

FT64F0AX

WDT Application note

目录

1. 看门狗定时器	3
1.1. 看门狗时钟源.....	3
1.2. WDT 相关寄存器汇总.....	5
2. 应用范例.....	6
联系信息	12

FT64F0Ax WDT 应用

1. 看门狗定时器

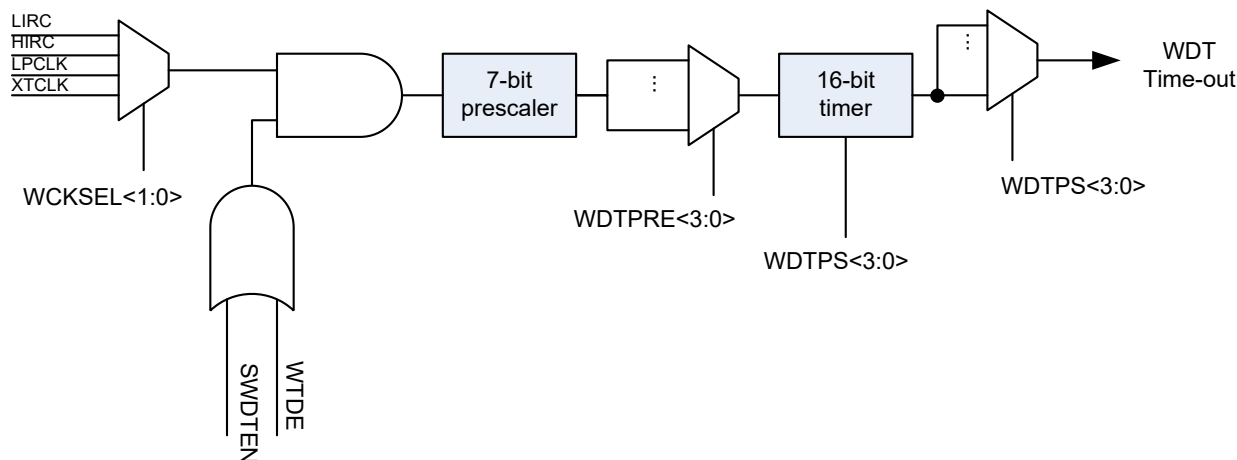


图 1-1 看门狗结构框图

看门狗是一个带 7 位预分频的 16 位计数器，其中预分频和周期可编程，分别由 WDTPRE 和 WDTPS 设置。

WDT 的硬件使能位位于配置寄存器 UCFG0 的第 3 位，WDTEN，软件使能位位于 WDTCON 寄存器的第 0 位，为 1 时表示使能看门狗，为 0 时禁止。

指令 CLRWDT、SLEEP 会清除看门狗计数器。

在使能了看门狗的情况下，处于睡眠时看门狗溢出事件可以作为一个唤醒源，而 MCU 正常工作时 WDT 则是作为一个复位源。

条件	看门狗状态
WDTEN 和 SWDTEN 同时为 0	清零
CLRWDT 指令	
进入 SLEEP、退出 SLEEP 时刻	
写 WDTCON	
写 WCKSEL	

注意：

- 如果内部慢时钟从 32k 切换到 256k 模式 (或反之从 256k 切换到 32k 模式，由 LFMOD 位控制)，都不影响看门狗计时。因为 WDT 时钟源选择 LIRC 时，固定使用 32k 时钟源，见[错误!未找到引用源。](#)的时钟框图；
- PWRT 和 OST 复用了 WDT 定时器，故 PWRT 或 OST 工作时，看门狗的复位功能是暂时屏蔽的；

1.1. 看门狗时钟源

WDT 有 4 种时钟源可选，由寄存器 MISC0 的 WCKSEL 位设置。在 WDT 使能的情况下，所选择的时钟

源被自动使能，并在 SLEEP 模式下保持。

1.2. WDT 相关寄存器汇总

名称	状态	寄存器	地址	复位值
WCKSEL	<u>WDT 时钟源</u> 00 = <u>LIRC</u> 01 = HIRC 10 = LP (只有当 FOSC 选择 LP 模式时才有效*) 11 = XT (只有当 FOSC 选择 XT 模式时才有效*) *否则配置错误, 无 WDT 时钟源	MISC0[1:0]	0x19D	RW-00
WDTPRE	<u>WDT 预分频器</u> 000 = 1 100 = 16 001 = 2 101 = 32 010 = 4 110 = 64 011 = 8 111 = <u>128 (默认)</u>	WDTCN[7:5]	0x97	RW-111
WDTPS	<u>WDT 周期</u> 0000 = 32 0111 = 4,096 0001 = 64 1000 = 8,192 0010 = 128 1001 = 16,384 0011 = 256 1010 = 32,768 0100 = <u>512 (默认)</u> 1011 = 65,536 0101 = 1,024 11xx = 65,536 0110 = 2,048	WDTCN[4:1]		RW-0100
SWDTEN	1 = WDT 使能 0 = <u>WDT 关闭</u> (当 WDTE 选择由 SWDTEN 控制时)	WDTCN[0]		RW-0

表 1-1 WDT 相关用户寄存器

名称	功能	默认
WDTE	<u>WDT</u> <ul style="list-style-type: none"> 使能 (指令不能禁止) <u>通过指令控制 (SWDTEN)</u> 	SWDTEN 控制

表 1-2 WDT 选项初始化配置寄存器

2. 应用范例

```

=====
/* 文件名: ASM_64F0Ax_WDT.ASM
* 功能:    FT64F0Ax_WDT 功能演示
* IC:      FT64F0A5  TSSOP20
* 内部:    16M/2T
* 说明:    程序中开启看门狗并将看门狗时间设置为 32ms
*          a. 主函数先在 DemoPortOut 脚输出一个高 3ms 低 3ms 的信号,
*          b. 然后循环输出高 1ms 低 1ms, 500Hz 的信号。
*          如果不在主程序和延时程序中清看门狗, 则每隔 32ms 单片机复位后, 会重复输
*          出 a+b 信号;
*          如果在主函数中清看门狗, 则一直输出 b 信号
*
*          FT64F0A5  TSSOP20
*          -----
* NC-----|1(PA5)      (PA4)20|-----NC
* NC-----|2(PA6)      (PA3)19|-----NC
* NC-----|3(PA7)      (PA2)18|-----NC
* NC-----|4(PC0)      (PA1)17|-----NC
* NC-----|5(PC1)      (PA0)16|-----NC
* NC-----|6(PB7)      (PB0)15|-----NC
* GND-----|7(GND)     (PB1)14|-----NC
* NC-----|8(PB6)      (PB2)13|-----NC
* VDD-----|9(VDD)     (PB3)12|--DemoPortOut
* NC-----|10(PB5)     (PB4)11|-----NC
*          -----
*/
=====
#include <FT64F0AX.INC>;
=====
;RAM DEFINE
=====
TEMP1      EQU    0X21
TEMP2      EQU    0X22

W_TMP      EQU    0X70
S_TMP      EQU    0X71
=====
;CONSTANT DEFINE
=====
INTCON_DEF EQU    B'00000000' ;禁止所有中断
OSCCON_DEF EQU    B'01110001' ;16MHz 1:1

WPUA_DEF   EQU    B'00000000' ;弱上拉的开关, 0-关, 1-开

```

```

WPUB_DEF EQU B'00000000'
WPUC_DEF EQU B'00000000'

WPDA_DEF EQU B'00000000' ;弱下拉的开关, 0-关, 1-开
WPDB_DEF EQU B'00000000'
WPDC_DEF EQU B'00000000'

TRISA_DEF EQU B'00000000' ;输入输出设置, 0-输出, 1-输入
TRISB_DEF EQU B'00000000' ;PB3-OUT
TRISC_DEF EQU B'00000000'

PSRC0_DEF EQU B'11111111' ;源电流设置最大
PSRC1_DEF EQU B'11111111'
PSRC2_DEF EQU B'00001111'

PSINK0_DEF EQU B'11111111' ;灌电流设置最大
PSINK1_DEF EQU B'11111111'
PSINK2_DEF EQU B'00000011'

ANSELA_DEF EQU B'00000000' ;设置对应的 IO 为数字 IO

WDTCON_DEF EQU B'00001011' ;打开看门狗, 预分频 1: 1,
                                ;定时器周期 1: 1024, 即看门狗周期为 32ms
MISC0_DEF EQU B'00000000' ;看门狗时钟 32kHz
;=====
;USER DEFINE
;=====
#define DemoPortOut PORTB,3
;=====
;PROGRAM START
;=====
ORG 0x0000
LJUMP RESTART
ORG 0x0004
STR W_TMP
SWAPR STATUS,W
STR S_TMP
LJUMP INT_PROGRAM
;=====
;SYSTEM START
;=====
RESTART:
LCALL INITIAL
LCALL WDT_INITIAL

```

```

    BSR          DemoPortOut
    LCALL        DELAY_3MS
    BCR          DemoPortOut
    LCALL        DELAY_3MS
MAIN:
    ;CLRWDT
    BANKSEL     PORTA
    BSR          DemoPortOut
    LCALL        DELAY_3MS
    BCR          DemoPortOut
    LCALL        DELAY_3MS
    LJUMP        MAIN
;=====
;INT_PROGRAM
;=====
INT_PROGRAM:
    SWAPR        S_TMP,0
    STR          STATUS
    SWAPR        W_TMP,1
    SWAPR        W_TMP,0
    RETI
;=====
;SYSTEM  INITIAL
;=====
INITIAL:
    BANKSEL     OSCCON
    LDWI        OSCCON_DEF
    STR          OSCCON

    BANKSEL     INTCON
    LDWI        INTCON_DEF
    STR          INTCON

    BANKSEL     PORTA
    LDWI        0X00
    STR          PORTA
    STR          PORTB
    STR          PORTC

    BANKSEL     TRISA
    LDWI        TRISA_DEF
    STR          TRISA
    LDWI        TRISB_DEF
    STR          TRISB

```



```
LDWI    TRISC_DEF
STR      TRISC
```

```
BANKSEL WPUA
LDWI     WPUA_DEF
STR      WPUA
LDWI     WPUB_DEF
STR      WPUB
LDWI     WPUC_DEF
STR      WPUC
```

```
BANKSEL WPDA
LDWI     WPDA_DEF
STR      WPDA
LDWI     WPDB_DEF
STR      WPDB
LDWI     WPDC_DEF
STR      WPDC
```

```
BANKSEL PSRC0
LDWI     PSRC0_DEF
STR      PSRC0
LDWI     PSRC1_DEF
STR      PSRC1
LDWI     PSRC2_DEF
STR      PSRC2
```

```
BANKSEL PSINK0
LDWI     PSINK0_DEF
STR      PSINK0
LDWI     PSINK1_DEF
STR      PSINK1
LDWI     PSINK2_DEF
STR      PSINK2
```

```
BANKSEL ANSELA
LDWI     ANSELA_DEF
STR      ANSELA
```

```
*****Clear SRAM*****
,
```

```
BANKSEL PORTA
LDWI     0X00
STR      FSR0H
```

CLEAR_RAM_BANK0:

```
LDWI     20H
```

```

        STR        FSR0L
CLEAR_RAM_BANK0_LOOP:
        CLRR       INDF0
        INCR       FSR0L,F
        LDWI       80H
        XORWR      FSR0L,W
        BTSS       STATUS,Z
        LJUMP      CLEAR_RAM_BANK0_LOOP
CLEAR_RAM_BANK1:
        LDWI       0A0H
        STR        FSR0L
CLEAR_RAM_BANK1_LOOP:
        CLRR       INDF0
        INCR       FSR0L,F
        LDWI       00H
        XORWR      FSR0L,W
        BTSS       STATUS,Z
        LJUMP      CLEAR_RAM_BANK1_LOOP
        INCR       FSR0H,F
CLEAR_RAM_LOOP:
        LDWI       10
        SUBWR      FSR0H,W
        BTSS       STATUS,0
        LJUMP      CLEAR_RAM_BANK0
        RET

;=====
;WDT_INITIAL
;=====
WDT_INITIAL:
        CLRWDT
        BANKSEL    MISC0
        LDWI       MISC0_DEF
        STR        MISC0
        BANKSEL    WDTCON
        LDWI       WDTCON_DEF
        STR        WDTCON
        RET

;=====
;DELAY_3MS(16M/2T)
;=====
DELAY_3MS:
        LDWI       H'C8'
        STR        TEMP1
DELAY_3MS_LOOP1:

```

```
LDWI      H'20'
STR        TEMP2
DELAY_3MS_LOOP2:
  DECRSZ   TEMP2,F
  LJUMP     DELAY_3MS_LOOP2
  DECRSZ   TEMP1,F
  LJUMP     DELAY_3MS_LOOP1
  RET

;=====
;DELAY_1MS(16M/2T, 实测为 1.1ms)
;=====
DELAY_1MS:
  LDWI      H'C8'
  STR        TEMP1
DELAY_1MS_LOOP1:
  LDWI      H'0C'
  STR        TEMP2
DELAY_1MS_LOOP2:
  DECRSZ   TEMP2,F
  LJUMP     DELAY_1MS_LOOP2
  DECRSZ   TEMP1,F
  LJUMP     DELAY_1MS_LOOP1
  RET

END
```

联系信息**Fremont Micro Devices (SZ) Corporation**

#5-8, 10/F, Changhong Building
Ke-Ji Nan 12 Road, Nanshan District,
Shenzhen, Guangdong, PRC 518057

Tel: (+86 755) 8611 7811

Fax: (+86 755) 8611 7810

Fremont Micro Devices (HK) Corporation

#16, 16/F, Block B, Veristrong Industrial Centre,
34-36 Au Pui Wan Street, Fotan, Shatin, Hong Kong SAR

Tel: (+852) 2781 1186

Fax: (+852) 2781 1144

<http://www.fremontmicro.com/>

* Information furnished is believed to be accurate and reliable. However, Fremont Micro Devices (SZ) Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents of other rights of third parties, which may result from its use. No license is granted by implication or otherwise under any patent rights of Fremont Micro Devices (SZ) Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. Fremont Micro Devices (SZ) Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of Fremont Micro Devices (SZ) Corporation. The FMD logo is a registered trademark of Fremont Micro Devices (SZ) Corporation. All other names are the property of their respective owners.