



PRODUCT NAME

TM58XX

TITLE

TM58XX 如何掃描 4*8 按鍵

APPLICATION NOTE

- 1.簡介
- 2.DEMO 程式
- 3.應用線路圖

簡介

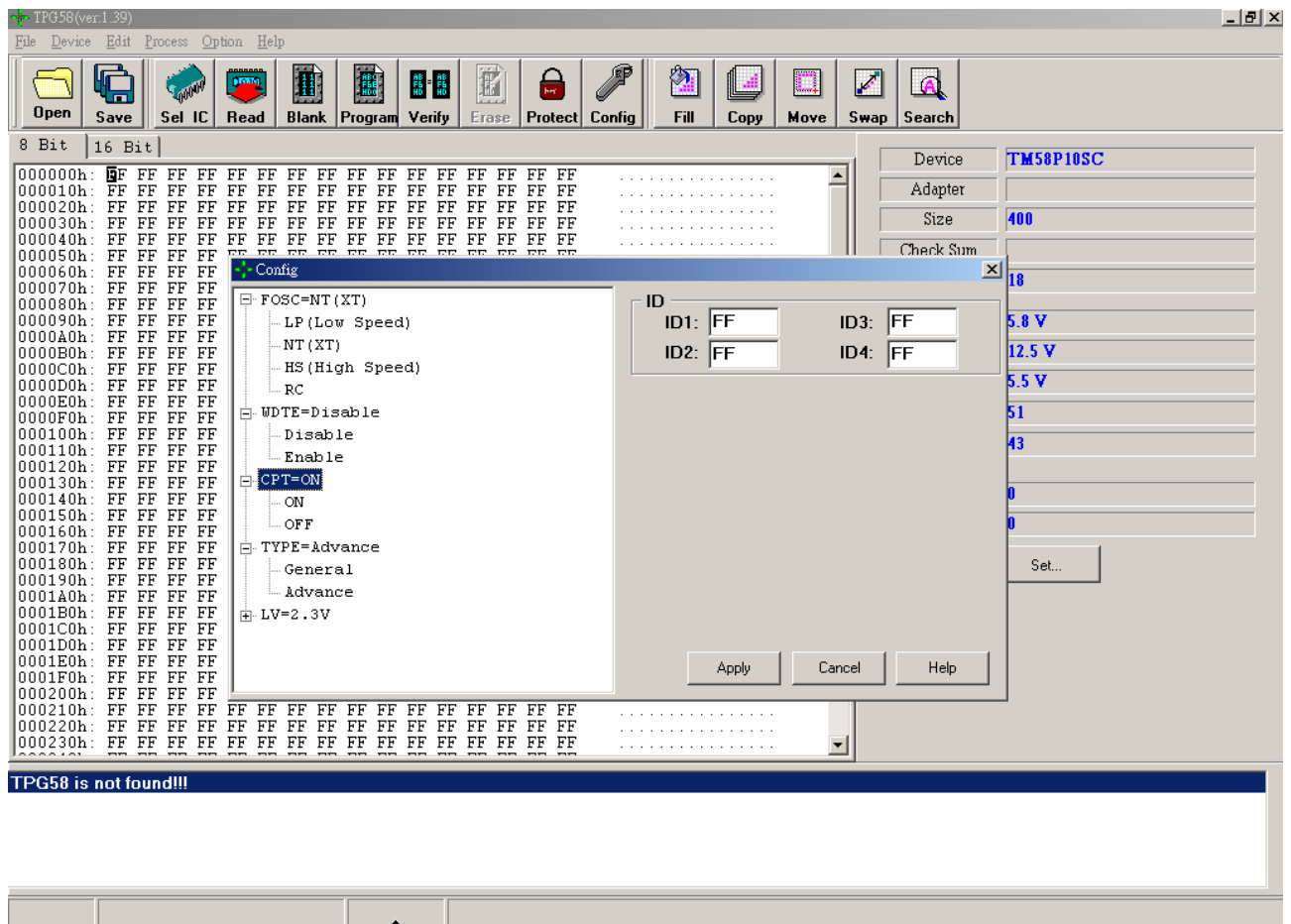
對於許多產品來說，按鍵輸入作為最基本的輸入是不可避免的。因此，如何有效的掃描按鍵，就顯得尤為重要。針對此目的，我們介紹以下 TM58 系列 MCU（以 TM58P10 為例）如何來掃描 4*8 按鍵。

在該常式中因 PORTA 口使用了上拉功能，所以用到了 TM58 系列 IC 的 ADVANCE（高級）模式。因為只有在高級模式下才提供 PORTA 口上拉功能。與 PORTA 口上拉功能有關的特殊功能暫存器有一個：喚醒功能控制暫存器 WAKE-UP(地址為 20H)，其具體定義如下：

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	喚醒允許位： 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： 1：置 PA0 做外部中斷請求腳 0：置 PA0 做普通 I/O 口
4	----	
3~1	Puh3~Puh1	PA 埠 bit3~1 上拉設定位： 1：允許上拉 在 WUE=1 時，上拉功能才生效。此時在設為上拉的 I/O 口輸入一個下降沿信號，將從睡眠狀態喚醒晶片 0：禁止上拉和外部喚醒
0	Puh0	PA 埠 bit0 上拉設定位： 1：允許上拉 在 WUE=1 或 EIS=1 時，上拉功能才生效。當 WUE=1 時，在設為上拉的 I/O 口輸入一個下降沿信號，將從睡眠狀態喚醒晶片；當 EIS=1 時，在設為上拉的 PA0 口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 的硬體的設定位需在 IC 燒錄時在 CONFIG WORD 中設定，CONFIG WORD 如下圖所示：



NOTE1 :

上拉功能必須在喚醒功能使能（"WUE 位"=1）時方可生效，否則無效。

下面是 TM58P10 掃描 4*8 按鍵的 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 : 4*8 按鍵掃描程式，鍵值存放在 key_value 中
;porta 口為列線，四個 I/O 作為按鍵掃描的輸入口，設為上拉
;portb 口為行掃描線，八個 I/O 作為按鍵掃描的輸出口
;32 個按鍵對應鍵值為:k1(01h),k2(02h),k3(03h),k4(04h),k5(05h),
;k6(06h),k7(07h),k8(08h),k9(09h),k10(0ah),k11(0bh),k12(0ch),

```

```
;k13(0dh),k14(0eh),k15(0fh),k16(10h),k17(11h),k18(12h),  
;k19(13h),k20(14h),k21(15h),k22(16h),k23(17h),k24(18h),k25(19h),  
;k26(1ah),k27(1bh),k28(1ch),k29(1dh),k30(1eh),k31(1fh),k32(20h)
```

```
;  
;  
;
```

```
=====
```

```
;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
exintmequ 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
;
;
;=====
;WAKEUP 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
;=====
key_value  equ   09h
time_cnt1  equ   0ah
time_cnt2  equ   0bh
key_buf    equ   0ch

```

```

scan_cnt    equ 0dh
;
;
;=====
;=====
;prog. start here
;=====
;=====
        org    3ffH    ;重定向量地址
        goto   main
        org    000h    ;程式在 ROM 中的起始地址
;
;
;=====
;延時 60ms 副程式
;=====
;定時時間：2+1+1+60*(1+1+250*(1+1+2)+1+2)+2=60306uS=60.306mS
delay60ms:
        movla  3ch
        movam  time_cnt1
loop1:
        movla  0fah
        movam  time_cnt2
loop2:
        nop
        decmsz time_cnt2
        lgoto  loop2
        decmsz time_cnt1
        lgoto  loop1
        ret
;
;
;=====
;按鍵掃描副程式
;=====
key_scan:
        clrm   portb    ;使行線全為低電位
        movm   porta,w
        andla  0fh      ;讀出 porta 埠低 4 位
        xorla  0fh
        btmsc  status,z ;看列線是否全 1
        ret           ;z=1,列線全 1,無鍵按下,退出副程式

```

```

    call    delay60ms ;z=0,列線不全 1,有鍵按下,延時 60ms 消除抖動
    movm   porta,w    ;再次確認是否有鍵按下-----+
    andla  0fh        ;                +
    xorla  0fh        ;                +
    btmsc  status,z   ;-----+
    ret    ;z=1,抖動所致,退出副程式
    clrm   key_buf    ;z=0,確實有鍵按下,清 key_buf
    movla  7fh
    movam  portb      ;行線指向 portb,7
    movla  08h
    movam  scan_cnt   ;掃描按鍵行數為 8 行
scan:
    incm   key_buf,f  ;key_buf 值加 1
    btmsc  porta,0    ;判斷 porta,0 口對應列是否有鍵按下
    ret    ;有鍵按下,退出掃描程式
    incm   key_buf,f  ;無鍵按下,key_buf 值加 1
    btmsc  porta,1    ;判斷 porta,1 口對應列是否有鍵按下
    ret    ;有鍵按下,退出掃描程式
    incm   key_buf,f  ;無鍵按下,key_buf 值加 1
    btmsc  porta,2    ;判斷 porta,2 口對應列是否有鍵按下
    ret    ;有鍵按下,退出掃描程式
    incm   key_buf,f  ;無鍵按下,key_buf 值加 1
    btmsc  porta,3    ;判斷 porta,3 口對應列是否有鍵按下
    ret    ;有鍵按下,退出掃描程式
    bsm    status,c   ;置狀態暫存器 status 進位元位元 c
    rrm    portb,f    ;置另一行線為 0
    decmsz scan_cnt   ;行值減 1
    goto   scan       ;8 行按鍵未全部掃描完,繼續掃描
    clrm   key_buf    ;8 行按鍵全部掃描完清 key_buf
    ret
;
;
;=====
;MAIN PROGRAM
;=====
main::初始化
    movla  00h
    iodir  portb      ;portb 口為輸出態
    movla  0fh

```

```

iodir    porta      ;porta 口為輸入態
movla    b'01001111'
movam    wakeup     ;porta 口四個 I/O 設為上拉
clrm     key_buf    ;清除 key_buf
clrm     key_value  ;清除 key_value
clrm     scan_cnt   ;清除 scan_cnt
;
;
start:;主迴圈
        lcall     key_scan  ;調按鍵掃描程式
        movm     key_buf,w
        movam    key_value ;得到鍵值送 key_value 中
        goto     start
;
;
;=====
        end          ;程式結束
;
;
;=====

```

;範例

;按鍵掃描過程為:採用行掃描法依次對每一行進行掃描，先使被掃描的行為低電位，其他
;所有的行均為高電位，接著檢測各列線的狀態。若各列均為高電位,則所按之鍵不在此行,
;繼續掃描下一行;若列線不為高電位,則所按之鍵在此行。此時退出掃描程式,key_buf 中
;的值即為鍵值。將其送入 key_value 中。例如,當 k6 鍵被按下時，在將行線 portb,6 置為
;低電位時就可檢測到列線 porta,1 為低電位。此時 key_buf 中的鍵值為 06h.

