M032 ISR運行在SRAM

NuMicro® 32 位系列微控制器範例代碼介紹

文件信息

應用簡述	本範例使用 M032 將中斷重映射到 SRAM 運行
BSP 版本	M031_Series_BSP_CMSIS_V3.03.000
開發平臺	NuMaker-M032SE V1.3

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

本示例使用M032 ISP功能將中斷重映射到SRAM運行，實現當FMC操作Flash時仍能正常使用UART功能。

芯唐支持映射中斷向量表到SRAM的晶片如下表，具體設置需根據不同情境而修改。

不支持映射中斷向量表至SRAM	支持映射中斷向量表至SRAM
M051 M058S M0518 M0519 Mini51 Mini55 Mini57 Mini58 Nano1x2 Nano100 NUC100 NUC122 NUC123 NUC131 NUC200 NUC029xAN NUC029xAE NUC029xDE NUC029xEE	Cortex®-M4 全系列 (VTOR 寄存器) Cortex®-M23 全系列 (VTOR 寄存器) M031 (FMC 支援) M0564 (FMC 支援) Nano103 (FMC 支援) NUC121/125 (FMC 支援) NUC126 (FMC 支援) NUC029xGE (FMC 支援)

表 1-1 支持映射中斷向量表重映射的晶片

1.1 原理

當Flash存儲控制器(FMC)進行操作時(運行程序與所操作Flash處於同一個bank)，CPU任何訪問Flash的操作都會造成CPU停機直到FMC完成操作。如果FMC執行頁擦除操作，可能會造成CPU長時間等待。

爲了避免這種情況，用戶如需要在FMC頁擦除時執行其他程序，可以把程序放在SRAM中執行。如需要響應中斷，可先將中斷向量空間及中斷處理函數拷貝到SRAM中，然後使用ISP功能重映射向量空間到SRAM。

在IAP模式下，ISP功能可將SRAM連續的512字節映射爲中斷向量空間。如中斷向量空間被重映射到SRAM中，CPU不需要訪問Flash獲取中斷向量地址。當程序在SRAM中執行，CPU

不會訪問Flash讀取指令。故當中斷產生時，CPU可以零等待讀取SRAM中的中斷向量并跳轉到SRAM中的中斷函數運行。圖 1-1展示了向量表映射的流程。

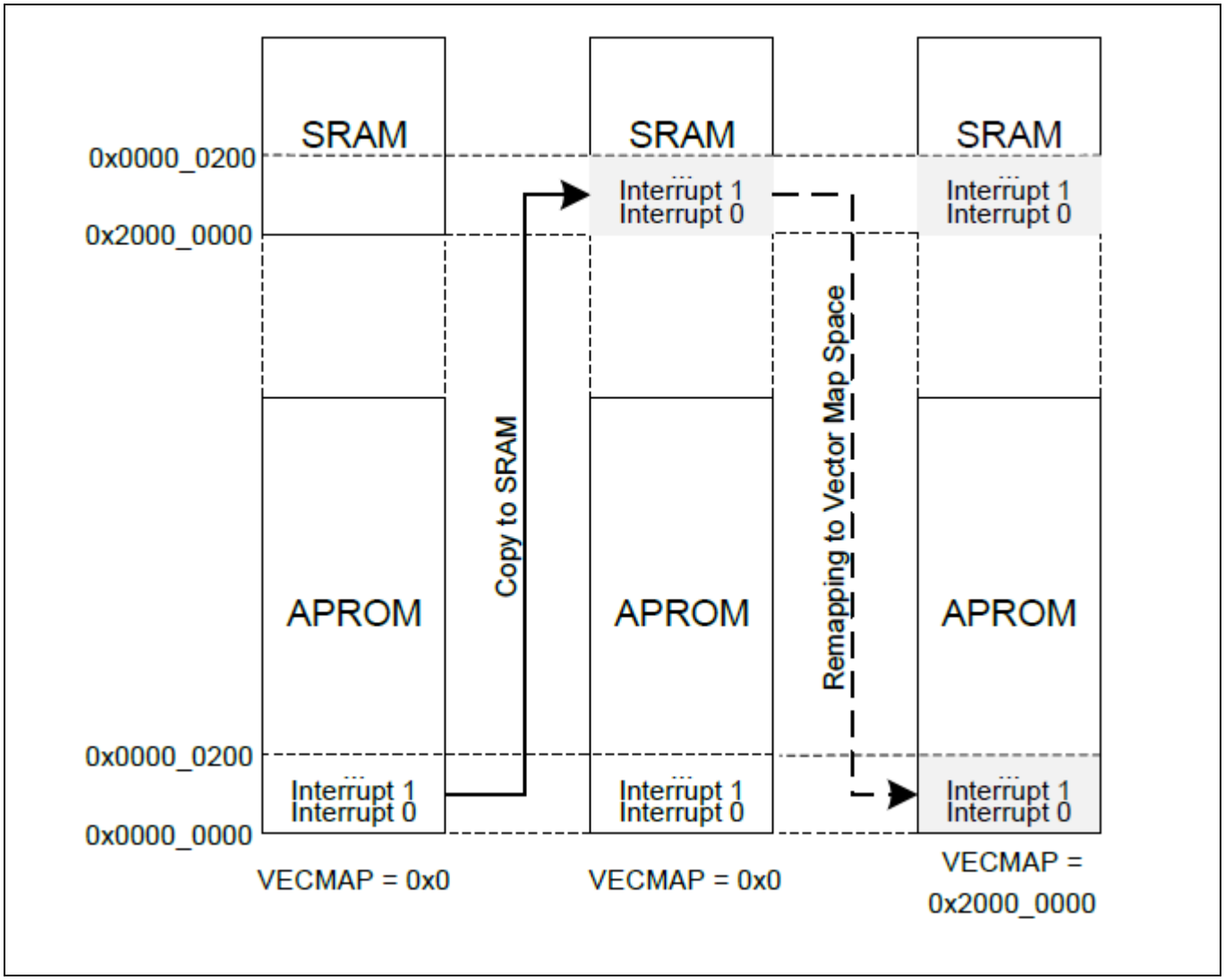

圖 1-1 使用ISP將中斷向量空間重映射到SRAM區域

此功能需使能晶片的 IAP 功能，晶片配置如下圖 1-2。

圖 1-2 晶片配置

1.2 執行結果

1.2.1 中斷向量無映射到 SRAM

在KEIL目錄下選擇**Project->Options for Target** 'ISR_In_SRAM'，在"Preprocessor Symbols"中去掉定義REMAP_ISR_IN_SRAM，再點擊Build編譯工程。待編譯完成下載到開發板。

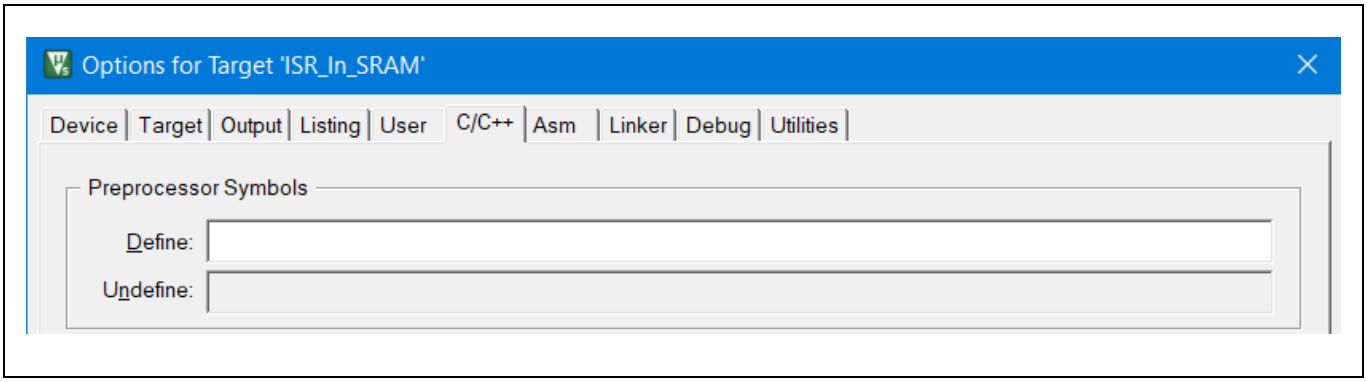

圖 1-3 去掉REMAP_ISR_IN_SRAM定義

打開串口工具eagleCom，並使用 "週期發送"功能，間隔1毫秒每次發送400個字符'1'，一直發送數據給開發板串口。運行結果如圖 1-4所示。可以看出，總共發送了95600字節，實際接收到6480字節。

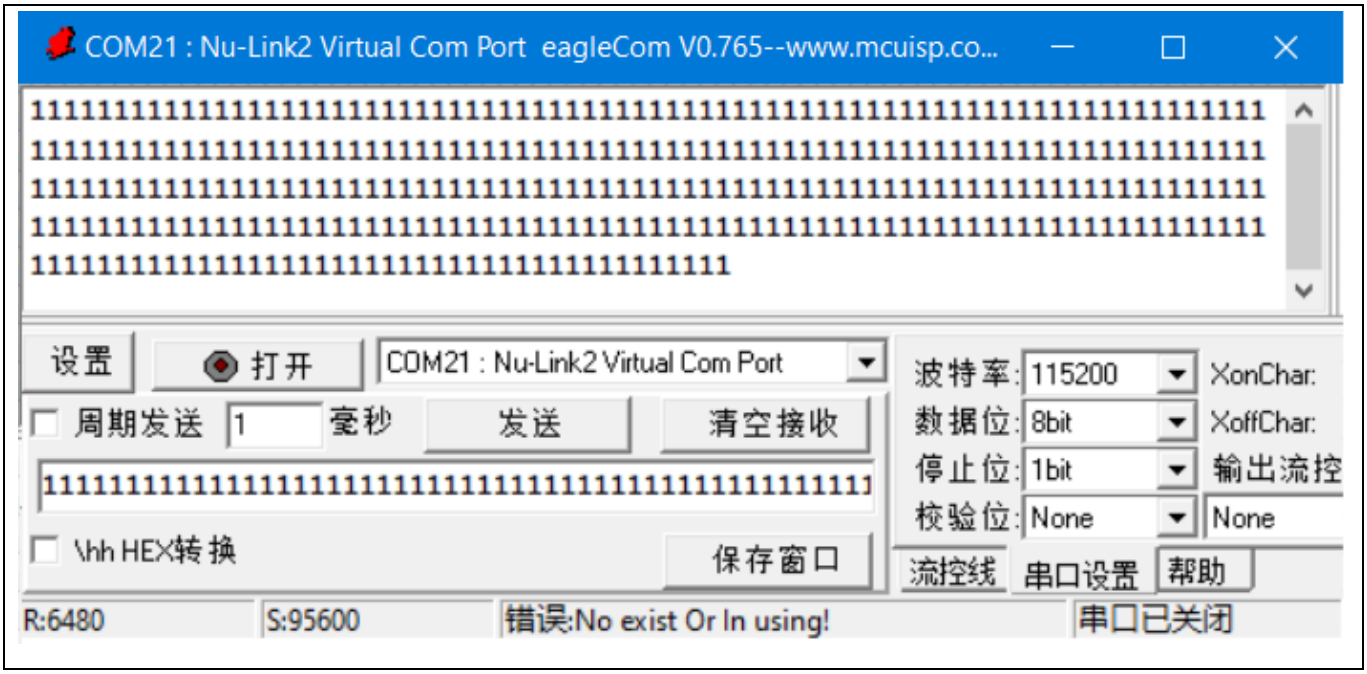

圖 1-4 中斷向量表在Flash上運行結果

使用邏輯分析儀抓取UART的數據波形，可以看到M032在收到UART數據後，存在19.2ms的時間段，沒有處理幷返回UART數據。這是因爲M032 FMC在擦除Flash指定區域時，晶片無法同時訪問處於同一bank的程序，導致存放在Flash的程序沒辦法及時執行。

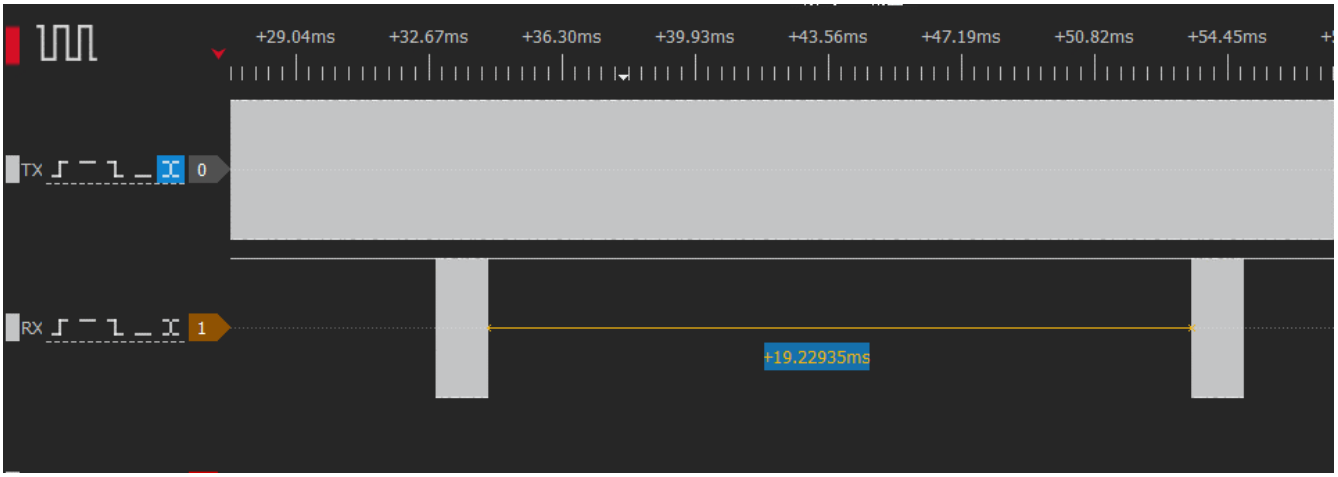

圖 1-5 當FMC訪問Flash時UART接收發送圖1

1.2.2 中斷向量映射到 SRAM

在KEIL目錄下選擇**Project->Options for Target** ‘ISR_In_SRAM’，在“Preprocessor Symbols”中添加定義REMAP_ISR_IN_SRAM。再點擊Build編譯工程。待編譯完成下載到開發板。

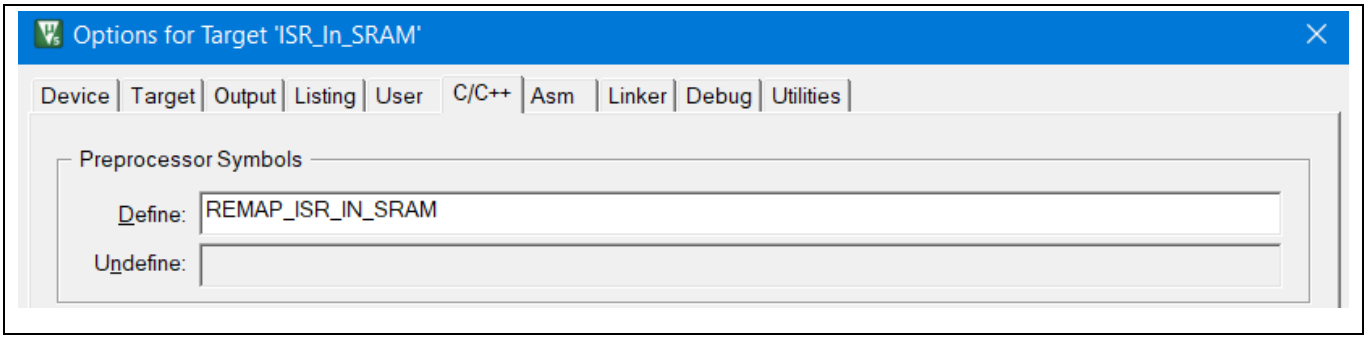

圖 1-6 增加REMAP_ISR_IN_SRAM定義

打開串口工具eagleCom，並使用 “週期發送”功能，間隔1毫秒每次發送400個字符‘1’，一直發送數據給開發板串口。運行結果如圖 1-7所示。可以看出，總共發送了59200字節，實際接收到59200字節。

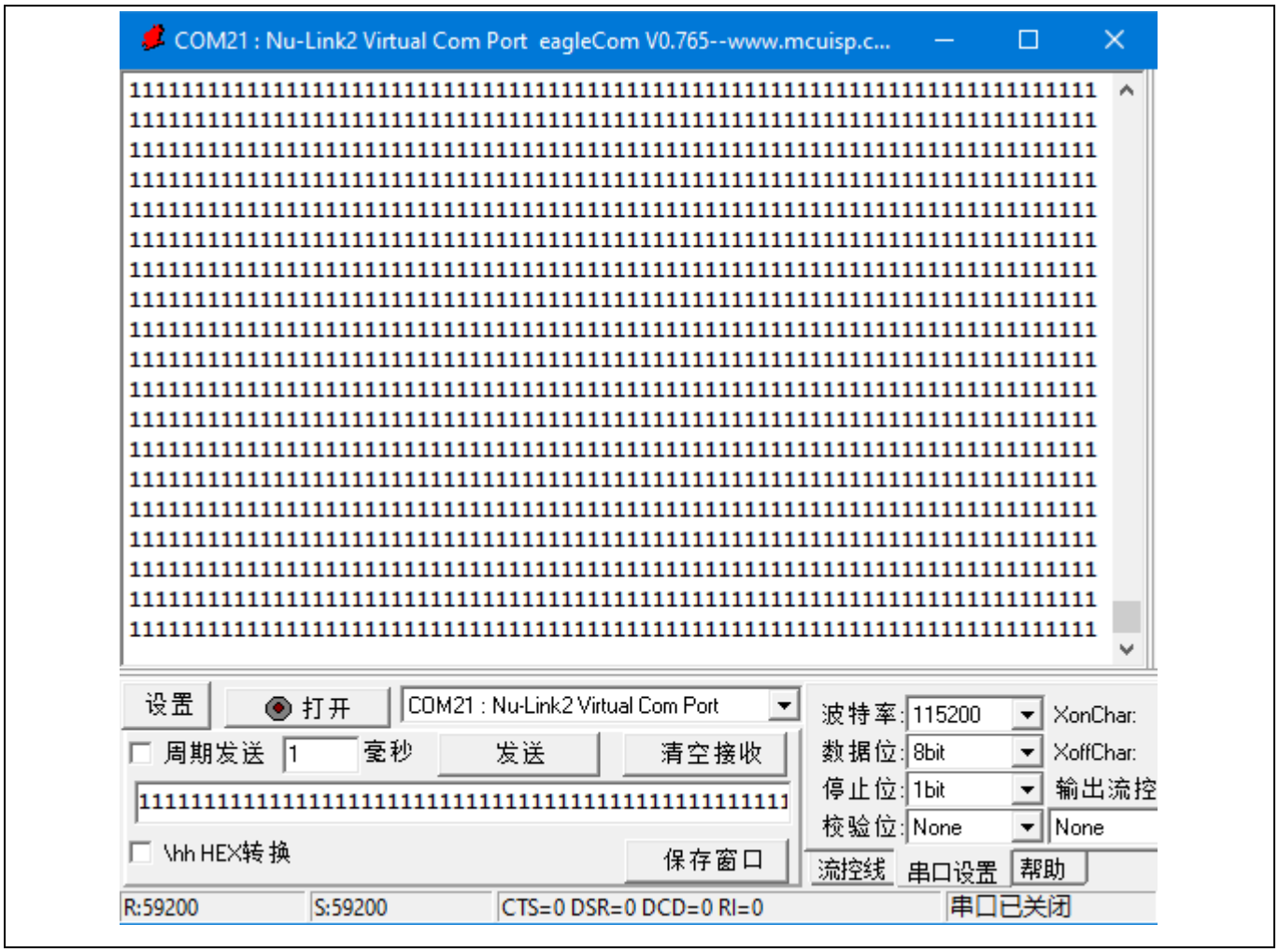

圖 1-7 中斷向量表在SRAM上運行結果

使用邏輯分析儀抓取UART的數據波形，如圖 1-8可以看到M032在收到UART數據後，可以正常接收處理。所以將程序放在SRAM中執行並重映射中斷向量空間到SRAM中可避免Flash編程時長時間停機等待。

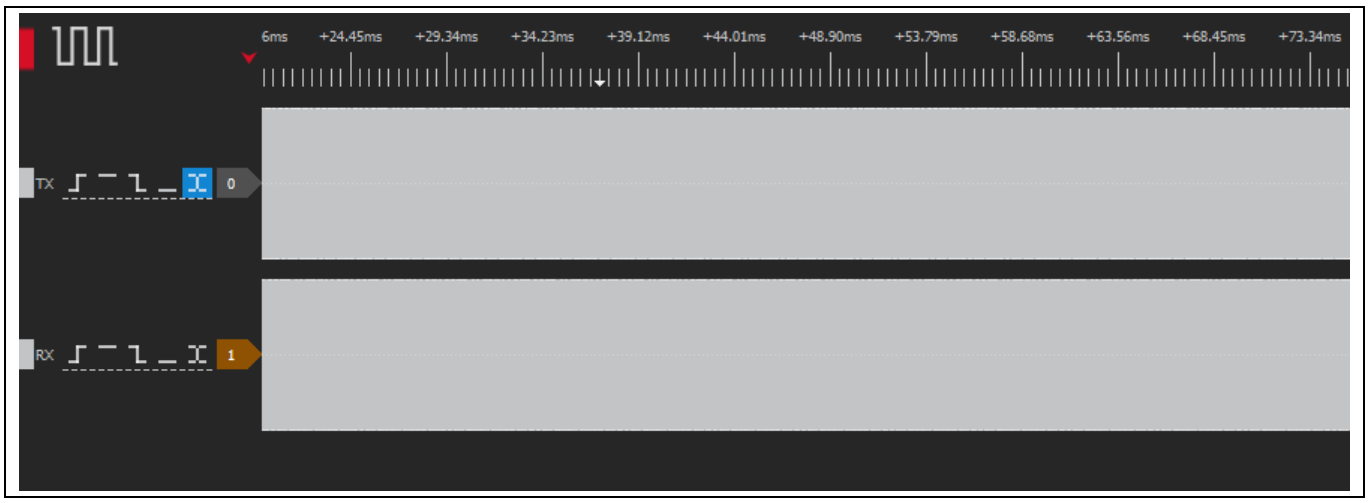

圖 1-8 當FMC訪問Flash時UART接收發送圖2

2 代碼介紹

使用 Scatter File 分配晶片內存存儲空間,在 SRAM 中劃分出中斷向量空間所需要的連續 512 字節地址空間。

```
FLASH 0x00000000
{
    ER_STARTU +0
    {
        startup_M031Series_SRAM.o(RESET, +First)
    }
    ER_RO +0
    {
        *(+RO)
    }

    ;on SRAM region
    VECTORS_SRAM 0x20000000 0x200 {
        *(VectorSection)
    }
    FASTCODE_INIT +0
    {
        uart.o(+RO)
        *(fastcode)
    }
    ER_RW +0
    {
       *(+RW, +ZI)
    }
}
```

在 SRAM 中自定義代碼段 FASTCODE_INIT，將中斷處理函數及其需要調用的函數放在此空間。請注意在此例程中 FMC 對 FLASH 的擦寫操作也需放在 SRAM 中。

```
#pragma arm section code="fastcode"

void UART02_IRQHandler(void)
{
    uint8_t u8InChar = 0xFF;

    if (UART_GET_INT_FLAG(UART0, UART_INTSTS_RDAINT_Msk))
    {
        /* Get all the input characters */
        while (UART_IS_RX_READY(UART0))
        {
            /* Get the character from UART Buffer */
            u8InChar = UART_READ(UART0);

            UART_WRITE(UART0, u8InChar);
        }
    }

    if (UART0->FIFOSTS & (UART_FIFOSTS_BIF_Msk | UART_FIFOSTS_FEF_Msk |
UART_FIFOSTS_PEF_Msk | UART_FIFOSTS_RXOVIF_Msk))
    {
        UART_ClearIntFlag(UART0, (UART_INTSTS_RLSINT_Msk | UART_INTSTS_BUFERRINT_Msk));
    }
}

void FMC_ISPGO(void)
{
```

```
    FMC->ISPTRG = FMC_ISPTRG_ISPGO_Msk;

    while (FMC->ISPTRG & FMC_ISPTRG_ISPGO_Msk) { }
}

#pragma arm section
```

拷貝中斷向量空間到 SRAM 指定地址幷設置中斷向量空間重映射地址。

```
#define VECMAP_SRAM_BASE    0x20000000UL
uint8_t u8Vectors_SRAM[512] __attribute__((section("VectorSection")));
    /* Lock protected registers */
    SYS_LockReg();

    if (FMC_GetVECMAP() != VECMAP_SRAM_BASE)
    {
        printf("Before setting VECMAP=0x%x\n\r", FMC_GetVECMAP());

        /* Mask all interrupt before changing VECMAP to avoid wrong interrupt handler
fetched */
        __set_PRIMASK(1);

        memcpy(u8Vectors_SRAM, &__Vectors, 512);

        /* Unlock protected register */
        SYS_UnlockReg();

        /* Enable FMC ISP function */
        FMC_Open();

        /* set VECMAP to SRAM */
        FMC_SetVectorPageAddr((uint32_t)&u8Vectors_SRAM);

        /* Disable FMC ISP function */
        FMC_Close();

        /* Lock protected Register */
        SYS_LockReg();

        __set_PRIMASK(0);

        printf("After  setting VECMAP=0x%x\n\r", FMC_GetVECMAP());
    }
```

3 軟體與硬體需求

- 軟件需求
  - BSP 版本
    - M031_Series_BSP_CMSIS_V3.03.000
  - IDE 版本
    - Keil uVersion 5.35
- 硬件需求
  - 電路組件
    - NuMaker-M032SE V1.3

4 目錄資訊

EC_M032_ISR_In_SRAM_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. 根據目錄信息章節進入 ExampleCode 路徑中的 KEIL 文件夾，雙擊 *ISR_In_SRAM.uvproj*。
2. 進入編譯模式界面
   a. 編譯
   b. 下載代碼至內存
3. 打開串口工具選擇對應串口
   a. 按下複位鍵後查看串口打印信息
   b. 使用邏輯分析儀抓取串口通信波形

6 修訂紀錄

Date	Revision	Description
2021.07.16	1.00	1. 初始發佈.

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.