Sekharan Natarajan, Dhaval Sonawane, Zhipeng Wei, Michael Xu

## (1) Format of header and packet that goes from Rasp Pi to Arduino Uno via LED and Photodiode

We have a series of 1's that make up the beginning and the end of the packet. The packet has 3 parts, a sequence number, followed by the data, followed by a checksum value.

Sequence number, the width is 8 bits; the checksum is 16 bits or 32 bits based on it's keystroke or file. The width of payload in each packet is 128 bits.

| Sequence number | Payload | Check sum |
|---|---|---|
| 8 bits | 128 bits or 64 bits | 16 bits or 32 bits |

## (2) Format of ACK that goes from laptop # 2 to Rasp Pi via TCP connection over WiFi

The ACK sends one byte (8 bit) integer over to Rasp Pi. This integer represents the packet number that we want the Rasp Pi to send and also serves as the ACK message.

## (3) How you modulate the LED to transmit a 0 or a 1

We have set Rasp Pi to leave the LED one or off for an interval of 100ms. We did this using Java code. To help out with clock drift, we actually leave the LED on for however long we have left in the time interval after the code has run, so it may actually vary depending on how fast the code runs, but the interval is still the same.

## (4) How do you synchronize the Rasp Pi and Arduino

We set both the Rasp Pi and Arduino to the same value (100ms) and when clock drift is detected through the packet checksum, we re-send the packet. Before sending actual packet, a sequence of 1 is sent. When the Arduino receives this data, it will extract the packet from the stream of data based on this sequence of 1.

## (5) How long do you take to transmit a bit

100ms, not counting any packet header/footer, packet sequence number, and packet checksum.

## (6) Explanation of the scheme you use to build reliability.

We implement ACK and checksum to confirm reliability.
Each packet contains a checksum. In case that some errors appear, these errors will be detected. Whenever a packet is received, PC2 will re-calculate the checksum of this packet of the data and compare this checksum with the checksum extracted from the packet, based on the comparing result, an ACK message is sent from PC2 to raspberry pi.