● 产品名称

MK7XXXX系列芯片

● 例案标题

如何运用MK7XXXX系列芯片来扫描 4x8 按键

● 简介

对于大多数电子产品而言，按键扫描电路是必不可少的，因此设计一个稳定的按键扫描电路，就显得尤为重要。针对此目的，我们介绍以下案例来供大家参考。

该案例用到 12 个 I/O，这里采用 MK7A10P 芯片来设计，电路图如图 1 所示。

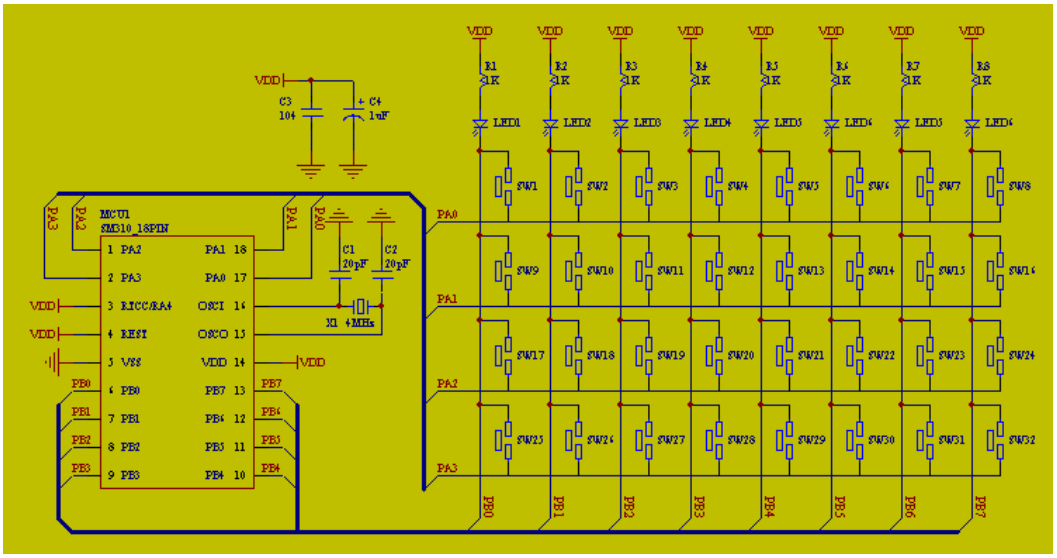


图 1　　MK7A10P 4x8 矩阵 按键扫描电路图

芯片上电后，所有 LED 全灭。如有按键按下，则相应的 LED 将点亮，按键与 LED 亮灭的对应关系见表 1。

案例中 PortA0～PortA3 为输出，PortB0～PortB7 则分时复用，既用作检测按键信号的输入脚，又用作点亮 LED 的输出脚，由于 PortB 端口有内部上拉功能，所以不用外接上拉电阻，启动 PortB 内部上拉需要进行如下设置：

1. 将 CONFIG 的 Bit7 设置为 1
2. PortB 设为输入模式

其中 CONFIG 寄存器为配置寄存器，内含 8 个 bit，其各 bit 的定义下：

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CONFIG	TYPE	RTCE	LV1	LV0	CPT	WDTE	FOSC1	FOSC0

Bit 1	Bit 0	OSC Type	Frequency Range
0	0	LS (low speed)	32~455KHz
0	1	NS(Normal speed)	1M~8MHz
1	0	HS (high speed)	10~40MHz
1	1	RC	32K~ 10MHz

Bit 2	Description
0	Watchdog timer disable
1	Watchdog timer enable

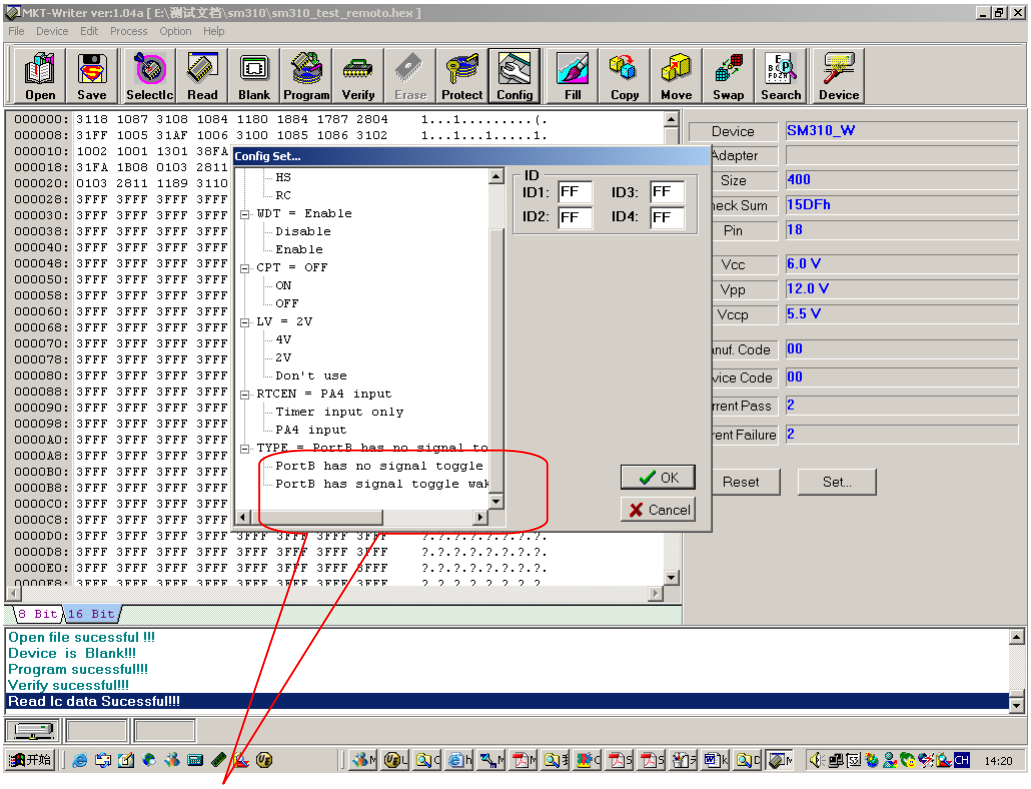
Bit 3	Description
0	Code Protection ON
1	Code Protection OFF

Bit 5	Bit 4	Detected Voltage Level
0	0	4V
0	1	No use
1	0	2V
1	1	No use

Bit 6	Description
0	Timer Input Only
1	PA4 Input

Bit 7	Description
0	Port B has no signal toggle wake up function
1	Port B has signal toggle wake up function

配置寄存器 CONFIG 的设定界面如图 2 所示。



要想启动 PortB 端口的上拉功能，此处必须选 Type=PortB has signal toggle wake up function

图 2 CONFIG 设置

由于程序中用到 Tmr0，所以有介绍一下 Tmr0 的必要， Tmr0 的预分频大小是由选择寄存器 Select 来控制的，Select 寄存器也是内含 8 个 bit，其各 bit 的定义下：

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SELECT	X	X	SUR0	EDGE0	PSA	PS2	PS1	PS0

Bit 2	Bit 1	Bit 0	Description	
PS2	PS1	PS0	TMR0 rate	WDT rate
0	0	0	1:2	1:1
0	0	1	1:4	1:2
0	1	0	1:8	1:4
0	1	1	1:16	1:8
1	0	0	1:32	1:16
1	0	1	1:64	1:32
1	1	0	1:128	1:64
1	1	1	1:256	1:128

Bit 3	Description
0	Prescaler assigned to TMR0
1	Prescaler assigned to WDT

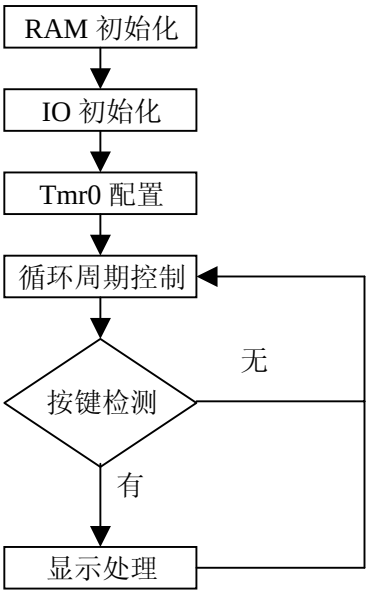
Bit 4	Description
0	increment when rising edge (Lo to Hi transition) on external clock
1	increment when falling edge (Hi to Lo transition) on external clock

Bit 5	Description
0	TMR0 clock source is (System Clock/4)
1	TMR0 clock source is RTCC input

在本程序中 Tmr0 的预分频比设为 1:8，因此只要将二进制数 00000010 输入到 Select 寄存器里面，执行如下两条指令即可：

```
movla       b'00000010'
select
```

- 程序流程图



按键	LED8	LED7	LED6	LED5	LED4	LED3	LED2	LED1
SW1	灭	灭	灭	灭	灭	灭	灭	亮
SW2	灭	灭	灭	灭	灭	灭	亮	灭
SW3	灭	灭	灭	灭	灭	灭	亮	亮
SW4	灭	灭	灭	灭	灭	亮	灭	灭
SW5	灭	灭	灭	灭	灭	亮	灭	亮
SW6	灭	灭	灭	灭	灭	亮	亮	灭
SW7	灭	灭	灭	灭	灭	亮	亮	亮
SW8	灭	灭	灭	灭	亮	灭	灭	灭
SW9	灭	灭	灭	灭	亮	灭	灭	亮
SW10	灭	灭	灭	灭	亮	灭	亮	灭
SW11	灭	灭	灭	灭	亮	灭	亮	亮
SW12	灭	灭	灭	灭	亮	亮	灭	灭
SW13	灭	灭	灭	灭	亮	亮	灭	亮
SW14	灭	灭	灭	灭	亮	亮	亮	灭
SW15	灭	灭	灭	灭	亮	亮	亮	亮
SW16	灭	灭	灭	亮	灭	灭	灭	灭
SW17	灭	灭	灭	亮	灭	灭	灭	亮
SW18	灭	灭	灭	亮	灭	灭	亮	灭
SW19	灭	灭	灭	亮	灭	灭	亮	亮
SW20	灭	灭	灭	亮	灭	亮	灭	灭
SW21	灭	灭	灭	亮	灭	亮	灭	亮
SW22	灭	灭	灭	亮	灭	亮	亮	灭
SW23	灭	灭	灭	亮	灭	亮	亮	亮
SW24	灭	灭	灭	亮	亮	灭	灭	灭
SW25	灭	灭	灭	亮	亮	灭	灭	亮
SW26	灭	灭	灭	亮	亮	灭	亮	灭
SW27	灭	灭	灭	亮	亮	灭	亮	亮
SW28	灭	灭	灭	亮	亮	亮	灭	灭
SW29	灭	灭	灭	亮	亮	亮	灭	亮
SW30	灭	灭	灭	亮	亮	亮	亮	灭
SW31	灭	灭	灭	亮	亮	亮	亮	亮
SW32	灭	灭	亮	灭	灭	灭	灭	灭

表 1　　按键与 LED 亮灭的对应关系表

● DEMO 程序

➢ 汇编程序文档

```
;-------------------------------------------------------------------------------------------------------------------
#include    "mk7a10p_hw.inc"            ;编译该文档需包含" mk7a10p_hw.inc "文件
;-------------------------------------------------------------------------------------------------------------------
;芯片型号（mk7a10p）
;-------------------------------------------------------------------------------------------------------------------
;配置寄存器设置说明（CONFIG）
;1----FOSC=NS                    ;LS,NS,HS,RC
;2----WDTE=Enable                ;Enable,Disable
;3----CPT=OFF                    ;ON,OFF
;4----LV=2v                      ;4V,2V,Don't use
;5----RTCEN=PA4 input            ;Timer input only,PA4 input
;6----TYPE=... has ... wake up ...  ;... has no ... wake up ...
                                 ;... has ... wake up ...
;-------------------------------------------------------------------------------------------------------------------
signal_new          equ     0x07    ;存储当前的按键信号
signal_old          equ     0x08    ;存储上次的按键信号
key_r               equ     0x09    ;检测信号的次数
row                 equ     0x0a    ;行
tier                equ     0x0b    ;列
key_val             equ     0x0c    ;键值
del_r               equ     0x0d    ;延时
flag                equ     0x0e    ;标志
pb_buf              equ     0x0f    ;rb 备用
;-------------------------------------------------------------------------------------------------------------------
#define             flag_key    flag,0      ;按键未松开标志
;-------------------------------------------------------------------------------------------------------------------
                    org     0x3ff       ;mk7a10p 的复位向量地址定义
                    lgoto   main        ;跳转到主程序入口
;-------------------------------------------------------------------------------------------------------------------
            org     0x000
row_table
            add     pc,m
            retla   b'11111110'     ;0 行
            retla   b'11111101'     ;1 行
            retla   b'11111011'     ;2 行
            retla   b'11110111'     ;3 行
;-------------------------------------------------------------------------------------------------------------------
```

```
key_table
        add     pc,m
        retla   .24     ;3 行
        retla   .0      ;0 行
        retla   .8      ;1 行
        retla   .16     ;2 行
;-----------------------------------------------------------------------------------------------------------------------
led_table
        add     pc,m
        retla   .1      ;SW1
        retla   .2      ;SW2
        retla   .3      ;SW3
        retla   .4      ;SW4
        retla   .5      ;SW5
        retla   .6      ;SW6
        retla   .7      ;SW7
        retla   .8      ;SW8
        retla   .9      ;SW9
        retla   .10     ;SW10
        retla   .11     ;SW11
        retla   .12     ;SW12
        retla   .13     ;SW13
        retla   .14     ;SW14
        retla   .15     ;SW15
        retla   .16     ;SW16
        retla   .17     ;SW17
        retla   .18     ;SW18
        retla   .19     ;SW19
        retla   .20     ;SW20
        retla   .21     ;SW21
        retla   .22     ;SW22
        retla   .23     ;SW23
        retla   .24     ;SW24
        retla   .25     ;SW25
        retla   .26     ;SW26
        retla   .27     ;SW27
        retla   .28     ;SW28
        retla   .29     ;SW29
        retla   .30     ;SW30
        retla   .31     ;SW31
```

```
        retla   .32     ;SW32
;-------------------------------------------------------------------------------------------------------
        org     0x100   ;主程序入口地址定义
main
        movla   0x07
        movam   fsr     ;将 0x07 送入 fsr 寄存器
        ;---------------------------------------------------------------------------------------------
clear_ram
        ;利用 indf 和 fsr 来进行间接寻址
        ;对 0x07-0x1f 的 RAM 进行 clear
        clr     indf
        mov     fsr,a
        andla   b'00011111'     ;将无关的数据虑除
        xorla   0x1f
        btsc    status,z
        lgoto   $+3
        inc     fsr,m
        lgoto   clear_ram
        clr     fsr             ;使用 fsr 时要注意 bank 的归位
        ;---------------------------------------------------------------------------------------------
        ;I/O 端口方向及状态设定
        movla   b'11110000'
        iodir   porta           ;PortA0-PortA3 为 output
        movla   b'00000000'
        iodir   portb           ;PortB0-PortB7 为 output
        clr     porta
        movla   0xff
        movam   portb           ;PortB 输出 High
        ;---------------------------------------------------------------------------------------------
        ;配置 TMR0,预分频比为 1:8
        ;TMR0 初始值为 0
        movla   b'00000010'
        select
        clr     tmr0
;-------------------------------------------------------------------------------------------------------
loop    ;程序循环入口
        ;程序循环周期控制处（T=2ms）
        clrwdt
        movla   .250
        xor     tmr0,a
```

```
                btss        status,z
                lgoto       $-4
                clr         tmr0
;-------------------------------------------------------------------------------------------------------------
scan_key    ;4*8 按键矩阵扫描处
                ;PortB 设为输入，为检测按键
                ;信号作准备
                movla               b'11111111'
                iodir               portb
                ;-------------------------------------------------------------------------------------------------------------
                ;查表将相应行对应的
                ;扫描值输入 PortA
                mov         row,a
                lcall       row_table
                movam       porta
                ;-------------------------------------------------------------------------------------------------------------
                ;延时等待 PortB 口信号稳定
                movla       .250
                movam       del_r
                nop
                nop
                decsz       del_r,m
                lgoto       $-3
                ;-------------------------------------------------------------------------------------------------------------
                ;获取 PortB 端口信号,并与上次
                ;信号比较,相同有效,否则无效
                ;同一行连续检测到有效信号达
                ;20 次,被认为信号稳定转信号
                ;处理，并进行行切换
                com         portb,a
                movam       signal_new
                xor         signal_old,a
                btss        status,z
                clr         key_r               ;信号无效清除 key_r
                ;-------------------------------------------------------------------------------------------------------------
                ;信号检测完成将 PortB 端口恢
                ;复为输出
                movla               b'00000000'
                iodir               portb
                ;-------------------------------------------------------------------------------------------------------------
```

```
;保存当前检测的信号
mov             signal_new,a
movam           signal_old
;-------------------------------------------------------------------------------------------
;连续有效信号次数累计
inc             key_r,m
movla           .10
xor             key_r,a
btss            status,z
lgoto           loop            ;未到次数转程序循环入口
;-------------------------------------------------------------------------------------------
;20 次到为下一行扫描作准备
clr             key_r
inc             row,m           ;row 行寄存器加 1
movla           .4
xor             row,a
btsc            status,z
clr             row             ;最后一行扫完将返回第一行继续
;-------------------------------------------------------------------------------------------
;按键信号处理
movla           b'00000001'
xor             signal_new,a
btsc            status,z
lgoto           check_free
movla           b'00000010'
xor             signal_new,a
btsc            status,z
lgoto           check_free
movla           b'00000100'
xor             signal_new,a
btsc            status,z
lgoto           check_free
movla           b'00001000'
xor             signal_new,a
btsc            status,z
lgoto           check_free
movla           b'00010000'
xor             signal_new,a
btsc            status,z
lgoto           check_free
```

```
                movla       b'00100000'
                xor         signal_new,a
                btsc        status,z
                lgoto       check_free
                movla       b'01000000'
                xor         signal_new,a
                btsc        status,z
                lgoto       check_free
                movla       b'10000000'
                xor         signal_new,a
                btsc        status,z
                lgoto       check_free
                bc          flag_key        ;清按键未松开标志
                lgoto       loop            ;无按键信号转程序循环入口
                ;-------------------------------------------------------------------------------------------
check_free
                ;判断按键是否松开
                btsc        flag_key
                lgoto       loop            ;按键未松开不进行按键信号处理
                ;-------------------------------------------------------------------------------------------
                ;计算键值
                mov         row,a
                lcall       key_table
                movam       key_val
                movla       .0                  ;0 列
                btsc        signal_new,1
                movla       .1                  ;1 列
                btsc        signal_new,2
                movla       .2                  ;2 列
                btsc        signal_new,3
                movla       .3                  ;3 列
                btsc        signal_new,4
                movla       .4                  ;4 列
                btsc        signal_new,5
                movla       .5                  ;5 列
                btsc        signal_new,6
                movla       .6                  ;6 列
                btsc        signal_new,7
                movla       .7                  ;7 列
                add         key_val,m
```

```
;-------------------------------------------------------------------------------------------
;值大于 32,clear PortB
;否则去显示键值
movla           .32
sub             key_val,a
btss            status,c
lgoto           $+3
clr             portb
lgoto           loop
;-------------------------------------------------------------------------------------------
;根据键值点亮相应的 led
;然后转程序循环入口
mov             key_val,a
lcall           led_table
movam           pb_buf
com             pb_buf,a
movam           portb
;-------------------------------------------------------------------------------------------
bs              flag_key            ;置按键未松开标志
lgoto           loop
;-------------------------------------------------------------------------------------------
end
```

- mk7a10p_hw.inc 文档

```
;---------Define special register(Define SFR) ---------------------------------
indf            equ             0x00
tmr0            equ             0x01
pc              equ             0x02
status          equ             0x03
fsr             equ             0x04
porta           equ             0x05            ;porta(0-3)
portb           equ             0x06            ;portb(0-7)
;--------Define [status  Register] special bit---------------------------------
c               equ             0
dc              equ             1
z               equ             2
pd              equ             3
to              equ             4
sa0             equ             5
;-------------------------------------------------------------------------------
```