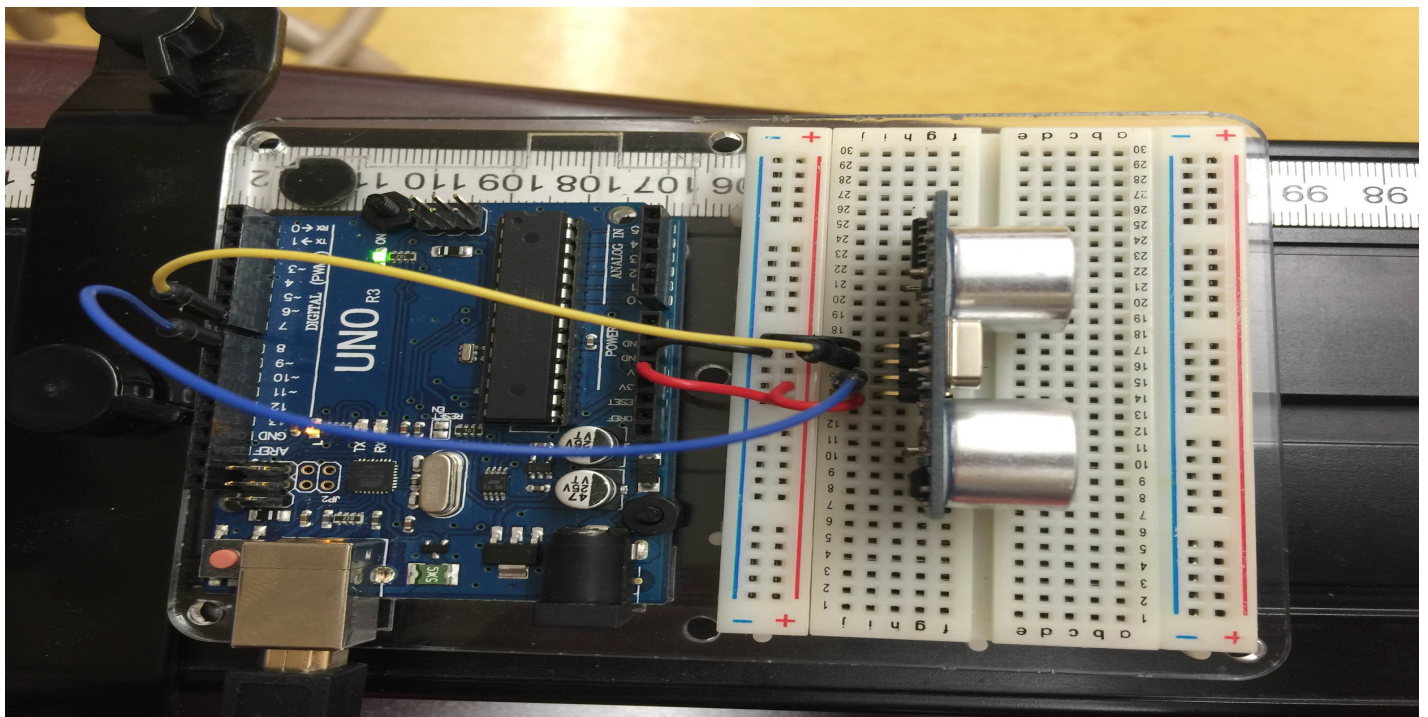
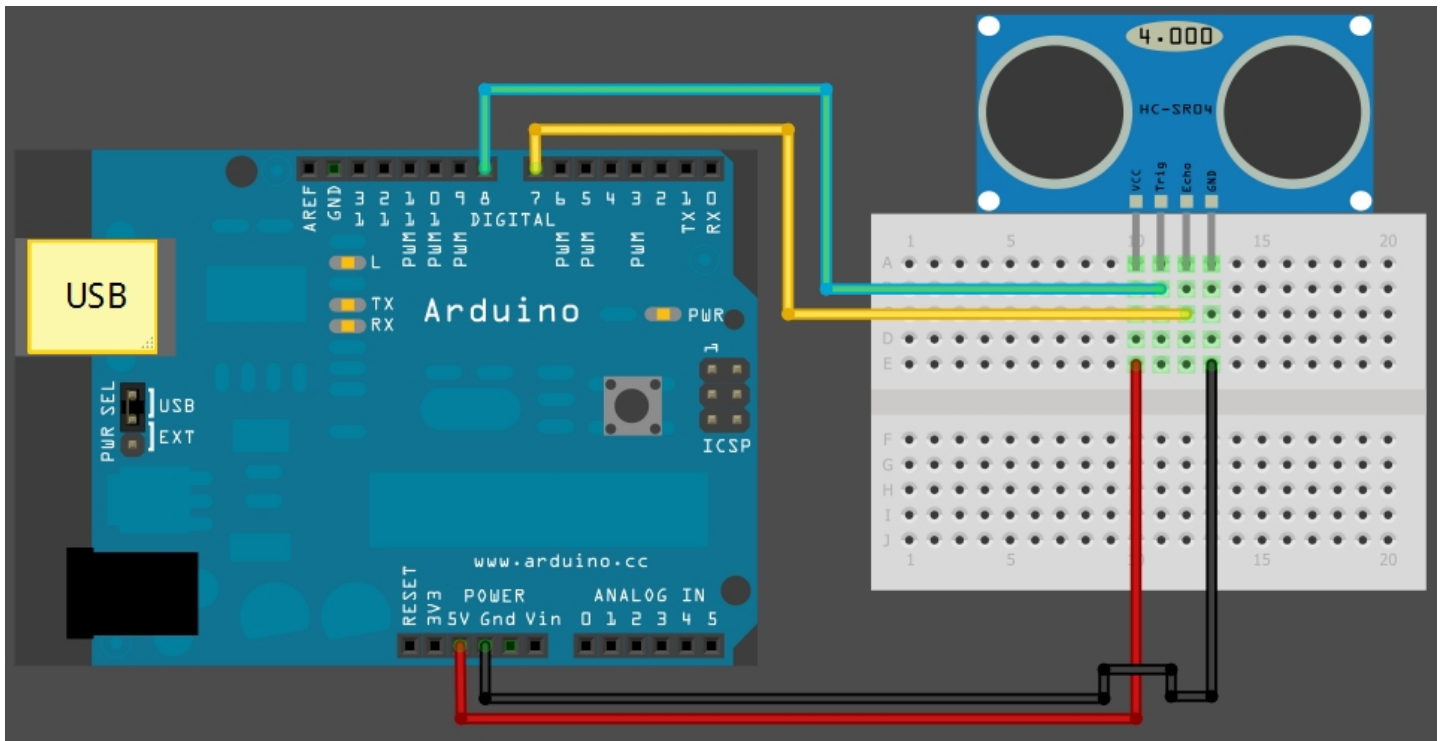


Using an Arduino and Ultrasonic Sensor for Data Acquisition

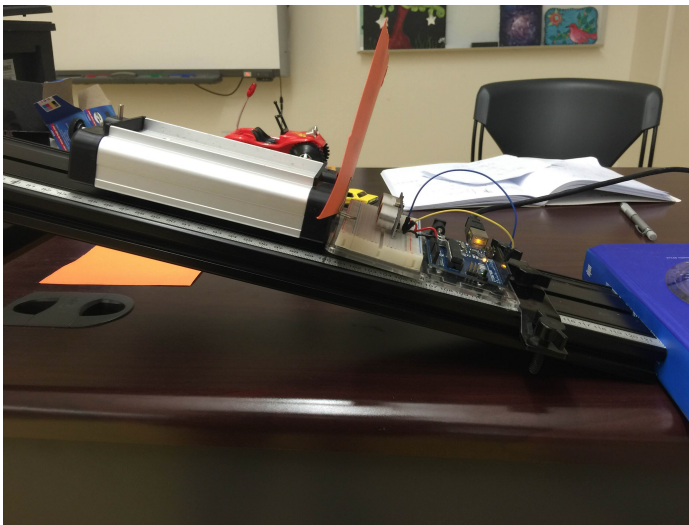
This investigation is all about getting data from the real world and turning that into a plot with your group.

Step 1: building the circuit. Follow the Fritzing diagram below and connect up the breadboard, the Arduino board, and the ultrasonic sensor. The color of the leads (wires) doesn't matter. Note the red connects to the 5V on the Arduino board and the black connects to the Gnd or ground. This is where the sensor gets it's power to send ultrasonic signals. The leads in pins 7 and 8 use the code on the Arduino board to actually operate the sensor and gather data.



Step 2: coding. Open [Rangefinder.ino](#) and copy and paste the code into the Arduino application. If you are using a student laptop, you will have to install the Arduino application from the software center. Otherwise you can [download and install](#) it yourself. Make sure the Arduino board is plugged into your computer and that you have the correct board selected in the Arduino application and that the correct port for your board is selected. Both of these settings are in the Tools menu. You are probably using an Arduino Uno if you're not sure.

```
RangeFinder
24 long duration, distance; // Duration used to calculate distance
25 int t = 0;
26 const int wait = 50;
27
28 void setup() {
29   Serial.begin(9600);
30   pinMode(trigPin, OUTPUT);
31   pinMode(echoPin, INPUT);
32   pinMode(LEDPin, OUTPUT); // Use LED indicator (if required)
33 }
34
35 void loop() {
36   /* The following trigPin/echoPin cycle is used to determine the
37      distance of the nearest object by bouncing soundwaves off of it. */
38   digitalWrite(trigPin, LOW);
39   delayMicroseconds(2);
40
41   digitalWrite(trigPin, HIGH);
42   delayMicroseconds(10);
43
44   digitalWrite(trigPin, LOW);
45   duration = pulseIn(echoPin, HIGH);
46
47   //Calculate the distance (in cm) based on the speed of sound.
48   distance = duration / 58.2;
49
50   if (distance >= maximumRange || distance <= minimumRange) {
Done compiling.
```



Step 3: build the experiment. You will need a flat open space to put your track and cart and sensor/Arduino/computer. The floor works just fine. Put the sensor circuit, which should be connected via a breadboard, at one end of the track. The circuit must be connected to the Arduino and the Arduino plugged into the computer. Make sure the program is working correctly by opening the Serial Monitor under the Tools menu in the Arduino application. Be sure to note how far the sensor is from the front of your cart! Do the values make sense if you place something in the path of the sensor? Once you are satisfied all is working, collect data.

Step 4: data gathering. With the Serial Monitor open and working as expected, put the cart at the other end of the track and give it a gentle push. **Don't ram the cart into the sensor but push hard enough to move about a meter or so.**

You should see time and distance data on the Serial Monitor window.

Uncheck the Autoscroll option and locate the time interval you care about.

You may need to experiment a bit to make sure you understand the data you are seeing.

What are the distance units? What are the time units? You will need 2 sets of data.

One with the cart flat and one with the cart at an angle around 15 degrees.

Use some simple geometry to determine your particular angle if needed.

```

/dev/cu.usbmodem14131 (Arduino/Genuino Uno)
18100 82
18150 82
18200 82
18250 82
18300 81
18350 80
18400 79
18450 76
18500 73
18550 69
18600 66
18650 59
18700 53
18750 47
18800 39
18850 31
18900 23
18950 14
19000 9
19050 6
19100 5
19150 4
19200 4
  
```

Step 5: plot and fit the data. Grab a set of data for a cart on a flat surface and then copy and paste that data into your spreadsheet of choice. Google Spreadsheets or Microsoft Excel are fine. Then select time as x and distance as y and add a scatter plot. Next add a trendline to the curve. What sort of fit seems to make sense for the given data? The two datasets may have different fits that make sense. You are to submit (as a group) links to your datasets and your plots.

	A	B
1	t (s)	x (m)
2	18.25	0.07589460785
3	18.26530612	0.07748621096
4	18.28061224	0.07966219505
5	18.29591837	0.08242256013
6	18.31122449	0.08576730619
7	18.32653061	0.08969643324
8	18.34183673	0.09420994127
9	18.35714286	0.09930783028
10	18.37244898	0.1049901003
11	18.3877551	0.1112567512
12	18.40306122	0.1181077832
13	18.41836735	0.1255431961
14	18.43367347	0.1335629901