Freescale MQX RTOS Example Guide

Mfs_sdcard example

This document explains the mfs_sdcard example, what to expect from the example and a brief introduction to the API.

The example

The application example code is used to demonstrate how to communicate with MQX File System (MFS) on SD card.

The example code opens SD card device and installs MFS. It allows user to perform some basic operation with the SD card through the terminal interface for example write/copy/create/rename. It shows how to work with the driver and how to use shell commands.

Running example

Check that the SHELLCFG_USES_MFS macro is set to 1 in the <MQX installation folder>/config/<board>/user_config.h.

Check the SD card's channel that is used by MCU to communicate with SD card. There are three available channels:

- BSP_SDCARD_ESDHC_CHANNEL
- BSP_SDCARD_SDHC_CHANNEL

And

- BSP_SDCARD_SPI_CHANNEL

If channel is defined, check corresponding macro is set to 1 in the user_config.h.

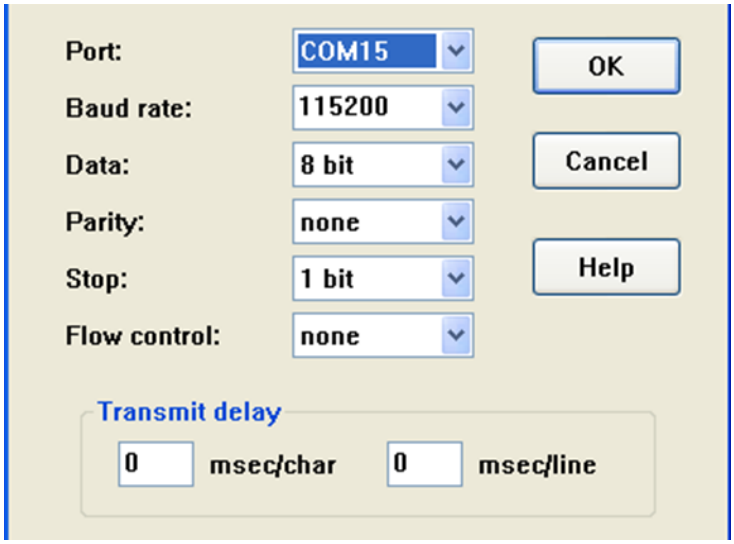
Then rebuild the BSP, PSP, MFS and SHELL projects for the target platform/IDE.

Hardware requirement

MCU board, SD card

If MCU board does not support micro SD interface. Primary and TWR_MEM are needed.

Start a terminal application on your PC and set the serial connection for 115200 baud, 8 data bits, 1 stop bit, no parity and no flow control.



The Shell function takes an array of commands and a pointer to a file as parameters. The Shell_commands array specifies a list of commands and relates each command to a function.

When a command is entered into the Shell input the corresponding function is executed.

List of shell commands

- "cd": Change the current working directory.
- "copy": Copy a file to another file.
- "create": Create a file.
- "del": Delete a file.
- "disect": Reads a sector of memory.
- "dir": List all files contain in a folder.
- "df": Prints out disk free information for current file system.
- "format": Format folder.
- "help": List all the commands.
- "mkdir": The command creates one or more new directories.
- "pwd": The command is used to output the path of the current working directory.
- "read": Read file.

- "ren": Rename a file.
- "rmdir" Removes the directory entry specified by each directory argument, provided the directory is empty.
- "type": SHELL utility to Ping a host.

Explaining example

The application example creates two AUTO_START tasks: shell_task and sdcard_task with different priority and FIFO scheduling policy.

Because of higher priority, sdcard_task will be running task and shell_task will be ready task.

Sdcard_task's code sets a communication with SD card device, opens and installs MFS (MQX file system) if SD card is inserted, closes and uninstalls if not.

When the task is started, task opens a low level communication device by calling following function.

```
fopen(SDCARD_COM_CHANNEL, (void *)(SPI_FLAG_FULL_DUPLEX));
```

SDCARD_COM_CHANNEL will be one of three channel which are mentioned previous part.

Then task will install GPIO pin for SD card pin.

- Install CS (Chip select) pin if SDCARD_COM_CHANNEL is SPI_COM_CHANNEL.
- Install Detect pin if BSP_SDCARD_GPIO_DETECT macro is defined.
- Install Protect pin if BSP_SDCARD_GPIO_PROTECT macro is defined.

After SD card's pins are installed, task will install SD card device. The code is

```
_io_sdcard_install("sdcard:", (void *)&_bsp_sdcard0_init, com_handle);
```

Task goes into an infinite loop, gets value of SD card pins that are defined.

Then task checks SD card's state.

SD card's state is changed.

- ✓ SD card is inserted
  - Open the device which MFS will be installed on
  - Set read only flag as needed
  - Install partition manager over SD card driver. Then open and validate partition for installing MFS file over. If partition cannot be opened, MFS file will be install over SD card.

- Open file system.

✓ SD card is removed.

- Close and uninstall file system.
- Close and uninstall partition manager.
- Close SD card device.

SD card’s sate is unchanged.

Finally, Task waits 200 milliseconds then goes to begin of infinite loop.

Note: Partition manager

The partition manager device driver is designed to be installed under the MFS device driver. It lets MFS work independently of the multiple partitions on a disk. It also enforces mutually exclusive access to the disk, which means that two concurrent write operations from two different MFS devices cannot conflict. The partition manager device driver can remove partitions, as well as create new ones. The partition manager device driver is able to work with multiple primary partitions. Extended partitions are not supported.

Shell_task

While Sdcard_task is waiting 200 milliseconds, Shell_task will become running task and waits commands which is entered by user.

After 200 milliseconds are expire. Shell_task will become ready task and Sdcard_task is running task again (because of preemption) and so on.