Read Bin File Data from SD Card to IAP

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This code demonstrates how to execute IAP process by reading bin file data from SD card to update user application.
BSP Version	NUC230/240 Series BSP CMSIS V3.01.001
Hardware	NuTiny-SDK-NUC240V V1.2 NuEdu-Advance02

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 Function Description

1.1 Introduction

This example code demonstrates how to execute IAP process by reading bin file data from SD card to update user application. If there is an update bin file inside the SD card, this example code will update user application. After finishing the IAP process, update process will remap the start address of new program to vector page address and call system reset function to run the new program. By changing different bin file data inside the SD card, user can easily update the program.

This example code provides two test program, one toggles LED at 1 Hz, another one toggles LED at 10 Hz. User can copy the bin file to SD card and rename it as “update.bin” to test the IAP process.

1.2 Principle

IAP (In-Application Programming) allows user to update the program when system is still in operation. User can use IAP to update the program for those devices, which cannot stop the operation when updating the program. There are several methods to get update file, for example, reading data from the SD card to update current user application.

When user enables CBS[0] (Config0[6]) value to 0, IAP function will be enabled. With IAP function, user can locate program anywhere in APROM or LDOM rather than the lowest address. By using the Vector Page Remap command, user can change the vector page address of user application to address 0x00000000. After system rest or CPU reset, MCU will execute the new user application.

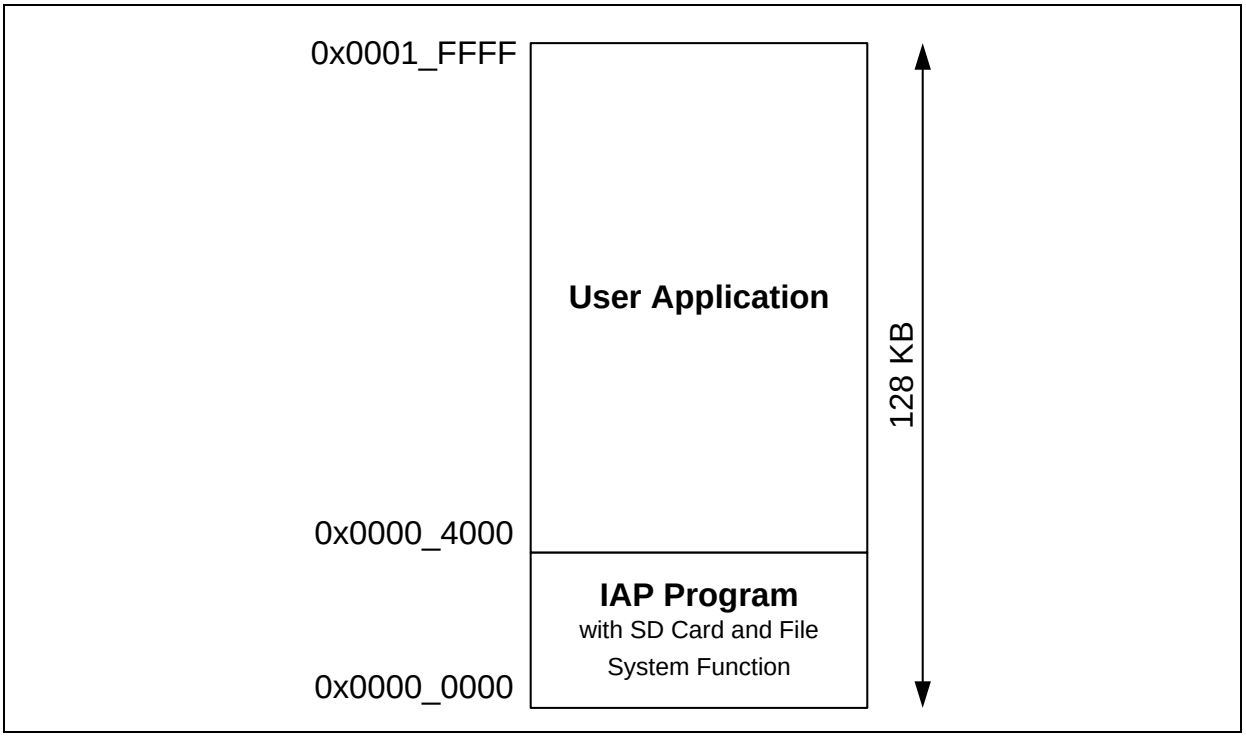
In this example code, IAP program with SD card and file system function is located to address 0x00000000, and user application is located to address 0x00004000. After power on or nRESET pin reset, MCU will execute IAP program first.

The IAP program checks if there is a SD card and if there is an update bin file named “update.bin” inside SD card. If there is an “update.bin” file inside SD card, IAP program will use the content data to update user application at address 0x00004000. After update is finish, IAP program uses the Vector Page Remap command to remap address 0x00004000 to vector page address 0x00000000, and call system reset function to run the new program.

If there is no SD card nor an update bin file inside SD card, IAP program will directly use the

Vector Page Remap command to remap 0x00004000 to vector page address 0x00000000, and call system reset function to run the current user application.

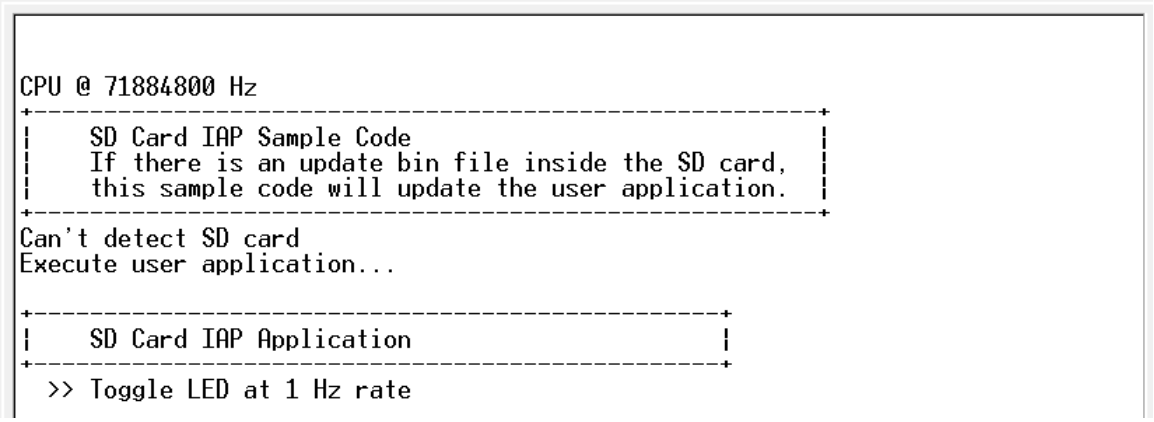
The flash management for this example code is shown below picture:



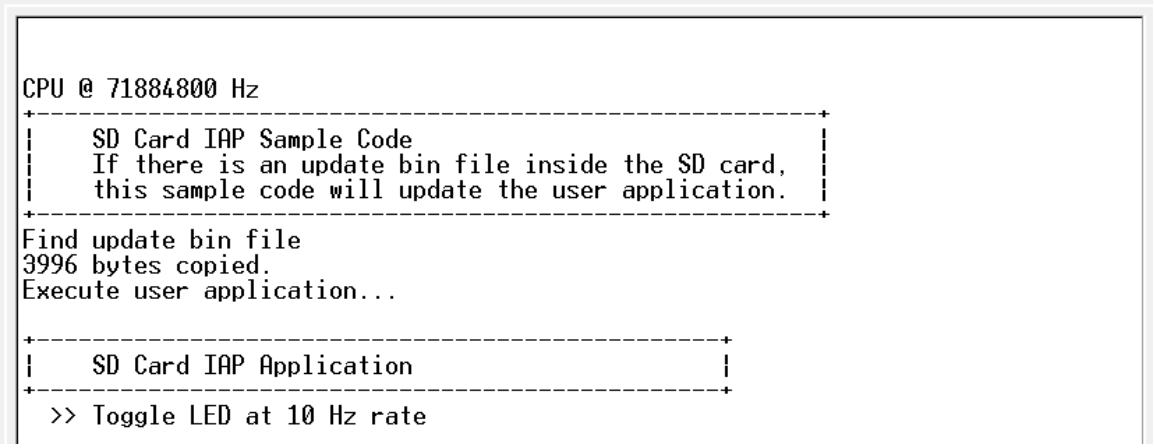
To read SD card, NUC240 uses SPI2 (PD.0, PD.1, PD.2 and PD.3) to connect with the SD card connector on NuEdu-Advance02.

1.3 Demo Result

If there is no SD card nor an update bin file inside SD card, this example code directly remaps the address 0x00004000 to vector page address 0x00000000, and uses system reset to execute the current user application. The result is shown below picture:



If there is an “update.bin” inside SD card, this example code reads the content data to update the user application. After update is finish, this example code remaps the address 0x00004000 to vector page address 0x00000000, and uses system reset to execute the new user application. The result is shown below picture:



2 Code Description

Unlock protected registers and enable FMC ISP function and APROM update.

```
    /* Unlock protected registers */
    SYS_UnlockReg();

    /* Enable FMC ISP function and APROM update */
    FMC_Open();
    FMC_EnableAPUpdate();
```

Initialize the SD card, mount file system and open update bin file. If one of above actions cannot be finish, jump to label “Boot_APROM”.

```
    /* Initialize the SD card */
    res = (FRESULT)disk_initialize(0);

    if (res)
    {
        //put_rc(res);
        printf("Can't detect SD card\n");
        goto Boot_APROM;
    }

    /* Mount file system */
    res = f_mount(0, &FatFs[0]);

    if (res)
    {
       //put_rc(res);
        printf("Can't mount file system\n");
        goto Boot_APROM;
    }

    /* Open update bin file */
    res = f_open(&file, "0:\\update.bin", FA_READ);

    if (res)
    {
        //put_rc(res);
        printf("Can't open file\n");
        goto Boot_APROM;
```

```
    }

    printf("Find update bin file\n");
```

When opening the update bin file, read data and program to APROM with offset 0x00004000. After the update process is finish, close the update bin file.

```
    while (1)
    {
        /* Read data from update bin file */
        res = f_read(&file, g_au8Buff, sizeof(g_au8Buff), &g_u32ReadSize);

        if (g_u32ReadSize > 0)
        {
            /* Program one page data */
            if (FMC_Read_Write_Flash((unsigned int *)g_au8Buff, g_u32Address,
g_u32ReadSize) != 1)
            {
                printf("Program error\n");

                while (1);
            }

            g_u32TotalSize += g_u32ReadSize;
            g_u32Address += g_u32ReadSize;
        }

        if (res || (g_u32ReadSize == 0))
            break;
    }

    printf("%d bytes copied.\n", g_u32TotalSize);
    f_close(&file);
```

When the update process is finish or there is no SD card nor update bin file inside SD card, IAP process remaps 0x00004000 to vector page address 0x00000000 and call system reset function to run the user application.

```
Boot_APROM:
    printf("Execute user application...\n");

    /* Wait UART0 print message done */
    while (!UART_IS_TX_EMPTY(UART0));
```

```
    /* Disable APROM update */
    FMC_DisableAPUpdate();

    /* Disable all interrupts before change VECMAP */
    NVIC->ICER[0] = 0xFFFFFFFF;

    /* Set vector table of startup AP address */
    FMC_SetVectorPageAddr(APROM_OFFSET);

    /* Reset All IP before boot to new AP */
    SYS->IPRSTC2 = 0xFFFFFFFF;
    SYS->IPRSTC2 = 0;

    /* System reset to execute user application */
    NVIC_SystemReset();

    while (1);
```

3 Software and Hardware Environment

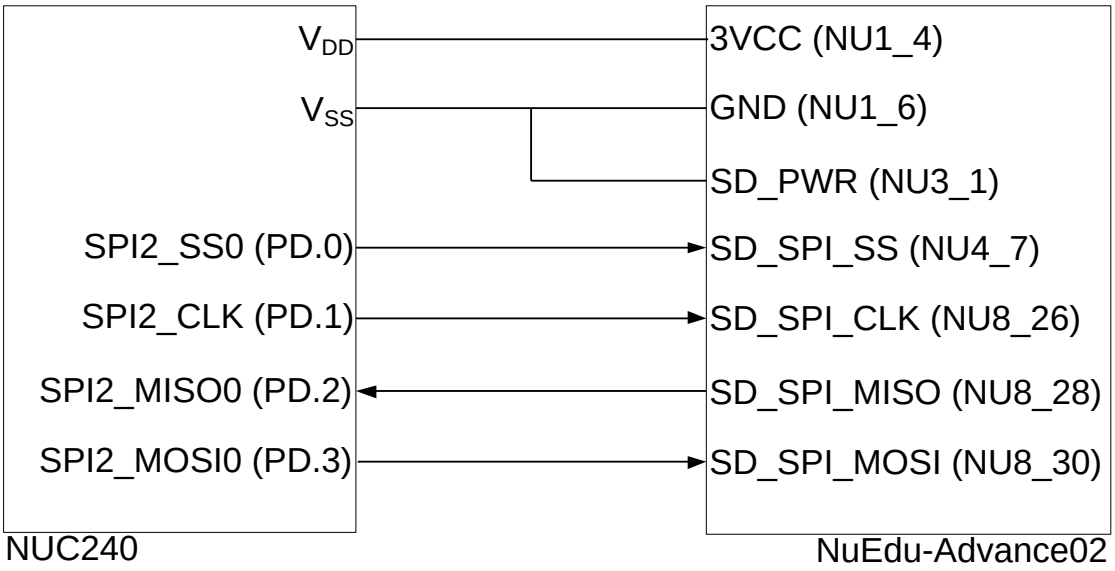
Software Environment

- BSP version
  - NUC230/240 Series BSP CMSIS V3.01.001
- IDE version
  - Keil uVersion 5.26

Hardware Environment

- Circuit components
  - NuTiny-SDK-NUC240V V1.2
  - NuEdu-Advance02
- Diagram

NUC240 uses SPI2 (PD.0, PD.1, PD.2 and PD.3) to connect with the SD card connector on NuEdu-Advance02.



User also needs to configure S1, S2 and S3 on NuEdu-Advance02. User should turn S2 pin 1 to pin 7 to “On” side and turn all the pins of S1 and S3 to another side.

4 Directory Information

EC_NUC240_SDCard_IAP_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
NUC240_SDCard_IAP	IAP program with SD card and file system function
LED_Toggle_1Hz	1st test program which toggles LED at 1 Hz
LED_Toggle_10Hz	2nd test program which toggles LED at 10 Hz

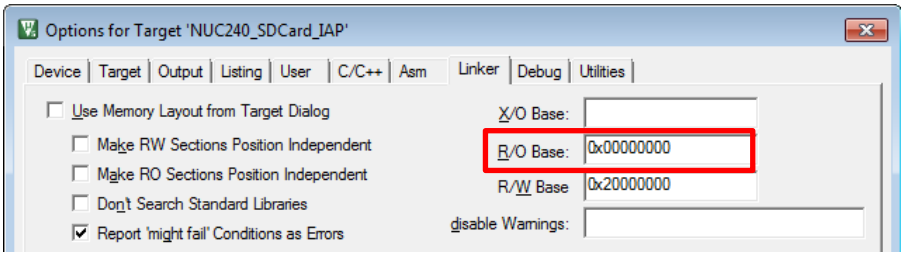
5 How to Execute Example Code

1. Browsing into sample code folder by Directory Information (section 4) and double click NUC240_SDCard_IAP.uvproj.

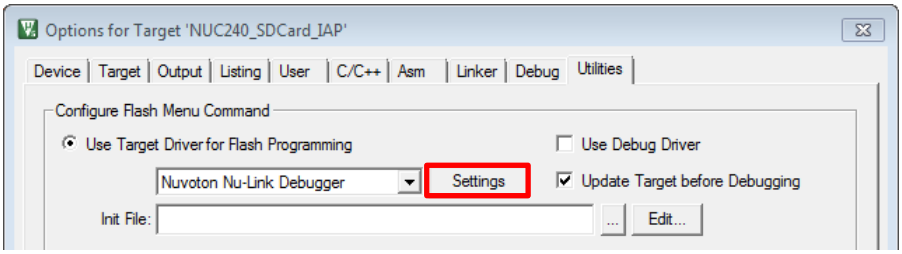
2. Check R/O Base address setting and enable IAP function

   a. Press Alt + F7 to open “Options for Target” setting page

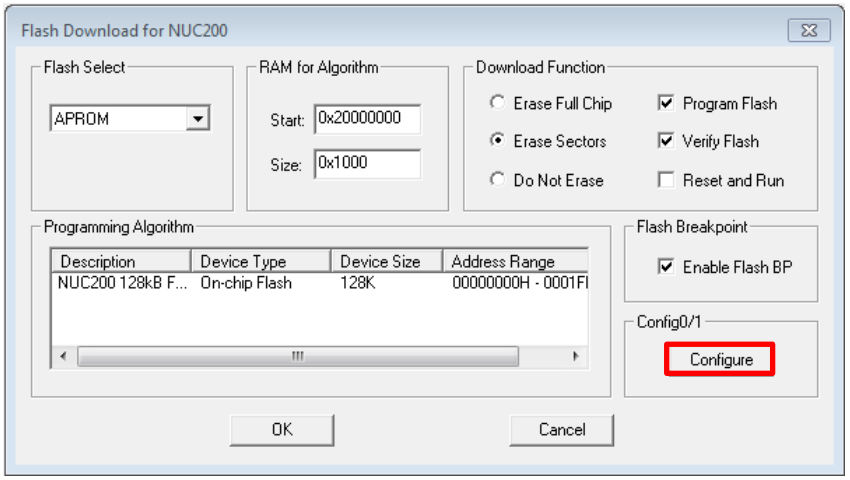
   b. Choose “Linker” dialog and confirm “R/O Base setting” is 0x00000000



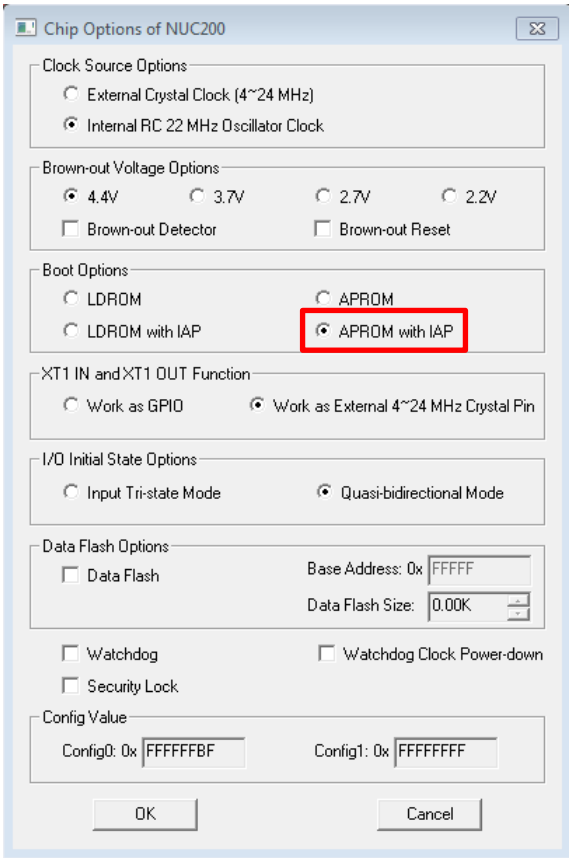
   c. Choose “Utilities” dialog and click “Settings”



   d. Click “Configure”



e. Check “APROM with IAP” to enable IAP function. Then click “OK” to make it effective.



3. Download FW to NUC240

   a. Build

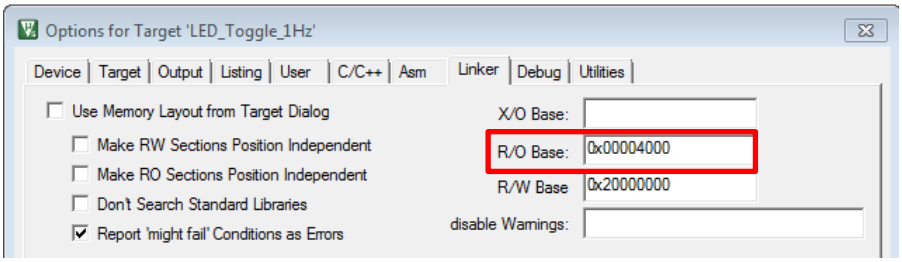
   b. Download

4. Browsing into sample code folder by Directory Information (section 4) and double click LED_Toggle_1Hz.uvproj.

5. Check R/O Base address setting

   a. Press Alt + F7 to open “Options for Target” setting page

b. Choose “Linker” dialog and confirm “R/O Base setting” is 0x00004000



6. Download FW to NUC240

   a. Build

   b. Download

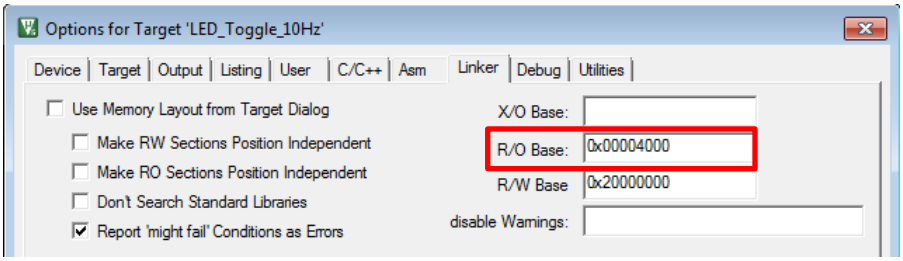
7. Test “no SD card nor an update bin file inside SD card” condition

8. Browsing into sample code folder by Directory Information (section 4) and double click LED_Toggle_10Hz.uvproj.

9. Check R/O Base address setting

   a. Press Alt + F7 to open “Options for Target” setting page

   b. Choose “Linker” dialog and confirm “R/O Base setting” is 0x00004000



10. Build FW and copy to SD card

    a. Build

    b. Copy bin file “LED_Toggle_10Hz.bin” from obj folder to SD card and change its file name to “update.bin”

11. Test “an ‘update.bin’ inside SD card” condition

6 Revision History

Date	Revision	Description
Aug 15, 2019	1.00	1. Initially issued.

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.