Freescale MQX RTOS Example Guide

CLKAPI example

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

The example

In this example reading data from quad spi serial flash at different clock rates is examined. The clock element of modules in Vybrid MCU is managed in a tree wherein each node represents current clock configuration of a module. The clock switching between different frequencies is accomplished via the clock management API functions. The clock switching processes and the statistic of the reading data process at different frequencies are recorded and shown on the terminal.

Running the example

This example is applicable to Vybrid tower board and Vybrid Autoevb board only.
The BSPCFG_ENABLE_CLKTREE_MGMT, BSPCFG_ENABLE_QUADSPI0 macro must be set to non-zero, BSPCFG_ENABLE_FLASHX must be set to zero in the user_config.h file prior to compilation of MQX libraries and the example itself.

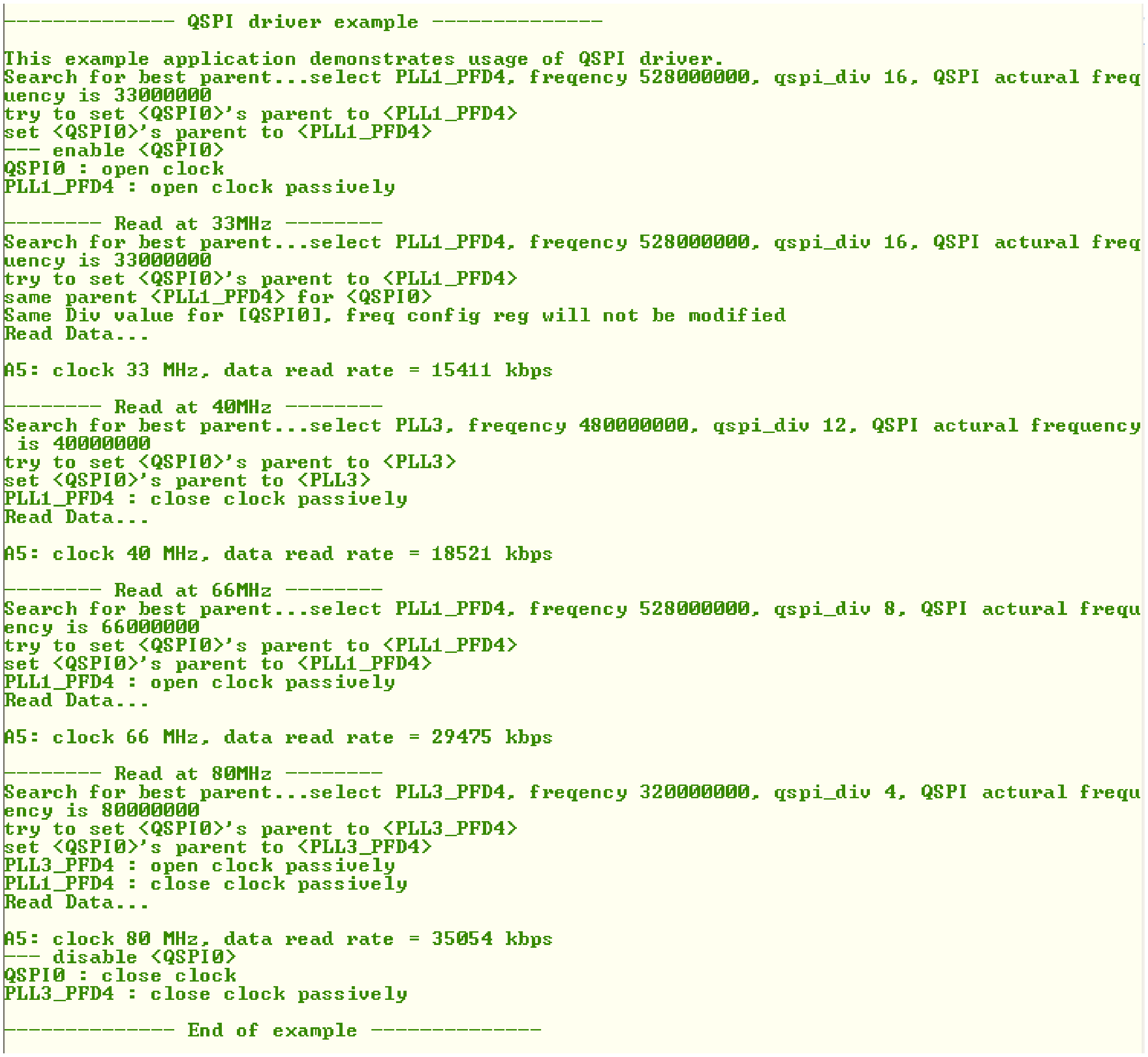
To run the example the corresponding IDE, compiler, debugger and a terminal program are needed.

Explaining the example

The example consists of three source files main.c, qpsi_memory.c and qpsi_memory.h. The qspi_memory.c and qspi_memory.h files define the constants and functions used in main.c file. The qspi API functions and cm API functions are used to provide access to the qspi serial flash memory and the clock module. In main.c file there is only one task called main_task which reads the data from qspi serial flash memory at four different clock rates: 33 MHz, 40 MHz, 66 MHz and 80 MHz. The read operation is accomplished through function *quadspi_valid_AHB_read()* which does the following jobs.

- Change the clock frequency of qspi serial flash for read operation using function *ioctl()* from qspi driver with the command *QuadSPI_IOCTL_SET_READ_SPEED* as one of the input parameters and the desired clock rate as other input parameter. The functions in cm driver are invoked to find the appropriate clock sources for the qspi serial flash memory according to different wanted clock rates. The statistic of clock switching process is displayed on the terminal.
- Read data from the qspi serial flash memory using AHB bus route with the call of function *memory_read_data()* from qspi_memory.c file. 1 Kbyte of data is read each time which is repeated over all the available sectors of a single qspi serial flash memory. The process is repeated 20 times to calculate the average reading rate. The statistic of reading process is recorded and shown on the terminal.

The main_task then closes the connection to the qspi serial flash memory and example is finished.
The output is shown as follow.



```
--------------- QSPI driver example ---------------

This example application demonstrates usage of QSPI driver.
Search for best parent...select PLL1_PFD4, freqency 528000000, qspi_div 16, QSPI actural freq
uency is 33000000
try to set <QSPI0>'s parent to <PLL1_PFD4>
set <QSPI0>'s parent to <PLL1_PFD4>
--- enable <QSPI0>
QSPI0 : open clock
PLL1_PFD4 : open clock passively

--------- Read at 33MHz ---------
Search for best parent...select PLL1_PFD4, freqency 528000000, qspi_div 16, QSPI actural freq
uency is 33000000
try to set <QSPI0>'s parent to <PLL1_PFD4>
same parent <PLL1_PFD4> for <QSPI0>
Same Div value for [QSPI0], freq config reg will not be modified
Read Data...

A5: clock 33 MHz, data read rate = 15411 kbps

--------- Read at 40MHz ---------
Search for best parent...select PLL3, freqency 480000000, qspi_div 12, QSPI actural frequency
 is 40000000
try to set <QSPI0>'s parent to <PLL3>
set <QSPI0>'s parent to <PLL3>
PLL1_PFD4 : close clock passively
Read Data...

A5: clock 40 MHz, data read rate = 18521 kbps

--------- Read at 66MHz ---------
Search for best parent...select PLL1_PFD4, freqency 528000000, qspi_div 8, QSPI actural frequ
ency is 66000000
try to set <QSPI0>'s parent to <PLL1_PFD4>
set <QSPI0>'s parent to <PLL1_PFD4>
PLL1_PFD4 : open clock passively
Read Data...

A5: clock 66 MHz, data read rate = 29475 kbps

--------- Read at 80MHz ---------
Search for best parent...select PLL3_PFD4, freqency 320000000, qspi_div 4, QSPI actural frequ
ency is 80000000
try to set <QSPI0>'s parent to <PLL3_PFD4>
set <QSPI0>'s parent to <PLL3_PFD4>
PLL3_PFD4 : open clock passively
PLL1_PFD4 : close clock passively
Read Data...

A5: clock 80 MHz, data read rate = 35054 kbps
--- disable <QSPI0>
QSPI0 : close clock
PLL3_PFD4 : close clock passively

--------------- End of example ---------------
```