Nano100 1Hz Timer with LIRC

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This document describes how to use Timer0 clock source from LIRC and generate interrupt every 1 second.
BSP Version	NANO100 Series BSP CMSIS V3.02.000
Hardware	NuTiny-EVB-NANO120-LQFP100_V001

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1 Function Description

1.1 Introduction

NANA100 has four timers, including Timer0, Timer1, Timer2, and Timer3. This example code uses Timer0 as sample. We set Timer0 clock source from LIRC and Timer0 frequency to 1 Hz.

1.2 Demo Result

We compile the example code and download it to the NuTiny-EVB-Nano120-LQFP100_V001 development board. Time0 generates interrupt every 1 second and print the interrupt message to UART console, as shown in Figure 1.

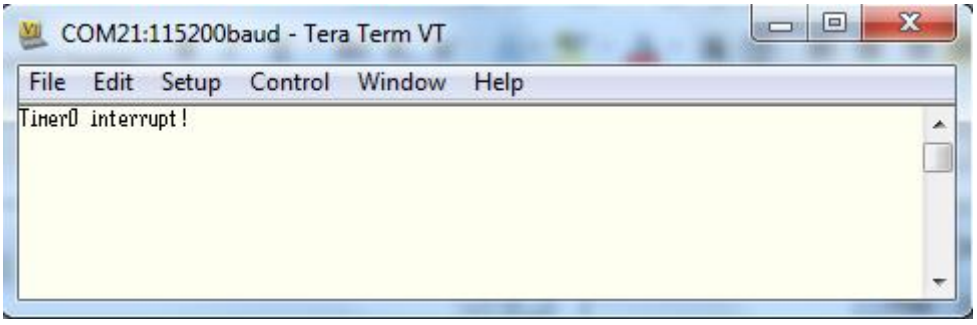


Figure 1 Timer0 generate interrupt every 1 second

Besides, GPIO port A0 output will toggle whenever Timer0 interrupt occurs.

2 Code Description

In this example, we have to enable Timer0 clock and set Timer0 clock source from LIRC. Besides, we use UART to print Timer0 interrupt message. Hence, we also have to enable UART0 clock and set UART0 clock source from HXT. Following is the related code segment in SYS_Init() of main.c

```
    /* Enable IP clock */
    CLK_EnableModuleClock(UART0_MODULE);
    CLK_EnableModuleClock(TMR0_MODULE);

    /* Select IP clock source */
    CLK_SetModuleClock(UART0_MODULE, CLK_CLKSEL1_UART_S_HXT, CLK_UART_CLK_DIVIDER(1));
    CLK_SetModuleClock(TMR0_MODULE, CLK_CLKSEL1_TMR0_S_LIRC, 0);
```

In main routine, we use TIMER_Open() to set Timer0 to periodic mode and Timer0 frequency to 1 Hz. Then, we use TIMER_EnableInt() to enable Timer0 interrupt and NVIC_EnableIRQ() to enable Timer0 NVIC.

```
    /* Set Timer0 frequency to 1Hz */
    TIMER_Open(TIMER0, TIMER_PERIODIC_MODE, 1);

    /* Enable Timer0 interrupt */
    TIMER_EnableInt(TIMER0);

    /* Enable Timer0 NVIC */
    NVIC_EnableIRQ(TMR0_IRQn);
```

TMR0_IRQHandler() is Timer0 interrupt service routine. Whenever Timer0 interrupt occurs, CPU executes TMR0_IRQHandler(). In TMR0_IRQHandler(), the output of GPIO port A0 toggles. Then using TIMER_ClearIntFlag() to clear Timer0 interrupt flag and print Timer0 interrupt message.

```
void TMR0_IRQHandler(void)
{
    PA0 = PA0 ^ 1;
    TIMER_ClearIntFlag(TIMER0);
    printf("Timer0 interrupt!\n");
}
```

3 Software and Hardware Environment

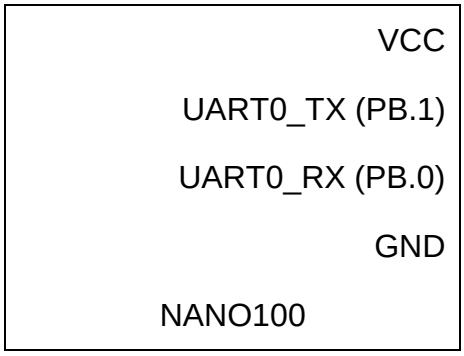
Software Environment

- BSP version
  - NANO100 Series BSP CMSIS v3.02.000
- IDE version
  - Keil uVersion 5.26

Hardware Environment

- Circuit components
  - NuTiny-EVB-Nano120-LQFP100_V001
- Diagram

We use printf function to output Timer0 interrupt message and the UART0 TX RX pin are PB.1 and PB.0 respectively.



4 Directory Information

- EC_Nano100_1Hz_Timer_with_LIRC_V1.00
  - Library — Sample code header and source files
    - CMSIS — Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
    - Device — CMSIS compliant device header file
    - StdDriver — All peripheral driver header and source files
  - SampleCode
    - ExampleCode — Source file of example code

5 How to Execute Example Code

1. Browsing into sample code folder by section 4 Directory Information and double click Nano100_1Hz_Timer_with_LIRC.uvproj.
2. Enter Keil compile mode
   a. Build
   b. Download
   c. Start/Stop debug session
3. Enter debug mode
   a. Run

6 Revision History

Date	Revision	Description
Jun. 17, 2019	1.00	1. Initially issued.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.