M032 ISP 更新 APROM 與 SPI Flash

NuMicro® 32 位元系列微控制器範例代碼介紹

文件資訊

應用簡述	本範例程式使用 ISP LDROM 程式更新 APROM 與 SPI Flash 中的數據
BSP 版本	M031_Series_BSP_CMSIS_V3.07.000
硬體平台	NuMaker-M032KI V1.0

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

APROM 通常儲存使用者代碼，而 LDROM 則用於儲存 Bootloader of In-System-Programming (ISP)。此 LDROM 範例程式透過 UART 介面接收使用者代碼，然後透過軟體暫存器控制將使用者代碼更新到 APROM 中，從而實現 In-System Programming (ISP) 功能。它還支援透過 SPI Flash 的指令集將資料更新到外部 SPI Flash。

1.1 原理

無論接收到的確認封包還是發送的命令封包，ISP 收發數據封包的大小均為 64 位元組。數據封包透過 UART 介面發送和接收。

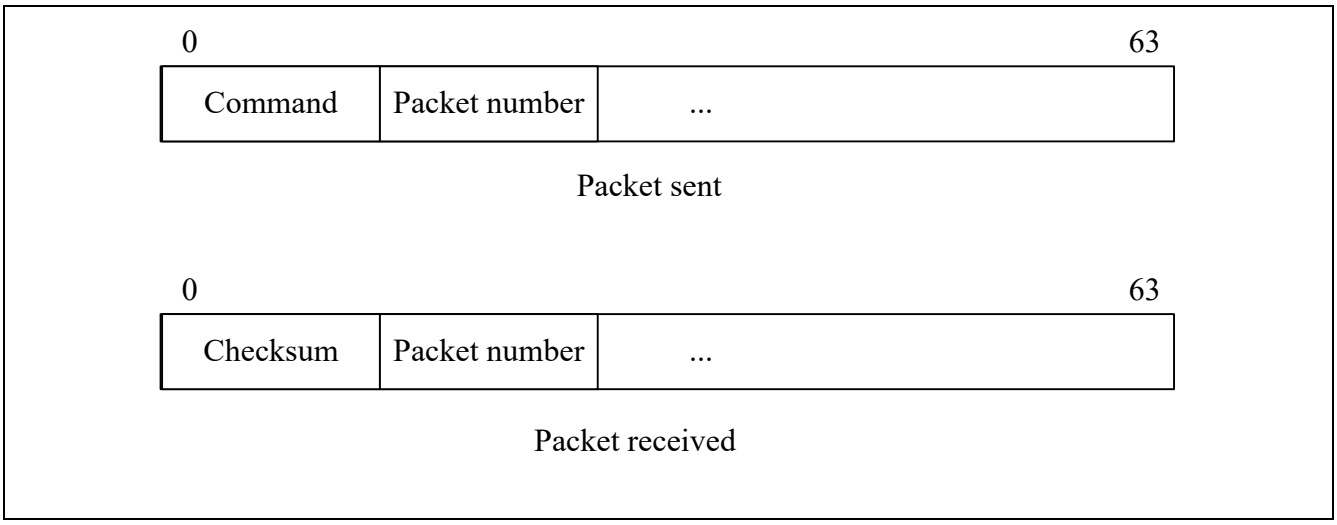

圖 1-1 ISP 封包的發送和接收

1.1.1 ISP 指令集

ISP 支援的命令如下所示。

Command	Command Code	ACK	Command Description
CMD_ERASE_SPI_FLASH	0x000000D0	Yes	從偏移位址（預設為 0）開始擦除 64 KB SPI Flash，直到所有區塊都被擦除。

CMD_UPDATE_SPI_FLASH	0x000000D1	Yes	從使用者指定的位址對 SPI Flash 進行燒錄。
CMD_GET_SPI_JEDEC	0x000000D2	Yes	取得 SPI Flash 的 JEDEC ID。

表 1-1 ISP 命令集

CMD_ERASE_SPI_FLASH 命令格式

Host Send (unit by byte):

4	4	4	4	n
CMD	Packet Number	Start Address	Data	Data……

Device ACK:

4	4	
Checksum	Packet Number	Total Checksum

此命令用於指示 ISP 擦除 SPI Flash。使用者可以指定要擦除的起始位址。當 ISP 收到命令封包後，它將發送 SPI Flash 的區塊擦除指令來擦除 64 KB 的 SPI Flash。擦除操作完成後，ISP 會計算接收到的整個資料封包的校驗值，然後向主機發送 ACK 封包。

主機收到 ACK 封包後，會檢查校驗值是否正確。如果校驗值正確，則表示資料傳輸成功。主機將繼續向 ISP 發送擦除命令，直到整個 SPI Flash 被擦除。

CMD_UPDATE_SPI_FLASH 命令格式

Host Send:

4	4	4	4	n
CMD	Packet Number	Start Address	Length	Data……

Device ACK:

4	4	
Checksum	Packet Number	Total Checksum

此命令用於指示 ISP 對 SPI Flash 進行燒錄。使用者可以指定要燒錄的起始位址。若程式資料過大，無法在單一封包中完成傳輸，ISP Programming Tool 會將資料分割成多個封包，透過多次傳輸方式送出。

ISP 會從接收到的封包中讀取程式資料，並將其燒錄至 SPI Flash 的指定位址。當燒錄作業完成後，ISP 會計算整個已接收資料封包的校驗值，然後向主機傳送 ACK 封包。

主機收到 ACK 封包後，會檢查校驗值是否正確。若校驗值正確，則表示資料傳輸成功。

CMD_GET_SPI_JEDEC 命令格式

Host Send:

4	4
CMD	Packet Number

Device ACK:

4	4	4
Checksum	Packet Number	JEDEC ID

此命令用於指示 ISP 讀取 SPI Flash 的 JEDEC ID 資訊，並將其傳送給主機。

1.1.2 ISP 更新 SPI Flash 流程

更新 SPI Flash 的步驟如下：

1. 使用 Keil project 開啟 **Chip Settings** 對話框，將 **Chips Booting Selection** 選項設為 LDROM，然後按下 **OK**，如圖 1-2 所示。

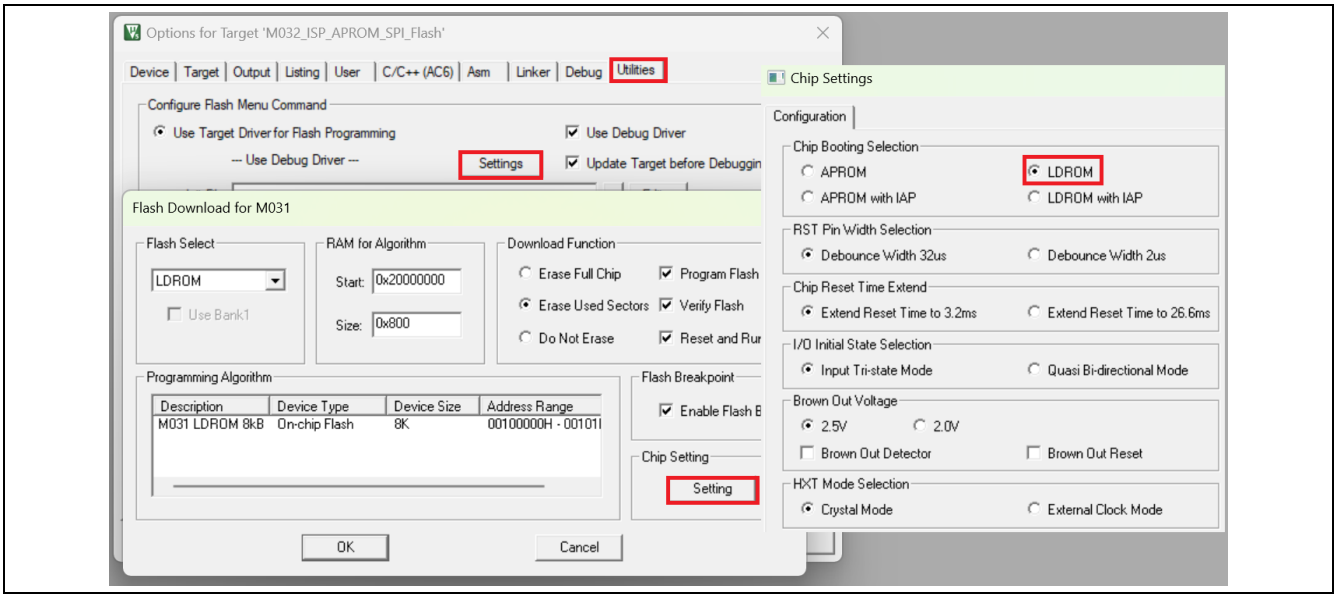

圖 1-2 從 LDROM 啟動的設定

2. 編譯程式碼並將其下載到目標晶片的 LDROM 中。

3. 如圖 1-3 所示，將 SPI Flash 的資訊加入 SPIFLASH ID.ini 檔案中。例如 **EF4016** 是 Winbond W25Q32JV SPI Flash 的 JEDEC ID，**4M** 表示其儲存容量為 4 MB。此步驟可確保 ISP Programming Tool 在連接到目標晶片時顯示正確的 SPI Flash 容量。

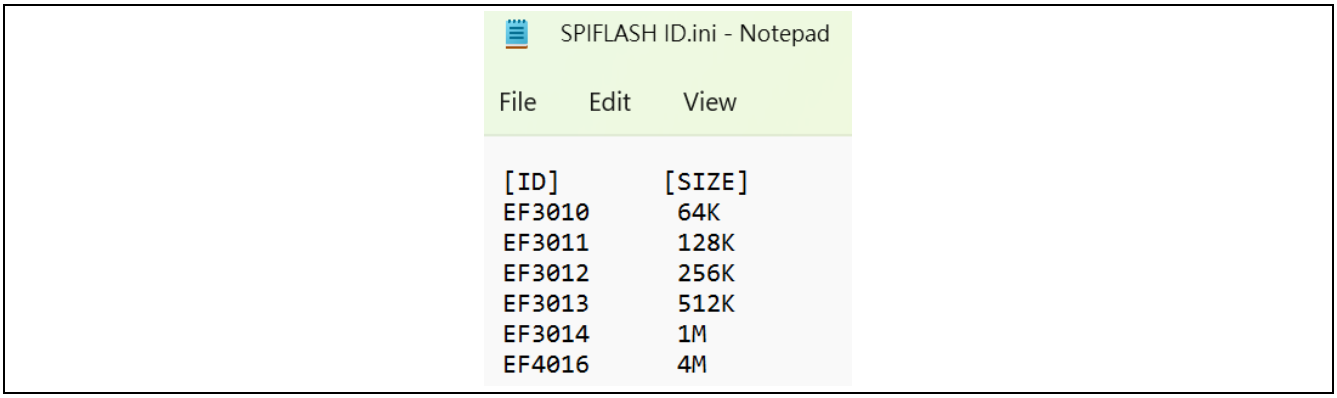

圖 1-3 SPI Flash 資訊檔案

4. 開啟 ISP Programming Tool，選擇 UART 介面進行連接。

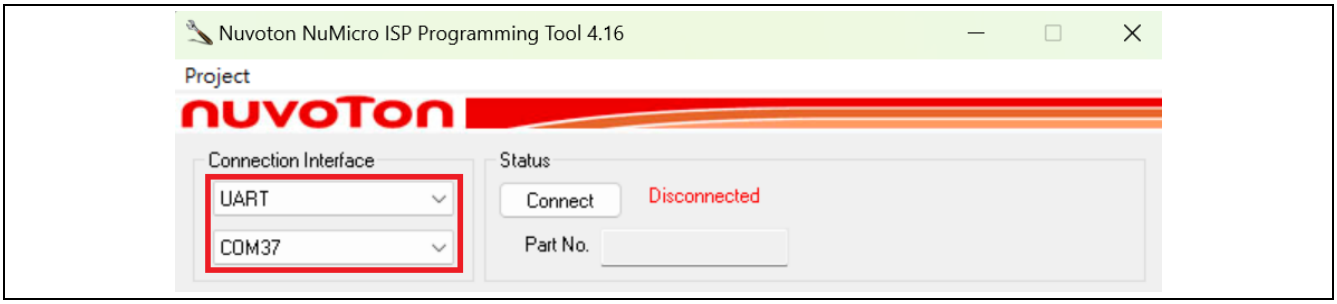

圖 1-4 設定 ISP 工具的連接接口

5. 點選 **Project** 按鈕，然後選擇 **Open SPI Flash** 選項，如圖 1-5 所示。

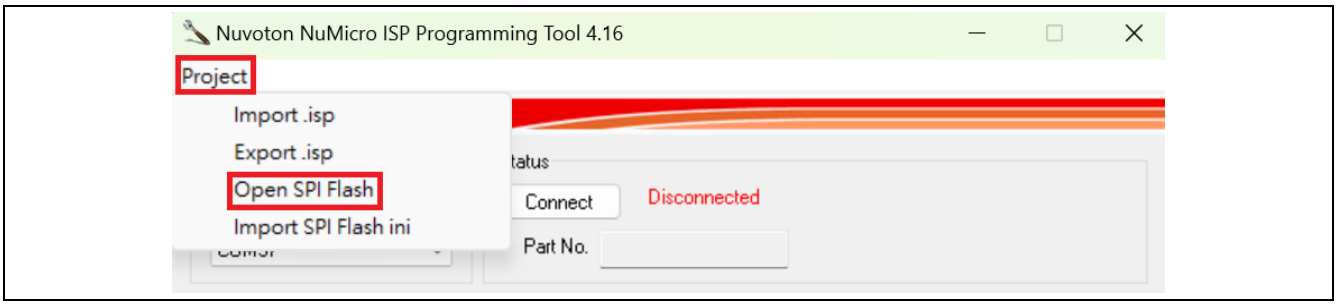

圖 1-5 Open SPI Flash 選項

6. 點選 **Project** 按鈕並匯入 SPI Flash ini 檔案，如圖 1-6 所示。

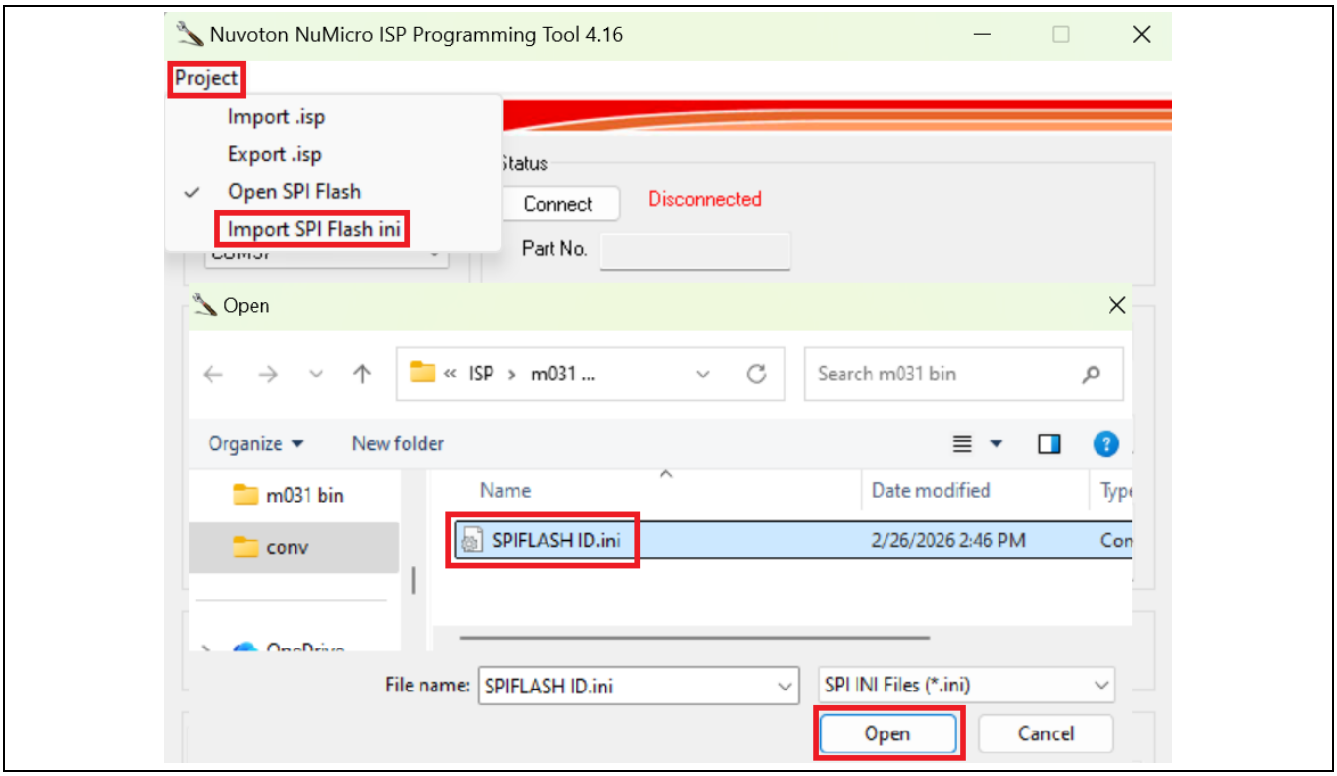

圖 1-6 匯入 SPI Flash 資訊檔案

7. 點擊 **Connect** 按鈕並重置目標晶片以執行 ISP 代碼。 ISP Programming Tool 將連接到目標晶片。

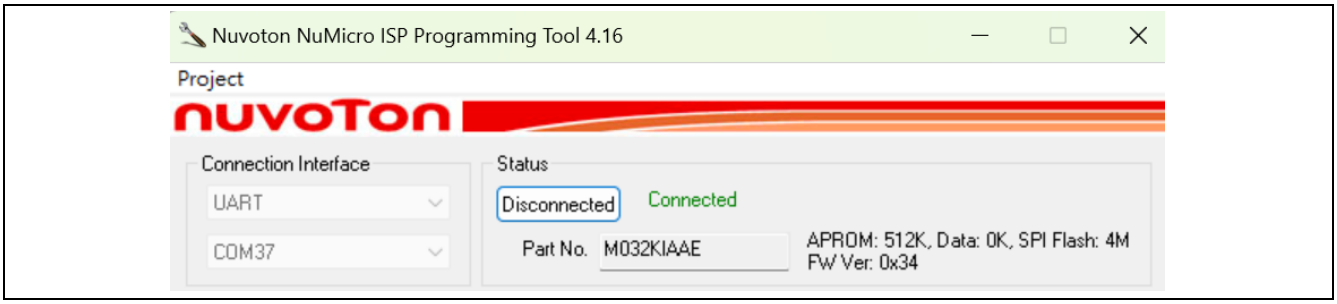

圖 1-7 透過 UART 介面連接到目標晶片

8. 使用者需要載入燒錄的文件，選擇燒錄選項為 **SPI Flash**，然後點擊 **Start** 按鈕。 ISP Programming Tool 將開始把檔案寫入到使用者指定的位址，如圖 1-8 所示。由於 ISP Programming Tool 在燒錄資料之前不會發出區塊擦除命令，如果使用者不確定 SPI Flash 的位址是否已被寫過資料，請選擇 **Erase SPI Flash** 選項，在寫入資料之前擦除整個 SPI Flash，以確保資料能夠正確寫入 SPI Flash。還有請注意 **Base Address** 選項存在 2 KB 對齊的限制。

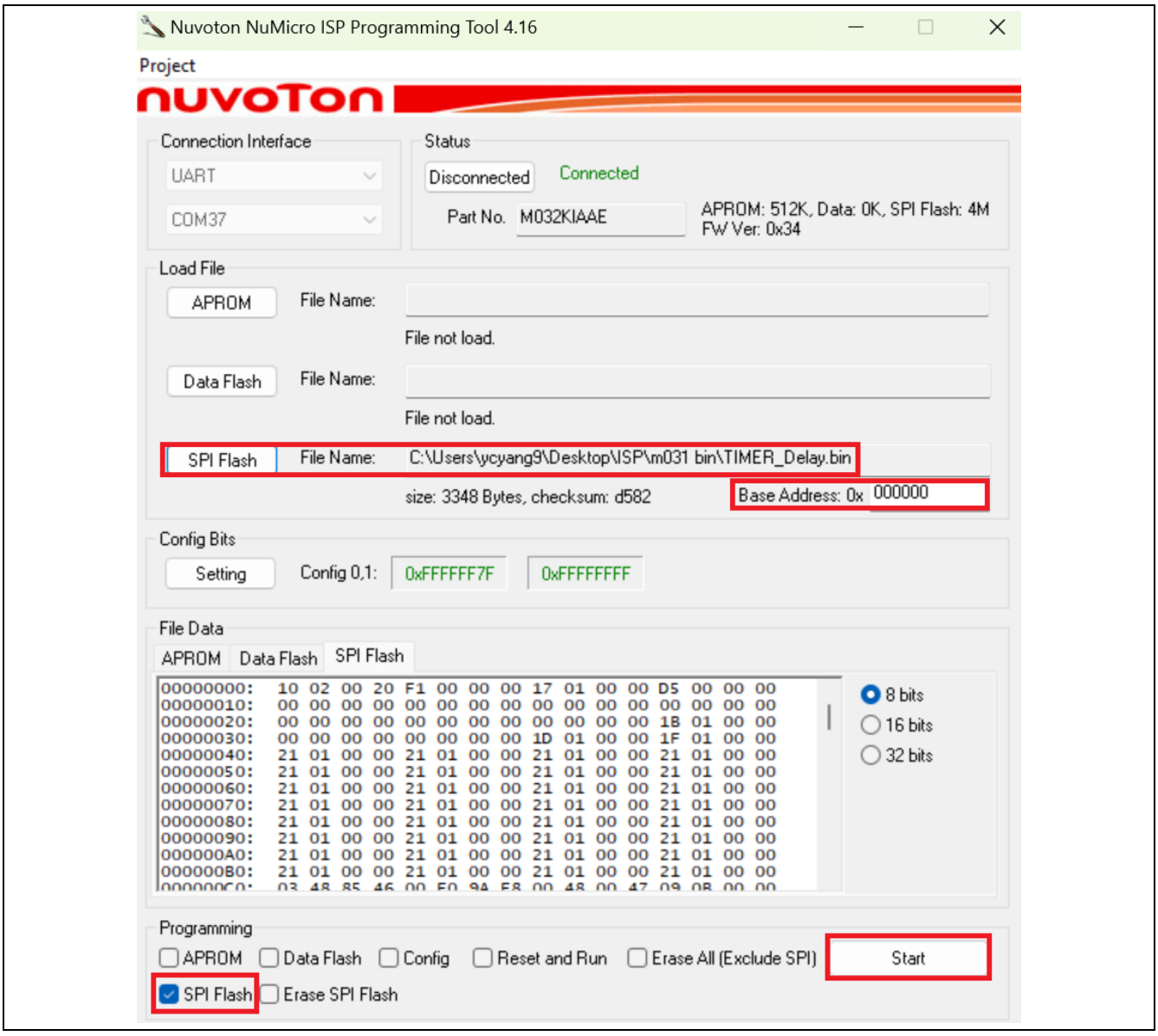

圖 1-8 透過 UART 介面將資料寫入外部 SPI Flash

1.2 執行結果

執行結果如圖 1-9 所示。

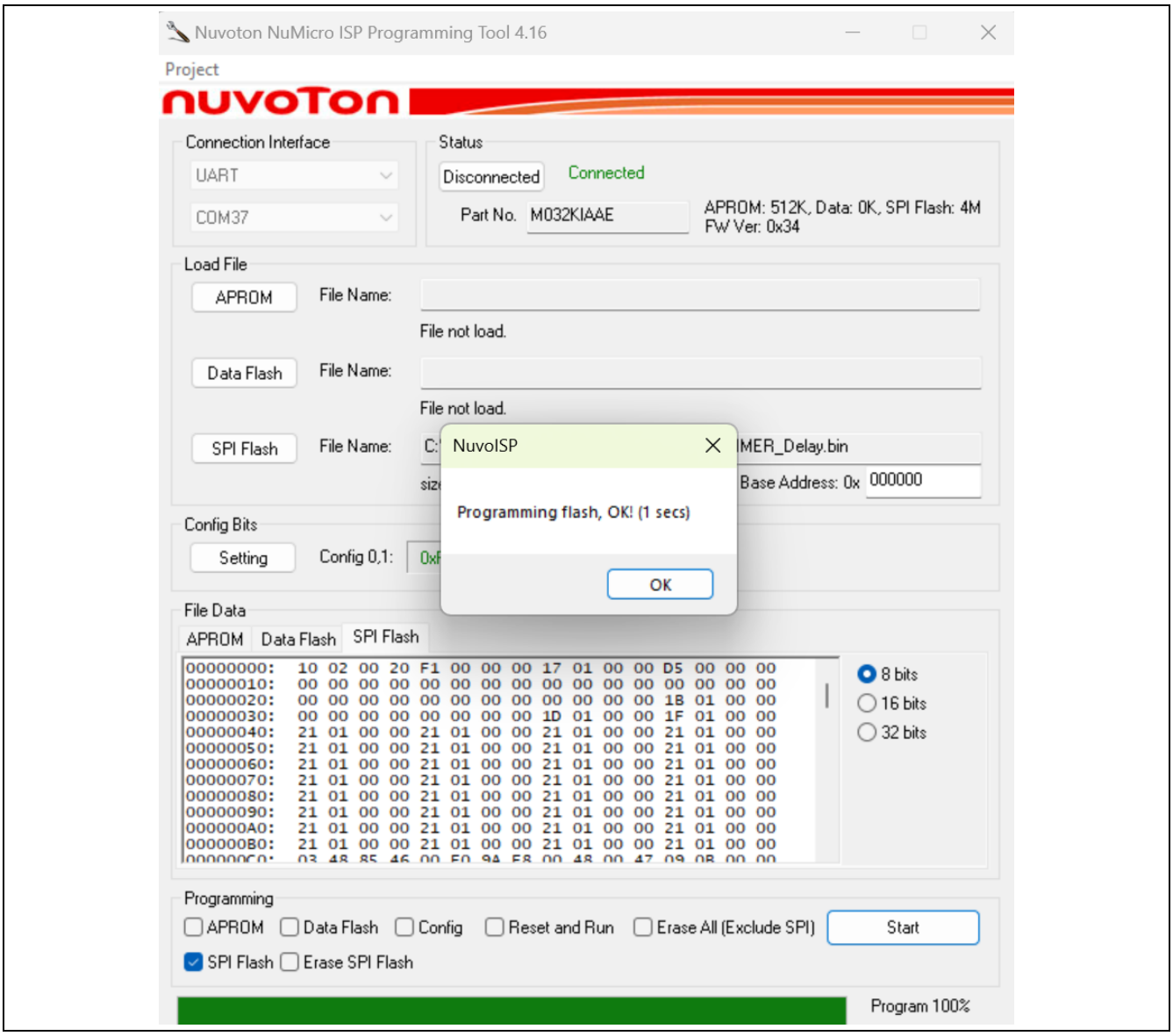
Nuvoton NuMicro ISP Programming Tool 4.16
Project
Connection Interface
UART
COM37
Status
Disconnected
Connected
Part No. M032KIAAE
APROM: 512K, Data: 0K, SPI Flash: 4M
FW Ver: 0x34
Load File
APROM
File Name:
File not load.
Data Flash
File Name:
File not load.
SPI Flash
File Name:
MER_Delay.bin
Base Address: 0x 000000
NuvoISP
Programming flash, OK! (1 secs)
OK
Config Bits
Setting
Config 0,1:
File Data
APROM
Data Flash
SPI Flash
00000000: 10 02 00 20 F1 00 00 00 17 01 00 00 D5 00 00 00
00000010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000020: 00 00 00 00 00 00 00 00 00 00 00 00 1B 01 00 00
00000030: 00 00 00 00 00 00 00 00 1D 01 00 00 1F 01 00 00
00000040: 21 01 00 00 21 01 00 00 21 01 00 00 21 01 00 00
00000050: 21 01 00 00 21 01 00 00 21 01 00 00 21 01 00 00
00000060: 21 01 00 00 21 01 00 00 21 01 00 00 21 01 00 00
00000070: 21 01 00 00 21 01 00 00 21 01 00 00 21 01 00 00
00000080: 21 01 00 00 21 01 00 00 21 01 00 00 21 01 00 00
00000090: 21 01 00 00 21 01 00 00 21 01 00 00 21 01 00 00
000000A0: 21 01 00 00 21 01 00 00 21 01 00 00 21 01 00 00
000000B0: 21 01 00 00 21 01 00 00 21 01 00 00 21 01 00 00
8 bits
16 bits
32 bits
Programming
APROM
Data Flash
Config
Reset and Run
Erase All (Exclude SPI)
Start
SPI Flash
Erase SPI Flash
Program 100%

圖 1-9 燒錄結果

2. 程式碼介紹

ISP Bootloader 的主程式設定系統頻率並初始化 UART 和 SPI 功能，它將使用 UART 與 ISP Programming Tool 通訊以進行更新。如果使用者不需要更新 APROM 或 SPI Flash，程式碼會將晶片設定為從 APROM 啟動並執行系統重置。

```
int32_t main(void)
{
    /* Init System, peripheral clock and multi-function I/O */
    SYS_Init();

    /* Init UART to 115200-8n1 */
    UART_Init();

    /* Configure SPI_FLASH_PORT as a master, MSB first, 8-bit transaction, SPI Mode-0
    timing, clock is 2MHz */
    SPI_Open(SPI_FLASH_PORT, SPI_MASTER, SPI_MODE_0, 8, 2000000);

    /* Disable auto SS function, control SS signal manually. */
    SPI_DisableAutoSS(SPI_FLASH_PORT);

    /* Checking if flash page size matches with target chip's */
    if ((GET_CHIP_SERIES_NUM == CHIP_SERIES_NUM_I) || (GET_CHIP_SERIES_NUM ==
    CHIP_SERIES_NUM_G))
    {
        if (FMC_FLASH_PAGE_SIZE != 2048)
        {
            /* FMC_FLASH_PAGE_SIZE is different from target device */
            /* Please enable the compiler option PAGE_SIZE_2048 in fmc.h */
            while(SYS->PDID);
        }
    }
    else
    {
        if (FMC_FLASH_PAGE_SIZE != 512)
        {
            /* FMC_FLASH_PAGE_SIZE is different from target device */
            /* Please disable the compiler option PAGE_SIZE_2048 in fmc.h */
            while(SYS->PDID);
        }
    }

    CLK->AHBCLK |= CLK_AHBCLK_ISPCKEN_Msk;

    /* Unlock protected registers */
    SYS_UnlockReg();

    /* Enable FMC ISP function. Before using FMC function, it should unlock system
    register first. */
    FMC->ISPCTL |= (FMC_ISPCTL_ISPEN_Msk | FMC_ISPCTL_APUEN_Msk);

    g_apromSize = GetApromSize();
```

```
    GetDataFlashInfo(&g_dataFlashAddr, &g_dataFlashSize);
    SysTick->LOAD = 300000 * CyclesPerUs;
    SysTick->VAL   = (0x00);
    SysTick->CTRL = SysTick->CTRL | SysTick_CTRL_CLKSOURCE_Msk | SysTick_CTRL_ENABLE_Msk;

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

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

        if (SysTick->CTRL & SysTick_CTRL_COUNTFLAG_Msk)
        {
            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);
        }
    }

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

此函式解析來自 ISP Programming Tool 的命令，以完成更新流程。

```
int ParseCmd(unsigned char *buffer, uint8_t len)
{
    static uint32_t StartAddress, TotalLen, LastDataLen;
    uint32_t lcmd, srclen, i, regcnf0, security;
    unsigned char *pSrc;
    static uint32_t gcmd;
    uint32_t u32Addr, u32NTx, u32Id;

    g_pu8Response = response_buff;
```

```
    pSrc = buffer;
    srclen = len;
    lcmd = inpw(pSrc);
    outpw(g_pu8Response + 4, 0);
    pSrc += 8;
    srclen -= 8;
    ReadData(Config0, Config0 + 16, (uint32_t *)(g_pu8Response + 8)); /* Read config */
    regcnf0 = *(uint32_t *)(g_pu8Response + 8);
    security = regcnf0 & 0x2;

    if (lcmd == CMD_SYNC_PACKNO)
    {
        g_packno = inpw(pSrc);
    }

    if ((lcmd) && (lcmd != CMD_RESEND_PACKET))
    {
        gcmd = lcmd;
    }

    if (lcmd == CMD_GET_FWVER)
    {
        g_pu8Response[8] = FW_VERSION;
    }
    else if (lcmd == CMD_GET_DEVICEID)
    {
        outpw(g_pu8Response + 8, SYS->PDID);

        ExitParseCmd(buffer, len);
        return (0);
    }
    else if (lcmd == CMD_RUN_APROM || lcmd == CMD_RUN_LDROM || lcmd == CMD_RESET)
    {
        SYS->RSTSTS = 3; /* Clear bit */

        /* Set BS */
        if (lcmd == CMD_RUN_APROM)
        {
            i = (FMC->ISPCTL & 0xFFFFFFFC);
        }
        else if (lcmd == CMD_RUN_LDROM)
        {
            i = (FMC->ISPCTL & 0xFFFFFFFC);
            i |= 0x00000002;
        }
        else
        {
            i = (FMC->ISPCTL & 0xFFFFFFFE);/* ISP disable */
        }

        FMC->ISPCTL = i;
        SCB->AIRCR = (V6M_AIRCR_VECTKEY_DATA | V6M_AIRCR_SYSRESETREQ);

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

```
    else if (lcmd == CMD_CONNECT)
    {
        g_packno = 1;

        ExitParseCmd(buffer, len);
        return (0);
    }
    else if ((lcmd == CMD_UPDATE_APROM) || (lcmd == CMD_ERASE_ALL))
    {
        EraseAP(FMC_APROM_BASE, (g_apromSize < g_dataFlashAddr) ? g_apromSize :
        g_dataFlashAddr); /* Erase APROM */

        if (lcmd == CMD_ERASE_ALL)
        {
            EraseAP(g_dataFlashAddr, g_dataFlashSize);
            *(uint32_t *)(g_pu8Response + 8) = regcnf0 | 0x02;
            UpdateConfig((uint32_t *)(g_pu8Response + 8), NULL);
        }

        bUpdateApromCmd = TRUE;
    }

    if ((lcmd == CMD_UPDATE_APROM) || (lcmd == CMD_UPDATE_DATAFLASH))
    {
        if (lcmd == CMD_UPDATE_DATAFLASH)
        {
            StartAddress = g_dataFlashAddr;

            if (g_dataFlashSize)   /*g_dataFlashAddr*/
            {
                EraseAP(g_dataFlashAddr, g_dataFlashSize);
            }
            else
            {
                ExitParseCmd(buffer, len);
                return (0);
            }
        }
        else
        {
            StartAddress = 0;
        }

        TotalLen = inpw(pSrc + 4);
        pSrc += 8;
        srclen -= 8;
    }
    else if (lcmd == CMD_UPDATE_CONFIG)
    {
        if ((security == 0) && (!bUpdateApromCmd))   /* Security lock */
        {
            ExitParseCmd(buffer, len);
            return (0);
        }

        UpdateConfig((uint32_t *)(pSrc), (uint32_t *)(g_pu8Response + 8));
```

```
        GetDataFlashInfo(&g_dataFlashAddr, &g_dataFlashSize);

        ExitParseCmd(buffer, len);
        return (0);
    }
    else if (lcmd == CMD_RESEND_PACKET)     /* For APROM&Data flash only */
    {
        uint32_t PageAddress;
        StartAddress -= LastDataLen;
        TotalLen += LastDataLen;
        PageAddress = StartAddress & (0x100000 - FMC_FLASH_PAGE_SIZE);

        if (PageAddress >= Config0)
        {
            ExitParseCmd(buffer, len);
            return (0);
        }

        ReadData(PageAddress, StartAddress, (uint32_t *)aprom_buf);
        FMC_Erase_User(PageAddress);
        WriteData(PageAddress, StartAddress, (uint32_t *)aprom_buf);

        if ((StartAddress % FMC_FLASH_PAGE_SIZE) >= (FMC_FLASH_PAGE_SIZE - LastDataLen))
        {
            FMC_Erase_User(PageAddress + FMC_FLASH_PAGE_SIZE);
        }

        ExitParseCmd(buffer, len);
        return (0);
    }
    else if (lcmd == CMD_GET_JEDECID)
    {
        u32Id = SpiFlash_ReadJEDECID();
        outpw(g_pu8Response + 8, u32Id);

        ExitParseCmd(buffer, len);
        return (0);
    }
    else if (lcmd == CMD_ERASE_SPIFLASH)
    {
        u32Addr = inpw(pSrc);
        SpiFlash_EraseBlock(u32Addr);
    }
    else if (lcmd == CMD_UPDATE_SPIFLASH)
    {
        u32Addr = inpw(pSrc);
        u32NTx = inpw(pSrc + 4);
        pSrc += 8;
        SpiFlash_NormalPageProgram(u32Addr, u32NTx, pSrc);
        memset(pSrc, 0, u32NTx);
        SpiFlash_NormalRead(u32Addr, u32NTx, pSrc);
    }

    if ((gcmd == CMD_UPDATE_APROM) || (gcmd == CMD_UPDATE_DATAFLASH))
    {
        if (TotalLen < srclen)
```

```
        {
            srclen = TotalLen;/* Prevent last package from over writing */
        }

        TotalLen -= srclen;
        WriteData(StartAddress, StartAddress + srclen, (uint32_t *)pSrc);
        memset(pSrc, 0, srclen);
        ReadData(StartAddress, StartAddress + srclen, (uint32_t *)pSrc);
        StartAddress += srclen;
        LastDataLen =  srclen;
    }

    ExitParseCmd(buffer, len);
    return (0);
}
```

ParseCmd() 函式的這一部分用於更新 SPI Flash，並將 JEDEC ID 回覆 ISP Programming Tool。

```
    else if (lcmd == CMD_GET_JEDECID)
    {
        u32Id = SpiFlash_ReadJEDECID();
        outpw(g_pu8Response + 8, u32Id);

        ExitParseCmd(buffer, len);
        return (0);
    }
    else if (lcmd == CMD_ERASE_SPIFLASH)
    {
        u32Addr = inpw(pSrc);
        SpiFlash_EraseBlock(u32Addr);
    }
    else if (lcmd == CMD_UPDATE_SPIFLASH)
    {
        u32Addr = inpw(pSrc);
        u32NTx = inpw(pSrc + 4);
        pSrc += 8;
        SpiFlash_NormalPageProgram(u32Addr, u32NTx, pSrc);
        memset(pSrc, 0, u32NTx);
        SpiFlash_NormalRead(u32Addr, u32NTx, pSrc);
    }
```

ParseCmd() 函式的這一部分用於更新 APROM 和 Data Flash。

```
    if ((gcmd == CMD_UPDATE_APROM) || (gcmd == CMD_UPDATE_DATAFLASH))
    {
        if (TotalLen < srclen)
        {
            srclen = TotalLen;/* Prevent last package from over writing */
        }

        TotalLen -= srclen;
        WriteData(StartAddress, StartAddress + srclen, (uint32_t *)pSrc);
        memset(pSrc, 0, srclen);
        ReadData(StartAddress, StartAddress + srclen, (uint32_t *)pSrc);
        StartAddress += srclen;
```

```
        LastDataLen =  srclen;
    }
```

3. 軟體與硬體需求

3.1 軟體需求

- BSP 版本
  - M031_Series_BSP_CMSIS_V3.07.000
- IDE 版本
  - Keil uVision 5.38

3.2 硬體需求

- 電路元件
  - NuMaker-M032KI V1.0
  - Level1-Training
- 腳位連結
  - 將 Nu-Link2-Me 的 USB 介面連接到 PC 的 USB 端口，以顯示範例代碼的執行結果。

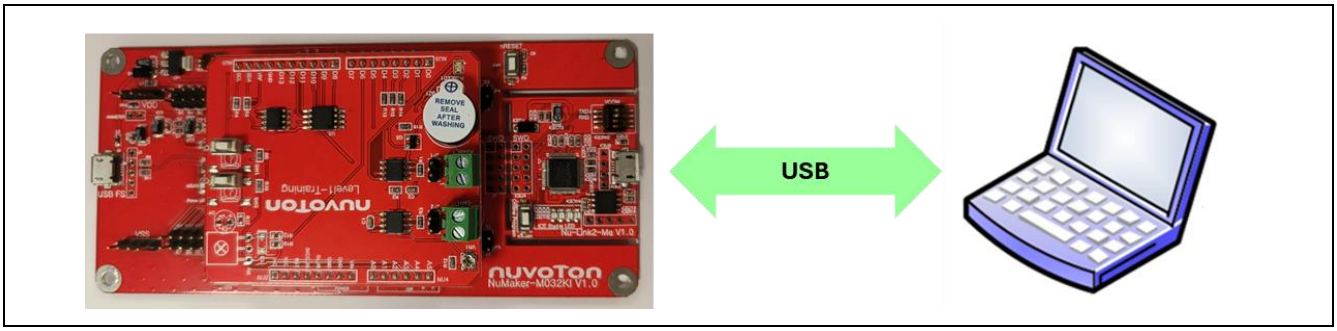

圖 3-1 腳位連結

4. 目錄資訊

目錄結構如下圖所示。

EC_M032_ISP_Update_APROM_and_SPI_Flash_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	
Project	Source file of example code
WindowsTool	ISP Programming Tool

圖 4-1 目錄資訊

5. 範例程式執行

5.1 Keil MDK

1. 在 Windows 檔案總管中進入 Keil 專案資料夾，再開啟 *M032_ISP_Update_APROM_and_SPI_Flash.uvprojx* 專案檔。
2. 編譯程式碼
3. 下載程式碼
4. 進入除錯模式
   - 執行程式碼

5.2 VS Code

1. 在 **VS Code** IDE 中開啟 **File** 選單
   - 選取 **Open Folder**
   - 選取專案目錄下的 **VSCode** 資料夾
2. 選擇 CMSIS
   - 點選 **Build solution** 編譯專案
   - 點選 **Load & Run application** 下載映像檔至目標晶片
   - 點選 **Load & Debug application** 進入偵錯模式

6. 修訂紀錄

Date	Revision	Description
2026.06.23	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.