Lab 6: USB Serial Communication
CSE 2100-001

Natnael Kebede

March 9, 2017

Date Performed:	March 02, 2017
Partners:	Natnael Kebede
	Asif Khan

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 : is an interface on a computer system which aids in information transfer (in or out) one bit at a time..

serial emulation : is a system that provides access to settings and status.

HID : is also known as Human interface device. it is a way by which a human interacts with an electronic information system by inputting data or providing output.

bulk transfer : is used to transfer large amounts of data. it also allows hardware error detection and involves a limited number of retries in the hardware.

isochronous : A sequence of events that occur regularly and take the same time period

2 Question 1

Name 3 different standards for serial communication

Asynchronous Synchronous Isochronous.

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 odds that the receiver would incorrectly interpret the first packet as valid is high. however, If the first packet has the same value of the checksum it would be valid. This means that the second package will contain an error since it is missing its checksum. Additionally, the odds for the modified 16-bit protocol are smaller than the first case.