NUC240 SPI FATFS

NuMicro® 32 位系列微控制器范例代码介绍

文件信息

代码简述	示范如何使用 FatFs 存取 SPI Flash
BSP 版本	NUC230/NUC240 Series BSP CMSIS v3.01.002
开发平台	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 功能介绍

1.1 简介

本范例将SPI Flash规划成一个储存装置。为了在存取SPI Flash时管理文件的数据结构，本范例使用FatFs来做文件系统的管理。

1.2 原理

1.2.1 FatFs

FatFs是一个为小型嵌入式系统设计的通用FAT(File Allocation Table)文件系统模块。FatFs模块的编写遵照ANSI C，且其与磁盘I/O层是完全分开的，因此，其可完全独立于硬件平台。此外，FatFs模块可以被嵌入至有限资源的小型微控制器，像是8051、ARM等等，而不需做任何修改。

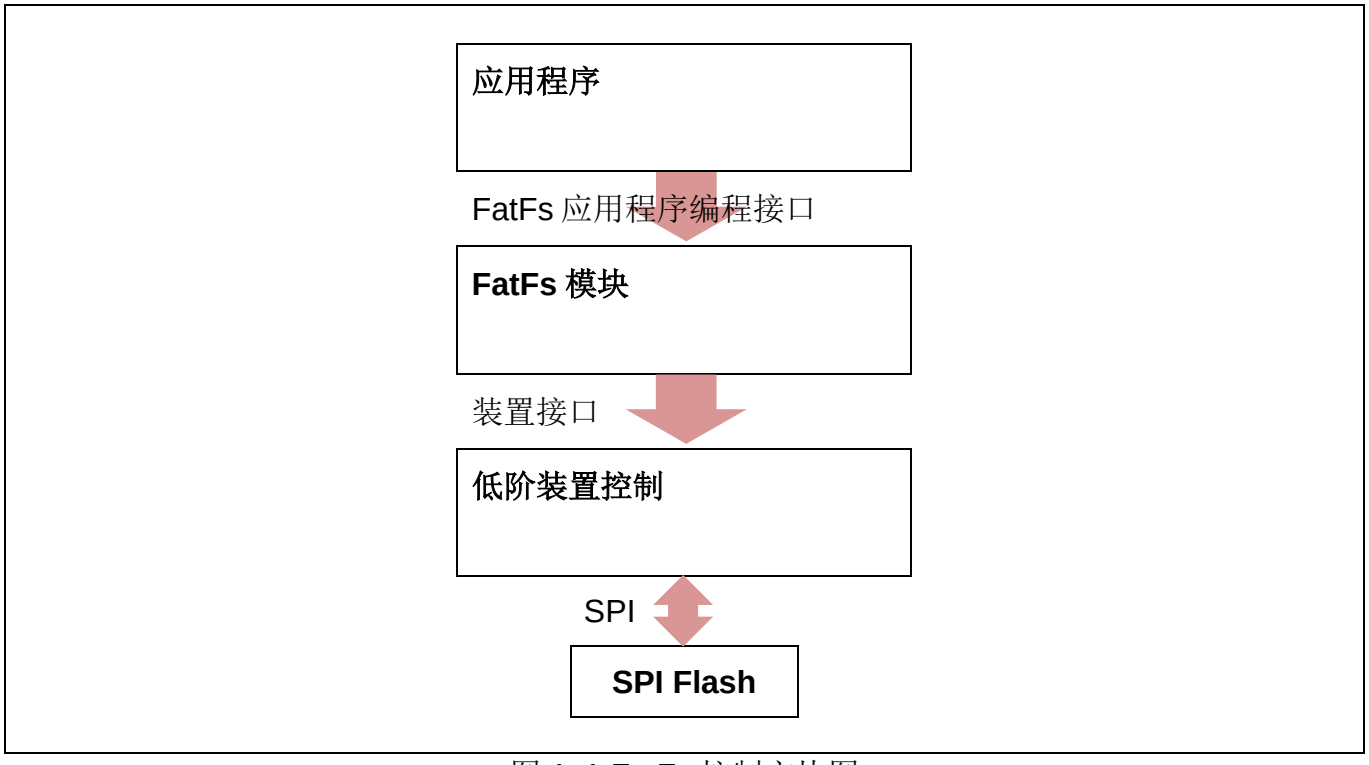

图 1-1 FatFs 控制方块图

FatFs控制方块图如图 1-1所示。FatFs应用程序编程接口为应用程序提供各种文件系统功能，在本范例中所使用到的应用程序编程接口被列在表 1。

FatFs应用程序编程接口	描述
f_mount	注册/注销一个工作区

f_mkfs	在逻辑驱动器上建立一个FAT系统
f_opendir	打开一个目录
f_readdir	读取目录里的档案及信息
f_getfree	回传系统全部空间及剩余空间

表 1 在范例程序中使用的应用程序接口

因为FatFs模块是独立于平台与储存媒体的文件系统层，故其可与像是SPI Flash的实体装置完全分开。如同图 1-1所示，低阶装置控制模块并非FatFs模块的任一部份，因此，本范例需实现装置接口以给FatFs存取SPI Flash使用。表 2列出了本范例所提供的媒体存取接口及其功能描述。如想得知更多关于应用程序接口与媒体存取接口的信息，请参考FatFs的官方网站：http://elm-chan.org/fsw/ff/00index_e.html。

装置接口	描述
disk_status	回传装置的状态
disk_initialize	初始化装置
disk_read	读取扇区
disk_write	写入扇区
disk_ioctl	控制装置相关的功能

表 2 在范例程序中实现的媒体存取接口

1.2.2 SPI Flash

本范例使用W25X16A串行闪存当作储存装置。W25X16A是通过一个SPI兼容的总线来进行存取，该总线由四个讯号组成，包括：串行时钟(Serial Clock, CLK)、芯片选择(Chip Select, /CS)、串行数据输入(Serial Data Input, DI)，以及串行数据输出(Serial Data Output, DO)。其引脚配置图如图 1-2所示。

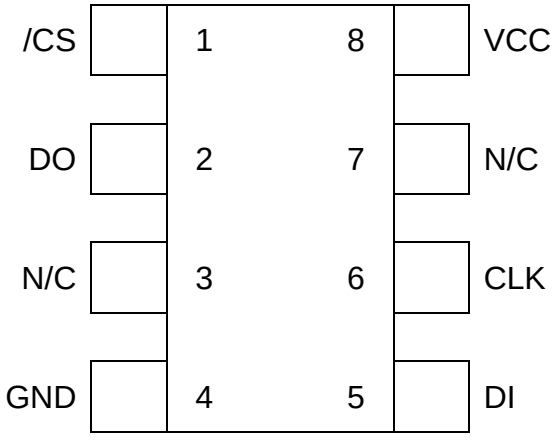

图 1-2 W25X16A 引脚配置图

标准的SPI指令使用DI输入引脚在CLK的上升缘将指令、地址或数据串行写入装置，而DO输出引脚则用来在CLK的下降缘读出装置的数据或者状态。表 3列出了本范例所使用到的W25X16A指令。此外，制造商标识与装置标识列于表 4中。

指令名称	命令	描述
DRVSPIFLASH_WRITE_ENABLE	0x06	致能写入
DRVSPIFLASH_WRITE_DISABLE	0x04	禁能写入
DRVSPIFLASH_READ_STATUS	0x05	读取状态缓存器 - 1
DRVSPIFLASH_FAST_READ	0x0B	快速读取
DRVSPIFLASH_PAGE_PROGRAM	0x02	页面编程
DRVSPIFLASH_SECTOR_ERASE	0x20	扇区擦除 (4KB)
DRVSPIFLASH_DEVICE_ID	0x90	制造商/设备 标识

表 3 在范例程序中使用的指令表

制造商标识	
Winbond Serial Flash	0xEF
设备标识	
W25X16A	0x14

表 4 制造商标识与设备标识

1.3 执行结果

以下执行结果会输出于UART鲍率设定为115200的终端机窗口。
当“FF_USE_MKFS”被定义为1时，文件系统将会被格式化。图 1-3为格式化文件系统后的目录信息，用户可以看到在SPI Flash中有多少可用的储存空间。

```
MID and DID = 0xEF14
SPI Flash Open success
SPI Flash Open success
SPI Flash Open success
SPI Flash Open success
rc=0 FR_OK
   0 File(s),          0 bytes total
   0 Dir(s),    2041856 bytes free
```

图 1-3 没有文件在内之目录信息

当使用者在SPI Flash内储存文件后，包括文档属性、修改日期、修改时间、大小、名称以及剩余多少可用空间的目录信息将会被列出，如图 1-4所示。

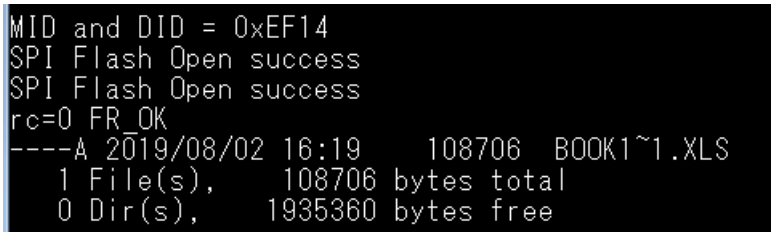

图 1-4 有文件在内之目录信息

2 代码介绍

初始化 SPI 及配置 SPI 多功能引脚：

```
/* 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);
```

读取 SPI Flash 的制造商标识与设备标识来确认 SPI Flash 是否已开放成功：

```
/* 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");
}
```

注册一个逻辑驱动器：

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

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

建立一个 FAT 系统，用户可以在 ffconf.h 定义“FF_USE_MKFS”为 1 来格式化文件系统：

```
#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
```

列出目录信息，用户可以从 UART 日志获得此信息：

```
    /* 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 软件与硬件环境

- 软件环境
  - BSP 版本
    - NUC230/NUC240 Series BSP CMSIS v3.01.002
  - IDE 版本
    - Keil uVision V5.26
- 硬件环境
  - 电路组件
    - NuTiny-EVB-NUC240
    - SPI Flash (W25X16A)
  - 示意图
    - 如图 3-1 所示，分别将 NUC240 的引脚：SPI0_SS0 (PC.0)、SPI0_CLK (PC.1)、SPI0_MISO0 (PC.2)、SPI0_MOSI0 (PC.3)与 W25X16A 的引脚：/CS、CLK、DO、DI 进行连接以藉由 SPI 存取 SPI Flash。

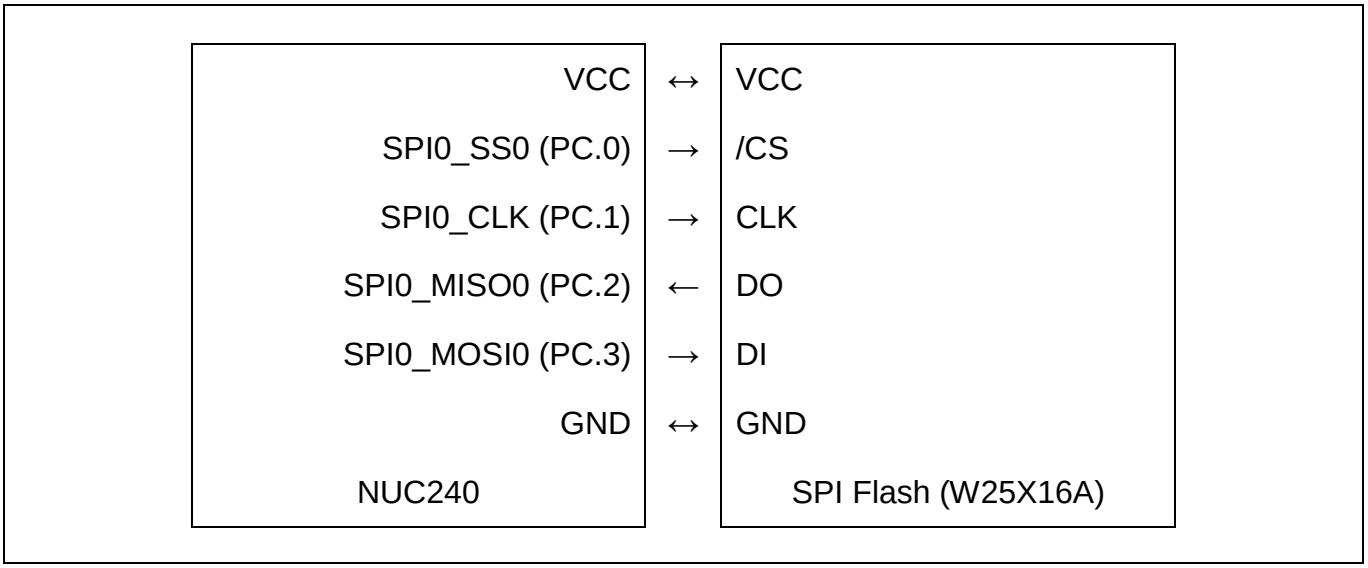

图 3-1 NUC240 与 W25X16A 引脚接线图

4 目录信息

- 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 如何执行范例程序

1. 根据目录信息章节进入 ExampleCode 路径中的 KEIL 文件夹，双击 NUC240_SPI_FATFS.uvproj。
2. 进入编译模式接口
   a. 编译
   b. 下载代码至内存
   c. 进入 / 离开除错模式
3. 进入除错模式接口
   a. 执行代码

6 修订纪录

Date	Revision	Description
Aug. 8, 2019	1.00	1. 初始发布.

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.