USBD MSD ISP

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	The Mass Storage (MSD) ISP supports to download firmware to APROM through mass storage interface. There is no any tool or driver installation required before downloading firmware.
BSP Version	M451 Series BSP CMSIS V3.01.003
Hardware	NuTiny-EVB-M451-LQFP100 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 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

Mass Storage ISP is one In-System Programming methodology to support download firmware to APROM of NuMicro® Family through mass-storage interface. By using mass-storage interface, we don't need to provide any tool or driver for PC to support firmware download. For a device which supports Mass Storage ISP, user just needs to plug-in the device by USB cable. Then PC will recognize it as a mass-storage disk and user just need to "COPY" a binary firmware image to the mass-storage disk to do "firmware download".

1.2 Demo Result

On Power-On, the Mass Storage ISP will enter Application Upgrade mode if the specified GPIO pin is low. Then, PC can detect a mass storage class device when the NuMicro® MCU is plugged into PC USB port.

Then, we drag or copy-and-paste the binary file of upgraded application code into window of the mass storage device.

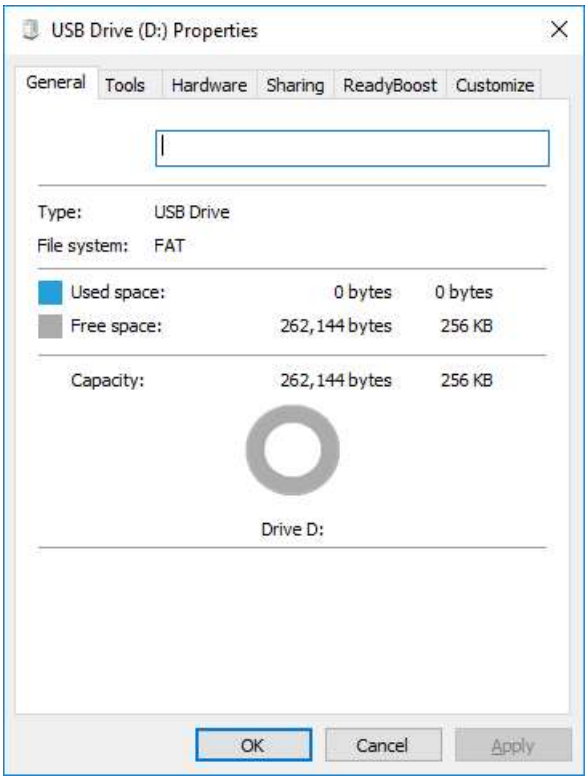


Figure 1 MSD ISP Device

2 Code Description

2.1 Mass Storage ISP Booting Flow

The Mass Storage ISP program will enter application upgrade mode or reset to boot from APROM according to the status of the specified GPIO pin. In the sample code, GPA0 is used to control Mass Storage ISP to reset to APROM or do mass-storage (application upgrade mode).

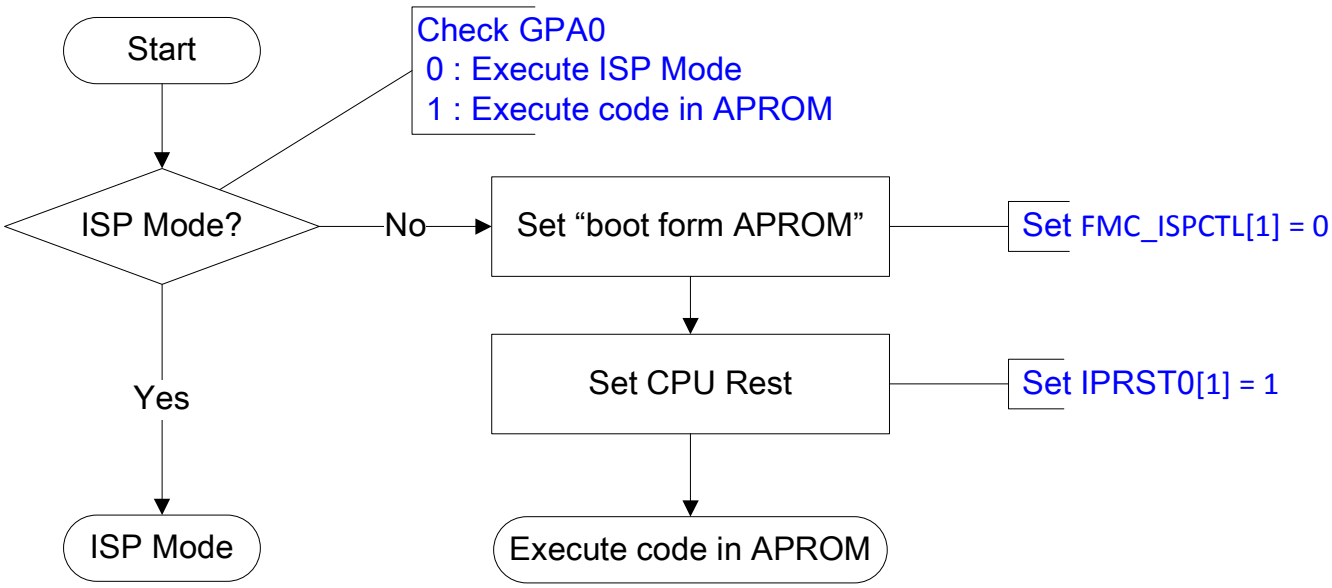


Figure 2 Boot Flow

2.2 Mass Storage ISP Main Flow

When entering application upgrade mode, the Mass Storage ISP program will use APROM size (256KB) for updating FAT table and configure NuMicro® MCU as mass storage class device. After all initial steps have been done successfully; it is ready to upgrade new application code transferred from Host.

After upgrade operation has been completed, User needs to set GPA0 as high and reset the NuMicro® MCU to execute the new firmware in APROM.

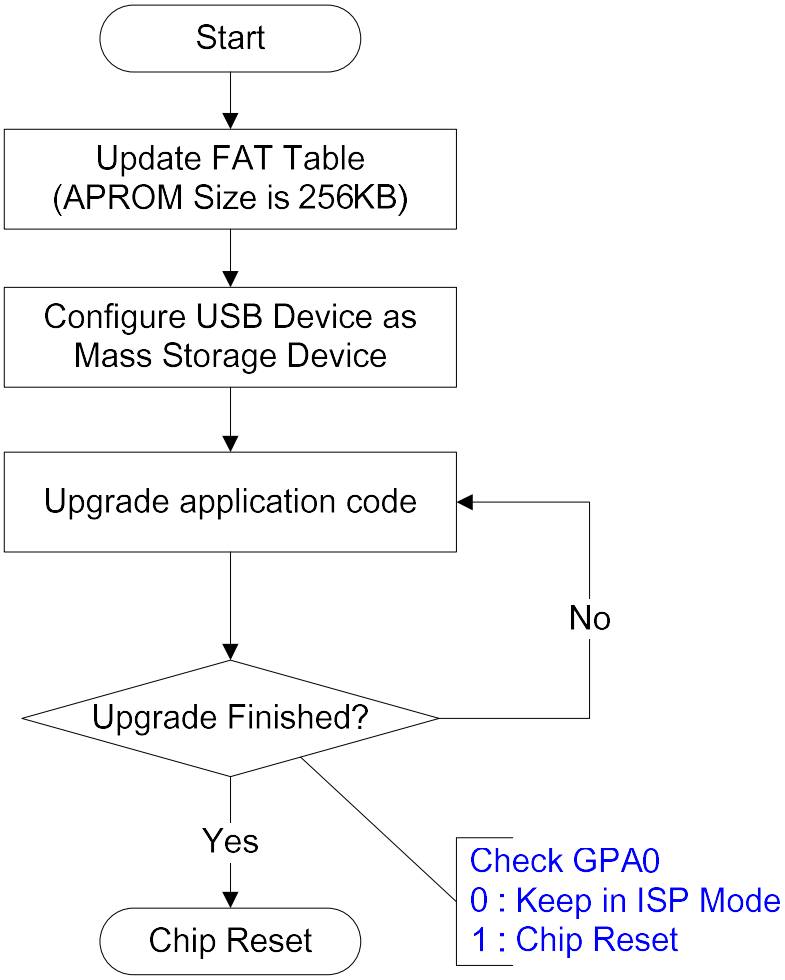


Figure 3 Mass Storage ISP Main Flow

2.3 USB MSD Endpoint Setting

This example code includes 1 interfaces that follow MSD (mass storage device) specification. Based on USB spec, endpoint 0 for control transfer is used for USB enumeration. There are 2 endpoints to upload and download Bulk report. In this example, control pipe (endpoint 0) is made of two hardware endpoints 0 and 1 in device. The interfaces and endpoints configuration are shown in Table 1.

Function	USB Composite Device	M451 Hardware setting
Control Transfer	Control Pipe endpoint 0: Control IN/OUT	endpoint 0: Control IN endpoint 1: Control OUT
Mass storage device	Interface 0 endpoint 2: Bulk IN endpoint 3: Bulk OUT	endpoint 2: Bulk IN endpoint 3 Bulk OUT

Table 1 Configuration of MSD interfaces and endpoints

2.4 FAT and Flash Program function

The DataFlashRead function is used to response the FAT table to PC and the DataFlashWrite() is used to program the new firmware into APROM.

```
void DataFlashRead(uint32_t u32Addr, uint32_t * u32Buf)
{
    USBD_MemReset((uint8_t *)u32Buf, STORAGE_BUFFER_SIZE);

    if (u32Addr == 0x00000000)
    {
        uint32_t u32Sectors = g_u32u32StorageSize / UDC_SECTOR_SIZE;

        u8FormatData[19] = u32Sectors & 0xFF;
        u8FormatData[20] = (uint8_t)(u32Sectors >> 8);

        USBD_MemCopy((uint8_t *)u32Buf, u8FormatData, sizeof(u8FormatData));

        u32Buf[STORAGE_BUFFER_SIZE/4-1] = 0xAA550000;
    }
    else if ((u32Addr == (RSVD_SEC_CNT * BYTE_PER_SEC)) ||
             (u32Addr == ((RSVD_SEC_CNT + FAT_SZ) * BYTE_PER_SEC)))
    {
        u32Buf[0] = 0x00FFFFF8;
    }
    else if (u32Addr == ((RSVD_SEC_CNT + FAT_SZ * NUM_FAT) * BYTE_PER_SEC)) /* root dir */
    {
        USBD_MemCopy((uint8_t *)u32Buf, u8RootDirData, sizeof(u8RootDirData));
    }
}
```

```
void DataFlashWrite(uint32_t u32Addr, uint32_t * u32Buf)
{
    /* This is low level write function of USB Mass Storage */
    if ((u32Addr >= DATA_SEC_ADDR) && (u32Addr < g_u32u32StorageSize))
    {
        u32Addr -= DATA_SEC_ADDR;

        if (g_u8LockBit)
        {
            if ((u32Addr & (FMC_FLASH_PAGE_SIZE - 1)) == 0)
                FMC_ISP(FMC_ISPCMD_PAGE_ERASE, u32Addr, 0);
        }
        /* For Security Lock */
        else if (u32Addr == 0)
        {
            uint32_t i;

            for (i = 0; i < (g_u32u32StorageSize - DATA_SEC_ADDR);
                  i += FMC_FLASH_PAGE_SIZE)
                FMC_ISP(FMC_ISPCMD_PAGE_ERASE, i, 0);
        }
        DataFlashProgramPage(u32Addr, u32Buf);
    }
}
```

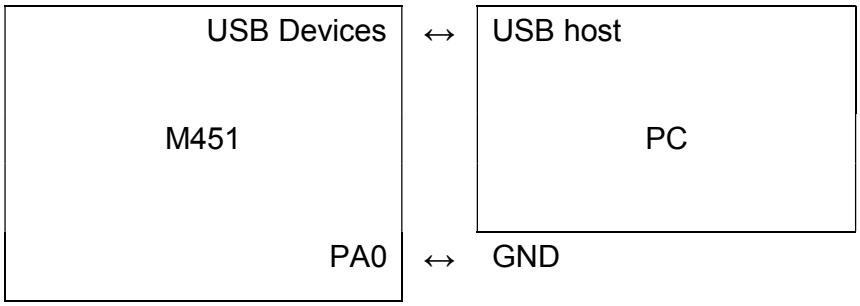
3 Software and Hardware Environment

Software Environment

- BSP version
  - M451 Series BSP CMSIS v3.01.003
- IDE version
  - Keil uVersion 5.26

Hardware Environment

- Circuit components
  - NuTiny-EVB-M451-LQFP100 V1.3
  - USB mini USB cable
  - M451 PA0 connect to GND
- Diagram



4 Directory Information

- EC_M451_USBD_MSD_ISP_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

5 How to Execute Example Code

1. This project supports Keil uVersion 5.26 or above.
2. Browsing into sample code folder by Directory Information (section 4) and double click USBD_MSD_ISP.uvproj.
3. Enter Keil compile mode
   a. Build
   b. Download
   c. Start/Stop debug session
4. Enter debug mode
   a. Run

6 Revision History

Date	Revision	Description
Nov. 01 , 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.