M251 QSPI 搭配 PDMA 模拟 PWM 信号信

NuMicro® 32 位系列微控制器范例代码介绍

文件信息

应用简述	本范例代码使用 M251 的 QSPI 搭配 PDMA 输出 PWM 信号
BSP 版本	M251_Series_BSP_CMSIS_V3.02.007
开发平台	NuMaker-M251SD-V1.1

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 and microprocessor 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. 概述

本范例代码使用 M251 的 QSPI 搭配 PDMA 来输出 PWM 信号。

大部分芯片都会自带 PWM 接口，但如果需要多个 PWM 的输出，可能会面临 Channel 数量不够的问题。如果在应用中有不会用到的其他外设，便可以藉由该外设模拟 PWM 信号。本范例代码使用 QSPI 接口搭配 PDMA 输出 PWM 信号，PDMA 使用 Scatter-Gather 模式持续进行传输。

本范例代码不需另接外部硬件模块，若需要观察输出信号，可使用示波器、逻辑分析仪等仪器对输出波型进行测量。

1.1 原理

要使用 QSPI 模拟 PWM 信号需要先决定可以控制的最小单位，用以推算出 QSPI 传送数据的单笔数据大小与一次传输需要的数据数量。本范例是以 0.1% 占空比(duty)为最小可控制单位，可控制范围在 0.0%~100.0% 占空比。根据上述，QSPI 传送的每笔数据大小为 25 bits，每次传送 40 笔数据，总共用 1000 个 bits 模拟 1 次 PWM 信号，因此 PWM 信号频率为 QSPI 频率的千分之一。若 QSPI 频率为 1MHz，则 PWM 信号频率为 1kHz。

传送的数据储存在元素数量为 40 的数组中，数组中元素的值用占空比做计算。需要三个变量来决定数组内的元素:占空比的十倍数、占空比的十倍数除以 25 的商数、占空比的十倍数除以 25 的余数。因为定义数据类型为整数 unsigned int，不希望包含小数信息，因此记录占空比的十倍数而不是占空比实际数值。透过将占空比做除法运算，得出来的商数可以决定在数组中的哪个元素发生信号转折，余数可以决定是该元素在哪个 bit 发生信号转折。

在一个完整的 PWM 信号周期中，在发生信号转折的前信号皆为 1，发生信号转折后信号皆为 0。以数组来表示，在发生信号转折的元素前元素值皆为 1，发生信号转折的元素后元素值皆为 0。发生信号转折的元素中，在第(25-余数)~(25-余数+1) 个 bit 会从 1 转换成 0。根据此规则来建立 QSPI 要传送的数组。

举例说明，想模拟占空比为 63.5%的 PWM，需要在第 25 个元素的第 15~16 个 bit 发生信号转折(635/25=25…10)，用移位和逻辑符号决定数组中第 25 个元素的值为 0x1FFFC00。第 1~24 个元素数值皆为 0x1FFFFFF 模拟信号值为 1; 第 26~40 个元素数值皆为 0 模拟信号值为 0。

因为使用 1000 个 bit 去模拟一个完整 PWM 周期信号，因此 PWM 信号频率为 QSPI bit clock 频率的千分之一。

因为只需要用到模拟传送 PWM 信号的引脚，因此在 multi-pin function 中只需要定义 MOSI 脚位做输出，CLK、MISO、SS 脚位元皆不需要做定义使用。

QSPI 的传送有搭配 PDMA 以加快执行效率。PDMA 执行于 Scatter-gather mode，使用 DSCT 表对 PDMA 进行设定，DSCT 表的内容包括对 PDMA 的传输设定值、数据来源地址、数据目的地以及传送完后的下一个 DSCT 表进行设定。

1.2 执行结果

本范例代码执行编译和烧录的结果如图 1-1 所示。

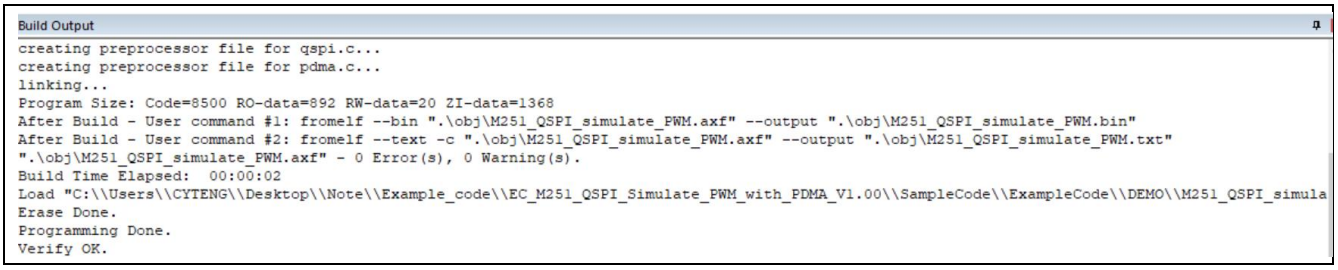

图 1-1 执行编译和烧录结果

使用逻辑分析仪量测输出 MOSI 信号脚位(PA0)，PWM 信号的占空比取决于传送数据的修改，图 1-2 和图 1-3 为占空比为 75%和 25%下的输出信号。

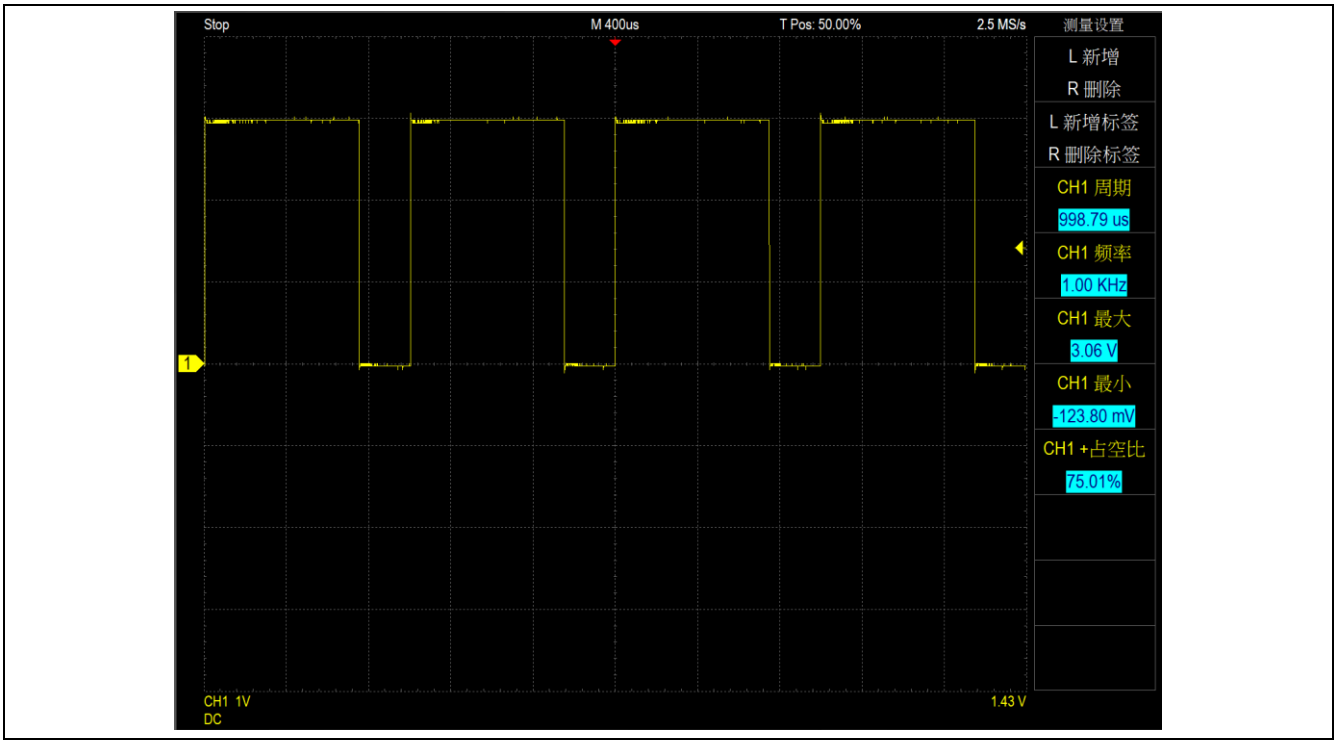

图 1-2 占空比为 75%下的输出信号

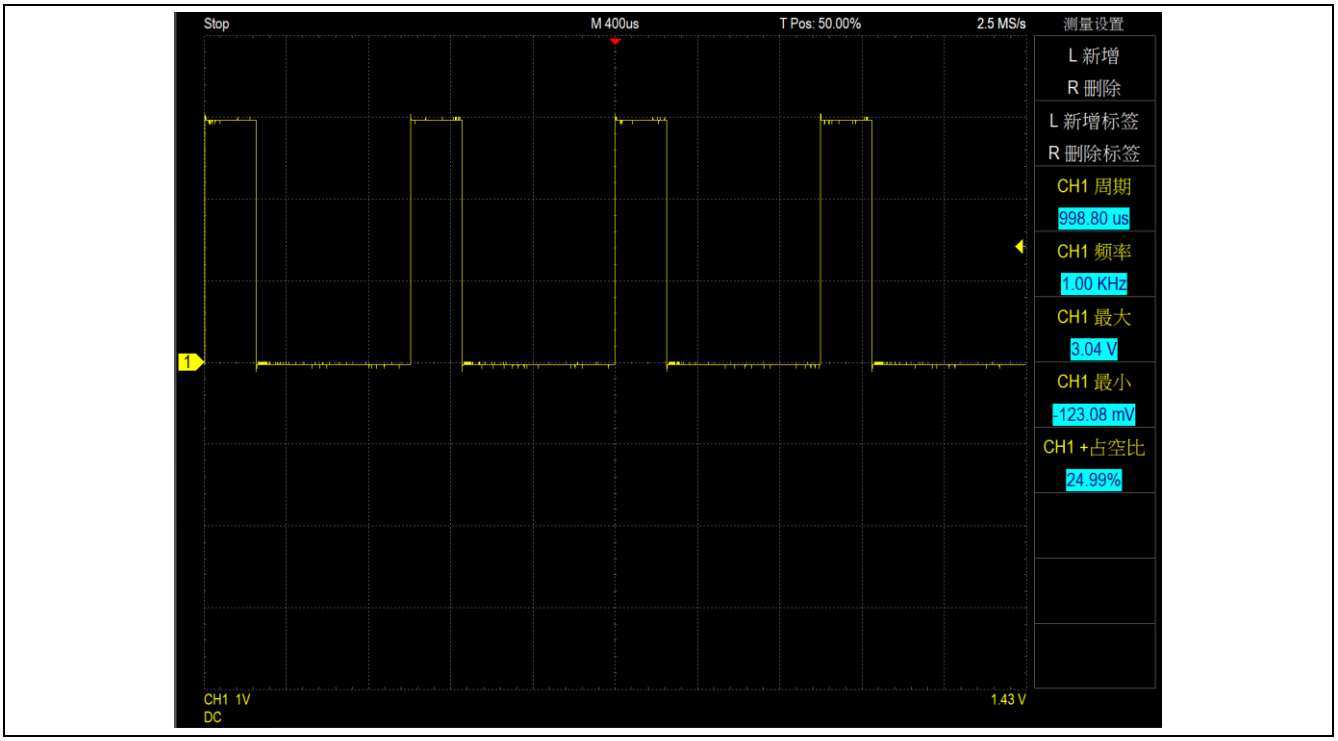

图 1-3 占空比为 25%下的输出信号

2. 代码介绍

初始化系统后，建立 QSPI 传送数据用到的数组，透过对变量 Duty_right_now 变量进行计算得到的商数与余数创建数组内容。代码位于 main.c。

```
int32_t main(void)
{
    /* Unlock protected registers */
    SYS_UnlockReg();

    /* Init System, IP clock and multi-function I/O */
    SYS_Init();

    /* Lock protected registers */
    SYS_LockReg();

    /* Init UART */
    UART0_Init();

    /* Calculate Duty array */
    Calculate_Duty_array(Duty_right_now);
```

函数 Calculate_Duty_array 内容，用于计算数组内容。

```
void Calculate_Duty_array(uint16_t Duty_Value)
{
    uint8_t Duty_remainder, Duty_Quotient;

    /* Calculate Duty parameter about quotient */
    Duty_Quotient = Duty_Value / 25;

    /* Calculate Duty parameter about remainder */
    Duty_remainder = Duty_Value % 25;

    /* Build transmission data array */
    for (uint8_t i = 0; i < 40; i++)
    {

        /*High signal before signal transition*/
        if (i <  Duty_Quotient)
        {
            Duty_array[i] = 0x1FFFFFF;
        }

        /*Signal transition located*/
        else if (i ==  Duty_Quotient)
        {
            Duty_array[i] = 0x1ffffff & (~(0x1ffffff+(1 << (25 - Duty_remainder))));
        }

        /*Low signal after signal transition*/
        else if (i >  Duty_Quotient)
        {
            Duty_array[i] = 0;
        }
    }

}
```

开启 PDMA 和 QSPI，并对 QSPI 数据传送间的间隔时间进行设定。

```
    /* Open Channel 1, 2 */
    PDMA_Open(PDMA, 1 | 2);

    /* Enable QSPI functions */
    /* Bit clock rate 1 MHz. */
    /* Master mode, 25-bit word width, mode 0. */
    QSPI_Open(QSPI0, SPI_MASTER, SPI_MODE_0, 25, 1000000);

    /* Set Suspend cycle */
    QSPI_SET_SUSPEND_CYCLE(QSPI0, 0);
```

对 PDMA 下 Scatter-gather mode 使用到的 DSCT 表进行设定，以结构体的形式对寄存器 PDMA_DSCTn_CTL、PDMA_DSCTn_SA、PDMA_DSCTn_DA、PDMA_DSCTn_NEXT 进行设定，并且开启 QSPI 与 PDMA。

```
    /* Define DSCT structure */
    DMA_DESC_T DMA_DESC[2];

    /*DSCT0 Description*/
    DMA_DESC[0].u32Ctl =
        ((40 - 1) << PDMA_DSCT_CTL_TXCNT_Pos) | /* Transfer count is 40 */ \
        PDMA_WIDTH_32 |     /* Transfer width is 32 bits(one word) */ \
        PDMA_SAR_INC |      /* Source increment size is 32 bits(one word) */ \
        PDMA_DAR_FIX |      /* Destination increment size is 32 bits(one word) */ \
        PDMA_REQ_SINGLE |   /* Transfer type is single transfer type */ \
        PDMA_BURST_128 |    /* No effect in single transfer type */ \
        PDMA_TBINTDIS_DISABLE |    /* Disable transfer done and table empty interrupt */ \
        PDMA_OP_SCATTER;    /* Operation mode is scatter-gather mode */

    /* Configure source address */
    DMA_DESC[0].u32Src = (uint32_t)Duty_array;

    /* Configure destination address */
    DMA_DESC[0].u32Dest = (uint32_t)&QSPI0->TX;

    /* Configure next descriptor table address */
    DMA_DESC[0].u32Offset = (uint32_t)&DMA_DESC[1] - (PDMA->SCATBA); /* next descriptor
    table is table 2 */

    DMA_DESC[1].u32Ctl =
        ((40 - 1) << PDMA_DSCT_CTL_TXCNT_Pos) | /* Transfer count is 40 */ \
        PDMA_WIDTH_32 |     /* Transfer width is 32 bits(one word) */ \
        PDMA_SAR_INC |      /* Source increment size is 32 bits(one word) */ \
        PDMA_DAR_FIX |      /* Destination increment size is 32 bits(one word) */ \
        PDMA_REQ_SINGLE |   /* Transfer type is single transfer type */ \
        PDMA_BURST_128 |    /* No effect in single transfer type */ \
        PDMA_TBINTDIS_DISABLE |    /* Disable transfer done and table empty interrupt */ \
        PDMA_OP_SCATTER;    /* Operation mode is scatter-gather mode */

    /* Configure source address */
    DMA_DESC[1].u32Src = (uint32_t)Duty_array;

    /* Configure destination address */
    DMA_DESC[1].u32Dest = (uint32_t)&QSPI0->TX;

    /* Configure next descriptor table address */
    DMA_DESC[1].u32Offset = (uint32_t)&DMA_DESC[0] - (PDMA->SCATBA); /* next descriptor
    table is table 1 */
```

设定 PDMA 的 Transfer mode，并开始 QSPI 搭配 PDMA 的传送。

```
    /* Set transfer mode*/
    PDMA_SetTransferMode(PDMA, 0, PDMA_QSPI0_TX, TRUE, (uint32_t)&DMA_DESC[0]);
    PDMA_SetTransferMode(PDMA, 1, PDMA_QSPI0_TX, TRUE, (uint32_t)&DMA_DESC[1]);

    /* Start QSPI with PDMA */
    QSPI_TRIGGER_TX_PDMA(QSPI0);
```

3. 软件与硬件需求

3.1 软件需求

- BSP 版本
  - M251_Series_BSP_CMSIS_V3.02.007
- IDE 版本
  - Keil uVision 5.39

3.2 硬件需求

- 电路组件
  - NuMaker-M251SD V1.1
- 线路示意图
  - 将 UART0 TX (PB.13) pin 脚连接到 PC UART RX，以显示范例代码的执行结果。若要测量 PWM 输出信号，可以将仪器接在 QSPI0 MOSI (PA.0) pin 脚上测量。

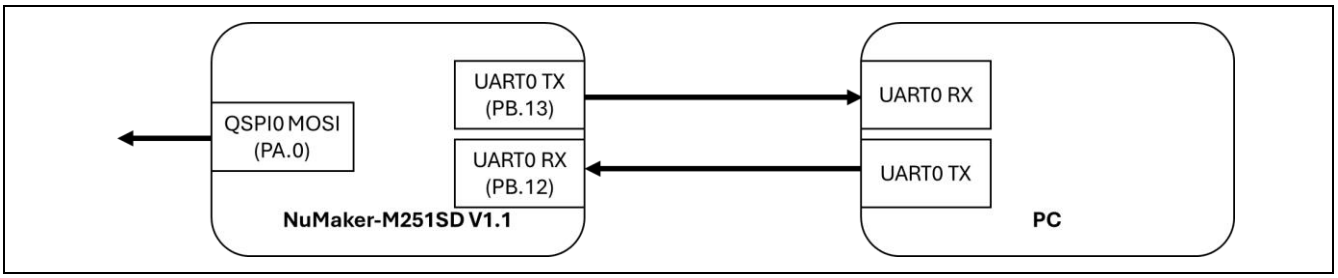

图 3-1 线路示意图

4. 目录信息

目录结构如下图所示。

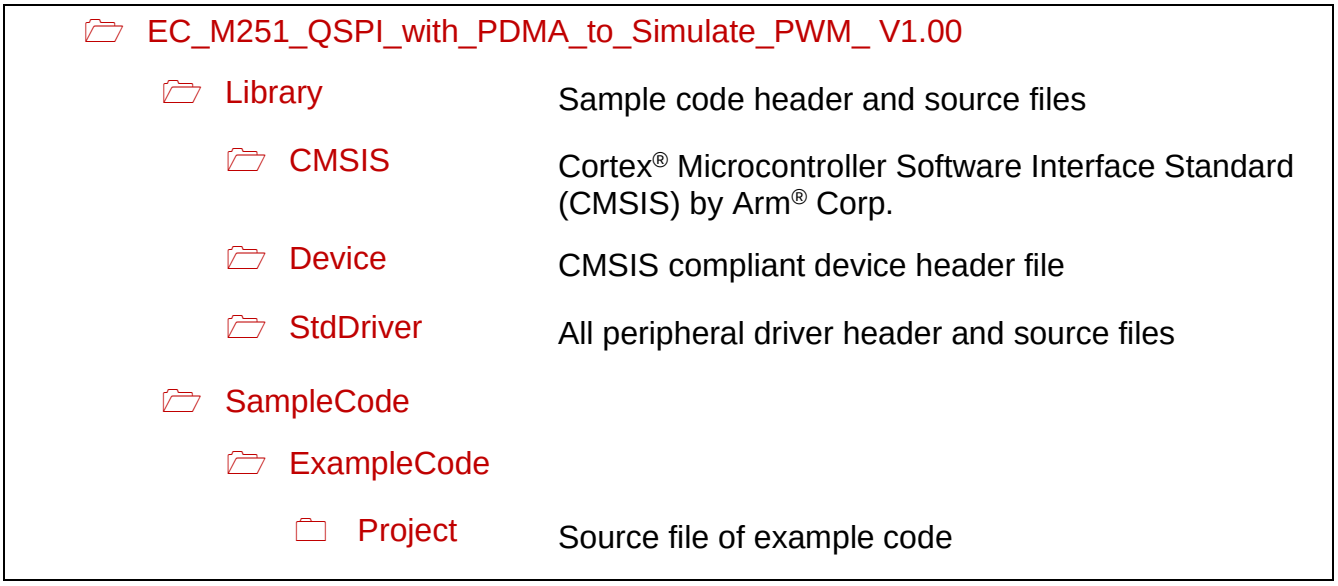

图 4-1 目录信息

5. 范例程式执行

1. 根据目录信息章节进入 ExampleCode 路径中的 KEIL 文件夹，双击 *M251_QSPI_with_PDMA_to_Simulate_PWM.uvprojx*。
2. 进入编译模式接口
   - 编译
   - 下载代码至内存
   - 进入 / 离开除错模式
3. 进入除错模式接口
   - 执行代码

6. 修订纪录

Date	Revision	Description
2025.01.15	1.00	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.