使用PWM捕获功能侦测UART鲍率

NuMicro® 32 位系列微控制器范例代码介绍

文件信息

代码简述	使用 NUC123 的 PWMA 通道 0 (PA.12) 的捕获功能侦测 UART1 鲍率
BSP 版本	NUC123 Series BSP CMSIS v3.01.001
开发平台	NuTiny-SDK-NUC123AE V1.0 Nu-Link2-Me

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 功能介绍

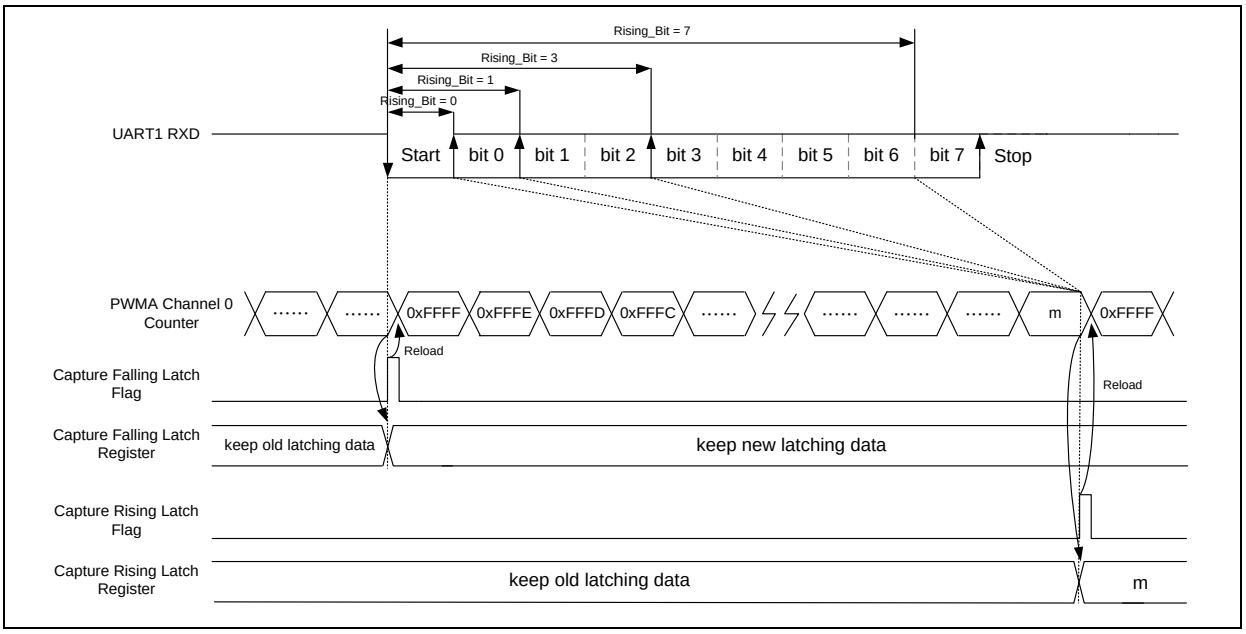
1.1 简介

使用PWMA通道0 (PA.12) 的捕获功能侦测UART1报率。PWM捕获功能可以在侦测到下降沿的时候，重置计数器的值，并且在侦测到上升沿的时候，锁存计数器的值。用户可以在RXD脚位传输特定的数据，利用上述方法计算低电平的时间，以侦测UART鲍率。

在使用这个范例程序前，用户需要将PWMA通道0 (PA.12) 接到 UART1的RXD (PB.4) 脚位。同时将Nu-Link2-Me的ICE_TX脚位和GND，分别接到NUC123的UART1的RXD脚位和GND，以传送测试数据。

1.2 原理

当接收到传输数据的Start位时，如果用户使能中断，PWM捕获功能将重置计数器的值。其值为CNR可设定的最大值0xFFFF。一旦PWM捕获功能侦测到上升沿的时候，PWM捕获功能将锁存当下计数器的值到缓存器。操作流程如下图所示：



用户可以传输不同的资料，使上升沿发生在不同的位，以变更低电平的时间。使用越长的低电平时间，可以得到越精准的结果。根据程序中定义的Rising_Bit，对应的传送数据如下表所示：

Rising_Bit	UART Data	Rising_Bit	UART Data
0	0xFF	4	0xF0
1	0xFE	5	0xE0

2	0xFC	6	0xC0
3	0xF8	7	0x80

UART鲍率的计算公式如下所示：

$$UART\,鲍率 = \frac{(Rising_Bit + 1)}{\left(0xFFFF - 上升沿锁存值\right) * (计数器的单位时间)}$$

计数器的单位时间计算公式如下所示：

$$计数器的单位时间 = \frac{PWM\,预除频\,(Prescaler)}{PWM\,时钟源频率}$$

1.3 执行结果

每当Nu-Link2-Me传送数据时，范例程序将侦测鲍率，并且使用计算结果设定UART1。结果如下图所示：

```
CPU @ 71884800 Hz
+------------------------------------------------------------+
|    Detect UART Baud Rate by PWM Capture Sample Code        |
|    Use PWMA Channel 0 to detect UART1 RXD                  |
|    Please connect PWMA Channel 0 (PA.12) to UART1 RXD (PB.4)    |
+------------------------------------------------------------+
Rising_Bit is 4.

PWM Capture counter value is 104.
Set UART1 baud rate to 192307.

PWM Capture counter value is 524.
Set UART1 baud rate to 38167.

PWM Capture counter value is 2092.
Set UART1 baud rate to 9560.

PWM Capture counter value is 16743.
Set UART1 baud rate to 1194.
```

Nu-Link2-Me依序使用UART鲍率194000、38400、9600以及1200，传送数据0xF0。计算误差分别为-0.873%、-0.607%、-0.417% 以及-0.500%。

2 代码介绍

初始化PWMA信道0捕获功能。设定计数器的单位时间为250 us。

```
void PWMA_Init(void)
{
    /* Set PWMA Channel 0 input clock divider */
    PWM_SET_DIVIDER(PWMA, 0, PWM_CLK_DIV_1);

    /* Set PWMA Channel 0 auto-reload mode */
    PWMA->PCR |= PWM_PCR_CH0MOD_Msk;

    /* Set PWMA Channel 0 prescaler value to make unit count time is 250 ns */
    /* PWMA clock input is HCLK = 72 MHz, prescaler value is HCLK*250/(10^9) - 1 = 17 */
    PWM_SET_PRESCALER(PWMA, 0, 17);

    /* Set PWMA Channel 0 CNR to max value */
    PWM_SET_CNR(PWMA, 0, 0xFFFF);

    /* Enable capture rising and falling edge interrupt for PWMA channel 0 */
    PWM_EnableCaptureInt(PWMA, PWM_CH0, PWM_CAPTURE_INT_RISING_LATCH |
PWM_CAPTURE_INT_FALLING_LATCH);

    /* Enable PWMA NVIC interrupt */
    NVIC_EnableIRQ(PWMA_IRQn);

    /* Enable Timer for PWMA channel 0  */
    PWM_Start(PWMA, BIT0);

    /* Enable Capture Function for PWMA channel 0 */
    PWM_EnableCapture(PWMA, BIT0);

    /* Clear detect flag */
    g_u8DetectFlag = 0;
}
```

在PWM中断处理中，在侦测到下降沿时，重置g_u32CaptureCounter的值，并且在侦测到上升沿时，取得计数器的值。

```
void PWMA_IRQHandler(void)
{
    uint32_t u32CaptureFlag = PWM_GetCaptureIntFlag(PWMA, 0);
```

```
    /* Detect rising edge */
    if (u32CaptureFlag == 1)
    {
        /* Clear rising edge flag */
        PWM_ClearCaptureIntFlag(PWMA, 0, PWM_CAPTURE_INT_RISING_LATCH);

        /* Detect rising bit, get counter value */
        g_u32CaptureCounter = 0xFFFF - PWM_GET_CAPTURE_RISING_DATA(PWMA, 0);

        /* Set detect flag */
        g_u8DetectFlag = 1;
    }
    /* Detect falling edge */
    else if (u32CaptureFlag == 2)
    {
        /* Clear falling edge flag */
        PWM_ClearCaptureIntFlag(PWMA, 0, PWM_CAPTURE_INT_FALLING_LATCH);

        /* Detect start bit, clear g_u32CaptureCounter to get counter value */
        g_u32CaptureCounter = 0;
    }
    /* Detect both rising and falling edge */
    else if (u32CaptureFlag == 3)
    {
        /* Clear rising edge flag */
        PWM_ClearCaptureIntFlag(PWMA, 0, PWM_CAPTURE_INT_RISING_LATCH);

        /* Clear falling edge flag */
        PWM_ClearCaptureIntFlag(PWMA, 0, PWM_CAPTURE_INT_FALLING_LATCH);
    }
}
```

计算UART1的新鲍率，并且设定到UART1。

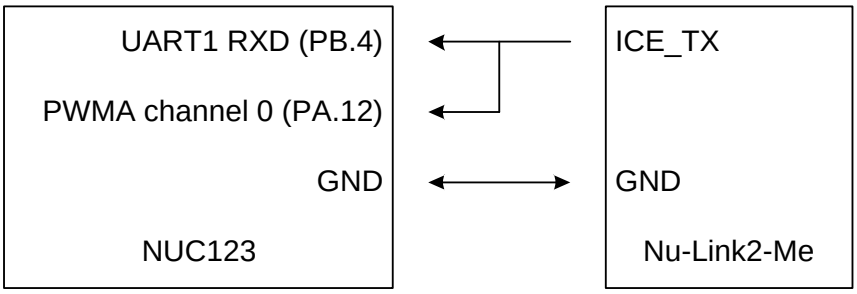
```
    while (1)
    {
        /* Wait until detection finish */
        while (!g_u8DetectFlag);

        /* Print PWM Capture result */
        printf("PWM Capture counter value is %d.\n", g_u32CaptureCounter);
```

```
        /* Set UART1 baud rate */
        u32Baudrate = (uint32_t)((Rising_Bit + 1) * 100000000) / (g_u32CaptureCounter *
25);
        printf("Set UART1 baud rate to %d.\n\n", u32Baudrate);

        /* Init UART1 */
        UART1_Init(u32Baudrate);

        /* Clear detect flag */
        g_u8DetectFlag = 0;
    }
```

3 软件与硬件环境

- 软件环境
  - BSP 版本
    - NUC230/240 Series BSP CMSIS V3.01.001
  - IDE 版本
    - Keil uVersion 5.26
- 硬件环境
  - 电路组件
    - NuTiny-SDK-NUC123AE V1.0
    - Nu-Link2-Me
  - 示意图

NUC123将UART1 RXD (PB.4) 脚位连接到Nu-Link2-Me的ICE_TX脚位，以接收数据。同时将NUC123的PWMA通道0 (PA.12) 连接到 UART1 RXD。



4 目录信息

- EC_NUC123_Detect_Baudrate_by_PWM_Capture_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 如何执行范例程序

1. 根据目录信息章节进入 ExampleCode 路径中的 KEIL 文件夹，双击 NUC123_Detect_Baudrate_by_PWM_Capture.uvproj。
2. 进入编译模式接口
   a. 编译
   b. 下载代码至内存
   c. 进入 / 离开除错模式
3. 进入除错模式接口
   a. 执行代码
4. 使用终端机软件，从 Nu-Link2-Me 传送数据给 NUC123 UART1 RXD。

6 修订纪录

Date	Revision	Description
July 1, 2019	1.00	1. 初始发布.

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.