FT61F02X

TIMER2 Application note

Table of contents

FT61F02x TIMER2application

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.

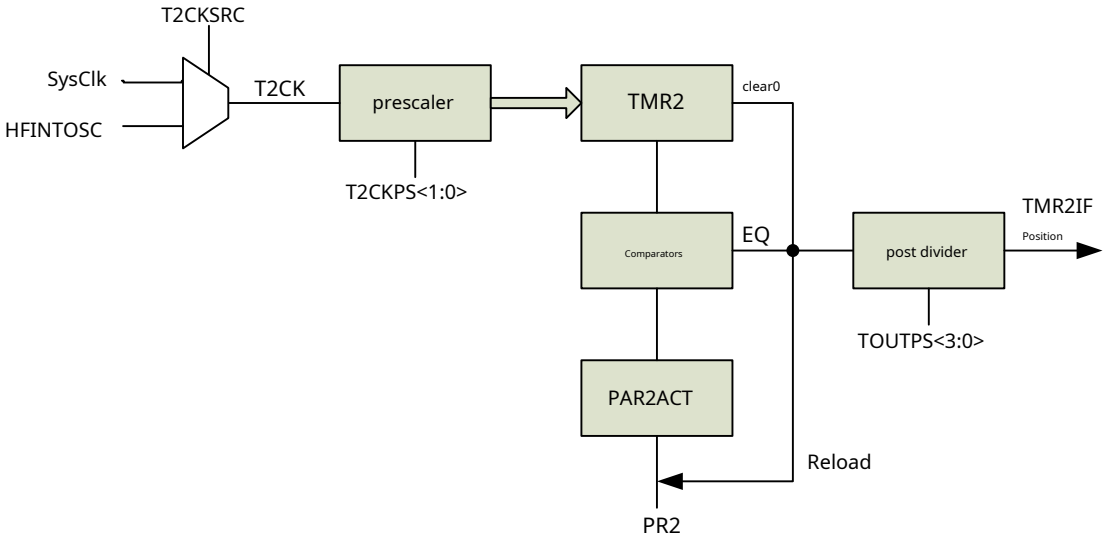
WDTThe postscaler (postscaler)andTimer0The prescaler (prescaler)Share the same hardware frequency division circuit. The hardware circuit is assigned by instruction selection toWDTorTimer0, but both cannot be used at the same time. For timers that are not assigned a divider, the divider ratio is "1".

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 event

1.1. timer2 (TIMER2)



picture1-1 Timer2Structure diagram

Timer2for8Bit timer, the clock source can be system clock or internal32MHzclock(HIRCof2multiplier), can be used forLIRCand HIRCCross Calibration Count (CKCNTI=1). The count match and postscaler overflow functions can be used simultaneously.

Timer2clock is fed intoTimer2Prescaler (the prescale ratio is1, 4or16), the output of the prescaler is used to increment theTMR2register,TMR2 From0x00starts incrementing until it matches thePR2match. When matching:

1. TMR2On the next increment cycle reset to0x00;

2. Timer2Postscaler increments;

3.whenTimer2The incremental output value of the postscaler and the postscaler setting value (1, 2 .... 15or16)when equal,Timer2overflow;

4.interrupt flagTMR2IFplace1, whether to triggerto interrupt and / orwake up from sleep depends on the corresponding enable control bit (GIE, PEIE andTMR2IE);

Note:

1.rightT2CONA write operation does not clear theTMR2register.

2. TMR2andPR2Both are read/write registers. At reset, their values are0x0000and0xFFFF.

3. Timer2The clock source ofMSCKCON.5control whenT2CKSRC=1select internal32MHzClock, independent of the currently running system clock.32MHzThe clock is driven by the internalHIRCmultiplier is obtained, so whenTimer2choose32MHzclock source and TMR2ON=1, even if the system clock selects the internal slow clock or external crystal clock,HIRCIt won't turn off unless it goes into sleep mode.

1.1.1. Timer2Summary of related registers

name	state		register	address	reset value
TOUTPS	Timer2Post divider ratio 0000 = 1 0100 = 5 1000 = 9 1100 = 13 0001 = 2 0101 = 6 1001 = 10 1101 = 14 0010 = 3 0110 = 7 1010 = 11 1110 = 15 0011 = 4 0111 = 8 1011 = 12 1111 = 16		T2CON[6:3]	0x12	RW – 0000
TMR2ON	Timer2enable bit	1 =Enable 0 =closure	T2CON[2]		RW–0
T2CKPS	Timer2Prescaler	00 = 1 1x = 16 01 = 4	T2CON[1:0]		RW–00
T2CKSRC	Timer2clock source	1 = 2x HIRC 0 =2xinstruction clock	MSCKCON[5]	0x1B	RW–0
PR2	PR2period register		PR2[7:0]	0x92	RW–1111 1111
TMR2	TMR2Count result register		TMR2[7:0]	0x11	RW–0000 0000

surface1-3 Timer2Related User Control Registers

name	state		register	address	reset value
GIE	global interrupt 1 =Enable 0 =global shutdown (PEIE, TMR2IEBe applicable) (wake up is not affected)		INTCON[7]	0x0B 0x8B 0x10B	RW–0
PEIE	Total Peripheral Interrupt	1 =Enable (TMR2IEBe applicable) 0 =closure (no wakeup)	INTCON[6]		RW–0
TMR2IE	Timer2andPR2match break	1 =Enable 0 =closure (no wakeup)	PIE1[1]	0x8C	RW–0
TMR2IF	Timer2andPR2match interrupt flag	1 =match (latch) 0 =Mismatch	PIR1[1]	0x0C	RW–0

surface1-4 Timer2Interrupt Enable and Status Bits

1.2. timerx (TIMER3,4,5)

TIMERxas a timer and can also be used to generatePWM (For more information see**chapter**Error! Reference source not found.**PWMx**).

2.Application example

```
//*****************************************************
********** /*file name:TEST_61F02x_Timer2.c
* Features:      FT61F02x-Timer2Demo
*IC:             FT61F023 SOP16
* Crystal:       16M/2T
* illustrate:    whenDemoPortInWhen floating or high level,
*                Demo Port Outoutput5kHzduty cycle50%The waveform of -Timer2achieve when
*                DemoPortInWhen grounded,Demo Port OutOutput high level. Off timer
*
*                      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
* DemoPortIn---|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    Demo Port Out      PA3
#define    DemoPortIn         PC3
/*-------------------------------------------------- -
* Function name:interrupt ISR
* Features:    timer2interrupt handling
* enter:    none
* output:    none
  ----------------------------------------------------
----- * / void interrupt ISR(void)
{
    if(TMR2IE && TMR2IF)                    //100μsinterrupt once
    {
        TMR2IF = 0;
        DemoPortOut = ~DemoPortOut;        //flip level
    }
}
/*-------------------------------------------------- -
* Function name:POWER_INITIAL
* Function: 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 = 0B00001000;           //PCinput Output0-output1-enter
                                  //PC3-enter

    WPUA = 0B00000000;            //PAPort pull-up control1-pull up0-close pull //
    WPUC = 0B00001000;            PCPort pull-up 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:TIMER2_INITIAL
* Function: Initialize and set the timer2
* Set the timing duration = (1/System clock frequency)*instruction cycle*prescaler value*postscaler value*PR2
*                    =1/16000000*2*4*1*200=100μs
  ------------------------------------------------
-------- * / void TIMER2_INITIAL (void)
{
    T2CON = 0B00000001;
    //Bit[6:3]=0000, T2Post divider ratio1:1 //
    Bit[1:0]=01, T2Clock Prescaler1:4

    TMR2 = 0;                     //TMR2assign initial value
    PR2 = 200;                    //assignmentPR2

    TMR2IF = 0;                   //clearTIMER2interrupt flag //
    TMR2IE = 1;                   EnableTIMER2interruption //
    TMR2ON = 1;                   EnableTIMER2start up
```

```
    PEIE = 1;                   //Enable peripheral interrupt
    GIE =  1;                   //enable global interrupt
}
/*--------------------------------------------- -
* Function name:main
* Features:   main function
* enter:    none
* output:   none
  ------------------------------------------------------
*/ void main()
{
    POWER_INITIAL();            //system initialization
    TIMER2_INITIAL();           //initializationT2

    while(1)
    {
        if(DemoPortIn == 1)     //Determine whether the input is high
        {
            TMR2IE = 1;         //start timer2
        }
        else
        {
            TMR2IE = 0;         //off timer2
            DemoPortOut = 1;
        }
    }
}
```

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