



● 产品名称

➤ MK7A21P

● 例案标题

MK7A21P 的 AD 转换功能使用说明

● 简介

MK7A21P 提供了一个与PortB共用引脚的AD转换模拟输入端口，用户可以通过寄存器AD_CTL3的Bit2~1来进行分配，下面是AD_CTL3寄存器功能结构表。

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AD_CTL3	--	--	--	--	--	PBSEL2	PBSEL1	PBSEL0

用户可以对照下表选择所需要的AD转换模拟输入端口。

Bit2	Bit1	Bit0	PB3~PB0 configurations
PBSEL2	PBSEL1	PBSEL0	
0	0	0	PB3, PB2, PB1, PB0
0	0	1	PB3, PB2, PB1, AN0
0	1	0	PB3, PB2, AN1, AN0
0	1	1	PB3, AN2, AN1, AN0
1	X	X	AN3, AN2, AN1, AN0

对MK7A21P来说，ADC功能分为“转换”和“比较”两种模式。这两种模式是由AD_CTL1寄存器的MODE(Bit5)位控制的。当MODE为0时为转换模式，也就是将对应输入端口中的模拟量转换成数据，存入AD_DAT寄存器里面；当MODE为1时为比较模式，也就是将对应输入端口中的模拟量转换成数据，并与事先存放在AD_DAT寄存器里面的数据进行比较，比较之后的结果通过AD_CTL2寄存器的RSUT(Bit7)位的值来体现，下面分别是AD_CTL1和AD_CTL2的功能结构表。

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AD_CTL1	EN	--	MODE	--	--	--	CHSEL1	CHSEL0

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AD_CTL2	RSUT	--	--	--	--	--	CKSEL1	CKSEL0

在对多个模拟输入通道进行AD转换操作的时候，用户可以根据下表来选择当前的转换通道。表中的CHSEL1和CHSEL0分别位于AD_CTL1寄存器的



Bit1和Bit0。

Bit1	Bit0	Input channel
CHSEL1	CHSEL0	
0	0	Channel 0, PB0
0	1	Channel 1, PB1
1	0	Channel 2, PB2
1	1	Channel 3, PB3

另外，在进行转换之前，用户应该根据实际需要由下表选择适当的转换时钟。表中的CKSEL1和CKSEL0分别位于AD_CTL2寄存器的Bit1和Bit0。

Bit1	Bit0	Conversion clock
CKSEL1	CKSEL0	
0	0	System clock X2
0	1	System clock X8
1	0	System clock X32
1	1	System clock X128

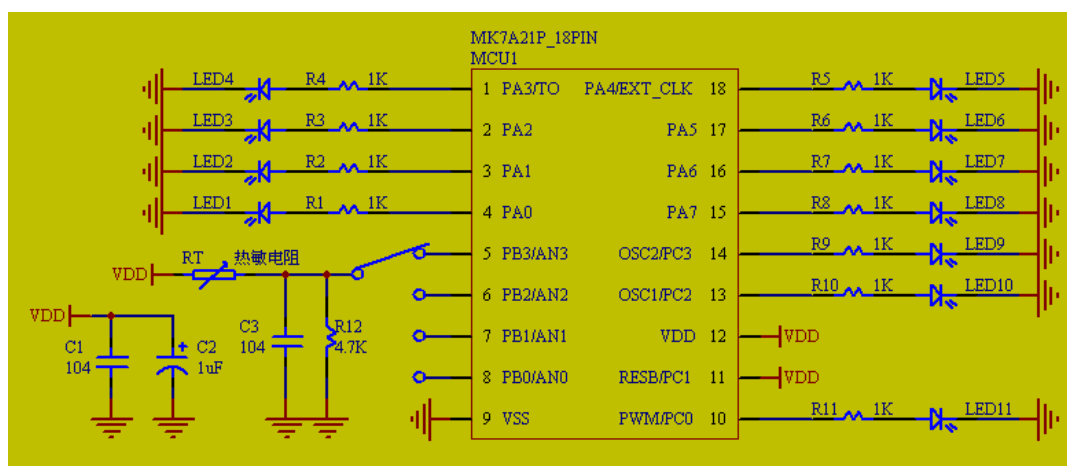


图1 MK7A21P的AD转换试验线路图

为了让用户更深入的了解MK7A21P的AD转换功能的使用方法，我们提供了如下实例。其线路图见图1。基本功能如下：



- ◆ 芯片每隔2.5s从4个模拟输入通道中按照AN0、AN1、AN2和AN3的顺序选择其中的一个通道进行AD转换，并将转换所得的数据送PortA端口显示。
- ◆ 图中的LED9~11的亮灭与当前转换的通道相对应，其关系见下表：

通道	LED9	LED10	LED11
AN0	灭	灭	灭
AN1	灭	灭	亮
AN2	灭	亮	灭
AN3	灭	亮	亮

● 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 ;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=...IO... ;...IO...,...timer source...  
;-----  
a_buf equ 0x40  
status_buf equ 0x41  
dat_0 equ 0x42  
dat_1 equ 0x43  
sec_r equ 0x44  
tm_r equ 0x45  
;-----  
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  
;-----
```



```
int                                     org      0x010

                                     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
                                     ;-----
                                     lgoto    int_end

                                     ;-----

int_tm2

                                     bc        irqf,tm2f
                                     ;-----
                                     ;TM2 中断服务程序
                                     ;-----
                                     lgoto    int_end

                                     ;-----

int_tm3

                                     bc        irqf,tm3f
```



```

;-----
;TM3 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
mov           ad_dat,a
movam         porta_dat
clr           portc_dat
btsc          sec_r,0
bs            portc_dat,0
btsc          sec_r,1
bs            portc_dat,2
btsc          sec_r,2
bs            portc_dat,3
;-----
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'00000000'
movam         porta_dir        ;PA0-7 输出

```



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'11111111'	
movam	portb_dir	;PB0-3 输入
movla	b'11110000'	
movam	portb_dat	;PB0-3 状态
movla	b'11111111'	
movam	pb_plu	;PB0-3 上拉使能
;-----		
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'01100010'	;以下为 TMR1 的初始化程序段
movam	tm1_ctl1	;TM1 用作捕捉模式,预分频 1:4
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'01000010'	
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  
movam      ad_ctl1        ;Bit7:adc 使能/Bit5:模式/Bit1-0 通道选择  
movla      b'00000001'  
movam      ad_ctl2        ;Bit7:比较结果/Bit1-0:adc 时钟倍频  
movla      b'00000100'  
movam      ad_ctl3        ;Bit3-0:PB0-3 复用管脚的选择,做 ADC 用  
movla      .0  
movam      ad_dat  
-----  
movla      b'11000000'    ;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      ;启动 tmr1  
;bs        tm2_ctl1,7  
;bs        tm3_ctl1,7  
-----  
clr        sec_r  
-----  
loop  
        btss      irqf,tm1f  
        lgoto     $-1  
        bc        irqf,tm1f  
-----  
inc        tm_r,m  
movla      .50  
xor        tm_r,a  
btss      status,z  
lgoto     loop  
clr        tm_r  
-----  
clr        porta_dat  
clr        portc_dat
```



```
-----  
inc          sec_r,m  
movla        .4  
sub          sec_r,a  
btsc         status,c  
clr          sec_r  
-----  
movla        b'00000111'  
movam        tab_bnk  
tabrdl       sec_r  
movam        ad_ctl1      ;根据查询值,选择将要进行 ad 转换的通道  
-----  
lgoto        loop
```

```
-----  
org          0x700  
dw           0080h  
dw           0081h  
dw           0082h  
dw           0083h  
-----
```

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
;-----		