M480 ISP韌體更新與修復

NuMicro® 32位系列微控制器範例代碼介紹

文件資訊

應用簡述	本範例程式為 LDROM 之應用，將 APROM 分割為 2 塊，在上電後對執行塊與備份塊進行比較與更新，以實現韌體更新與修復功能。
BSP 版本	M480 Series BSP CMSIS V3.05.003
開發平台	NuMaker-PFM-M487 V3.0

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 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

本範例為 LDROM 之應用，支援了以 UART 更新 APROM 並透過 Bank 拆分的方式對其進行韌體校驗與修復。 由於 M480 系列 IC 沒有切換 Bank 映射的功能，因此本範例將 APROM 的 2 個 bank 區分為 Bank0 的執行塊與 Bank1 的備份塊如圖 1-1 所示，當使用 ICE 或 UART 進行韌體更新時，將程式更新至 APROM_SIZE/2 的位址，待重新上電後，本範例將對 Bank0 與 Bank1 逐頁進行比較，若出現差異，則更新相對應的頁，以達到修復應用程式的目的。在一般情況下重新上電，也會對 Bank0 與 Bank1 進行比較，修正完錯誤的頁後才執行應用。

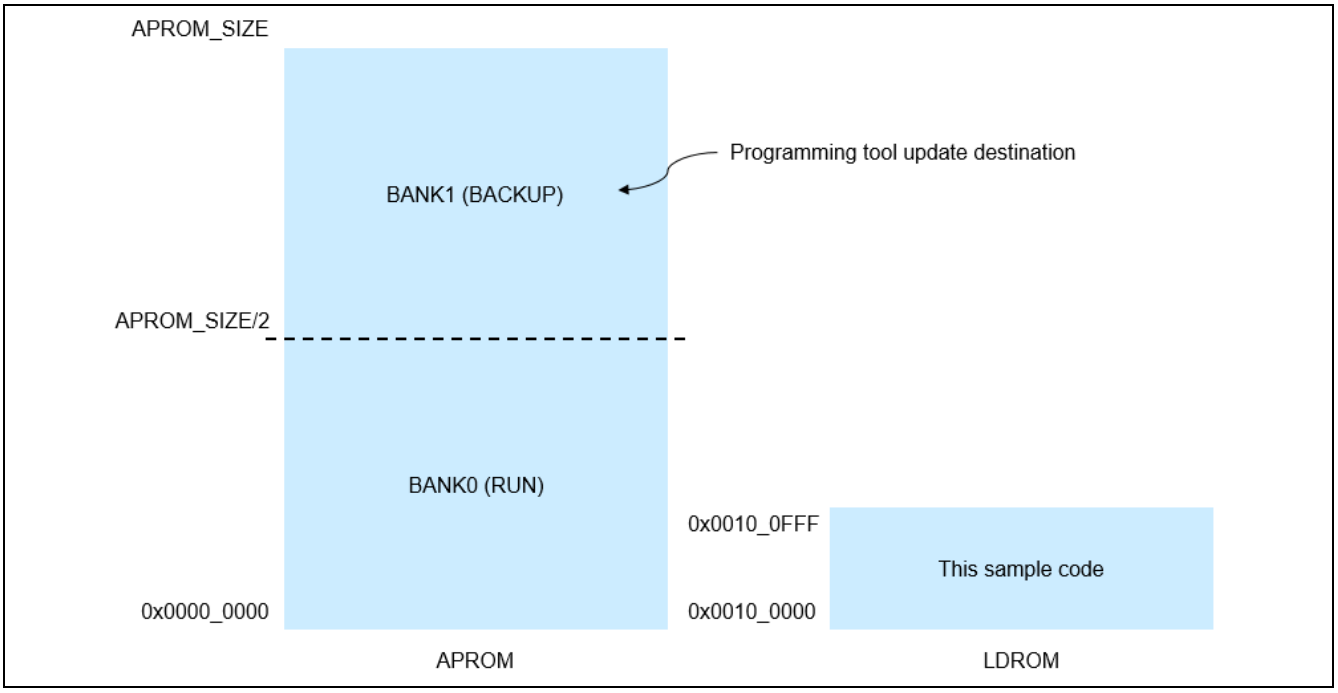

圖 1-1 快閃記憶體分配

1.1 軟體架構

軟體流程如圖 1-2 所示， 啟動模式選擇由 LDROM 啟動，此時位於 LDROM 的程式會在等待 UART 命令的同時對 Bank0 與 Bank1 做逐頁比對並記錄結果，比較的長度為 APROM_SIZE 的一半，當比對完成且接收 UART 命令超時，則會根據比對結果區分為: 1. 比對內容一致，跳轉至 APROM 執行應用，2. 內容不一致，針對 Bank0 相對應的頁進行擦除與燒錄，待執行完畢，再跳轉至 APROM 執行應用。

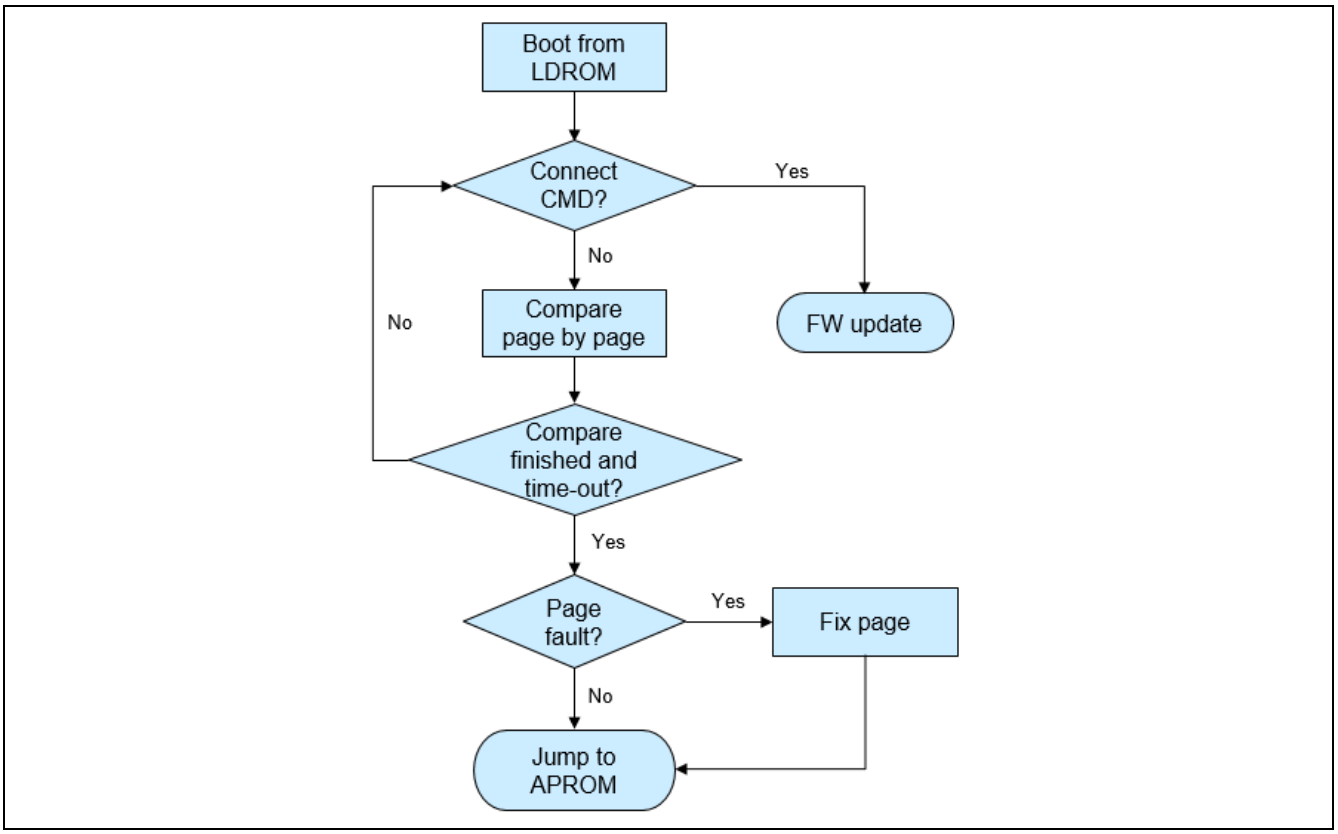

圖1-2 軟體流程圖

1.2 功能介紹

1.2.1 啟動模式設定

- 啟動模式：

  M480 系列 MCU 支援了多種啟動模式，包含 LDROM with IAP，LDROM without IAP，APROM with IAP，APROM without IAP，其中設定啟動模式的位址為 0x0030_0000 的 CONFIG0[7:6] CBS，本範例建立於 LDROM，使用 CBS 為 01 的 LDROM 啟動，使韌體更新與 APROM 的應用彼此獨立。
  當使用 LDROM 開機時，會將 LDROM 映射至起始位址為 0 處，此時若想存取 APROM 的位址則需使用 FMC 模塊。

1.2.2 FMC 操作

- 擦除：

  M480 系列 IC 擦除的最小單位為頁，操作時將 FMC_ISPCMD 填入擦除命令 0x22，FMC_ISPADDR 填入每頁的實體起始位址，再將 FMC_ISPTRG 寫 1 完成一次擦除流程。

- 讀：

  讀取每次以 32 位元為單位，操作時將 FMC_ISPCMD 寫入讀命令 0x0，FMC_ISPADDR 填入欲讀取的實體位址，再將 FMC_ISPTRG 進行讀取，讀取的資料可從 FMC_ISPDAT 取得。

- 寫：

  寫的最小單位為 32 位元，操作時將 FMC_ISPCMD 填入寫命令 0x21，FMC_ISPDAT 填入欲寫入的資料，FMC_ISPADDR 填入目標實體位址，再將 FMC_ISPTRG 寫 1 完成寫入流程。

1.2.3 執行說明

1.2.3.1 前置作業

第一次燒錄需連接 ICE 並搭配 Nuvoton NuMicro ICP Programming Tool 將此範例程式燒錄至 LDROM，並將 CONFIG0[7:6] CBS 寫為 0x1，使啟動模式改為由 LDROM 啟動，應用程式的燒錄以及後續的更新亦可使用 Nuvoton NuMicro ISP Programming tool 透過 UART 進行韌體更新，燒錄前須注意應用程式的大小需小於 APROM 大小的一半。

1.2.3.2 執行結果

實驗以大小為 512KB 之 APROM 的韌體更新為例：

韌體更新前，程式#1 在位址 0x200 的資料如圖 1-3。

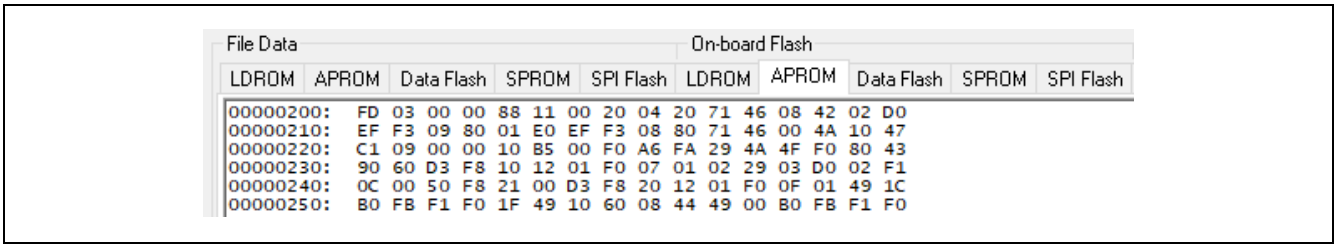

圖 1-3 更新前執行區塊

將程式#2 更新至起始位址為 APROM 大小的一半，位址 0x40200 的資料如圖 1-4 。

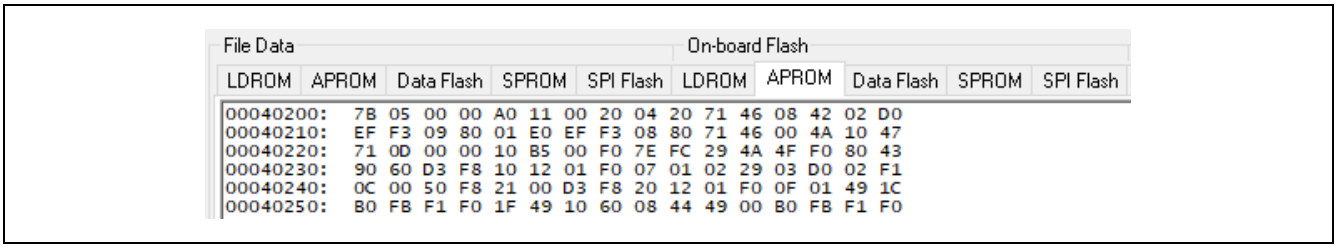

圖 1-4 更新後備份區塊

重新上電後，位於 LDROM 的範例程式對 Bank0 與 Bank1 進行比較與更新，可得在位址 0x200 與 0x40200 資料相同如圖 1-5。

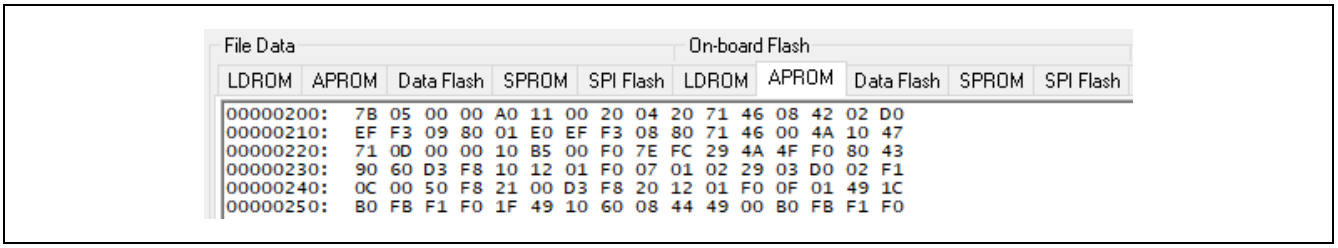
File Data
On-board Flash
LDROM APROM Data Flash SPROM SPI Flash LDROM APROM Data Flash SPROM SPI Flash
00000200: 7B 05 00 00 A0 11 00 20 04 20 71 46 08 42 02 D0
00000210: EF F3 09 80 01 E0 EF F3 08 80 71 46 00 4A 10 47
00000220: 71 0D 00 00 10 B5 00 F0 7E FC 29 4A 4F F0 80 43
00000230: 90 60 D3 F8 10 12 01 F0 07 01 02 29 03 D0 02 F1
00000240: 0C 00 50 F8 21 00 D3 F8 20 12 01 F0 0F 01 49 1C
00000250: B0 FB F1 F0 1F 49 10 60 08 44 49 00 B0 FB F1 F0

圖 1-5 更新後執行區塊

2. 程式介紹

以下程式實現了章節 1.1 的架構，包含 1.2.2 所述的 FMC 使用流程。

```
/* Word compare between source address and destination address */
int flashcmp(uint32_t desAddr, uint32_t srcAddr)
{
    uint32_t srcData, desData;
    FMC->ISPCMD = FMC_ISPCMD_READ;
    FMC->ISPADDR = srcAddr;
    FMC->ISPTRG = 0x1;
    __ISB();
    while (FMC->ISPTRG & 0x1) ;
    srcData = FMC->ISPDAT;

    FMC->ISPCMD = FMC_ISPCMD_READ;
    FMC->ISPADDR = desAddr;
    FMC->ISPTRG = 0x1;
    __ISB();
    while (FMC->ISPTRG & 0x1) ;
    desData = FMC->ISPDAT;

    return srcData != desData;
}

/* Word copy from source address to destination address */
void flashcpy(uint32_t desAddr, uint32_t srcAddr)
{
    uint32_t srcData;
    FMC->ISPCMD = FMC_ISPCMD_READ;
    FMC->ISPADDR = srcAddr;
    FMC->ISPTRG = 0x1;
    __ISB();
    while (FMC->ISPTRG & 0x1) ;
    srcData = FMC->ISPDAT;

    FMC->ISPCMD = FMC_ISPCMD_PROGRAM;
    FMC->ISPADDR = desAddr;
    FMC->ISPDAT = srcData;
    FMC->ISPTRG = 0x1;
    __ISB();
    while (FMC->ISPTRG & 0x1) ;
}

int main(void)
{
    uint16_t cp[4] = {0,0,0,0};
    uint32_t i = 0, j, checked = 0;
    uint32_t bsize, pnum;

    /* Unlock protected registers */
    SYS_UnlockReg();
    /* Init System, peripheral clock and multi-function I/O */
    SYS_Init();
```

```
    /* Init UART to 115200-8n1 */
    UART_Init();
    CLK->AHBCLK |= CLK_AHBCLK_ISPCKEN_Msk;
    FMC->ISPCTL |= FMC_ISPCTL_ISPEN_Msk;
    g_apromSize = GetApromSize();
    bsize = g_apromSize >> 1;
    pnum = bsize >> 12;
    GetDataFlashInfo(&g_dataFlashAddr, &g_dataFlashSize);
    SysTick->LOAD = 300000 * CyclesPerUs;
    SysTick->VAL   = (0x00);
    //using cpu clock
    SysTick->CTRL = SysTick->CTRL | SysTick_CTRL_CLKSOURCE_Msk | SysTick_CTRL_ENABLE_Msk;

    while (1)
    {
    /* Command catch */
        if ((bufhead >= 4) || (bUartDataReady == TRUE))
        {
            uint32_t lcmd;
            lcmd = inpw(uart_rcvbuf);

            if (lcmd == CMD_CONNECT)
            {
                goto _ISP;
            }
            else
            {
                bUartDataReady = FALSE;
                bufhead = 0;
            }
        }

    /* Command timeout and flash compare finished */
        if ((SysTick->CTRL & SysTick_CTRL_COUNTFLAG_Msk) && checked)
        {
         // Fix page
         for(i = 0; i < pnum; i++)
         {
             if(cp[i/16] & (1 << (i%16)))
             {
                 FMC->ISPCMD = FMC_ISPCMD_PAGE_ERASE;
                 FMC->ISPADDR = FMC_APROM_BASE + PAGE_SIZE * i;
                 FMC->ISPTRG = 0x1;
                 __ISB();
                 while (FMC->ISPTRG & 0x1) ;

                 for(j = 0; j < PAGE_SIZE; j += 4)
                 {
                     flashcpy(PAGE_SIZE * i + j, PAGE_SIZE * i + j + bsize);
                 }
             }
         }
            goto _APROM;
        }

    /* Compare flash page by page */
    if(!checked)
```

```
    {
        for(j = 0; j < PAGE_SIZE; j+=4)
        {
            if(flashcmp(i * PAGE_SIZE + j, i * PAGE_SIZE + j + bsize))
            {
                cp[i/16] |= 1 << (i%16);
                break;
            }
        }

        if(++i == pnum)
            checked = 1;
    }
    }

_ISP:

    while (1)
    {
        if (bUartDataReady == TRUE)
        {
            bUartDataReady = FALSE;
            ParseCmd(uart_rcvbuf, 64);
            PutString();
        }
    }

_APROM:
    SYS->RSTSTS = (SYS_RSTSTS_PORF_Msk | SYS_RSTSTS_PINRF_Msk);
    FMC->ISPCTL &= ~(FMC_ISPCTL_ISPEN_Msk | FMC_ISPCTL_BS_Msk);
    SCB->AIRCR = (V6M_AIRCR_VECTKEY_DATA | V6M_AIRCR_SYSRESETREQ);

    /* Trap the CPU */
    while (1);
}
```

3. 軟硬體需求

3.1 軟體需求

- BSP 版本
  - M480 Series BSP CMSIS V3.05.003
- IDE 版本
  - Keil uVersion 5.28
- 燒錄工具
  - Nuvoton NuMicro ICP Programming Tool
  - Nuvoton NuMicro ISP Programming Tool(可選)

3.2 硬體需求

- NuMaker-PFM-M487 V3.0

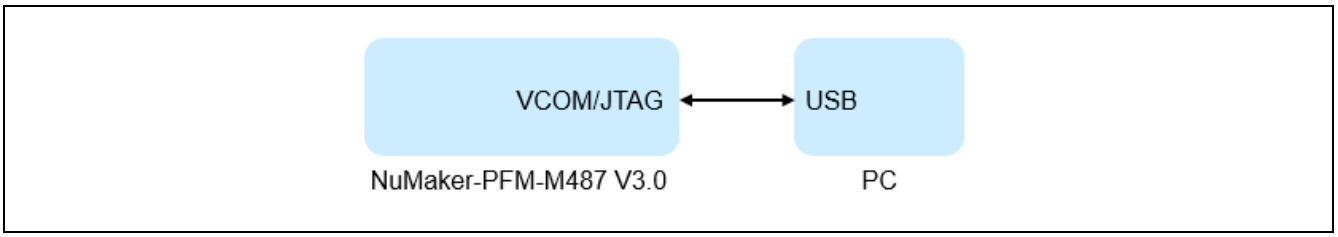

圖 3-1 連接至電腦

4. 目錄資訊

目錄架構如下所示。

EC_M480_ISP_Firmware_Update_and_Restore_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

圖 4-1 目錄資訊

5. 範例程式執行

1. 編譯範例程式。
2. 透過 ICE 連接 Nuvoton NuMicro ICP Programming Tool，選擇 M480 Series。
   - 勾選 LDROM 以及 Options 內 Erase Whole Chip，點擊 LDROM 選擇編譯後的檔案後 Start 開始燒錄 LDROM。
   - 勾選 Config 並取消勾選 LDROM 與 Options 內 Erase Whole Chip，點擊 Setting 勾選 LDROM，確認後 Start 開始燒錄 CONFIG。
   - 勾選 APROM 並取消勾選 Config，於 Offset 處填入 APROM_SIZE/2，點擊 APROM 選擇 BSP 內任意編譯過的範例程式後 Start 開始燒錄 APROM。
3. 燒錄完成後程式將由 LDROM 啟動並開始更新執行區塊，執行結果即為寫入備份區塊的應用程式。

6. 修訂紀錄

Date	Revision	Description
2022.08.24	1.00	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.