Freescale MQX Example Guide

The lwmsgq_lite example

This client/server model shows communication and task synchronization using message passing.

Server task initializes the message queues, creates three client tasks, and then waits for a message.

After receiving a message, the task returns the message to the sender.

Client task sends a message to the server_task and then waits for a reply.

Running the example

The user only needs to do compilation of MQX libraries, ksdk library and the example without any further step.

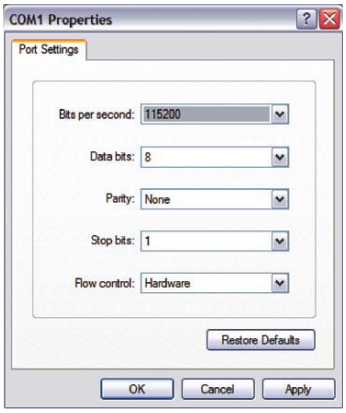
If the platform supports floating point, you have to disable floating point in <MQX_folder>\rtos\mqx\config\mcu\<board>\mqx_sdk_config.h:

```
#define MQXCFG_ENABLE_FP                0
#define MQXCFG_ALLOCATOR                MQX_ALLOCATOR_LWMEM
```

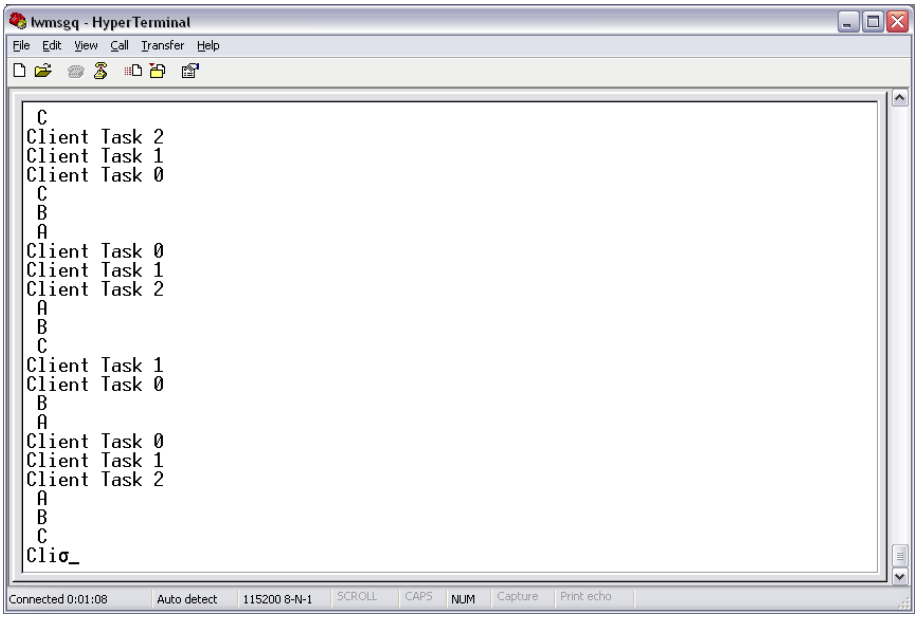
And rebuild MQX library.

Start HyperTerminal on the PC (Start menu->Programs->Accessories->Communications).
Make a connection to the serial port that is connected to the board (usually will be COM1).

Set it for 115200 baud, no parity, 8 bits and click OK.

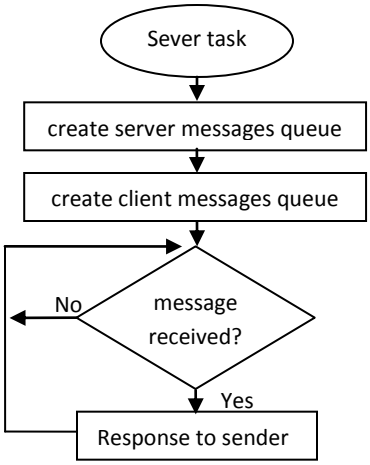


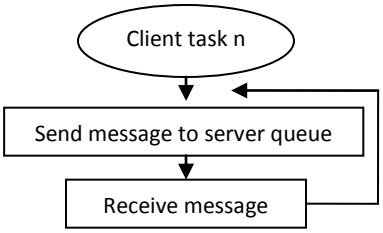
After running the MCU, you will see the printed message as the following picture.



Explanation of the example

The flow of the tasks is described in the next figure. There is a server task that receives all the incoming data and responds to them. There are three client tasks. Each client sends a message to the server and receives the answer back from it.


Sever task
create server messages queue
create client messages queue
message received?
No
Yes
Response to sender


Client task n
Send message to server queue
Receive message