Pending IRQ On WakeUp

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	Use TIMER to wake up the system and pending the interrupt service, first execute the main program
BSP Version	M051 Series BSP CMSIS v3.01.002
Hardware	NuTiny-EVB-M051_V3.0

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

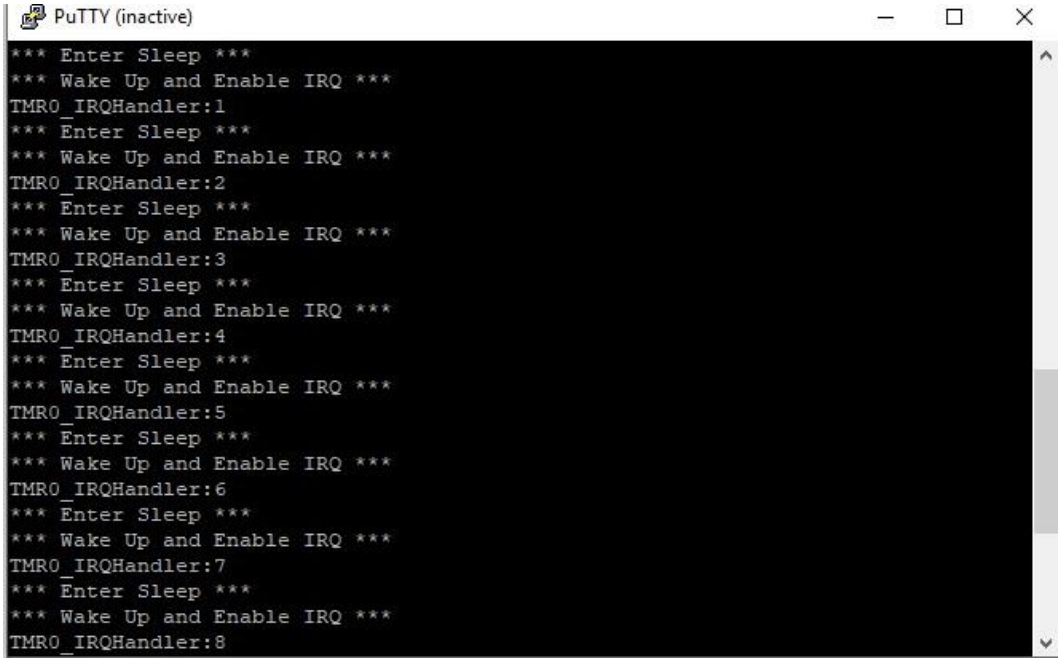
1.1 Introduction

Under normal circumstances, when the power saving mode is woken up, it will first enter the interrupt service routine, and then execute the part of the main program that has not been executed after the wakeup. This example is an instruction provided by ARM, which allows the system to delay entering the interrupt service after the power-saving mode is woken up, execute the main program first, and then execute the interrupt service routine.

1.2 Principle

First, set the counter count interrupt mode and interrupt wake-up of the TIMER0. After the TIMER0 is turned on, use the ARM-provided disable interrupt instruction, and then the system enters the power-saving mode. Wait for the TIMER0 to count until the interrupt is generated. After the system wakes up, use the ARM-provided enable interrupt instruction to enter the interrupt service routine.

1.3 Demo Result



2 Code Description

TIMER interrupt service routine, clear interrupt flag and wake-up flag, print out the number of times to enter the interrupt service routine：

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

        g_au32TMRINTCount[0]++;
        printf("TMR0_IRQHandler:%d\n", g_au32TMRINTCount[0]);
    }
}
```

Set the TIMER function and the corresponding interrupt service routine to wake up the system：

```
/* Open Timer0 frequency to 1 Hz in periodic mode, and enable interrupt */
TIMER_Open(TIMER0, TIMER_PERIODIC_MODE, 1);
TIMER_EnableInt(TIMER0);
TIMER_EnableWakeup(TIMER0);
/* Enable Timer0 NVIC */
NVIC_EnableIRQ(TMR0_IRQn);
/* Start Timer0 counting */
TIMER_Start(TIMER0);
```

Use the interrupt service switch instruction and print out the point in time when the program is executed：

```
    __disable_irq();

    printf("*** Enter Sleep ***\n");
    UART_WAIT_TX_EMPTY(UART0);
    CLK_PowerDown();

    printf("*** Wake Up and Enable IRQ ***\n");
    __enable_irq();
```

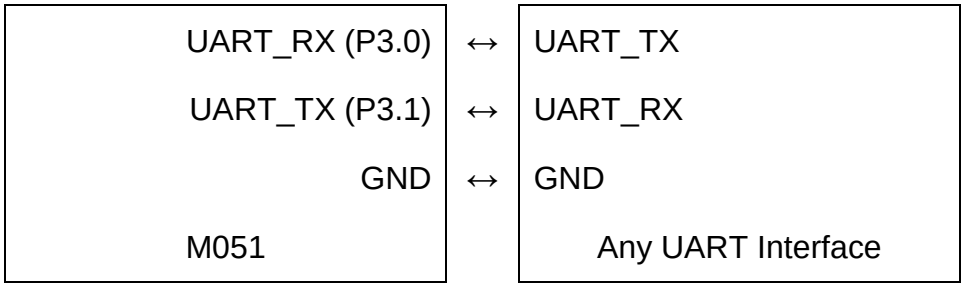
3 Software and Hardware Environment

Software Environment

- BSP version
  - M051 Series BSP CMSIS v3.01.002
- IDE version
  - Keil uVersion 5.26

Hardware Environment

- Circuit components
  - NuTiny-EVB-M051_V3.0
- Diagram



4 Directory Information

EC_M051_Pending_IRQ_On_WakeUp_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 Directory Information (section 4) and double click M051_Pending_IRQ_On_WakeUp.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
Aug. 30, 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.