Lab 6: USB Serial Communication
CSE 2100-001

Thomas Vu
John Jones

March 12, 2020

Date Performed:	March 05, 2020
Partners:	John Jones
	Thomas Vu

1 Objective

Program the Teensy 3.2 microcontroller with the packetized serial communication program on the class GitHub repository (serial_communication_variable.ino). Verify working bidirectional communication using the CuteCom terminal program on your Raspberry Pi.
Once you have your communications working correctly, modify the firmware (serial_communication_variable.ino) to extend the checksum from 8-bits to 16-bits (2 byte fields instead of 1). When generating the checksum, use the same cumulative XOR method, but perform using two bytes for each operand. For example, in the packet...

0xAA 0x07 0x01 0x02 0x03 [checksum]

the 16 bit checksum would be...

0xAA07 XOR 0x0102 XOR 0x0300 = A805

For payloads with odd numbers of bytes (such as above), use the last payload byte as the first (leftmost) byte and 0x00 as the second when performing the final XOR.
Demonstrate your modified packeting protocol with CuteCom using the test cases provided by the lab instructors.

1.1 Definitions

serial port An interface used for serial communication with devices such as terminals and various peripherals.

serial emulation Emulation of a standard serial port where software allows the creation of a port that has the ability to enable many serial ports without the need for additional hardware installation.

HID Human interface device such as a keyboard or mouse that takes input from humans and gives output to humans.

bulk transfer The method a software utilizes to move large data files using a variety of techniques such as compression and buffering in order to save bandwidth and optimize transmission speeds.

isochronous Describes processes that require timing coordination to be successful such as audio or video transmissions.

2 Question 1

Name 3 different standards for serial communication

Human interface device (HID), serial port data transfer (SATA) and serial emulation(USB).

3 Question 2

Suppose we transmit a packet and the final byte (the checksum) of the unmodified packeting strategy is lost by the receiver. Immediately after the transmission, another packet is sent and the receiver interprets the start byte of the 2nd packet as the checksum of the previous one. What are the odds that the receiver would incorrectly interpret the first packet as valid? What would be the odds for the modified (16-bit) protocol?

The XOR loop would have determined if there was an error in the checksum and therefore, the odds would be that the receiver would not incorrectly interpret the first packet as valid.