

## **Freescale MQX Example Guide**

### **ipc\_shmem 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 inter processor communication via the shared memory. This memory area is shared between individual cores of the multicore system like Freescale Vybrid MCU.

The example can be executed on dual core platforms. Two projects are prepared, sharing the same source files:

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

The responder project runs another MQX task (responder\_task), but on the second core. 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 back to the sender.

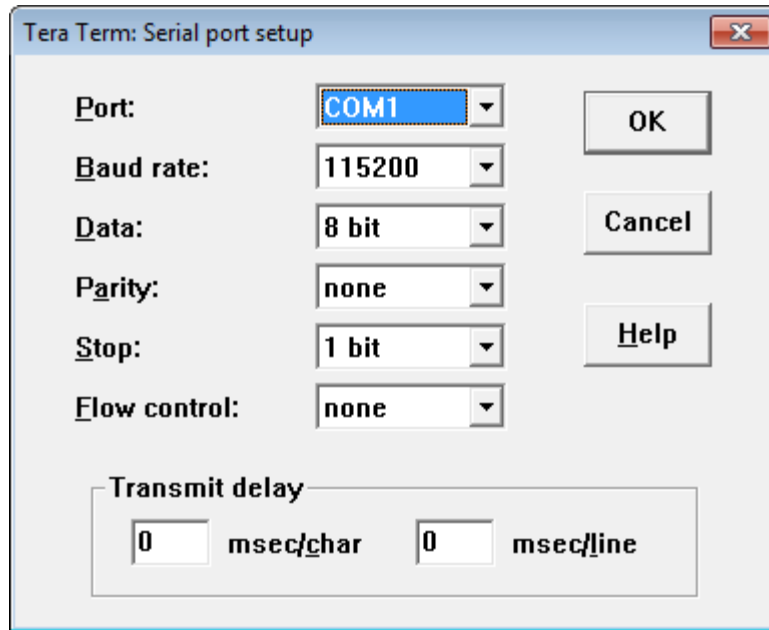
### **Running the example**

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
```

After user\_config.h modification rebuild the BSP and PSP projects for the target platform/IDE. Then build both ipc\_shmem\_<first\_core> and ipc\_shmem\_<second\_core> projects for both sides of the dual-core system.

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_shmem_<first_core>` and `ipc_shmem_<second_core>` 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`).

The responder application (`responder_task`) should be started as a first one (just responses message) while the sender application (`main_task`) should be started afterwards.

The following outputs can be seen on the serial console for individual cores of a dual-core system.

```
COM3:115200baud - Tera Term VT
File Edit Setup Control Window KanjiCode Help

Main task started
Main task sending
Main task receiving...done
Message: Size=c, SQID= 20a, TQID=109, DATA = 1
Main task sending
Main task receiving...done
Message: Size=c, SQID= 20a, TQID=109, DATA = 3
Main task sending
Main task receiving...done
Message: Size=c, SQID= 20a, TQID=109, DATA = 5
Main task sending
Main task receiving...done
Message: Size=c, SQID= 20a, TQID=109, DATA = 7
Main task sending
Main task receiving...done
Message: Size=c, SQID= 20a, TQID=109, DATA = 9
Main task sending
Main task receiving...done
Message: Size=c, SQID= 20a, TQID=109, DATA = b
Main task sending
Main task receiving...done
Message: Size=c, SQID= 20a, TQID=109, DATA = d
Main task sending
Main task receiving...done
Message: Size=c, SQID= 20a, TQID=109, DATA = f
Main task sending
Main task receiving...done
Message: Size=c, SQID= 20a, TQID=109, DATA = 11
Main task sending
Main task receiving...done
Message: Size=c, SQID= 20a, TQID=109, DATA = 13
```

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

Responder task started
Responder task receiving...done
Message: Size=c, SQID= 109, TQID=20a, DATA = 0
Responder task receiving...done
Message: Size=c, SQID= 109, TQID=20a, DATA = 2
Responder task receiving...done
Message: Size=c, SQID= 109, TQID=20a, DATA = 4
Responder task receiving...done
Message: Size=c, SQID= 109, TQID=20a, DATA = 6
Responder task receiving...done
Message: Size=c, SQID= 109, TQID=20a, DATA = 8
Responder task receiving...done
Message: Size=c, SQID= 109, TQID=20a, DATA = a
Responder task receiving...done
Message: Size=c, SQID= 109, TQID=20a, DATA = c
Responder task receiving...done
Message: Size=c, SQID= 109, TQID=20a, DATA = e
Responder task receiving...done
Message: Size=c, SQID= 109, TQID=20a, DATA = 10
Responder task receiving...done
Message: Size=c, SQID= 109, TQID=20a, DATA = 12
Responder task receiving...done
Message: Size=c, SQID= 109, TQID=20a, DATA = 14
Responder task receiving...done
Message: Size=c, SQID= 109, TQID=20a, DATA = 16
Responder task receiving...done
Message: Size=c, SQID= 109, TQID=20a, DATA = 18
Responder task receiving...done
Message: Size=c, SQID= 109, TQID=20a, DATA = 1a
Responder task receiving...done
Message: Size=c, SQID= 109, TQID=20a, DATA = 1c
```

