

# Freescal e MQX RTOS Example Guide

## Low Power Multi-Task example

This document explains the low power multi-task example for imx6sx board; the example consists of several tasks for stress test in order to measure the power consumption.

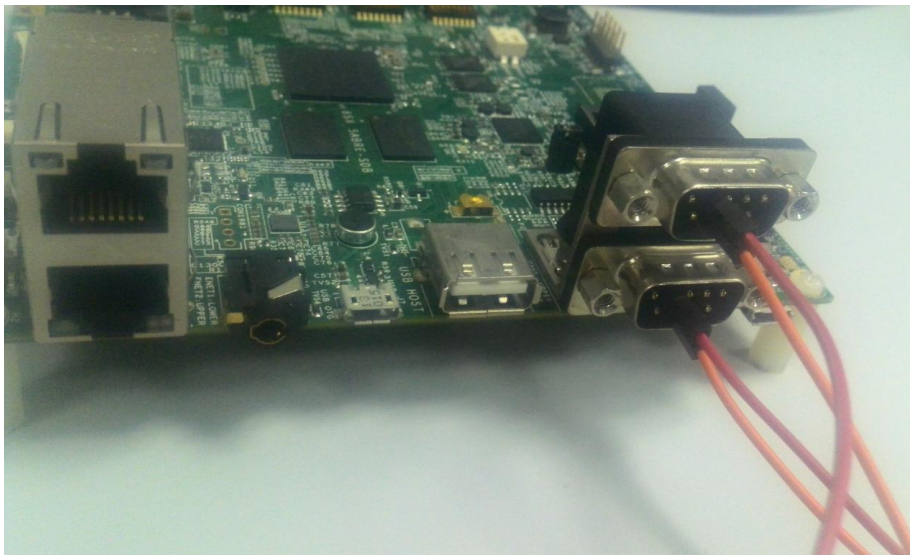
## The example

The example contains several tasks for adc, i2c and flexCAN. It aims to show power consumption state when these tasks are running at the same time. These tasks are running with time slice scheduling strategy except the main task, so it will never suspend at any task.

## Running the example

Due to most of example tasks are running with time slice scheduling, so the MQX\_HAS\_TIME\_SLICE macro must be set to non-zero in the user\_config.h file prior to compilation of MQX kernel libraries and the example itself.

You need to connect pins between two flexCAN instances on the board for flexCAN test.



**Figure 1**  
**Connect instance 1 and 2**

To run the example, the corresponding IDE, compiler, debugger and a terminal program are needed.

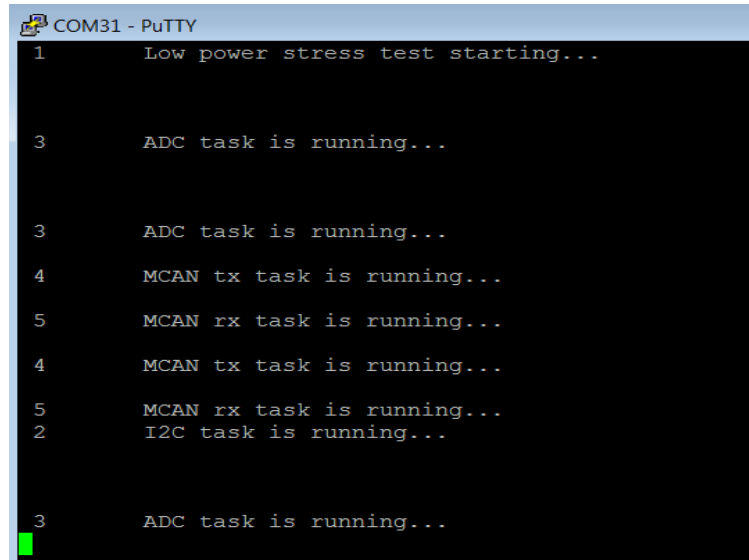
## Explaining the example

The application example creates five tasks:

- **main\_task:** this task initializes and creates different tasks for adc, i2c and flexCAN.
- **i2c\_task:** shows the usage of i2c driver as a master using either polling or interruption driver or an Accelerometer Sensor

- MMA8451Q as slave device. It will open the IIC driver and read data from MMA8451Q device.
- **adc\_task**: demonstrates how to read converted values from analogue inputs using lwadc api. It will loop forever to read adc input.
  - **flexcan\_tx\_task**: sending data to receive side.
  - **flexcan\_rx\_task**: receiving data from transmit side.

The expected console output will like below:



```
COM31 - PuTTY
1      Low power stress test starting...

3      ADC task is running...

3      ADC task is running...
4      MCAN tx task is running...
5      MCAN rx task is running...
4      MCAN tx task is running...
5      MCAN rx task is running...
2      I2C task is running...

3      ADC task is running...
```

**Figure 2**  
**Expected console output result**

The number at the beginning of each line is the task index.  
Each task name corresponds to task index is below:

```
#define MAIN_TASK    1
#define I2C_TASK     2
#define ADC_TASK     3
#define TX_TASK      4
#define RX_TASK      5
```