Scope

This document describes how to test USB Host MSD fatfs example.

The MFS filesystem is not contained in usb stack, it could be found in SDK_SOURCE/filesystem/mfs/examples directory .

Preparation

Host

A board, i.e. twrk22f120m, which is running host_msd_fatfs_twrk22f120m example.

Device

A u-disk

Libraries dependency

The libraries dependency for various RTOS lists as following,

BM

Library project path:

- *<install_dir>/usb/usb_core/host/build/<tool_chain>/usbh_sdk_<board>_bm*
- *<install_dir>/lib/ksdk_platform_lib/<tool_chain>/<platform>*

FreeRTOS

Library project path:

- *<install_dir>/usb/usb_core/host/build/<tool_chain>/usbh_sdk_<board>_freertos*
- *<install_dir>/lib/ksdk_freertos_lib/<tool_chain>/<platform>*

MQX

Library project path:

- *<install_dir>/rtos/mqx/mqx/build/<tool_chain>/mqx_<board>*
- *<install_dir>/rtos/mqx/mqx_stdlib/build/<tool_chain>/mqx_stdlib_<board>*
- *<install_dir>/usb/usb_core/host/build/<tool_chain>/usbh_sdk_<board>_mqx*
- *<install_dir>/lib/ksdk_mqx_lib/<tool_chain>/<platform>*

uCOSii

Library project path:

- *<install_dir>/usb/usb_core/host/build/<tool_chain>/usbh_sdk_<board>_ucosii*
- *<install_dir>/lib/ksdk_ucosii_lib/<tool_chain>/<platform>*

uCOSiii

Library project path:

- *<install_dir>/usb/usb_core/host/build/<tool_chain>/usbh_sdk_<board>_ucosiii*
- *<install_dir>/lib/ksdk_ucosiii_lib/<tool_chain>/<platform>*

Refer to **Integration of the USB Stack and Kinetis SDK_review.pdf(***<install_dir>/doc***)** on how to build the corresponding libraries.

Steps

Follow the steps to run the MSD fatfs demo.

1. Run the host_msd_fatfs_twrk22f120m example and you will see the printed guide note.
2. Plug-in the u-disk and you will see some attach information printed out.
3. If you run the demo in bm, test result as follows:

```
*********************************************************************************
*                               FatFS DEMO                                      *
*          Configuration:  LNF Enabled, Code page  =437                          *
*********************************************************************************
*********************************************************************************
*                             DRIVER OPERATION                                  *
*********************************************************************************
1. Demo function: f_mount

  Initializing logical drive 0...
  Initialization complete
…
*-----------------------------   DEMO COMPLETED    ------------------------ *
```

```
*****************************************************************************
```

4. If you run the demo in mqx, you can type string ”help” to get shell command. And according to the prompt information, you can do something. For example, you can input string "dir" to get the file list of the u-disk in the current directory.
5. If you want to test throughput, you should set THROUGHPUT_TEST_ENABLE as (1) in file mfs_usb.h for mqx, file msd_diskio.h for sdk. Then an additional 64K ram is required to test the throughput, the macro is unsupported for the board that doesn’t have the enough ram. Then run the demo the result is as follows:

```
*****************************************************************************
*                                file.dat for write and read speed                                *
*****************************************************************************
**************************1 write 100M test***********************************
write test results: time = 247577ms  speed = 413K/s
**************************1 read 100M test************************************
read test results: time = 153873ms  speed = 665K/s
*****************************************************************************
*                                      TEST DONE                                                  *
*****************************************************************************
```

Note:

1. The macro “THROUGHPUT_TEST_ENABLE” is unsupported in Vybrid series board.
2. In KSDK, The macro “THROUGHPUT_TEST_ENABLE” only support twrk64f120m. For KDS, Atollic and armgcc, the linker file need to swap m_data and m_data_2 as follow:

```
m_data      (RW)  : ORIGIN = 0x1FFF0000, LENGTH = 0x00010000
m_data_2    (RW)  : ORIGIN = 0x20000000, LENGTH = 0x00030000
```

Change to

```
m_data_2    (RW)  : ORIGIN = 0x1FFF0000, LENGTH = 0x00010000
m_data      (RW)  : ORIGIN = 0x20000000, LENGTH = 0x00030000
```