# UART wake up and compensation

Example Code Introduction for 32-bit NuMicro® Family

## Information

| | |
|---|---|
| Application | This sample code uses the UART to wake up the M480 from power down mode and perform frequency compensation. |
| BSP Version | M480 Series BSP CMSIS V3.04.000 |
| Hardware | NuMaker-PFM-M487 Ver 3.0 |

*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.*

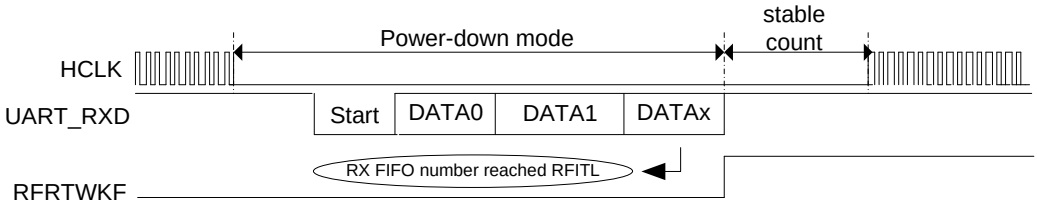
www.nuvoton.com

# 1 Function Description

## 1.1 Introduction

In this example code, the M480 will enter the power down mode. When the received UART data reaches the threshold, it will wake up the M480. Since the UART clock source needs to be set to LXT in power down mode, UART baud rate compensation is used to achieve a more accurate frequency. Set the register WKRFRTEN (UART_WKCTL[2]) to enable the UART wake-up function.

After entering the power down mode, when the data received in the RX FIFO reaches the threshold setting RFITL (UART_FIFO[7:4]), the receive data flag RFRTWKF (UART_WKSTS[2]) is set. The following figure shows the wake-up of the UART received data:




## 1.2 Principle

User can set the bit to be compensated using BRCOMP (UART_BRCOMP [8: 0]) and set the positive or negative compensation using BRCOMPDEC (UART_BRCOMP [31]) to increase or decrease the bit data time by 1 UART clock.

Low Power UART registers:

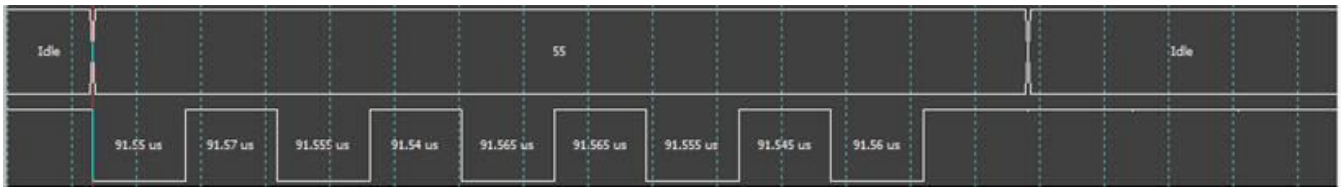
| Bits | Description | |
|---|---|---|
| [31] | **BRCOMPDEC** | **Baud Rate Compensation Decrease**<br>0 = Positive (increase one module clock) compensation for each compensated bit.<br>1 = Negative (decrease one module clock) compensation for each compensated bit. |
| [30:9] | **Reserved** | Reserved. |
| [8:0] | **BRCOMP** | **Baud Rate Compensation Patten**<br>These 9-bits are used to define the relative bit is compensated or not.<br>BRCOMP[7:0] is used to define the compensation of UART_DAT[7:0] and BRCOM[8] is used to define the parity bit. |

Example:

UART peripheral clock = 32.768K (LXT)

UART Baud rate = 9600, 1bit = 104.167us

The original waveform represents 1 bit UART data ( (1 / 32.768K) * 3 = 91.55us) according to 3 UART clocks. The waveform is as follows:




The register setting is as follows:

BRCOMP[8:0] = 0x0010100101

BRCOMPDEC[31] = 0

1 UART clock time compensated at bit0, bit2, bit5, bit7 = ((1/32.768K) * (3+1) = 122.07us)

The waveform is as follows:

When the compensation function is enabled, the UART baud rate error can be controlled within ± 0.5 UART clock.

## 2 Code Description

First, set baud rate of UART1 to 9600, and perform frequency compensation for bit0, bit2, bit5, and bit7. Then enable the UART wake-up function, and then set the threshold of the UART receiving data. The code is as follows:

```
void UART_FunctionTest()
{
    SYS_UnlockReg();

    /* Configure UART1 and set UART1 baud rate */
    UART_Open(UART1, 9600);

    /* Set Compensation in bit0, bit2, bit5, bit7 */
    UART1->BRCOMP = 0xA5;

    g_bWait = 0;

    UART1->WKCTL = UART_WKCTL_WKRFRTEN_Msk;

    /* Enable UART RDA/THRE/Time-out interrupt */
    NVIC_EnableIRQ(UART1_IRQn);
    UART_EnableInt(UART1, (UART_INTEN_WKIEN_Msk));

    CLK->PWRCTL |= CLK_PWRCTL_PDWKIEN_Msk;
    NVIC_EnableIRQ(PWRWU_IRQn);

    /* clear status */
    UART1->WKSTS = UART1->WKSTS;

    /* Wait debug message finish */
    while((UART0->FIFOSTS & UART_FIFOSTS_TXEMPTYF_Msk) == 0);

    UART1->FIFO &=~ UART_FIFO_RFITL_Msk;
    UART1->FIFO |= UART_FIFO_RFITL_4BYTES;

    /* Enter to Power-down mode */
    CLK_PowerDown();

    while(!g_bWait);
```

```
    CLK_SysTickLongDelay(1000000);
    /* Disable UART RDA/THRE/Time-out interrupt */
    UART_DisableInt(UART1, UART_INTEN_WKIEN_Msk);
    g_bWait = TRUE;

    printf("\n\n%x\n",UART_READ(UART1));
    printf("\n\n%x\n",UART_READ(UART1));
    printf("\n\n%x\n",UART_READ(UART1));
    printf("\n\n%x\n",UART_READ(UART1));

    printf("\n\nUART Wakeup Compensation Sample Demo End.\n");
}
```

# 3 Software and Hardware Environment

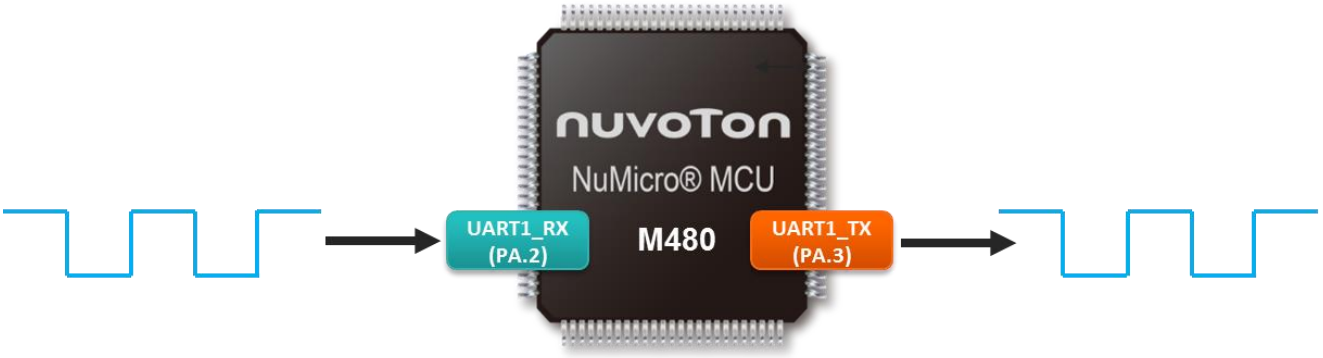
## Software environment

- BSP version
  - M480 Series BSP CMSIS V3.04.000
- IDE version
  - Keil uVersion 5.26

## Hardware environment

- Circuit components
  - NuMaker-IoT-M487 or other M480 Development Board
- Diagram

Input data from UART1_RX (PA.2) will wake up M480 when the data reaches the threshold.



# 4 Directory Information

- EC_M480_UART_Wakeup_Compensation_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 UART_Wakeup_Compensation.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 |
|---|---|---|
| Oct. 1, 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.*