Freescale MQX RTOS Example Guide

SPI_legacy example

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

The example

The example demonstrates the usage of spi driver to write data to and to read data from a SPI serial flash memory over the SPI bus. The example shows two mechanisms associated with SPI peripheral module: polling and interrupt. The polling mode and interrupt mode are tested with the same sequence of writing data to and reading data from flash memory. The output displayed on the terminal shows that the two methods produces similar result. The example shows the read and write of a data byte as well as the read and write of a long sequence of data bytes into flash memory.

Running the example

The BSPCFG_ENABLE_SPI0 and BSPCFG_ENABLE_ISPI0 macros 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

The application example uses only one task called main_task which implements the polling method and interrupt method of the SPI module to read data from and write data to serial flash memory. The example consists of 4 source files including the memory_int.c file defining the functions to access serial flash memory in interrupt mode, the memory_polled.c file defining the functions to access serial flash memory in polling mode, the memory.h and spi.c files. The logic flow can be described into two sections regarding to the polling method and interrupt method as follow.

- Polling method:
    - open the connection to the SPI peripheral module in polling mode and install the external chip select callback function associated with a gpio pin to manually enable the serial flash module in case the SPI module does not handle it.
    - examine the current baudrate over the SPI bus and set up the new baudrate value. The idle state of SPI clock is configured to logic low. The SPI module is set up as the master and the serial flash memory is the slave on the SPI bus.
    - display the statistics of communication over SPI bus. The result includes the number of interrupts, the packets transferred and the number of times the error occurs.
    - examine the current status of the serial flash memory, remove memory protection from the flash memory and erase the whole flash memory.
    - Write a byte into the serial flash memory and read it back to verify the operation of the SPI module. The statistic of

communication process over SPI bus is displayed over terminal.

  - Perform the read and write with a long sequence of data bytes. The example uses two strings with 12 characters and 72 characters consecutively at this step. This is to demonstrate the internal memory layout of serial flash memory which has memory pages with size of 256 bytes for each page.
  - Illustrate the characteristic of the SPI communication protocol where the master node and the slave node exchange data simultaneously. In this case the master exchanges a message including the address in the memory of the slave and some dummy data bytes. The received data from the serial flash memory and the transmitted data from the master are displayed over terminal.
- Interrupt method:
  - This case is similar to the polling method discussed previously except the SPI module is operated in the interrupt mode. The same process as in the case of the polling method is repeated for this case without the last operation where the simultaneous data exchange between master and slave of the SPI communication protocol was examined. However the memory addresses in the serial flash memory for the polling method to write data to and to read data from are different from the interrupt method’s addresses.

The output of this example is similar to the output of the SPI example. Please refer to the _REAMDE file of SPI example to see the example output.