

FreescalE MQX RTOS Example Guide

TFS example

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

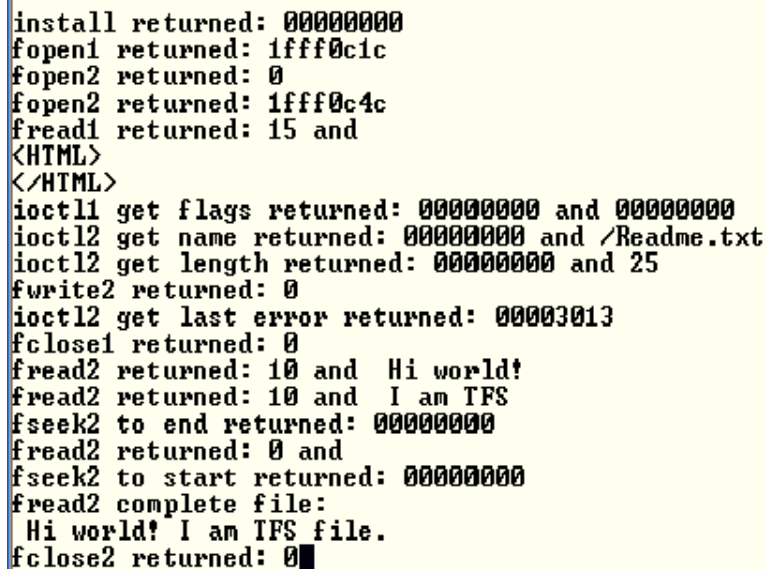
The example

The example code shows how to communicate with file system. The task performs some basic operation with the file system example open/read/write/close.

Running example

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.

Start TFS example on the target platform. After running example the result will be as the following picture



```
install returned: 00000000
fopen1 returned: 1fff0c1c
fopen2 returned: 0
fopen2 returned: 1fff0c4c
fread1 returned: 15 and
<HTML>
</HTML>
ioctl1 get flags returned: 00000000 and 00000000
ioctl2 get name returned: 00000000 and /Readme.txt
ioctl2 get length returned: 00000000 and 25
fwrite2 returned: 0
ioctl2 get last error returned: 00003013
fclose1 returned: 0
fread2 returned: 10 and Hi world!
fread2 returned: 10 and I am TFS
fseek2 to end returned: 00000000
fread2 returned: 0 and
fseek2 to start returned: 00000000
fread2 complete file:
Hi world! I am TFS file.
fclose2 returned: 0
```

Explaining the example

The application example creates only one main task. The main task starts to initialize the trivial file system.

```
_io_tfs_install ("tfs:", tfs_data);
```

After installing, Task opens two file systems in data folder: index.html and readme.txt. Then reading the content and getting the ATTRIBUTES, NAME and LENGTH of two file systems.

The task also tries to write the content of the first file system to second file system. But error code will return because disk is protected to write. The code to do that is:

- `result = fwrite(buffer, 1, 10, file2);`
- `printf ("\nfwrite2 returned: %d", result);`
- `result = ioctl(file2, IO_IOCTL_TFS_GET_LAST_ERROR, NULL);`
- `printf ("\nioclt2 get last error returned: 0x%X", result);`

Then the task closes index.html file and reads 20 first bytes of readme.txt file. After reading, Task seeks to the end of the file and tries to read the content.

Finally, Task seeks to begin of the file reads whole content and closes it.

