Mini51DE_Software_RTC

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This document describes how to use timer periodic mode to emulate RTC function.
BSP Version	Mini51DE Series BSP CMSIS V3.02.000
Hardware	NuTiny-EVB-Mini51_V2.1

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

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

www.nuvoton.com

1 Function Description

1.1 Introduction

This example code demonstrates how to use timer periodic mode to emulate RTC function. We configure Timer0 in periodic mode. Enable interrupt and one interrupt tick per second. Hence, we can get the latest time related information every second.

1.2 Demo Result

We compile the example code and download it to the NuTiny-EVB-Mini51_V2.1 development board. The program shows current time related information including year, month, day, hour, minute, and second, as shown in Figure 1.

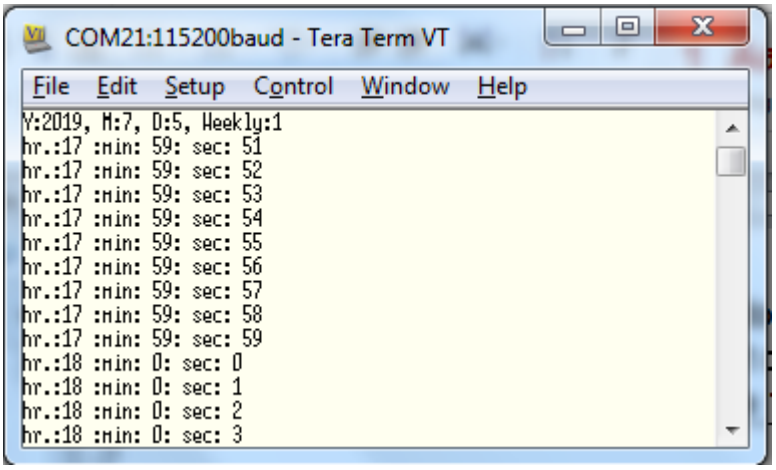


Figure 1 Software RTC

2 Code Description

In this example, we have to configure UART and Timer0 clock setting. Following is the related code segment in SYS_Init() of main.c

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

    /* Select IP clock source */
    CLK_SetModuleClock(UART_MODULE, CLK_CLKSEL1_UART_S_XTAL, CLK_CLKDIV_UART(1));
    CLK_SetModuleClock(TMR0_MODULE, CLK_CLKSEL1_TMR0_S_XTAL, 0);
```

In main routine, initializing Timer0 as following

- Set Timer0 to periodic mode
- Enable interrupt and 1 interrupt tick per second
- Enable Timer0 NVIC

```
    /* Open Timer0 in periodic mode, enable interrupt and 1 interrupt tick per second */
    TIMER_Open(TIMER0, TIMER_PERIODIC_MODE, 1);
    TIMER_EnableInt(TIMER0);

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

    /* Start Timer0 counting */
    TIMER_Start(TIMER0);
```

TMR0_IRQHandler () is Timer0 interrupt service routine. Since Timer0 interrupt occurs every second, setting a flag, t_flag, to notify main routine that time related information should be updated.

```
void TMR0_IRQHandler(void)
{
    if(TIMER_GetIntFlag(TIMER0) == 1)
    {
        /* Clear Timer0 time-out interrupt flag */
        TIMER_ClearIntFlag(TIMER0);
        t_flag=1;

    }
}
```

In main routine, polling the flag, t_flag, till it is set to 1. Then update time related information.

```
void SoftwareRTC(void)
{
    if(t_flag == 0)
        return;
    sec++;
    if(sec == 60)
    {
        sec = 0;
        min++;
    }
    if(min == 60)
    {
        min=0;
        hour++;
    }
    if(hour == 24)
    {
        hour = 0;
        g_day++;
        g_weekly++;
    }
    printf("hr.:%d :min: %d: sec: %d\n\r",hour,min,sec);
    t_flag = 0;
}
```

3 Software and Hardware Environment

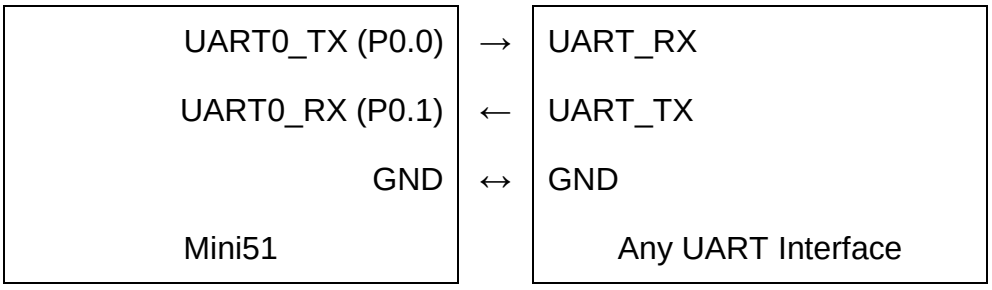
Software Environment

- BSP version
  - Mini51DE Series BSP CMSIS V3.02.000
- IDE version
  - Keil uVersion 5.26

Hardware Environment

- Circuit components
  - NuTiny-EVB-Mini51_V001
- Diagram

We use printf function to output current time related message. The UART0 TX RX pins are P0.0 and P0.1 respectively.



4 Directory Information

- Mini51DE Series BSP CMSIS V3.02.000 – Software RTC
  - 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 Software_RTC.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
July. 11, 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.

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.