Application Note

32-bit Cortex™-M0 MCU NuMicro® Family

How to use Timer?

Table of Contents-

1 INTRODUCTION

The timer module includes four channels, TIMER0~TIMER3 (TIMER0 and TIMER1 are at APB1 and TIMER2 and TIMER3 are at APB2), which allow user to easily implement a counting scheme or timing control for applications. The timer can perform functions like frequency measurement, event counting, interval measurement, clock generation, delay timing, and so on. The timer can generates an interrupt signal upon timeout, or provide the current value of count during operation. The details of feature and structure are as follows.

1.1 Feature

- Independent clock source for each channel (TMR0_CLK, TMR1_CLK, TMR2_CLK, TMR3_CLK).
- Time out period = (Period of timer clock input) * (8-bit Prescale + 1) * (24-bit TCMP)
- Maximum counting cycle time = (1 / 25 MHz) * (2^8) * (2^24) =171.8 second, if TCLK = 25 MHz.
- Internal 24-bit up counter is readable through TDR (Timer Data Register).
- Include three operation mode (one shot, periodic and continuous counting mode)

1.2 Structure

Each channel is equipped with an 8-bit pre-scale counter, a 24-bit up-counter, a 24-bit compare register and an interrupt request signal. Refer to Figure 1 for the timer controller block diagram. There are five options of clock sources for each channel, Figure 2 illustrate the clock source control function.

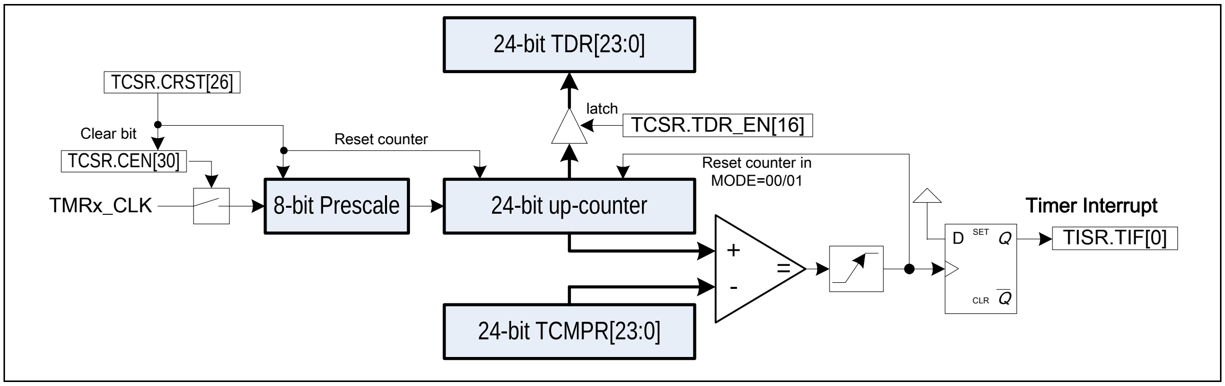

Figure 1 Timer Controller Block Diagram

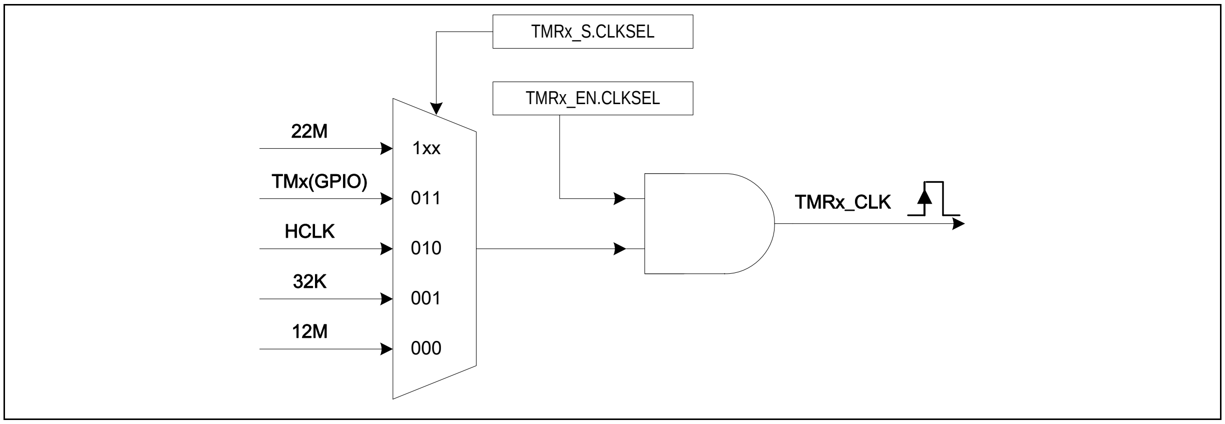

Figure 2 Clock Source of Timer Controller

2 HOW TO PROGRAM TIMER

2.1 PROGRAM FLOW OF TIMERx

1. Select **TMRx_S** bit in **CLKSEL1** register and set the **TMRx_EN** bit in **APBCLK** register to enable **TIMERx** clock source
2. Select **MODE** bit in **TCSR** register for operation mode
3. Select **PRESCALE** bit in **CLKSEL1** register and **TICR(TCMP)** register to define Time out period = (Period of timer clock input) * (8-bit Prescale + 1) * (24-bit TCMP)
4. Set **IE** bit in **TCSR** register and **ISER** to enable interrupt.
5. Set **CEN** bit in **TCSR** register to enable Timer module

2.2 Sample code

```
#include <stdio.h>
#include "NUC1xx.h"
/*---------------------------------------------------------------------------
  Defline variable
  ---------------------------------------------------------------------------*/
            static uint32_t TimerCOUN=0;
/*---------------------------------------------------------------------------
  Function subroutine
  ---------------------------------------------------------------------------*/
void Delay(uint32_t delayCnt)
{
    while(delayCnt--)
    {
        __NOP();
        __NOP();
    }
}
uint32_t GetTDR(void)
{
            return    TIMER0->TDR;
}
/*---------------------------------------------------------------------------
  Interrupt subroutine
  ---------------------------------------------------------------------------*/
void TMR0_IRQHandler(void) // Timer0 interrupt subroutine
```

```
{
        TIMER0->TISR.TIF =1;
        TimerCOUN++;
}
/*---------------------------------------------------------------------------
  MAIN function
  ---------------------------------------------------------------------------*/
int32_t main (void)
{
        uint32_t TEMP=0;

        NVIC_DisableIRQ(TMR0_IRQn);         //Disable Timer0 interrupt
        outpw(&TIMER0->TCSR ,0 );           //Disable Timer0

        /* Step 1. Enable and Select Timer clock source */
        SYSCLK->CLKSEL1.TMR0_S = 4;         //Select 22Mhz for Timer0 clock source
        SYSCLK->APBCLK.TMR0_EN =1;          //Enable Timer0 clock source

        /* Step 2. Select Operation mode */
        TIMER0->TCSR.MODE=1;                //Select periodic mode for operation mode

        /* Step 3. Select Time out period = (Period of timer clock input) * (8-bit Prescale + 1) * (24-bit
        TCMP)*/
        TIMER0->TCSR.PRESCALE=0;            // Set Prescale [0~255]
        TIMER0->TICR = 2765;                // Set TICR(TCMP) [0~16777215]
                                        // (1/22118400)*(0+1)*(2765)= 125.01usec or 7999.42Hz

        /* Step 4. Enable interrupt */
        TIMER0->TCSR.IE = 1;
        TIMER0->TISR.TIF = 1;               //Write 1 to clear for safety
        NVIC_EnableIRQ(TMR0_IRQn);//Enable Timer0 Interrupt

        /* Step 5. Enable Timer module */
        TIMER0->TCSR.CRST = 1;              //Reset up counter
        TIMER0->TCSR.CEN = 1;               //Enable Timer0
        TIMER0->TCSR.TDR_EN=1;                   // Enable TDR function

        while(1)
        {
                Delay(1000);
                TEMP = GetTDR();                 //Get value of up counter

                if (TimerCOUN>40000) break;      //Wait 5 seconds
        }

        NVIC_DisableIRQ(TMR0_IRQn);              //Disable Timer0 interrupt
        outpw(&TIMER0->TCSR ,0 );                //Disable Timer0

    return 0;
}
```

3 REVISION HISTORY

REV.	DATE	DESCRIPTION
1.00	March 1, 2010	1. Initially issued.
1.01	April 8, 2010	1. Remove register description

Important Notice

Nuvoton products are not designed, intended, authorized or warranted for use as components in systems or equipment intended for surgical implantation, atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for other applications intended to support or sustain life. Further more, Nuvoton products are not intended for applications wherein failure of Nuvoton products could result or lead to a situation wherein personal injury, death or severe property or environmental damage could occur.

Nuvoton customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from such improper use or sales.

Please note that all data and specifications are subject to change without notice. All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.