PRODUCT NAME

TM58XX

TITLE

如何使用 TM58 系列的外部中斷

APPLICATION NOTE

1.簡介
2.DEMO 程式
3.應用線路圖

簡介

在當前的許多應用程式中，對於外部設備的請求處理使用外部中斷功能可以簡化程式的複雜性，提高 MCU 的利用率。針對此目的，我們介紹以下 TM58 系列 MCU（以 TM58P10 爲例）如何來使用外部中斷。

在使用外部中斷時，首先要用到 TM58 系列 IC 的 ADVANCE（高級）模式。因爲只有在高級模式才提供中斷功能。在該模式下，TM58P10 的位址 3FEH 爲中斷向量地址，TM58P20 的地址 7FEH 爲中斷向量地址。與外部中斷有關的特殊功能暫存器有三個：中斷遮罩暫存器 IRQM(地址爲 21H)，喚醒功能控制暫存器 WAKE-UP(位址爲 20H)和中斷旗標暫存器 IRQF（位址爲 22H），其具體定義如下：

IRQM：

位	符號	描述
7	INTM	整體中斷致能位元： 該位元較其他中斷致能信號具有更高的優先順序 1：允許 0：禁止
6~3	----	
2	EXINTM	外部中斷致能位元： 1：允許中斷 0：禁止中斷
1	----	
0	TMR0M	TMR0 中斷致能位元： 1：允許中斷 0：禁止中斷

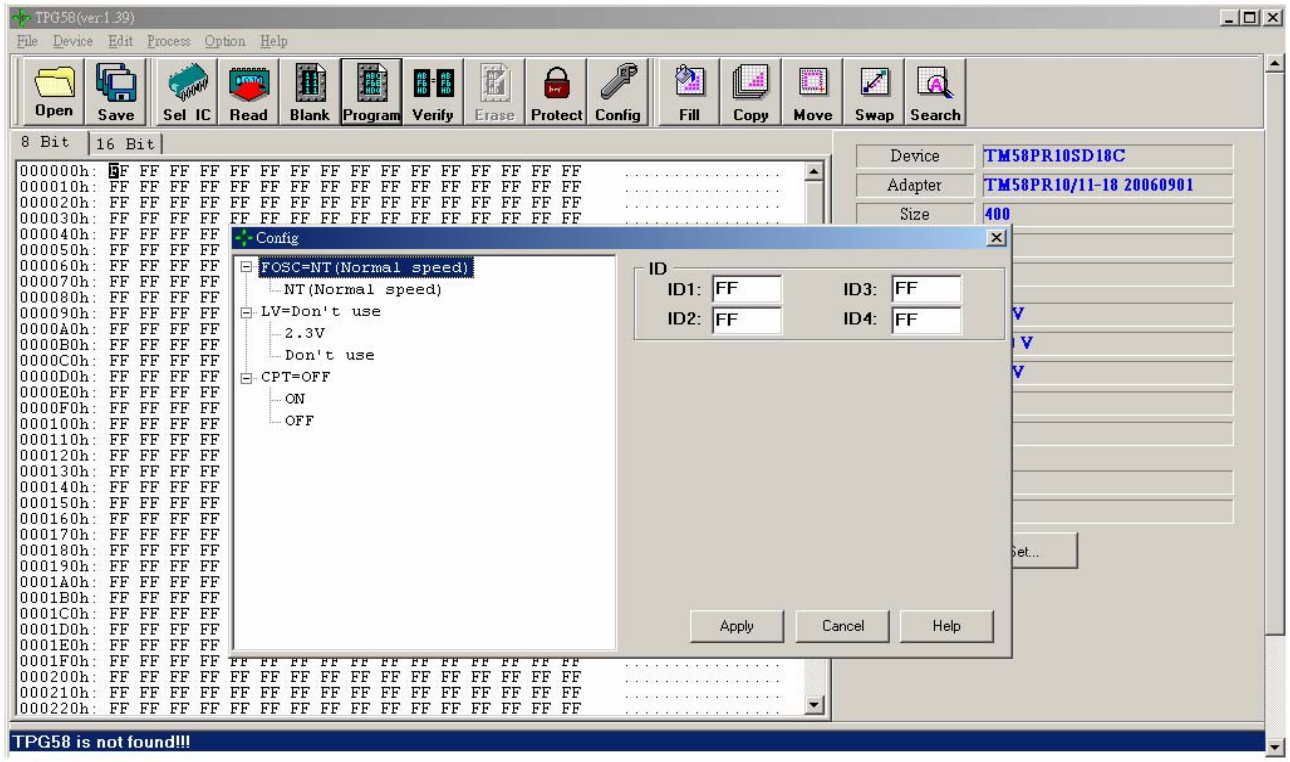
IRQF：

位	符號	描述
7~3	----	
2	EXINTF	外部中斷旗標： 1：外部中斷產生的外部中斷請求
1	----	
0	TMR0F	TMR0 中斷旗標： 1：TMR0 計數器溢出產生的中斷請求

WAKE_UP:

位	符號	描述
7	WDTS	Watch Dog 計時器控制位元： TM58P10/20 有 2 個 WDT 控制位元（WDTE，WDTS），WDTE 是在 configuration word 中由硬體設置的，而 WDTS 是在控制暫存器中由軟體設置的。只有當 WDTE 被設置後，WDTS 才是有效的，也就是說，WDTE 比 WDTS 有更高的優先順序。 1：允許 0：禁止
6	WUE	喚醒致能位元： 1：允許外部喚醒功能 0：不允許外部喚醒
5	EIS	外部中斷選擇位元： 1：PA0 當做外部中斷腳使用 0：PA0 當做普通 I/O 使用
4	----	
3~1	Puh3~Puh1	PA 埠 bit3~1 上拉設定位元： 1：允許上拉 (當(WUE=1)和(PUHn=1)時，若在 PUHn 輸入一個下降沿信號,將從睡眠狀態喚醒晶片。n 可以是 3、2、1 。 0：禁止外部喚醒
0	Puh0	PA 埠 bit0 上拉設定位元： 1：允許上拉 當(WUE=1)和(PUH0=1)時，若在 PUH0 輸入一個下降沿信號,將從睡眠狀態喚醒晶片； 當(EIS=1)和(PUH0=1)時，若在 PUH0 輸入一個下降沿信號，將引起外部中斷。 0：禁止上拉，外部喚醒和外部中斷 注：如果 PUH0，WUE，EIS 被置成 ‘1’ ，那麼 PA0 被定義成中斷請求輸入腳。

工作模式的選擇及看門狗 WDT 的設定需在 IC 燒錄時在 CONFIG WORD 中設定，CONFIG WORD 如下圖所示：



NOTE1：

在程式中應注意進中斷時變數的保存及完成中斷時變數的恢復。

NOTE2：

當有外部中斷時，將由硬體置位元中斷標誌暫存器的 EXINTF 位元（EXINTF 位元由硬體設定位元，由軟體清零），爲避免進入中斷閉環，必須在退出中斷副程式時用軟體清除該標誌位元。系統在進入中斷時會自動關中斷，在出中斷時由 RETI 指令自動開中斷。

下面是 TM58P10 使用外中斷時的 DEMO 程式和線路圖：
程式的功能請見程式中的功能描述。

```
;=========================================
;模式（mode):高級（advanced)
;晶振：4MHZ
;看門狗（WATCHDOG）：禁止（DISABLE）
;功能:使用 PORTA,0 當作外部中斷的輸入腳，讓
;燈 led1,led2,led3,led4,led5,led6,led7 以跑馬燈的方式作閃滅的動作。
;( 時間間隔爲:0.5S)。燈 led8 的亮滅由按鍵控制,當按下一次按鍵
```

```
;時燈 led8 點亮,再一次按下按鍵時則熄滅。按鍵的回應過程由外部中
;斷功能實現。
;
;
;==========================================
;特殊功能暫存器定義
;==========================================
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
;
;
;==========================================
;目的暫存器控制位元定義
;==========================================
w               equ     00h
f               equ     01h
;
;
;==========================================
;狀態暫存器(status)位元定義
;==========================================
c               equ     00h
dc              equ     01h
z               equ     02h
pd              equ     03h
to              equ     04h
pa0             equ     05h
pa1             equ     06h
pa2             equ     07h
;
;
;==========================================
；中斷遮罩暫存器(irqm)位定義
;==========================================
intm            equ     07h
exintm          equ     02h
tmr0m           equ     00h
;
;
;==========================================
;中斷請求標誌暫存器（irqf)位元定義
;==========================================
```

```
tmr0f           equ     00h
exintf          equ     02h
;
;
;=========================================
;select 暫存器位定義
;=========================================
ps0             equ 00h
ps1             equ 01h
ps2             equ 02h
psa             equ 03h
rte             equ 04h
rts             equ 05h
;
;
;=========================================
;WAKE_UP 暫存器位定義
;=========================================
wdts            equ 07h
wue             equ 06h
eis             equ 05h
puh3            equ 03h
puh2            equ 02h
puh1            equ 01h
puh0            equ 00h
;
;
;=========================================
;位置定義
;=========================================
led_flag        equ 08h
count1          equ 09h
del_cnt         equ 0ah
a_temp          equ 0bh
status_temp     equ 0ch
time_temp       equ 0dh
time_buf        equ 0eh
del_flag        equ 0fh
;
;
;=========================================
;portb □        I/O 定義
;=========================================
led1            equ 00h
led2            equ 01h
led3            equ 02h
led4            equ 03h
led5            equ 04h
led6            equ 05h
led7            equ 06h
```

```
led8            equ 07h
;
;
;=========================================
;=========================================
;prog. start here
;=========================================
;=========================================
        org     3ffh            ;重定向量地址
        goto    main
        org     3feh            ;中斷副程式入口位址
        goto    extint
        org     000h            ;程式在 ROM 中的起始地址
;
;
;=========================================
;tmr0 時間處理副程式
;=========================================
readtime:
        movm    tmr0,w          ;讀取 tmr0 的值,結果放入累加器中
        xoram   time_temp,w     ;tmr0 的值與 time_temp 值相異或,結果放入累加器中
        movam   time_buf        ;將 tmr0 與 time_temp XOR 的結果送入 time_buf 中
        xoram   time_temp,f     ;tmr0 的值送入 time_temp 中
        btmss   time_buf,7      ;判斷 128*64us/1000=8.192ms 是否已到
        ret                     ;8.192ms 未到，則退出副程式
        btmsc   del_flag,0      ;8.192ms 已到,判斷是否處於遮罩外中斷狀態
        incm    del_cnt         ;處於遮罩外中斷狀態,則 del_cnt 值加 1
        incm    count1          ;count1 值加 1

        movla   3dh             ;判斷 61*8.192ms=499.712ms 約爲 0.5S 是否已到
        xoram   count1,w
        btmss   status,z
        ret                     ;499.712ms 未到，則退出副程式
        clrm    count1          ;499.712ms 已到，清 count1
        btmsc   led_flag,led1   ;判斷 led1 控制標誌位元是否爲 1？
        goto    turn_on_led1    ;led1 控制標誌位元爲 1,轉去開 led1
        btmsc   led_flag,led2   ;led1 控制標誌位元不爲 1,判斷 led2 控制旗標位元是否爲 1？
        goto    turn_on_led2    ;led2 控制標誌位元爲 1,轉去開 led2
        btmsc   led_flag,led3   ;led2 控制標誌位元不爲 1,判斷 led3 控制旗標位元是否爲 1？
        goto    turn_on_led3    ;led3 控制標誌位元爲 1,轉去開 led3
        btmsc   led_flag,led4   ;led3 控制標誌位元不爲 1,判斷 led4 控制旗標位元是否爲 1？
        Goto    turn_on_led4    ;led4 控制標誌位元爲 1,轉去開 led4
        Btmsc   led_flag,led5   ;led4 控制標誌位元不爲 1,判斷 led5 控制旗標位元是否爲 1？
        Goto    turn_on_led5    ;led5 控制標誌位元爲 1,轉去開 led5
        btmsc   led_flag,led6   ;led5 控制標誌位元不爲 1,判斷 led6 控制旗標位元是否爲 1？
        goto    turn_on_led6    ;led6 控制旗標位元爲 1,轉去開 led7
```

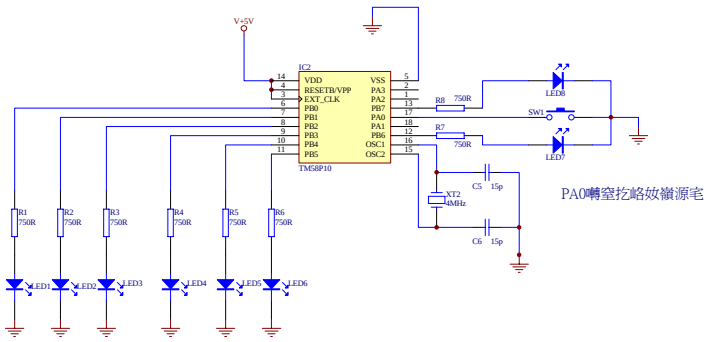
```
        goto    turn_on_led7   ;led7 控制旗標位元爲 1,轉去開 led7
turn_on_led1:
        bcm     portb,led7     ;關閉燈 led7
        nop
        nop
        nop
        bsm     portb,led1     ;打開燈 led1
        movla   02h
        movam   led_flag       ;給 led_flag 值 02h,使其指向燈 led2
        ret                    ;退出副程式
turn_on_led2:
        bcm     portb,led1     ;關閉燈 led1
        nop
        nop
        nop
        bsm     portb,led2     ;打開燈 led2
        movla   04h
        movam   led_flag       ;給 led_flag 值 04h,使其指向燈 led3
        ret                    ;退出副程式
turn_on_led3:
        bcm     portb,led2     ;關閉燈 led2
        nop
        nop
        nop
        bsm     portb,led3     ;打開燈 led3
        movla   08h
        movam   led_flag       ;給 led_flag 值 08h,使其指向燈 led4
        ret                    ;退出副程式
turn_on_led4:
        bcm     portb,led3     ;關閉燈 led3
        nop
        nop
        nop
        bsm     portb,led4     ;打開燈 led4
        movla   10h
        movam   led_flag       ;給 led_flag 值 10h,使其指向燈 led5
        ret                    ;退出副程式
turn_on_led5:
        bcm     portb,led4     ;關閉燈 led4
        nop
        nop
        nop
        bsm     portb,led5     ;打開燈 led5
        movla   20h
        movam   led_flag       ;給 led_flag 值 20h,使其指向燈 led6
        ret                    ;退出副程式
```

```
turn_on_led6:
        bcm     portb,led5      ;關閉燈 led5
        nop
        nop
        nop
        bsm     portb,led6      ;打開燈 led6
        movla   40h
        movam   led_flag        ;給 led_flag 值 40h,使其指向燈 led7
        ret                     ;退出副程式
turn_on_led7:
        bcm     portb,led6      ;關閉燈 led6
        nop
        nop
        nop
        bsm     portb,led7      ;打開燈 led7
        movla   01h
        movam   led_flag        ;給 led_flag 值 01h,使其指向燈 led1
        ret                     ;退出副程式
;
;
;=========================================
;MAIN PROGRAM
;=========================================
main:;初始化
        clrm    count1          ;清 count1
        clrm    del_cnt         ;清 del_cnt
        clrm    del_flag        ;清 del_flag
        movla   02h             ;
        movam   led_flag        ;給 led_flag 值 02h,使其指向燈 led2
        movla   01h             ;
        movam   portb           ;打開燈 led1
        clra                    ;
        iodir   portb           ;將 portb 埠設爲輸出態
        movam   porta           ;porta 埠輸出爲 0
        movla   0fh             ;
        iodir   porta           ;將 porta 埠設爲輸入態
        movla   05h             ;
        select                  ;tmr0 採用內部時鐘計數,預除頻器分配給 tmr0,
                                ;除頻比爲 1:64
        movla   b'00100001'     ;
        movam   wakeup          ;設定 porta,0 口爲外部中斷輸入
        movla   b'10000100'     ;
```

```
        movam irqm              ;開放外中斷功能
        movm   tmr0,w       ;
        movam time_temp         ;讀 tmr0 的值結果送入 time_temp 中
;
;
start:;主迴圈
        call      readtime      ;調 tmr0 時間處理副程式
        btmsc     irqm,exintm   ;判斷外部中斷功能是否關閉
        goto      start         ;未關閉外部中斷功能,返回 start 處執行
        btmss     porta,0       ;外部中斷功能關閉,判斷按建是否彈起。
        goto      start         ;按鍵未彈起,返回 start 處執行
        bsm       del_flag,0    ;按鍵彈起,置控制延時標誌 del_flag
        movla     05h
        xoram     del_cnt,w
        btmss     status,z      ;判斷延時消抖時間 5*8.192ms=40.96ms 到否
        goto      start         ;延時消抖時間未到,返回 start 處執行
        clrm      del_cnt       ;延時消抖時間到,清 del_cnt
        btmss     porta,0       ;再次確認按鍵是否真的彈起
        goto      start         ;按鍵未彈起,返回 start 處執行
        bcm       del_flag,0    ;按鍵確實擡彈起,清控制延時旗標 del_flag
        bcm       irqf,exintf   ;清外部中斷請求旗標位元
        bsm       irqm,exintm   ;開放外部中斷功能
        goto      start         ;返回 start 處執行
;
;
;=========================================
;外部中斷回應副程式
;=========================================
extint:
        movam  a_temp           ;將累加器內容送入 a_temp 中進行保護
        swapm  status,w         ;將狀態暫存器 status 高低半位元組內容
                                ;相互交換結果送累加器中
        movam status_temp       ;最後將結果送入 status_temp 中進行保護
        btmss    portb,led8     ;判斷燈 led8 的開/關狀態
        goto     turn_on_led8   ;燈 led8 處於關閉狀態,則轉去開燈
turn_off_led8:
        bcm      portb,led8     ;開燈 led8
        goto     end_int        ;離開中斷處
turn_on_led8:
        bsm      portb,led8     ;關燈 led8
end_int:
        bcm      irqm,exintm    ;關閉外中斷功能
```

```
        bcm     irqf,exintf     ;清外中斷請求旗標
        swapm   status_temp,w;將 status_temp 暫存器高低半位元組內容相互交換
                                ;結果送累加器中
        movam   status          ;累加器內容送狀態暫存器 status,以恢復進中
                                ;斷前的 status 狀態
        swapm   a_temp,f        ;將 a_temp 的高低半位元組內容相互交換,
                                ;結果送入 a_temp 中
        swapm   a_temp,w        ;將 a_temp 的高低半位元組內容相互交換,
                                ;結果送入累加器中,以恢復進中斷前的累加器內容
        reti                    ;中斷返回，執行該指令後，程式將自動開中斷
;
;
;=========================================
        end                     ;程式結束
;
;
;=========================================
;範例
;=========================================
;系統開電後燈 led1,led2,led3,led4,led5,led6,led7 即開始以 0.5S 時間爲間隔
;做跑馬燈。燈 led8 處於關閉狀態。此時按下按鍵,燈 led8 點亮。再按一次
;則燈 led8 熄滅。按鍵的回應過程由外中斷功能實現。按鍵在彈起的過程中
;由於抖動原因,會引起多次中斷,導致燈 led8 誤動作。所以程式中對按鍵作了除彈跳的處
理。
;處理方式如下：在退出中斷前先將外中斷功能關閉,然後在中斷程式外檢測按鍵有無彈
;起,直到檢測到按鍵彈起後才開放外中斷功能。由此,可有效避免一次按鍵多次中斷的
;現象。爲避免中斷閉環,在出中斷時必須清除相對應的中斷請求旗標位元。
```

應用線路圖



	Title: TM58烚蹈偲窒筷剿APN盅繚芣		
tenx	Size: B	Number:	Revision:
	Date:		Sheet of
	File:		Drawn by: