# RMIT University

## School of Engineering

### EEET2256 Introduction into Embedded Systems

### Major Project - Report

**Lecturer:**
Assoc. Professor Paul Becket

**Tutor:**
Raktim Kumar Mondol

**Laboratory Group:**
Tuesday 2:30-4:30 and Thursday 2:30-4:30, Group #1

**Submission Date:**
10-October 2017

**Students' Names:**
Callum Jones, Simon Coles, Ryan Prekop

**Students' Numbers:**
s3601120, s3401569, s3544394

# 1. Introduction:

The project our group has selected, is the RS232 Transmitter/Receiver project. It requires two OUSB boards, the first to act as a transmitter and the second a receiver. The transmitting board is required to scan a keypad, and if a key is pressed, send it over the serial port, to the second OUSB board, which will either display the binary version of the number on the LED's or display the number directly to a LCD screen (via serial).

Some constraints must be defined first. The transmitter board will only send one key at a time, and will implement a delay long enough, such that a key is only pushed once. If multiple keys are pushed down, the first key scanned, will be sent over.

The OUSB boards are a microprocessor development board made for the ATMEGA32 (designed by PJ Radcliffe, please visit: https://pjradcliffe.wordpress.com/ ). The ATMEGA32 has four IO (input/output) ports available for use. Each port has 8 pins available to it. On the OUSB board, each port has a different function. PORTA are ADC pins (analogue to digital converters), of which a couple of the pins have an LDR and Potentiometer connected. PORTB has an LED connected to each pin. PORTC has an 8-dip switch connected to it, while PORTD is used by the board for communicating to a PC.

The RS232 serial protocol is way of sending data across a single wire (not including the ground connection), or data in both directions using two wires (not including the ground connection). The key to this protocol is both the receiver and transmitter must have a pre-set speed known as the baud rate. The baud rate is the speed of data transferring over the link. For example, in this project a baud of 9600 will be used, which also represents 9600 bits/second. To find out how long it takes one bit to transfer, simply invert the baud rate: $\frac{1}{9600} = 0.104\ ms$

LCD

Once the serial connection is up and running, the next stage of the project is to use an RF module to transmit the serial data (ASCII codes) wirelessly.

# 2. Flow Chart (RX):
# 3. Flow Chart (TX):
# 4. Block Diagram:
# 5. Testing Process:

The testing process consisted of multiple stages. Each stage was a small piece of the larger project. For this project, it generally meant implementing each piece of hardware separately in software, before bringing it all together in the final software solution.

**Stage 1 (Individual Hardware Tests):**

The first step was to understand and program the 4x3 keypad. The testing process for this was rather simple: using a single OUSB board, individually push each key, and it should display the correct number on the LED's on PORTB. IMAGE.

At the same time, software was developed for transmitting serial (RS232 protocol) over PORTB pin 7. Rather than use the PC to decode the serial data, an Arduino Nano was set up as a receiver, which echoed results to the PC. The purpose of doing this was to prove, the programmed serial port, is working with external hardware, and because I know the Nano very reliable serial communication (hence why our serial code wasn't between the two boards to begin with). A success test would result in the pc showing letter 'C' appearing every 2-3 seconds or so. IMAGE

After the transmitting code was developed, some modifications were made to use PORTB pin 7 as a receiver for serial data (RS232 protocol). Much like before, the Arduino Nano was used as a transmitter, which was read by the OUSB PORTC pin 7 and displayed on PORTB LED's. A successful test would result in the ASCII code for each byte received show on PORTB. IMAGE

The same code used for transmitting serial, was also used to test the LCD screen, that is, instead of wiring the Arduino Nano to the transmission pin, the LCD RX pin was connected to it instead. Just like before, a successful test, will result in the letter 'C' appearing every 2-3 seconds or so, on the LCD.

**Stage 2 (Joining and Interfacing the Code Together):**

As mentioned in the introduction, the base project consists of two OUSB boards. The first is acting as a transmitter, while the second is the receiver. As the block diagram suggests, the transmitter must use both the transmission code, as well as the keypad code. Once these two pieces of code have been joined together and are interfaced properly, then a successful test will result in the program waiting for a key to be pressed, and sending the ASCII equivalent of the key pressed over the serial port. In our case, the Arduino Nano was used to test this. IMAGE

The receiver, in the base project doesn't need adjusted, and was already tested above. However, the next stage is to implement the serial receive from an external source, and serial transmit to an LCD screen. A successful test will result in the Serial data sent to the board, showing on the LCD screen. Just like before, an Arduino Nano was used to transmit data in ASCII to the OUSB board. IMAGE

**Stage 3 (The Final RX and TX Code):**

This is the final stage of the project, where the two pieces of code from the previous code will be uploaded to different OUSB boards, and tested. The test involves wiring up the receive pin from the receiver board to the transmit pin on the transmitter. A 2$^{nd}$ wire (ground) must also be connected between the boards for a reference. A successful test will demonstrate that when a key is pressed on the keypad, the corresponding number will show on PORTB of the 2$^{nd}$ OUSB board.