

Computer System Design Lab # 4

USB Signal Analysis

Charlie Coleman
Lab Partner: Amy Guo

February 5, 2018

Pre-Lab: N.A.

Objective: The objective of this experiment is to create 4 different types of USB communication packets to analyze using the oscilloscope. These signals will include Full speed and Low speed devices.

Circuit Diagram: N.A.

Outcome Predictions: We expect to capture and identify four different communication packets from USB devices in communication with our PC.

Equipment:

- Oscilloscope
- USB extension cable with pins exposed
- PC
- Keyboard
- Mouse
- Flash drive

Procedure:

1. Connect the oscilloscope to the D+ and D- pins of the USB cable provided.
2. Connect the female end of the wire to the device to be recorded.
3. Capture the resultant waveform on the oscilloscope
4. Repeat for the following packet types:
 - (a) Full speed hub
 - (b) High speed device plugged into a full speed hub
 - (c) Full speed function
 - (d) Low speed function

Recalculations and Predictions: N.A

Data and Observations: For this lab, we captured various USB signals and practiced decoding them. The signals were as follows.

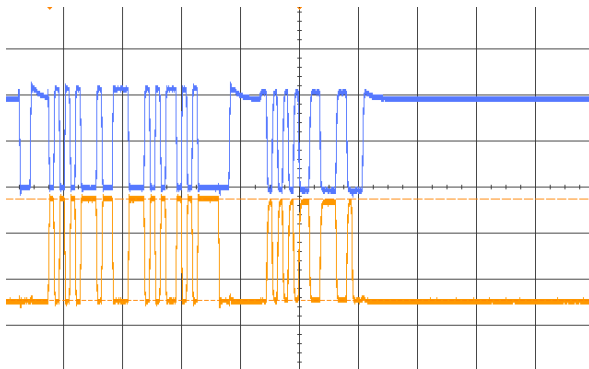


Figure 1: Packet from full speed hub

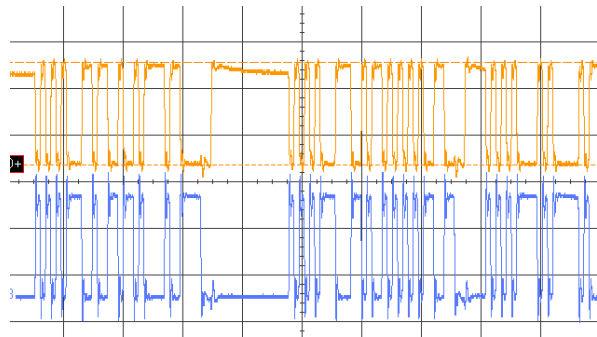


Figure 2: Packet from high speed device plugged into a full speed hub

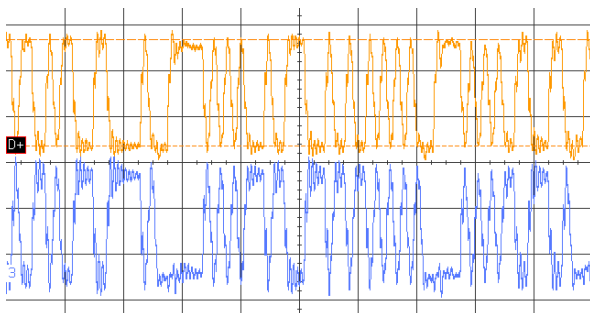


Figure 3: Packet from full speed function

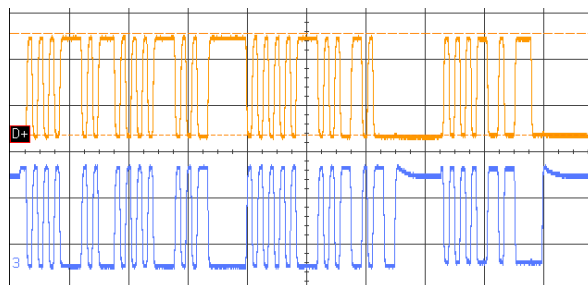


Figure 4: Packet from a low speed function

Figure 1: 0000 0001 0101 1010 SE0 - Handshake NAK

Figure 2: 0000 0001 1010 0101 0010 0011 1001 0111 SE0 - SOF

Figure 3: 0000 0001 0101 1010 SE0 - Handshake NAK

Figure 4: 0000 0001 0100 1011 SE0 - Handshake ACK

Analysis & Discussion: The waveforms matched our expectations, as the messages started with the requisite sync pattern, then went into the PID/data sections. We were able to clearly identify the start of frame and end of frame (single ended zero) easily in each of the waveforms captured.

Lab Questions:

Q: Describe in detail NRZI.

A: Non-Return to Zero Inverted code is the method of transmitting binary code used in USB signals. The signal is interpreted as a 0 when the state changes, and as a 1 when it remains constant.

Q: How can you tell if a function is a full speed or low speed, without looking at the waveform timing?

A: You can look at the state of D+ and D- during idle states. This is assuming that the waveforms are labeled. If we are also missing this info, it is not possible to identify the speed of the device.

Q: Describe what is meant by differential signaling.

A: Differential signaling is transferring data over a line using 2 data lines. Each of the lines is the inverse of the other when transmitting data. For example, in USB, D+ being high and D- being low means that a differential 1 is being sent across the line.

Q: Describe bit stuffing and why it is used by USB devices.

A: When a piece of data is being transmitted and it contains more than 6 ones in a row, the device must insert a 0 in order to keep the clock synchronized.

Q: What is the maximum throughput in data characters per second for a full speed device, discuss the design constraints that impact this.

A: For 8 bit characters, the throughput is approximately 1.4×10^6 characters per second. One design constraint is the amount of overhead per message / the maximum bytes per message. Minimizing overhead is important to maximizing speed of a channel. Full speed USB has fairly low overhead when the packet is using the maximum number of bytes, so this is what we assume for our calculations. Another important constraint is the frequency with which we send data. Increases in the frequency improve the speed directly.

Results: Our experiment was successful as the waveforms were successfully captured and the message types and connection speed were successfully identified.

Conclusions: In this lab, we learned to capture waveforms being transmitted over a USB line and were able to identify important characteristics of that data. This will be important in future projects as the intuition behind manually translating a USB signal can be applied to hardware and software design that involves these signals. Even though manually identifying USB signals is uncommon, it is important to understand how it is being done inside different components in order to debug problems or create new technologies.