FT61F02X

TIMER1 Application note

Table of contents

1.timer (TIMERS) 3

1.1. timer1 (TIMER1).. 4

1.1.1. Timer1Summary of related registers............................... 5

1.1.2. Timer1Register read/write operations................................ 6

1.1.3. Timer1Clock source.. 7

1.1.4. Timer1Prescaler.. 8

1.1.5. Timer1Working in Asynchronous Counter Mode................................. 8

1.1.6. Timer1Gating... 8

1.1.7. Timer1Capture/Compare Time Base................................ 8

2.Application examples... 9

Contact information... 12

FT61F02x TIMER1application

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.

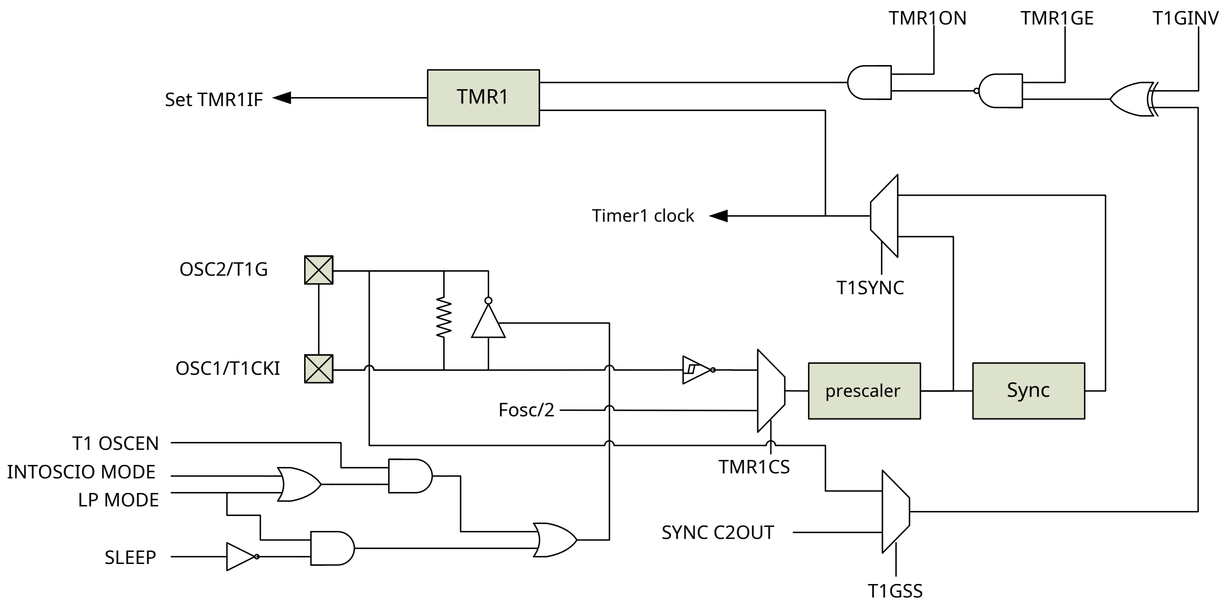
WDTThe postscaler (postscaler)andTimer0The prescaler (prescaler)Share the same hardware frequency division circuit. The hardware circuit is assigned by instruction selection toWDTorTimer0, but both cannot be used at the same time. For timers that are not assigned a divider, the divider ratio is "1".

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. timer1 (TIMER1)



picture1-1 Timer1Structure diagram

TIMER1Is a16Bit timers and counters have the following characteristics:

- a pair16Bit Timer/Counter Register (TMR1H:TMR1L) Programmable
- internal or external clock source
- 3bit prescaler
- optionalLPoscillator
- synchronous or asynchronous operation
- via a comparator orT1GPin'sTimer1Gating (count enable) Overflow
- interrupt
- Wake-up on overflow (external clock only and in asynchronous mode)
- Time Base for Capture/Compare Functions
- Special event triggers (withECCP)
- Comparator output withTimer1Clock synchronization

Timer1module is16Bit up counter. rightTMR1HorTMR1LA write operation will directly update the counter.

When used with an internal clock source, this module is a timer. When used with an external clock source, the module can be used as a timer or counter.

Timer1A pair of registers (TMR1H:TMR1L) increments toFFFFhreturn after0000h.Timer1When the full return,PIR1registerTimer1interrupt flag (TMR1IF)be placed1. Whether to triggerto interrupt and / orwake up from sleep depends on the corresponding enable/disable control bit (TMR1ON, GIE, PEIEandTMR1IE). In the interrupt service routine willTMR1IFClearing the bit will clear the interrupt.

To wake up from sleep, set to asynchronous counter mode. In this mode, an external crystal or clock source signal can be used to increment the counter. otherwise Timer1will stop counting and maintain the count value it had before going to sleep. The required register configuration is as follows:

- must beT1CONregisterTMR1ONLocation1
- must bePIE1registerTMR1IELocation1

- must beINTCONregisterPEIELocation1
- must beT1CONregisterT1SYNCLocation1
- must beT1CONregisterTMR1CSLocation1
- Can beT1CONregisterT1 OSCENLocation1

1.1.1. Timer1Summary of related registers

name	state		register	address	reset value
T1GINV	gating toggle bit	1 =Active high (starts when the gate is high dynamic count) 0 =active low	T1CON[7]	0x10	RW−0
TMR1GE	Gate enable bit	1 =Enable (Clock source is not instruction clock) 0 =no	T1CON[3]		RW−0
T1CKPS	Timer1Prescaler 00 =1 01 = 2 10 = 4 11 = 8		T1CON[5:4]		RW−00
T1 OSCEN	LPoscillator enable bit 1 = LPThe oscillator is enabled for Timer1 0 = LPOscillator off Note: When the clock is configured asLPorLIRCmode, this bit is valid, otherwise this bit is ignored.		T1CON[3]		RW-0
T1SYNC	Timer1External clock input synchronization control bit whenTMR1CS = 1: 1 =out of sync 0 =Synchronize Note: whenTMR1CS = 0, this bit is ignored,Timer1use internal clock		T1CON[2]		RW-0
TMR1CS	Timer1clock source selection	1 = PA7/T1CKI(rising edge) 0 = instruction clock	T1CON[1]		RW-0
TMR1ON	enable bit	1 =Enable 0 =closure	T1CON[0]		RW-0
T1GSS	gating source option bit	1 = T1Gpin (configured as a digital input) 0 =ComparatorsC2Output	CMCON1[1]	0x1A	RW-1
C2SYNC	Comparators2output sync bit 1 =output withTimer1Synchronous to the falling edge of the clock 0 =asynchronous output		CMCON1[0]		RW−0
TMR1L	TMR1Timing and counting result register low8bit		TMR1L[7:0]	0x0E	RW−0000 0000
TMR1H	TMR1Timing and counting result register high8bit		TMR1H[7:0]	0x0F	

surface1-3 Timer1Related User Control Registers

name	state		register	address	reset value
GIE	global interrupt	1 =Enable (PEIE, TMR1IEBe applicable) 0 =global shutdown (wake up is not affected)	INTCON[7]	0x0B 0x8B 0x10B	RW−0
PEIE	Total Peripheral Interrupt	1 =Enable (TMR1IEBe applicable) 0 =closure (no wakeup)	INTCON[6]		RW−0
TMR1IE	Timer1andPR1overflow interrupt	1 =Enable 0 =closure (no wakeup)	PIE1[0]	0x8C	RW−0
TMR1IF	Timer1andPR1overflow interrupt flag	1 =overflow (latch) 0 =not overflow	PIR1[0]	0x0C	RW−0

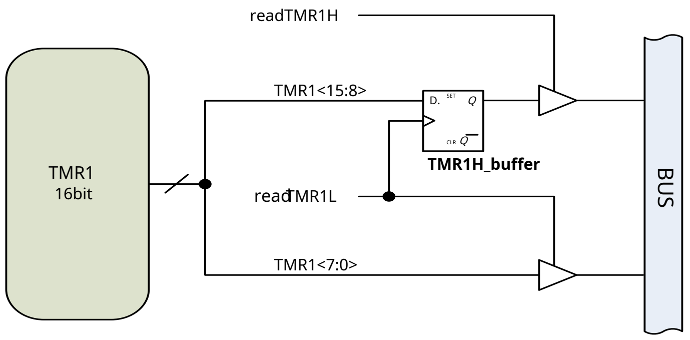
surface1-4 Timer1Interrupt Enable and Status Bits

1.1.2. Timer1Register read/write operations

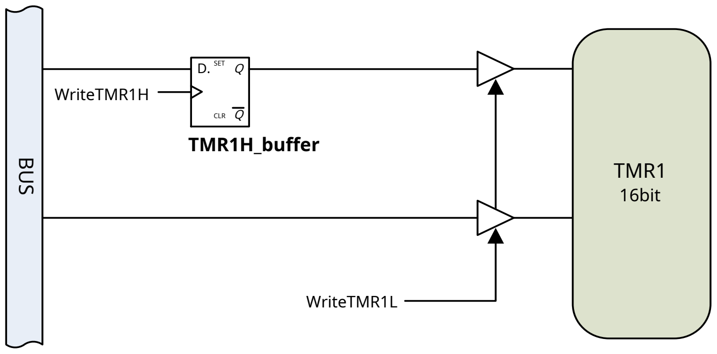
When the timer runs on an external asynchronous clock,TMR1HandTMR1LCannot read or write at the same time. passTMR1Hinternal cache of TMR1H_buf

To resolve this issue, the following read and write sequences must be followed:

- readTMR1when, read firstTMR1L,at this timeTMR1HThe value of will be latched intoTMR1H_buf, then readTMR1H. whenTimer1 When the clock source is not the instruction clock, you need to set "TMR1ON=0"to stop counting, then read theTMR1Executed before1stripNOP instruction.
- WriteTMR1when, first writeTMR1H,at this timeTMR1HThe value of will be stored inTMR1H_buffermiddle. then writeTMR1L,at this time TMR1Hand TMR1Lwill be updated to the count value at the same time. In addition, in order to avoid the contention between write operation and count, before write operation, should set "TMR1ON = 0"to stop counting.



picture1-2TMR1Read operation block diagram



picture1-3 TMR1Write Operation Structure Box

1.1.3.Timer1clock source

T1CONregisterTMR1CSBits are used to select the clock source. whenTMR1CS=0hour, is the instruction clock (2Tmode). whenTMR1CS=1 When, the clock source is provided externally (T1CKIpins).

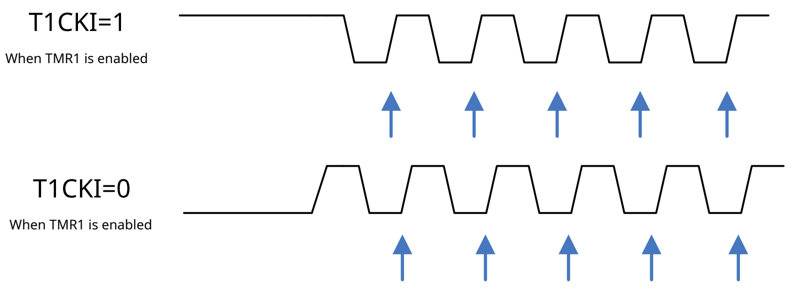
When an external clock source is selected,Timer1The module can work as a timer and counter. when counting,Timer1The external clock inputT1CKI The rising edge increments.

When the clock is configured asLPorINTOSCIOmode,Timer1be usableLPOscillator as clock source.

Notice:

In Counter mode, the counter must experience a falling edge before the first rising edge increments after any one or more of the following conditions:

- POREnable after resetTimer1
- to writeTMR1HorTMR1L
- Timer1Prohibited
- T1CKIwhen highTimer1Prohibited(TMR1ON=0), then inT1CKIwhen lowTimer1enabled (TMR1ON=1).



Notice:

1. The edge pointed by the arrow is the counter increment;

2. In counter mode, a falling edge must pass before the counter increments.

picture1-4TIMER1Incremental edge indication

OSC1(input) pin withOSC2(output) pins connected between low power32.768kHzcrystal, willT1CONregisterT1 OSCEN control position1Enable the oscillator, the oscillator will continue to work during sleep.

becauseTimer1Oscillators and SystemsLPOscillator shared, theTimer1Only when the main system clock comes from the internal oscillator or the oscillator is in LPThis mode can only be used in this mode. The user must provide a software delay to ensure proper oscillator start-up.

whenTimer1The oscillator is enabled when the PORTA[7],PORTA[6]The output drive is disabled, and thePA7andPA6bits read as0. butTRISA7, TRISA6Keep the original value.

Notice:

1.Because the oscillator needs a certain start-up and stabilization time. So in willT1 OSCENplace1and enableTimer1An appropriate delay should be added before;

2.When configured in Oscillator mode, theT1Gfixed output1, so it cannot be used to gateTIMER1.

1.1.4.Timer1prescaler

Timer1There are four prescaler options for clock input1,2,4or8crossover.T1CONregisterT1CKPSbit controls the prescaler counter. The prescaler counter cannot be read and written directly;TMR1HorTMR1Lduring a write operation, orTIMER1 When closed (TMR1ONfor0), the prescaler counter is cleared.

1.1.5.Timer1Works in asynchronous counter mode

whenT1CONRegister Control BitsT1SYNCplace1, the external clock input is not synchronized. The timer increments asynchronously to the internal phase clock. If an external clock source is selected, the timer will continue to run during sleep and can generate an interrupt on overflow to wake up the processor. However, special care should be taken when reading and writing timers (see**chapter1.1.2**).

Notice:

It is possible to miss an increment when switching from synchronous to asynchronous mode. When switching from asynchronous mode to synchronous mode, it is possible to generate one more increment.

1.1.6.Timer1gating

Timer1The gating source is software configurable asT1Gpin or comparatorC2output, the device can be used directlyT1GTime an external event, or use a comparatorC2Timing of simulated events.Timer1For gating source selection, seeCMCON1register.

useT1CONregisterT1GINVbit flippedTimer1Gating, regardless of its originT1Gpin or comparatorC2Output. This will configure Timer1to ensure that there is an active-low or active-high time between events.

1.1.7.Timer1Capture/Compare Timebase

When operating in capture or compare mode,ECCPThe module uses a pair ofTMR1H:TMR1Lregister as the time base.

1.In capture mode,TMR1H:TMR1LThe values of this pair of registers are copied to theCCPR1H:CCPR1L This pair of registers.

2.In compare mode, whenCCPR1H:CCPR1LThis pair of register values andTMR1H:TMR1LWhen the value of matches, an event will be fired. This event can be a special event trigger.

For more information see**chapter**Error! Reference source not found."enhanced capture/compare/PWMmodule".

2.Application example

```
//*****************************************************
********** /*file name:TEST_61F02x_Timer1.c
* Features:      FT61F02x-Timer1Demo
*IC:             FT61F023 SOP16
* Crystal:       16M/2T
* illustrate:    whenDemoPortInWhen floating or high level,
*                Demo Port Outoutput1kHzduty cycle50%The waveform of -Timer1achieve when
*                DemoPortInWhen grounded,Demo Port OutOutput high level. Off timer interrupt
*
*                     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
* Function: Timer1interrupt handling
* set upTimer1Timing duration =(1/system clock cycle)*instruction cycle*prescaler value*(65536-TMR1H:TMR1L)
*                             =(1/16000000)*2*1*4000=500μs
  ---------------------------------------------------
- - - - - * / void interrupt ISR(void)
{
        if(TMR1IF)
      {
            TMR1IF = 0;
            TMR1L = 0X60;                    //timing500μs=>TMR1=4000*0.125μs=500μs //
                                             initial value =65536-4000=61536=>0XF060 //Assign
            TMR1H = 0XF0;                    initial value =>TMR1H=0XF0;TMR1L=0X60 //flip level
            DemoPortOut = ~DemoPortOut;
      }
}
```

```
/*------------------------------------------------ -
* 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 = 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:TIMER1_INITIAL
* Function: Initialize and set the timer1
* Set the timing duration =(1/system clock cycle)*instruction cycle*prescaler value*(65536-TMR1H:TMR1L)
*                       =(1/16000000)*2*1*4000=500µs
  ------------------------------------------------
-------- * / void TIMER1_INITIAL (void)
{
        //Need to reassign the initial value in the interrupt
        T1CON = 0B00000000;
        //Bit[5:4]=00,T2clock frequency division1:1 //
        Bit1=0,T1Clock source selection internal clock
        TMR1L = 0X60;                       //TMR1Assign the initial value to the lower eight bits
        TMR1H = 0XF0;                       //TMR1Assign the initial value to the upper eight bits

        TMR1IE = 1;                         //EnableTMER1interruption
        TMR1ON = 1;                         //EnableTMER1start 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
    TIMER1_INITIAL();                       //initializationT1
    while(1)
    {
        if(DemoPortIn == 1)                 //Determine whether the input is high
        {
            TMR1IE = 1;                     //start timer1to interrupt
        }
        else
        {
            TMR1IE = 0;                     //off timer1to interrupt
            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