是一个 16 位递减计数器，它的特性与下面 7 个寄存器有关，这 7 个寄存器分别是：TM1_CTL1、TM1_CTL2、CLR_CNT、TM1L_LA、TM1H_LA、TM1L_CNT 和 TM1H_CNT。如下是对这几个寄存器的说明：

A) TM1_CTL1 (0X13)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1_CTL1	TM1_EN	WR_CNT	SUR1	SUR0	EDGE	PRE2	PRE1	PRE0

✧ TM1_EN (Bit7)

1 表示 TM1 开启。

0 表示 TM1 停止。

✧ WR_CNT (Bit6)

1 表示程序指令在给 TM1L_LA 和 TM1H_LA 送数据后，立即将此数据传送给 TM1L_CNT 和 TM1H_CNT。

0 表示程序指令在给 TM1L_LA 和 TM1H_LA 送数据后，并不立即将此数据传送给 TM1L_CNT 和 TM1H_CNT。

✧ SUR1 和 SUR0 (Bit5 和 Bit4)

此两位为 TM1 时钟源的选择位，用户可以根据下表选择适当的时钟源。

Bit5	Bit4	TM1 clock source
SUR1	SUR0	
0	0	EXT_CLK (PA4)
0	1	Crystal mode OSC1
1	0	RC mode (Ext. & Internal RC) OSC1
1	1	Clock source is system clock and capture input is PA4

✧ EDGE (Bit3)

1 表示下降沿有效。

0 表示上升沿有效。

✧ PRE2、PRE1 和 PRE0 (Bit3-0)

此三位决定着 TM1 的预分频比，用户可以根据下表选择。



Bit2	Bit1	Bit0	TMR1 Prescaler rate
PRE2	PRE1	PRE0	
0	0	0	1:1
0	0	1	1:2
0	1	0	1:4
0	1	1	1:8
1	0	0	1:16
1	0	1	1:32
1	1	0	1:64
1	1	1	1:128

B) TM1_CTL2 (0X1F)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1_CTL2	ENC	--	--	--	--	--	--	--

◇ ENC(Bit7)

1 表示在 TM1 为捕捉模式时，一旦捕捉到相应信号，硬件自动将 TM1L_CNT 和 TM1H_CNT 计数器清零。

0 表示在 TM1 为捕捉模式时，一旦捕捉到相应信号，要通过写 CLR_CNT 寄存器将 TM1L_CNT 和 TM1H_CNT 计数器清零。

C) CLR_CNT (0X21)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CLR_CNT	CLR_CNT	--	--	--	--	--	--	--

当 TM1_CTL2 的 Bit7 置 1 时，可通过写 CLR_CNT 寄存器清零 TM1L_CNT 和 TM1H_CNT 计数器。

D) TM1L_LA/TM1H_LA 和 TM1L_CNT/TM1H_CNT (0X14、0X15、0X16 和 0X17)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1L_LA	D7	D6	D5	D4	D3	D2	D1	D0
TM1H_LA	D7	D6	D5	D4	D3	D2	D1	D0
TM1L_CNT	D7	D6	D5	D4	D3	D2	D1	D0
TM1H_CNT	D7	D6	D5	D4	D3	D2	D1	D0

TM1L_CNT/TM1H_CNT 为只可读计数器，用户不能对此寄存器进行写操作。当 TM1 作时钟用时，TM1L_LA/TM1H_LA 为 TM1L_CNT/TM1H_CNT 的预载寄存器；当 TM1 作捕捉用时，硬件将捕捉到的数据存入 TM1L_LA/TM1H_LA 里面，以供用户读取。



为了让用户更深入地了解 TM1 的捕捉功能的用法，我们特地提供了如下试验，以供用户参考。此次试验的线路图见图 1，基本功能如下：

- 启动芯片的 PWM 功能，让芯片的第 10 脚（即 PWM/PC0）输出周期为 800us，占空比为 50% 的方波。
- 利用 TM1 的捕捉功能来检测以上方波的周期大小，并将捕捉到的数据按 16 进制从低到高通过图中的 LED1~13 分别显示出来（这里正确的数据应该是 0X320，误差±3）。

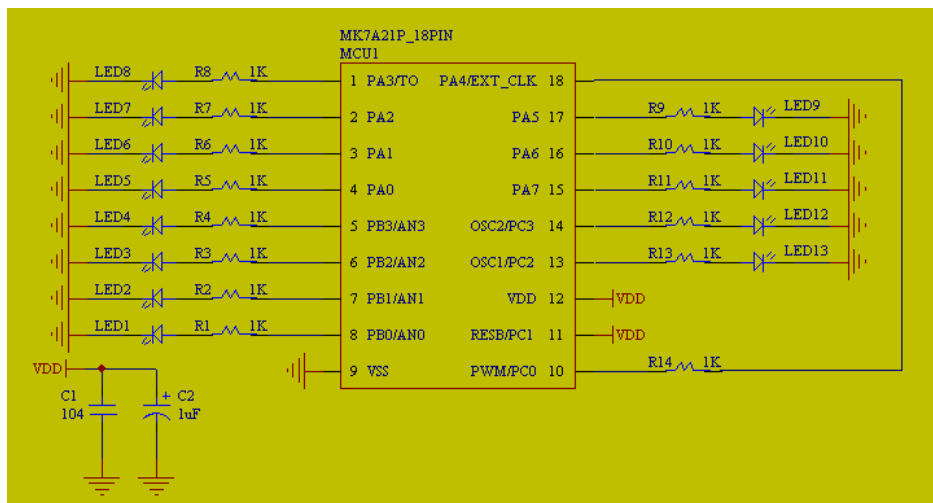


图 1 TM1 捕捉功能试验线路图

● DEMO 程序

➤ 汇编程序文档

```
-----  
#include "mk7a21p_hw.inc" ;编译该文档需包含"mk7a21p_hw.inc"文件  
-----  
;芯片型号 (mk7a21p)  
-----  
;配置寄存器设置说明 (CONFIG)  
;1-----FOSC=INRC&NS ;LS,NS,HS,RC,INRC (单时钟)  
; ;INRC&LS,INRC&NS,INRC&HS  
;2-----CPT=OFF ;ON,OFF  
;3-----WDTE=Disable ;Enable,Disable  
;4-----LV=2.3V ;4V,2.3V,Don't use  
;5-----RST_DEF=...input... ;...input...,...reset...  
;6-----EXT_CLK=...timer source... ;...IO...,...timer source...  
-----  
a_buf equ 0x40  
status_buf equ 0x41
```



```
dat_0      equ      0x42
dat_1      equ      0x43

;-----

      org      0x000      ;mk7a21p 的复位向量地址定义
      lgoto    main      ;跳转到主程序入口
      org      0x004      ;mk7a21p 的中断向量地址定义
      lgoto    int        ;跳转到中断程序入口

;-----

      org      0x010

int

      movam    a_buf
      swap     status,a
      movam    status_buf      ;保护现场

;-----

      btsc     irqf,tm1f
      lgoto    int_tm1      ;进入 TM1 有关的中断服务程序
      btsc     irqf,tm2f
      lgoto    int_tm2      ;进入 TM2 有关的中断服务程序
      btsc     irqf,tm3f
      lgoto    int_tm3      ;进入 TM3 有关的中断服务程序
      btsc     irqf,paf
      lgoto    int_pa       ;进入 PortA 中断服务程序
      btsc     irqf,adcf
      lgoto    int_adc      ;进入 AD 转换中断服务程序

;-----

int_end

      swap     status_buf,a
      movam    status
      swap     a_buf,m
      swap     a_buf,a      ;恢复现场
      reti

;-----

int_tm1

      bc       irqf,tm1f

;-----

;TM1 中断服务程序
      com      tm1l_la,a
      movam    dat_0
      com      tm1h_la,a
      movam    dat_1
```



```
lcall    display
lgoto    int_end
;-----
int_tm2
    bc    irqf,tm2f
;-----
;TM2 中断服务程序
;-----
lgoto    int_end
;-----
int_tm3
    bc    irqf,tm2f
;-----
;TM2 中断服务程序
;-----
lgoto    int_end
;-----
int_pa
    bc    irqf,paf
;-----
;PortA 中断服务程序
;-----
lgoto    int_end
;-----
int_adc
    bc    irqf,adcf
;-----
;adc 中断服务程序
;-----
lgoto    int_end
;-----
main
    movla    b'00000000'
    movam    sys_ctl    ;单时钟模式时此设置无效
                        ;双时钟模式时:Bit7---系统时钟选择
                        ;双时钟模式时:Bit1---内部 RC 振荡控制
                        ;双时钟模式时:Bit0---外部振荡控制
;-----
    movla    b'10000011'
    movam    wdt_ctl    ;WDT 的使能及其预分频为 1:8
```



```
-----  
;-----  
movla      b'00010000'  
movam      porta_dir      ;PA4 输入  
movla      b'00000000'  
movam      porta_dat      ;PA0-7 状态  
movla      b'00000000'  
movam      pa_plu        ;PA0-7 上拉禁止  
movla      b'00000000'  
movam      wake_up       ;PA0-7 唤醒禁止  
;-----  
movla      b'11110000'  
movam      portb_dir     ;PB0-3 输出  
movla      b'11110000'  
movam      portb_dat     ;PB0-3 状态  
movla      b'11110000'  
movam      pb_plu       ;PB0-3 上拉禁止  
;-----  
movla      b'11110000'  
movam      portc_dir     ;PC0,2,3 输出,PC1 只能作为输入  
movla      b'11110000'  
movam      portc_dat     ;PC0-3 状态  
movla      b'11110000'  
movam      pc_plu       ;PC0-3 上拉禁止  
;-----  
movla      b'00110000'    ;以下为 TMR1 的初始化程序段  
movam      tm1_ctl1      ;TM1 用作捕捉模式,预分频 1:1  
movla      b'00000000'  
movam      tm1_ctl2      ;Bit7:比较计数器自动清零使能  
;-----  
mov        clr_cnt,m     ;写这个寄存器将清零比较计数器  
;-----  
;movla      0x50  
;movam      tm1l_la  
;movla      0xc3  
;movam      tm1h_la  
;-----  
movla      b'01100010'    ;以下为 TMR2 的初始化程序段  
movam      tm2_ctl1  
movla      b'11000000'  
movam      tm2_ctl2      ;Bit7:模式选择/Bit6:PWM 初始状态
```



;Bit5:IO 或 TO_E/Bit3-0:PWM 预分频

```
movla    .200
movam    tm2_la
```

```
movla    b'01100010'    ;以下为 TMR3 的初始化程序段
movam    tm3_ctl1
movla    .100
movam    tm3_la
```

```
movla    b'00000000'    ;以下是 ADC 的初始化设置
movla    ad_ctl1        ;Bit7:adc 使能/Bit5:模式/Bit1-0 通道选择
movla    b'00000000'
movla    ad_ctl2        ;Bit7:比较结果/Bit1-0:adc 时钟源
movla    b'00000000'
movla    ad_ctl3        ;Bit3-0:PB0-3 复用管脚的选择
movla    .0
movla    ad_dat
```

```
movla    b'10000010'    ;以下是中断设置
movam    irqm            ;Bit1:TM1/Bit2:TM2/Bit3:TM3
                                ;Bit4:PA/Bit6:ADC/Bit7:中断总使能 Bit
clr      irqf
```

```
bs      tm1_ctl1,7
bs      tm2_ctl1,7
;bs     tm3_ctl1,7
nop
lgoto   $
```

display

```
mov      dat_0,a
andla    b'00001111'
movam    portb_dat
```

```
rr      dat_0,m
rr      dat_0,m
rr      dat_0,m
rr      dat_0,m
mov      dat_0,a
andla    b'00001111'
```



movam porta_dat

;

btsc dat_1,0

bs porta_dat,5

btss dat_1,0

bc porta_dat,5

;

btsc dat_1,1

bs porta_dat,6

btss dat_1,1

bc porta_dat,6

;

btsc dat_1,2

bs porta_dat,7

btss dat_1,2

bc porta_dat,7

;

btsc dat_1,3

bs portc_dat,3

btss dat_1,3

bc portc_dat,3

;

btsc dat_1,4

bs portc_dat,2

btss dat_1,4

bc portc_dat,2

ret

;

end

➤ mk7a21p_hw.inc 文档

;-----Define special register(Define SFR) -----

indf equ 0x00

pcl equ 0x01

pch equ 0x02

status equ 0x03

fsr equ 0x04

;

porta_dir equ 0x05

porta_dat equ 0x06

portb_dir equ 0x07



portb_dat	equ	0x08
portc_dir	equ	0x09
portc_dat	equ	0x0a

tm1_ctl1	equ	0x13
tm1_ctl2	equ	0x1f
clr_cnt	equ	0x21
tm1l_la	equ	0x14
tm1h_la	equ	0x15
tm1l_cnt	equ	0x16
tm1h_cnt	equ	0x17

tm2_ctl1	equ	0x18
tm2_ctl2	equ	0x19
tm2_la	equ	0x1a
tm2_cnt	equ	0x1c

tm3_ctl1	equ	0x1e
tm3_la	equ	0x20
tm3_cnt	equ	0x22

irqm	equ	0x25
irqf	equ	0x26

ad_ctl1	equ	0x29
ad_ctl2	equ	0x2a
ad_ctl3	equ	0x2b
ad_dat	equ	0x2d

pa_plu	equ	0x31
pb_plu	equ	0x33
pc_plu	equ	0x35
wake_up	equ	0x3a
wdt_ctl	equ	0x3d
tab_bnk	equ	0x3e
sys_ctl	equ	0x3f

-----Define [status Register] special bit-----

c	equ	0
dc	equ	1
z	equ	2



pd	equ	3
to	equ	4
;-----Define [tm1_ctl1 Register] special bit-----		
pre0	equ	0
pre1	equ	1
pre2	equ	2
edge	equ	3
sur0	equ	4
sur1	equ	5
wr_cnt	equ	6
tm1_en	equ	7
;-----Define [tm1_ctl2 Register] special bit-----		
enc	equ	7
;-----Define [tm2_ctl1 Register] special bit-----		
;pre0	equ	0
;pre1	equ	1
;pre2	equ	2
;edge	equ	3
;sur0	equ	4
;sur1	equ	5
;wr_cnt	equ	6
tm2_en	equ	7
;-----Define [tm2_ctl2 Register] special bit-----		
pos0	equ	0
pos1	equ	1
pos2	equ	2
pos3	equ	3
to_e	equ	5
pwm_os	equ	6
mod	equ	7
;-----Define [tm3_ctl1 Register] special bit-----		
;pre0	equ	0
;pre1	equ	1
;pre2	equ	2
;edge	equ	3
;sur0	equ	4
;sur1	equ	5
;wr_cnt	equ	6
tm3_en	equ	7
;-----Define [irqm Register] special bit-----		



tm1m	equ	1
tm2m	equ	2
tm3m	equ	3
pam	equ	4
adcm	equ	6
intm	equ	7
;-----Define [irqf Register] special bit-----		
tm1f	equ	1
tm2f	equ	2
tm3f	equ	3
paf	equ	4
adcf	equ	6
;-----Define [ad_ctl1 Register] special bit-----		
chsel0	equ	0
chsel1	equ	1
mode	equ	5
en	equ	7
;-----Define [ad_ctl2 Register] special bit-----		
cksel0	equ	0
cksel1	equ	1
rsut	equ	7
;-----Define [ad_ctl3 Register] special bit-----		
pbsel0	equ	0
pbsel1	equ	1
pbsel2	equ	2
;-----Define [pa_plu Register] special bit-----		
ua0	equ	0
ua1	equ	1
ua2	equ	2
ua3	equ	3
ua4	equ	4
ua5	equ	5
ua6	equ	6
ua7	equ	7
;-----Define [pb_plu Register] special bit-----		
ub0	equ	0
ub1	equ	1
ub2	equ	2
ub3	equ	3
;-----Define [pc_plu Register] special bit-----		



uc0	equ	0
uc2	equ	2
uc3	equ	3
;-----Define [wake_up Register] special bit-----		
en0	equ	0
en1	equ	1
en2	equ	2
en3	equ	3
en4	equ	4
en5	equ	5
en6	equ	6
en7	equ	7
;-----Define [wdt_ctl Register] special bit-----		
;pre0	equ	0
;pre1	equ	1
;pre2	equ	2
wdten	equ	7
;-----Define [tab_bnk Register] special bit-----		
Bnk0	equ	0
bnk1	equ	1
bnk2	equ	3
;-----Define [sys_ctl Register] special bit-----		
stp0	equ	0
stp1	equ	1
clks	equ	7
;-----		