Freescale MQX RTOS Example Guide

Flashx_swap example

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

The example

The memory address range of the flash memory is divided into the lower code addresses and upper code addresses (or flash space). The Flashx_swap example shows swapping of the two base addresses (or the base addresses of flash spaces) of the flash memory blocks. User is prompted to input a message which is then written into the upper flash space. The message is then read from the lower flash space and displayed on terminal.

Running the example

The Flashx_swap application belongs to the set of examples of MQX handling the flash memory. The BSPCFG_ENABLE_FLASHX macro must be set to non-zero in the user_config.h file prior to compilation of MQX libraries and the example itself.

To run the example the corresponding IDE, compiler, debugger and a terminal program are needed.

Explaining the example

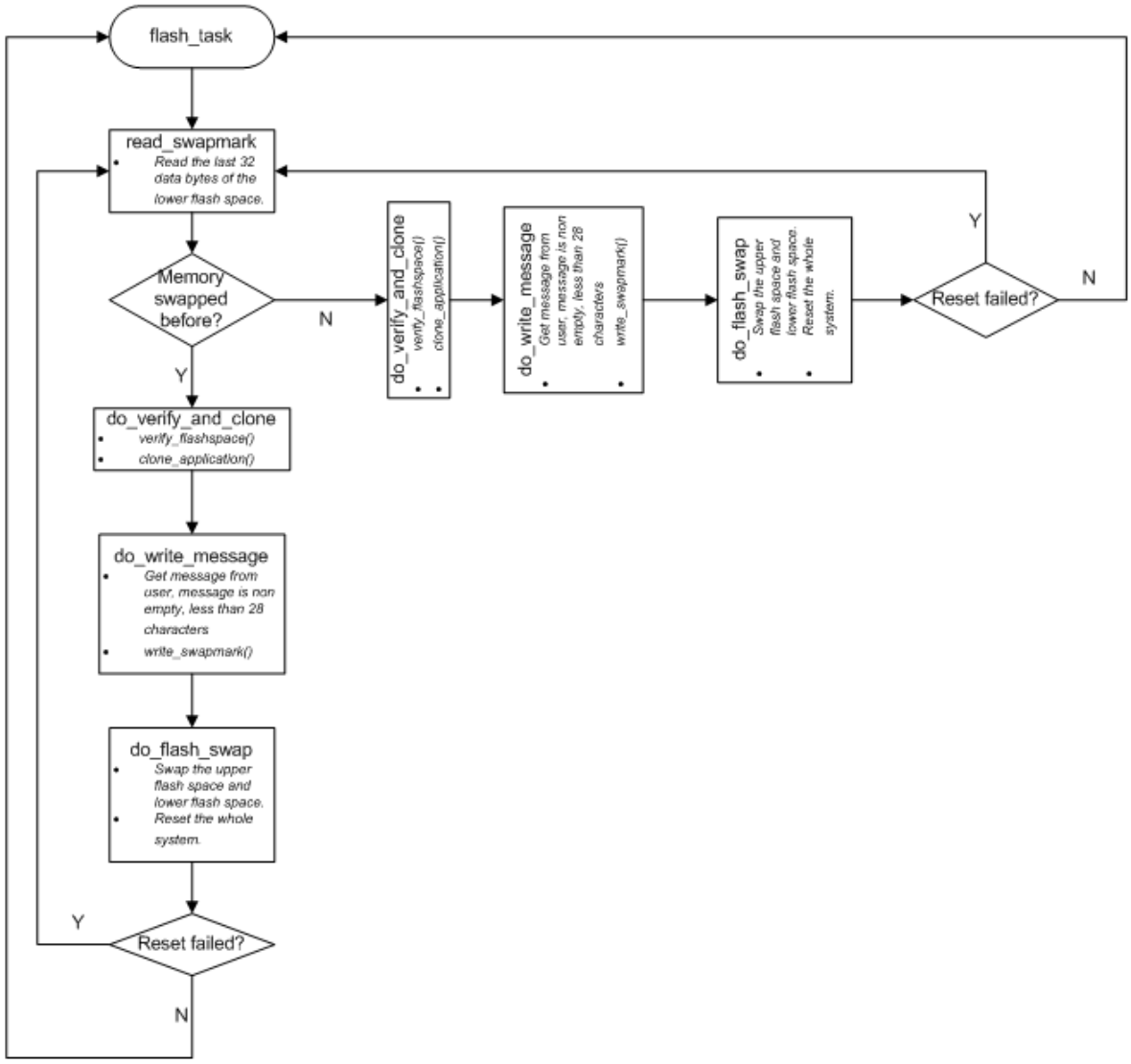
There is only one task flash_task is created. The flash_task task first reads the data in the lower flash space by calling function read_swapmark(). All the data in the lower flash space is then copied into upper flash space as function do_verify_and_clone() is invoked. As the function do_write_message() is called the user is prompted to input a message string which is then written into the upper flash space. The base addresses of the upper flash space and lower flash space is then swapped when function do_flash_swap() is invoked. The diagram in the next page explains the logic flow in more detail.

The flashx driver is used to read data from and to write data into the flash memory.

- The verify_flashspace() function checks the size of upper flash space and lower flash space to make sure that as swapping is carried out the upper flash space is large enough to hold the data from the lower flash space.
- The clone_application() function copies the data in the lower flash space into the upper flash space by calling series of read() and write() function from the flashx driver.
- The write_swapmark() function writes a number of bytes of data to the end of the selected flash space.
- The read_swapmark() function reads a number of bytes of data from the end of the selected flash space.

The following output is expected on the terminal.

```
------------------------------------------------------------
running application for the first time (no swapflash mark was found)
log : flash space verification start
log : flash space verification done
log : copying application start
log : copying application done
put your message ( max 27 characters ):
this is a test string more t
log : writing message to upper flash space - 'SWAP1' file
log : swapping flash space
------------------------------------------------------------
running application after swap (swapflash found)
reading message from lower flash space - 'SWAP0' file
your message is:

 this is a test string more

log : flash space verification start
log : flash space verification done
log : copying application start
log : copying application done
put your message ( max 27 characters ):
less than 27 chars

log : writing message to upper flash space - 'SWAP1' file
log : swapping flash space
------------------------------------------------------------
running application after swap (swapflash found)
reading message from lower flash space - 'SWAP0' file
your message is:

 less than 27 chars

log : flash space verification start
log : flash space verification done
log : copying application start
log : copying application done
put your message ( max 27 characters ):
more than 27 chars 012345678
log : writing message to upper flash space - 'SWAP1' file
log : swapping flash space
------------------------------------------------------------
running application after swap (swapflash found)
reading message from lower flash space - 'SWAP0' file
your message is:

 more than 27 chars 01234567

log : flash space verification start
log : flash space verification done
log : copying application start
log : copying application done
put your message ( max 27 characters ):
```


flash_task
read_swapmark
Read the last 32 data bytes of the lower flash space.
Memory swapped before?
N
Y
do_verify_and_clone
verify_flashspace()
clone_application()
do_write_message
Get message from user, message is non empty, less than 28 characters
write_swapmark()
do_flash_swap
Swap the upper flash space and lower flash space.
Reset the whole system.
Reset failed?
Y
N
do_verify_and_clone
verify_flashspace()
clone_application()
do_write_message
Get message from user, message is non empty, less than 28 characters
write_swapmark()
do_flash_swap
Swap the upper flash space and lower flash space.
Reset the whole system.
Reset failed?
Y
N