

## FreescalE MQX Example Guide

### ipc example

This document describes the ipc (Inter Processor Communication) example application. The example shows how to work with the IPC MQX component and how to use it for serial-line-based inter processor communication. The example consists of two different projects targeted for two CPUs/boards. The cpu1 project serves as message sender while the cpu2 project is the message responder. The boards have to be connected via the serial line cable.

The sender project runs the MQX task on the first CPU. This task creates a message pool and a message queue. It sends a message to a queue on the second CPU. It waits for a return message, validating the message before sending a new message.

The responder project runs another MQX task, but on the second CPU. This task creates a message queue and then waits for a message. Upon receiving the message the data is incremented before the message is returned to the sender.

### Running the example

Before building MQX libraries it is necessary to enable the IPC functionality (disabled by default). The following options have to be added/enabled in the user\_config.h file:

```
#define MQX_USE_IPC 1
#define MQX_IS_MULTI_PROCESSOR 1
```

It is also necessary to consider which serial line will be used for the IPC and which serial port will be used for the application console. Typically, the serial port for the default serial console is defined in the BSP code (BSP\_DEFAULT\_IO\_CHANNEL) and there is no need to setup this explicitly. However, the serial port for the IPC has to be selected and enabled in the user\_config.h file. For example, if there are two serial connectors on the target board, one is enabled by default and used for the serial console. If we would like to use the second port for the IPC it has to be enabled in the user\_config.h, i.e. the following option has to be added/enabled in the user\_config.h file:

```
#define BSPCFG_ENABLE_ITTYB 1
```

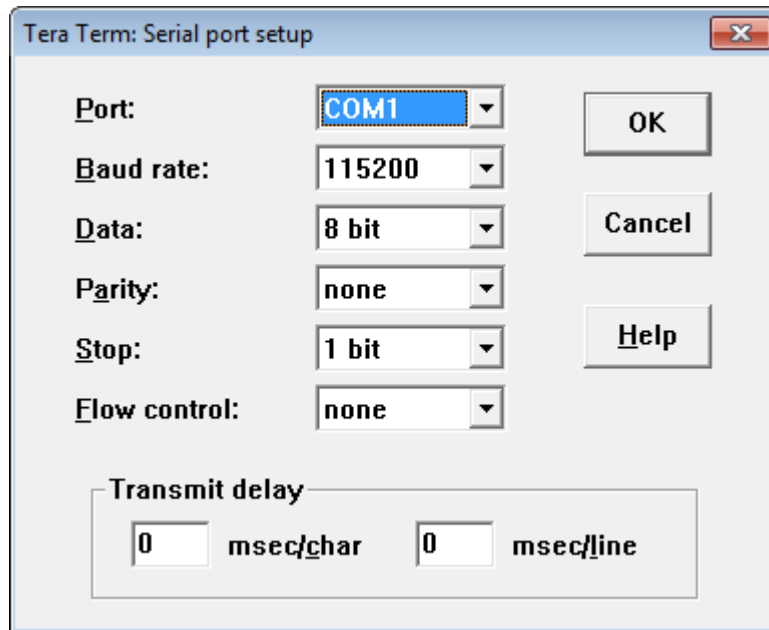
After user\_config.h modification rebuild the BSP and PSP projects for the target platform/IDE.

Before building ipc\_serial\_cpu1 and ipc\_serial\_cpu2 projects check that the IO\_PORT\_NAME member of the pcb\_mqxa\_init structure matches the IPC port number specified in the user\_config.h. As we have selected ITTYB ("I" means interrupt driven driver) in the user\_config.h the following declaration has to be placed in the ipc1.c/ipc2.c file:

```
#define DEFAULT_SERIAL_IPC_PORT_NAME "ittyb:"
```

Then build both ipc\_serial\_cpu1 and ipc\_serial\_cpu2 projects.

Start two terminal applications (for both sender and responder sides) on your PC and set the serial connection for 115200 baud, 8 data bits, 1 stop bit, no parity and no flow control.



Start `ipc_serial_cpu1` and `ipc_serial_cpu2` applications on target platforms. For instructions how to do that in different IDEs and for different debuggers, see the MQX documentation (<MQX installation folder>/doc/tools).

`ipc_serial_cpu2` application should be started as a first one (just responses message) while the `ipc_serial_cpu1` application should be started afterwards.

The following outputs can be seen on the serial console for individual boards.



