

## ● 产品名称

➤ MK7A21P

## ● 标题

如何使用 MK7A21P 的 PortA 中断功能

## ● 简介

MK7A21P 芯片具备 PortA 中断功能，这给需要处理外部设备请求的用户提供了极大的方便。为了让用户更深入的了解这种功能，我们提供了如下试验例程。

本例程的线路图见图 1。基本功能是利用 PortA 中断功能来检测图中的 8 个按键，并将按键的编号通过图中的 4 个 LED 显示出来。

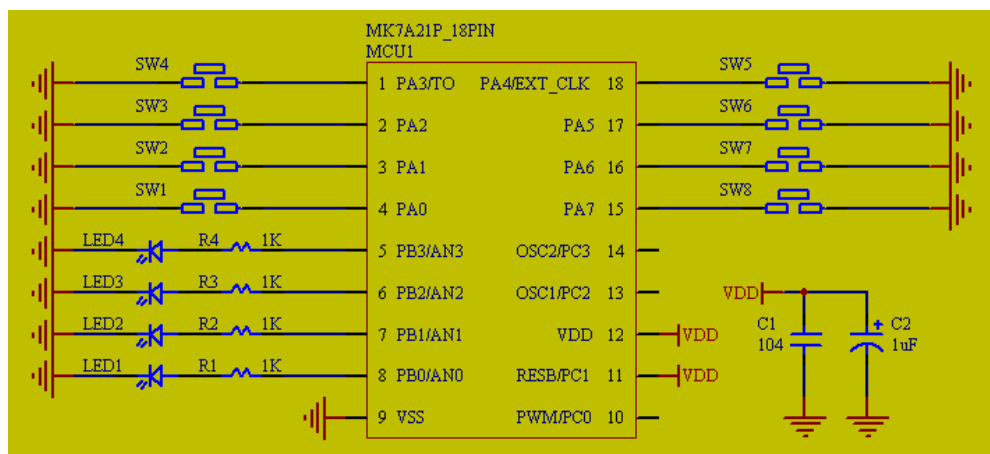


图 1 MK7A21P 的 PortA 中断试验线路图

## ● 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 ;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
flag       equ      0x42
del_r0     equ      0x43
del_r1     equ      0x44
key_buf    equ      0x45

;-----
#define     flag_key  flag,0

;-----
                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,paf
                lgoto    int_pa     ;进入 PortA 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      ;推出中断时不立即开中断

;-----
int_pa

                bc       irqf,paf

;-----
;PortA 中断服务程序
                bs       flag_key

;-----
```



```
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'11111111'
    movam   porta_dir    ;PA0-7 输入
    movla   b'00000000'
    movam   porta_dat    ;PA0-7 状态
    movla   b'11111111'
    movam   pa_plu       ;PA0-7 上拉使能
    movla   b'11111111'
    movam   wake_up      ;PA0-7 唤醒使能
;-----
    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'10010000'  ;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
;-----
```



loop

```
btss    flag_key
lgoto   $-1      ;等待按键标志出现
bc      flag_key  ;清除按键标志
```

-----

```
com      porta_dat,a
movam    key_buf
btsc     status,z
lgoto    delay    ;按键放开不再进行处理
```

-----

```
clra
btsc     key_buf,0
movla    .1
btsc     key_buf,1
movla    .2
btsc     key_buf,2
movla    .3
btsc     key_buf,3
movla    .4
btsc     key_buf,4
movla    .5
btsc     key_buf,5
movla    .6
btsc     key_buf,6
movla    .7
btsc     key_buf,7
movla    .8
movam    portb_dat    ;送显示
```

-----

delay

```
incsz    del_r0,m    ;延时消抖
lgoto    delay
incsz    del_r1,m
lgoto    delay
```

-----

```
lgoto    loop
```

-----

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
;-----		