Freescale MQX RTOS Example Guide

Rs485 example

This document describes the rs485 component example application. The rs485 example demonstrates how to transmit and receive data through rs485 data's channel.

The example consists of two tasks: rs485_write_task and rs485_read_task.

Rs485_write_task sends a data buffer to rs485's channel. Data will be transfer from tower 1 to tower 2 and prints out in terminal of tower 2.

Rs485_read_task waits for data in rs485's channel which is entered in terminal of tower 2.

Running example

Requirement of hardware

Need 2 towers to run the example:

- Tower 1: Primary, TWR_SER1(or TWR_SER2), TWR-ELVE Secondary, board
- Tower 2: Primary, TWR_SER1(or TWR)SER2), TWR-ELVE Secondary

The BSPCFG_ENABLE_ITTYB macro must be set to non-zero in the file user_config.h prior to compilation of MQX kernel libraries and the example itself. Because of UART1 is connected to an RS_485 transmitter. Prefer to read TWSER User's Manual to know about the jumper setting.

The RS485_TRANSMIT macro must be set to non-zero if board is transmitter, and must be clear if board is receiver.

The HAS_485_HW_FLOW_CONTROL macro must be set to non-zero if device has RTS (Request to send) pin hardware support. If not clearing it.

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

Explaining the example

The application example creates two tasks: rs485_write_task for transmitting data and rs485_read_task for receiving data.

The rs485_write task

The rs485_write task sends a data_buffer to rs485 channel. And rs485_read task reads data from this channel.

The rs485_write task starts to open rs485 channel. The code to do this depend on device supports RTS or not.

The example uses just 2 wires so rs485 standard just support half duplex. Receiver should be disables before transmitting.

- ioctl(rs485_dev, IO_IOCTL_SERIAL_DISABLE_RX, &disable_rx);

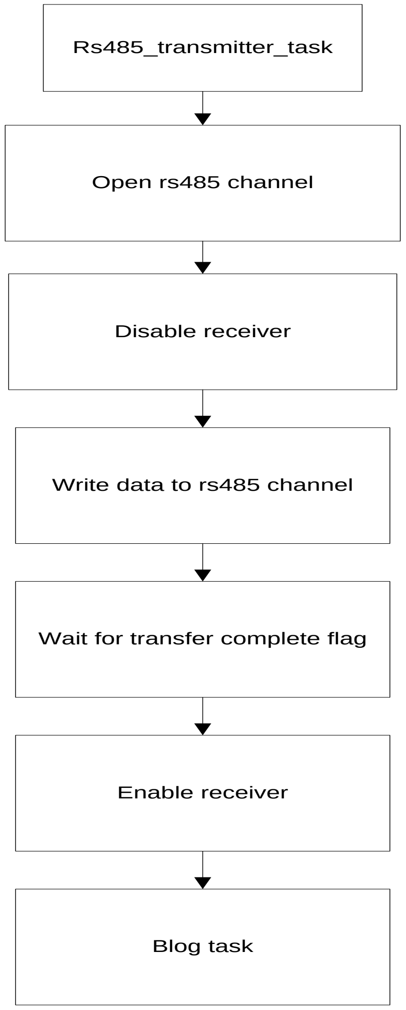
Then task writes a data_buffer to rs485 channel and waits for transfer complete flag.

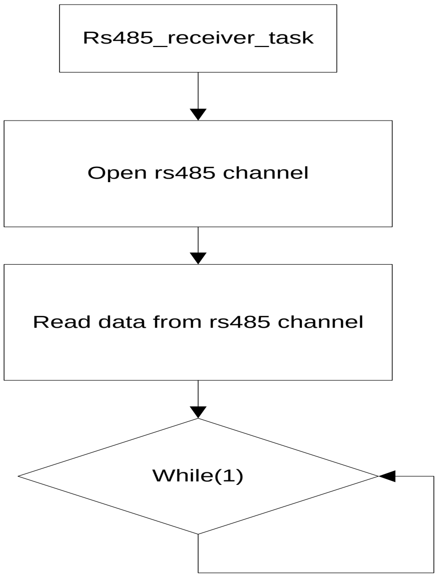
Finally task enables receiver which is disabled and blocks itself.

The rs485_read task

The res485_read task starts to open rs485 channel, then waits for data which is entered from terminal of tower 2.

Finally task goes infinite loop.


Rs485_transmitter_task
Open rs485 channel
Disable receiver
Write data to rs485 channel
Wait for transfer complete flag
Enable receiver
Blog task


Rs485_receiver_task
Open rs485 channel
Read data from rs485 channel
While(1)