NANO100B Timer Wakeup

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This code initial some peripheral and demo wakeup by Timer
BSP Version	Nano100B Series BSP CMSIS v3.02.002
Hardware	NuTiny-EVB-Nano130-LQFP128 v1.0

1 Function Description

1.1 Introduction

This example code will demo how to initial the GPIO, Timer, ADC, PWM and SPI. When the system power down, it will wake up by Timer interrupt.

1.2 Principle

The program setup different timer period, demo how to initial external interrupt, PWM, ADC, and SPI, and the system power down will wake up by Timer interrupt.

1.3 Demo Result

```
Hello World.
PLL Clock = 84000000 Hz
Core Clock = 42000000 Hz

timer count 1
ADC done!
CH0 = 0x0fff, CH1 = 0x0fff
SPI TX done!
timer count 2
ADC done!
CH0 = 0x0fff, CH1 = 0x0fff
SPI TX done!
timer count 3
ADC done!
CH0 = 0x0fff, CH1 = 0x0fff
SPI TX done!
timer count 4
ADC done!
CH0 = 0x0fff, CH1 = 0x0fff
SPI TX done!
timer count 5
ADC done!
CH0 = 0x0fff, CH1 = 0x0fff
SPI TX done!
```

2 Code Description

Timer initial:

```
    void Timer0_Init(void)
    {
        /* Set Timer0 as Peridic mode and 5 sec */
        TIMER0->CMPR = 0xC350;        // 5 sec.
        TIMER0->PRECNT = 0;
        TIMER0->CTL = TIMER_CTL_TMR_EN_Msk | TIMER_CTL_WAKE_EN_Msk | TIMER_PERIODIC_MODE;
        /* Enable Timer0 interrupt */
        TIMER_EnableInt(TIMER0);
        NVIC_EnableIRQ(TMR0_IRQn);
        /* Clear Timer0_flag */
        Timer0_flag = 0;
    }
```

External interrupt 0:

```
void INT0_Init(void)
{
    /* Set PB.9 as Input */
    GPIO_SetMode(PB, BIT9, GPIO_PMD_INPUT);
    /* Enable PB.9 Pull-up resister */
    PB->PUEN |= 0x200;
    /* Enable interrupt by falling edge trigger */
    GPIO_EnableEINT0(PB, 9, GPIO_INT_FALLING);
    NVIC_EnableIRQ(EINT0_IRQn);
}
```

ADC Initial:

```
void ADC_Init(void)
{
    /* Enable channel 0~3 and set as Single Cycle mode */
    ADC_Open(ADC, ADC_INPUT_MODE_SINGLE_END, ADC_OPERATION_MODE_SINGLE_CYCLE,
(ADC_CH_0_MASK | ADC_CH_1_MASK | ADC_CH_2_MASK | ADC_CH_3_MASK));
    /* Power on ADC */
    ADC_POWER_ON(ADC);
    /* Enable ADC ADC_IF interrupt */
    ADC_EnableInt(ADC, ADC_ADF_INT);
    NVIC_EnableIRQ(ADC_IRQn);
    /* Clear ADC_flag */
    ADC_flag = 0;
```

```
}
```

SPI initial:

```
void SPI_Init(void)
{
    /* Configure SPI0 as a master, MSB first, 8-bit transaction, SPI Mode-0 timing, clock is 2MHz */
    SPI_Open(SPI0, SPI_MASTER, SPI_MODE_0, 8, 2000000);
    /* SPI serial clock = 42 MHz/ (6+1) = 6 MHz */
    SPI0->CLKDIV = 6;
    /* Enable the automatic hardware slave select function. Select the SS0 pin and configure as low-active. */
    SPI_EnableAutoSS(SPI0, SPI_SS0, SPI_SS0_ACTIVE_LOW);
    SPI_EnableFIFO(SPI0, 4, 4);
}
```

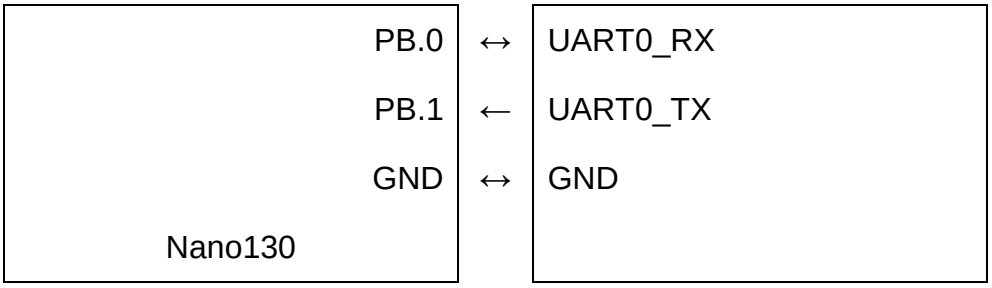
3 Software and Hardware Environment

Software Environment

- BSP version
  - Nano100B Series BSP CMSIS v3.02.002
- IDE version
  - Keil uVersion 5.15

Hardware Environment

- Circuit components
  - NuTiny-EVB-Nano130-LQFP128 V1.0
- Diagram



4 Directory Information

- EC_NANO100B_PeripheralInit_and_TimerWakeup_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 NANO100B_PeripheralInit_and_TimerWakeup.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
June. 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.