在 RTX 下利用 PDMA 存取 SPI Flash

NuMicro® 32 位系列微控制器範例代碼介紹

文件資訊

代碼簡述	示範如何在一個 RTX 系統下使用 PDMA 存取 SPI Flash
BSP 版本	NUC230/240 Series BSP CMSIS v3.01.002
開發平台	NuEdu-EVB-NUC240 V2.1 NuEdu-Basic01 V2.1

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 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 功能介紹

1.1 簡介

此範例程式將示範如何在 NuMicro® NUC240 系列 MCU 上運行 RTX 系統，並透過 PDMA 來存取 SPI Flash。

1.2 原理

Keil RTX 是一套專為 ARM 和 Cortex-M 硬體裝置設計的免付費版本的實時操作系統。這套系統允許開發者可創建同時執行多個功能的程序，並幫助創建結構更好且更易於維護的應用程序。Keil RTX 的詳細功能和優點可參考官網，http://www.keil.com/arm/rl-arm/kernel.asp。

此示例代碼實現了使用 NuMicro® NUC240 MCU 在 NuEdu-EVB-NUC240 V2.1 電路板上移植 Keil RTX，並創建了一個 RTX 任務，以便在 NuEdu-Basic01 V2.1 電路板上使用 PDMA 讀取外部 SPI Flash。
NuEdu-Basic01 V2.1 板的相關驅動程序位於 BSP\Library\NuEdu 資料夾內。

下列 圖 1-1 展示了此示例代碼的操作流程。

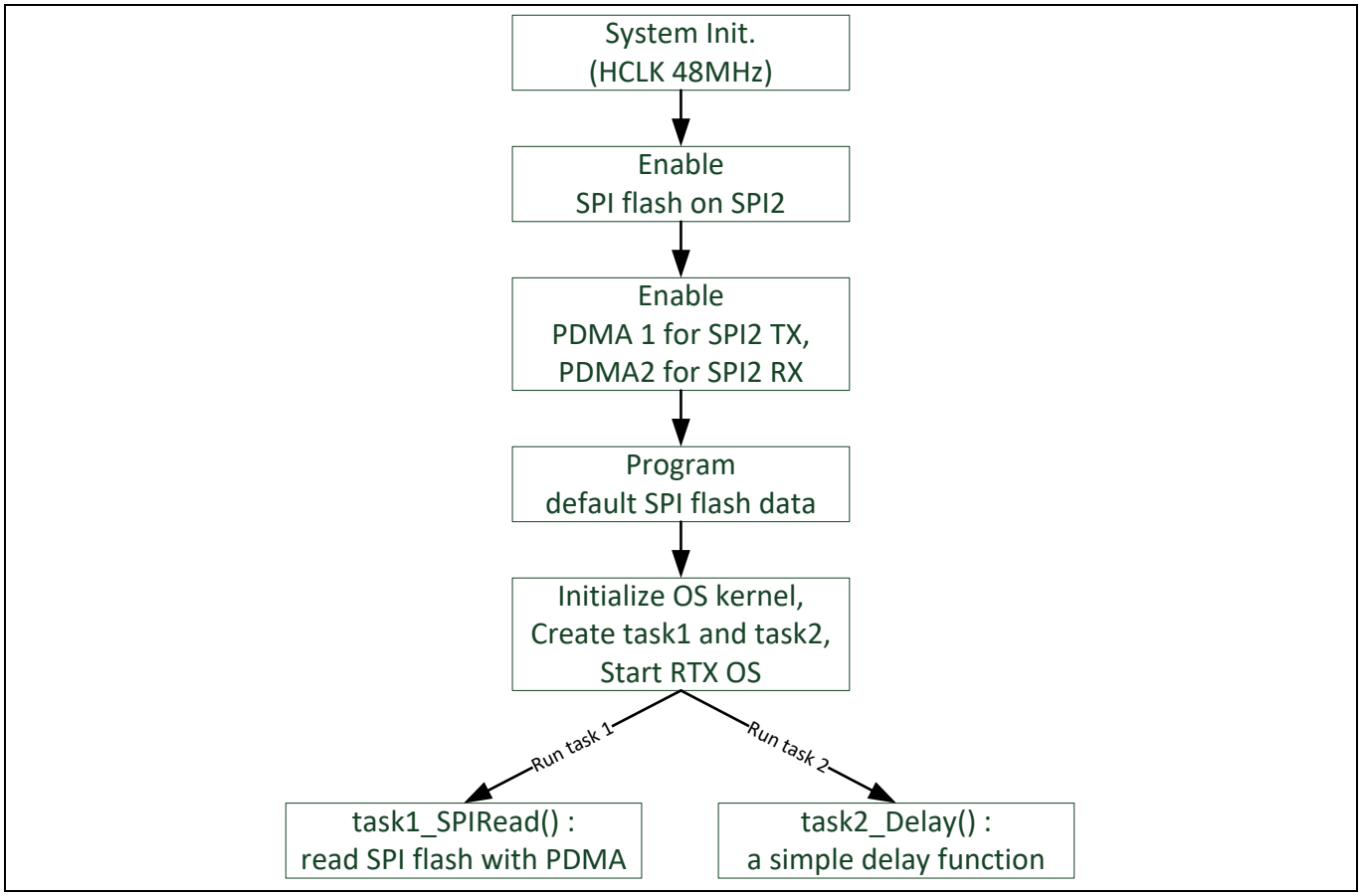

圖 1-1 操作流程圖

1.3 執行結果

程序執行結果可經由串口鮑率 115200 的輸出到 PC 做顯示，如下圖 1-2 所示。

```
CPU @ 48000000 Hz
+-----------------------------------------------------+
|    Initialize RTX to access SPI flash with PDMA     |
+-----------------------------------------------------+

SPI Flash MID and DID = 0xef14
Flash Erasing... Done!

Flash Programming... Done!

task1: Read SPI flash was PASS.
task2: A delay task.
task2: A delay task.
task1: Read SPI flash was PASS.
task2: A delay task.
task2: A delay task.
task1: Read SPI flash was PASS.
task2: A delay task.
task2: A delay task.
```

圖 1-2 串口執行結果

2 代碼介紹

在 main.c 內和 RTC 系統有關的變數和函式說明。

```
osThreadId  t_task1; /* thread ID of task 1 */
osThreadId  t_task2; /* thread ID of task 2 */

void task1_SPIRead(void const *argument); /* thread function of task 1 */
void task2_Delay(void const *argument);   /* thread function of task 2 */

/* Thread Definition with function, priority, and stack requirements */
osThreadDef(task1_SPIRead, osPriorityNormal, 1, 0);
osThreadDef(task2_Delay, osPriorityNormal, 1, 0);

/*----------------------------------------------------------------------------
-----------------*/
/* task 1 function: read SPI flash with PDMA
*/
/*----------------------------------------------------------------------------
-----------------*/
void task1_SPIRead(void const *argument)
{
    ……
}

/*----------------------------------------------------------------------------
-----------------*/
/* task 2 function: a simple delay function
*/
/*----------------------------------------------------------------------------
-----------------*/
void task2_Delay(void const *argument)
{
    ……
}
```

在 main.c 內初始化 PDMA 和 SPI 功能，並寫入 SPI Flash 的默認數據。

大部分在函式 InitSPIFlashDataAndPDMA( ) 內被調用的 API 是直接引用 NuEdu 驅動程序庫內的 NuEdu-Basic01_SPI_Flash_w_PDMA.c。

```
void InitSPIFlashDataAndPDMA(PDMA_T *pdma)
```

```
{
    unsigned int u32ByteCount;
    unsigned int u32MidDid;

    /* Initial PDMA Channels */
    Init_PDMA_CH1_for_SPI2_TX((uint32_t)gau32SrcArray);
    Init_PDMA_CH2_for_SPI2_RX((uint32_t)gau32DestArray);

    /* Enable PDMA IRQ */
    NVIC_EnableIRQ(PDMA_IRQn);

    /* Read MID & DID */
    u32MidDid = SpiFlash_w_PDMA_ReadMidDid();
    printf("\nSPI Flash MID and DID = 0x%x", u32MidDid);

    /* Erase SPI Flash */
    SpiFlash_w_PDMA_ChipErase();
    printf("\nFlash Erasing... ");

    /* Wait ready */
    SpiFlash_w_PDMA_WaitReady();
    printf("Done!");

    /* Fill the Source Data and clear Destination Data Buffer */
    for (u32ByteCount = 0; u32ByteCount < 256; u32ByteCount++)
        gau32SrcArray[u32ByteCount] = ((u32ByteCount+0x10)&0xFF);

    /*=== Program SPI Flash ===*/
    printf("\n\nFlash Programming... ");

    /* Trigger PDMA specified Channel */
    pdma->CSR |= (PDMA_CSR_TRIG_EN_Msk | PDMA_CSR_PDMACEN_Msk);

    /* Page Program */
    SpiFlash_w_PDMA_PageProgram(0x0, 256);
    SpiFlash_w_PDMA_WaitReady();
    printf("Done!\n\n");
}
```

初始化 RTOS，創建任務並啟動 RTOS 以運行 main.c 中的所有任務。

```
    /* Initialize RTOS */
    osKernelInitialize();


    /* Create task1 to read SPI flash with PDMA channel 2, and task 2 as a simple delay
loop */
    t_task1 = osThreadCreate(osThread(task1_SPIRead), PDMA_CH2);
    t_task2 = osThreadCreate(osThread(task2_Delay), NULL);

    /* Start the RTOS */
    osKernelStart();
```

3 軟體與硬體環境

- **軟體環境**
  - BSP 版本
    - NUC230/240 Series BSP CMSIS v3.01.002
  - IDE 版本
    - Keil uVersion 5.28
- **硬體環境**
  - 電路元件
    - NuEdu-EVB-NUC240 V2.1
    - NuEdu-Basic01 V2.1
  - 示意圖
    - 將 NuEdu-Basic01 V2.1 板上的 NUC240 UART TX (PB.1) 引腳連接到 PC UART RX，以在 PC 上顯示示例代碼的執行結果。

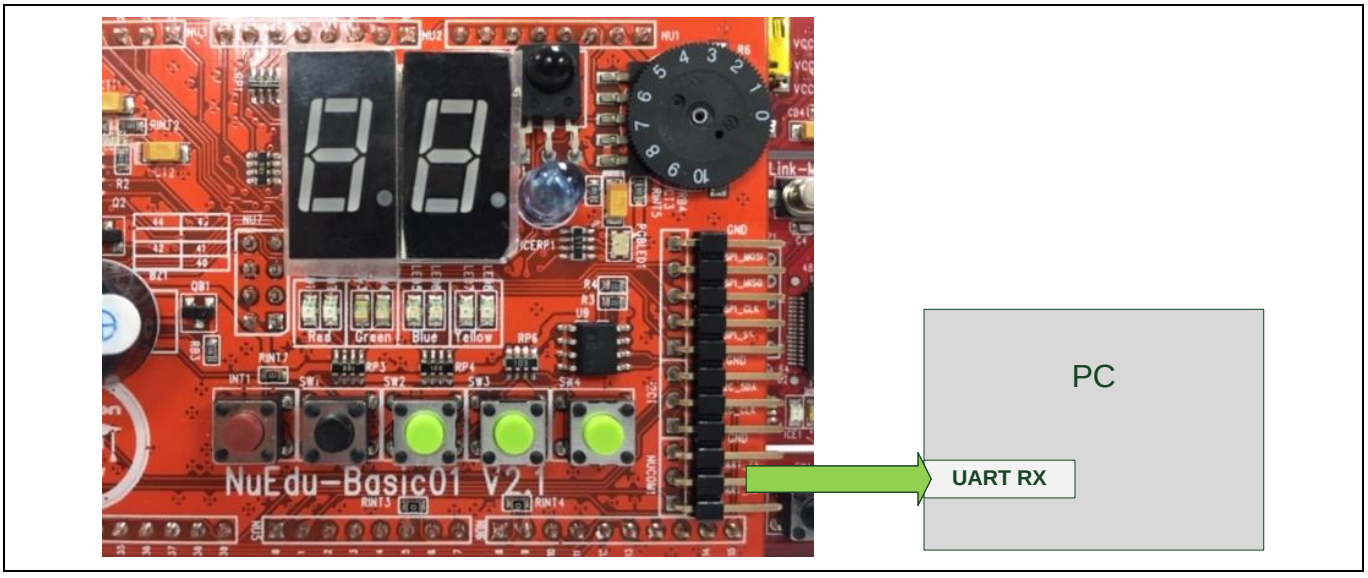

圖 3-1 NUC240 UART TX 和 PC UART RX 連接

◆ 將 NuEdu-EVB-NUC240 V2.1 板上的 NUC240 SPI2 引腳連接到 NuEdu-Basic01 V2.1 板上的外部 SPI Flash。

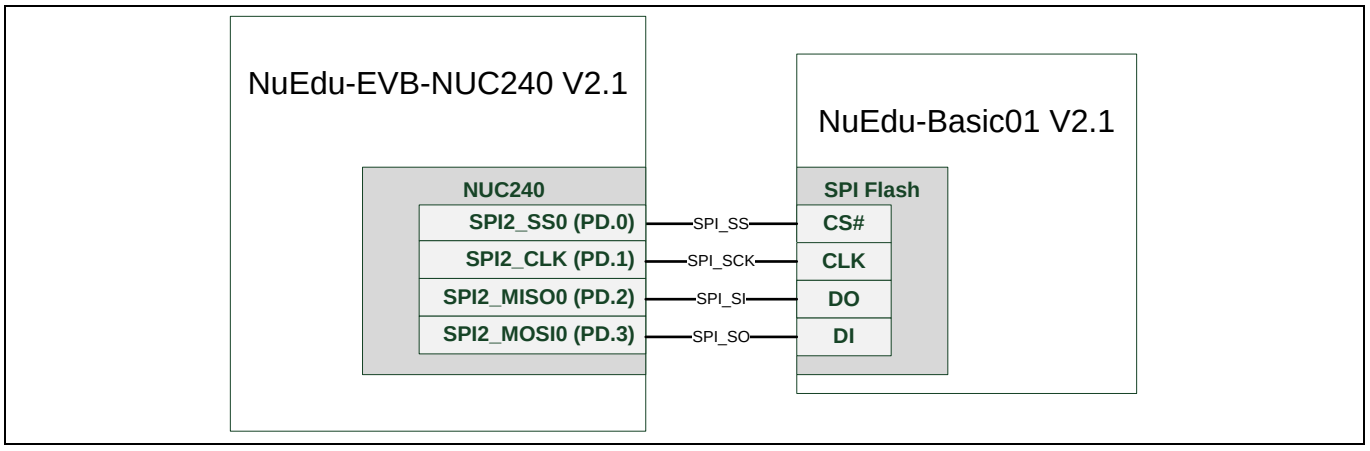

圖 3-2 NUC240 SPI2 和 外部 SPI Flash 連接

4 目錄資訊

- EC_NUC240_RTX_SPI_Flash_wPDMA_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
    - NuEdu — Library for NuEdu-SDK-xxxx board
    - StdDriver — All peripheral driver header and source files
  - SampleCode
    - ExampleCode — Source file of example code

5 如何執行範例程式

1. 根據目錄資訊章節進入 ExampleCode 路徑中的 KEIL 資料夾，雙擊 RTX_SPI_Flash_wPDMA.uvprojx。
2. 進入編譯模式介面
   a. 編譯
   b. 下載代碼至記憶體
   c. 進入 / 離開除錯模式
3. 進入除錯模式介面
   a. 執行代碼

6 修訂紀錄

Date	Revision	Description
Aug 21, 2019	1.00	1. 初始發布。

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.