

Computer System Design Lab # 2

RS232 Signal Generation - Software

Amy Guo
Lab Partner: Charlie Coleman

January 28, 2018

Pre-Lab: N.A.

Objective: The objective of this lab was to program a microcontroller in order to interact with a Bluetooth radio.

Circuit Diagram: N.A

Outcome Predictions: We will be able to successfully communicate with a Bluetooth device using the BlueFruit Radio. We will analyze the output signals using an oscilloscope to identify data.

Equipment:

- BlueFruit BLE Radio
- Oscilloscope
- PC Terminal
- C8051F320 microcontroller

Procedure:

1. Connect the C8051F320 to the PC using the debugger
2. Using the provided code and instructions, setup the C8051F320
 - (a) Disable the watchdog timer so that the microcontroller does not restart regularly
 - (b) Need to implement the correct number of data bits, stop bits, set the parity bit
3. Upload the code
4. Connect the output of the C8051F320 UART to the PC UART to verify that the correct info is being sent
5. Send various commands to the BlueFruit
 - (a) Rename the device to something easily recognizable
 - (b) Add some different variables, edit their values, etc.
6. Check the Bluetooth connection on a cellphone to ensure the changes have gone into effect

Recalculations and Predictions: N.A

Data and Observations: The BlueFruit responded as expected. We were able to change the name on the BlueFruit easily so that the name would show up on our cellphones. When we added variables to the device and entered the value, we were able to see that info when using the nRF Connect app on a phone.

Analysis & Discussion: We saw lots of RS232 communication happen. We did not collect data for this lab, so there is no data to analyze. We saw the correct responses from the devices, so that was good. We were also able to make the changes we wanted to the device, so that was good too.

Lab Questions:

Q: How are PC baud rates historically generated?

A: Baud clocks were generated by dividing the master clock in the system

Q: How are baud rates generated in the C8051F320?

A: The baud rate is given by the equation $\frac{T1_{clk}}{256 - T1H} \times \frac{1}{2}$, where $T1_{clk}$ is the frequency of the clock supplied to timer 1, and T1H is the high byte of timer 1 (reload value)

Q: What is the percent difference in the PC baud rate and the baud rate of your designs?

A: The master clock on the C8051F320 is 12MHz, which can be divided evenly into 9600 baud. This means there is likely very minimal difference in PC baud and our design.

Q: What is the maximum allowed difference in baud rates at the rate you choose, demonstrate your answer with a plot, annotation, and text.

A: It is recommended to be within 1-2% of the desired baud rate, so at 9600 baud, we can be off by about 96-192. If we sample at the middle of each bit, we will be able to sample 25-50 bits before the drift will cause losses.

Results: Overall, the experiment worked as expected. We were able to observe changes in the BlueFruit configuration and look at the output of the C8051F320 using the PC.

Conclusions: This lab helped us to understand the configuration steps that are required to get multiple RS232 devices to communicate to each other. The lab helped us learn to use different software to implement code on microcontrollers, and familiarized us with different concepts like watchdog timers. These skills will be very helpful when we need to use any sort of microcontroller or UART in the future.