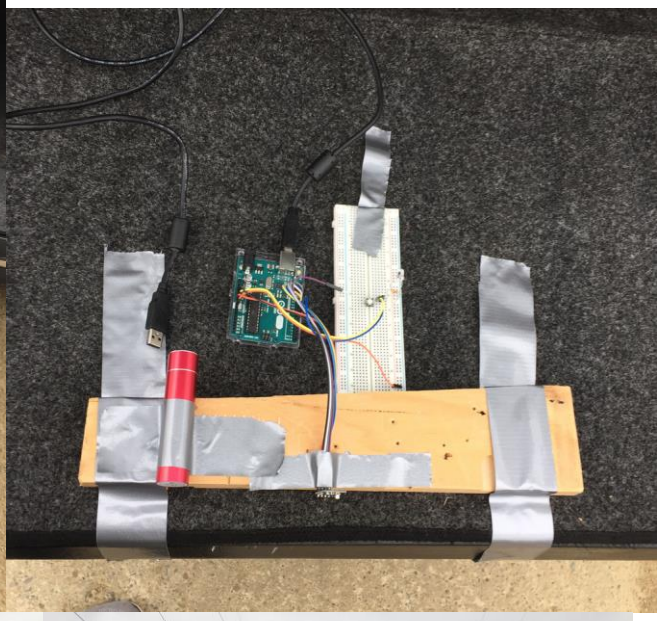
nRF24L01 Range Test

3/17/17 –Tim Lindquist

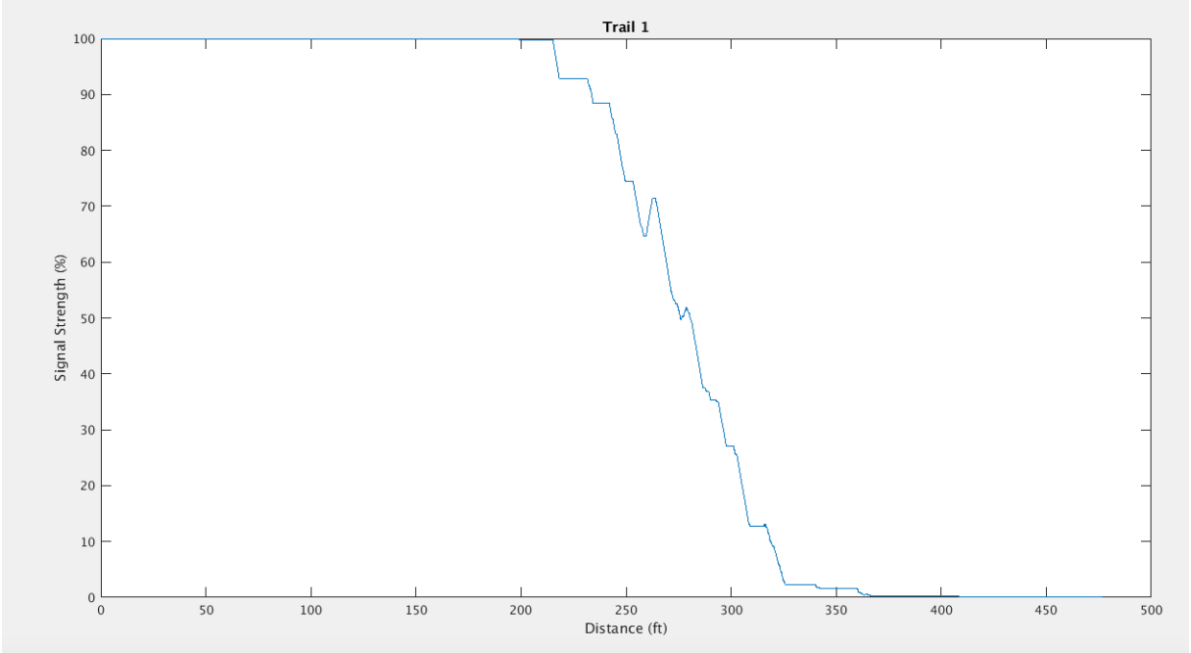
```
#include <Arduino.h>
#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"

int SUM=500;

long msg[1];
RF24 radio(9,10);
const uint64_t pipe = 0xE8E8F0F0E1LL; //channel to recieve
int LED1 = 3;
int i=0;
int j;
int data[500]={0};     //array size must be same as SUM size
double avg;


void setup(void){
   Serial.begin(9600);
   radio.begin();
   radio.openReadingPipe(1,pipe);
   radio.startListening();
   pinMode(LED1, OUTPUT);}

void loop(void){
   if (radio.available()) {
      bool done = false;
      while (!done) {
         done = radio.read(msg, 4);  //byte value
         if (!done) {
            //Serial.println(msg[0]);}
         }
         avg = 0;
         data[i] = 1;
         i++;
         if (i > (SUM-1)) {     //check if i>SUM-1
            i = 0;
         }
         for (j=0;j<SUM;j++) {
            avg += data[j];
         }
         Serial.print(avg/5);
         Serial.print(", ");
      }
   }
   else{
      //Serial.println("No radio available");
      avg=0;
      data[i]=0;
      i++;
      if(i>(SUM-1)){
         i=0;
         }
      for(j=0;j<SUM;j++){
            avg+=data[j];
         }
      Serial.print(avg/5);
      Serial.print(", ");
      }


}
```

Program checks the number of correctly received packages and displays the data as a signal strength percentage. 500 data points are collected and stored in an array, if a package is correct, that array index is 1 and if it is incorrect or missing the array index receives a 0. After which a continual averaging system sums the entire array and divides by 5 to get a number between 0 and 100 corresponding to percentage. The array is incremented and reset upon overflow. A size of 500 was chosen to create a large enough window for some stability during the discrete time averaging. The program finally outputs the percentage value to be displayed over serial interface.

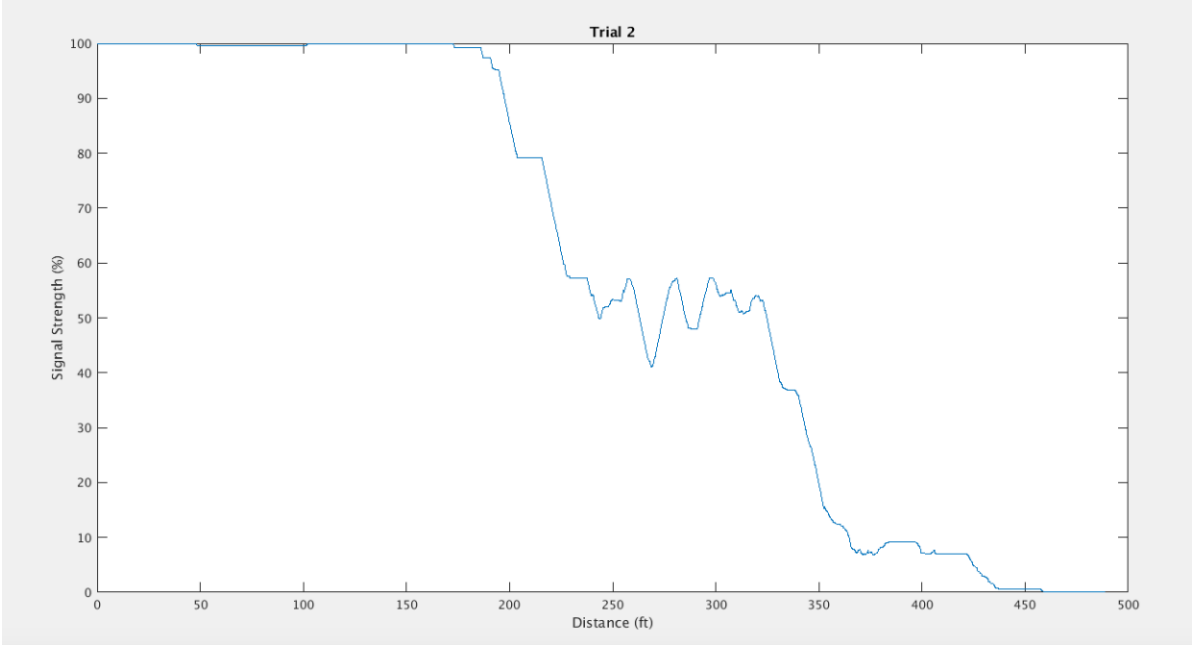
Testing was done by going to a rural road in an open area. The transmitter was attached to the gate of the truck and the receiver was mounted on a stationary table in the road. The initial starting point was recorded using GPS coordinates. Communication with the driver was done over cell phone. The test began by plugging in the transmitter and starting the serial logging of the signal strength data. At that point the truck drove forward at a constant 10 mph. When the data zeroed out, I signaled the driver to stop as well as stopping the data log. The end point was recorded using GPS coordinated. A graph was constructed of the signal strength vs distance by determining the distance between the two points and then mapping the incremented values to the data points.



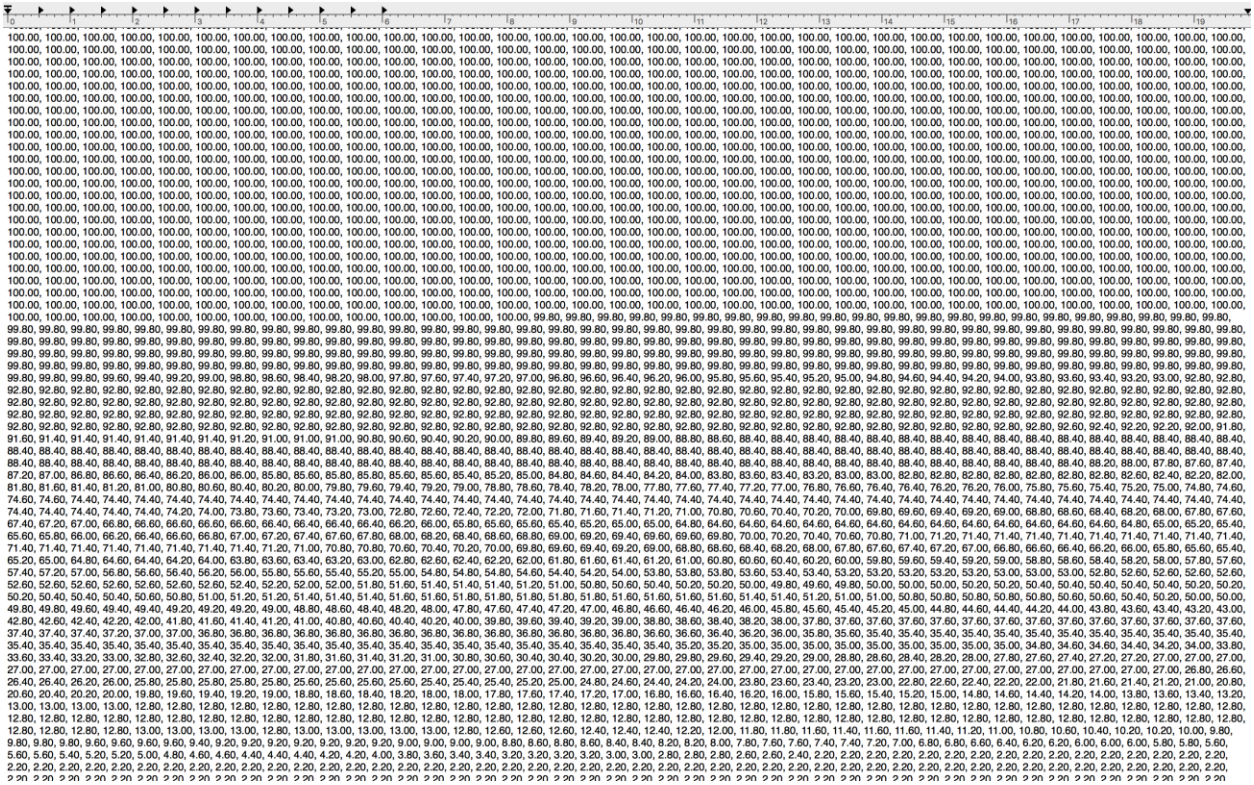
Pictures above show the testing site. Data is shown below. To note, during Trial 2 a truck drove by during the test and can explain the discrepancy during the fall out.



Trial1:
(Deg,min,sec)
Start: 44°05'38.2"N 92°33'45.3"W
End: 44°05'38.2"N 92°33'51.8"W
Distance=477ft 145m



Trial2:
(Deg,min,sec)
Start: 44°05'38.2"N 92°33'45.2"W
End: 44°05'38.2"N 92°33'51.9"W
Distance=489ft 149m



Above shows a sample of what the collected data looks like. In total there were over 10,000 data points per trial.