

Access SPI Flash by USB MSD

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	The code is a demonstration of how to use USB MSD to access file in SPI flash
BSP Version	NUC230_240 Series BSP CMSIS v3.01.001
Hardware	NuEdu-EVB-NUC240 v2.0

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

USB MSD is an abbreviation of USB Mass storage device. We use USB MSD to access file in SPI flash. In this way, we access the SPI flash data as easily as accessing the U disk.

1.2 Principle

SPI Flash is connected to SPI2 (PD0-PD3) using W25Q16, and the USB port of NUC240 is connected to PC.

After executing the program, the NUC240 is listed as a USB DISK, and the user can directly perform file operations on the SPI Flash.

1.3 Demo Result



2 Code Description

The user is mainly concerned with the SPI interface when using this code. The code in the USB part is basically standardized and does not need to be modified. If you want to change to an SPI interface, you need to modify the following code.

```

#define SPI_Flash SPI2
#define Check_SPI_BUSY          while(SPI_IS_BUSY(SPI_Flash))
#define SPI_SetBitLength(n)     SPI_SET_DATA_WIDTH(SPI_Flash,8*(n))
#define SPI_WRITE(n)            SPI_WRITE_TX0(SPI_Flash, n)
#define SPI_READ                SPI_READ_RX0(SPI_Flash)
#define SS_HIGH                 SPI_SET_SS0_HIGH(SPI_Flash)
#define SS_LOW                  SPI_SET_SS0_LOW(SPI_Flash)
#define SPI_GO                   SPI_TRIGGER(SPI_Flash)

void SpiInit(void)
{
    /* Setup SPI2 multi-function pins */
    SYS->GPD_MFP |= SYS_GPD_MFP_PD0_SPI2_SS0 | SYS_GPD_MFP_PD1_SPI2_CLK |
        SYS_GPD_MFP_PD2_SPI2_MISO0 | SYS_GPD_MFP_PD3_SPI2_MOSI0;

    /* Enable SPI2 IP clock */
    CLK->APBCLK |= CLK_APBCLK_SPI2_EN_Msk;
    /* Init GPIO for SPI2 */
    /* Select HCLK as the clock source of SPI2 */
    CLK_SetModuleClock(SPI2_MODULE, CLK_CLKSEL1_SPI2_S_HCLK, MODULE_NoMsk);
    /* Enable SPI2 peripheral clock */
    CLK_EnableModuleClock(SPI2_MODULE);

    /* Configure SPI2 as a master, SPI clock rate 24 MHz,
    clock idle low, 32-bit transaction, drive output on falling clock edge and latch
    input on rising edge. */
    SPI_Open(SPI_Flash, SPI_MASTER, SPI_MODE_0, 32, 12000000);

    /* Disable the automatic hardware slave select function. Select the SS pin and
    configure as low-active. */
    SPI_DisableAutoSS(SPI_Flash);
    SPI_SET_SS0_HIGH(SPI_Flash);
}
    
```

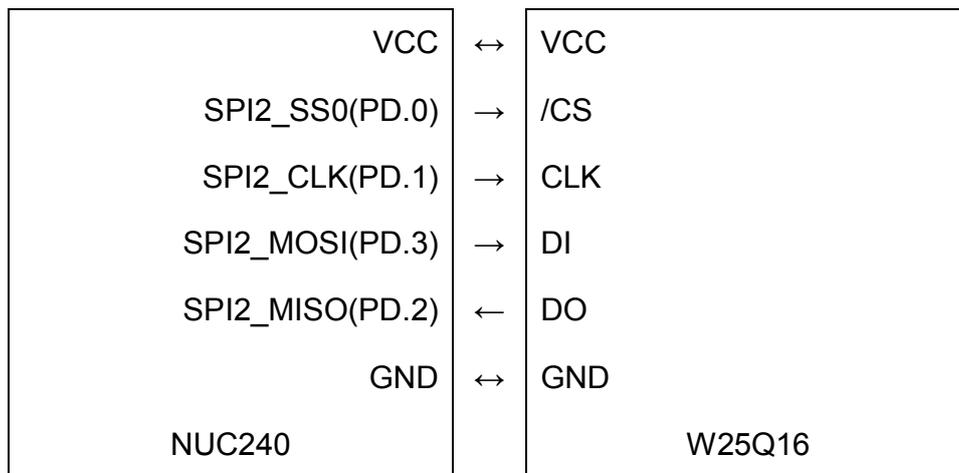
3 Software and Hardware Environment

- **Software Environment**

- BSP version
 - ◆ NUC230_240 Series BSP CMSIS v3.01.001
- IDE version
 - ◆ Keil uVersion 5.26

- **Hardware Environment**

- Circuit components
 - ◆ NuEdu-EVB-NUC240
- Diagram



4 Directory Information

📁 EC_NUC240_USBD_MSD_SPIFlash_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 NUC240_USBD_MSD_SPIFlash.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. 29, 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.*