- 产品名称
  - MK7A21P
- 例案标题

  MK7A21P 的 PWM 使用说明
- 简介

  MK7A21P 提供了一个与PC0 共用引脚的PWM端口，可以由寄存器来控制。通过对TM2和TM3寄存器进行设置，PWM功能可输出占空比可调的波形。

  本案例MK7A21P MCU的PWM使用方法。

  设置PWM的方法如下：

  a.设置TM2预分频比
  b.设置TM2计数边沿
  c.设置TM2时钟来源
  d.设置PWM预分频比
  e.设置TM2为PWM模式
  f.设置TM2预分频比
  g.设置TM3计数边沿
  h.设置TM3时钟来源
  i.设置PWM的周期
  j.设置PWM的占空比

  该案例采用MK7A21P，使用PWM/PC0为PWM输出，电路图如图1所示。

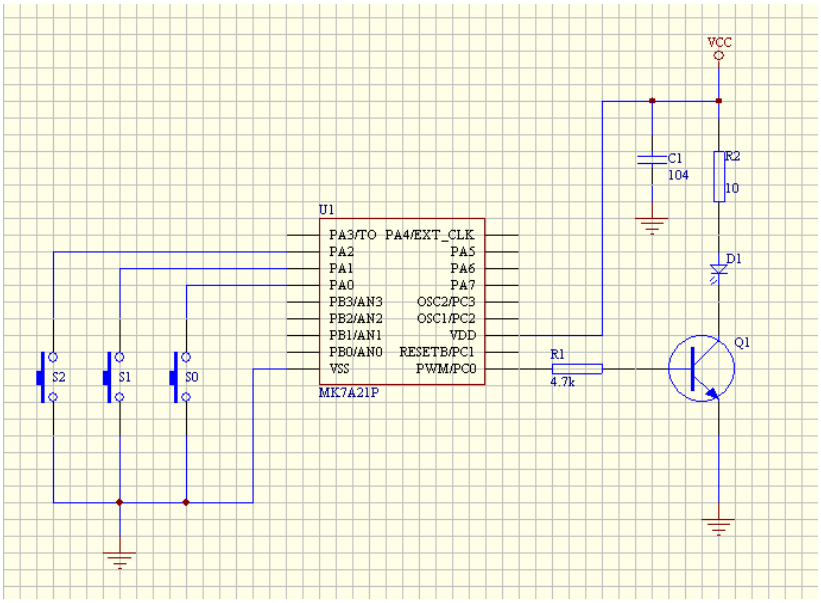


图 1

本例中使用3个按键，当某一按键被按下并松开时，其所对应的数据码将在PWM口中发送。其编码格式为：1 位起始信号+8 位数据信号，且只发送一次。

A.TM2_CTL1 ($18H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM2_CTL1	TM2_EN	WR_CNT	SUR1	SUR0	EDGE	PRE2	PRE1	PRE0

bit7:TM2使能位

0：TM2关闭

1：TM2打开

bit6:缓存数据写入计数器控制

0：不写入

1：写入

bit5,4:时钟来源控制

00：外部时钟输入

01：晶体

10：RC

11：不使用

bit3:时钟边沿控制位

0：上升边沿计数

1：下降边沿计数

bit2,1,0:设置TM2预分频比

Bit2	Bit1	Bit0	TM2 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.TM2_CTL2 ($19H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM2_CTL2	MOD	PWM_OS	TO_E	--	POS3	POS2	POS1	POS0

bit7: 模式控制位

0：定时器模式

1：PWM模式

bit6:时钟边沿控制位

0：上升边沿计数

1：下降边沿计数

bit5:定时器输出控制位

0：设置PA3为普通IO口

1：设置PA3为定时器输出

bit3,2,1,0: PWM预分频比控制位

Bit3	Bit2	Bit1	Bit0	PWM Poscaler rate
POS3	POS2	POS1	POS0	
0	0	0	0	1:1
0	0	0	1	1:2
0	0	1	0	1:3
. .	. .	. .	. .	. .
1	1	1	0	1:15
1	1	1	1	1:16

A.TM3_CTL ($1EH):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM3_CTL	TMR3_EN	WR_CNT	SUR1	SUR0	EDGE	PRE2	PRE1	PRE0

bit7:TM3使能位

0：TM3关闭

1：TM3打开

bit6:缓存数据写入计数器控制

0：不写入

1：写入

bit5,4:时钟来源控制

00：外部时钟输入

01：晶体

10：RC

11：不使用

bit3:时钟边沿控制位

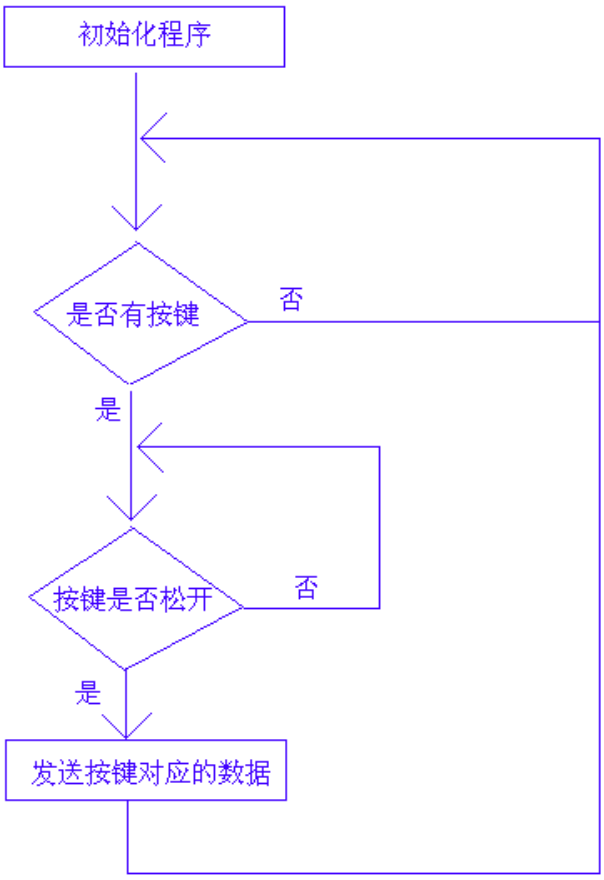
0：上升边沿计数

1：下降边沿计数

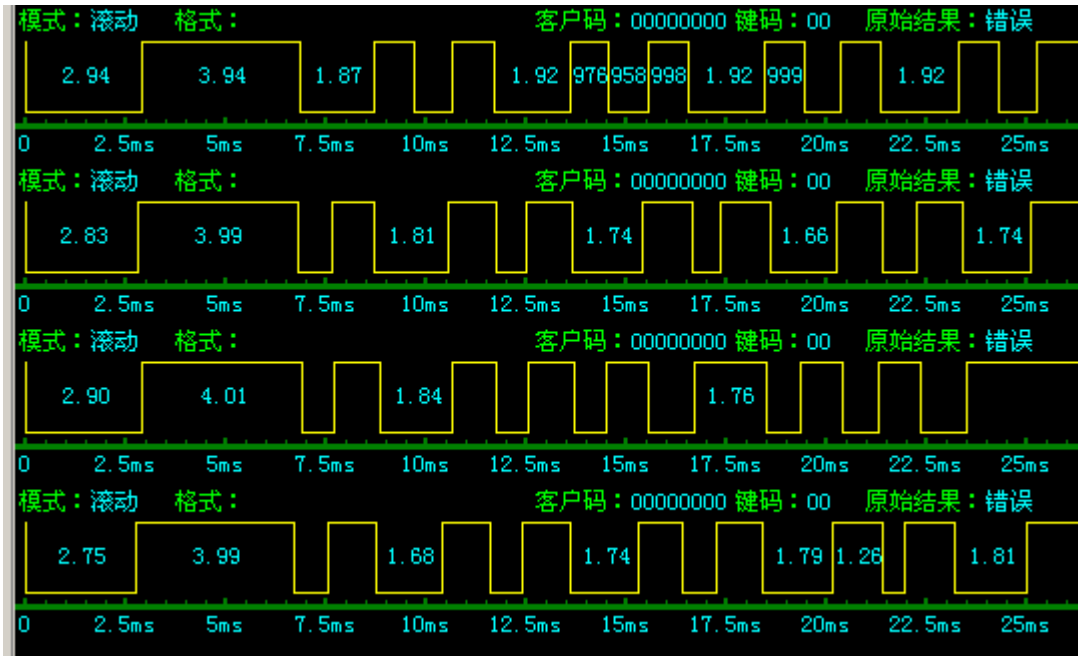
bit2,1,0:设置TM3预分频比

Bit2	Bit1	Bit0	TM3 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

- 程序流程图



- 本程序经红外遥控编码分析仪所得到的波形如下：



- DEMO程序
  - 程序说明：

    以下程序只是一个简单的使用介绍，其编码方式非常简单，并且只能完成8bit 数据的发送，不能直接应用到实际的系统中。其主要目的不是编码和解码的算法，而是通过这个例程来介绍MK7A21P的PWM使用使用方法。
  - 汇编程序文档

```
;-----------------------------------
#include "mk7a21p.inc"
length_counter equ 40h          ;延时
delay_counter1 equ 41h          ;延时
delay_counter2 equ 42h          ;延时
delay_counter3 equ 43h          ;延时
s0_data        equ 44h          ;按键S0的发送数据码
s1_data        equ 45h          ;按键S1的发送数据码
s2_data        equ 46h          ;按键S2的发送数据码
s_temp         equ 47h          ;发送数据码暂存
byte_counter   equ 48h          ;发送数据位数
;-----------------------------------
        org 00h
        lcall initial           ;调用初始化程序
main_loop:
```

```
        lcall scan_key_loop     ;调用按键扫描程序
        movla 04fh
        movam length_counter
        lcall delay             ;延时
        lgoto main_loop
;-----------------------------------
;延时子程序
delay:
       mov length_counter,a
       movam delay_counter1
loop4: decsz delay_counter1,m
       lgoto loop1
       lgoto return
loop1:
       mov length_counter,a
       movam delay_counter2
loop6: decsz delay_counter2,m
       lgoto loop2
       lgoto loop4
loop2:
       mov length_counter,a
       movam delay_counter3
loop3:
       nop
       decsz delay_counter3,m
       lgoto loop3
       lgoto loop6
return:
       ret
;-----------------------------------
;按键扫描子程序
scan_key_loop:
        btss pa_dat,b0          ;扫描S0按键
        lgoto scan_key_loop4
        lgoto scan_key_loop1
scan_key_loop4:
        btss pa_dat,b0          ;判断S0是否松开
        lgoto scan_key_loop4
        lcall s0_send           ;调用S0按键信号处理子程序
        lgoto scan_key_loop3
```

```
scan_key_loop1:
        btss pa_dat,b1          ;扫描S1按键
        lgoto scan_key_loop5
        lgoto scan_key_loop2
scan_key_loop5:
        btss pa_dat,b1          ;判断S1是否松开
        lgoto scan_key_loop5
        lcall s1_send           ;调用S1按键信号处理子程序
        lgoto scan_key_loop3
scan_key_loop2:
        btss pa_dat,b2          ;扫描S2按键
        lgoto scan_key_loop6
        lgoto scan_key_loop
scan_key_loop6:
        btss pa_dat,b2          ;判断S2是否松开
        lgoto scan_key_loop6
        lcall s2_send           ;调用S2按键信号处理子程序
        lgoto scan_key_loop3
scan_key_loop3:
        ret
;-----------------------------------
;S0按键信号处理子程序
s0_send:
        lcall lead_send         ;调用起始信号发射子程序
        mov s0_data,a
        movam s_temp
        movla 08h               ;发送数据位数为8，即一字节
        movam byte_counter
s0_send_loop4:
        btsc s_temp,b0          ;判断第0位是1还是0
        lgoto s0_send_loop3
        lgoto s0_send_loop2
s0_send_loop3:
        rr s_temp,m             ;数据右移
        lcall bit1_send         ;调用发射BIT1子程序
        decsz byte_counter,m
        lgoto s0_send_loop4
        lgoto s0_send_loop1
s0_send_loop2:
        rr s_temp,m             ;数据右移
```

```
        lcall bit0_send         ;调用发射BIT0子程序
        decsz byte_counter,m
        lgoto s0_send_loop4
        lgoto s0_send_loop1
s0_send_loop1:
        bc   tm2_ctl1,b7
        ret
;-----------------------------------
;S1按键信号处理子程序
s1_send:
        lcall lead_send         ;调用起始信号发射子程序
        mov s1_data,a
        movam s_temp
        movla 08h               ;发送数据位数为8，即一字节
        movam byte_counter
s1_send_loop4:
        btsc s_temp,b0          ;判断第0位是1还是0
        lgoto s1_send_loop3
        lgoto s1_send_loop2
s1_send_loop3:
        rr s_temp,m             ;数据右移
        lcall bit1_send         ;调用发射BIT1子程序
        decsz byte_counter,m
        lgoto s1_send_loop4
        lgoto s1_send_loop1
s1_send_loop2:
        rr s_temp,m             ;数据右移
        lcall bit0_send         ;调用发射BIT0子程序
        decsz byte_counter,m
        lgoto s1_send_loop4
        lgoto s1_send_loop1
s1_send_loop1:
        bc   tm2_ctl1,b7
        ret
;-----------------------------------
;S2按键信号处理子程序
s2_send:
        lcall lead_send         ;调用起始信号发射子程序
        mov s2_data,a
        movam s_temp
```

```
        movla 08h               ;发送数据位数为8，即一字节
        movam byte_counter
s2_send_loop4:
        btsc s_temp,b0          ;判断第0位是1还是0
        lgoto s2_send_loop3
        lgoto s2_send_loop2
s2_send_loop3:
        rr s_temp,m             ;数据右移
        lcall bit1_send         ;调用发射BIT1子程序
        decsz byte_counter,m
        lgoto s2_send_loop4
        lgoto s2_send_loop1
s2_send_loop2:
        rr s_temp,m             ;数据右移
        lcall bit0_send         ;调用发射BIT0子程序
        decsz byte_counter,m
        lgoto s2_send_loop4
        lgoto s2_send_loop1
s2_send_loop1:
        bc   tm2_ctl1,b7
        ret
;-----------------------------------
; 起始信号发射子程序
lead_send:
        bs     tm2_ctl1,b7
        movla  08h
        movam  length_counter
        lcall  delay
        lcall  delay
        lcall  delay
        bc     tm2_ctl1,b7
        movla  08h
        movam  length_counter
        lcall  delay
        lcall  delay
        lcall  delay
        ret
;-----------------------------------
;发射BIT0子程序
bit0_send:
```

```
        bc      tm2_ctl1,b7
        movla   08h
        movam   length_counter
        lcall   delay
        bs      tm2_ctl1,b7
        movla   08h
        movam   length_counter
        lcall   delay
        ret
;-----------------------------------
;发射BIT1子程序
bit1_send:
        bc      tm2_ctl1,b7
        movla   08h
        movam   length_counter
        lcall   delay
        bs      tm2_ctl1,b7
        movla   08h
        movam   length_counter
        lcall   delay
        lcall   delay
        ret
;-----------------------------------
;初始化程序
initial:
;按键S0，S1，S2的发送数据码设置
        movla b'10010010'       ;S0被按下时，b'10010010'将被发送
        movam s0_data
        movla b'10101010'       ;S1被按下时，b'10101010'将被发送
        movam s1_data
        movla b'01010101'       ;S2被按下时，b'01010101'将被发送
        movam s2_data
;IO方向及上拉电阻设置
        bs  pa_dir,b0
        bs  pa_dir,b1
        bs  pa_dir,b2
        bs  pa_plu,b0
        bs  pa_plu,b1
        bs  pa_plu,b2
        bc  pc_dir,b0
```

```
;TM2及TM3的PWM设置
        bs tm2_ctl1,b0
        bs tm2_ctl1,b1
        bc tm2_ctl1,b2          ;设置预分频比为1：8
        bc tm2_ctl1,b3          ;设置上升边沿
        bc tm2_ctl1,b4
        bs tm2_ctl1,b5          ;设置时钟来源为RC
        bs tm2_ctl1,b6
        bc tm2_ctl1,b7          ;关闭PWM
        bc tm2_ctl2,b0
        bc tm2_ctl2,b1
        bc tm2_ctl2,b2          ;设置PWM预分频比为1:1
        bc tm2_ctl2,b6
        bs tm2_ctl2,b7          ;设置TM2为PWM模式
        bs tm3_ctl,b0
        bs tm3_ctl,b1
        bc tm3_ctl,b2           ;设置预分频比为1：8
        bc tm3_ctl,b3           ;设置上升边沿
        bc tm3_ctl,b4
        bs tm3_ctl,b5           ;设置时钟来源为RC
        bs tm3_ctl,b6
        movla 0ch               ;设置PWM的周期
        movam tm2_la
        movla 05h               ;设置PWM的占空比
        movam tm3_la
        ret
;---------------------------------
        end
```

➢ mk7a21p.inc文档

```
INDF            EQU         0x00
PCL             EQU         0x01
PCH             EQU         0x02
STATUS          EQU         0x03
FSR             EQU         0x04

PA_DIR          EQU         0x05
PA_DAT          EQU         0x06
PB_DIR          EQU         0x07
PB_DAT          EQU         0x08
PC_DIR          EQU         0x09
```

```
PC_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_CTL         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

;=================================================================
;Special Purpose Register bit define
;-----------------------------------------------------------------
; Define [STATUS Register] special bit
C               EQU          0
```

```
DC              EQU         1
Z               EQU         2
PD              EQU         3
TO              EQU         4

; Define [PA_DIR Register] special bit
IOA0            EQU         0
IOA1            EQU         1
IOA2            EQU         2
IOA3            EQU         3
IOA4            EQU         4
IOA5            EQU         5
IOA6            EQU         6
IOA7            EQU         7

; Define [PA_DAT Register] special bit
DA0             EQU         0
DA1             EQU         1
DA2             EQU         2
DA3             EQU         3
DA4             EQU         4
DA5             EQU         5
DA6             EQU         6
DA7             EQU         7

; Define [PB_DIR Register] special bit
IOB0            EQU         0
IOB1            EQU         1
IOB2            EQU         2
IOB3            EQU         3

; Define [PB_DAT Register] special bit
DB0             EQU         0
DB1             EQU         1
DB2             EQU         2
DB3             EQU         3

; Define [PC_DIR Register] special bit
IOC0            EQU         0
IOC2            EQU         2
```

```
IOC3            EQU         3

; Define [PC_DAT Register] special bit
DC0             EQU         0
DC1             EQU         1
DC2             EQU         2
DC3             EQU         3

; Define [TM1_CTL1 Register] special bit
PRE0            EQU         0           ; share with
TM1_CTL2/TM3_CTL1/WDT_CTL
PRE1            EQU         1           ; share with
TM1_CTL2/TM3_CTL1/WDT_CTL
PRE2            EQU         2           ; share with
TM1_CTL2/TM3_CTL1/WDT_CTL
EDGE            EQU         3           ; share with
TM1_CTL2/TM3_CTL1/WDT_CTL
SUR0            EQU         4           ; share with
TM1_CTL2/TM3_CTL1/WDT_CTL
SUR1            EQU         5           ; share with
TM1_CTL2/TM3_CTL1/WDT_CTL
WR_CNT          EQU         6           ; share with
TM1_CTL2/TM3_CTL1/WDT_CTL
TM1_EN          EQU         7

; Define [TM1_CTL2 Register] special bit
ENC             EQU         7

; Define [CLR_CNT Register] special bit
CLR_CPT         EQU         7           ; Clear capture counter

; 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         2

; Define [SYS_CTL Register] special bit
STP0            EQU         0
STP1            EQU         1
CLKS            EQU         7
```