

# Computer System Design Lab # 5

## USB Signal Generation

Charlie Coleman  
Lab Partner: Amy Guo

March 5th, 2018

**Pre-Lab:** Not Applicable

**Objective:** The objective of this experiment was to print our names onto the console by requesting information from the USB device. We did this by using USBexpress APIs from SiLabs to create a USB device that would transfer our names to a requesting application.

**Circuit Diagram:** Not Applicable

**Outcome Predictions:** We expect to transfer the data for our names from the 8051 to the application.

**Equipment:**

- SiLabs
- USBexpress
- PC
- 8051

**Procedure:**

1. Test the SiLabs USB demo to ensure that the microcontroller is connected properly and all files needed are included in the project.
2. Modify the existing SiLabs USB demo to transfer some string across the connection.
3. Upload the code to the microcontroller using the JTAG debugger.
4. Connect the device to the computer via USB.
5. Write a C++ program that reads the output data from the microcontroller.

**Recalculations and Predictions:** N.A

**Data and Observations:** The string was transmitted successfully. With the default demo, the speed at which the strings were being transmitted/received was very slow compared to a theoretical USB connection, but each character was transmitted as desired.

**Analysis & Discussion:** No real data was collected during this lab. For the most part, it was based off of observations to make sure that the device was working as intended. The only data collected was for lab questions, and that data was surprising, as the character/second speed of the device was much lower than expected, but after some tweaking we were able to discover the reasons for this and correct them.

**Lab Questions:**

**Q:** Did you find the documentation supplied with the APIs sufficient to complete this task?

**A:** Yes, after we got used to interpreting and using the APIs, they were very useful.

**Q:** If not what additional documentation would you have found helpful?

**A:** Though the documentation was complete enough for our use, some more examples using the API would have been helpful.

**Q:** Is the C8051F320 a full speed device or low speed device and what is the theoretical and actual maximum throughput in characters per second of this class of device?

**A:** The C8051F320 can operate as either a full speed or low speed device. The theoretical characters/second of a full speed device is  $1.4 \times 10^6$  characters per second. The actual speeds that it is capable of is less than half of the theoretical, due to overhead and due to the fact that messages must traverse the bus twice.

**Q:** What is the maximum throughput you were able to obtain from the C8051F320?

**A:** With the default configuration, we were able to obtain speeds of around 3 characters/second. After some changes to the code, we were able to obtain 30 characters/second, a ten-fold increase.

**Q:** What factors contributed to the difference between what was obtainable and the maximum throughput?

**A:** Much of the difference is due to the overhead introduced by the API. Also, the demo that was being used had many points where it was waiting on things like ADCs that run much slower than USB is capable of. These things combined slowed down the USB connection significantly.

**Results:** Overall, the experiment worked as intended. The proper characters were sent across the USB connection and recognized by the computer. The transmission speed was much lower than expected, but that was common for all students and was due to factors that were outside of the scope of this lab exercise.

**Conclusions:** In this lab, we familiarized ourselves with different APIs for communicating with USB devices. This is important because, even if the API that is being used is different, a lot of the skills transfer easily. Being able to understand custom data types and pointers/references in API documentation is important for all types of devices, not just USB. Overall this lab was a success.