



● 产品名称

MK7XXXX系列芯片

● 标题

如何使用MK7XXXX系列芯片的WDT唤醒功能

● 简介

当前市场上许多电子产品是用电池来供电的，因此提高电池的使用寿命已经成为所有 IC 设计厂家共同面对的一项课题。对此 MKT 为自己的芯片提供了 SLEEP 模式功能。在 SLEEP 模式中，由于系统时钟停止振荡，所以芯片耗电非常低，芯片在进入了 SLEEP 模式之后，可以通过两种途径唤醒到正常工作模式，这两种途径分别为 IO 唤醒和 WDT 唤醒。下面我们就以 MK7A11P 的 WDT 唤醒试验为例，看看 MKT 的芯片是如何通过 WDT 来唤醒的。

本次试验电路的线路图见图 1，基本功能要求如下：

- 电源上电，LED 全灭
- 第 1 次按下按键时，LED1 闪烁。
- 第 2 次按下按键时，LED2 闪烁。
- 第 3 次按下按键时，LED3 闪烁。
- 第 4 次按下按键时，LED4 闪烁。
- 第 5 次按下按键时，LED 全灭，以后依次循环。

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PB_PUP	UB7	UB6	UB5	UB4	--	UB2	UB1	UB0

表 1 PortB 端口上拉使能寄存器

由于 MK7A11P 芯片的 PortB 端口具有内部上拉功能（上拉使能寄存器见表 1, PB3 除外），所以 PB0 不用外接上拉电阻。启动 WDT 只要设置 CONFIG 寄存器的 WDTE 位（在表 2 中可以找到）即可，设置方法请参考图 2。



Application Note

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CONFIG	ADJ0	RESETE	LV	WDTE	CPT	INRC	FOSC1	FOSC0
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
	--	--	ADJ6	ADJ5	ADJ4	ADJ3	ADJ2	ADJ1

表 2 MK7A11P 的 CONFIG

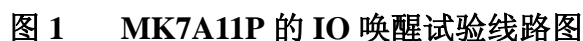
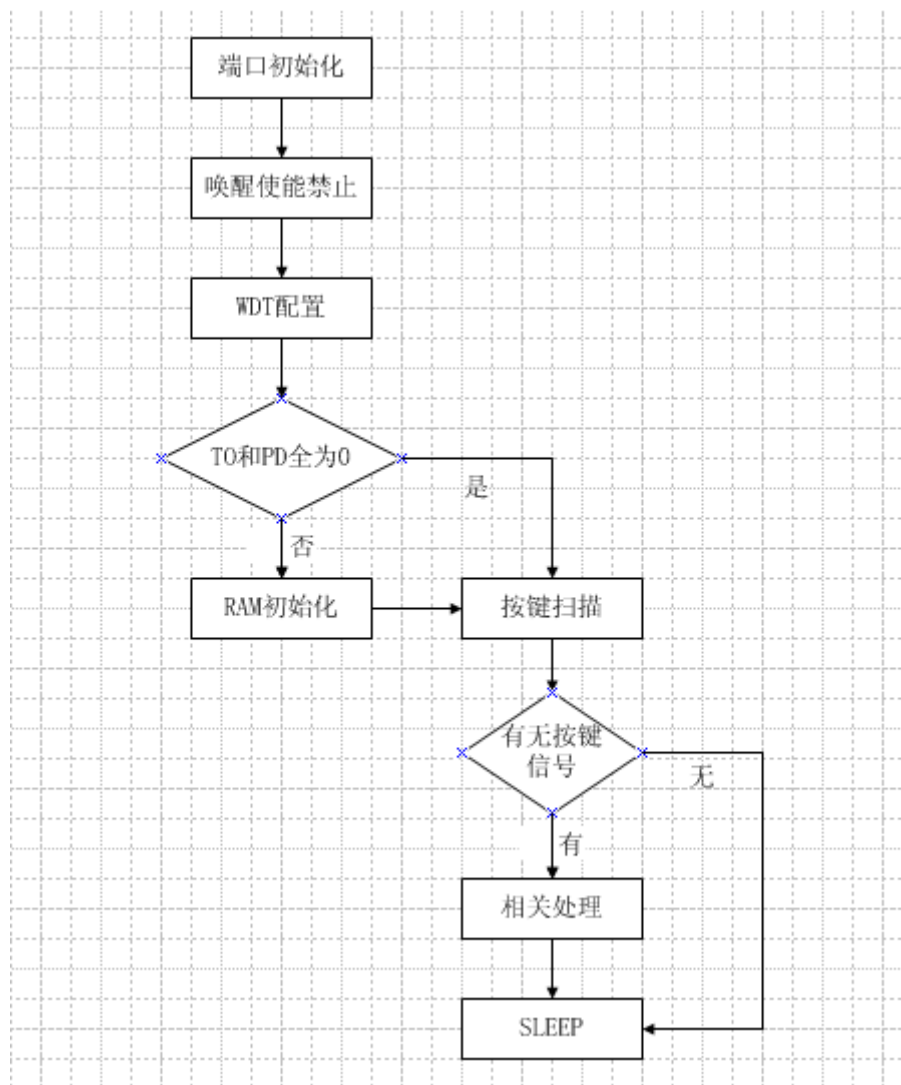


图2 WDT 的设置方法



● 程序流程图



● DEMO 程序

➤ 汇编程序文档

```
;-----  
#include "mk7a11p_hw.inc" ;编译该文档需包含"mk7a11p_hw.inc"文件  
;-----  
;芯片型号 (mk7a11p)  
;-----  
;配置寄存器设置说明 (CONFIG)  
;1-----FOSC=RC ;LS,NS,HS,RC  
;2-----INRC=ON ;ON,OFF
```



```
;3-----CPT=OFF                ;ON,OFF
;4-----WDTE=Enable             ;Enable,Disable
;5-----LV=Low Vol Reset ON     ;Low Vol Reset ON,Low Vol Reset OFF
;6-----RESET=...input...       ;...input...,...reset...
;-----
signal_new      equ      0x20      ;存储当前的按键信号
signal_old      equ      0x21      ;存储上次的按键信号
key_r           equ      0x22      ;检测信号的次数
flag            equ      0x23      ;标志
dis_r           equ      0x24      ;按键次数
;-----
#define          flag_key         flag,0      ;按键未松开标志
#define          flag_t           flag,1      ;重新计时标志
;-----
                org      0x3ff      ;mk7a11p 的复位向量地址定义
                lgoto     main      ;跳转到主程序入口
;-----
                org      0x000
led_table
                add       pc,m
                retla     b'00000000'
                retla     b'00000001'
                retla     b'00000010'
                retla     b'00000100'
                retla     b'00001000'
;-----
                org      0x100      ;主程序入口地址定义
main
                ;PortA 端口
                movla     b'11110000'
                movam      pa_pdm
                iodir      porta
;-----
                ;PortB 端口
                movla     b'00000001'
                movam      pb_pup      ;PB0 上拉使能
                iodir      portb
                clr        pb_pod
                clr        pb_pdm
;-----
```



```
;PortB 端口 8 个 IO 唤醒禁止
movla      b'00000000'
movam      wake_up
;-----
;配置 WDT,预分频比为 1:1
movla      b'00001000'
select
;-----
btsc       status,to
lgoto      ram_init      ;非 WDT 从 LEEP 模式中复位转 RAM 初始化
btsc       status,pd
lgoto      ram_init      ;非 WDT 从 LEEP 模式中复位转 RAM 初始化
lgoto      display       ;WDT 从 LEEP 模式中复位不进行 RAM 初始化
;-----

ram_init

movla      0x20
movam      fsr           ;将 0x20 送入 fsr 寄存器
;-----

clear_ram

;利用 indf 和 fsr 来进行间接寻址
;对 0x20-0x2f 的 RAM 进行 clear
clr        indf
mov        fsr,a
andla      b'00111111'   ;将无关的数据清除
xorla      0x2f
btsc       status,z
lgoto      $+3
inc        fsr,m
lgoto      clear_ram
clr        fsr           ;使用 fsr 时要注意 bank 的归位
;-----

clr        porta
clr        portb
;-----

display

;根据按键的次数
;查表 LED 该显示的数据
;并送显示
mov        dis_r,a
lcall      led_table
```



```
movam    porta

;-----
scan_key                                ;按键扫描处

;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
andla    b'00000001'
movam    signal_new
xor      signal_old,a
btss     status,z
clr      key_r                        ;信号无效清除 key_r

;-----
;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    .3
xor      key_r,a
btss     status,z
lgoto    into_sleep                  ;未到次数转 SLEEP 入口

;-----
;按键信号判断
movla    0x01
xor      signal_new,a
btsc     status,z
lgoto    check_key                  ;判断按键是否处理过

;-----
;无按键信号出现
bc       flag_key                  ;清按键处理过标志
lgoto    into_sleep                  ;无按键信号出现转 SLEEP 入口

;-----
check_key                                ;判断按键是否处理过
btsc     flag_key
lgoto    into_sleep                  ;按键处理过不再进处理
```



```
;-----  
;按键信号处理  
bs      flag_key      ;置按键处理过标志  
;-----  
;按键次数累计  
inc     dis_r,m  
movla   .5  
xor     dis_r,a  
btsc    status,z  
clr     dis_r      ;按键次数等于 4 归零  
;-----  
into_sleep  
        sleep  
        nop  
        nop  
        nop  
;-----  
        end
```

➤ mk7a11p_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)  
;-----  
irqm      equ      0x09  
irqf      equ      0x0a  
;-----  
pa_pdm    equ      0x0b  
pb_pup    equ      0x0c  
pb_pdm    equ      0x0d  
pb_pod    equ      0x0e  
wake_up   equ      0x0f  
;-----Define [status Register] special bit-----  
c          equ      0  
dc         equ      1
```



z	equ	2
pd	equ	3
to	equ	4
;-----Define [irqm Register] special bit-----		
tm0m	equ	0
extm	equ	1
intm	equ	7
;-----Define [irqf Register] special bit-----		
tm0f	equ	0
extf	equ	1
;-----Define [pa_pdm Register] special bit-----		
da0	equ	0
da1	equ	1
da2	equ	2
da3	equ	3
;-----Define [pb_pup Register] special bit-----		
ub0	equ	0
ub1	equ	1
ub2	equ	2
ub4	equ	4
ub5	equ	5
ub6	equ	6
ub7	equ	7
;-----Define [pb_pdm Register] special bit-----		
db0	equ	0
db1	equ	1
db2	equ	2
inte	equ	6
rtce	equ	7
;-----Define [pb_pod Register] special bit-----		
ob0	equ	0
ob1	equ	1
ob2	equ	2
ob4	equ	4
ob5	equ	5
ob6	equ	6
ob7	equ	7



```
;-----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  
;-----
```
