使用 Data Flash 模擬 EEPROM

NuMicro® 32 位系列微控制器範例代碼介紹

文件資訊

代碼簡述	用一塊 Data Flash 模擬 EEPROM，並做 EEPROM 寫/讀
BSP 版本	NUC230/240 Series BSP CMSIS v3.01.002
開發平台	NuTiny–EVB–NUC240 V1.2

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用一塊 Data Flash 來模擬 EEPROM 做讀/寫。

1.2 原理

在 NuMicro® NUC240 系列 MCU 內配備了嵌入式閃存，以運行程序和儲存數據。
寫/讀閃存操作的限制是訪問地址必須是 32 位對齊地址，數據資料值也是 32 位數據。 每個位元的數據值可以從 1 寫成 0，並在執行擦除閃存操作完成後會變為 1。 NuMicro® NUC240系列僅支持頁擦除閃存的操作，一頁大小為 512 字節。

在寫/讀 EEPROM 時，寫/讀操作支持字節地址對齊，數據長度為一個字節，每個位數據是可以直接寫入 1 或 0。

在此示例代碼中，將預先分配指定一塊 Data Flash 區域來模擬為 EEPROM 存儲空間。函式 SIM_EEPROM_READ 用於從指定的 EEPROM 地址讀取一筆 8 位資料，函式 SIM_EEPROM_WRITE 用於對指定的 EEPROM 地址寫入一筆 8 位資料。
對 EEPROM 做寫入的範例流程，如下 圖 1-1 所示。

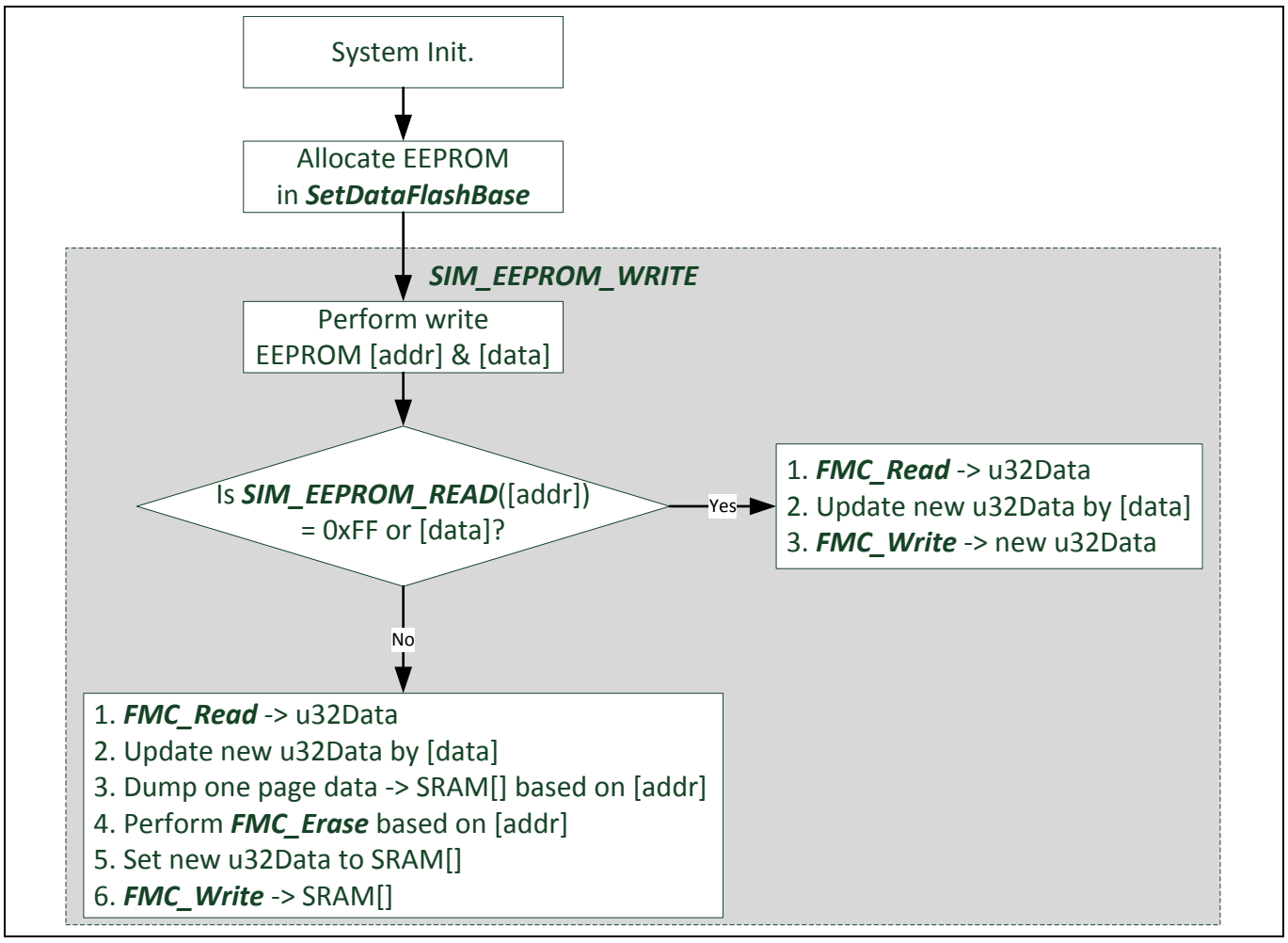

圖 1-1 寫入 EEPROM 的流程

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 @ 72000000 Hz

[Simulate EEPROM base address at Data Flash 0xf000]

W EEPROM 0000: 0x0 PASS.
W EEPROM 0000: 0xff PASS.
W EEPROM 0000: 0x55 PASS.

W EEPROM 0001: 0x0 PASS.
W EEPROM 0001: 0xff PASS.
W EEPROM 0001: 0xaa PASS.

W EEPROM 0512: 0x0 PASS.
W EEPROM 0512: 0xff PASS.
W EEPROM 0512: 0x5a PASS.

W EEPROM 0513: 0x0 PASS.
W EEPROM 0513: 0xff PASS.
W EEPROM 0513: 0xa5 PASS.
```

圖 1-2 串口執行結果

2 代碼介紹

分配一塊 Data Flash 區域來模擬 EEPROM 存儲空間。

```
int32_t SetDataFlashBase(uint32_t u32DFBA)
{
    uint32_t au32Config[2];

    /* Read current User Configuration */
    FMC_ReadConfig(au32Config, 1);

    /* Just return when Data Flash has been enabled */
    if(!(au32Config[0] & 0x1))
        return 0;

    /* Enable User Configuration Update */
    FMC_EnableConfigUpdate();

    /* Erase User Configuration */
    FMC_Erase(FMC_CONFIG_BASE);

    /* Write User Configuration to Enable Data Flash */
    au32Config[0] &= ~0x1;
    au32Config[1] = u32DFBA;
    if(FMC_WriteConfig(au32Config, 2))
        return -1;

    /* Perform chip reset to make new User Config take effect */
    SYS->IPRSTC1 |= SYS_IPRSTC1_CHIP_RST_Msk;

    return 0;
}
```

讀 EEPROM 的函式。

```
uint8_t SIM_EEPROM_READ(uint32_t address)
{
    uint32_t u32Data;

    u32Data = FMC_Read(((address/4)*4) + EEPROM_BASE);
```

```
    return ((u32Data>>((address%4)*8)) & 0xFF);
}
```

寫 EEPROM 的函式。

```
void SIM_EEPROM_WRITE(uint32_t address, uint8_t data)
{
    ……
    if((SIM_EEPROM_READ(address)==0xFF) || (SIM_EEPROM_READ(address)==data))
    {
        /* Original flash data is 0xFF, or data is matched. Do not perform flash page
erase. */

        /* Read oroginal data */
        u32Data = FMC_Read(((address/4)*4) + EEPROM_BASE);
        /* Assign 32-bit updated data */
        u32Data = ~(0xFF<<((address%4)*8));
        u32Data |= (data<<((address%4)*8));
        /* Write updated data */
        FMC_Write((((address/4)*4) + EEPROM_BASE), u32Data);
    }
    else
    {
        /* Need to update one page flash data. Perform flash page erase is mandatory
before writing updated data. */

        /* Assign 32-bit updated data */
        u32Data = FMC_Read(((address/4)*4) + EEPROM_BASE);
        u32Data &= ~(0xFF<<((address%4)*8));
        u32Data |= (data<<((address%4)*8));

        /* Dump target page data to SRAM buffer */
        u32Start = (EEPROM_BASE + ((address/FMC_FLASH_PAGE_SIZE)*FMC_FLASH_PAGE_SIZE)); //
page-size alignment
        u32End   = (u32Start + FMC_FLASH_PAGE_SIZE);
        pu32DataBuf = (uint32_t *)au32DataBuf;
        for(i=u32Start; i<u32End; i+=4)
        {
            *pu32DataBuf++ = FMC_Read(i);
        }

        /* Erase target page */
```

```
        FMC_Erase(EEPROM_BASE+((address/FMC_FLASH_PAGE_SIZE)*FMC_FLASH_PAGE_SIZE));

        /* Set updated data to SRAM buffer */
        au32DataBuf[((address%FMC_FLASH_PAGE_SIZE)/4)] = u32Data;

        /* Write target flash data from SRAM buffer */
        pu32DataBuf = (uint32_t *)au32DataBuf;
        for(i=u32Start; i<u32End; i+=4)
        {
            FMC_Write(i, *pu32DataBuf++);
        }
    }
}
```

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
  - ◆ NuTiny–EVB–NUC240 V1.2
- ■ 示意圖
  - ◆ 將 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。

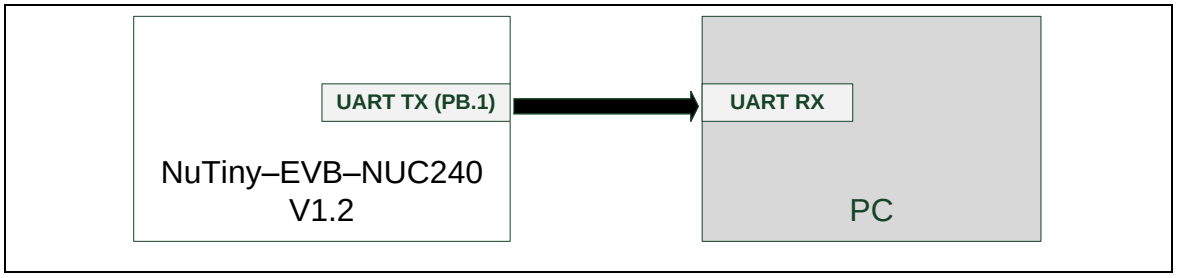

圖 3-1 NUC240 UART TX 和 PC UART RX 連接

4 目錄資訊

- EC_NUC240_ Simulate_EEPROM_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 Simulate_EEPROM.uvproj。
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
Jul 30, 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.