SPI flash as a storage for USB MSC

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This sample code uses SPI Flash as back end storage through SPIM interface
BSP Version	M480 Series BSP CMSIS V3.04.000
Hardware	NuMaker-PFM-M487 Ver 3.0

1 Function Description

1.1 Introduction

In the sample code, the M480 acts as a USB 2.0 flash drive. It uses the SPIM interface as an back end storage for SPI Flash and has a file system. This function can be used as an alternative to the SD card as a storage space. The write speed is approximately 128 bytes per second.

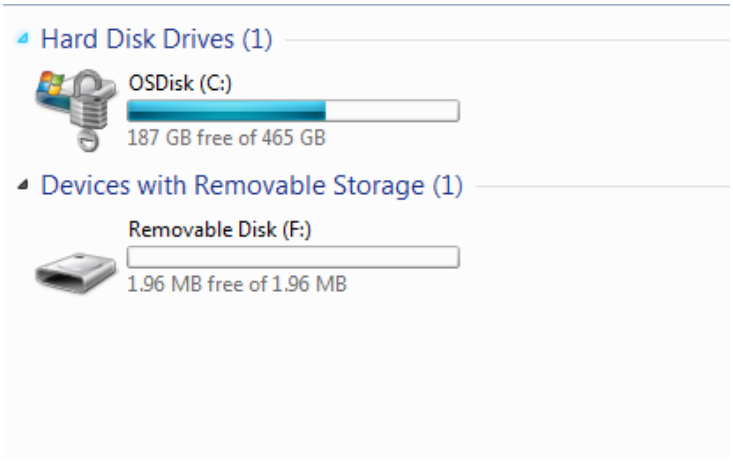
The user must set the storage memory starting position and the total memory space size through MSC_Init(SectorOffset, TotalSector). A Sector is 512 KB. Since the SPI Flash limit requires at least 4KB to be cleared at a time, 4KB of content is read first, and deleted and then rewritten. The USB handler is in MSC_ProcessCmd(), and M480 will perform the corresponding processing according to the received command.

1.2 Demo Result

After execution, the following information will be printed.

```
+------------------------------------------------------------+
|             HSUSB Mass Storage Sample Code                 |
|  use SPI flash as back end storage through SPIM interface  |
+------------------------------------------------------------+
disk size is : 2048 KB
SPIM get JEDEC ID=0xEF, 0x40, 0x16
```

After connecting the USB2.0 slot on the NuMaker-PFM-M487 to the computer, the following icon appears



2 Code Description

First, do initialize in main.c and read the external SPI flash ID, then set the storage location and storage size of the flash drive, and finally wait for the connection to the computer and execute the USB processing program in MSC_ProcessCmd():

```
int32_t main(void)
{
    /* Init System, IP clock and multi-function I/O
       In the end of SYS_Init() will issue SYS_LockReg()
       to lock protected register. If user want to write
       protected register, please issue SYS_UnlockReg()
       to unlock protected register if necessary */
    SYS_Init();

    /* Init UART to 115200-8n1 for print message */
    UART_Open(UART0, 115200);

    printf("+------------------------------------------------------------+\n");
    printf("|             HSUSB Mass Storage Sample Code                 |\n");
    printf("|  use SPI flash as back end storage through SPIM interface |\n");
    printf("+------------------------------------------------------------+\n");
    printf("disk size is : %d KB\n", (disk_size*512)/1024);

    SYS_UnlockReg();

    /* SPIM init */
    SPIM_Init();

    HSUSBD_Open(&gsHSInfo, MSC_ClassRequest, NULL);

    /* Massstorage init */
    MSC_Init(memory_offset, disk_size);

    /* Enable USBD interrupt */
    NVIC_EnableIRQ(USBD20_IRQn);

    /* Start transaction */
    while (1) {
        if (HSUSBD_IS_ATTACHED()) {
```

```
            HSUSBD_Start();
            break;
        }
    }
    /* Massstorage process */
    while (1) {
        if (g_hsusbd_Configured)
            MSC_ProcessCmd();
    }
}
```

Read and write SPI flash code as follows:

```
/* Read data through SPIM */
void MSC_ReadMedia(uint32_t addr, uint32_t size, uint8_t *buffer)
{
    SPIM_Read(addr + g_SectorsOffset, size, buffer);
}
/* Write data through SPIM */
void MSC_WriteMedia(uint32_t addr, uint32_t size, uint8_t *buffer)
{
    SPIM_Write(addr + g_SectorsOffset, size, buffer);
}
```

3 Software and Hardware Environment

Software environment

- BSP version
  - M480 Series BSP CMSIS V3.04.000
- IDE version
  - Keil uVersion 5.26

Hardware environment

- Circuit components
  - NuMaker-PFM-M487 or other M480 Development Board
- Diagram

The SPI flash is connected through the SPIM interface, and the transmission protocol is processed in a hardware to improve the read/write speed of the SPI flash. Use the USB2.0 OTG slot to connect to computer and you will see the disk on your computer.

SPIM_MOSI (PC.0)	↔	IO0
SPIM_MISO (PC.1)	↔	IO1
SPIM_CLK (PC.2)		CLK
SPIM_SS (PC.3)		/CS
SPIM_D2 (PC.5)		IO2
SPIM_D3 (PC.4)		IO3
M480		W25Q32

4 Directory Information

- EC_M480_ HSUSBD_MassStorage_SPIMFlash_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. Browsing into sample code folder by Directory Information (section 4) and double click HSUSBD_MassStorage_SPIMFlash.uvproj
2. Enter Keil compile mode
   a. Build
   b. Download
   c. Start/Stop debug session
3. Enter debug mode
   a. Run

6 Revision History

Date	Revision	Description
Oct. 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.