Bootloader with IAP

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This code uses memory layout and enables IAP function for bootloader to manage memory distribution when bootloader size is larger than LDROM size.
BSP Version	M031 Series BSP CMSIS v3.01.000
Hardware	NuMaker-M032LD V1.0

1 Function Description

1.1 Introduction

This example code uses memory layout and enables IAP function for bootloader to manage memory distribution when LDROM size is smaller than bootloader size.

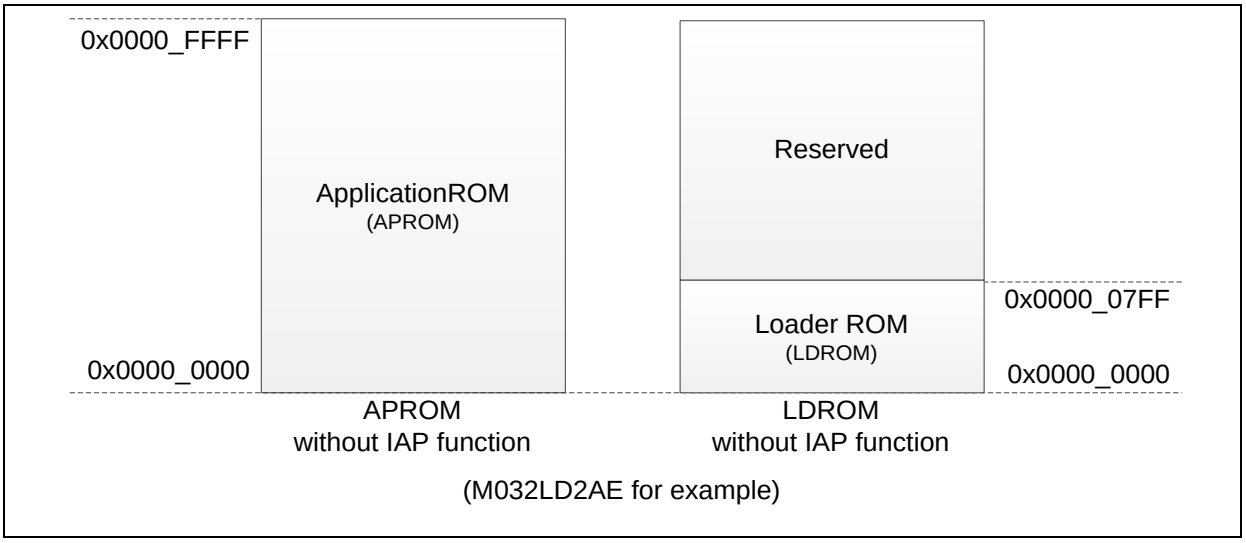
User can use bootloader to update current application program in APROM through several common communication interfaces, such as UART, USB and so on. User can also add some additional actions to perfect the update process, like checksum confirmation and encryption/decryption for update file. These actions will enlarge the bootloader size and the size may be over the LDROM size. If the bootloader size is larger than the LDROM size, user can use memory layout and enables IAP function to manage memory distribution in APROM and LDROM.

This example code uses bootloader for NuMaker-M032LD through USB interface and manages the memory distribution to LDROM and the last 4 KB of APROM.

1.2 Principle

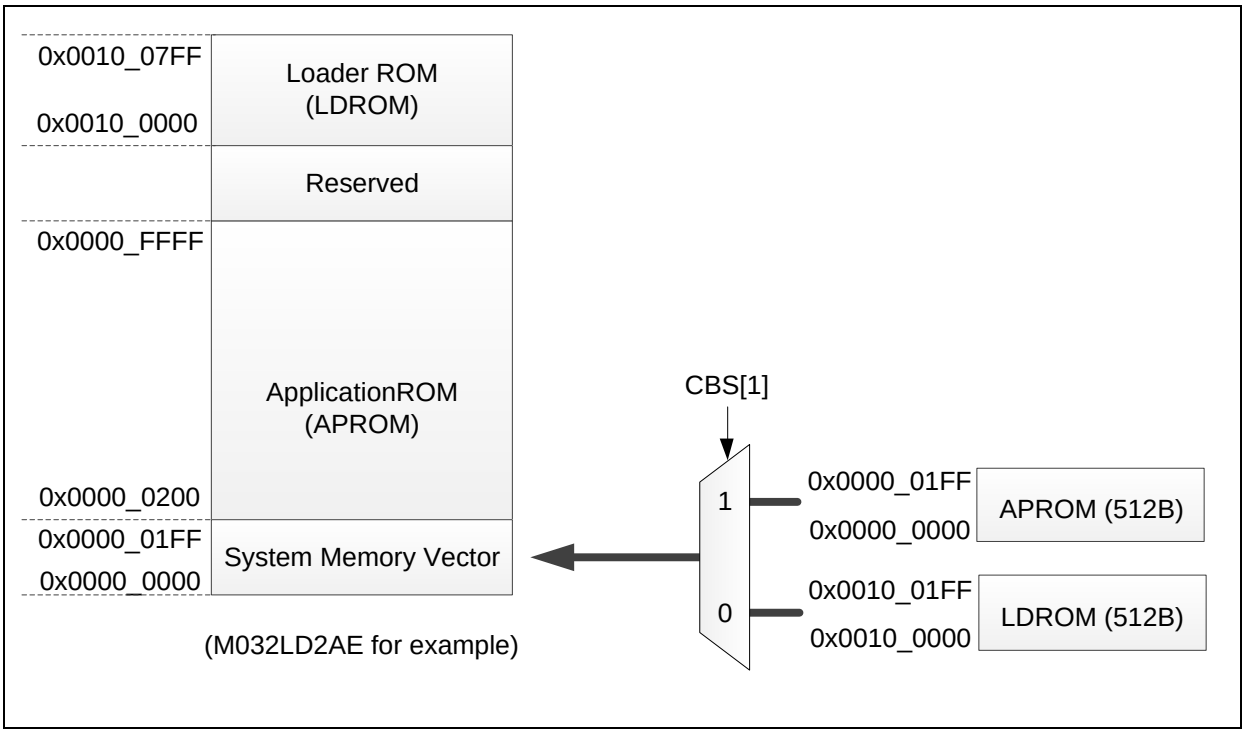
The system memory map is used by CPU to fetch code or data from flash memory map based on chip boot selection CBS (Config0[7:6]). CBS[0] decides whether the IAP function is enabled and CBS[1] decides the boot area.

If CBS[0] = 0, the IAP function is disabled, and FMC only maps LDROM or APROM to the system memory based on CBS[1]. The system memory map without IAP function is shown below:



Without IAP function, CPU can't fetch code in different area. If the boot selection is LDROM, CPU can't fetch the code in APROM.

If CBS[0] = 0, the IAP function is enabled, and FMC maps both LDROM and APROM to the system memory. The first page of system memory map is vector page from LDROM or APROM based on CBS[1]. The system memory map with IAP function is shown below:



With IAP function, CPU can fetch code in different area. If bootloader size is larger than LDROM size, user can enable IAP function and distribute parts of bootloader to APROM.

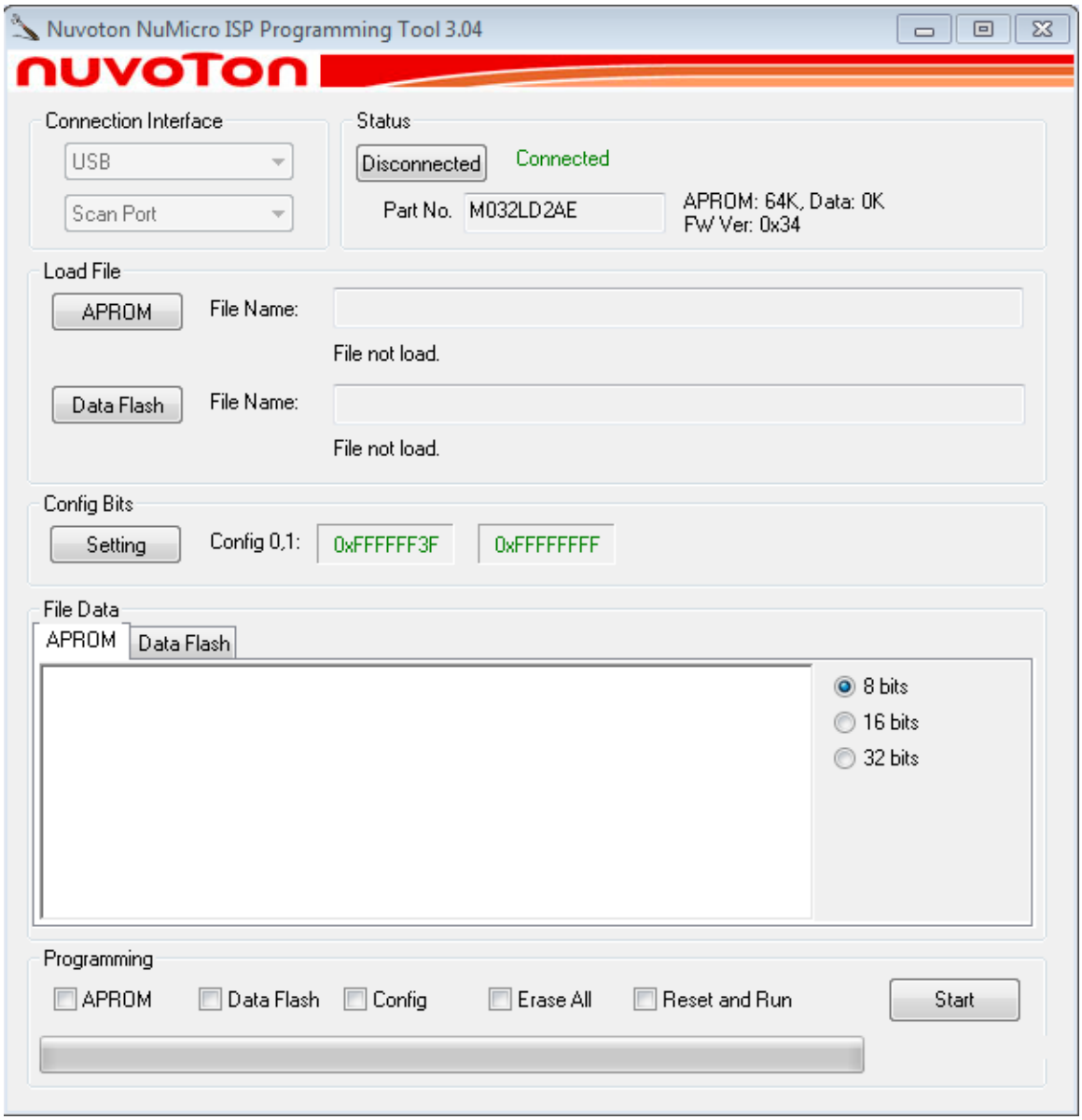
In Keil RVMDK, user can configure the linker setting in "Options for Target" setting page. By enabling "Use Memory Layout from Target Dialog" in "Linker" dialog, linker creates a scatter file from the "Read/Only Memory Areas" setting in "Target" dialog. User can manage one area in LDROM and another area in APROM to make linker manage the memory distribution. If the bootloader size is larger than LDROM size, linker will distribute to APROM based on the setting. With IAP function, bootloader can execute the code, which is in APROM.

1.3 Demo Result

This example code uses memory layout and enables IAP function for bootloader to manage memory distribution when LDROM size is smaller than bootloader size. User can use the appended ISP Programming Tool "NuvoISP_v3_04.exe" to test update process. To enter

update process, user has to connect PB.12 (pin 44 on NuMaker-M032LD) to VSS before power-on. Then user can connect USB cable between PC and connector J2 on NuMaker-M032LD. Choose interface to USB and click “Connect” button in ISP Programming Tool. After connect, user can do the update process as normal. Since the last 4 KB of APROM is used for bootloader, the update file size should smaller than 60 KB

The result is shown below picture:



2 Code Description

Define function pointer for reset handler, which is used when jumping from bootloader to application. Declare function __set_SP to set stack pointer base address which application uses.

```
/*---------------------------------------------------------------------------*/
/* Define                                                                    */
/*---------------------------------------------------------------------------*/
typedef void (FUNC_PTR)(void);

/*---------------------------------------------------------------------------*/
/* Functions                                                                 */
/*---------------------------------------------------------------------------*/
#ifdef __ARMCC_VERSION
__asm __set_SP(uint32_t _sp)
{
    MSR MSP, r0
    BX lr
}
#endif
```

Check if DetectPin is low as update condition.

```
while (DetectPin == 0)
{
    if (bUsbDataReady == TRUE)
    {
        ParseCmd((uint8_t *)usb_rcvbuf, EP3_MAX_PKT_SIZE);
        EP2_Handler();
        bUsbDataReady = FALSE;
    }
}
```

When jumping to APROM to execute application:

1. Change system memory vector page address to APROM
2. Point to the reset handler address, which is at FMC_APROM_BASE + 4.
3. Set stack pointer base address, which is at FMC_APROM_BASE.
4. Execute reset handler.

```
_APROM:
    /* Change vector page address to APROM */
```

```
    FMC_SetVectorPageAddr(FMC_APROM_BASE);
    /* Point to the reset handler address which application uses */
    func = (FUNC_PTR *)*(uint32_t *)(FMC_APROM_BASE + 4);
    /* Set stack pointer base address to the one which application uses */
    __set_SP(*(uint32_t *)FMC_APROM_BASE);
    func();

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

3 Software and Hardware Environment

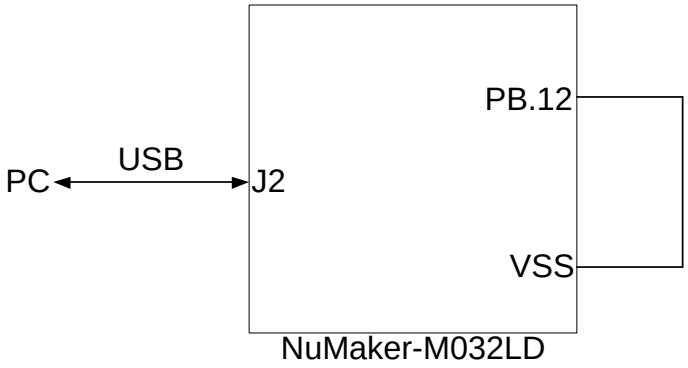
Software Environment

- BSP version
  - M031 Series BSP CMSIS v3.01.000
- IDE version
  - Keil uVersion 5.26

Hardware Environment

- Circuit components
  - NuMaker-M032LD V1.0
- Diagram

This example code uses PB.12 as update detect pin. To execute update process, user needs to connect PB.12 (pin 44 on NuMaker-M032LD) to VSS. Then connect USB cable between PC and connector J2.



4 Directory Information

EC_M031_Bootloader_w_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
ISP Programming Tool	ISP Programming Tool to test update process

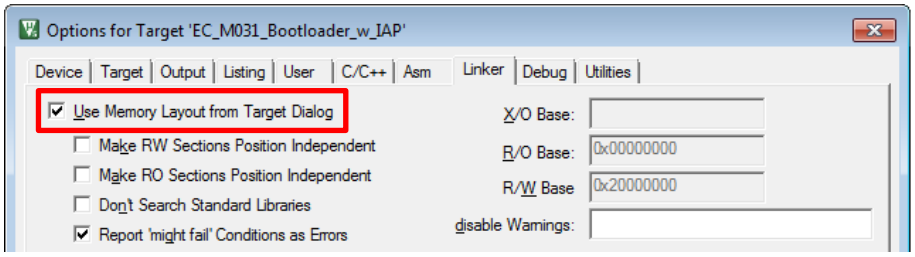
5 How to Execute Example Code

1. Browsing into sample code folder by Directory Information (section 4) and double click EC_M031_Bootloader_w_IAP_V1.00.uvproj.

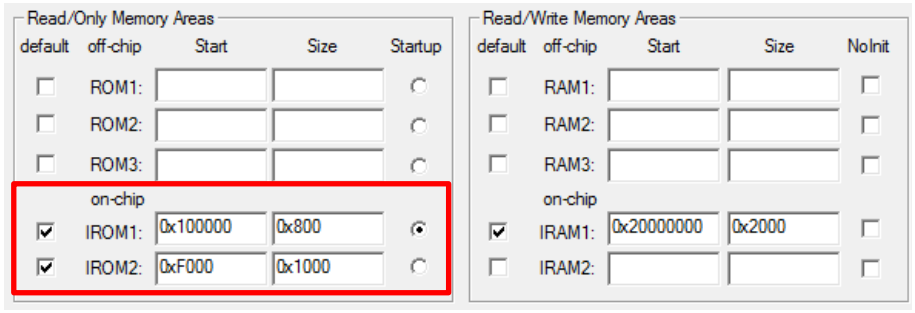
2. Use memory layout and enable IAP function

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

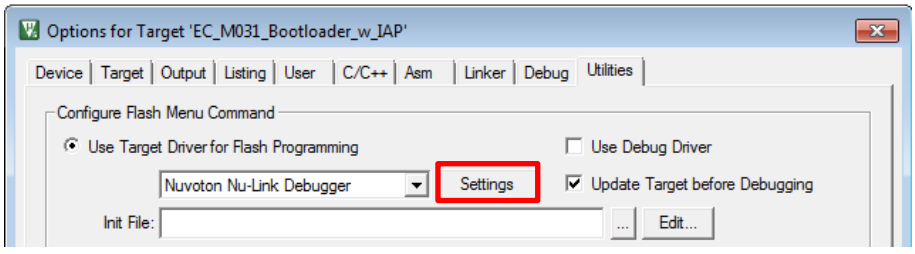
   b. Choose “Linker” dialog and enable “Use Memory Layout from Target Dialog”



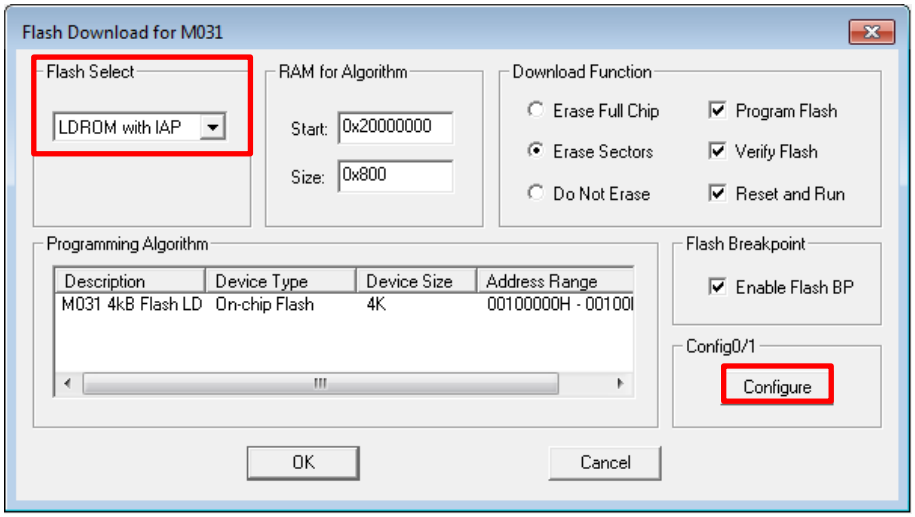
   c. Choose “Target” dialog and configure on-chip IROM1 and IROM2 in “Read/Only Memory Areas”



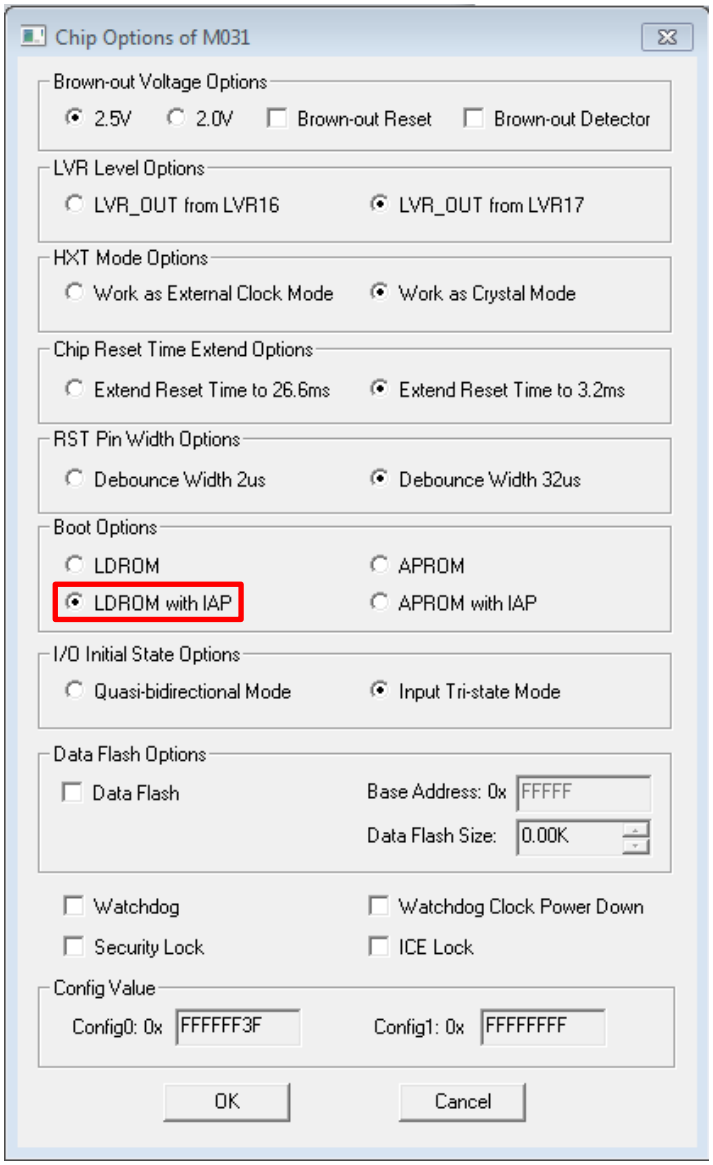
   d. Choose “Utilities” dialog and click “Settings”



   e. Change “Flash Select” to “LDROM with IAP” and click “Configure”



f. Check “LDROM with IAP” to enable IAP function. Then click “OK” to make it effective.



3. Download FW to NuMaker-M032LD
    a. Build
    b. Download
4. Connect PB.12 and VSS.
5. Connect USB cable between PC and connector J2.
6. Execute ISP Programming Tool “NuvoISP_v3_04.exe” and test the update process.

6 Revision History

Date	Revision	Description
Oct. 10, 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.