**Freescale MQX Example Guide**

**Hello example**

This document describes the hello example application. The example hello handles two different tasks. Every task prints text to a console and ends.

**Running the example**

Start a terminal application on your PC and set the serial connection for 115200 baud, 8 data bits, 1 stop bit, no parity and no flow control.

Start hello example on the target platform. For instructions how to do that in different IDEs and for different debuggers, see the MQX documentation (<MQX installation folder>/doc/tools).

After starting the application, you will see the printed message as the following.

**Hello**

**World**

**Explanation of the example**

There are two tasks in the example (WORLD\_TASK, HELLO\_TASK). WORLD\_TASK starts automatically and try to create higher priority task HELLO\_TASK. If creations of HELLO\_TASK succeed, HELLO\_TASK only prints string “\nHello \n” and ends. After HELLO\_TASK ended, WORLD\_TASK prints string “World” and also ends.

WORLD\_TASK:

* Creates HELLO\_TASK by \_task\_create function. If creating failed, error message is printed out to the console.
* HELLO\_TASK is created with higher priority and is activated after creating.
* When scheduler activates WORLD\_TASK it prints out the string “World \n” by printf function.
* Calls \_task\_block function to end the task.

HELLO\_TASK:

* After creating and activating this task, m the string “\n Hello\n” is printed out by printf function.
* Calls \_task\_block function to end the task.