

MOD-BT Bluetooth with DE0 Nano

Nigel Himmelreich

ECE 492 – Group 2 – Winter 2015

Introduction

The MOD-BT is a Bluetooth module by Olimex and allows serial communication between the Altera DE0 Nano and a Bluetooth connected device. The module connects to the DE0 Nano via the GPIO pins routed through the RS232 UART component provided [1]. The code provided uses the provided Altera RS232 library to extend it allow for strings of data to be written over Bluetooth. The setup for this module is based partially on winter 2014 G3 app note provided [2]. The code was created to interface with the provided Altera RS232 UART library [1] to simplify the task for setting up the module and sending data.

Setup

It is assumed in this setup that the user has already created a project in Quartus and created a system in Qsys much like completed in lab #1.

Hardware

1. Provide 3.3V and GND to pins 1 and 2 of the MOD-BT respectively (the numbers are printed on the PCB to help align the connector) [3]
2. Choose 2 GPIO pins and connect pins 3 (RXD) and pin 4 (TXD) of the MOD-BT. Note that the RXD pin (3) noted on the MOD-BT needs to be connected to the transmission side of the DE0 Nano (uart_0_external_interface_TXD) board, and the TXD pin (4) of the MOD-BT connected to the receiving side of the DE0 Nano (uart_0_external_interface_RXD).

Qsys

1. With the existing system, add the serial component. Located under, University Program -> Communications -> RS232 UART. Leave items as default but ensure the baud rate is set to 115200. Rename the component as desired.
2. Connect the following:
 - a. clock_reset to altpll_0.c1
 - b. clock_reset_reset to both clk_0.clk_reset and nios2_qsys_0.jtag_debug_module_reset
 - c. avalon_rs232_slave to nios2_qsys_0.data_master
 - d. Scroll to the right and connect the interrupt
 - e. Export the external_interface (double click the text under the export column)
3. Go to System-> Assign Base Addresses
4. Click on the Generate tab and generate the new design

Quartus

1. Edit the existing top level VHDL file to include the GPIO in the ports in the entity with:
 - GPIO_0 : inout std_logic_vector(33 downto 0) := (others => 'X')
2. In the top level, add to the niosII_system ports:
 - uart_0_external_interface_RXD : in std_logic := 'X'; -- RXD
 - uart_0_external_interface_TXD : out std_logic; -- TXD

3. Also in the top level, add to the niosII_system under the u0 component ports, where X is the GPIO pin number connected to pin 4 of the MOD-BT and Y is the GPIO pin number connected to pin 3 of the MOD-BT.
 - `uart_0_external_interface_RXD` => `GPIO_0(X)`, --`uart_0_external_interface.RXD`
 - `uart_0_external_interface_TXD` => `GPIO_0(Y)`, --`.TXD`
4. Compile the design and upload to the DE0 Nano board

Eclipse

1. Create a new project in Eclipse using the generated BSP from Quartus and using the MicroC/OS Hello World as the template.
2. Download and copy to the project directory the `btSerial.c` and `btSerial.h` found with this app note to the project folder.
3. (optional) You may need to rebuild the project includes index by right-click project folder, select index and then rebuild to find the rs232 library.

Software API

Below if the function calls created to allow for easier interfacing with the Bluetooth module as defined the `btSerial.c` & `.h` files.

- `singleWrite`
 - Inputs:
 - `alt_u8 c` Single character to be sent to the device
 - `alt_up_rs232_dev * serial_bt` Connection to device
 - Outputs:
 - none
 - Disables reading, checks for space in the device FIFO queue and writes the character to the queue to be sent to the device
- `multiWrite`
 - Inputs:
 - `char data[]` String to be sent to the device
 - `alt_up_rs232_dev * serial_bt` Connection to device
 - Outputs:
 - none
 - Disables reading and sends entire string character by character to the device using the `singleWrite` function
- `initDevice`
 - Inputs:
 - `char device[]` Device name found in `system.h` for the module
 - Outputs:
 - `alt_up_rs232_dev * serial_bt` Connection to device
 - Initializes the connection to the device with the given device name
- `setupModule`
 - Inputs:
 - `alt_up_rs232_dev * serial_bt` Connection to device
 - `char name[]` String for name of Bluetooth connection
 - `char pin[]` String for the pin for the Bluetooth connection

- Outputs:
 - None
- Sets up the module to enter command mode, sets the name, disables echo (module sending back any commands given), allows auto connections, sets the Bluetooth pin, and then starts the server to respond to connections

Sample Code

After following the above steps to setup the project, using the existing Hello World MicoC/OS example provided, the following code can be used in one of the default tasks created.

This program loops sending 0 to 100 to the Bluetooth connected device.

```
#include <stdio.h>
#include <stdint.h>
#include "includes.h"
#include "altera_up_avalon_rs232.h"
#include "altera_up_avalon_rs232_regs.h"
#include "btSerial.h"

/*
 * Task for the Bluetooth functionality
 *
 * Initializes the Bluetooth module and sends commands as needed.
 */
void task_bt(void* pdata) {
    printf("Start bt task\n");
    //Init the connection to the Bluetooth unit (MOD-BT) over serial
    alt_up_rs232_dev * serial_bt = initDevice(UART_0_NAME);

    //Where G2 M15 is the name and 2000 is the pin, change as necessary
    setupModule(serial_bt, "AT+BTLNM=\"G2 M15\"\r", "AT+BTPIN=\"2000\"\r");

    //Delay start of data from connection (adjust or remove with testing)
    printf("Wait for delay\n");
    OSTimeDlyHMSM(0, 0, 30, 0);
    printf("End of delay\n");

    //Testing, loop through and send 0 to 100
    int i = 0;
    char c[10];
    while (1) {
        sprintf(c, "%d\n", i);
        multiWrite(c, serial_bt);
        i++;
        OSTimeDlyHMSM(0, 0, 1, 0);
        if (i > 100)
            i = 0;
    }
}
```

Sources

[1] Altera, CA, USA. *RS232 UART Core for Altera's DE2/DE1 Boards* (2006). [Online]. Available: ftp://ftp.altera.com/up/pub/University_Program_IP_Cores/RS232.pdf Accessed on: March 1, 2015.

[2] K.Y. Lau, *Tutorial of Interfacing with RS232 UART*. University of Alberta, Edmonton, AB, CAN. [Online]. Available: https://www.ualberta.ca/~delliott/local/ece492/appnotes/2014w/G3_UART_Interface/G3_UART_Appnote_V2.pdf Accessed on: March 1, 2015.

[3] Olimex, Bulgaria. *MOD-BT development board*. (2009) [Online]. Available: <https://www.olimex.com/Products/Modules/RF/MOD-BT/resources/MOD-BT.pdf> Accessed on: March 1, 2015.