Ashot Kulikyan

PID: A11310640

**Blink Test and Basic chat**

**Lab 1**

**BlinkTest:** Blink Test was tested by downloading Arduino tools and FTDI drivers which allows to upload software from pc to ATmega128RFA1.

**Basic Chat Test:**

To measure the amount of data that can be sent in one second from each board, I sent 127 bytes at a time for one second and measured the amount of data that was sent from each board. The amount of data for the boards is the following:

Board #1: 11684 Chars/sec=11684\*10 bits/sec = 116.84 Kb/sec

Board #2: 11684 Chars/sec=11684\*10 bits/sec = 116.84 Kb/sec

Next, I send 18 bytes at a time in 1 second and measured the amount of bytes sent in one second; the number 18 bytes at once is the amount that the receiver could properly receive and output. When I sent 19 bytes at the time for 1 sec the receiver would output less data than sent.

One way communication: 10638 Char/sec (1Char=1byte=8bits)

10638 Char/sec does not include the 2 bits for start and end of the packet. If those bits are included in the data transfer then the amount of data that the TRX sends is 10638\*10 bits/sec=106380 bits/sec

**Total bits send in one way communication is 106380 bits/sec= 106.38 Kbits/s**

When I increased the distance between two boards to about 20 meters the amount of data sent was the same. However, the receiving end only got 7110 Chars/sec=7110\*10bits/sec which means that 3528 Chars were lost because of the distance. 100% of the packets was received in less than ~10m range.

For two way communication I sent some data and once the receiver board gets some data it sends the same amount of data back. However, each time I ran the system I would get different number of data back as a result of two way communication because the second board gets the data and when it tries to send the data there is another packet coming in from the first board so it gets rewritten. This means that the board cannot send and receive data at the same time. It has to either send or receive. When I made some delay between receiving and sending, both boards could successfully send and receive data with 160.38 Kbits/s.

I used Word Document’s character counter to get the number of characters sent and received so I can overcome overwriting issues as well as to keep the code very simple.