Lab 6: USB Serial Communication
CSE 2100-001

Apar Pokhrel

October 8, 2019

Date Performed:	October 01, 2019
Partners:	Apar Pokhrel
	Bivash Yadav

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 : a serial communication interface through which information transfers in or out one bit at a time

serial emulation :mimicking physical serial ports or old school serial ports through the use of virtual serial ports and emulator

HID : Human Interface Device -a class of peripheral devices that enables people to input data or interact directly with the computer, such as with a mouse, keyboard and other input devices.

bulk transfer : unidirectional communication using a large amount of data in any given time trhough a peripherial device such as a flash drive.

isochronous : transmitting real-time information such as audio and video data delivered at the desired rate

2 Question 1

Name 3 different standards for serial communication
RS-232, RS-422, and RS-485

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 modified protocol would be accepted.