移植FATFS文件系統存取SD Card

NuMicro® 32 位系列微控制器範例代碼介紹

文件資訊

代碼簡述	本範例代碼移植 FATFS 文件檔案系統，來存取 SD 卡
BSP 版本	NUC230/240 Series BSP v3.01.002
開發平台	NUC240VE3AN NuEdu Board V2.0 + NuEdu-Advance02 V1.1

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

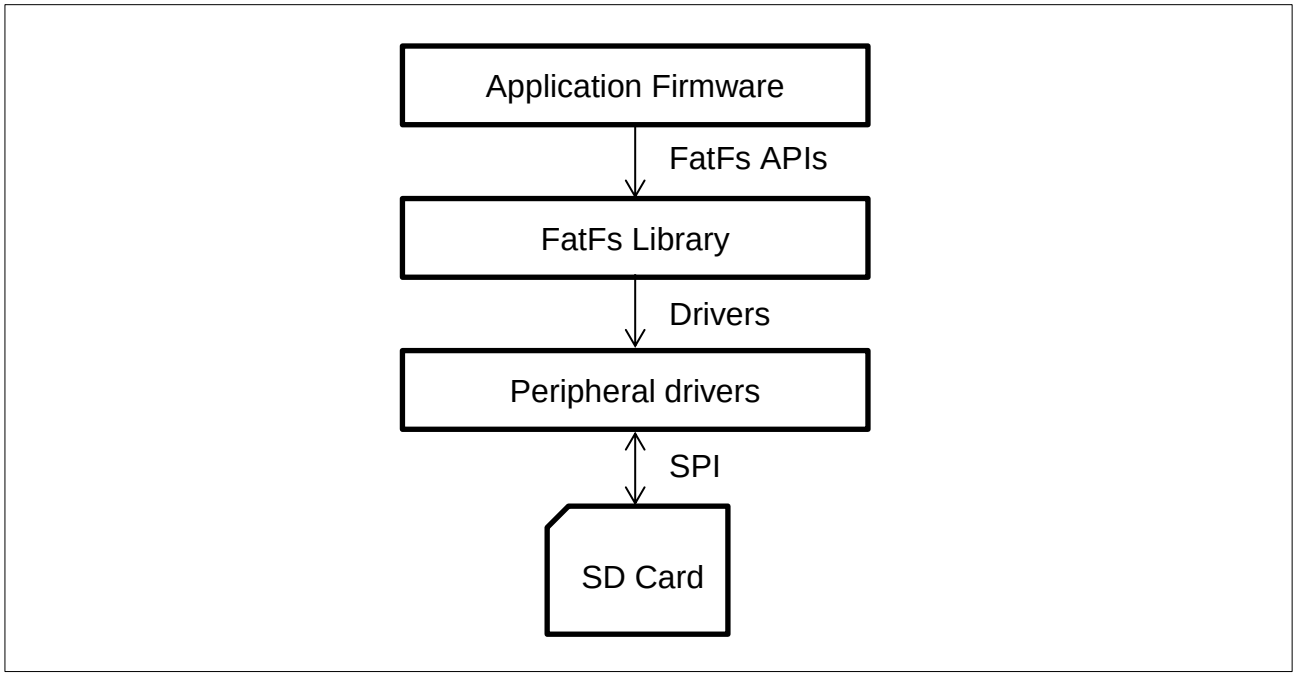
1 功能介紹

1.1 簡介

SD 記憶卡系統支援兩種通訊模式：SD 模式和 SPI 模式。此範例使用 SPI 模式和 SD 卡通訊，和讀寫 SD 卡 FAT 文件系統上的檔案。SD 卡的文件系統層是採用第三方的 FatFs 函式庫，而 FatFs 為一個專為小型嵌入式系統設計的通用 FAT 文件系統模組。

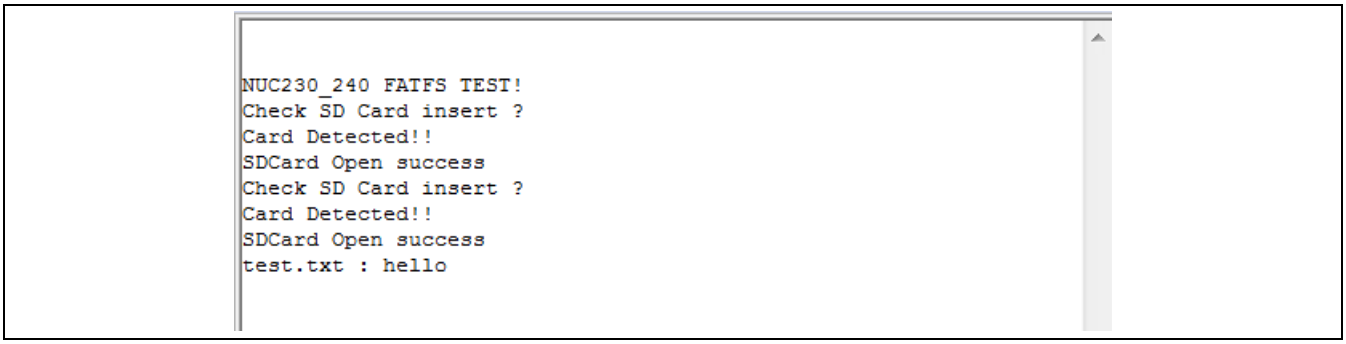
1.2 原理

SD 卡透過 SPI 模式存取的架構顯示如下圖：基本的存取流程為應用端韌體呼叫 FatFs APIs 讀寫檔案，接著 FatFs APIs 呼叫 SPI 驅動介面來和 SD 卡接收和傳送資料。



1.3 執行結果

範例執行結果，可進入Keil MDK 偵錯模式，打開UART#1視窗，即可看到輸出訊息如下。



2 代碼介紹

FatFs 函式庫提供文件系統的函式，下表列出此範例會使用到的函式如下：

FUNCTION NAME	FUNCTION DESCRIPTION
f_open	Open or create a file
f_close	Close an open file object
f_read	Read data from the file
f_write	Write data to the file
f_lseek	Move file pointer of the file object
f_mount	Mount/Unmount a logical drive

有五個 FatFs 介面程式需要實作，其檔案在 EC_NUC230_240_FATFS_SDCard_V1.0\SampleCode\ExampleCode\NUC230_240_FATFS_SDCard\diskio.c。

FUNCTION NAME	FUNCTION DESCRIPTION
disk_initialize	Initialize device
disk_status	Get device status
disk_read	Read sector(s)
disk_write	Write sector(s)
disk_ioctl	Control device dependent functions

有些 FatFs 介面程式需要呼叫 SD 卡的驅動程式來存取 SD 卡，如下表所列。這些 SD 卡驅動程式以 SPI 實作在 sdcard.c 檔案，sdcard.c 的路徑位於 EC_NUC230_240_FATFS_SDCard_V1.0\SampleCode\ExampleCode\NUC230_240_FATFS_SDCard 目錄下。

FUNCTION NAME	FUNCTION DESCRIPTION
SDCARD_Open	Open GPIO function and initial SD card
SDCARD_Close	Close SD card
MMC_Command_Exec	Send SDCARD CMD and Receive Response
GetLogicSector	Get logic sector size
SDCARD_GetCardSize	Get card total sector after SDCARD is opened
SpiRead	Get data from SD card
SpiWrite	Store data into SD card

範例程式如下所示。在 SYS_Init() 中開啟晶振和 PLL，打開 UART0 和 SPI1 模組並選擇其時鐘源，設置所有使用腳位的 MFP 功能。

設置腳位的 MFP 功能如下：

- PB.0 設置為 UART0_RXD
- PB.1 設置為 UART0_TXD
- PC.9 設置為 SPI1_CLK
- PC.10 設置為 SPI1_MISO
- PC.11 設置為 SPI1_MOSI

```
void SYS_Init(void)
{
    /* Unlock protected registers */
    SYS_UnlockReg();

    /*---------------------------------------------------------------------------------------------------------*/
    /* Init System Clock                                                                                       */
    /*---------------------------------------------------------------------------------------------------------*/

    /* Enable Internal RC 22.1184MHz clock */
    CLK_EnableXtalRC(CLK_PWRCON_OSC22M_EN_Msk);

    /* Waiting for Internal RC 22M clock ready */
    CLK_WaitClockReady(CLK_CLKSTATUS_OSC22M_STB_Msk);

    /* Switch HCLK clock source to Internal RC and HCLK source divide 1 */
    CLK_SetHCLK(CLK_CLKSEL0_HCLK_S_HIRC, CLK_CLKDIV_HCLK(1));

    /* Enable external XTAL 12MHz clock */
    CLK_EnableXtalRC(CLK_PWRCON_XTL12M_EN_Msk);

    /* Waiting for external XTAL clock ready */
    CLK_WaitClockReady(CLK_CLKSTATUS_XTL12M_STB_Msk);

    /* Set core clock as PLL_CLOCK from PLL */
    CLK_SetCoreClock(PLL_CLOCK);
```

```
    /* Enable UART module clock */
    CLK_EnableModuleClock(UART0_MODULE);

    /* Select UART module clock source */
    CLK_SetModuleClock(UART0_MODULE, CLK_CLKSEL1_UART_S_PLL, CLK_CLKDIV_UART(1));

    /* Enable SPI1 peripheral clock */
    CLK_EnableModuleClock(SPI1_MODULE);

    /* Select HCLK as the clock source of SPI1 */
    CLK_SetModuleClock(SPI1_MODULE, CLK_CLKSEL1_SPI1_S_HCLK, MODULE_NoMsk);

    /*---------------------------------------------------------------------------------------------------------*/
    /* Init I/O Multi-function                                                                                 */
    /*---------------------------------------------------------------------------------------------------------*/

    /* Set GPB multi-function pins for UART0 RXD and TXD */
    SYS->GPB_MFP &= ~(SYS_GPB_MFP_PB0_Msk | SYS_GPB_MFP_PB1_Msk);
    SYS->GPB_MFP |= SYS_GPB_MFP_PB0_UART0_RXD | SYS_GPB_MFP_PB1_UART0_TXD;

    /* Setup SPI1 multi-function pins */
    #ifdef DEFINE_SS
    SYS->GPC_MFP = SYS_GPC_MFP_PC9_SPI1_CLK | SYS_GPC_MFP_PC10_SPI1_MISO0 |
SYS_GPC_MFP_PC11_SPI1_MOSI0;
    SYS->ALT_MFP = SYS_ALT_MFP_PC9_SPI1_CLK | SYS_ALT_MFP_PC10_SPI1_MISO0 |
SYS_ALT_MFP_PC11_SPI1_MOSI0;
    #else
    SYS->GPC_MFP = SYS_GPC_MFP_PC8_SPI1_SS0 | SYS_GPC_MFP_PC9_SPI1_CLK |
SYS_GPC_MFP_PC10_SPI1_MISO0 | SYS_GPC_MFP_PC11_SPI1_MOSI0;
    SYS->ALT_MFP = SYS_ALT_MFP_PC8_SPI1_SS0 | SYS_ALT_MFP_PC9_SPI1_CLK |
SYS_ALT_MFP_PC10_SPI1_MISO0 | SYS_ALT_MFP_PC11_SPI1_MOSI0;
    #endif

    /* Lock protected registers */
    SYS_LockReg();
}
```

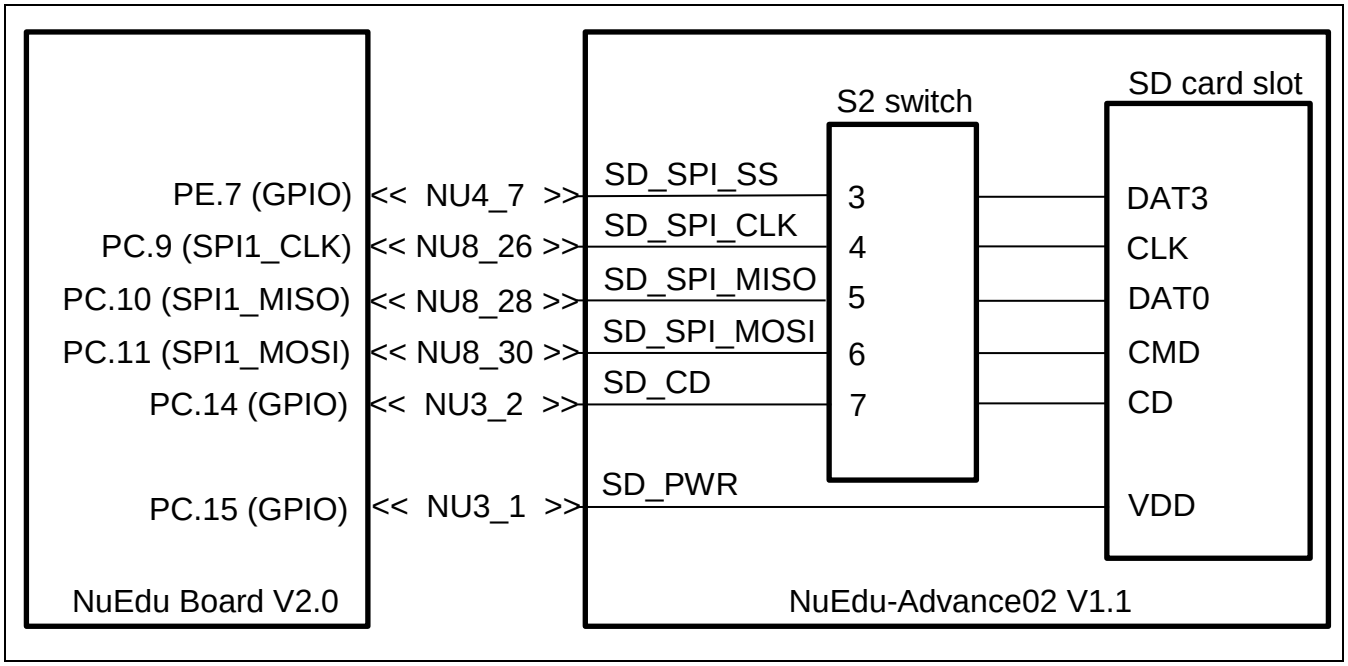
在main()中呼叫disk_initialize()來初始化磁碟、呼叫f_mount()來掛載一個邏輯磁碟、呼叫f_open()來開啟檔案、呼叫f_write()來寫入資料、呼叫f_read()來讀取資料、呼叫f_close()來關閉檔案。

```
int32_t main(void)
{
    uint32_t u32Count;
    FIL file;       /* File objects */
    char achBufRead[512];
    FRESULT res;

    /* Init System, IP clock and multi-function I/O */
    SYS_Init();

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

    printf("\n\nNUC230_240 FATFS TEST!\n");

    /* SD card initializetion */
    if (disk_initialize(0) != RES_OK)
    {
        printf("SDCard initialization failed\n");
    }

    disk_read(0, g_aBuff, 2, 1);

    /* Mount a logical drive */
    res = f_mount(&g_aFatFs[0], g_aSD_Path, 1);
    if (res)
    {
        put_rc(res);
        printf("[ERROR] Mount file system is failed!\n");
        while(1);
    }

    /* Open test.txt file */
    f_open(&file, "test.txt", FA_OPEN_ALWAYS | FA_READ |FA_WRITE);

    /* Move the file pointer to end of file */
    f_lseek(&file, file.obj.objsize);

    /* Write hello text into test.txt */
    f_write(&file, "hello ", 6, &u32Count);
```

```
    /* Move the file pointer to start of file */
    f_lseek(&file, 0);

    /* Read content in test.txt */
    f_read(&file, achBufRead, file.obj.objsize, &u32Count);

    printf("test.txt : %s \n", achBufRead);

    /* Close file */
    f_close(&file);

    while(1);
}
```

3 軟體與硬體環境

- **軟體環境**
  - BSP 版本
    - NUC230/240 Series BSP v3.01.002
  - IDE 版本
    - Keil uVersion 5.26
- **硬體環境**
  - 電路元件
    - NUC240VE3AN NuEdu Board V2.0
    - NuEdu-Advance02 V1.1(SD 卡介面透過 SPI, S2 開關上的 1、2、3、4、5、6、7 設為 ON)
  - 示意圖



- PE.7: GPIO 腳位，設成 output 模式， 為 SPI 的從選擇腳位，連接到 SD 卡槽的 DAT3。
- PC.9: SPI1_CLK 腳位，連接到 SD 卡槽的 CLK。
- PC.10: SPI1_MISO 腳位，連接到 SD 卡槽的 DAT0。
- PC.11: SPI1_MOSI 腳位，連接到 SD 卡槽的 CMD。
- PC.14: GPIO 腳位，設成 input 模式，為卡偵測腳位，連接到 SD 卡槽的 CD。
- PC.15: GPIO 腳位，設成 output 模式，連接到 SD 卡槽的 VDD。

4 目錄資訊

- EC_ NUC230_240_FATFS_SDCard_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 NUC230_240_FATFS_SDCard.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. 21, 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.