



## Watch Dog Timer (WDT)功能使用說明(使用外部晶振頻率)

適用產品：SM8952A、SM89516、SM8951B、SM5964、SM5953、SM8952B、SM8954B、SM8958B、SM89516B、SM5952E、SM5954、SM5958、SM5964B

應用說明：

- 1 針對使用新茂 (SyncMOS) MCU，如有因程式設計之關係，需防止當機之發生時，可啟動 Watch Dog 功能，以防止當機之發生。
- 2 此 WDT 功能共有八組重置時間可供選擇。
- 3 應用範圍：1、防當機裝置。2、自動 S/W Reset 功能。

使用方法：

1. 特殊功能暫存器說明：

1.1. WDT 控制暫存器(WDTC, \$9F)

	bit-7					bit-0		
	WDTE	Reserve	Clear	Unused	Unused	PS2	PS1	PS0
Read / Write:	R/W	-	R/W	-	-	R/W	R/W	R/W
Reset value:	0	*	0	*	*	0	0	0

WDTE：WDT 致能位元

當設為 1 時則啟動 WDT 計時功能，初使值為 0。

當 WDT Reset 發生時，此位元會被清成 0，故如果要再次啟動 WDT，則需再一次將此位元設成 1

Clear：WDT 重置位元

當設為 1 時則是將 WDT 16 位元計數值清除為 0，當計數值清除為 0 後，CLEAR 會自動恢復為 0。

PS2~PS0：溢位週期選擇位元

WDT 溢位時間的表示如下圖：

PS[2:0]	Divider (OSC in)	Timer Period (ms) @40MHz
000	8	13.1
001	16	26.21
010	32	52.42
011	64	104.8
100	128	209.71
101	256	419.43
110	512	838.86
111	1024	1677.72

使用 WDT 的方法

1. 如何計算 WDT 在溢位時的休息時間

16 位元計數器 = 65536 clocks 時溢位



40MHz 晶體振盪 = 0.025us/clock

- 當 PS[2:0] = 000 時，分配值 = 8  
WDT 溢位的休息時間 = 0.025us x 65536 x 8 = 13.107ms
- 當 PS[2:0] = 111 時，分配值 = 1024  
WDT 溢位的休息時間 = 0.025us x 65536 x 1024 = 1677.72ms

## 1.2. 系統控制暫存器 (SCONF, \$BF)

	bit-7					bit-0		
	WDR	Unused	Unused	Unused	Unused	ISPE	OME	ALEI
Read / Write:	R/W	-	-	-	-	R/W	R/W	R/W
Reset value:	0	*	*	*	*	0	0	0

WDR: Watch Dog Timer 重置位元

當系統重置時是藉著 WDT 溢位時產生 WDR 將被設定為 1，初使值 = 0。

## 2. 程式範例：

### 2.1. Assembly language：

```

.***** SET DATA *****
,
WDTC      .REG    09FH          ;SFR 內 WDTC 的位址
WDTKEY    .REG    097H        ;SFR 內 WDTKEY 的位址
SCONF     .REG    0BFH        ;SFR 內 SCONF 的位址
.*****
,
      ORG    0000H          ;設定目前程式的起始位址
.***** 判斷 SCONF 的 WDR 是否有 Reset 產生 *****
,
reset:
      MOV    a,SCONF        ;將 SCONF 暫存器的內容放至 A 暫存器中
      JNB   acc.7,poweron_reset ;acc.7 的值若為 0 則跳至 power on_reset
.***** acc.7 為 1 時的執行程式(即 WDT Reset 動作) *****
,
wdt_reset:
      MOV    P1,#00          ;如果進入 WDT RESET 副程序,則 P1 會全亮全滅一次。
      CALL  delay
      ANL   a,#7fh          ;將 SCONF 的 WDR 清除為 0,為了做下一次是否被 RESET 的偵測
      MOV   SCONF,a
      MOV   WDTC,#10000110b ; 啟動 WDT Timer 且開始計數及選擇內部時脈除頻大小
      MOV   P1,#0ffh
      CALL  delay
      JMP   start

```



```

;***** acc.7 為 0 時的執行程式(即設定 WDT 功能) *****
poweron_reset:
    MOV    WDTC,#10000110b    ; 啟動 WDT Timer 且開始計數及選擇內部時脈除頻大小
;***** 執行 main program (LED 閃爍程式由 Port 0 輸出) *****
start:
    MOV    a,#55h
    MOV    P0,a
    CALL   delay
    MOV    a,#0aah
    MOV    P0,a
    CALL   delay
    JMP    start

;***** 執行 Delay 的動作 *****
delay:
    MOV    20H,#08h
b3:    MOV    21H,#0ffh
b2:    CALL   clearWDT        ;呼叫 clearWDT 副程式
    MOV    22H,#0ffh
b1:    DJNZ   22H,b1
    DJNZ   21H,b2
    DJNZ   20H,b3
    RET

;***** 設定 WDTC 的 Clear Bit *****
clearWDT:
    MOV    a,WDTC            ;將 WDTC 的 CLEAR 設定為 1,執行清除計數器的動作
    ORL    a,#20h
    MOV    WDTC,a
    RET
    END

```

## 2.2. C language :

```

#include "SM5964B.h"
#include<intrins.h>/--use "_nop_();"
void WDT_Enable(void) // Watch Dog enable
{

```



```
    WDTC      =  0x86;
}
void WDT_Clear(void) // Watch Dog Disable and counter clear
{
    WDTC      =  WDTC | 0x20;
}
void Delay_x1ms(int time)
{
    int count=0,T_base;
    TMOD=TMOD | 0x01;           // set timer0 mode1
    T_base=0x10000-0x733; //0x10000-0x733;
    while(count<time)
    {
        TH0=T_base/0x100; //catch hi byte
        TL0=T_base%0x100; //catch low byte
        TR0=1;           //set TR0, start timer1 mode1
        while(TF0==0) //TMOD bit-7 is TF0 , if TF0=1 overflow
        {
            _nop_();
            WDT_Clear();
        }
        TR0=0;
        TF0=0;           //clr TF0
        count++;
    }
    TMOD=TMOD & 0xFE;           // stop timer0 mode1
}
void WDT_Reset(void) // Watch Dog Reset
{
    unsigned char temp;
    temp = SCONF & 0x80;
    if (temp==0x80)
    {
        P1 = 0X00;
        SCONF = SCONF & 0X7F;
        Delay_x1ms(300);
    }
}
```



```
    P1 = 0Xff;
    Delay_x1ms(300);
}
}
//=====
void main(void)
{
    WDT_Enable();
    WDT_Reset();
    while(1)
    {
        P0 = 0x55;
        Delay_x1ms(300);
        P0 = 0xAA;
        Delay_x1ms(300);
    }
}
```