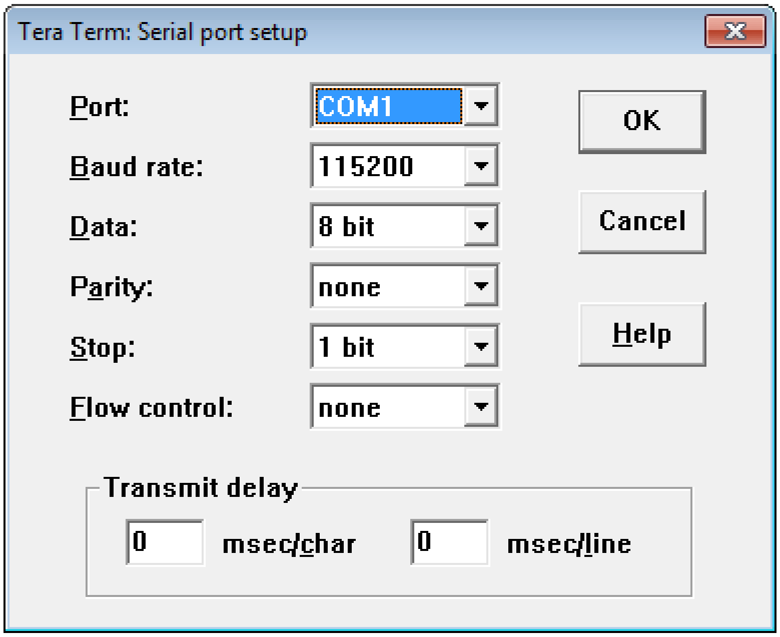
Freescale MQX Example Guide
Nandflash example

This document describes the Nandflash driver example application. The example opens the NAND Flash device, gets and displays basic NAND Flash organization data (ID, number of blocks, block size, physical page size, etc.), performs several erase/read/write operations and finally closes the device. It shows how to work with the driver and how to use dedicated IOCTL commands.

Running the example

Check that the BSPCFG_ENABLE_NANDFLASH macro is set to 1 in the <MQX installation folder>/config/<board>/user_config.h. Then re-build the BSP and PSP projects for the target platform/IDE.

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.



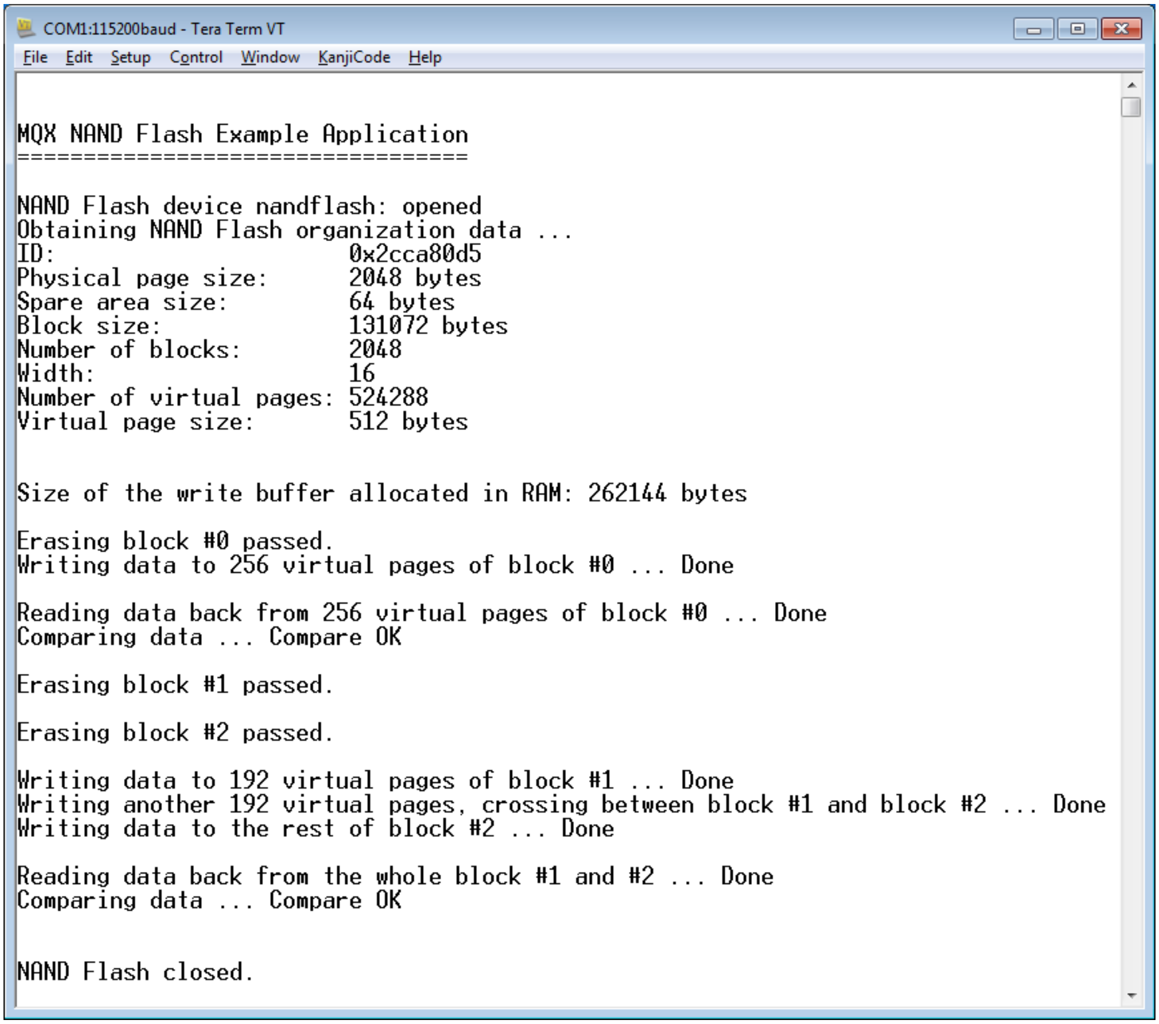
Tera Term: Serial port setup

Port: COM1
Baud rate: 115200
Data: 8 bit
Parity: none
Stop: 1 bit
Flow control: none

Transmit delay
0 msec/char 0 msec/line

Start nandlflash example on the target platform. For instructions how to do that in different IDEs and for different debuggers, see the MQX documentation (<MQX installation folder>/doc/tools).

After starting the application, you will see the printed message as the following picture.


COM1:115200baud - Tera Term VT
File Edit Setup Control Window KanjiCode Help

```
MQX NAND Flash Example Application
==================================

NAND Flash device nandflash: opened
Obtaining NAND Flash organization data ...
ID:                      0x2cca80d5
Physical page size:      2048 bytes
Spare area size:         64 bytes
Block size:              131072 bytes
Number of blocks:        2048
Width:                   16
Number of virtual pages: 524288
Virtual page size:       512 bytes


Size of the write buffer allocated in RAM: 262144 bytes

Erasing block #0 passed.
Writing data to 256 virtual pages of block #0 ... Done

Reading data back from 256 virtual pages of block #0 ... Done
Comparing data ... Compare OK

Erasing block #1 passed.

Erasing block #2 passed.

Writing data to 192 virtual pages of block #1 ... Done
Writing another 192 virtual pages, crossing between block #1 and block #2 ... Done
Writing data to the rest of block #2 ... Done

Reading data back from the whole block #1 and #2 ... Done
Comparing data ... Compare OK


NAND Flash closed.
```

Explanation of the example

There is just one task (nandflash_task) that performs the following sequential operations above the NAND flash device:

- Open the NAND flash device

- Get NAND Flash organization data via IOCTL calls:
    - ID – read out from the NAND flash chip
    - Physical page size in bytes - read out from the NANDflash info structure
    - Spare area size in bytes - read out from the NANDflash info structure
    - Block size in bytes - read out from the NANDflash info structure
    - Number of blocks - read out from the NANDflash info structure
    - Width - read out from the NANDflash info structure
    - Number of virtual pages - read out from the NANDflash info structure
    - Virtual page size in bytes - read out from the NANDflash info structure

- Get Bad Block Table, i.e. the list of bad blocks

- Allocate read and write buffers in RAM
- Calculate the number of virtual pages per one block
- Fill data to write buffer
- Erase block #0 before writing to the first page of the block
- Write data into the block #0
- Read data back from the block #0 and compare with the write buffer

- If enabled via NANDFLASH_TEST_ERASE_CHIP macro erase the whole NAND flash

- The next set of operations shows how bad block management is performed. Block #1 is marked as bad first and checked afterwards. Then, the block marked as bad is forced to be erased causing the block #1 is marked as good again which is checked afterwards.

- If enough RAM has been allocated (at least the 4 * block size) perform next writing/reading operations, otherwise skip this block of operations.
- Fill new data to write buffer
- Write data into 3/4 of virtual pages of the block #1
- Write data into another X virtual pages, crossing between block #1 and block #2
- Write the rest of the block #2
- Read data back from the whole block #1 and #2 and compare with the write buffer

- Close the NAND flash device