

Freescale MQX RTOS Example Guide

IRDA example

This document explains the IrDA transmitter and receiver examples, what to expect from the examples and a brief introduction to the API used.

UART IrDA MODE

This mode is a UART's feature. In the mode, The UART provides the capability of transmitting narrow pulses to an IR LED and receiving narrow pulses and transforming them to serial bits, which are sent to the UART. The IrDA physical layer specification defines a half-duplex infrared communication link for exchanging data.

IOCTL COMMAND FOR IRDA MODE

- ***IO_IOCTL_SERIAL_SET_IRDA_RX:***
Depending on parameter, the command is used to configure the IrDA RX channel is enabled or disabled
- ***IO_IOCTL_SERIAL_SET_IRDA_TX:***
Depending on parameter, the command is used to configure the IrDA TX channel is enabled or disabled

The example

This section describes the IrDA mode application. The example shows how to work with the IrDA and how to use it.

The example consists of two different projects:

- The transmitter project runs on the first board. It will transfer a number of messages to receiver and wait for acknowledge.
- The receiver project runs on another board. It will receive data from transmitter and send acknowledge message.

Running the example

The transmitter sends 50 messages to receiver via Infrared. The receiver will send an ACK for each message to transmitter.

Before building MQX libraries it is necessary to enable `BSP_ENABLE_TTYx` in `user_config.h` for IrDA channel. It depends on using board, most of all are using TTYA for IrDA channel.

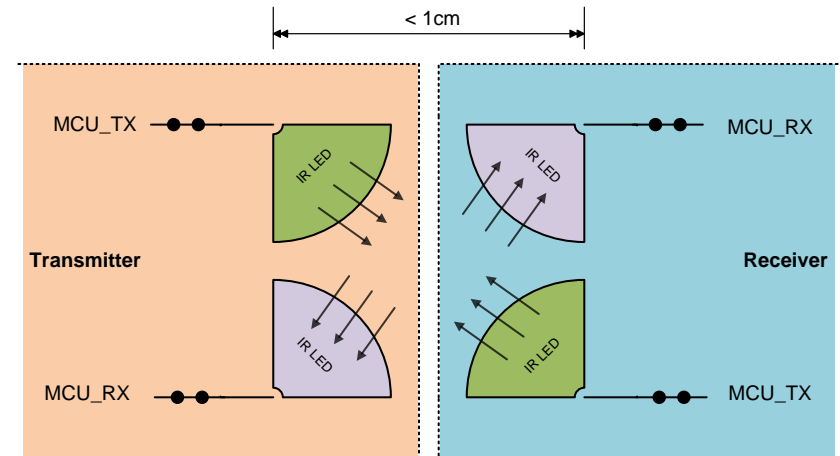
Start transmitter and receiver applications on target platforms. For instructions how to do that in different IDEs and for different debuggers, see the MQX documentation (<MQX installation folder>/doc/tools).

Receiver application should be started as a first one (just responses message) while the transmitter application should be started afterwards.

NOTE: Due to hardware design of the Infrared LED Transmit/Receive do not have electric current strength enough. The test setup should be as these two scenarios below.

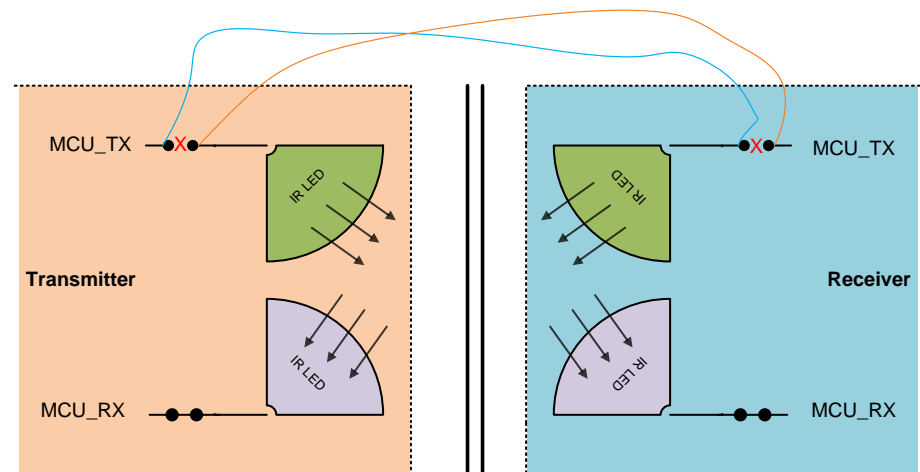
Scenario 1:

The transmitter and receiver must be closer less than 1cm distance.



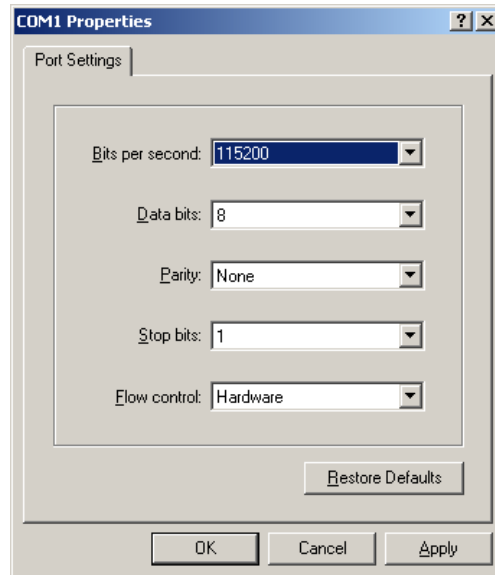
Scenario 2:

Transfer data between 2 Infrared LEDs that are fixed on each board. Use external wire connector connects TX pin of transmitter to TX LED of receiver and vice versa as below picture:



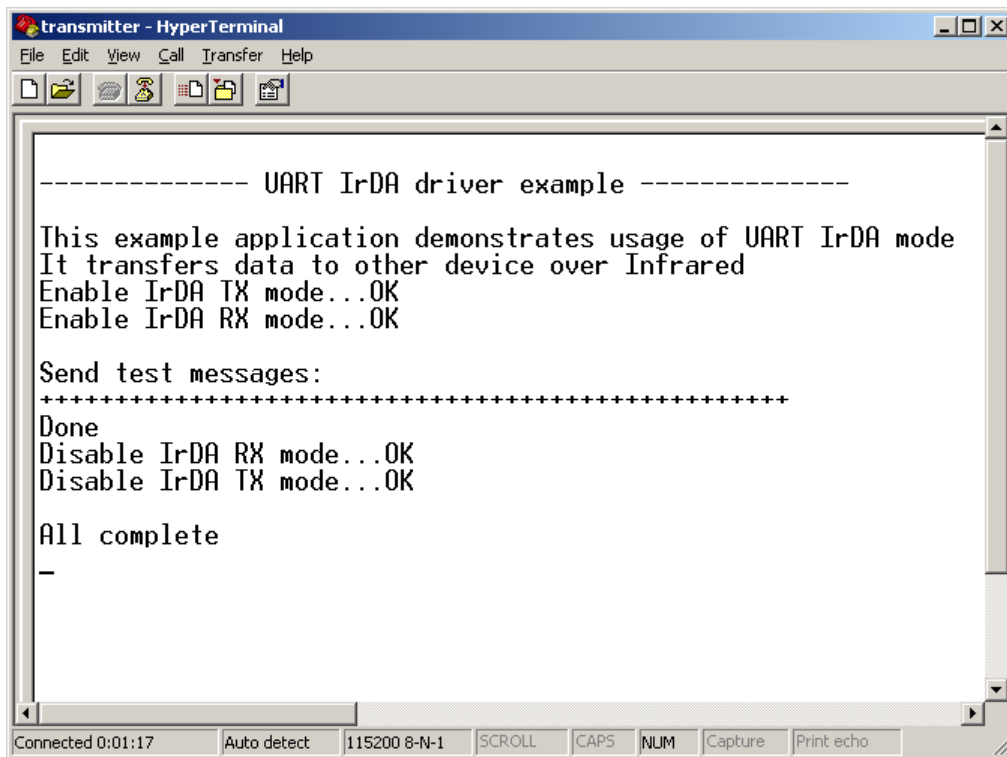
This setting is recommended because it more stable than the first once. The distance of transmitter and receiver is fixed and close enough.

To display the result, please turn on three terminal applications (for master and two slaves) on your PC and configure them as following:
115200 baud, 8 data bits, 1 stop bit, no parity and no flow control.



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

Transmitter board:



Receiver board:

The screenshot displays a HyperTerminal session. The title bar reads "Receiver - HyperTerminal". The menu bar includes "File", "Edit", "View", "Call", "Transfer", and "Help". Below the menu is a toolbar with icons for file operations and communication. The main text area contains the following output:

```
Receiver running...  
Message received  
Message received  
Message received  
Message received  
Message received  
Message received  
Message received  
Message received  
Message received  
Message received  
Message received  
Message received  
Message received  
Message received  
Message received  
Message received  
Message received  
Message received  
Message received  
Message received  
Message received  
Message received  
Message received
```

The status bar at the bottom provides connection details and controls: "Connected 0:00:57", "Auto detect", "115200 8-N-1", and buttons for "SCROLL", "CAPS", "NUM", "Capture", and "Print echo".

Explanation of the example

