NUC240 SPI FATFS

NuMicro® 32 位系列微控制器範例代碼介紹

文件資訊

代碼簡述	示範如何使用 FatFs 存取 SPI Flash
BSP 版本	NUC230/NUC240 Series BSP CMSIS v3.01.002
開發平台	NuTiny-EVB-NUC240 V1.2

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 功能介紹

1.1 簡介

本範例將SPI Flash規劃成一個儲存裝置。為了在存取SPI Flash時管理文件的資料結構，本範例使用FatFs來做文件系統的管理。

1.2 原理

1.2.1 FatFs

FatFs是一個為小型嵌入式系統設計的通用FAT(File Allocation Table)文件系統模組。FatFs模組的編寫遵照ANSI C，且其與磁盤I/O層是完全分開的，因此，其可完全獨立於硬件平台。此外，FatFs模組可以被嵌入至有限資源的小型微控制器，像是8051、ARM等等，而不需做任何修改。

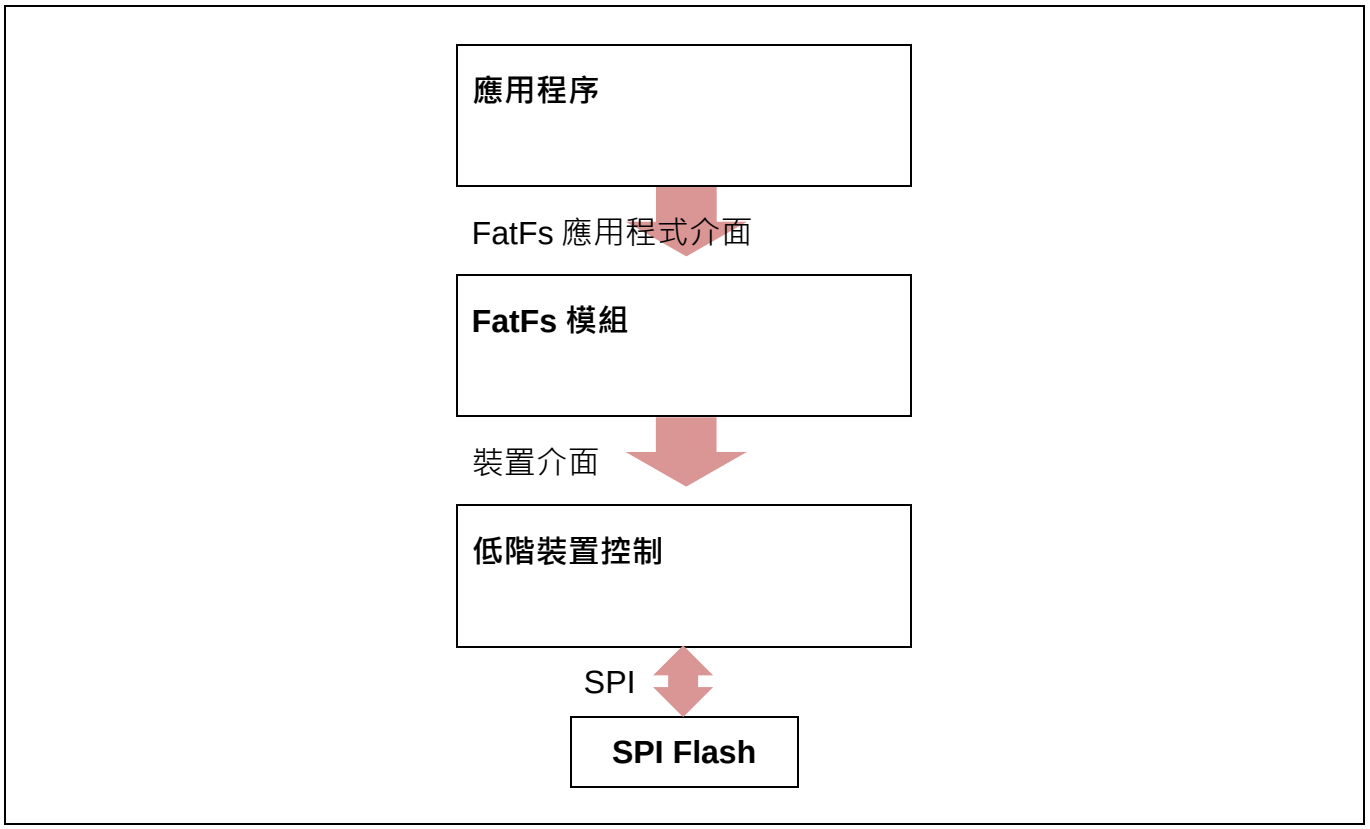

圖 1-1 FatFs 控制方塊圖

FatFs控制方塊圖如圖 1-1所示。FatFs應用程式介面為應用程序提供各種文件系統功能，在本

範例中所使用到的應用程式介面被列在表 1。

FatFs應用程式介面	描述
f_mount	註冊/註銷一個工作區
f_mkfs	在邏輯驅動器上建立一個FAT系統
f_opendir	打開一個目錄
f_readdir	讀取目錄裡的檔案及資訊
f_getfree	回傳系統全部空間及剩餘空間

表 1 在範例程式中使用的應用程序介面

因為FatFs模組是獨立於平台與儲存媒體的文件系統層，故其可與像是SPI Flash的實體裝置完全分開。如同圖 1-1所示，低階裝置控制模組並非FatFs模組的任一部份，因此，本範例需實現裝置介面以給FatFs存取SPI Flash使用。表 2列出了本範例所提供的媒體存取介面及其功能描述。如想得知更多關於應用程序介面與媒體存取介面的資訊，請參考FatFs的官方網站：http://elm-chan.org/fsw/ff/00index_e.html。

裝置介面	描述
disk_status	回傳裝置的狀態
disk_initialize	初始化裝置
disk_read	讀取扇區
disk_write	寫入扇區
disk_ioctl	控制裝置相關的功能

表 2 在範例程式中實現的媒體存取介面

1.2.2 SPI Flash

本範例使用W25X16A串行快閃記憶體當作儲存裝置。W25X16A是通過一個SPI兼容的總線來進行存取，該總線由四個訊號組成，包括：串行時鐘(Serial Clock, CLK)、芯片選擇(Chip Select, /CS)、串行數據輸入(Serial Data Input, DI)，以及串行數據輸出(Serial Data Output, DO)。其引腳配置圖如圖 1-2所示。

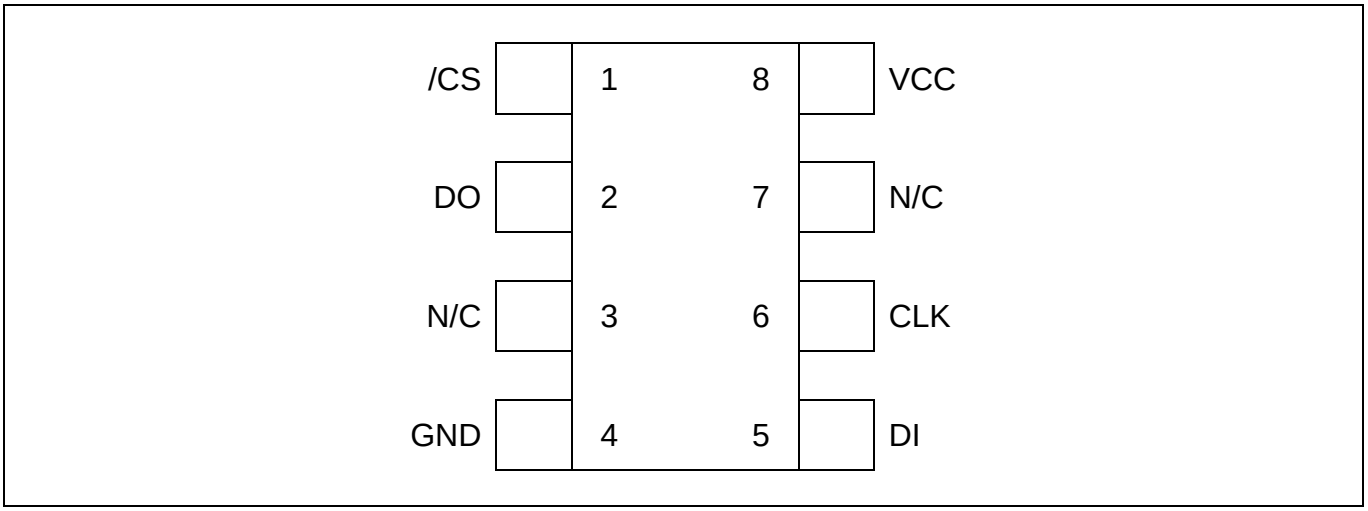

圖 1-2 W25X16A 引腳配置圖

標準的SPI指令使用DI輸入引腳在CLK的上升緣將指令、位址或數據串行寫入裝置，而DO輸出引腳則用來在CLK的下降緣讀出裝置的數據或者狀態。表 3列出了本範例所使用到的W25X16A指令。此外，製造商標識與裝置標識列於表 4中。

指令名稱	命令	描述
DRVSPIFLASH_WRITE_ENABLE	0x06	致能寫入
DRVSPIFLASH_WRITE_DISABLE	0x04	禁能寫入
DRVSPIFLASH_READ_STATUS	0x05	讀取狀態暫存器 - 1
DRVSPIFLASH_FAST_READ	0x0B	快速讀取
DRVSPIFLASH_PAGE_PROGRAM	0x02	頁面編程
DRVSPIFLASH_SECTOR_ERASE	0x20	扇區擦除 (4KB)
DRVSPIFLASH_DEVICE_ID	0x90	製造商/設備 標識

表 3 在範例程式中使用的指令表

製造商標識	
Winbond Serial Flash	0xEF
設備標識	
W25X16A	0x14

表 4 製造商標識與設備標識

1.3 執行結果

以下執行結果會輸出於UART鮑率設定為115200的終端機視窗。

當“FF_USE_MKFS”被定義為1時，文件系統將會被格式化。圖 1-3為格式化文件系統後的目錄信息，使用者可以看到在SPI Flash中有多少可用的儲存空間。

圖 1-3 沒有文件在內之目錄信息

當使用者在SPI Flash內儲存文件後，包括文件屬性、修改日期、修改時間、大小、名稱以及剩餘多少可用空間的目錄信息將會被列出，如圖 1-4所示。

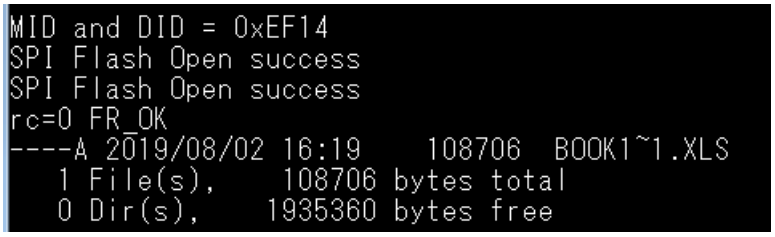

圖 1-4 有文件在內之目錄信息

2 代碼介紹

初始化 SPI 及配置 SPI 多功能引腳：

```
    /* Setup SPI0 multi-function pins */
    SYS->GPC_MFP &= ~(SYS_GPC_MFP_PC0_Msk | SYS_GPC_MFP_PC1_Msk | SYS_GPC_MFP_PC2_Msk |
                      SYS_GPC_MFP_PC3_Msk);
    SYS->GPC_MFP |= SYS_GPC_MFP_PC0_SPI0_SS0 | SYS_GPC_MFP_PC1_SPI0_CLK |
                    SYS_GPC_MFP_PC2_SPI0_MISO0 | SYS_GPC_MFP_PC3_SPI0_MOSI0;
    SYS->ALT_MFP &= ~(SYS_ALT_MFP_PC0_Msk | SYS_ALT_MFP_PC1_Msk | SYS_ALT_MFP_PC2_Msk |
                      SYS_ALT_MFP_PC3_Msk);
    SYS->ALT_MFP |= SYS_ALT_MFP_PC0_SPI0_SS0 | SYS_ALT_MFP_PC1_SPI0_CLK |
                    SYS_ALT_MFP_PC2_SPI0_MISO0 | SYS_ALT_MFP_PC3_SPI0_MOSI0;


    /* Select HCLK as the clock source of SPI0 */
    CLK_SetModuleClock(SPI0_MODULE, CLK_CLKSEL1_SPI0_S_HCLK, MODULE_NoMsk);


    /* Enable SPI0 peripheral clock */
    CLK_EnableModuleClock(SPI0_MODULE);


    /* Configure SPI0 as a master, SPI clock rate 12 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 automatic slave selection function and set SPIn_SS0 pin to high state. */
    SPI_DisableAutoSS(SPI_Flash);
    SPI_SET_SS0_HIGH(SPI_Flash);
```

讀取 SPI Flash 的製造商標識與設備標識來確認 SPI Flash 是否已開放成功：

```
    /* Read MID & DID */
    u32SpiMidDid = SpiReadMidDid();


    if ((u32SpiMidDid & 0xFFFF) == 0xEF14)
    {
        stat = RES_OK;
        printf("SPI Flash Open success\n");
    }
    else
    {
        stat = STA_NOINIT;
        printf("SPI Flash Open failed\n");
```

```
    }
```

註冊一個邏輯驅動器：

```
    /* Mount a logical drive */
    res = f_mount(&FatFs[0], ptr, 1);

    if (res)
    {
        put_rc(res);
        printf("Don't mount file system\n");
    }
```

建立一個 FAT 系統，使用者可以在 ffconf.h 定義“FF_USE_MKFS”為 1 來格式化文件系統：

```
#if (FF_USE_MKFS)
    /* Create a FAT volume */
    res = f_mkfs(ptr, FM_FAT, 0, Buff, sizeof(Buff));

    if (res)
    {
        put_rc(res);
        printf("Don't format\n");
    }

#endif
```

列出目錄信息，使用者可以從 UART 日誌獲得此信息：

```
    /* List directory information */
    put_rc(f_opendir(&dir, ptr));
    p1 = s1 = s2 = 0;

    for (;;)
    {
        /* Read directory entries in sequence */
        res = f_readdir(&dir, &Finfo);

        if ((res != FR_OK) || !Finfo.fname[0])
            break;
```

```
        if (Finfo.fattrib & AM_DIR)
        {
            s2++;
        }
        else
        {
            s1++;
            p1 += Finfo.fsize;
        }

        printf("%c%c%c%c%c %u/%02u/%02u %02u:%02u %9lu  %s\n",
               (Finfo.fattrib & AM_DIR) ? 'D' : '-',
               (Finfo.fattrib & AM_RDO) ? 'R' : '-',
               (Finfo.fattrib & AM_HID) ? 'H' : '-',
               (Finfo.fattrib & AM_SYS) ? 'S' : '-',
               (Finfo.fattrib & AM_ARC) ? 'A' : '-',
               (Finfo.fdate >> 9) + 1980, (Finfo.fdate >> 5) & 15, Finfo.fdate & 31,
               (Finfo.ftime >> 11), (Finfo.ftime >> 5) & 63, Finfo.fsize, Finfo.fname);
    }

    printf("%4u File(s),%10u bytes total\n%4u Dir(s)", s1, p1, s2);

    /* Get number of free clusters */
    if (f_getfree(ptr, (DWORD *)&p1, &fs) == FR_OK)
        printf(", %10u bytes free\n", p1 * fs->csize * 512);
```

3 軟體與硬體環境

● 軟體環境

- ■ BSP 版本
  - ◆ NUC230/NUC240 Series BSP CMSIS v3.01.002
- ■ IDE 版本
  - ◆ Keil uVision V5.26

● 硬體環境

- ■ 電路元件
  - ◆ NuTiny-EVB-NUC240
  - ◆ SPI Flash (W25X16A)
- ■ 示意圖
  - ◆ 如圖 3-1 所示，分別將 NUC240 的引腳：SPI0_SS0 (PC.0)、SPI0_CLK (PC.1)、SPI0_MISO0 (PC.2)、SPI0_MOSI0 (PC.3)與 W25X16A 的引腳：/CS、CLK、DO、DI 進行連接以藉由 SPI 存取 SPI Flash。

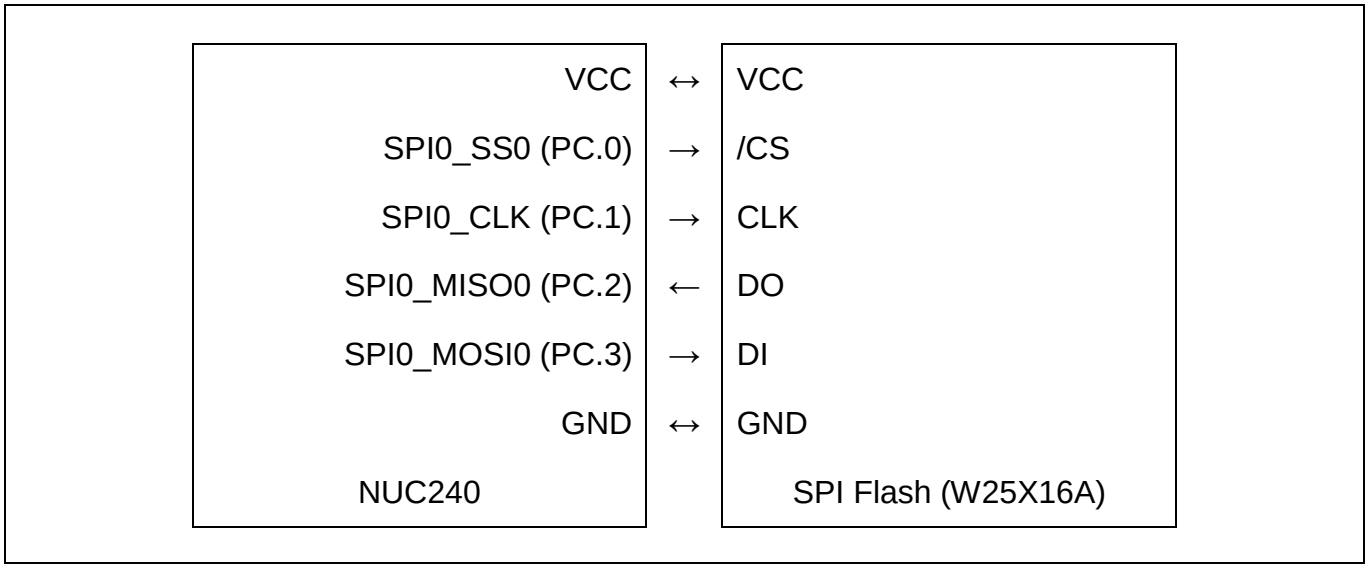

圖 3-1 NUC240 與 W25X16A 引腳接線圖

4 目錄資訊

- EC_NUC240_SPI_FATFS_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
  - ThirdParty
    - FatFs — A generic FAT file system module for small embedded systems. Its official website is: http://elm-chan.org/fsw/ff/00index_e.html.

5 如何執行範例程式

1. 根據目錄資訊章節進入 ExampleCode 路徑中的 KEIL 資料夾，雙擊 NUC240_SPI_FATFS.uvproj。
2. 進入編譯模式介面
   a. 編譯
   b. 下載代碼至記憶體
   c. 進入 / 離開除錯模式
3. 進入除錯模式介面
   a. 執行代碼

6 修訂紀錄

Date	Revision	Description
Aug. 8, 2019	1.00	1. 初始發布.

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.