FT61F02X

TIMER0 Application note

Table of contents

FT61F02x TIMER0application

1.timer (TIMERS)

in total7timers, including the watchdog timer (WDT)inside.

	WDT	Timer0	Timer1	Timer2	Timer3/4/5
Prescaler (bits)	–	8 (andWDTshared)	3 (1x, 2x, 4x, 8x)	4 (1x, 4x, 16x)	7 (1x, 2x, 4x, 8x, 16x, 32x, 64x, 128x)
counter (bit)	16	8	16	8	12
Postscaler (bits)	7 (andTimer0shared)	–	–	4 (1 – 16x)	–
clock source	-LIRC	- instruction clock - PA2/T0CKI (transition edge count device)	- instruction clock - LP - PA7/T1CKI (rising edge count device)	- 2xcommand bell - 2x HIRC	- HIRC - 2xinstruction clock - PA2/T0CKI (transition edge counter) - PA7/T1CKI (rising edge counter)

surface1-1timer resource

Note: If the clock source of the timer is not the instruction clock, after changingTMRxBefore setting the "TMRxON = 0".

When the timer is enabled, its selected clock source is automatically turned on. When the timer selectsLPWhen the oscillator is used as the clock source, theFOSCmust be configured accordinglyLPmode or choiceINTOSCIOmode, otherwiseLPThe oscillator will be off and will not generate counts.

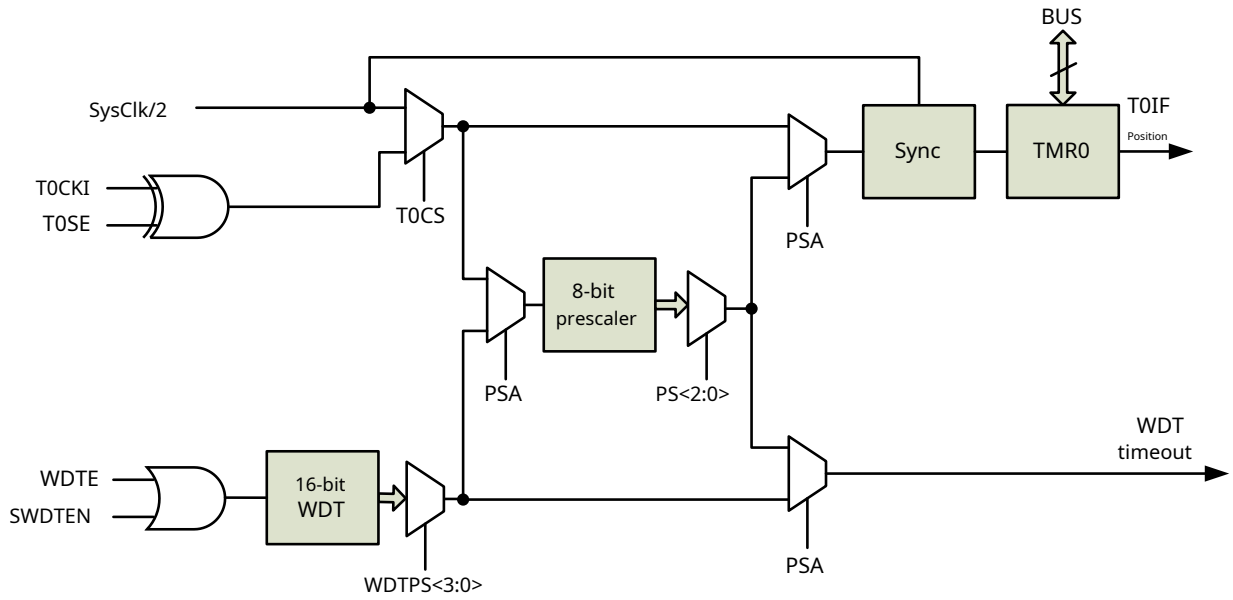
WDTThe postscaler (postscaler)andTimer0The prescaler (prescaler)Share the same hardware frequency division circuit. The hardware circuit is assigned by instruction selection toWDTorTimer0, but both cannot be used at the same time. For timers that are not assigned a divider, the divider ratio is "1".

existPORor system reset, exceptTimer0counter of (counter)The counters, prescalers, and postscalers of all other timers will be reset except . The following events will also reset the counter and divider of the corresponding timer:

	WDT	Timer0	Timer1	Timer2	Timer3/4/5
prescaler	–	- WriteTMR0 - PSAto switch	- TMR1ON = 0 - WriteTMR1L/H	- LIRCandHIRC Cross Calibration Start - WriteT2CON, TMR2L/H - any reset action	- WriteTMRxL/H - WriteTxCKDIV
counter	- WDT, OSToverflow - enter/exitSLEEP - CLRWDT - WriteWDTCON	-Timer0overflow	- TMR1 = PR1 (match, special event trigger) - ECCPtrigger special special event	- TMR2 = PR2 (match)	- TMRx = PRx (BUZZERmode match)
post divider	- except writeWDTCONoutside all of the above conditions - PSAto switch	–		- WriteT2CON, TMR2L/H - any reset action	–

surface1-2 Timer counter and divider reset events

1.1. timer0 (TIMER0)



picture1-1 Timer0Structure diagram

Timer0can be used asI/O "PA2−T0CKI"Rising edge/falling edge counters, or timing timers (the clock source is the instruction clock).

Timer0Count and timeout time = *TMR0[7:0] * Timer0_prescaler*

Timer0An overflow will set the interrupt flag bit (T0IF), whether to triggerto interrupt depends on the corresponding enable control bit (T0IEandGIE).

Note:

1.rightTMR0after a write operation2In an instruction cycle,Timer0stop incrementing;

2.existSLEEPmode,Timer0It will stop counting and maintain the count value before going to sleep;

3.ifTimer0used to pairT0CKIis counted, then relative toTimer0,rightT0CKIThere are minimum period, high/low pulse width requirements. unlessT0CKIvery fast andT_{T0CK}Very slow, otherwise these constraints are usually met;

T0CKI	minimum value	unit	condition
High/Low Pulse Width	0.5*T_{T0CK}+ 20	ns	no prescaler
	10	ns	with prescaler
cycle	20and (T_{T0CK}+40)/Nthe greater of	ns	N = 1, 2, 4, ..., 256 (with prescaler) N = 1 (no prescaler)

4.About "inTimer0andWDTswitching between divider circuits" see**chapter1.1.1**;

1.1.1. Timer0Summary of related registers

name	state				register	address	reset value
T0CS	Timer0clock source		1 =PA2/T0CKI (counter) 0 = Instruction clock (timer)		OPTION[5]	0x81	RW−1
T0SE	Counter trigger edge		1 =falling edge 0 =rising edge		OPTON[4]		RW−1
PSA	1 =The divider circuit is assigned toWDTpost divider 0 = The divider circuit is assigned toTimer0prescaler				OPTION[3]		RW−1
P.S.		WDTPost divider ratio		TIMER0Prescaler	OPTION[2:0]		RW−111
	000		1	2			
	001		2	4			
	010		4	8			
	011	(PSA=1)	18	(PSA=0) 16			
	100		16	32			
	101		32	64			
	110		64	128			
	111		128	256			
	xxx	(PSA=0)	1	(PSA=1) 1			
TMR0[7:0]	Timer0count value				TMR0[7:0]	0x01	RW−xxxx xxxx

surface1-3 Timer0Related User Control Registers

name	state			register	address	reset value
GIE	global interrupt	1 =Enable 0 =global shutdown	(T0IEBe applicable) (wake up is not affected)	INTCON[7]	0x0B 0x8B 0x10B	RW−0
T0IE	Timer0overflow interrupt control bit	1 =Enable 0 =closure	(no wakeup)	INTCON[5]		RW−0
T0IF	Timer0overflow interrupt flag	1 =has overflowed 0 =not overflow	(Latch)	INTCON[2]		RW−0

surface1-4 Timer0Interrupt Enable and Status Bits

1.1.2.existTimer0andWDTSwitch between divider circuits

Shared hardware divider circuitry can be assigned toTimer0orWDTuse when inTimer0andWDTSwitching between frequency divider circuits may cause false reset of the system.

Assign the divider circuit from theTimer0switch toWDT, the following command sequence must be followed:

```
BANKSEL TMR0                ; Can skip if already in TMR0 bank ;
CLRWDT                      Clear WDT
CLRR TMR0                   ; Clear TMR0 and scaler
BANKSEL OPTION
BSR OPTION, PSA             ;Select WDT

LDWI b'11111000'            ; Mask scaler bits (PS2-0)
ANDWR OPTION, W
IORWI b'00000101'           ; Set WDT scaler bits to 32 (or any value desired)
STR OPTION
```

Assign the divider circuit from theWDTswitch toTimer0, the following command sequence must be followed:

```
CLRWDT                      ; Clear WDT and scaler
BANKSEL OPTION
LDWI b'11110000'            ; Mask TMR0 select and scaler bits (PSA, PS2-0)
ANDWR OPTION, W
IORWI b'00000011'           ; Set Timer0 scale to 16 (or any value desired)
STR OPTION
```

2.Application example

```
//*************************************************
********** /*file name:TEST_61F02x_Timer0.c
* Features:      FT61F02x-Timer0Demo
*IC:             FT61F023 SOP16
* Crystal:       16M/2T
* illustrate:    Demo Port Outoutput60Hzduty cycle50%The waveform of -Timer0accomplish
*
*                    FT61F023 SOP16
*                     ------------------------
* VDD----------|1(VDD)      (VSS)16|----------GND
* NC-----------|2(PA7)       (PA0)15|----------NC
* NC-----------|3(PA6)       (PA1)14|--- ----------NC
* NC-----------|4(PA5)       (PA2)13|----------NC
* NC-----------|5(PC3)       (PA3)12|--DemoPortOut
* NC-----------|6(PC2)       (PC0)11|----------NC
* NC-----------|7(PA4)       (PC1)10|----------NC
* NC-----------|8(PC5)       (PC4)09|- ----------NC
*                     ------------------------
*/
//************************************************* ***********
# include "SYSCFG.h"
//************************Macro definition ***************************
# define DemoPortOut PA3 /
*---------------------------------------------- -----
*   Function name:interrupt ISR
*   Features:    timer0interrupt handling
*   enter:    none
*   output:    none
  ------------------------------------------------------
- - - - - * / void interrupt ISR(void)
{
    if(T0IE && T0IF)                    //8.16msFlip once ≈60Hz
    {
        T0IF = 0;
        DemoPortOut = ~DemoPortOut; //flip level
    }
}
/*---------------------------------------------- -
  *   Function name:POWER_INITIAL
  *   Features:    Power-on system initialization
  *   enter:    none
  *   output:    none
  ------------------------------------------------------*/
```

```
void POWER_INITIAL (void) {

    OSCCON = 0B01110001;          //IRCF=111=16MHz/2T=8MHz,0.125μs//Temporarily
    INTCON = 0;                   disable all interrupts
    PORTA = 0B00000000;
    TRISA = 0B00000000;           //PAinput Output0-output1-enter //
                                  PA3->output
    PORTC = 0B00000000;
    TRISC = 0B00000000;           //PCinput Output0-output1-enter //PAPort pull-
    WPUA = 0B00000000;            up control1-pull up0-close pull //PCPort pull-up
    WPUC = 0B0000000;             control1-pull up0-close pull

    OPTION = 0B00001000;          //Bit3=1, WDT MODE, PS=000=WDT RATE 1:1
    MSCKCON = 0B00000000;
    //Bit6->0,prohibitPA4,PC5Regulated output
    //Bit5->0,TIMER2the clock isFosc //Bit4->0,
    prohibitLVR

    CMCON0 = 0B00000111;          //turn off the comparator,Cxfor numbersIOmouth
}
/*---------------------------------------------- -
 *  Function name:TIMER0_INITIAL
 *  Function: Initialize and set the timer0
 *  set upTMR0Timing duration =(1/System clock frequency)*instruction cycle*prescaler value*255
 *                            =(1/16000000)*2*256*255=8.16ms
 ----------------------------------------------
-------- * / void TIMER0_INITIAL (void)
{
    OPTION = 0B00000111;
    //Bit5:   T0CS Timer0Clock Source Selection
    //        1-External pin level changeT0CKI 0-internal clock (FOSC/2)
    //Bit4:   T0CKIpin trigger mode1-falling edge PSA Prescaler Allocation
    //Bit3:   Bits0-Timer0                        1-WDT
    //Bit[2:0]: PS 8prescaler111-1:256

    T0IF = 0;                     //emptyT0software interrupt
}
/*---------------------------------------------- -
 *  Function name:main
 *  Features:   main function
 *  enter:   none
 *  output:  none
 ----------------------------------------------------------
*/ void main()
```

```
{
    POWER_INITIAL();                    //system initialization
    TIMER0_INITIAL();
    GIE =   1;                          //open interrupt
    T0IE = 1;                           //ON Timer/Counter0to interrupt
    while(1)
    {
        NOP();
    }
}
```

Contact information

Fremont Micro Devices 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) Limited

16, 16/F, Block B, Veristong 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