



**PRODUCT NAME**  
TM58XX

**TITLE**  
How to use SLEEP function in the TM58xx series

## APPLICATION NOTE

1. Introduction
2. DEMO Code
3. Diagrams of the applicable circuits

### 1. Introduction

Many of the current products require IC with low power consumption, in order to extend the lifespan of the batteries of the products. The TM58 series IC provides the SLEEP function in response to this demand. The usage of the SLEEP function in the TM58 series MCU (using model TM58P10 as an example) will be described below.

The SFR (Special Function Register) that is related to the SLEEP function is WAKE\_UP (its address is 20H), and is defined as follows:

#### WAKE\_UP:

Bit	Symbol	Description
7	WDTS	Watch Dog timer register software control bit: TM58P10/20 has 2 WDT control bits (WDTE and WDTS), WDTE is set in configuration word by hardware and WDT is set in control register by software. If WDTS is valid only if WDTE has been set, i.e. WDTE has higher priority than WDTS. 1: Enable 0: Disable
6	WUE	Wake_Up enable bit: 1: Enable external wakeup 0: Disable external wakeup
5	EIS	External interrupt select: 1: Set PA0 as an external IRQ pin 0: Set PA0 as a bi-directional I/O pin
4	----	
3~1	PUH <sub>3</sub> ~PUH <sub>1</sub>	Pull High Port A bit3~1: 1: Enable pull-high If (WUE) & (PUH <sub>N</sub> ) & (input a falling edge signal at PA <sub>N</sub> ) than wake_up chip form sleep. N can be 3、2 or 1, but it must keep consistent.

		0: Disable pull-high and external wake_up
0	Puh0 PUH0	<p>Pull High Port A bit0:                      1: Enable pull-high                      If (WUE) &amp; (PUH0) &amp; (input a falling an edge signal at PA0) than Wake_Up chip form sleep. Or if (EIS) &amp; (PUH0) &amp; (input a falling edge signal at PA0) then generate an IRQ.                      0: Disable pull-high, external wakeup, and external interrupt.  <b>Note:</b> if PUH0, WUE, or EIS set as '1', then PA0 is defined as IRQ input pin</p>

**NOTE-1:** if PUH0, WUE, or EIS is set as '1', then PA0 is defined as the interrupt request pin, but the SLEEP mode cannot be awakened under this condition. Only when "EIS"=0 & "WUE"=1 can PA0 be defined as the wake-up input port, and only then can the SLEEP mode be awakened.

**NOTE-2:** The PORTA pull-high function is only effective under the condition of WUE=1(enable external wakeup).

**NOTE-3:** Select register:

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		SUR0	EDGE3	PSA	PS2	PS1	PS0

If the "PSA" of the select register is set to "0" before entering the SLEEP mode, the prescaler will be assigned to TMTR0, and then the execution of the WAKE\_UP function will lead to the resetting of the chip after it is awakened. To avoid the problem, the following method is used: the "PSA" is set to "1" before entering the SLEEP mode, assign the prescaler to WDT and disable the WDT function; consequently, the IC will not be rest after being awakened.

**2. DEMO Code**

The following DEMO code and the diagrams of circuits indicate the process in which TM58P10 makes use of the wakeup function and interruption of timing by using TIMER: (For the functions of the programming, please read the functional description in the programming lines).

```

;=====
;
;Mode: Advanced
;Crystal vibration: 4MHZ
;WATCHDOG: DISABLE
;Function: Programming for the pull-high wakeup function at PORTA port
;When a button is pressed, the corresponding LED light alights (the correspondence
between the buttons and the lights is: SW1 corresponds to LED1, SW2 corresponds to
LED2, SW3 corresponds to LED3, SW4 corresponds to LED4); if no button is pressed, the
system re-enters the SLEEP mode after 5S, all of the LED lights go off before entering
SLEEP mode.
;
;
;=====
;Definitions of the special function register
;=====

```

```
indf      equ    00h
tmr0      equ    01h
pc        equ    02h
status    equ    03h
fsr       equ    04h
porta     equ    05h
portb     equ    06h
wakeup    equ    20h
irqm      equ    21h
irqf      equ    22h
```

```
;
```

```
=====
```

```
;Bit definitions of the destination register
```

```
=====
```

```
w         equ    00h
f         equ    01h
```

```
;
```

```
=====
```

```
;Bit definitions of the status register
```

```
=====
```

```
c         equ    00h
dc        equ    01h
z         equ    02h
pd        equ    03h
to        equ    04h
pa0       equ    05h
pa1       equ    06h
pa2       equ    07h
```

```
;
```

```
=====
```

```
;Bit definitions of the interrupt mask register (irqm)
```

```
=====
```

```
intm      equ    07h
exintm    equ    02h
tmr0m     equ    00h
```

```
;
```

```
=====
```

```
;Bit definitions of the interrupt flag register (irqf)
```

```
=====
```

```
tmr0f     equ    00h
exintf    equ    02h
```

```
;
```

```
=====
```

```
;Bit definitions of the select register
```

```
=====
```

```

ps0      equ    00h
ps1      equ    01h
ps2      equ    02h
psa      equ    03h
rte      equ    04h
rts      equ    05h
;
;
;
=====

```

```

;Bit definitions of the WAKEUP register
;
=====

```

```

wdts     equ    07h
wue      equ    06h
eis      equ    05h
puh3     equ    03h
puh2     equ    02h
puh1     equ    01h
puh0     equ    00h
;
;
;
=====

```

```

;Bit definitions
;
=====

```

```

count1   equ    08h
count2   equ    09h
timer1   equ    0ah
timer2   equ    0bh
flag     equ    0ch
buffer   equ    0dh
s_buffer equ    0eh
;
;
;
=====

```

```

;prog. start here
;
=====

```

```

    org    3ffh      ; Resets vector address
    goto   main
    org    3feh      ; Interrupts entrance address of the subprogram
    goto   timer
    org    000h      ; The initialization address of programs in ROM
    nop
;
;
;
;
=====

```

```

;MAIN PROGRAM
;
=====

```

```

main:                ; initialization
    clrm    portb    ; initial value of portb is 00h
    clrm    porta    ; initial value of porta is 00h
    movla   0fh      ; Sets porta port as the import port
    iodir   porta    ;
    movla   00h      ; Sets portb port as the export port
    iodir   portb    ;
    movla   0ah      ; Prescaling ratio of prescaler is set at 1:8
    select  ; Sets PSA=1 before entering SLEEP mode, assigns prescaler
              to WATCHDOG
    movla   06h      ; Sets the initial value of TMR0 as 06h
    movam   tmr0     ;
    movla   4fh      ; Sets the initial value of wakeup register as 4fh, sets WUE=1 to
              ; enable external wakeup function, sets the four I/O of porta
              port at pull-high to give it the external wakeup
    movam   wakeup
    movla   81h      ; Sets the initial value of irqm register as 81h, sets
              ; INTM=1&TMR0M=1, enables TMR0 interrupt
    movam   irqm    ;
    clrm    irqf     ; Clears the irqf
    clrm    count1   ; Clears count1
    clrm    count2   ; Clears count2
    clrm    timer1   ; Clears timer1
    clrm    timer2   ; Clears timer2
    clrm    flag     ; Clears flag
;
;
;
s_loop:              ; sleep program sections
    nop
    nop
    sleep          ; Enters SLEEP mode
    nop
    nop
    movla   02h      ; After awaking from SLEEP mode, sets PSA=0, assigns
              prescaler to tmr0
    select
    goto    start    ; Program turns to start for execution
;
;
;
=====
;Delay 10ms subprogram
=====
;time setting : 2+1+1+10*(1+1+250*(1+1+2)+1+2)+2=10056uS=10.056mS
delay_10:
    movla   0ah
    movam   count2
del1_1:
    movla   0fah
    movam   count1

```

```

del1_2:
    nop
    decmsz count1
    goto del1_2
    decmsz count2
    goto del1_1
    ret
;
;
;
;=====
start:                                ; start program sections
    btmsc porta,0                      ; Determines whether button sw1 is pressed or not
    goto porta_1                      ; Button sw1 is not pressed, turns to porta_1 for
                                        execution
    call delay_10                     ; Button sw1 is pressed, adjusts the delay 10mS
                                        subprogram,
                                        ; delays 10mS to remove joggles
    btmsc porta,0                      ; Re-determines whether button sw1 is pressed or not
    goto porta_1                      ; Button sw1 is not pressed due to joggles, turns to
                                        porta_1 for execution
    movla 01h                          ; Sets the value "01h" to flag
    movam flag                          ;
    clrm timer1                        ; Clears timer1
    clrm timer2                        ; Clears timer2
    goto out_pb                         ; for execution
porta_1:
    btmsc porta,1                      ; Determines whether button sw2 is pressed or not
    goto porta_2                      ; Button sw2 is not pressed, turns to porta_2 for
                                        execution
    call delay_10                     ; Button sw2 is pressed, adjusts the delay 10mS
                                        subprogram,
                                        ; delays 10mS to remove joggles
    btmsc porta,1                      ; Re-determines whether button sw2 is pressed or not
    goto porta_2                      ; Button sw2 is not pressed due to joggles, turns to
                                        porta_2 for execution
    movla 02h                          ; Sets the value "02h" to flag
    movam flag                          ;
    clrm timer1                        ; Clears timer1
    clrm timer2                        ; Clears timer2
    goto out_pb                         ; Turns to out_pb for execution
porta_2:
    btmsc porta,2                      ; Determines whether button sw3 is pressed or not
    goto porta_3                      ; Button sw3 is not pressed, turns to porta_3 for
                                        execution
    call delay_10                     ; Button sw3 is pressed, adjusts the delay 10mS
                                        subprogram,
                                        ; delays 10mS to remove joggles
    btmsc porta,2                      ; Re-determines whether button sw3 is pressed or not
    goto porta_3                      ; Button sw3 is not pressed due to joggles, turns to

```

```

                                porta_3 for execution
movla 04h                        ; Sets the value "04h" to flag
movam flag                       ;
clrm timer1                      ; Clears timer1
clrm timer2                      ; Clears timer2
goto out_pb                      ; Turns to out_pb for execution
porta_3:
btmsc porta,3                   ; Determines whether button sw4 is pressed or not
goto out_pb                     ; Button sw4 is not pressed, turns to out_pb for
                                execution
call delay_10                   ; Button sw4 is pressed, adjusts the delay 10mS
                                subprogram, delays 10mS to remove joggles
btmsc porta,3                   ; Re-determines whether button sw4 is pressed or not
goto out_pb                     ; Button sw4 is not pressed due to joggles, turns to
                                out_pb for execution
movla 08h                        ; Sets the value "08h" to flag
movam flag                       ;
clrm timer1                      ; Clears timer1
clrm timer2                      ; Clears timer2
out_pb:
btmsc flag,0                    ; Determines whether the control flag of led1 is 1 or not
goto turn_on_led1              ; Control flag of led1 is 1, turns to turn on led1
btmsc flag,1                    ; Control flag of led1 is not 1, determines whether the
                                control flag of led2 is 1 or not
goto turn_on_led2              ; Control flag of led2 is 1, turns to turn on led2
btmsc flag,2                    ; Control flag of led2 is not 1, determines whether the
                                control flag of led3 is 1 or not
goto turn_on_led3              ; Control flag of led3 is 1, turns to turn on led3
btmsc flag,3                    ; Control flag of led3 is not 1, determines whether the
                                control flag of led4 is 1 or not
goto turn_on_led4              ; Control flag of led4 is 1, turns to turn on led4
goto main                      ; Returns to the main section at the beginning of the
                                main program for execution
turn_on_led1:
movla 01h                        ; Turns on led1
movam portb                     ;
goto start                      ; Returns to start for execution
turn_on_led2:
movla 02h                        ; Turns on led2
movam portb                     ;
goto start                      ; Returns to start for execution
turn_on_led3:
movla 04h                        ; Turns on led3
movam portb                     ;
goto start                      ; Returns to start for execution
turn_on_led4:
movla 08h                        ; Turns on led4
movam portb                     ;

```