M480 FreeRTOS SD FATFS

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	Access a SD card using FatFs with FreeRTOS.
BSP Version	M480 Series BSP CMSIS V3.03.001
Hardware	NuMaker-PFM-M487 Ver 3.0

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

NuMicro® M480 series support SD/SDHC card. In M480 BSP, the sample bsp\SampleCode\StdDriver\SDH_FATFS shows ability to access an SD card in FAT format. The file system layer adopts the third-party library FATFS, which is an open-source project for FAT16/32 file system.
The example code M480_FreeRTOS_SD_FATFS modifies BSP sample SDH_FATFS to work under FreeRTOS, which is a real-time operating system. M480_FreeRTOS_SD_FATFS creates a task to execute the SDH_FATFS sample code.

1.2 Principle

This example code creates the “Check” task and the “SD_FATFS” task. The function of “SD_FATFS” task is identical to BSP sample SDH_FATFS.
“SD_FATFS” task provides a command-line interface to access the SD/SDHC card. The example code detects SD card insert and remove automatically. Once a SD/SDHC card inserted, the example code detects and mounts FATFS on it. The following commands can be used to access SD/SDHC card:

Command	Description
q	Exit “SD_FATFS” task
dd	Dump a sector
di	Re-initialize the SD card
bd	Dump r/w buffer
be	Edit r/w buffer
br	Read SD card sectors into r/w buffer
bw	Write r/w buffer into SD card sectors
bf	Fill r/w buffer
fs	Show logical drive status
fl	Directory listing
fo	Open a file
fc	Close a file
fe	Seek file pointer
fd	Read and dump file from current file pointer
fr	Read file
fw	Write file
fn	Change file/directory name
fu	Unlink a file or directory
fv	Truncate file

fk	Create a directory
fa	Change file/directory attribute
ft	Change time stamp
fx	Copy file
fg	Change current directory
fj	Change current drive
fz	Change r/w length of fr/fw/fx command
?	Show usage

1.3 Demo Result

```
FreeRTOS is starting ...
***** card insert !
>fl
D-HS- 2019/06/12 09:56         0  SYSTEM~1
D---- 2019/05/31 13:07         0  THIRDP~1
D---- 2019/05/31 13:02         0  LIBRARY
D---- 2019/05/31 13:05         0  SAMPLE~1
   0 File(s),         0 bytes total
   4 Dir(s),  848003072 bytes free
>dd 0
Sector:0
00000000  FA B8 00 10 8E D0 BC 00 B0 B8 00 00 8E D8 8E C0 ................
00000010  FB BE 00 7C BF 00 06 B9 00 02 F3 A4 EA 21 06 00 ...|.........!..
00000020  00 BE BE 07 38 04 75 0B 83 C6 10 81 FE FE 07 75 ....8.u........u
00000030  F3 EB 16 B4 02 B0 01 BB 00 7C B2 80 8A 74 01 8B .........|...t..
00000040  4C 02 CD 13 EA 00 7C 00 00 EB FE 00 00 00 00 00 L.....|.........
00000050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000000A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000000B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000000C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000000D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000000E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000000F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000100  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000110  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000120  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
```

```
00000130  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000140  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000150  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000160  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000170  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000180  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000190  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000001A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000001B0  00 00 00 00 00 00 00 00 23 2E 90 3C 00 00 00 20 ........#..<...
000001C0  21 00 0C FE FF FF 00 08 00 00 00 30 9B 03 00 00 !..........0....
000001D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000001E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000001F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 55 AA ..............U.
>
```

2 Code Description

Initialize SD Host Controller and multiple function pins

```
    /* select multi-function pin */
    SYS->GPE_MFPL &= ~(SYS_GPE_MFPL_PE7MFP_Msk | SYS_GPE_MFPL_PE6MFP_Msk |
                       SYS_GPE_MFPL_PE3MFP_Msk | SYS_GPE_MFPL_PE2MFP_Msk);
    SYS->GPE_MFPL |= (SYS_GPE_MFPL_PE7MFP_SD0_CMD | SYS_GPE_MFPL_PE6MFP_SD0_CLK |
                      SYS_GPE_MFPL_PE3MFP_SD0_DAT1 | SYS_GPE_MFPL_PE2MFP_SD0_DAT0);
    SYS->GPB_MFPL &= ~(SYS_GPB_MFPL_PB5MFP_Msk | SYS_GPB_MFPL_PB4MFP_Msk);
    SYS->GPB_MFPL |=  (SYS_GPB_MFPL_PB5MFP_SD0_DAT3 | SYS_GPB_MFPL_PB4MFP_SD0_DAT2);


    SYS->GPD_MFPH &= ~(SYS_GPD_MFPH_PD13MFP_Msk);
    SYS->GPD_MFPH |=  (SYS_GPD_MFPH_PD13MFP_SD0_nCD);


    /* Enable SD Host Controller engine clock */
    CLK_EnableModuleClock(SDH0_MODULE);


    /* Select SD clock source from PLL/10 */
    CLK_SetModuleClock(SDH0_MODULE, CLK_CLKSEL0_SDH0SEL_PLL, CLK_CLKDIV0_SDH0(10));


    /* SD initial state needs 400KHz clock output, driver will use HIRC for SD initial
       clock source. And then switch back to the user's setting.  */
    SDH_Open_Disk(SDH0, CardDetect_From_GPIO);
```

Create SD_FATFS task:

```
    /* Create SD_FATFS task */
    xTaskCreate( vSDFATFSTask, "SD_FATFS", mainCHECK_TASK_STACK_SIZE, NULL,
mainQUEUE_POLL_PRIORITY, NULL );


    /* Start the FreeRTOS scheduler. */
    vTaskStartScheduler();
```

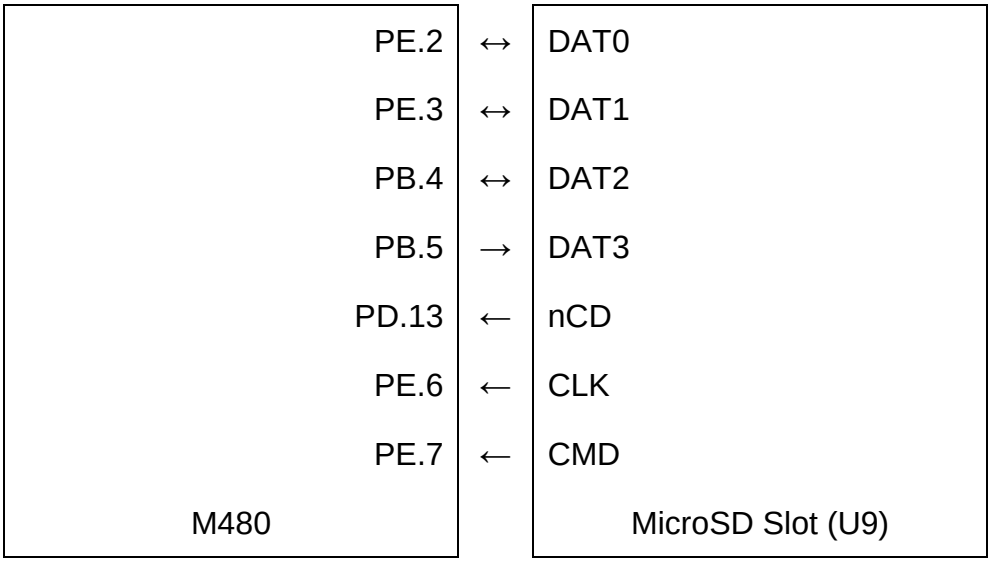
3 Software and Hardware Environment

Software Environment

- BSP version
  - M480 Series BSP CMSIS v3.03.001
- IDE version
  - Keil uVersion 5.26

Hardware Environment

- Circuit components
  - NuMaker-FPM-M487
  - MicroSD Slot (U9)
- Diagram



4 Directory Information

- EC_M480_FreeRTOS_SD_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.html.
    - FreeRTOS — A real time operating system available for free download. Its official website is: http://www.freertos.org/

5 How to Execute Example Code

1. Browsing into sample code folder by 4.Directory Information and double click FreeRTOS_SD_FATFS.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
June 12, 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.