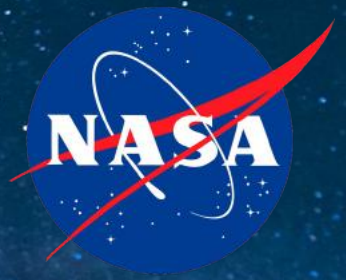
HiBall Balloon Payload Workshop

Sensors Part 2



Partner

Part 1 – Arduino Test Drive Sensors

A. LED Visual Display
B. Analog vs. Digital
C. Balloon Shield Build
D. Thermometer

Part 2 – Arduino Road Trip Sensors

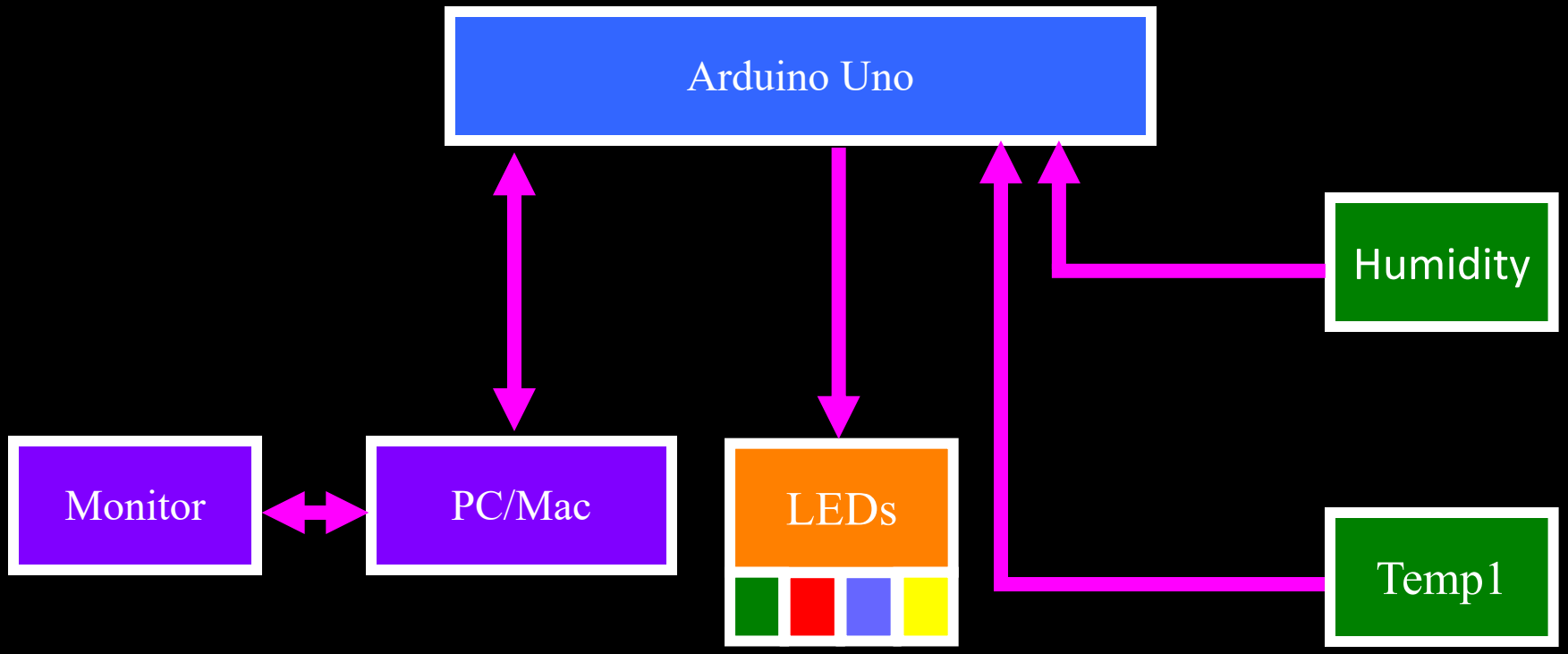
A. Humidity Sensor
B. Pressure Sensor
C. Accelerometers
D. External Temp Sensor

Part 2 – Arduino Road Trip Sensors

A. Humidity Sensor
B. Pressure Sensor
C. Accelerometers
D. External Temp Sensor

Humidity Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER



Humidity Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

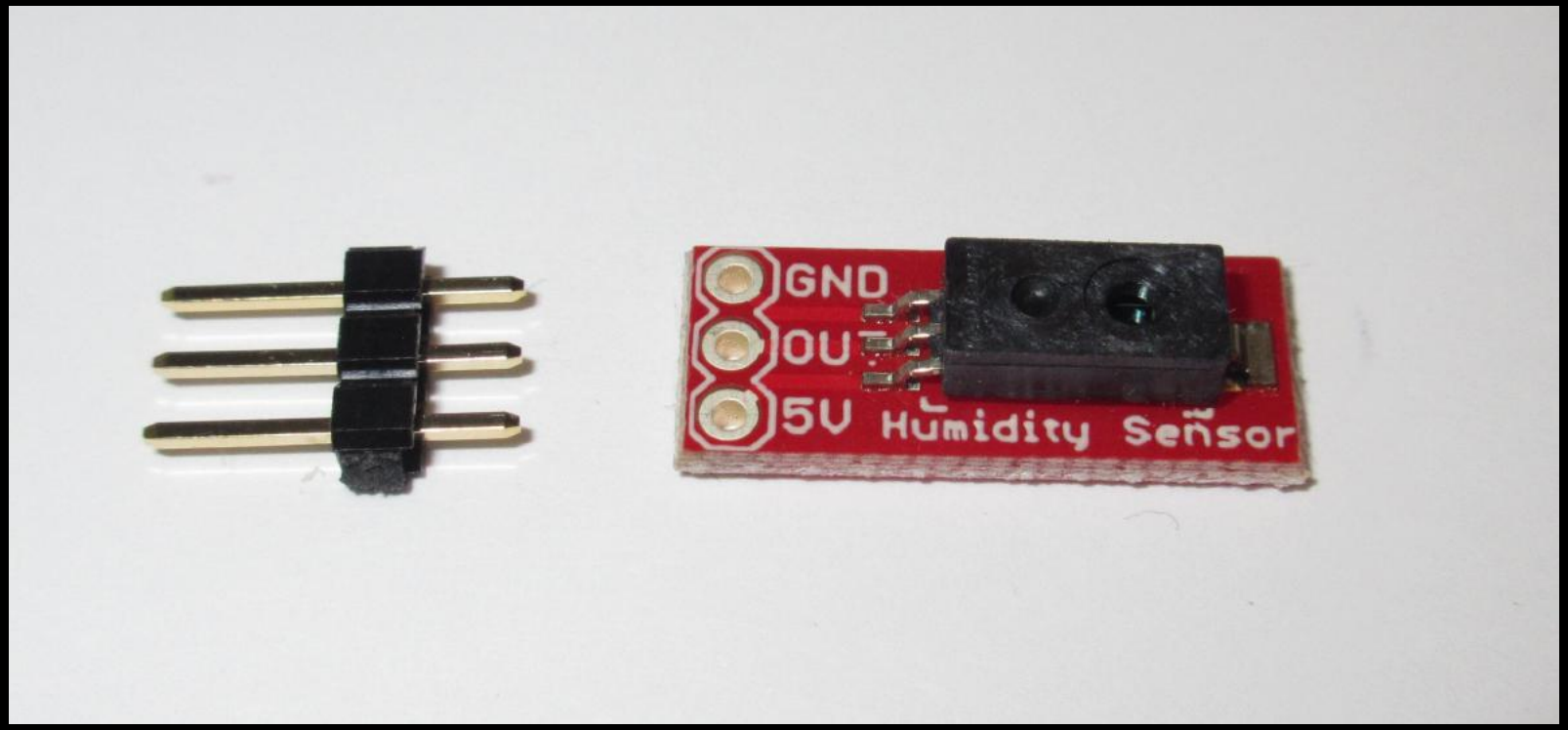
- Humidity sensor (or the Darth Vader Sensor)

- It measures moisture in the air, which is great for balloon flights (condensation failures)

Humidity Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

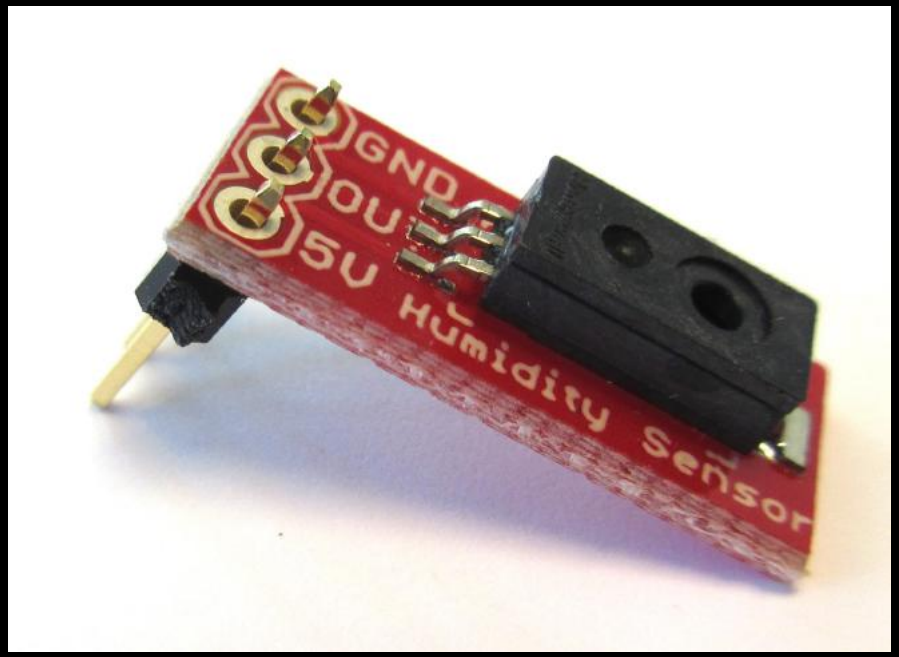
- First need to solder header to sensor

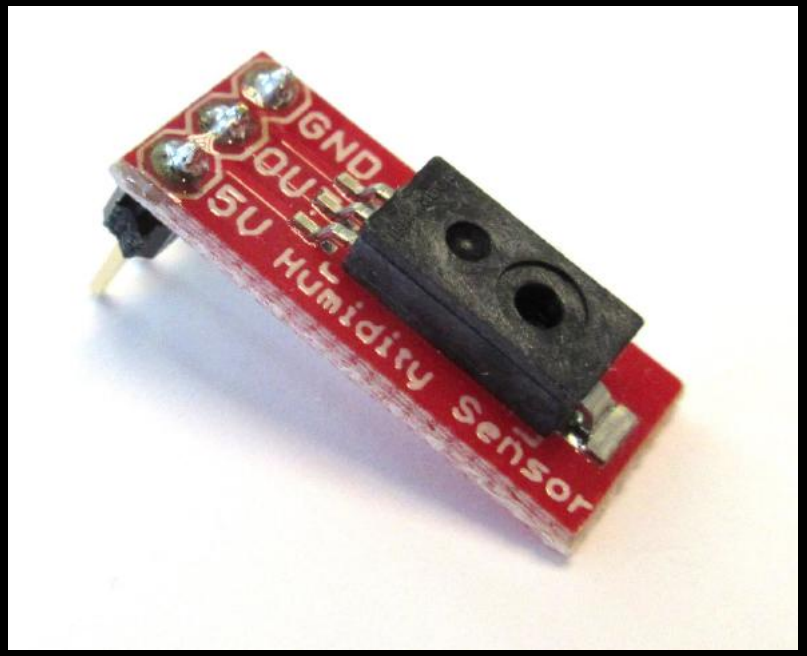


Humidity Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Install header like shown and solder from top of board
- Short side through the bottom of the board
- Keep header perpendicular to board





Humidity Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

Leave your Balloon Shield attached to Arduino

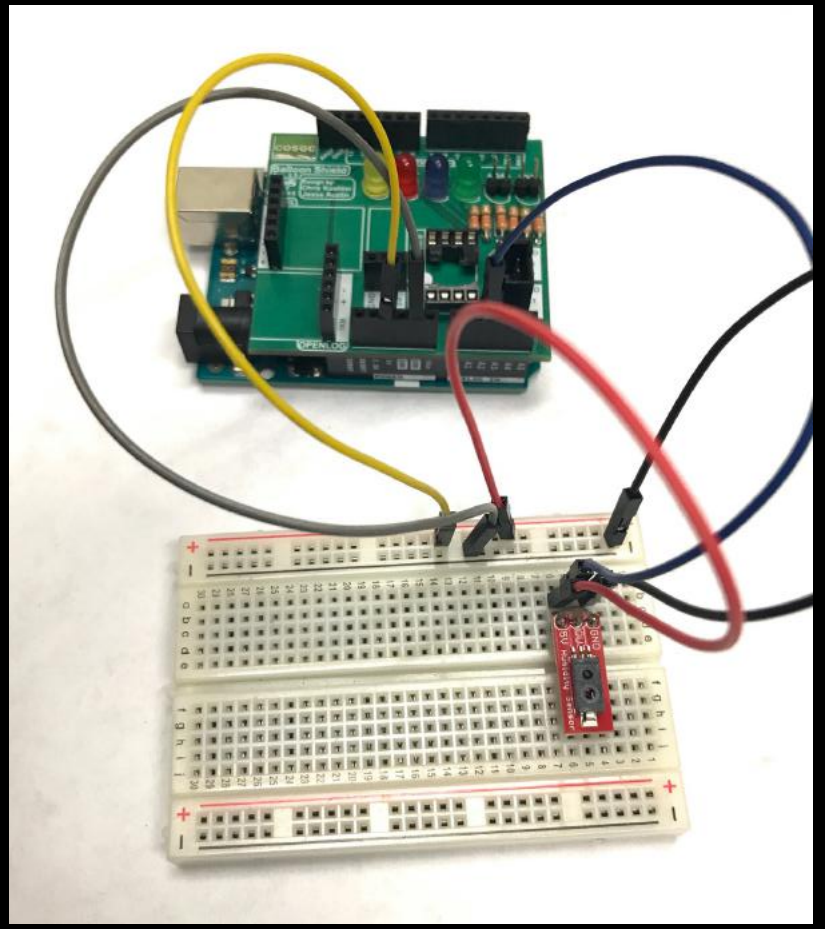
- Wire **Arduino 5V** to Breadboard **(BB) 5V PWR Rail**

- Wire **Arduino GND** to **BB GND Rail**

- Wire **Sensor 5V** to **BB 5V Rail**

- Wire **Sensor GND** to **BB GND Rail**

- Wire **Sensor OUT** to **Arduino A2**



Humidity Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

Leave your Balloon
Shield attached to Arduino

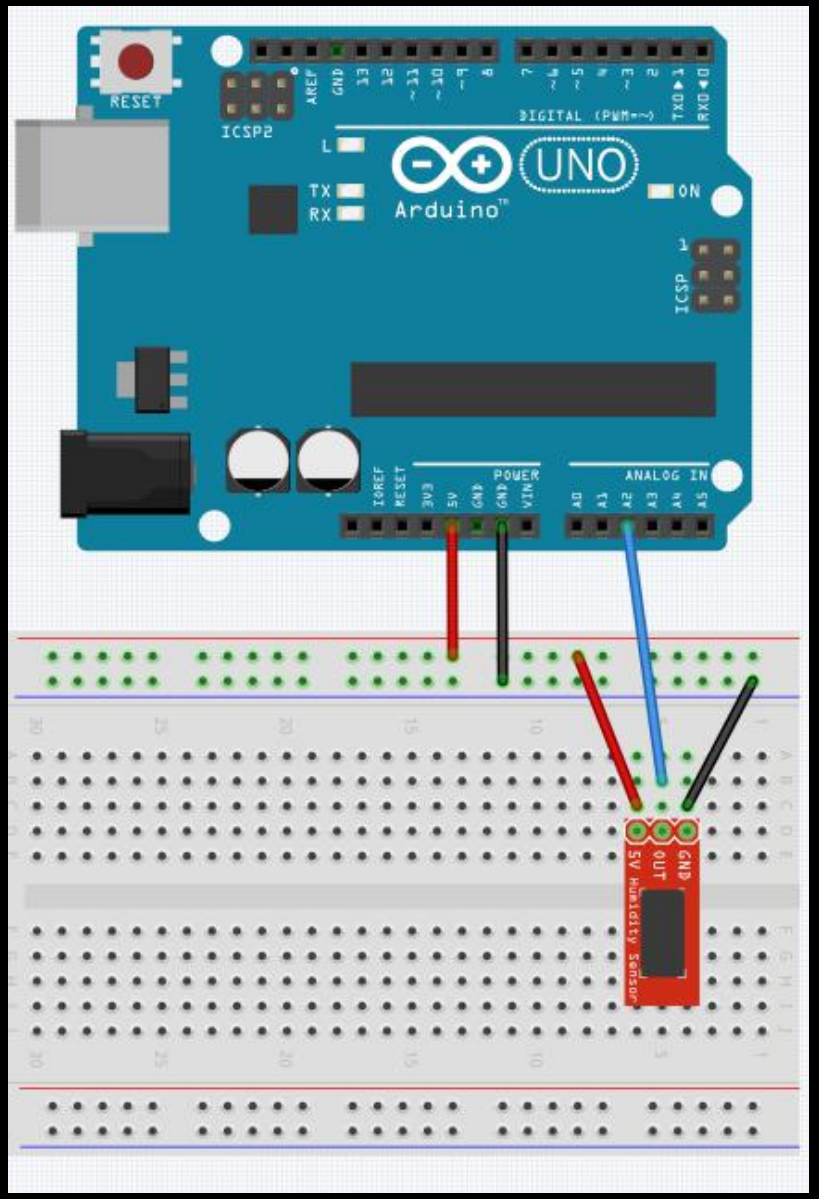
- Wire **Arduino 5V** to Breadboard **(BB) 5V PWR Rail**

- Wire **Arduino GND** to **BB GND Rail**

- Wire **Sensor 5V** to **BB 5V Rail**

- Wire **Sensor GND** to **BB GND Rail**

- Wire **Sensor OUT** to **Arduino A2**



Humidity Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

Leave your Balloon
Shield attached to Arduino

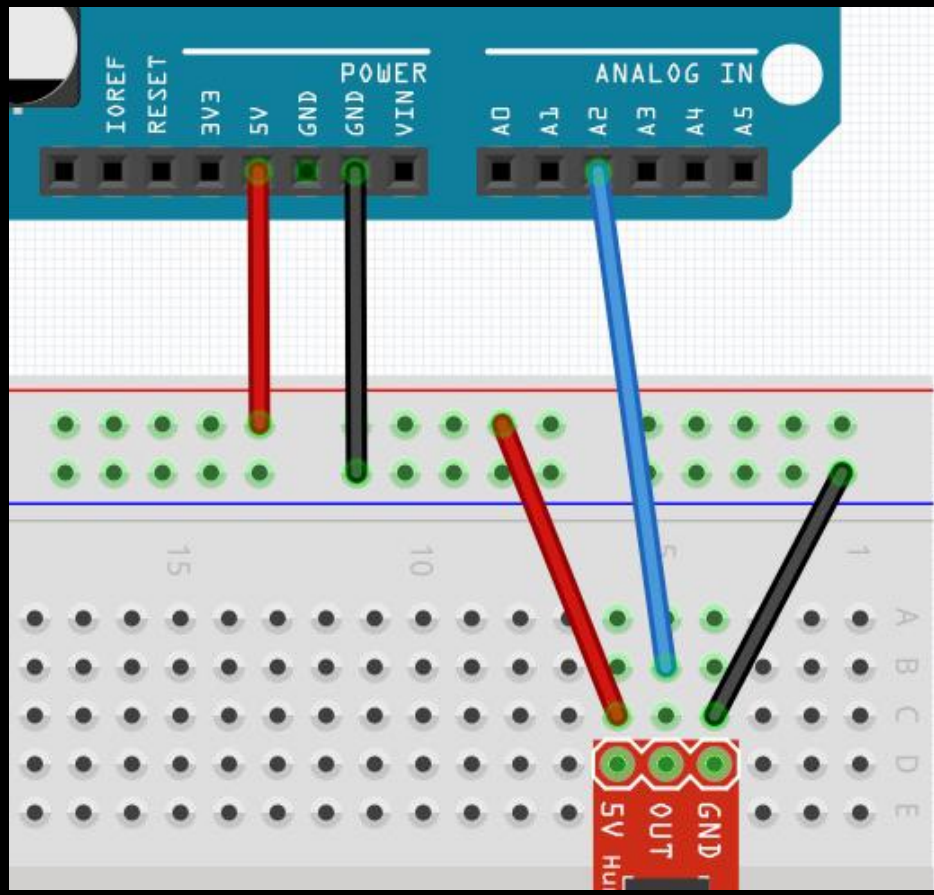
- Wire **Arduino 5V** to Breadboard **(BB) 5V PWR Rail**

- Wire **Arduino GND** to **BB GND Rail**

- Wire **Sensor 5V** to **BB 5V Rail**

- Wire **Sensor GND** to **BB GND Rail**

- Wire **Sensor OUT** to **Arduino A2**



Humidity Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Modify sketch to read new sensor on A2

```
// Definitions
      int sensor;
      float sensorVolt;
```

```
void loop() {
//    put your main code here, to run

      sensor = analogRead(A2);
      sensorVolt = sensor*(5.0/1023);
      Serial.print(sensor);
      Serial.print("\t voltage ");
      Serial.println(sensorVolt);

//    Turn script running leds OFF at begir
      digitalWrite(5, LOW);  //Green LED
```

```
if(sensorVolt > 1.25) {
digitalWrite(5, HIGH);
}
if(sensorVolt > 1.75) {
digitalWrite(6, HIGH);
}
if(sensorVolt > 2.25) {
digitalWrite(7, HIGH);
}
if(sensorVolt > 2.75) {
digitalWrite(9, HIGH);
}
delay(100);
```

EXTRA

Humidity Sensor:

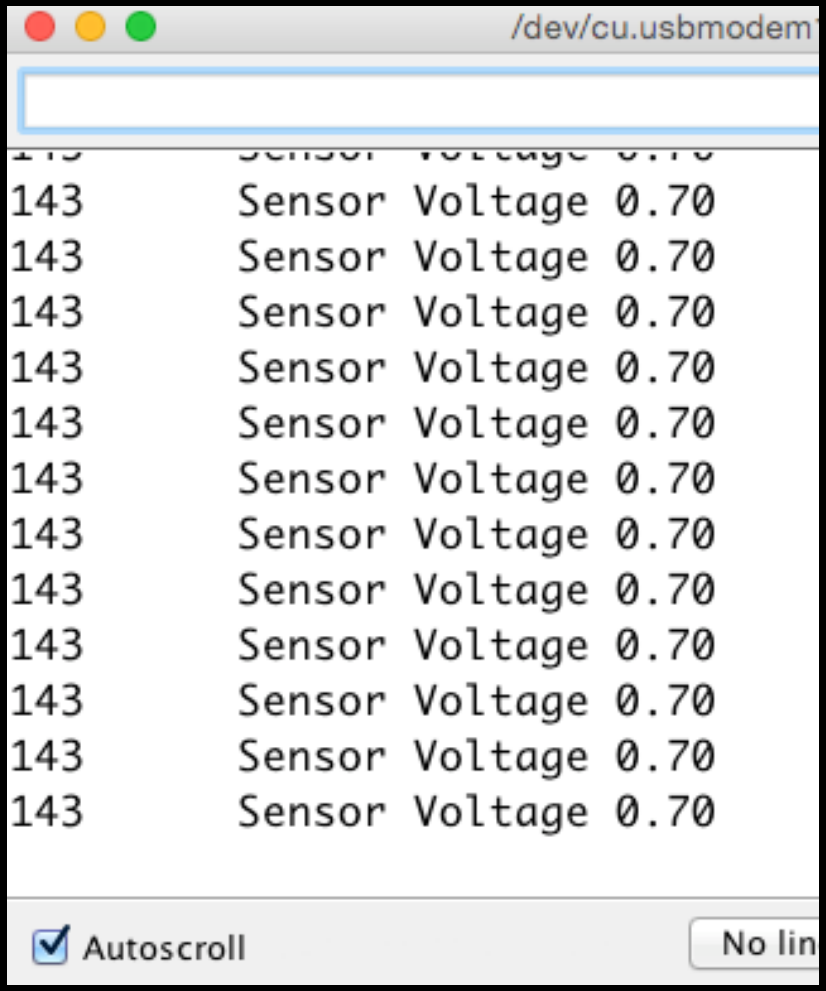
Space Minor
UNIVERSITY OF COLORADO BOULDER

- Compile and Upload

- Start Serial Monitor

- Breathe on humidity sensor like Darth Vader

- Watch LEDs on Shield



- Next, let's convert volts to % humidity

Humidity Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Look at the data sheet to understand output of the sensor

- We know Vout and Vsupply so using algebra


Honeywell

HIH-4030/31 Series

Humidity Sensors

DESCRIPTION

Honeywell has expanded our HIH Series to include an SMD (Surface Mount Device) product line: the new HIH 4030/4031. The HIH 4030/4031 complements our existing line of non-SMD humidity sensors. SMD packaging on tape and reel allows for use in high volume, automated pick and place manufacturing, eliminating lead misalignment to printed circuit board through-hole.

The HIH-4030/4031 Series Humidity Sensors are designed specifically for high volume OEM (Original Equipment Manufacturer) users.

Direct input to a controller or other device is made possible by this sensor's near linear voltage output. With a typical current draw of only 200 µA, the HIH-4030/4031 Series is often ideally suited for low drain, battery operated systems.

Tight sensor interchangeability reduces or eliminates OEM production calibration costs. Individual sensor calibration is available.

The HIH-4030/4031 Series delivers instrumentation-quality RH (Relative Humidity) sensing performance in a competitively priced, solderable SMD.

The HIH-4030 is a covered integrated circuit humidity sensor.

The HIH-4031 is a covered, condensation-resistant, integrated circuit humidity sensor that is factory-fitted with a hydrophobic filter allowing it to be used in condensing environments including industrial, medical and commercial applications.

The RH sensor uses a laser trimmed, thermoset polymer capacitive sensing element with on-chip integrated signal conditioning.

The sensing element's multilayer constru... excellent resistance to most ... condensation, dust, ... chemicals.

FEATURES

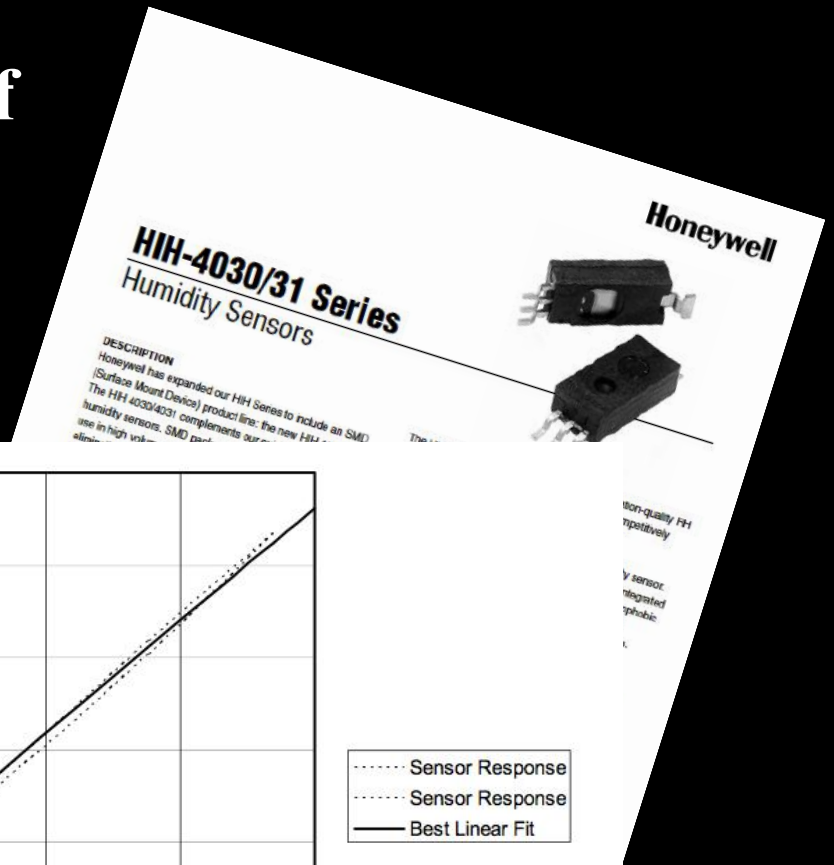
Voltage output (1st order curve fit)	$V_{OUT}=(V_{SUPPLY})(0.0062(\text{sensor RH}) + 0.16)$, typical at 25 °C
Temperature compensation	True RH = (Sensor RH)/(1.0546 – 0.00216T), T in °C

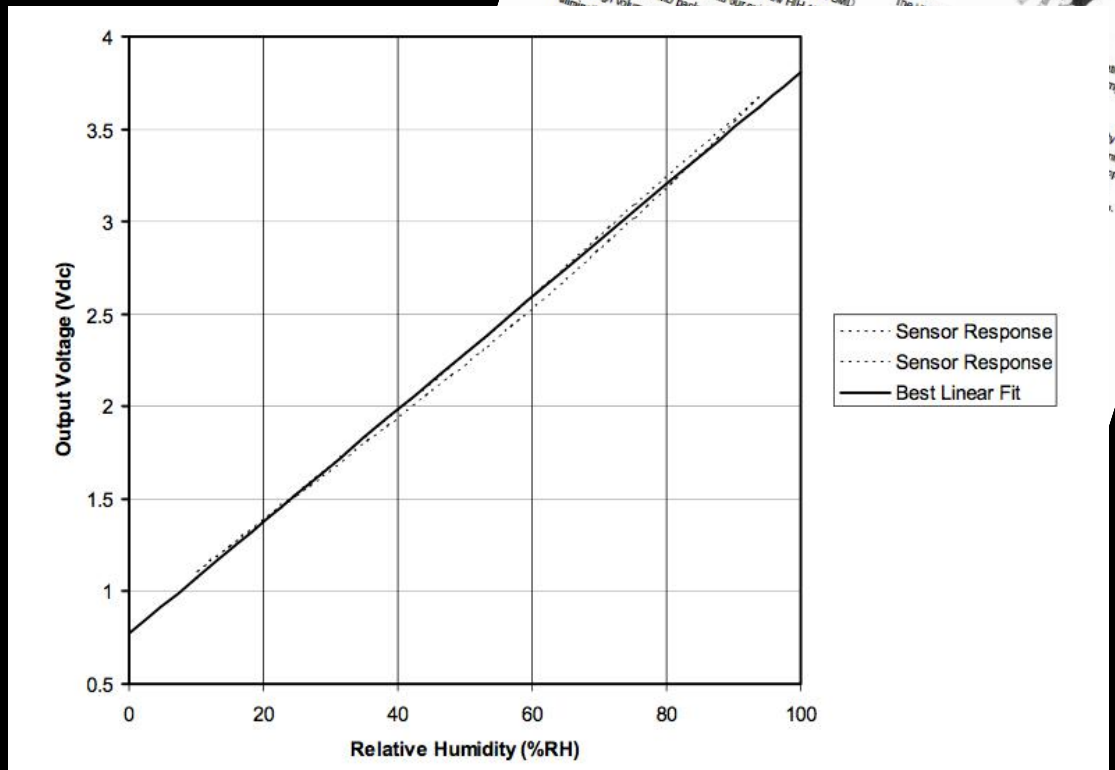
Humidity Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- % RH is a linear function of voltage

- 100% RH looks like ~3.7 V





Humidity Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- **Here's the algebra and the equation to code**

$$V_{OUT} = \left(V_{SUPPLY}\right)\left(0.0062\left(sensorRH\right) + 0.16\right)$$

$$\frac{\left(\frac{V_{OUT}}{\left(V_{SUPPLY}\right)} - 0.16\right)}{0.0062} = sensorRH$$

$$sensorRH = \frac{\left(\frac{V_{OUT}}{\left(5.0\ V\right)} - 0.16\right)}{0.0062}$$

Humidity Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

```
// Definitions
    int sensor;
    float sensorVolt;
    float sensorUnits;
```

```
void loop() {
//   put your main code here, to run repeatedly:

    sensor = analogRead(A2);
    sensorVolt = sensor*(5.0/1023);
    sensorUnits = (((sensorVolt/5.0)-0.16)/0.0062);
    Serial.print(sensor);
    Serial.print("\t voltage ");
    Serial.print(sensorVolt);
    Serial.print("\t units ");
    Serial.println(sensorUnits);
```

```
if(sensorUnits > 10) {
digitalWrite(5, HIGH);
}
if(sensorUnits > 20) {
digitalWrite(6, HIGH);
}
if(sensorUnits > 30) {
digitalWrite(7, HIGH);
}
if(sensorUnits > 40) {
digitalWrite(9, HIGH);
}
delay(100);
```

Humidity Sensor:

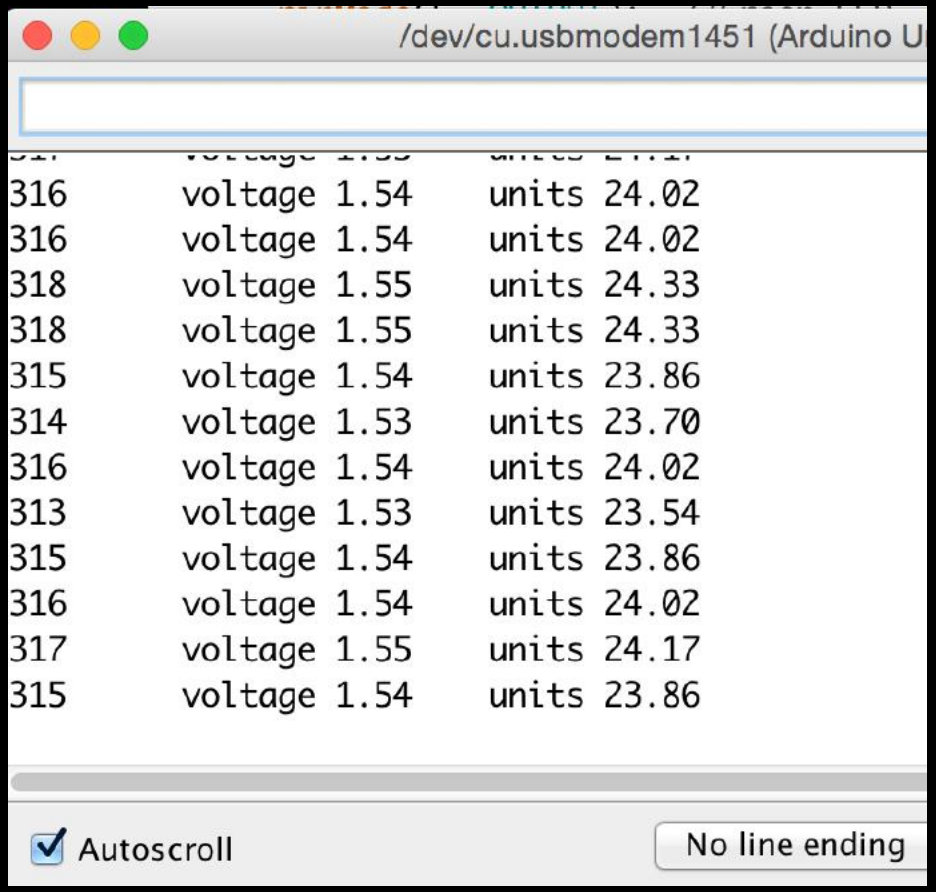
Space Minor
UNIVERSITY OF COLORADO BOULDER

- Verify and upload your code

- Launch serial monitor

- Breathe on humidity sensor like Darth Vader

- Watch LEDs on Shield



Humidity Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Play with your new sensor some to make sure you understand how it works!

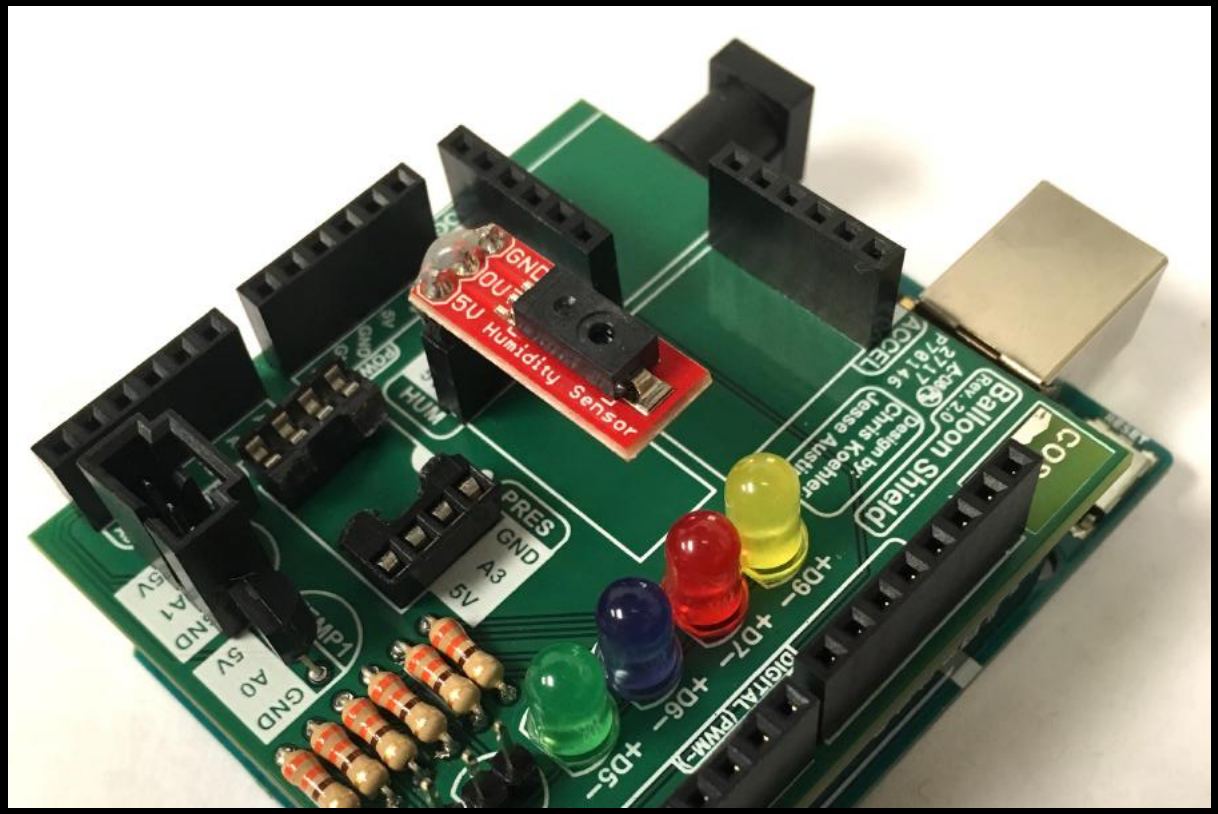
- Also, look at the data sheet and determine the voltage at maximum humidity

PLEASE SAVE YOUR SKETCH FILE

Balloon Shield Build Part 3:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- **Disconnect you Balloon Shield and add the Humidity Sensor**



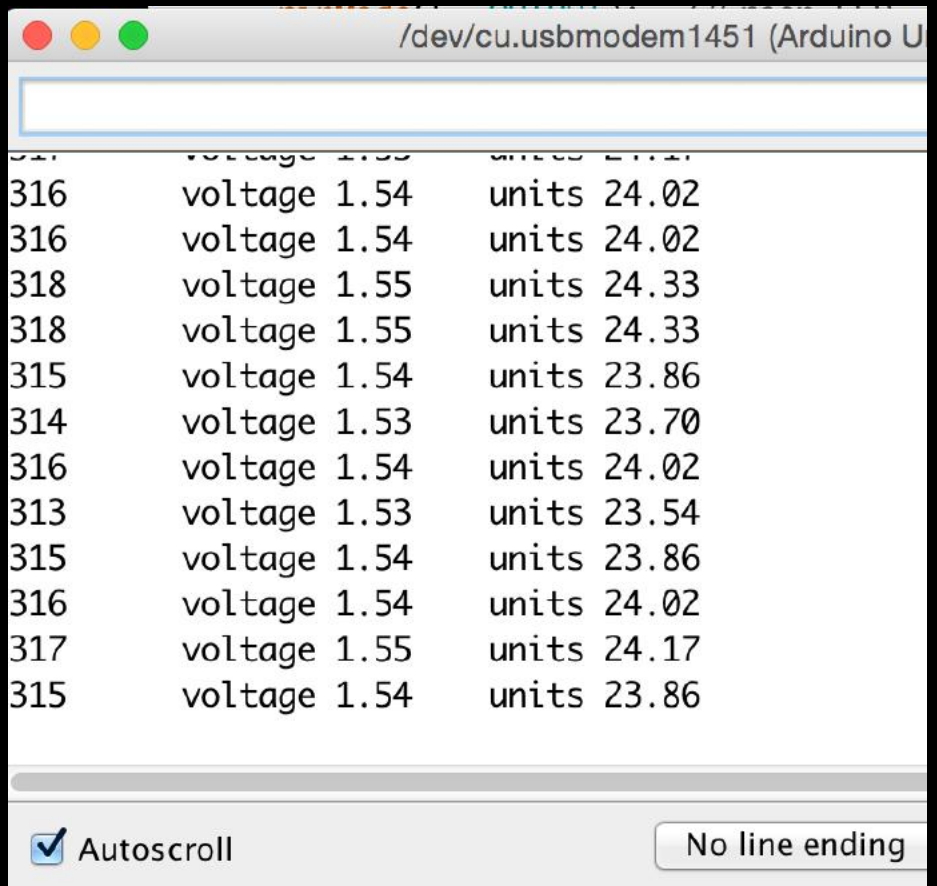
Balloon Shield Build Part 2:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Reconnect your Balloon Shield to the Arduino

- Connect USB and reload code

- Verify same results

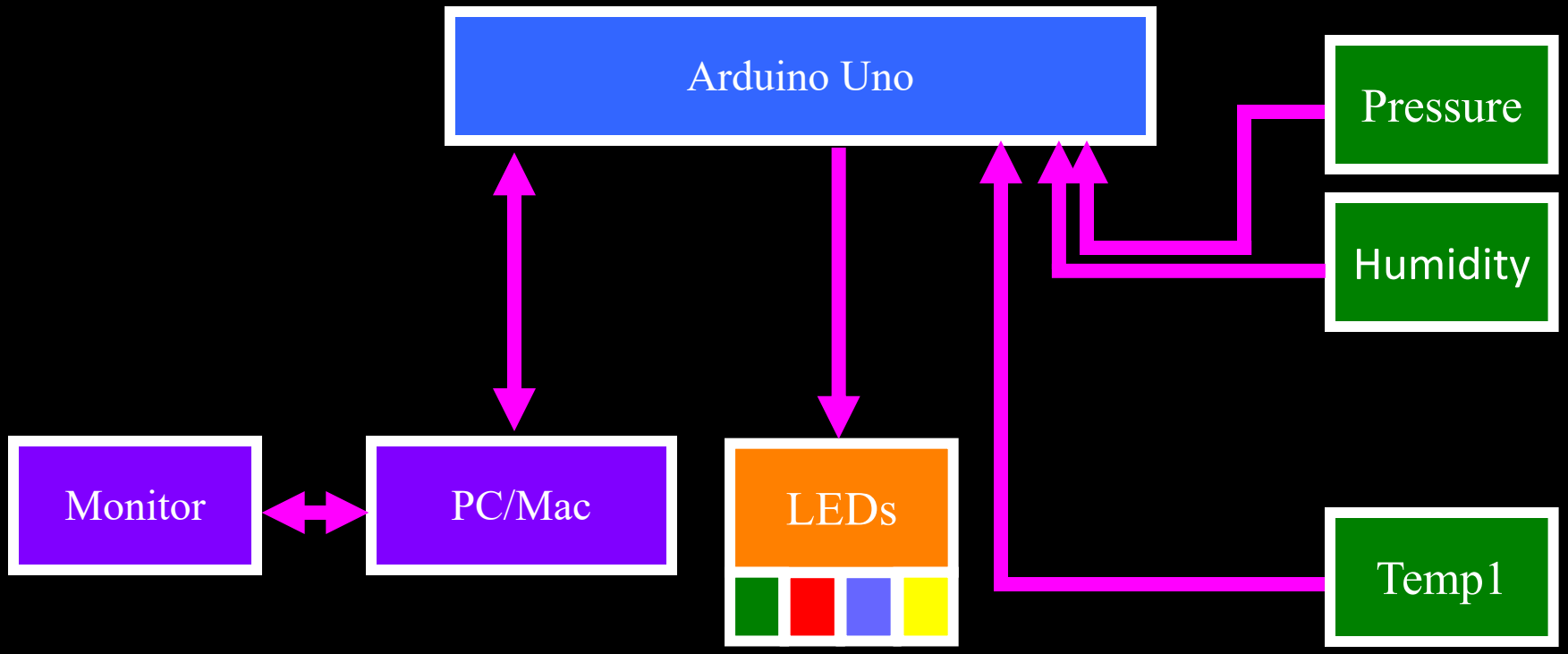


Part 2 – Arduino Road Trip Sensors

A. Humidity Sensor
B. Pressure Sensor
C. Accelerometers
D. External Temp Sensor

Pressure Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER



Pressure Sensor:

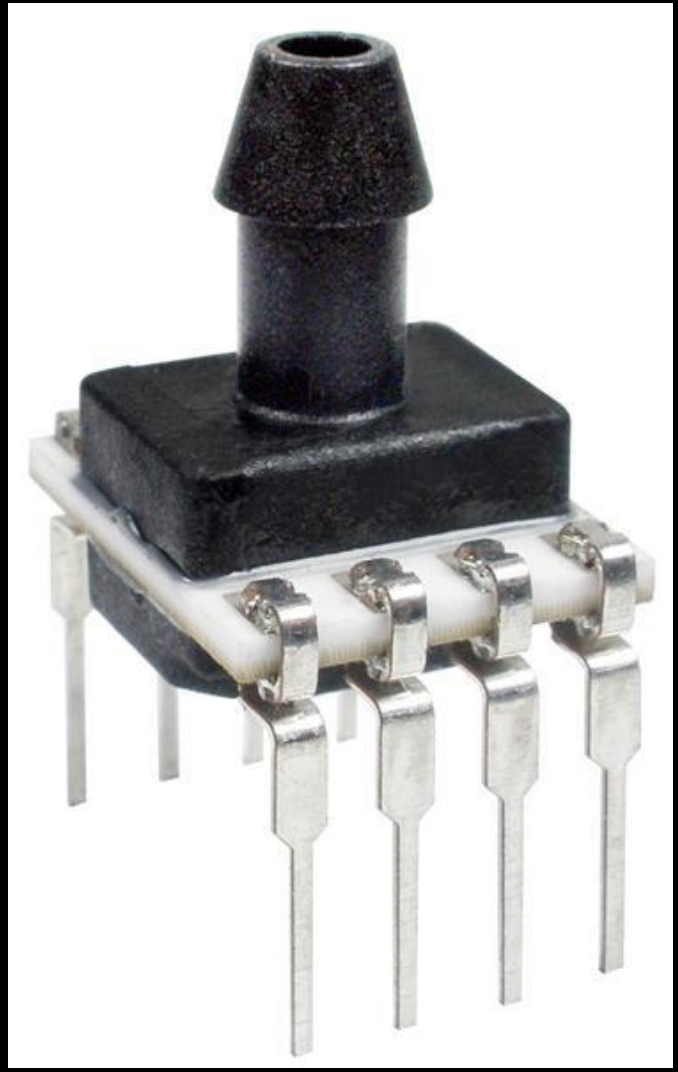
Space Minor
UNIVERSITY OF COLORADO BOULDER

- Pressure Sensors is fragile and $$$

- A bit tricky to see the markings to install correctly

- Can use it to determine pressure/altitude of payload

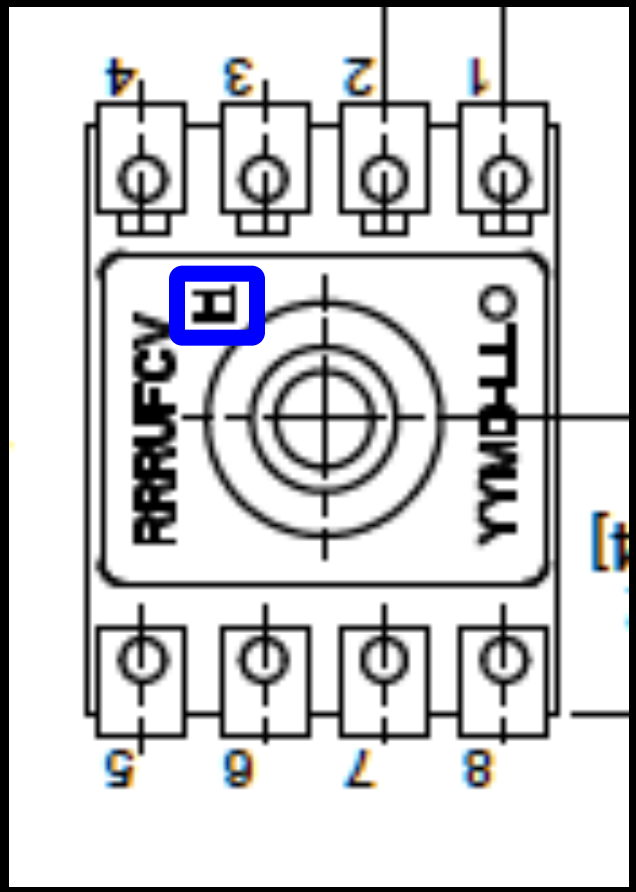
- To be safe, please disconnect power from your Arduino

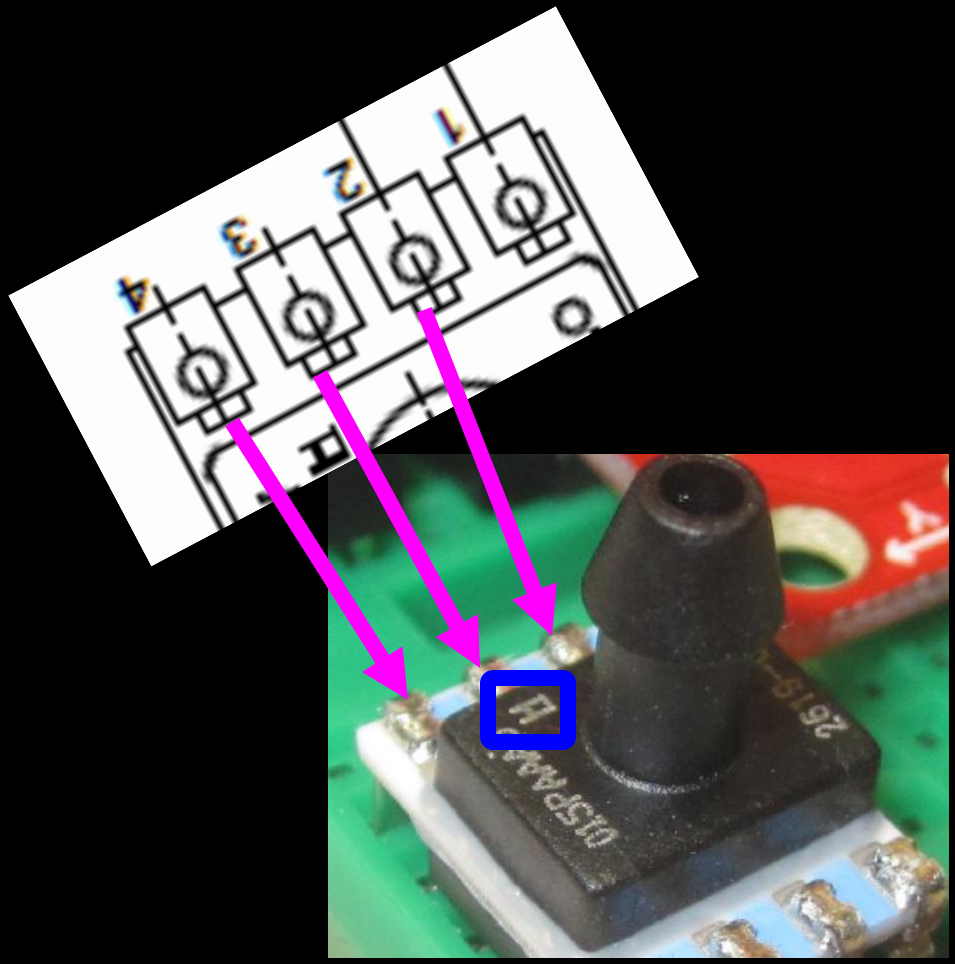


Pressure Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Pressure sensor orientation

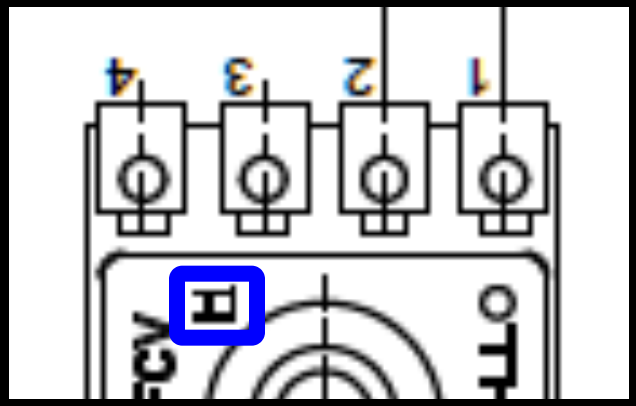


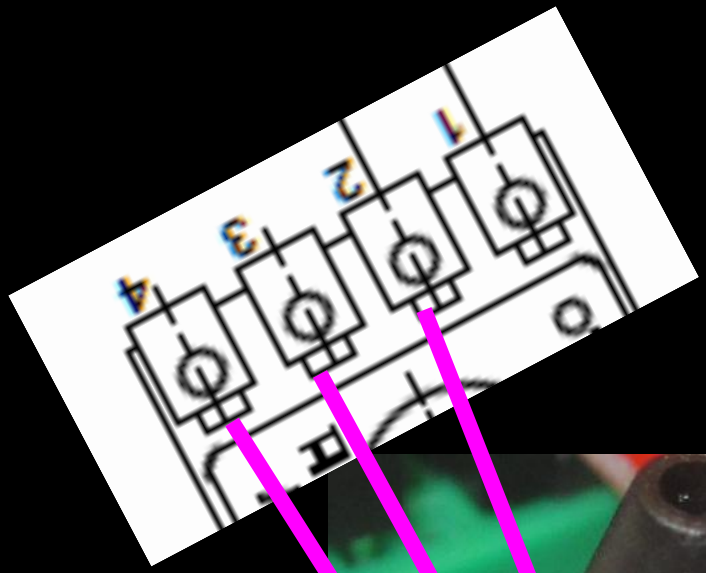


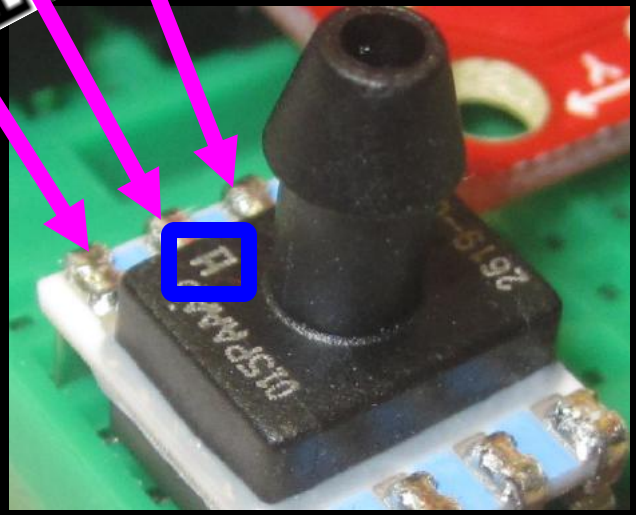
Pressure Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Pressure sensor orientation







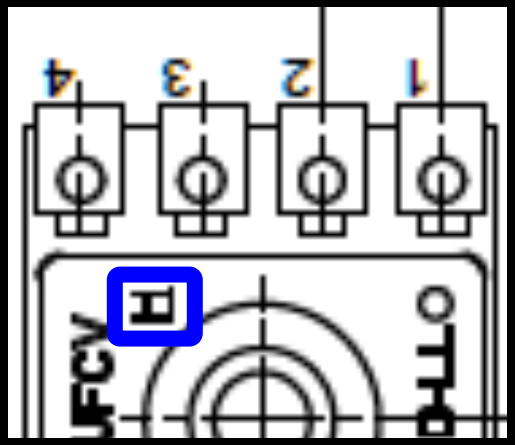
Pin 1	Pin 2	Pin 3	Pin 4
NC	V_{supply}	OUTPUT+	GND

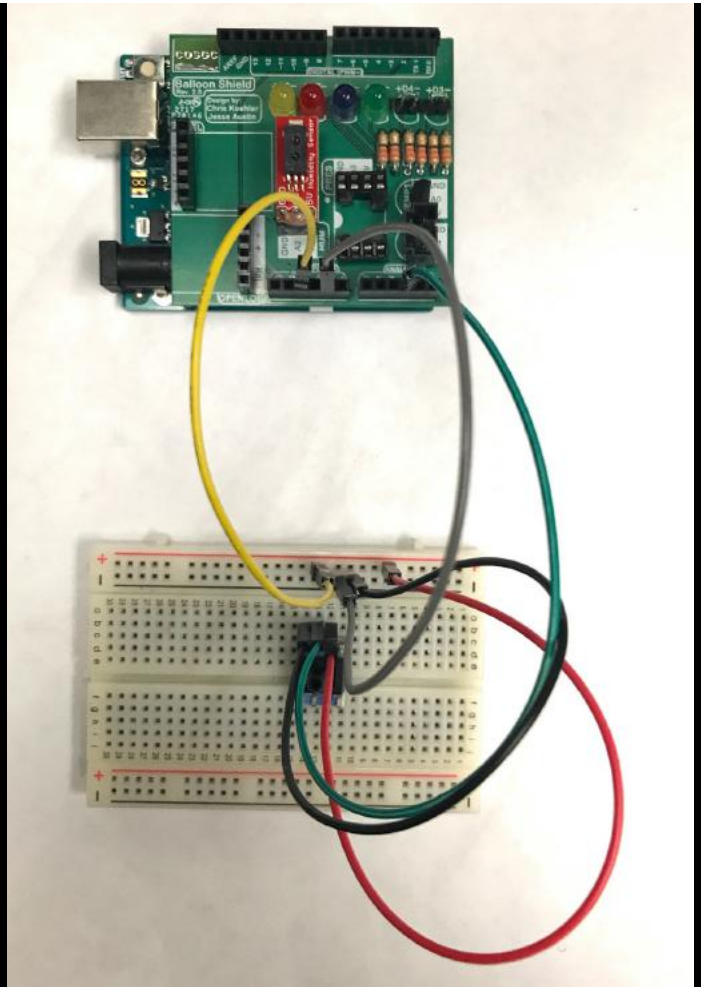
Pressure Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Connect GND to Pin 4, 5V to Pin 2, and Pin 3 to A3 on the Arduino

Pin 1	Pin 2	Pin 3	Pin 4
NC	V_{supply}	OUTPUT+	GND

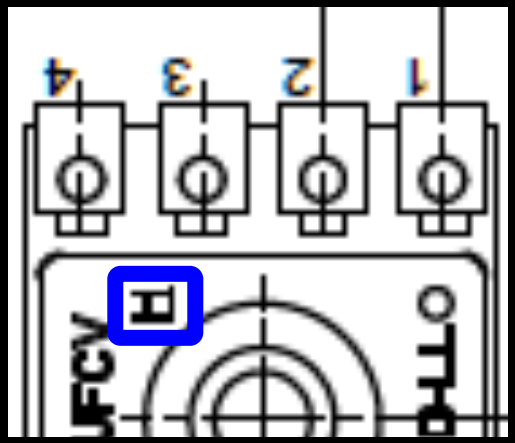




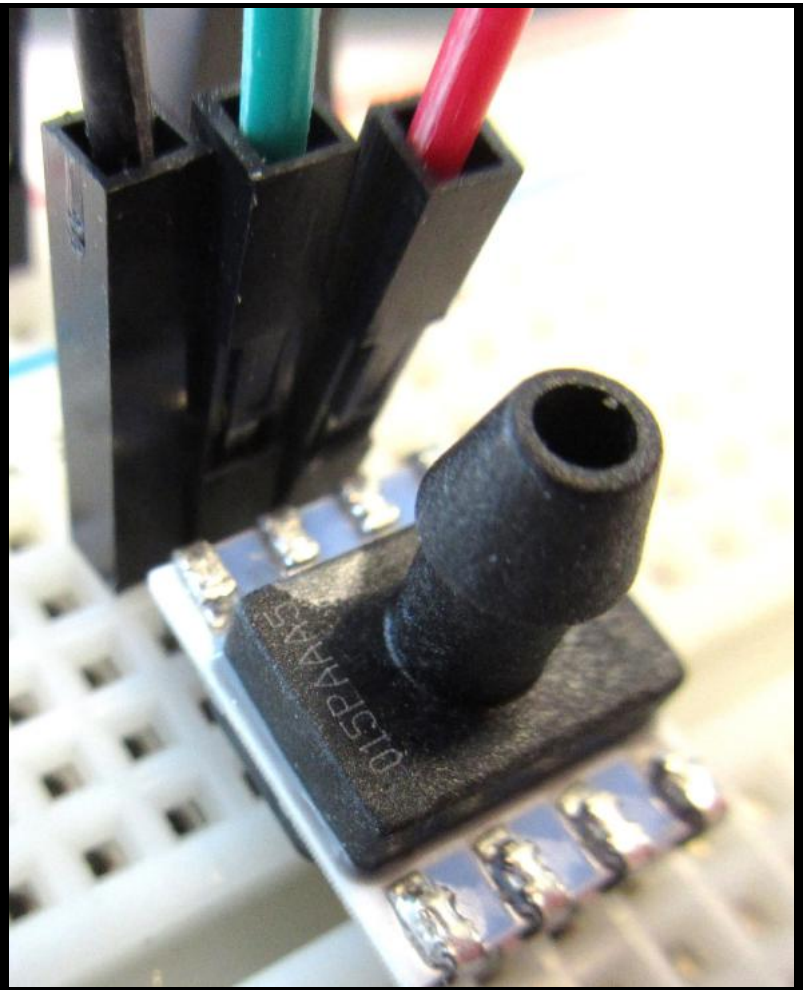
Pressure Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Connect GND to Pin 4, 5V to Pin 2, and Pin 3 to A3 on the Arduino



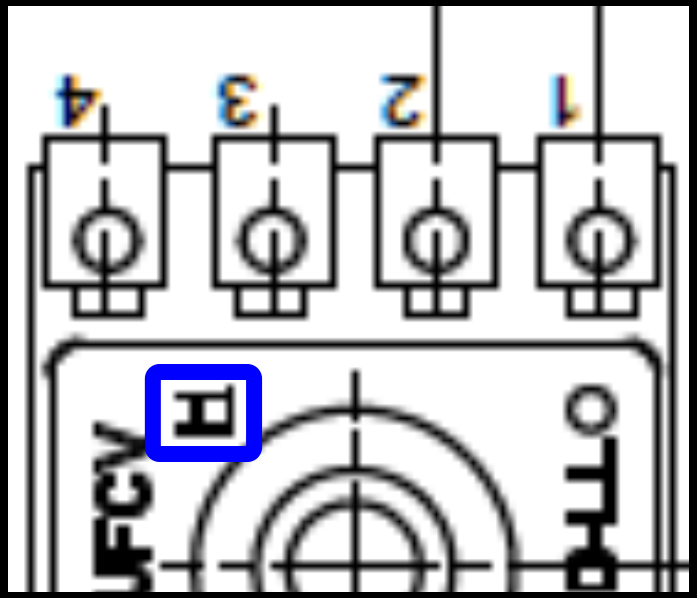
Pin 1	Pin 2	Pin 3	Pin 4
NC	V_{supply}	OUTPUT+	GND



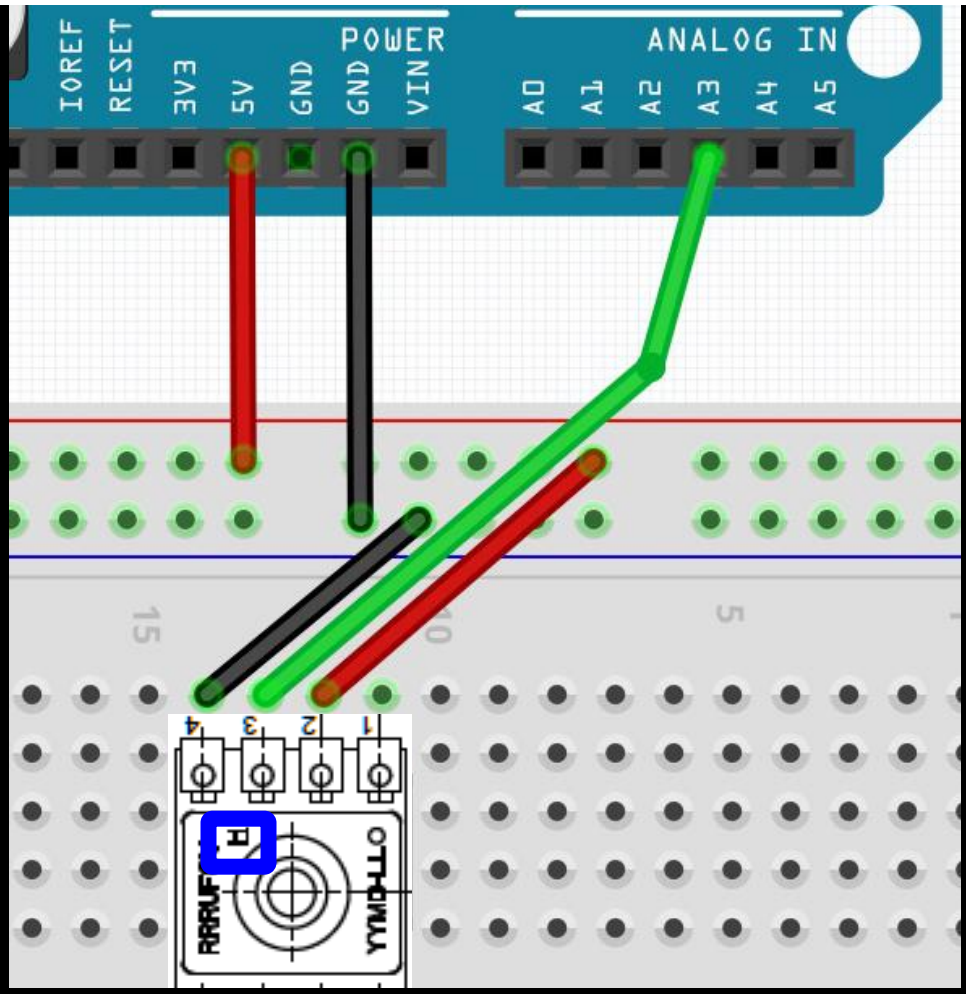
Pressure Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Connect GND to Pin 4, 5V to Pin 2, and Pin 3 to A3 on the Arduino



Pin 1	Pin 2	Pin 3	Pin 4
NC	V_{supply}	OUTPUT+	GND



Pressure Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Look at the data sheet to understand output of the sensor

- Known:
  Vsupply = 5.0 V
  Pmax = 15.0 psi
  Pmin = 0.0 psi
  Output(V) = measured
  Pressure applied = solve

TruStability®
Silicon Pressure Sensors:
SSC Series–Standard Accuracy
±2% Total Error Band,
Amplified Compensated Analog Output,
1 psi to 150 psi (60 mbar to 10 bar)

Honeywell

DESCRIPTION

FEATURES

$$\text{Output (V)} = \frac{0.8 \times V_{supply}}{P_{max.} - P_{min.}} \times (\text{Pressure}_{applied} - P_{min.}) + 0.10 \times V_{supply}$$

Pressure Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- **Here's the algebra and the equation to code**

$$Output(V) = \frac{(0.8 * V_{SUPPLY})}{(P_{\max} - P_{\min})} * (pressure_{applied} - P_{\min}) + 0.10 * V_{\sup ply}$$

$$Output(V) = \frac{(0.8 * 5.0)}{(15.0 - 0.0)} * (pressure_{applied} - 0.0) + 0.10 * 5.0$$

$$Output(V) = \frac{(4.0)}{(15.0)} * (pressure_{applied}) + 0.5$$

$$\frac{15.0}{4.0} * (-0.5 + Output(V)) = pressure_{applied}$$

Pressure Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

```
// Definitions
      int sensor;
      float sensorVolt;
      float sensorUnits;
//      float sensorUnitsC;
```

```
void loop() {
//   put your main code here, to run repeatedly:

      sensor = analogRea(A3);
      sensorVolt = sensor*(5.0/1023);
      sensorUnits = (sensorVolt-0.5)*(15.0/4.0);
      Serial.print(sensor);
      Serial.print("\t voltage ");
      Serial.print(sensorVolt);
      Serial.print("\t units ");
      Serial.println(sensorUnits);
```

```
if(sensorUnits < 12.20) {
digitalWrite(5, HIGH);
}
if(sensorUnits < 10.10) {
digitalWrite(6, HIGH);
}
if(sensorUnits < 8.10) {
digitalWrite(7, HIGH);
}
if(sensorUnits < 3.10) {
digitalWrite(9, HIGH);
}
delay(100);
```

> to <

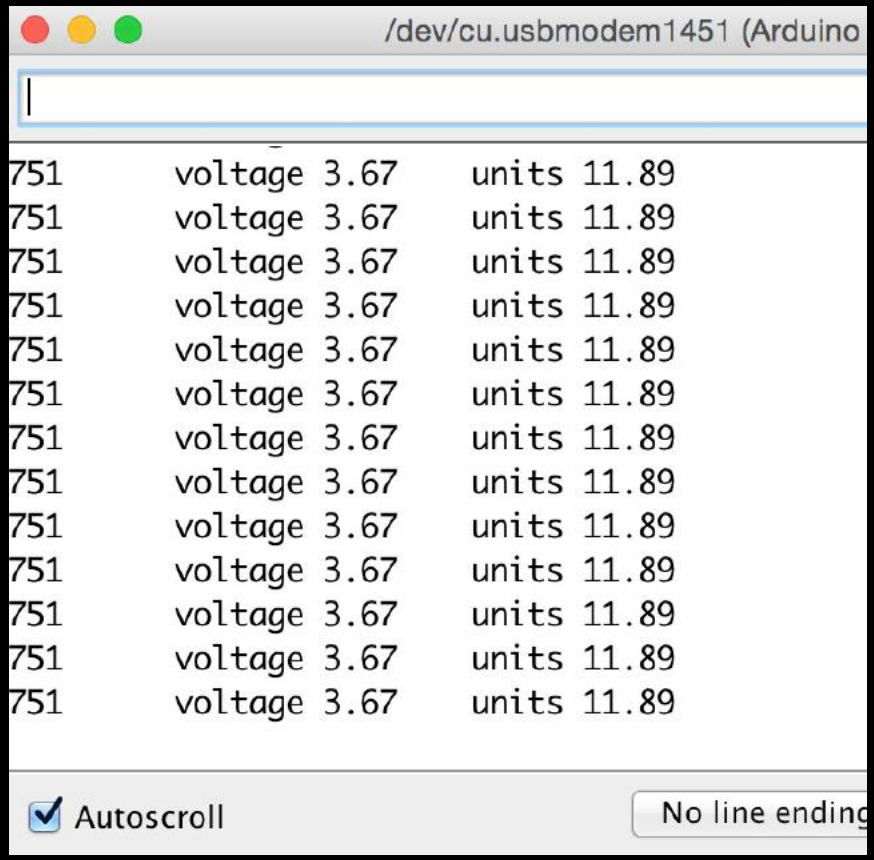
Pressure Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Build and Upload

- **DO NOT BLOW** or **DO NOT APPLY PRESSURE**; it will break the sensor

- Use solder sucker



PLEASE SAVE YOUR SKETCH FILE

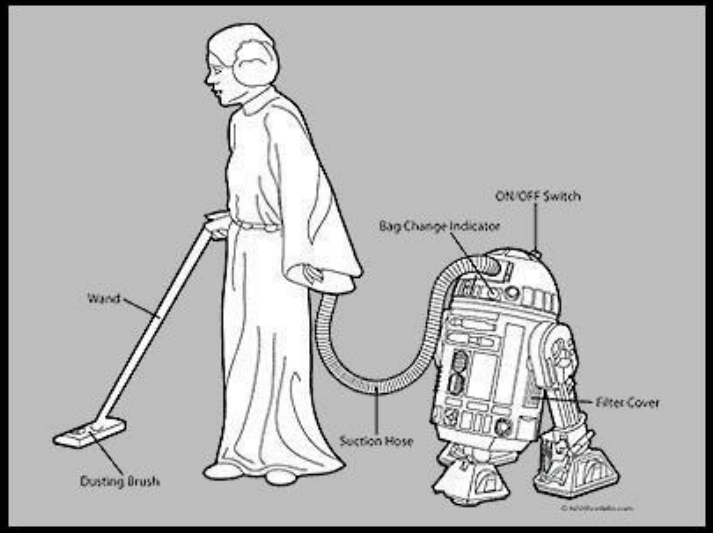
Pressure Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Play with your new sensor to get a feel for how it works

- Try to get your sensor to zero

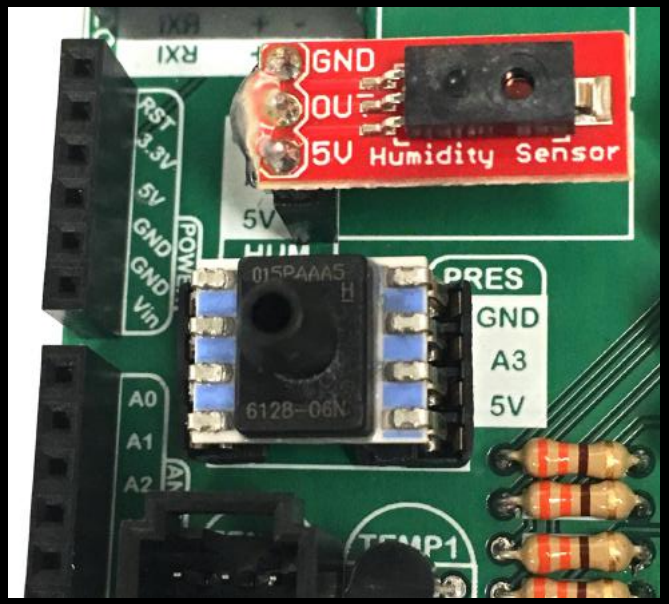
PLEASE SAVE YOUR SKETCH FILE

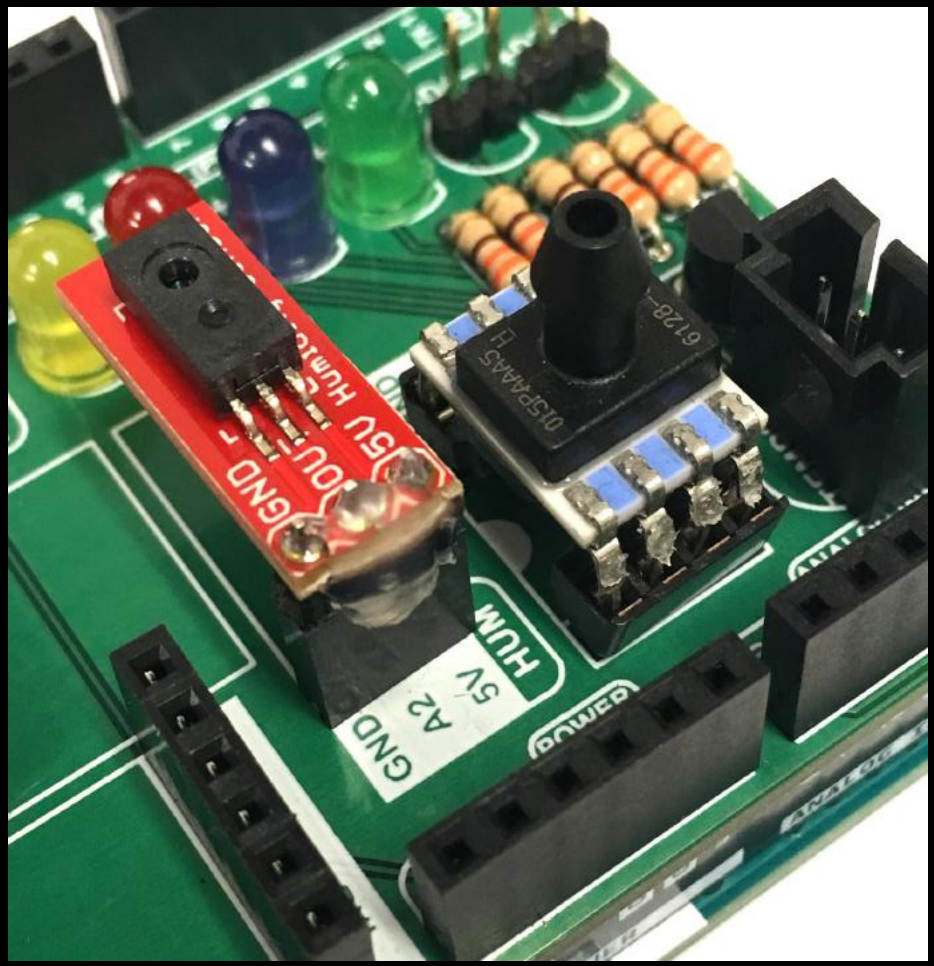


Pressure Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Install Pressure Sensor into headers





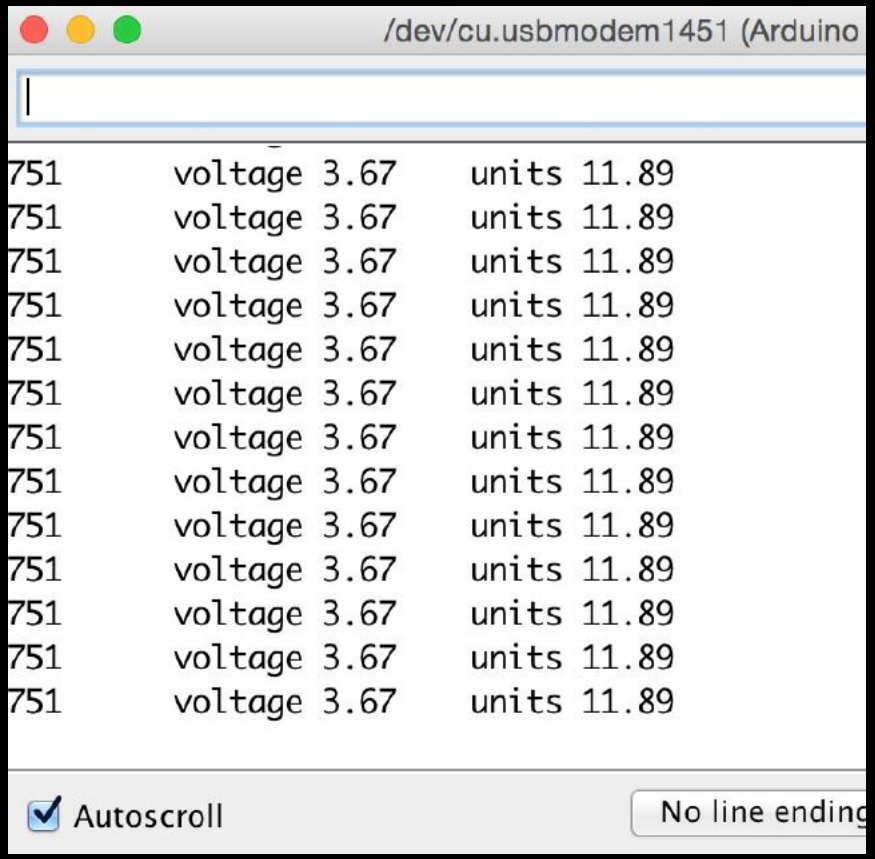
Pressure Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Reconnect your Balloon Shield to the Arduino

- Connect USB and reload code

- Verify same results

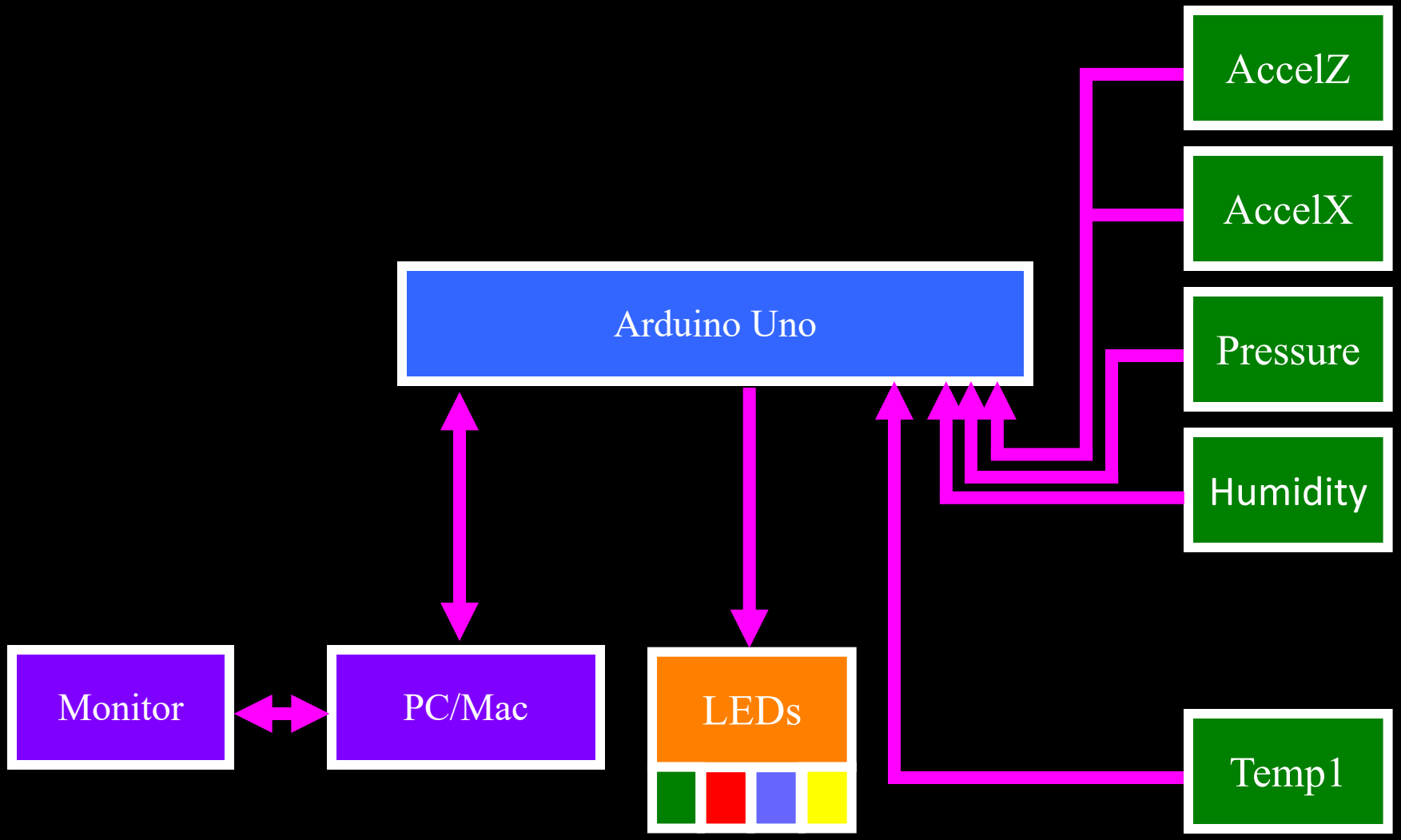


Part 2 – Arduino Road Trip Sensors

A. Humidity Sensor
B. Pressure Sensor
C. Accelerometers
D. External Temp Sensor

Accelerometer:

Space Minor
UNIVERSITY OF COLORADO BOULDER



Accelerometer:

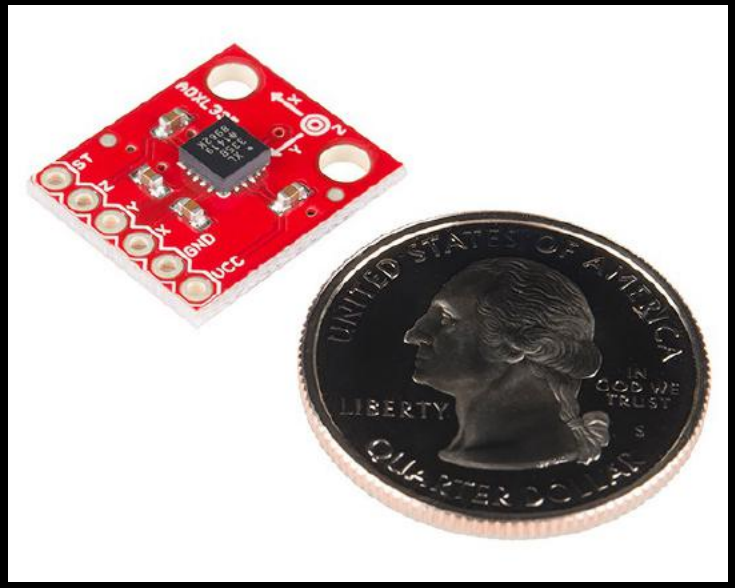
Space Minor
UNIVERSITY OF COLORADO BOULDER

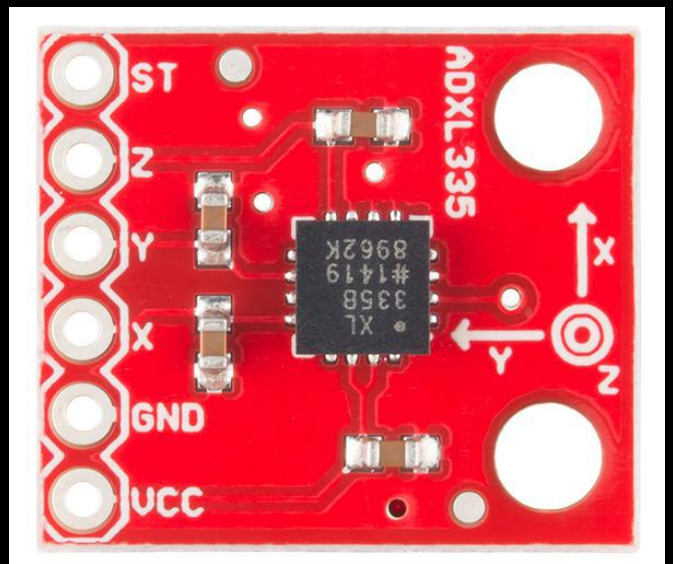
- Accelerometers are used to detect forces acting on a payload

- This is a 3 axis accelerometer

- Measures g forces in X, Y, and Z directions

- **Only have two analog** channels left so X and Z

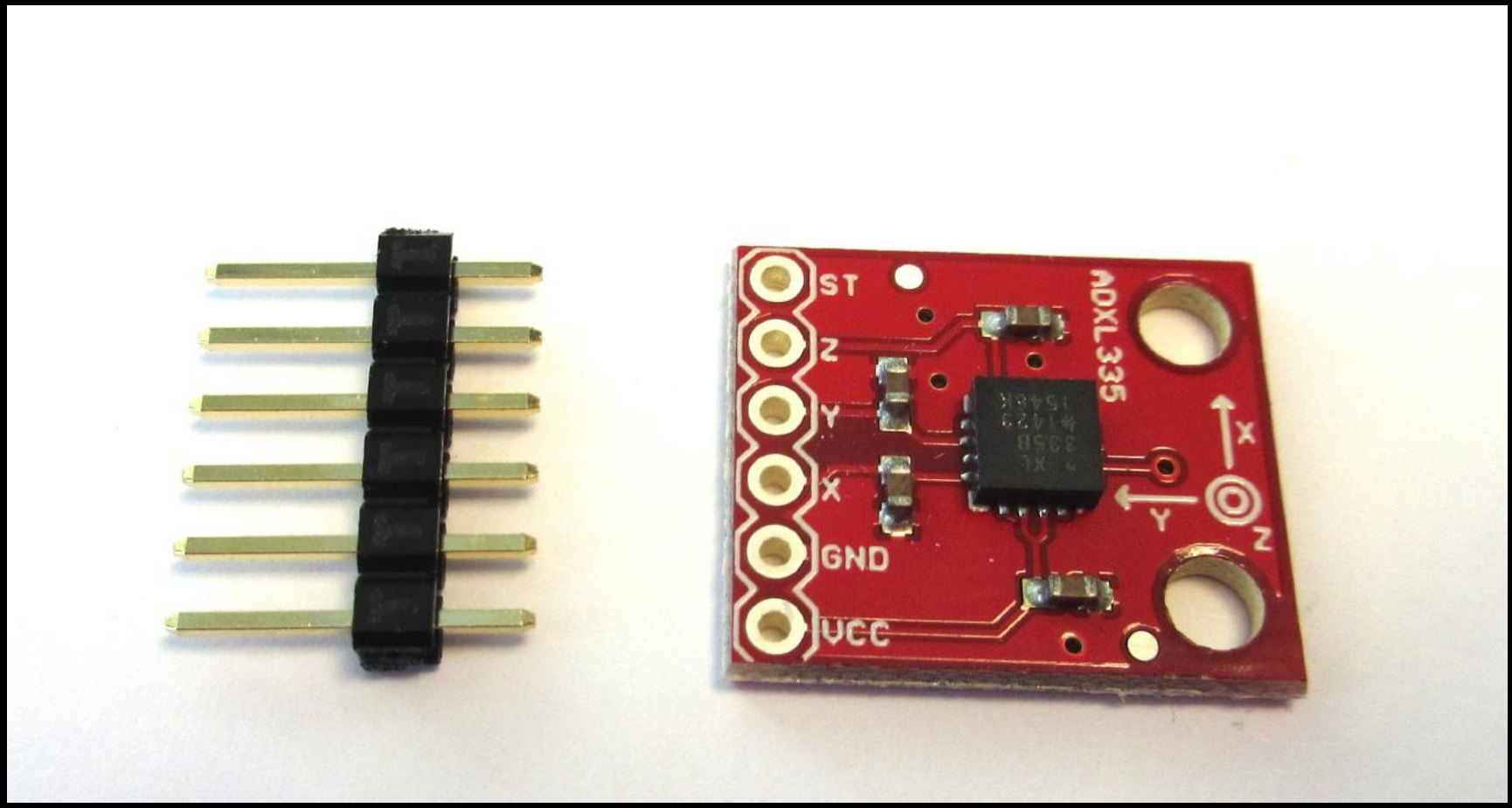




Accelerometer:

Space Minor
UNIVERSITY OF COLORADO BOULDER

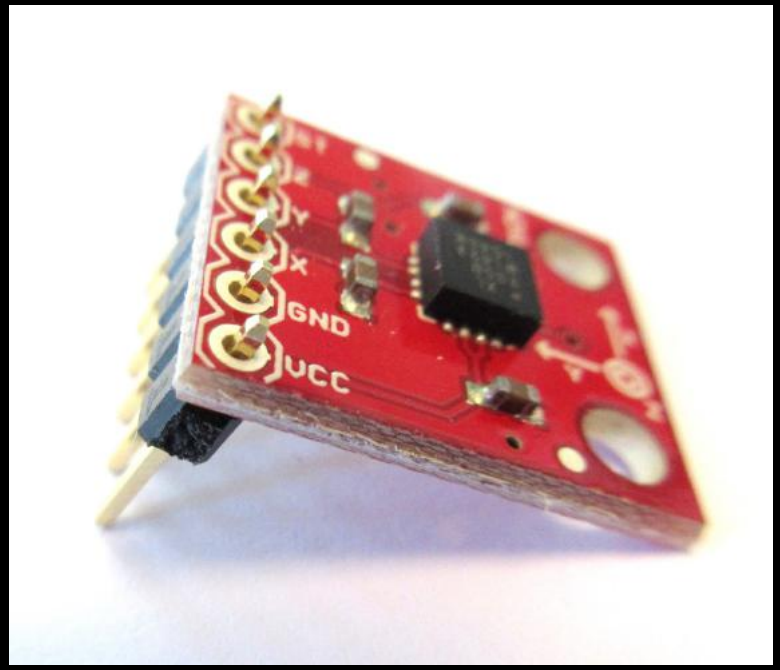
- Solder 6 pin header to board

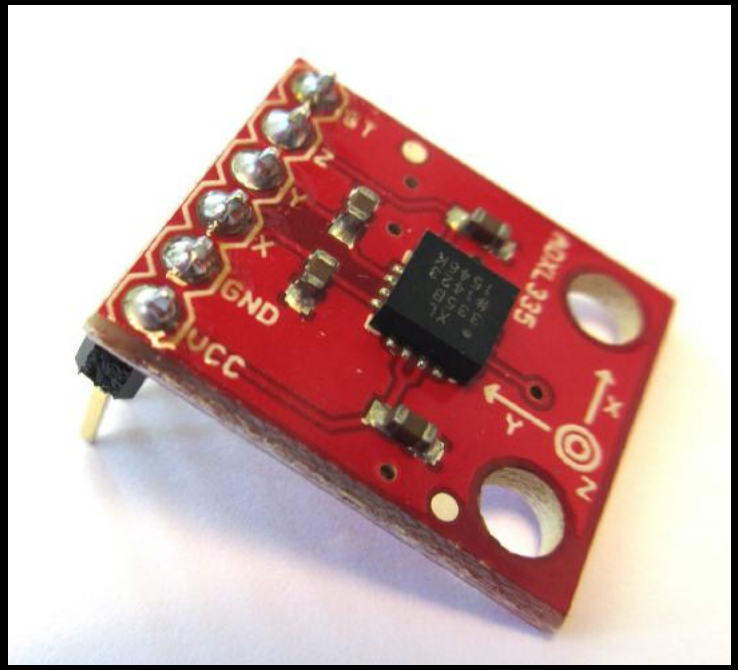


Accelerometer:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Solder 6 pin header to board
- Short side through the bottom of the board
- Keep header perpendicular to board



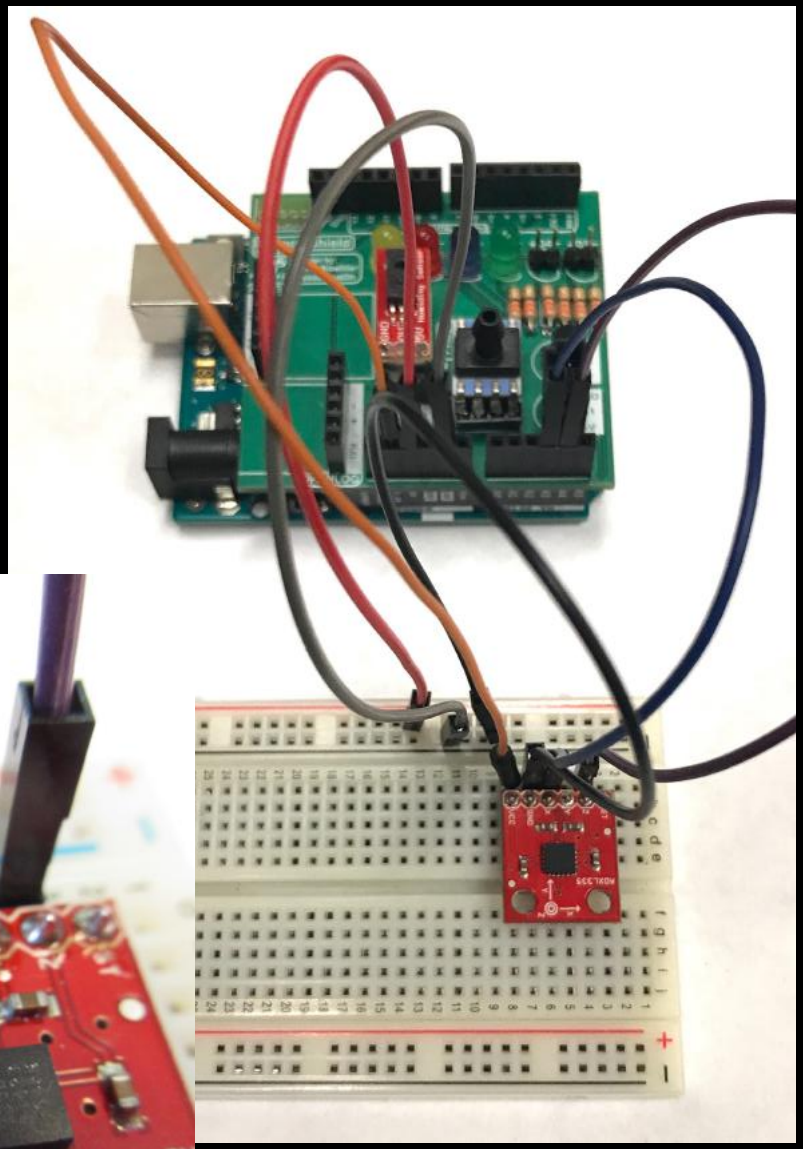


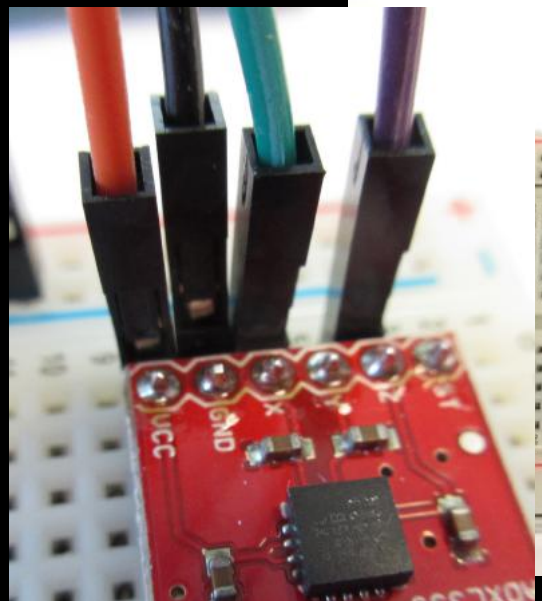
Accelerometer:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Wire accelerometer as shown

  Vcc is to 3.3V
  GND is to GND
  X is to A4
  Z is to A5



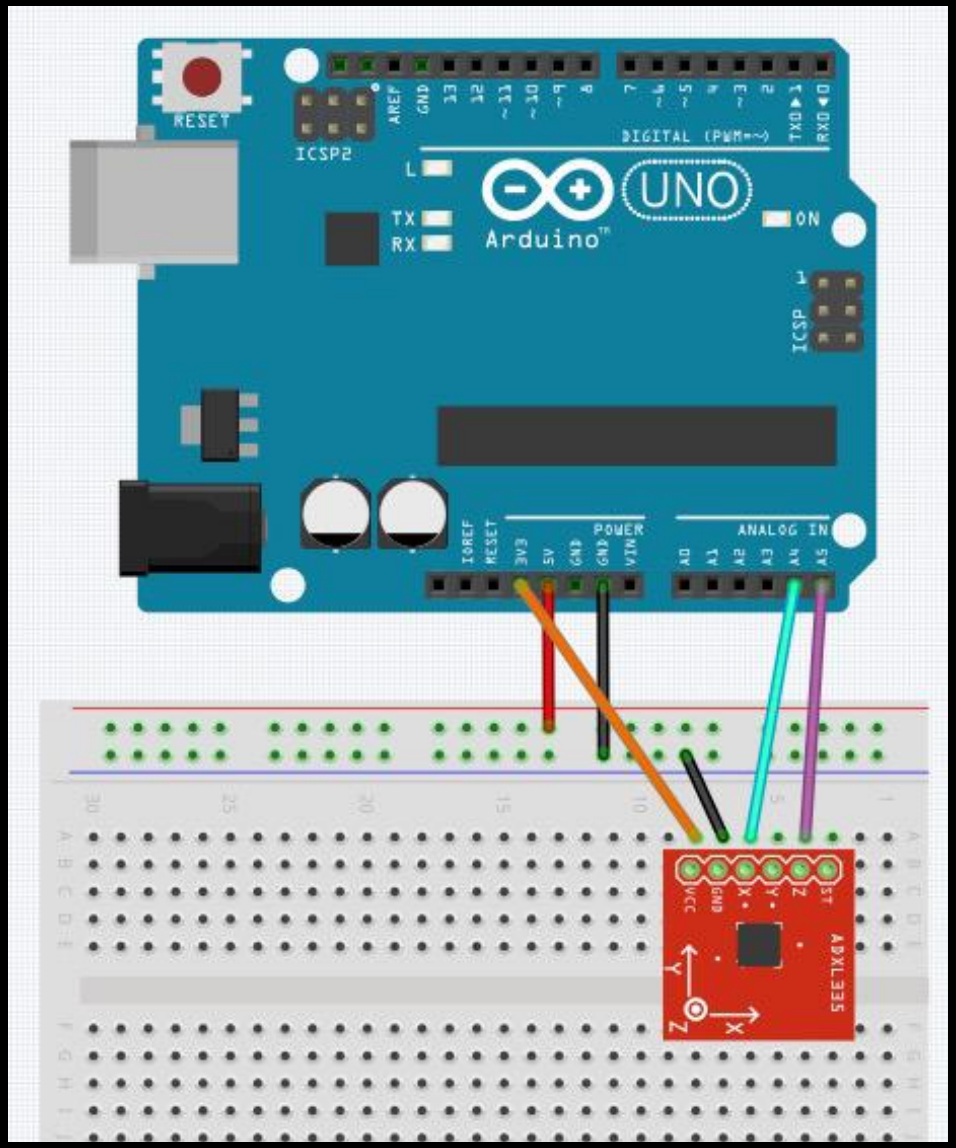


Accelerometer:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Wire accelerometer as shown

Vcc is to 3.3V
GND is to GND
X is to A4
Z is to A5

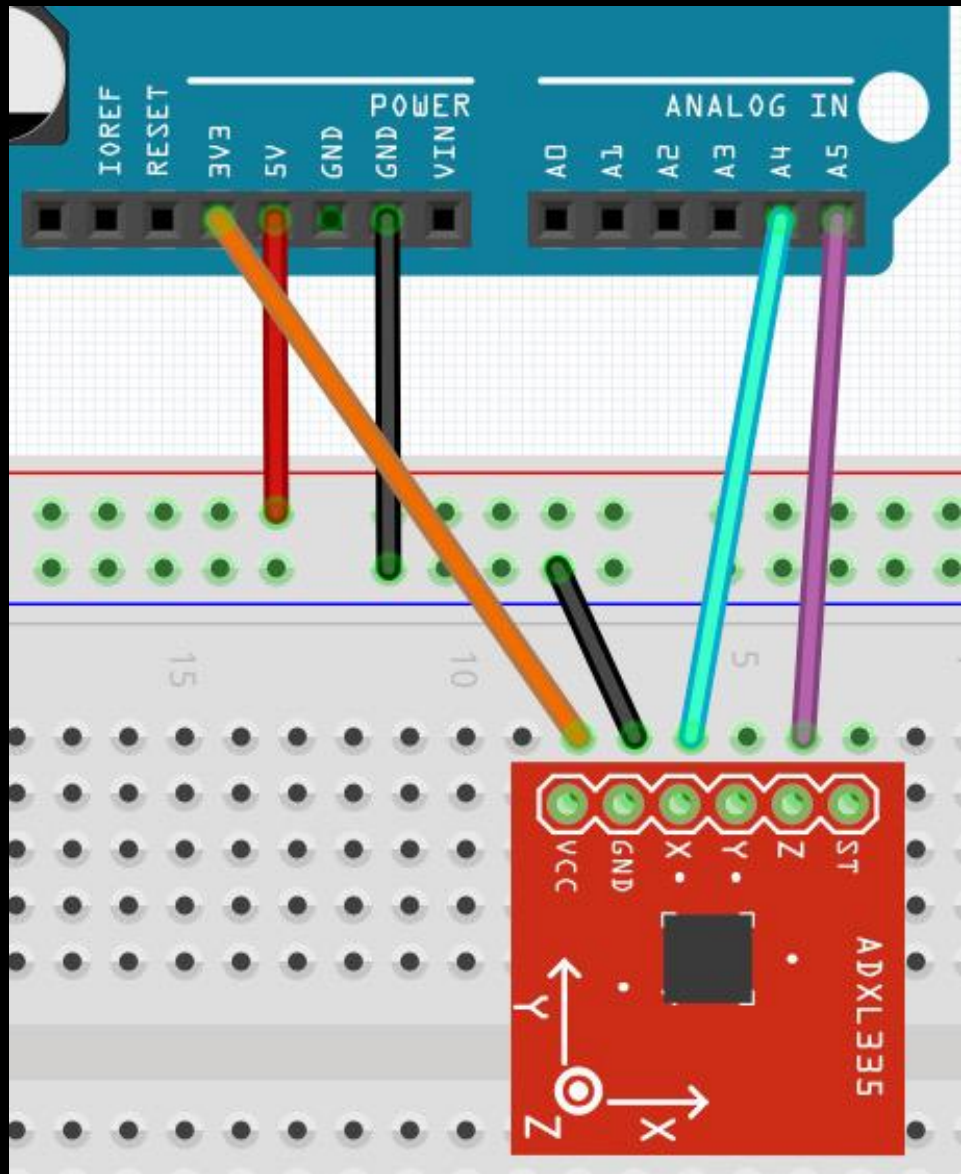


Accelerometer:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Wire accelerometer as shown

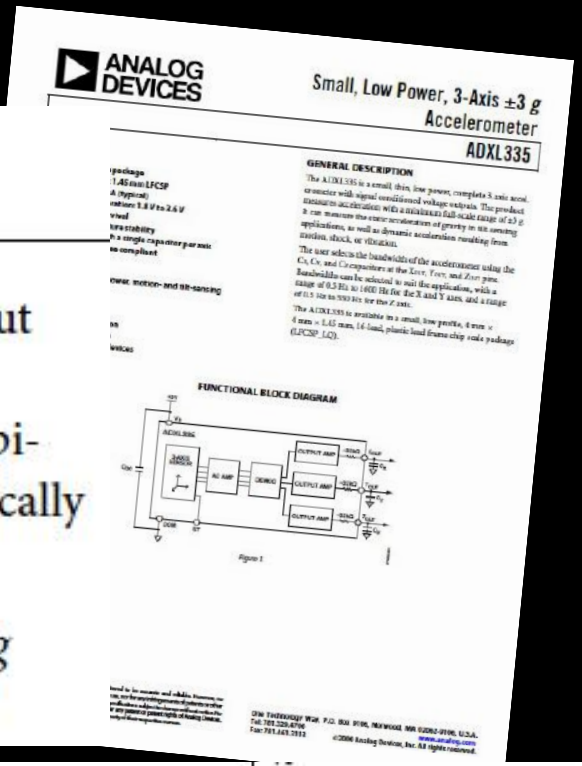
Vcc is to 3.3V
GND is to GND
X is to A4
Z is to A5



Accelerometer:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Looking at the data sheet…


ANALOG DEVICES

Small, Low Power, 3-Axis ±3 *g* Accelerometer

ADXL335

GENERAL DESCRIPTION

FUNCTIONAL BLOCK DIAGRAM

ADXL335

The ADXL335 output is ratiometric, therefore, the output sensitivity (or scale factor) varies proportionally to the supply voltage. At V_S = 3.6 V, the output sensitivity is typically 360 mV/*g*. At V_S = 2 V, the output sensitivity is typically 195 mV/*g*.

The zero *g* bias output is also ratiometric, thus the zero *g* output is nominally equal to V_S/2 at all supply voltages.

SENSITIVITY (RATIOMETRIC)[2]	Each axis				
Sensitivity at X_{OUT}, Y_{OUT}, Z_{OUT}	V_S = 3 V	270	300	330	mV/*g*
Sensitivity Change Due to Temperature[3]	V_S = 3 V		±0.01		%/°C
ZERO *g* BIAS LEVEL (RATIOMETRIC)					
0 *g* Voltage at X_{OUT}, Y_{OUT}	V_S = 3 V	1.35	1.5	1.65	V
0 *g* Voltage at Z_{OUT}	V_S = 3 V	1.2	1.5	1.8	V
0 *g* Offset vs. Temperature			±1		mg/°C
NOISE PERFORMANCE					

Accelerometer:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- 3.3V/2 is what it should read at “zero G” orientation or 1.65V

- Then 330 mV for every G so…

Gs = (Accelvoltage – 1.65 V) / (0.330 V)

ADXL335

The ADXL335 output is ratiometric, therefore, the output sensitivity (or scale factor) varies proportionally to the supply voltage. At V_S = 3.6 V, the output sensitivity is typically 360 mV/*g*. At V_S = 2 V, the output sensitivity is typically 195 mV/*g*.

The zero *g* bias output is also ratiometric, thus the zero *g* output is nominally equal to $V_S/2$ at all supply voltages.

Accelerometer:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- **A4, A5,**
  comment out LED ifs

```
// Definitions
int sensorX;
int sensorZ;
float sensorVoltX;
float sensorVoltZ;
float sensorUnitsX;
float sensorUnitsZ;
```

```
void loop() {
//   put your main code here, to run

     sensorX = analogRead(A4);
     sensorZ = analogRead(A5);
     sensorVoltX = sensorX*(5.0/1023);
     sensorVoltZ = sensorZ*(5.0/1023);
     sensorUnitsX = (sensorVoltX-(3.3/2))/(0.330);
     sensorUnitsZ = (sensorVoltZ-(3.3/2))/(0.330);
     Serial.print("X ");
     Serial.print(sensorUnitsX);
     Serial.print("\t Z ");
     Serial.println(sensorUnitsZ);
```

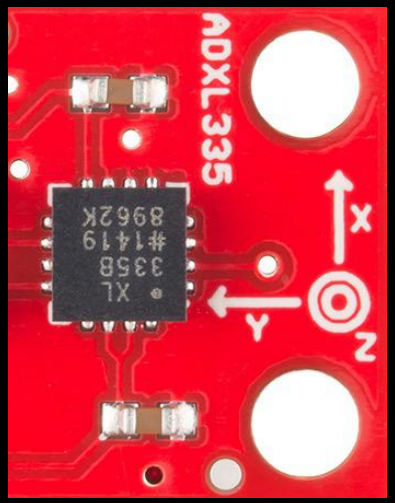
Accelerometer:

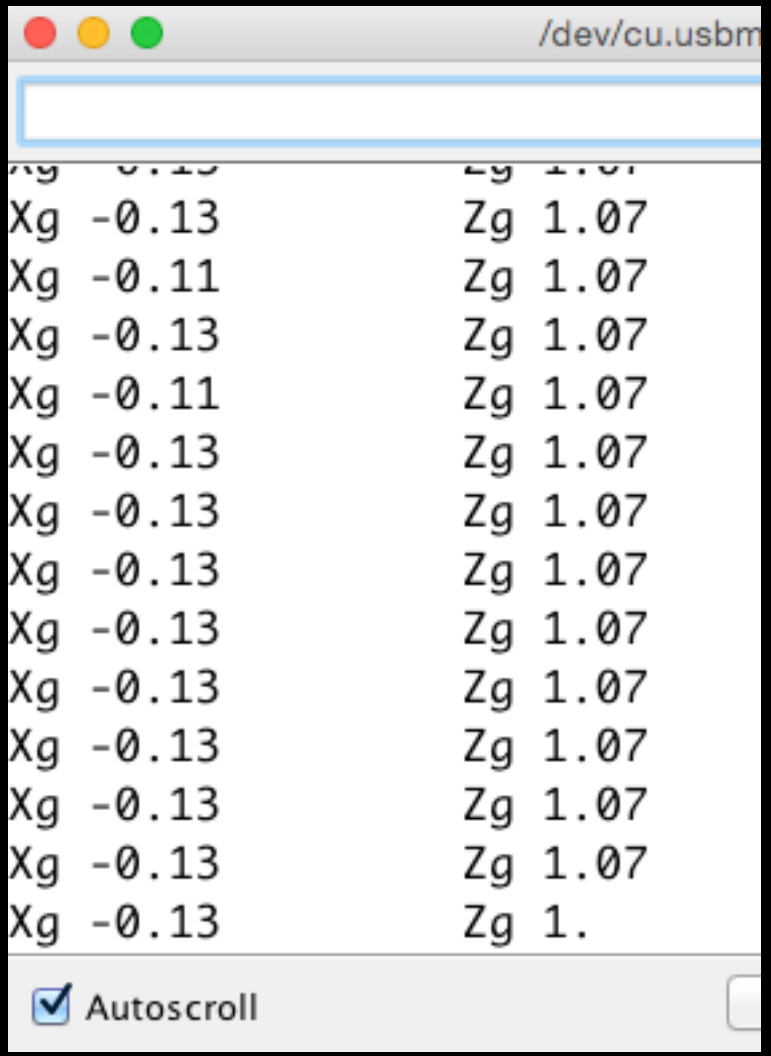
Space Minor
UNIVERSITY OF COLORADO BOULDER

- Upload you code and launch your serial monitor (no LEDs this time)

- Rotate your breadboard and look for changes in both X and Z

- X up and X down
- Z up and Z down



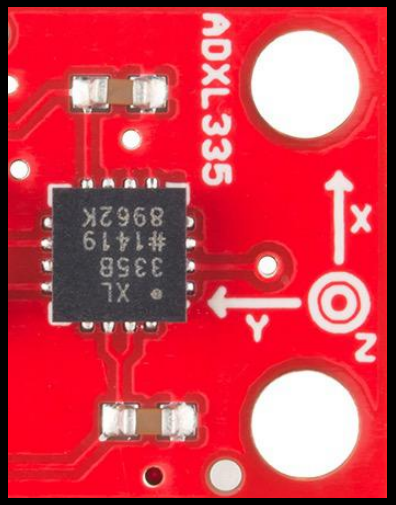


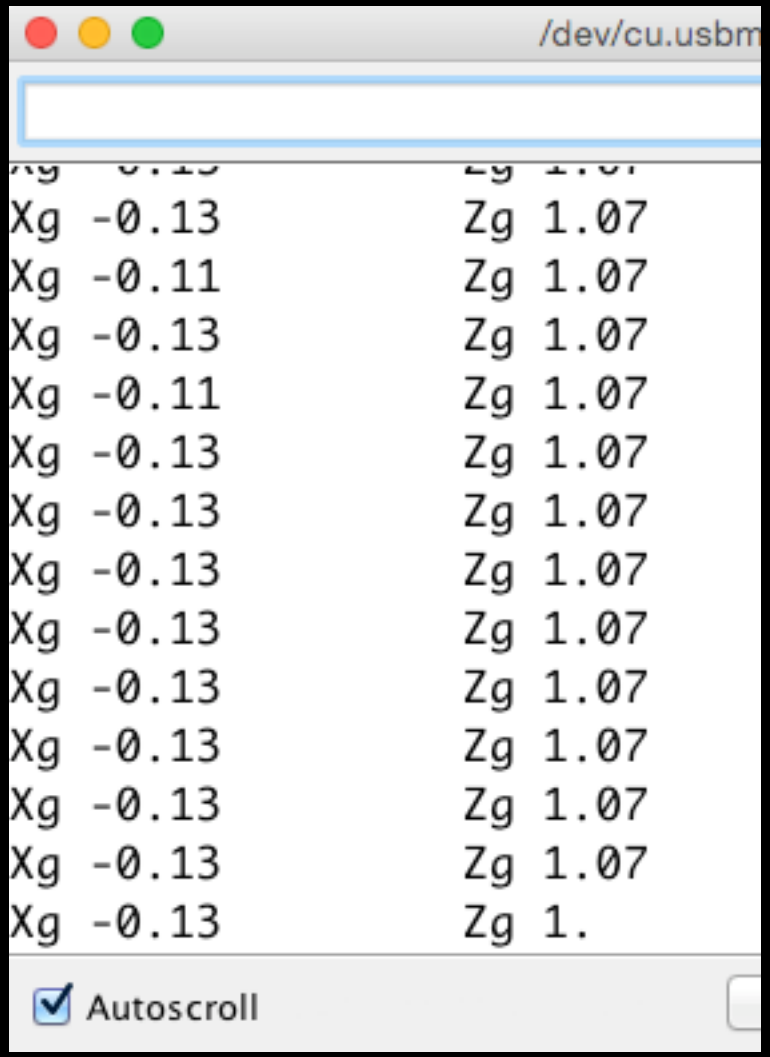
Accelerometer:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Upload you code and launch your serial monitor

- When Z up ~ 1.0G
- When Z down ~ -1.0G
- When X up ~ 1.0G
- When X down ~ -1.0G

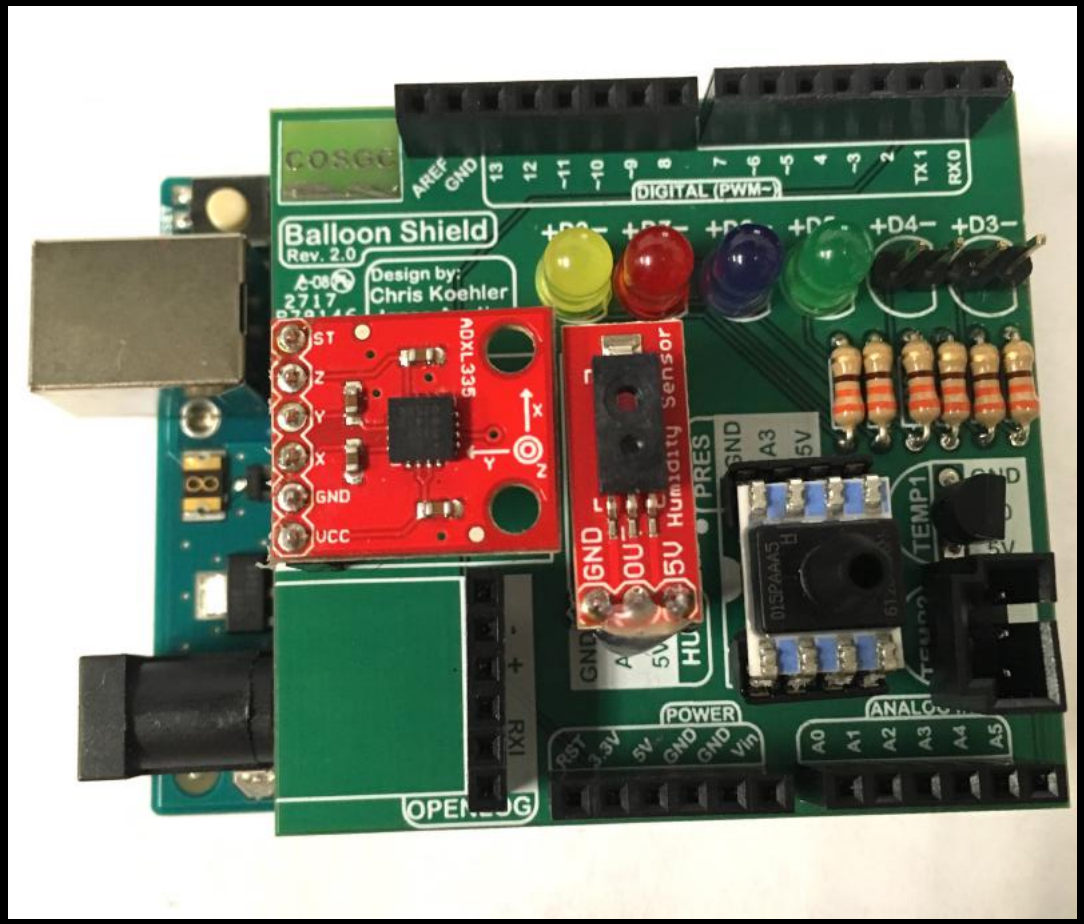




Accelerometer:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- **Disconnect your Balloon Shield and add the Accelerometer**



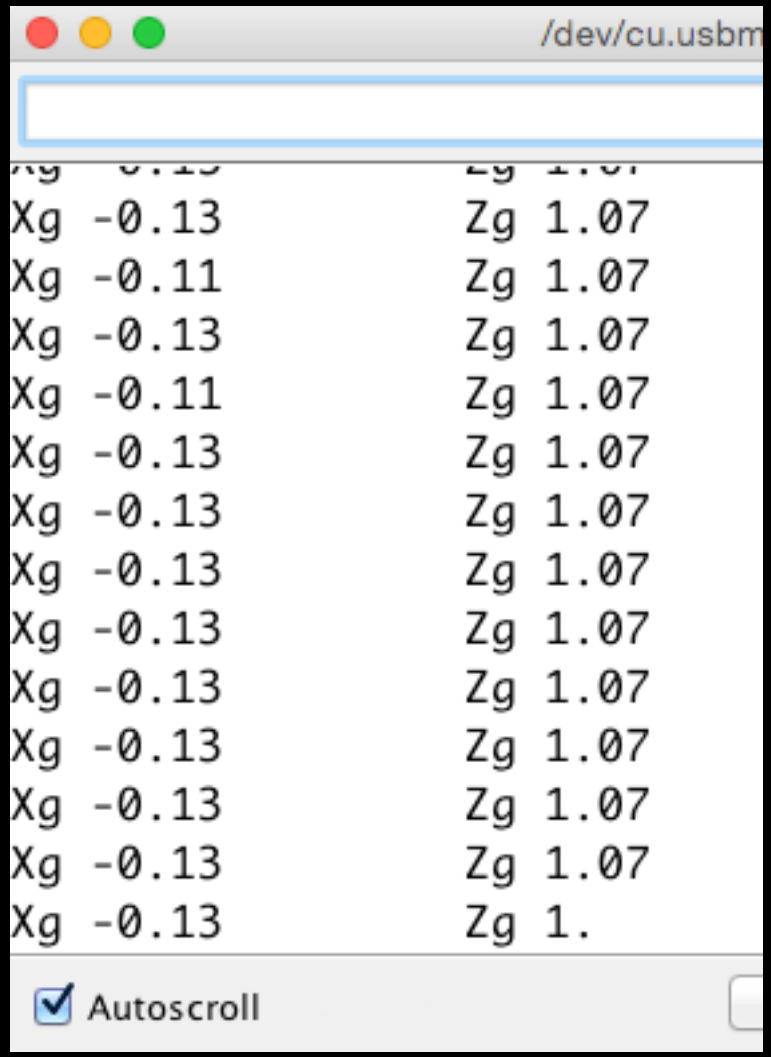
Accelerometer:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Reconnect your Balloon Shield to the Arduino

- Connect USB and reload code

- Verify same results

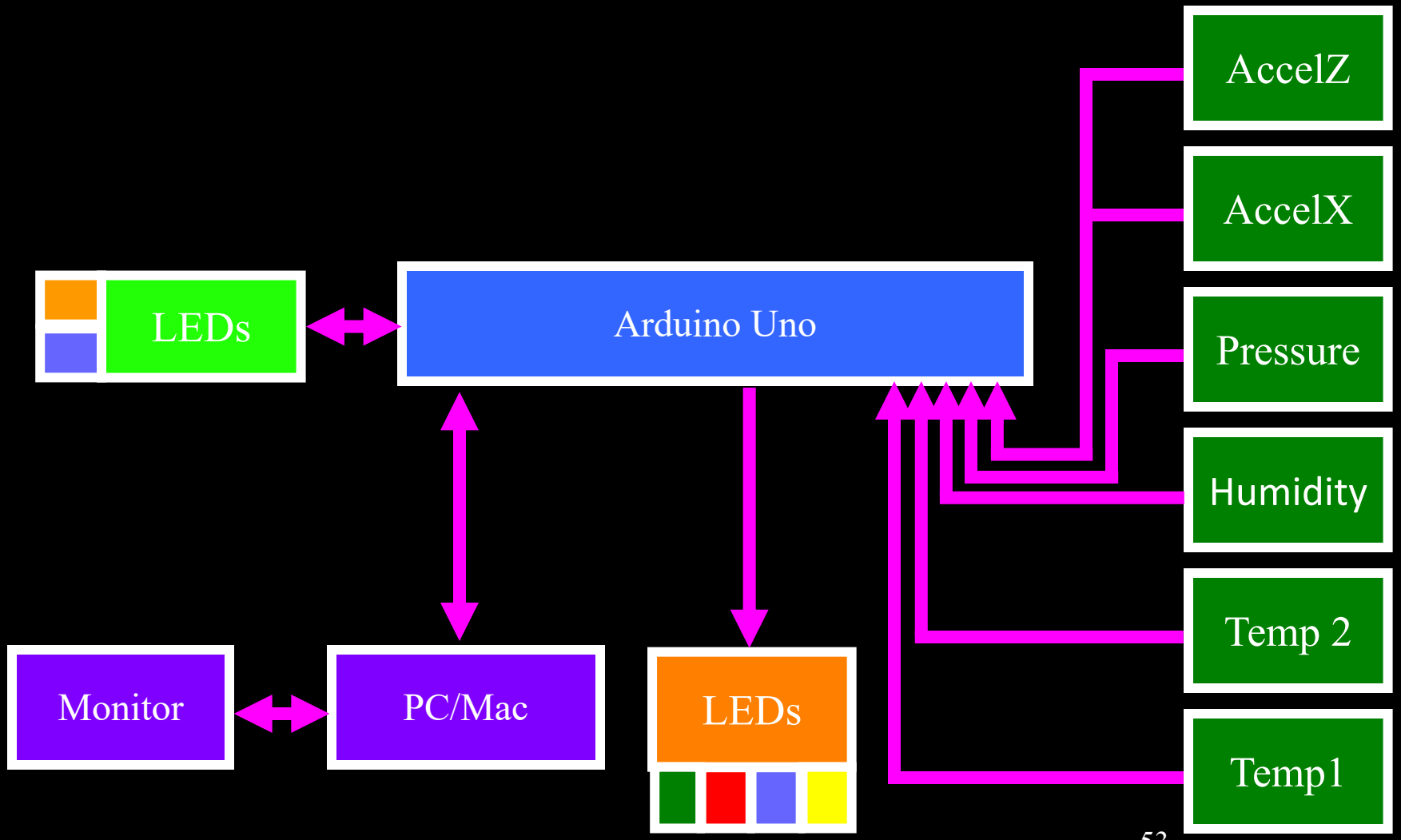


Part 2 – Arduino Road Trip Sensors

A. Humidity Sensor
B. Pressure Sensor
C. Accelerometers
D. External Temp Sensor

External Temperature Sensor:

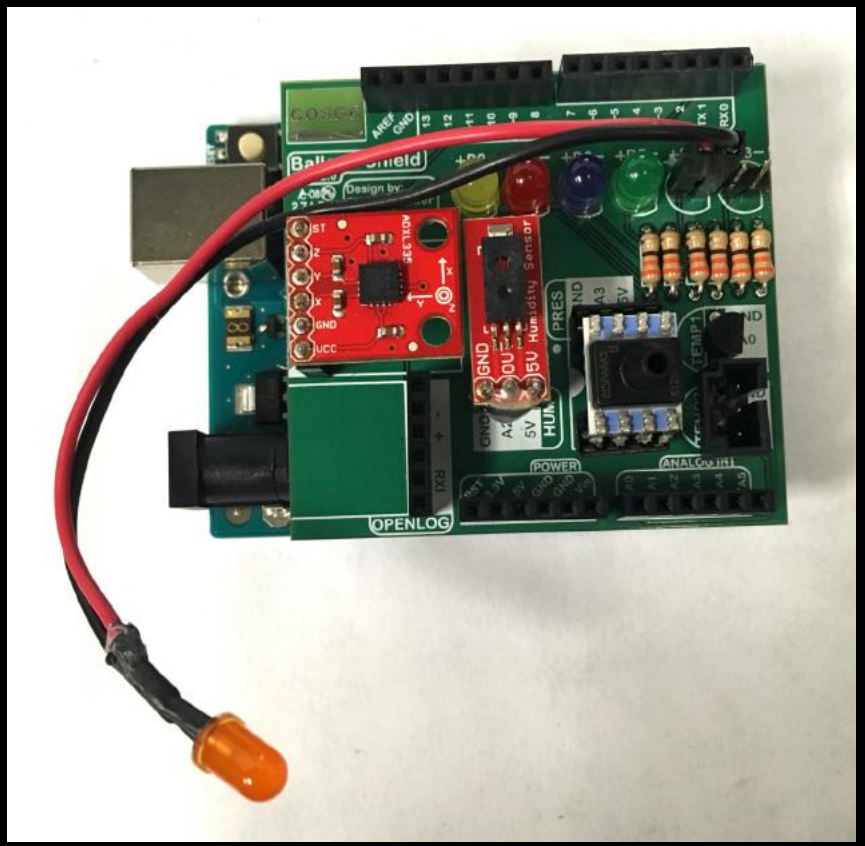
Space Minor
UNIVERSITY OF COLORADO BOULDER

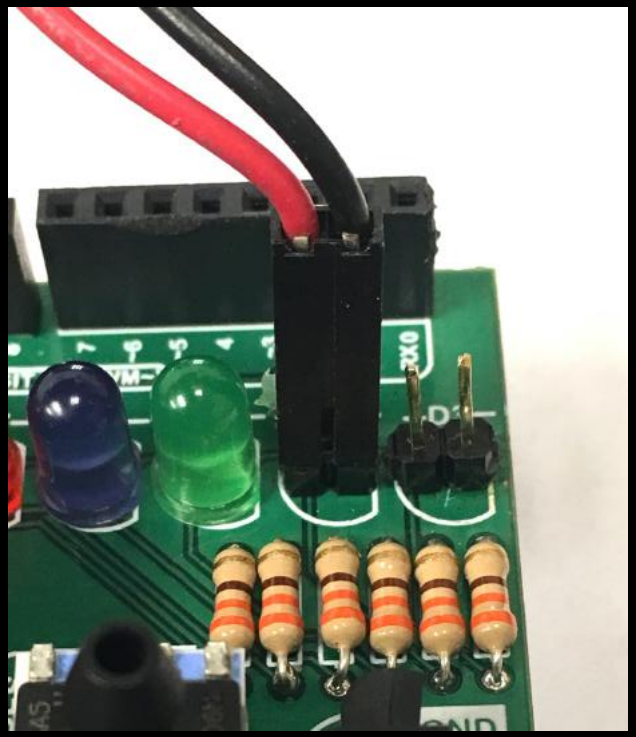


External Temperature Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Add Orange LED to D4
- Red wire to + and Black wire to -

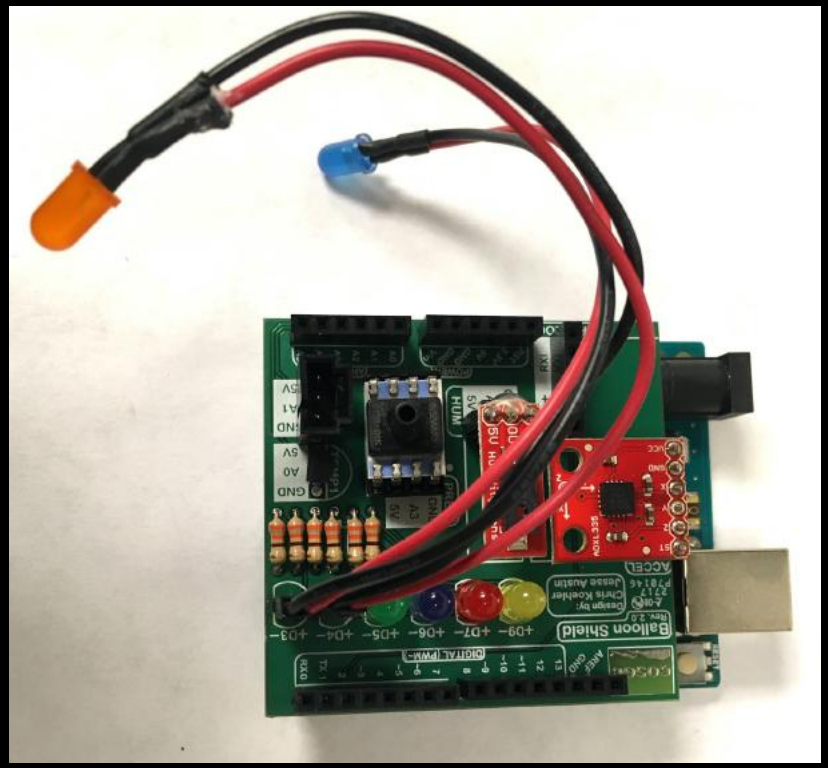


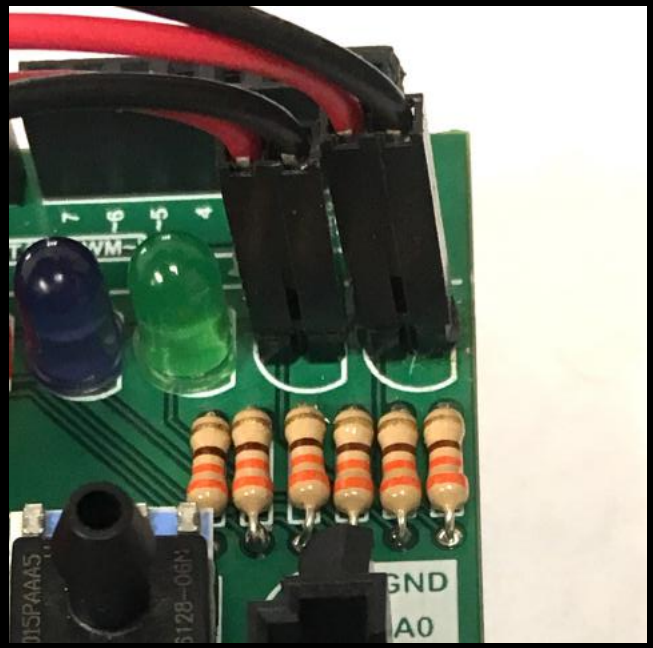


External Temperature Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Add Blue LED to D3
- Red wire to + and Black wire to -

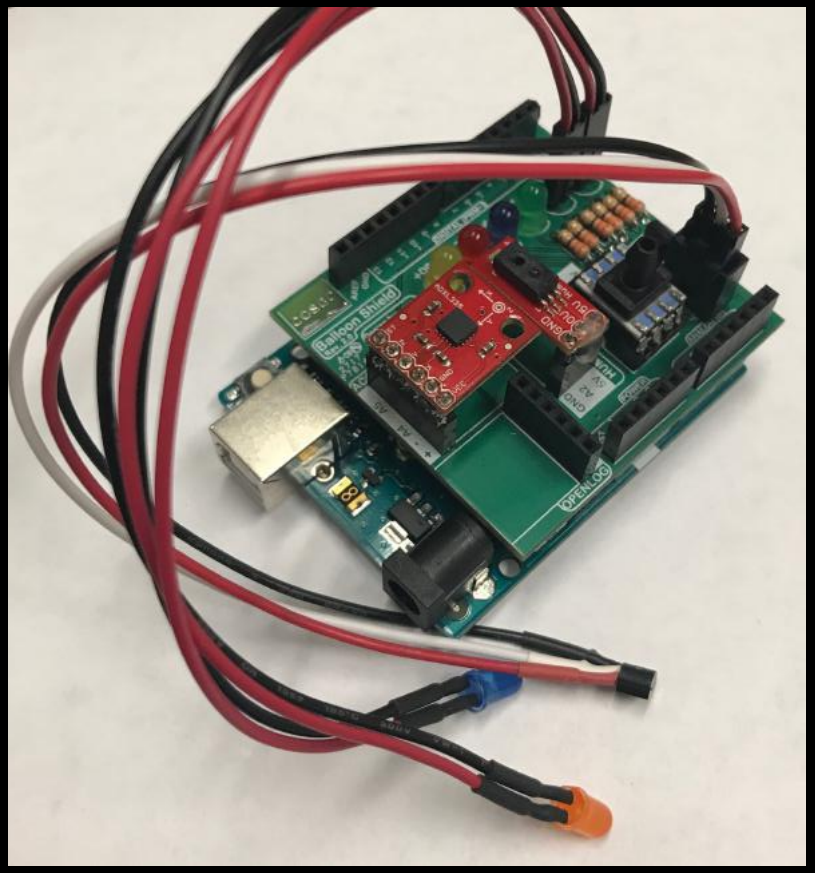


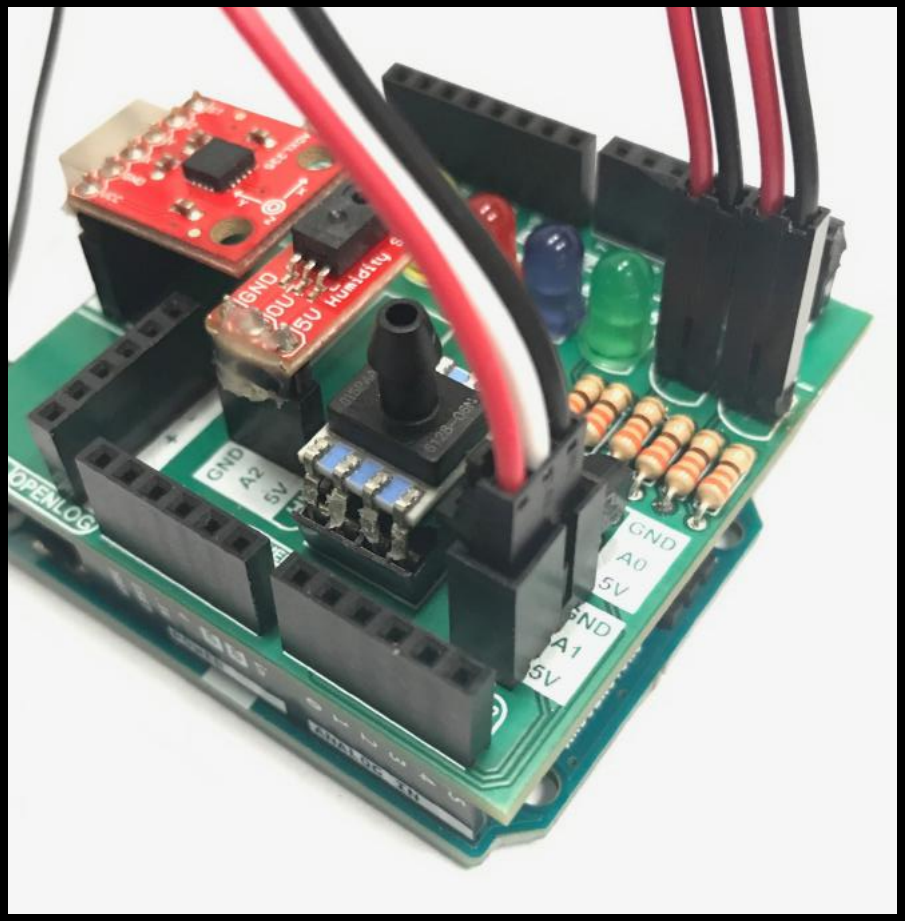


External Temperature Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Add Temp2 to Temp2
- Note wire colors





External Temperature Sensor:

- **Open Temp1 Sketch; save as Temp2**

```
// Definitions
      int sensor;
      float sensorVolt;
      float sensorUnits;
      float sensorUnitsC;
```

```
void setup() {
// put your setup code here, to run once:

      Serial.begin(9600);

// setup the LED Visual Display
      pinMode(3, OUTPUT);  //Blue LED
      pinMode(4, OUTPUT);  //Orange LED
      pinMode(5, OUTPUT);  //Green LED
      pinMode(6, OUTPUT);  //Purple LED
      pinMode(7, OUTPUT);  //Red LED
      pinMode(9, OUTPUT);  //Yellow LED
}
```

Balloon Shield Build Part 6:

Space Minor
UNIVERSITY OF COLORADO BOULDER

```
void loop() {
//   put your main code here, to run repeatedly:

     sensor = analogRead(A1);
     sensorVolt = sensor*(5.0/1023);
     sensorUnitsC = (sensorVolt - 0.5)/(0.01);
     sensorUnits = (sensorUnitsC*(9.0/5.0) + 32);
     Serial.print(sensor);
     Serial.print("\t voltage ");
     Serial.print(sensorVolt);
     Serial.print("\t units ");
     Serial.println(sensorUnits);

//   Turn script running leds OFF at begining of l
     digitalWrite(3, LOW);  //Blue LED
     digitalWrite(4, LOW);  //Orange LED
     digitalWrite(5, LOW);  //Green LED
     digitalWrite(6, LOW);  //Purple LED
     digitalWrite(7, LOW);  //Red LED
     digitalWrite(9, LOW);  //Yellow LED
```

```
if(sensorUnits > 78.0) {
digitalWrite(5, HIGH);
}
if(sensorUnits > 79.0) {
digitalWrite(6, HIGH);
}
if(sensorUnits > 80.0) {
digitalWrite(7, HIGH);
}
if(sensorUnits > 81.0) {
digitalWrite(9, HIGH);
}
digitalWrite(3, HIGH);
digitalWrite(4, HIGH);
delay(100);
```

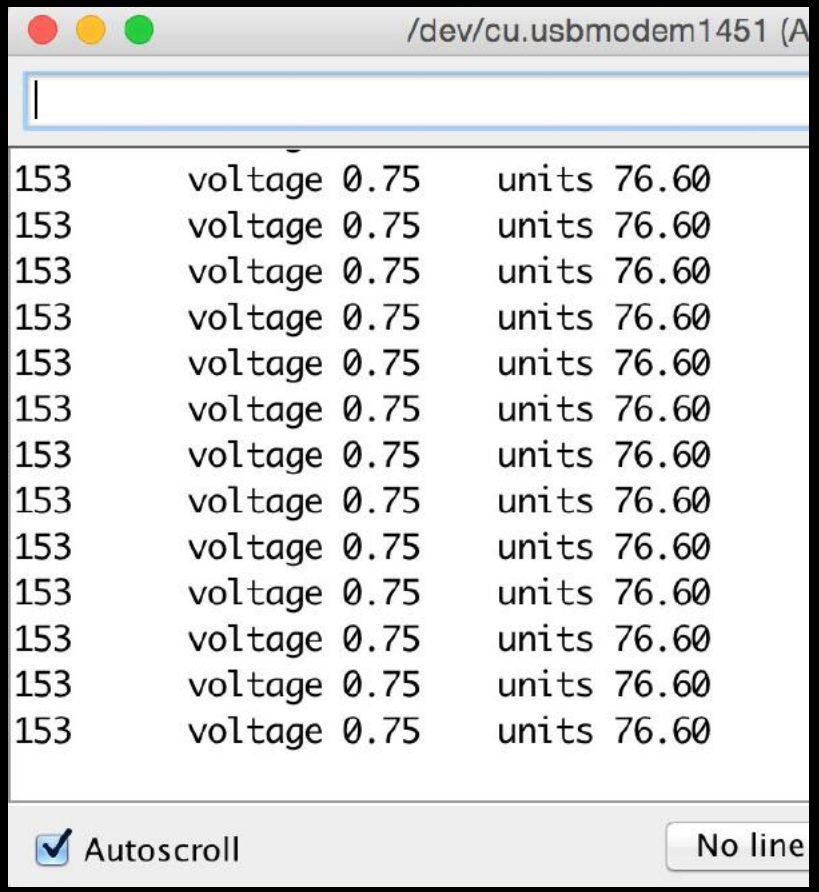
External Temperature Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- **Build and upload your sketch**

- **Temp2 will stick outside your BalloonSat**

- **LED 3 and 4, will also stick outside your BalloonSat**



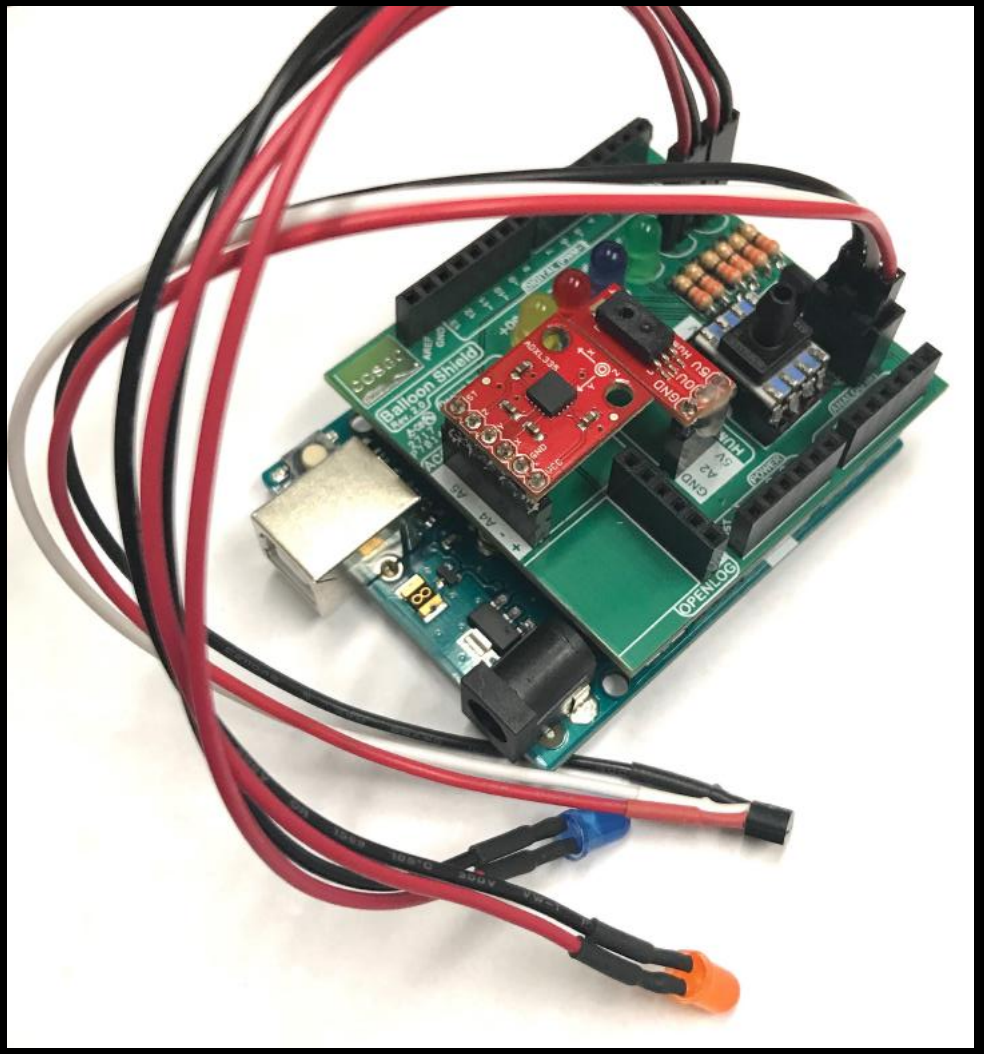
External Temperature Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Build and upload your sketch

- Temp2 will stick outside your BalloonSat

- LED 3 and 4, will also stick outside your BalloonSat

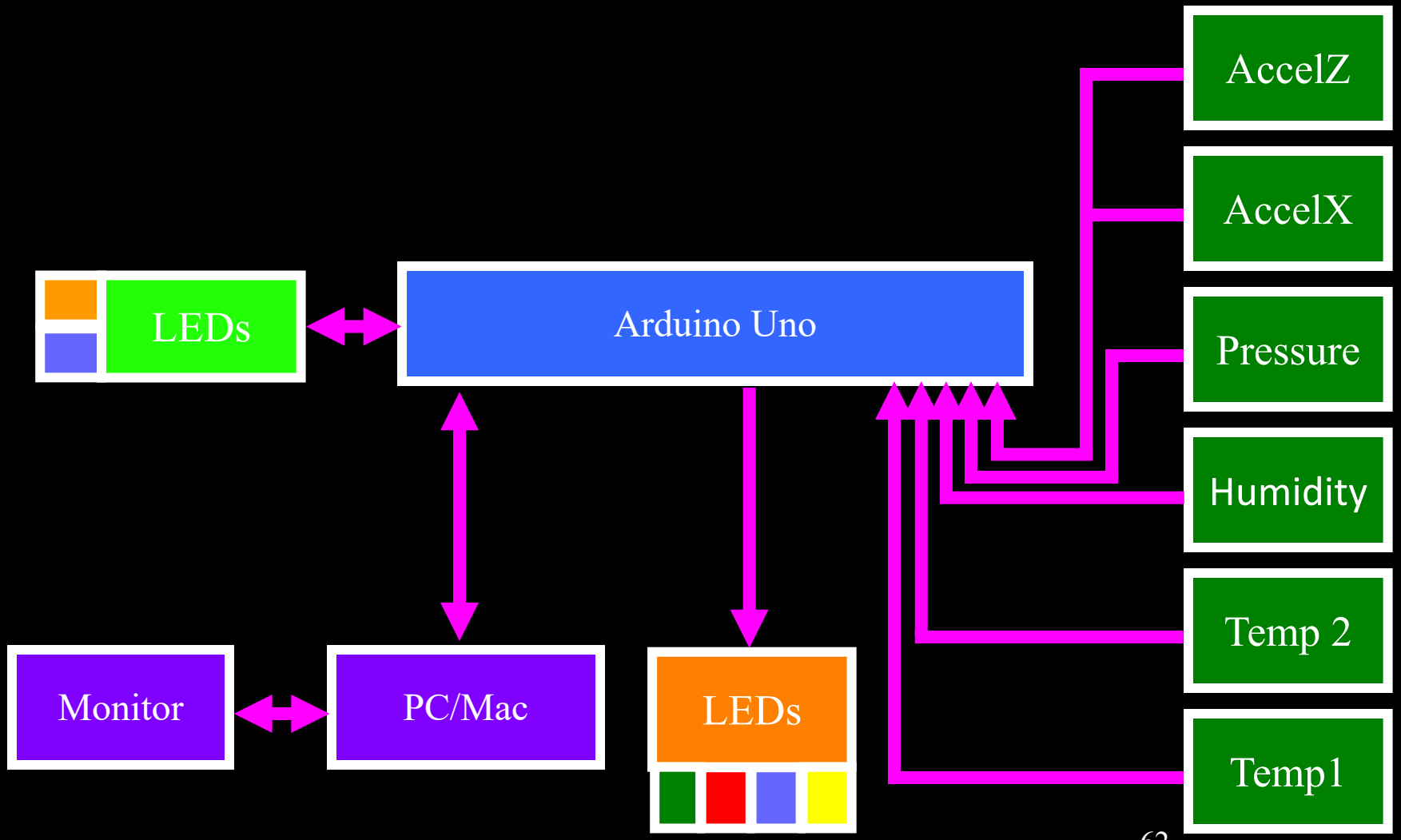


Part 2 – Arduino Road Trip Sensors

A. Humidity Sensor
B. Pressure Sensor
C. Accelerometers
D. External Temp Sensor

External Temperature Sensor:

Space Minor
UNIVERSITY OF COLORADO BOULDER



Part 2 – Arduino Road Trip Sensors

A. Humidity Sensor
B. Pressure Sensor
C. Accelerometers
D. External Temp Sensor

Full Sensor Code Testing:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- **Now let's integrate all the code and sensors together and test**

- **We will review code but you will use a pre-coded sketch**

- **Everything should look familiar**

- **Download code from spacegrant.colorado.edu**
  - **Statewide Programs**
  - **DemoSat Program**

Full Sensor Code Testing:

Space Minor
UNIVERSITY OF COLORADO BOULDER

```
// Definitions
// Temperature Sensor #1
    int temp1;
    float temp1Volt;
    float temp1C;
    float temp1F;

// Temperature Sensor #2
    int temp2;
    float temp2Volt;
    float temp2C;
    float temp2F;

// Humidity Sensor
    int humidity;
    float humidityVolt;
    float RH;
```

```
// Presure Sensor
    int pressure;
    float pressureVolt;
    float psi;

// Accelerometer X
    int accelX;
    float accelXVolt;
    float accelXG;

// Accelerometer Z
    int accelZ;
    float accelZVolt;
    float accelZG;
```

Full Sensor Code Testing:

Space Minor
UNIVERSITY OF COLORADO BOULDER

```
// Time keeper
// The time stamp used when recording data points
    uint32_t timeStamp = 0;
```

Full Sensor Code Testing:

Space Minor
UNIVERSITY OF COLORADO BOULDER

```
void setup() {
// put your setup code here, to run once:

      Serial.begin(9600);

// setup the LED Visual Display
      pinMode(3, OUTPUT);  //Arduino on
      pinMode(4, OUTPUT);  //Internal Temp
      pinMode(5, OUTPUT);  //External Temp
      pinMode(6, OUTPUT);  //Humidity
      pinMode(7, OUTPUT);  //Pressure
      pinMode(9, OUTPUT);  //Accels

// turn on Arduino LED
      digitalWrite(3, HIGH);  // Leave on while power is on

// Print Column Headers
      Serial.println("Time,Temp1F,Temp2F,RH,Pres,AccX,AccZ");
}
```

Full Sensor Code Testing:

Space Minor
UNIVERSITY OF COLORADO BOULDER

```
void loop() {
//   put your main code here, to run repeatedly:

//   Turn script running leds OFF at begining of loop
     digitalWrite(4, LOW);
     digitalWrite(5, LOW);
     digitalWrite(6, LOW);
     digitalWrite(7, LOW);
     digitalWrite(9, LOW);

     delay(500);  //Amount of time between samples (milliseconds)

// Log the time
     timeStamp = millis();
     Serial.print(timeStamp);
```

Full Sensor Code Testing:

Space Minor
UNIVERSITY OF COLORADO BOULDER

```
temp1 = analogRead(A0);
temp1Volt = temp1*(5.0/1023);
temp1C = (temp1Volt - 0.5)/(0.01);
temp1F = (temp1C*(9.0/5.0) + 32);
Serial.print(",");
Serial.print(temp1F, 2);
digitalWrite(4, HIGH);

temp2 = analogRead(A1);
temp2Volt = temp2*(5.0/1023);
temp2C = (temp2Volt - 0.5)/(0.01);
temp2F = (temp2C*(9.0/5.0) + 32);
Serial.print(",");
Serial.print(temp2F, 2);
digitalWrite(5, HIGH);
```

Full Sensor Code Testing:

Space Minor
UNIVERSITY OF COLORADO BOULDER

```
humidity = analogRead(A2);
humidityVolt = humidity*(5.0/1023);
RH = (((humidityVolt/5.0)-0.16)/0.0062);
Serial.print(",");
Serial.print(RH, 2);
digitalWrite(6, HIGH);

pressure = analogRead(A3);
pressureVolt = pressure*(5.0/1023);
psi = (pressureVolt-0.5)*(15.0/4.0);
Serial.print(",");
Serial.print(psi, 2);
digitalWrite(7, HIGH);
```

Full Sensor Code Testing:

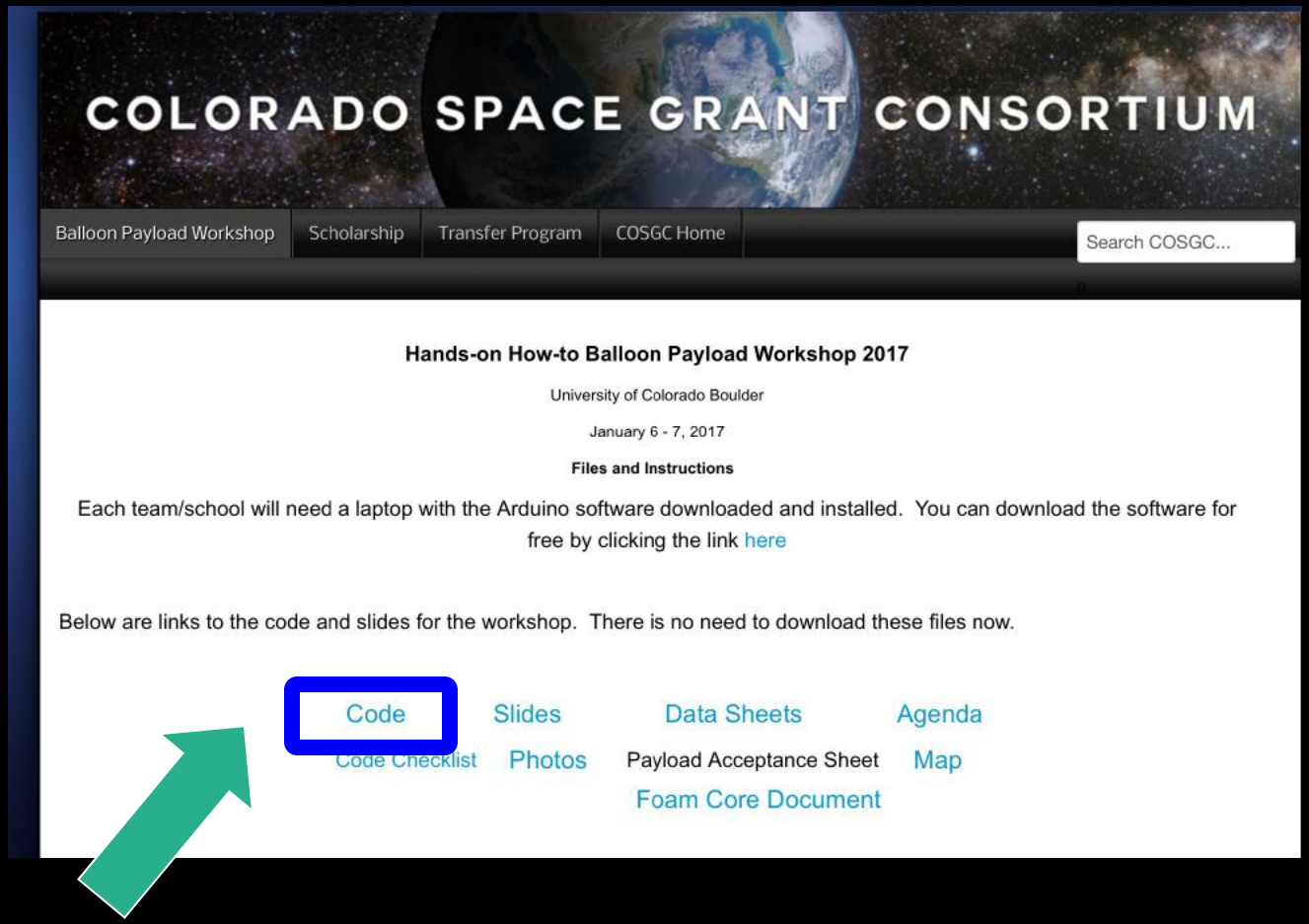
Space Minor
UNIVERSITY OF COLORADO BOULDER

```
    accelX = analogRead(A4);
    accelXVolt = accelX*(5.0/1023);
    accelXG = (accelXVolt - (3.3/2))/(0.330);
    Serial.print(",");
    Serial.print(accelXG,3);

    accelZ = analogRead(A5);
    accelZVolt = accelZ*(5.0/1023);
    accelZG = (accelZVolt - (3.3/2))/(0.330);
    Serial.print(",");
    Serial.print(accelZG,3);
    digitalWrite(9, HIGH);

    Serial.println();

}
```

Full Sensor Code Testing:

Space Minor
UNIVERSITY OF COLORADO BOULDER

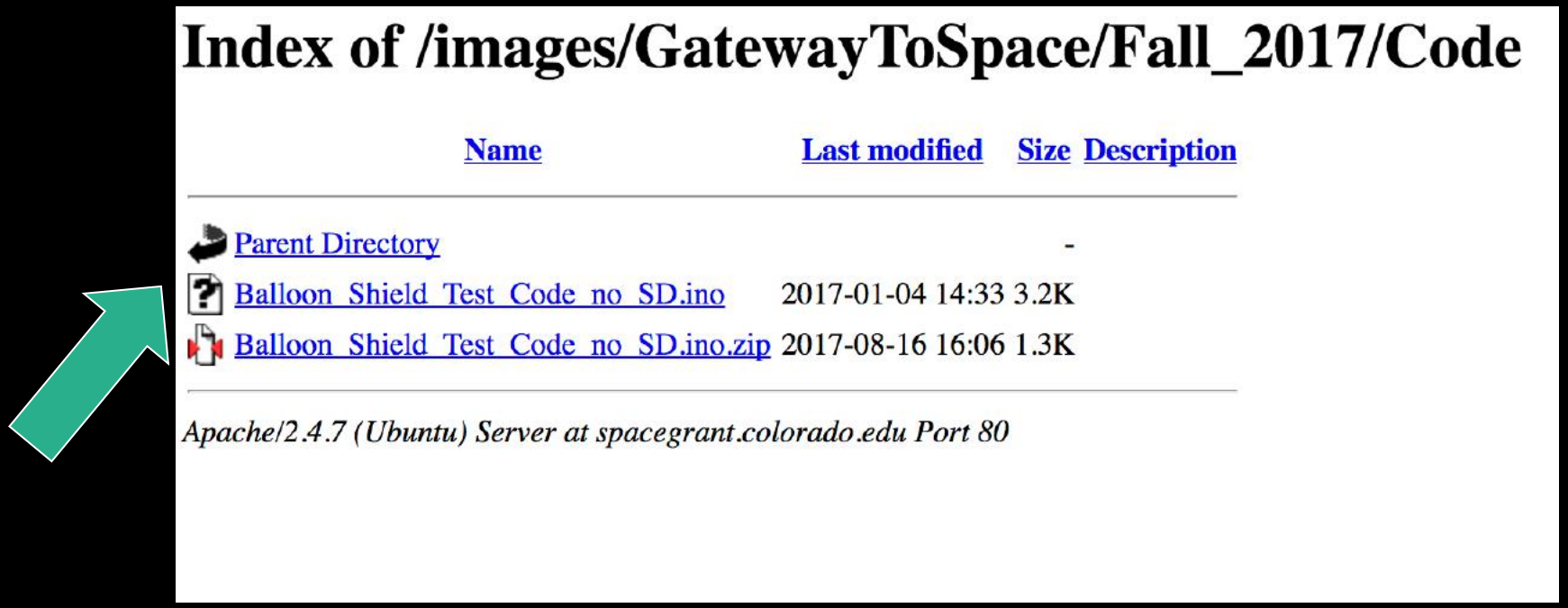
- **Download code or get from desktop and run and verify it works….**



Full Sensor Code Testing:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- **Download code or get from desktop and run and verify it works….**



Index of /images/GatewayToSpace/Fall_2017/Code

Name	Last modified	Size	Description
Parent Directory		-	
Balloon_Shield_Test_Code_no_SD.ino	2017-01-04 14:33	3.2K	
Balloon_Shield_Test_Code_no_SD.ino.zip	2017-08-16 16:06	1.3K	

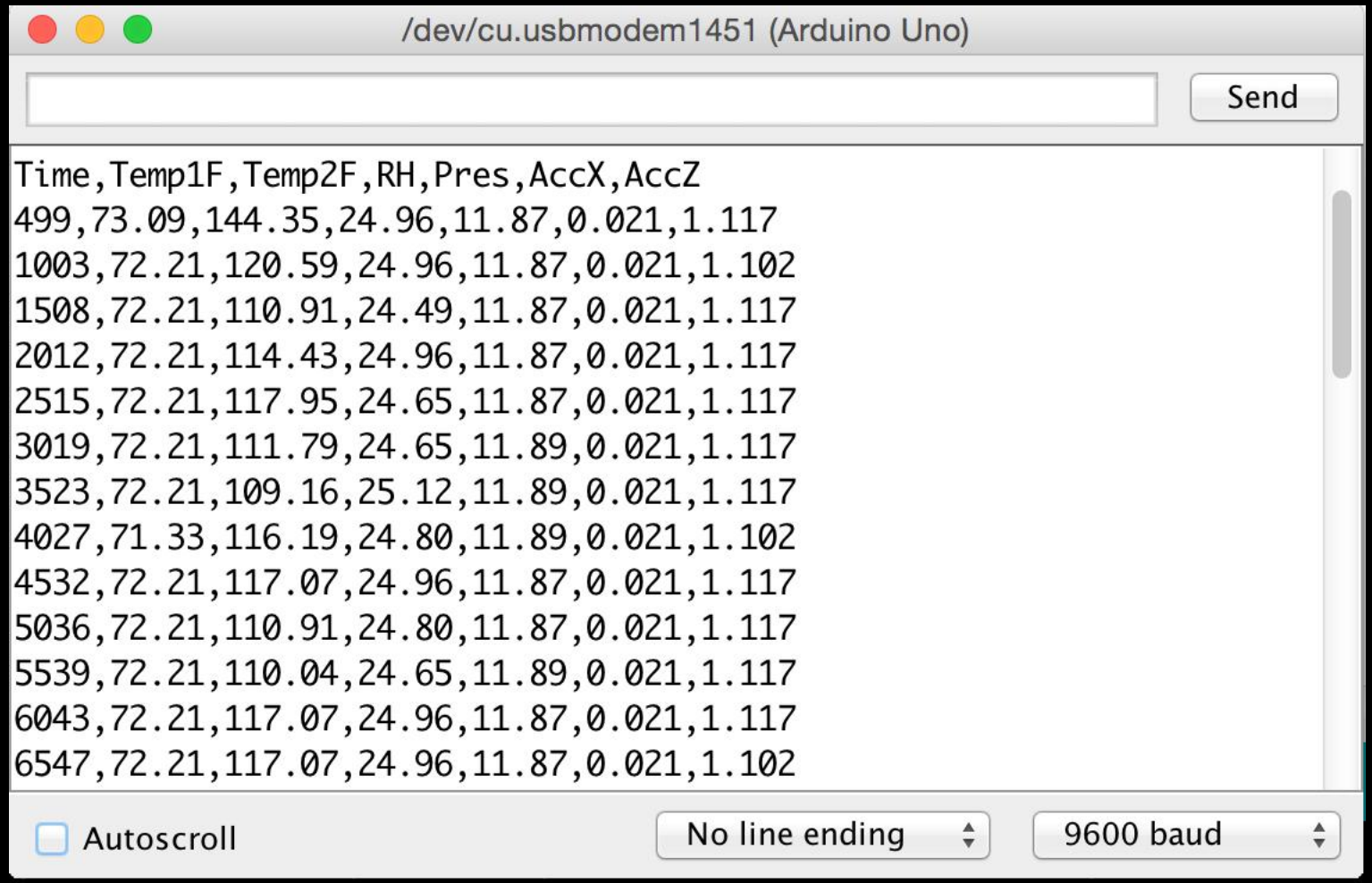
Apache/2.4.7 (Ubuntu) Server at spacegrant.colorado.edu Port 80

If .ino file doesn't work, try downloading the .zip version

Full Sensor Code Testing:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- **Should look like this**



Part 2 – Arduino Race Track Sensors

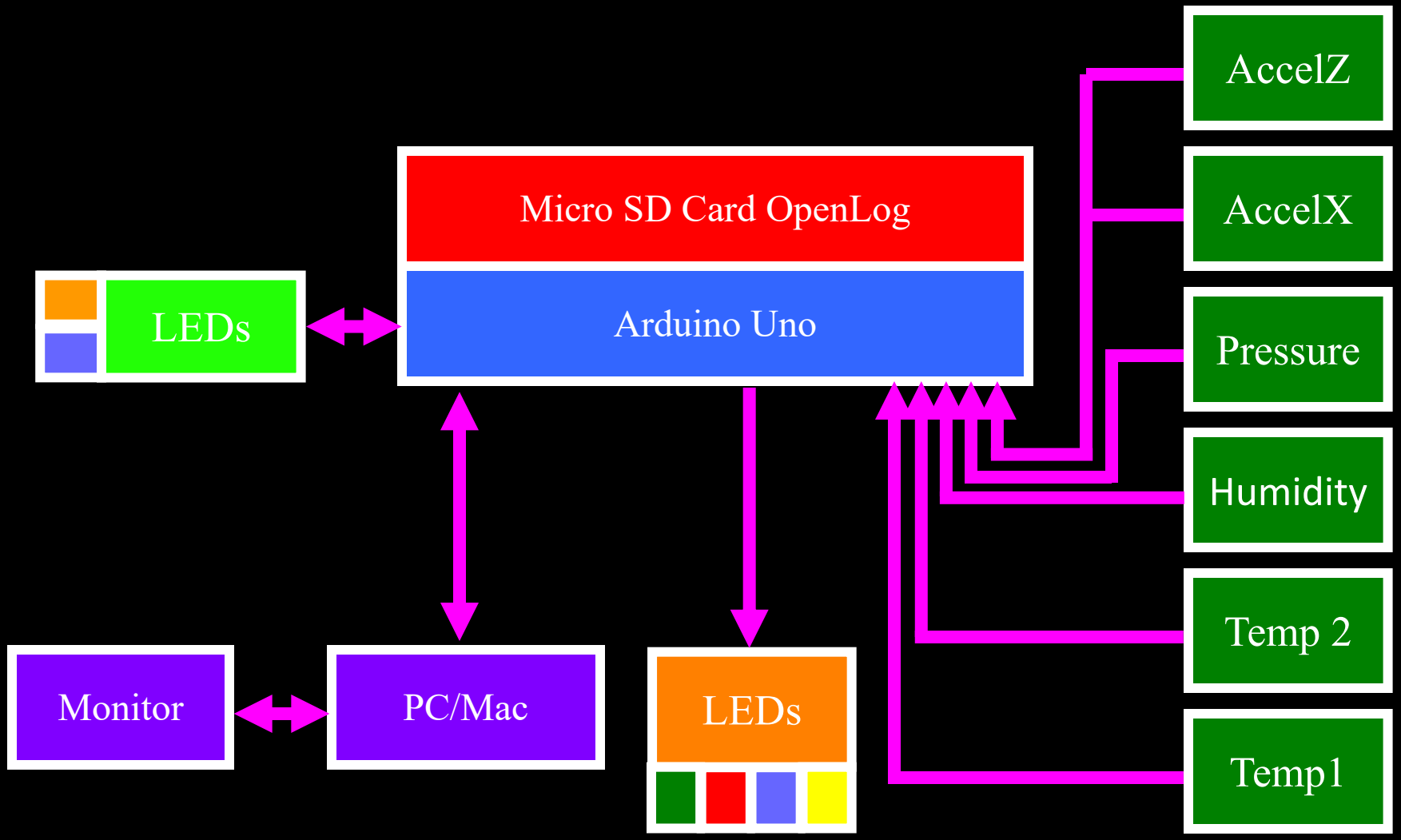
A. OpenLog Integration
B. OpenLog Code Integration
C. Data Retrieval

Part 2 – Arduino Race Track Sensors

A. OpenLog Integration
B. OpenLog Code Integration
C. Data Retrieval

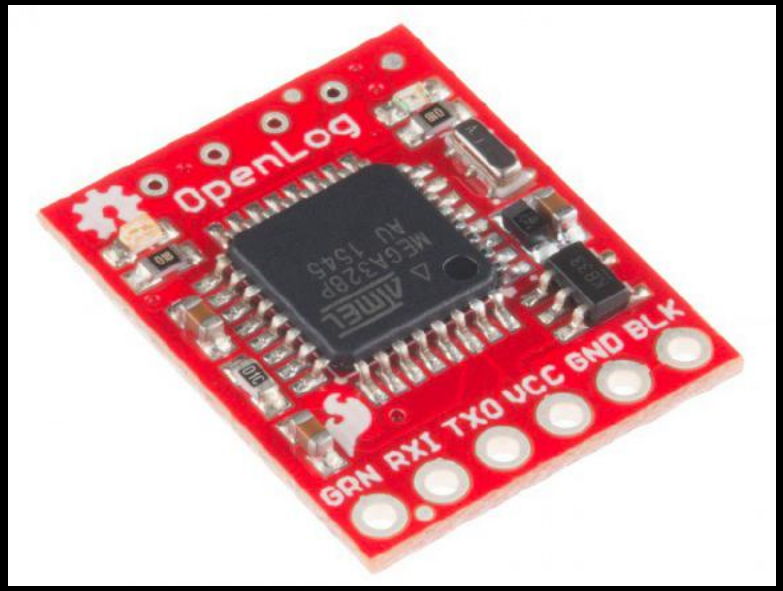
MicroSD Card Shield:

Space Minor
UNIVERSITY OF COLORADO BOULDER



OpenLog:

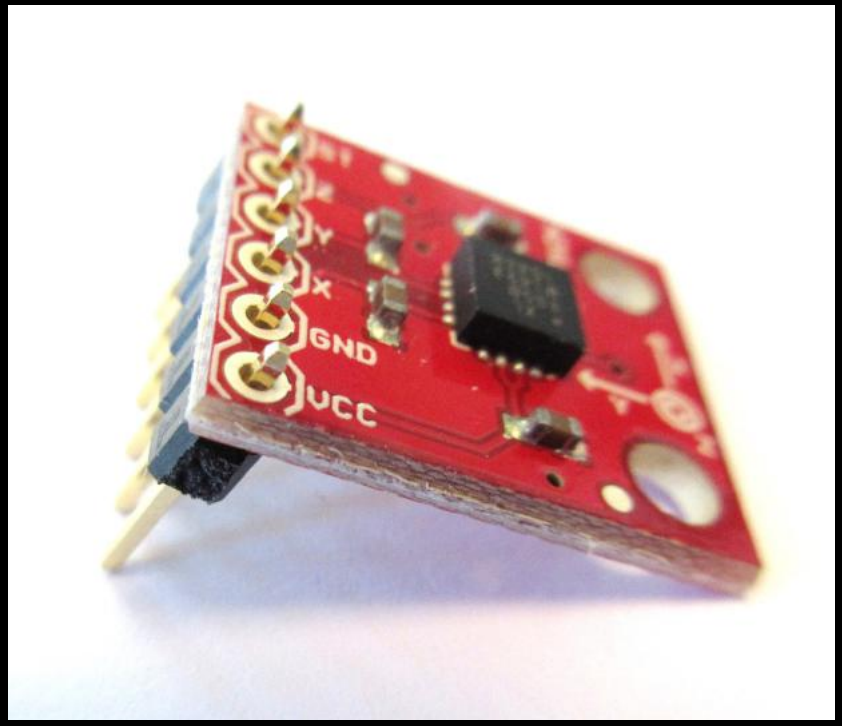

Kingston
TECHNOLOGY
4GB
microSDHC
Kingston
ADAPTER


OpenLog
GRN RXI TXO VCC GND BLK

OpenLog:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Solder 6 pin header to board
- Short side through the bottom of the board
- Keep header perpendicular to board

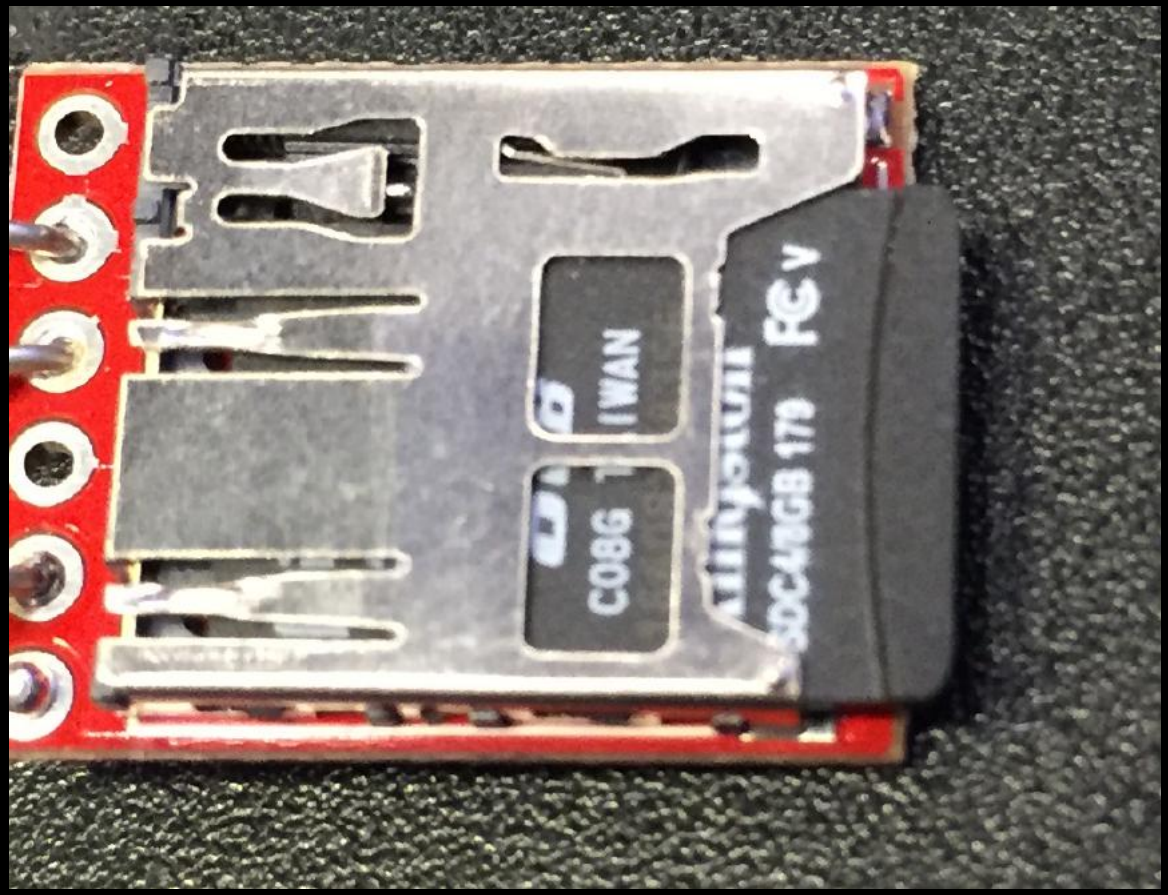


Similar to accelerometer shown here.

Micro SD Card OpenLog:

Space Minor
UNIVERSITY OF COLORADO BOULDER

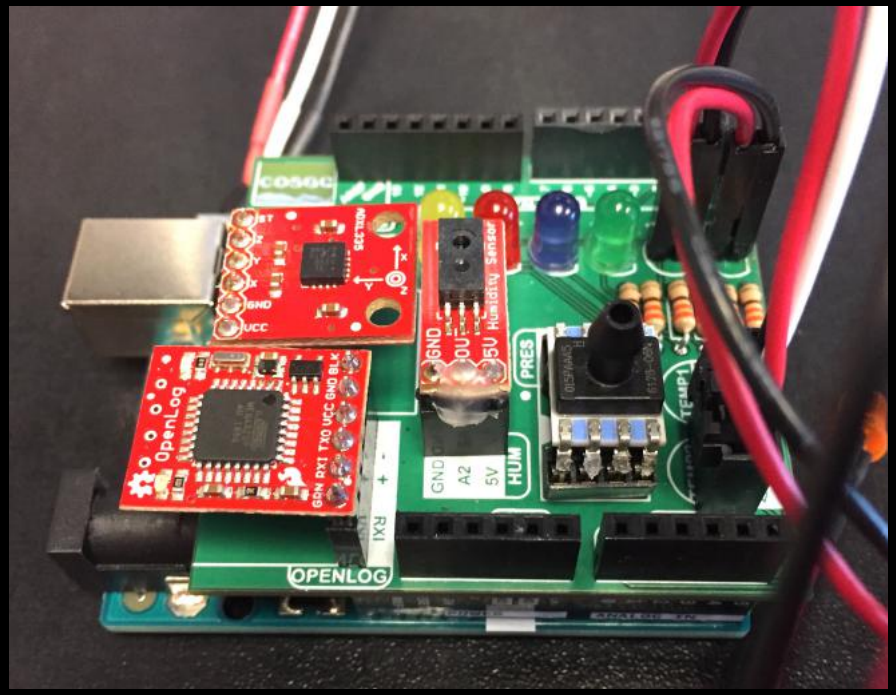
- Insert MicroSD card as shown



Open Log:

Space Minor
UNIVERSITY OF COLORADO **BOULDER**

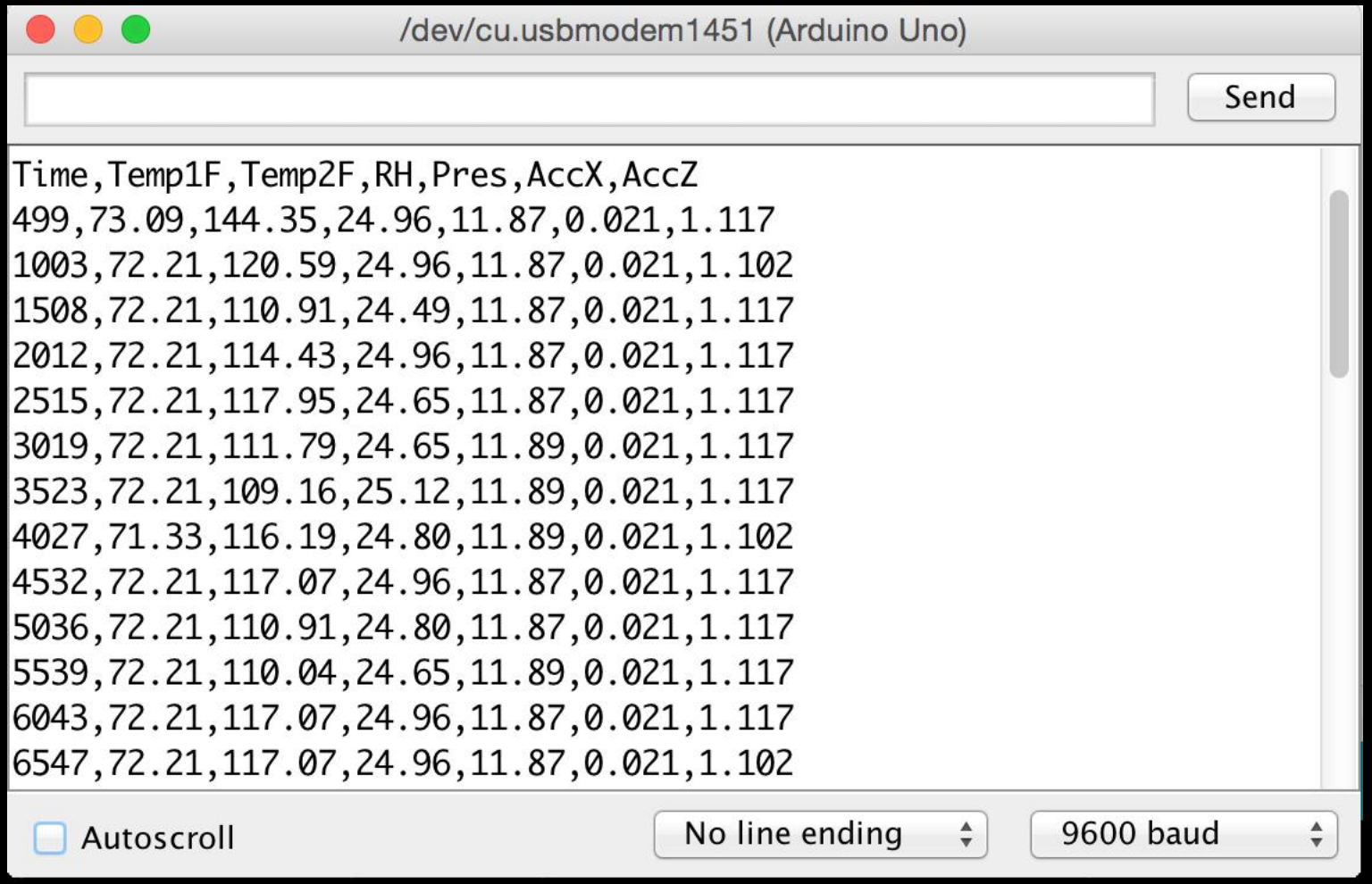
Place OpenLog in correct spot on Balloon Shield



Open Log:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- **Reconnect USB and rerun same code**



Part 2 – Arduino Race Track Sensors

A. OpenLog Integration
B. OpenLog Code
C. Data Retrieval

OpenLog Code:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- **Now let's explore the code needed to record this data to the OpenLog**

OpenLog Code:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- The super cool thing about OpenLog is that anything you serial print is written to the OpenLog

- A new file is created if power is removed

- A new file is created if sd card is removed and re-inserted

- Can eject sd card while powered

Part 2 – Arduino Race Track Sensors

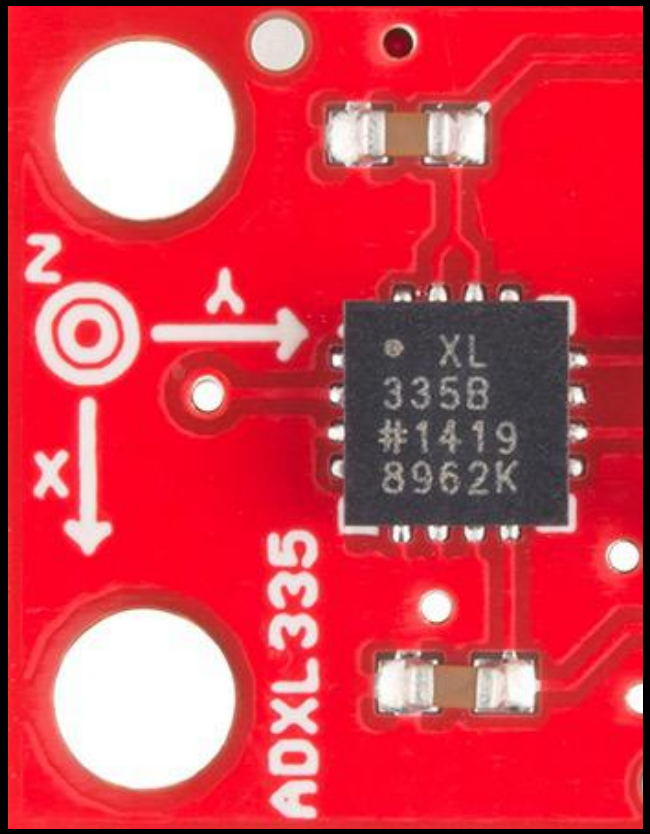
A. OpenLog Integration
B. OpenLog Code
C. Data Retrieval

Sensor Testing:

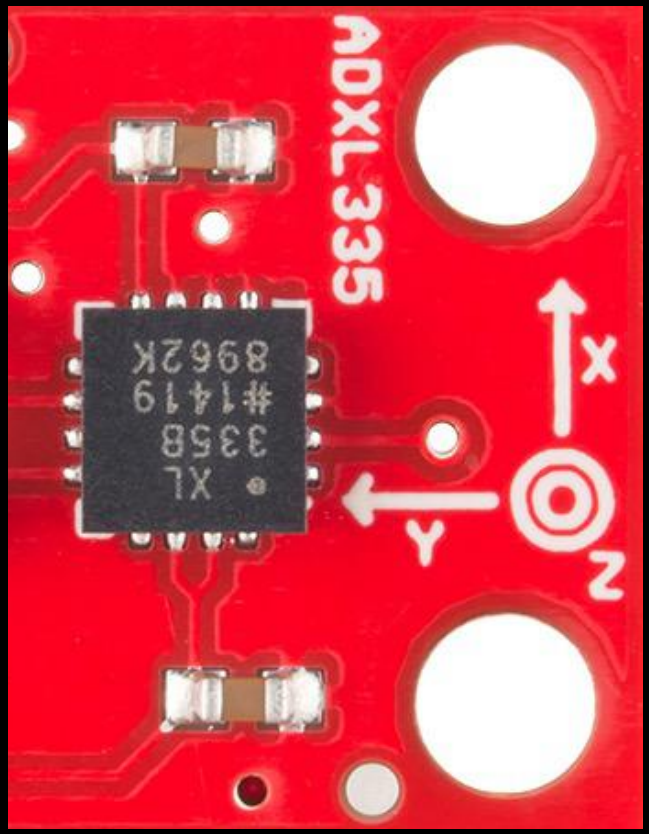
Space Minor
UNIVERSITY OF COLORADO BOULDER

- Rotate your accelerometer like…

4. X Down



5. X Up

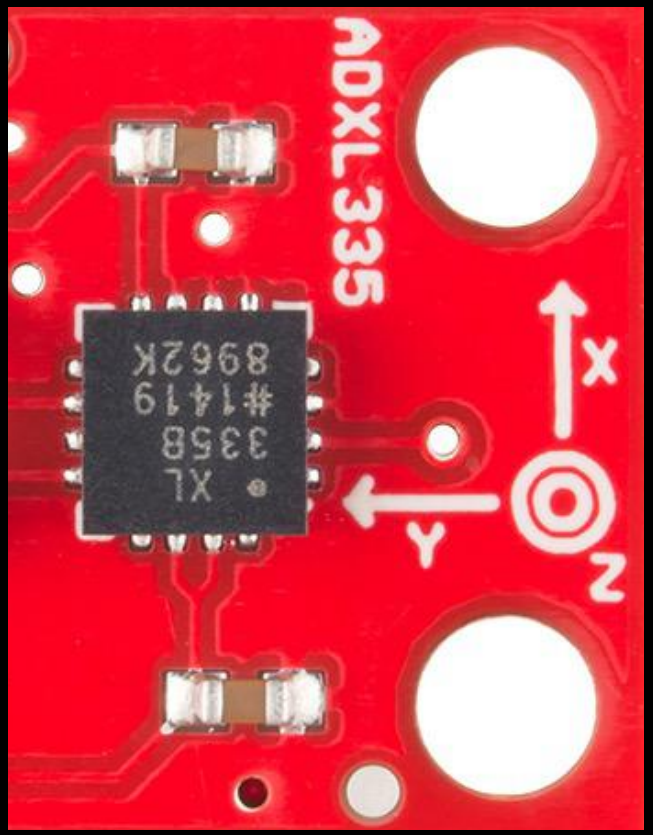


Sensor Testing:

Space Minor
UNIVERSITY OF COLORADO BOULDER

8. Z Down

9. Z UP



Data Retrieval:

Space Minor
UNIVERSITY OF COLORADO BOULDER

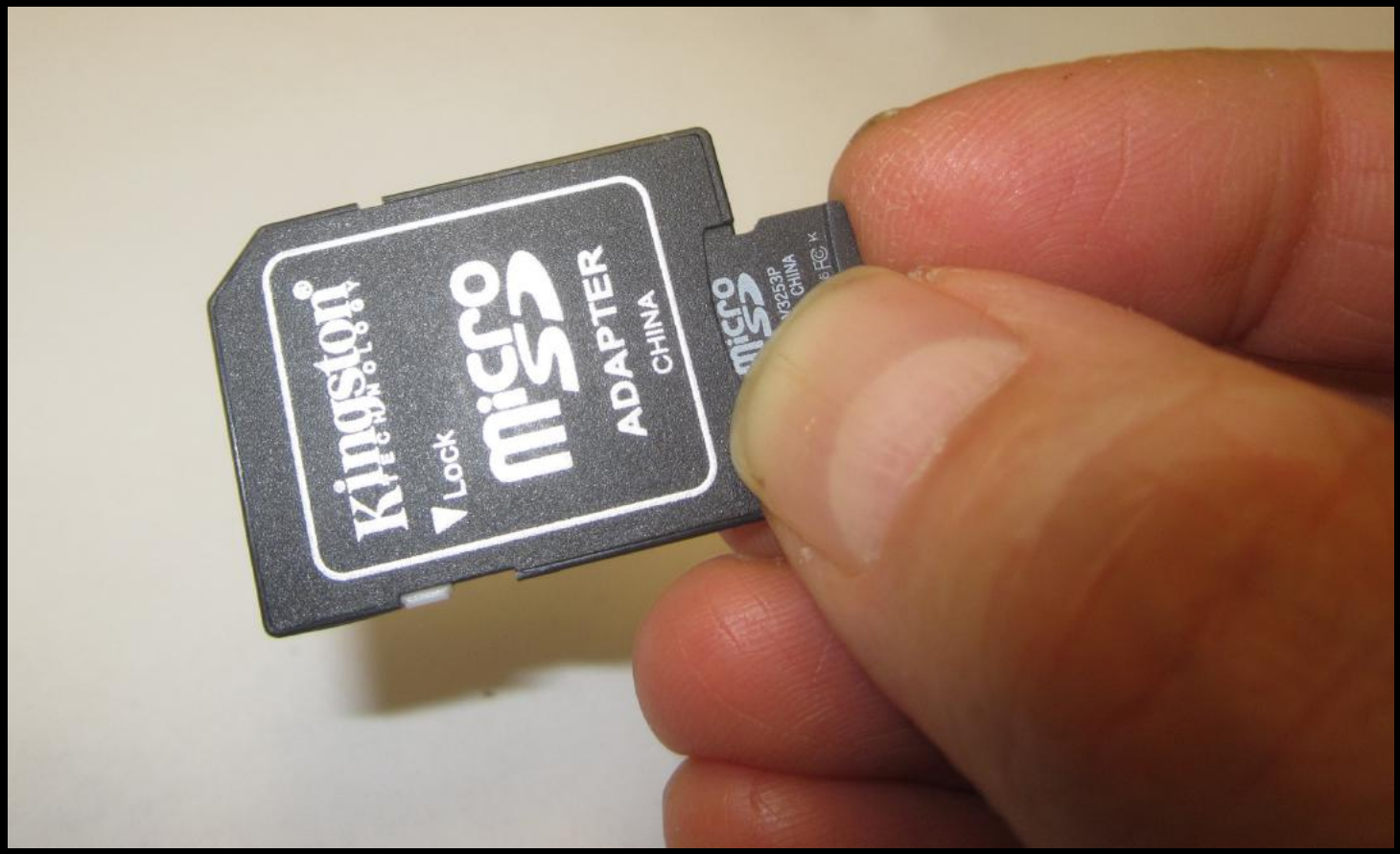
- **Eject the SD card and re-insert.**
- **Then record data as follows:**

1. **Breath on your humidity sensor twice**
2. **Suck on pressure sensor twice**
3. **Touch both temp sensors for 5 seconds each**
4. **Orient your accelerometer (Z up/down, X up/down) 10 seconds each direction**
5. **Breath on your humidity sensor twice**
6. **Suck on pressure sensor twice**
7. **Disconnect USB from Arduino**

Data Retrieval:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- **Remove microSD card from Uno and insert into SD card adapter**



Data Retrieval:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Remove microSD card from Uno and insert into SD card adapter

Data Retrieval:

Space Minor
UNIVERSITY OF COLORADO BOULDER

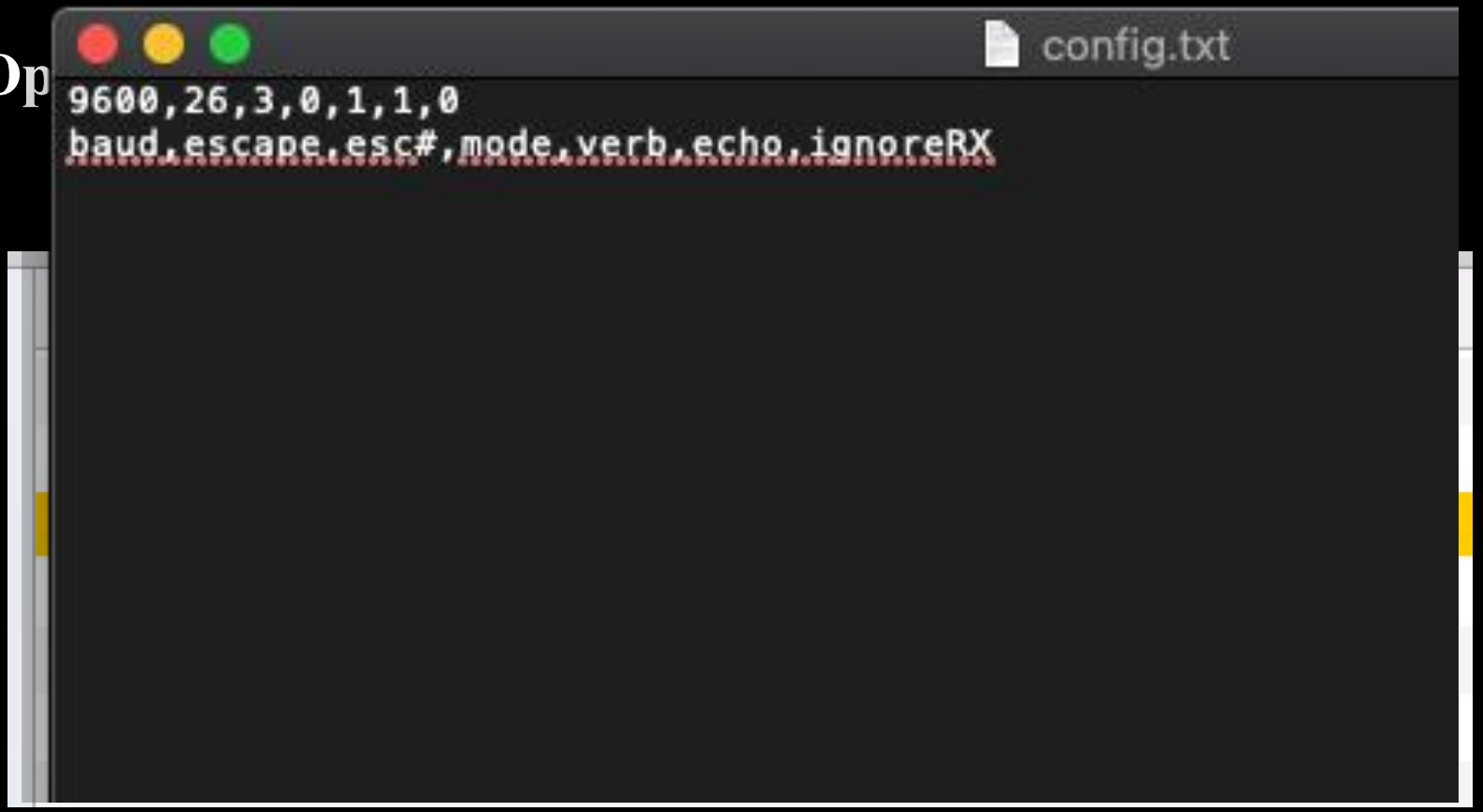
- Insert SD card adapter into your laptop

Data Retrieval:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- Navigate to card and copy last LOG file to your desktop

- Op

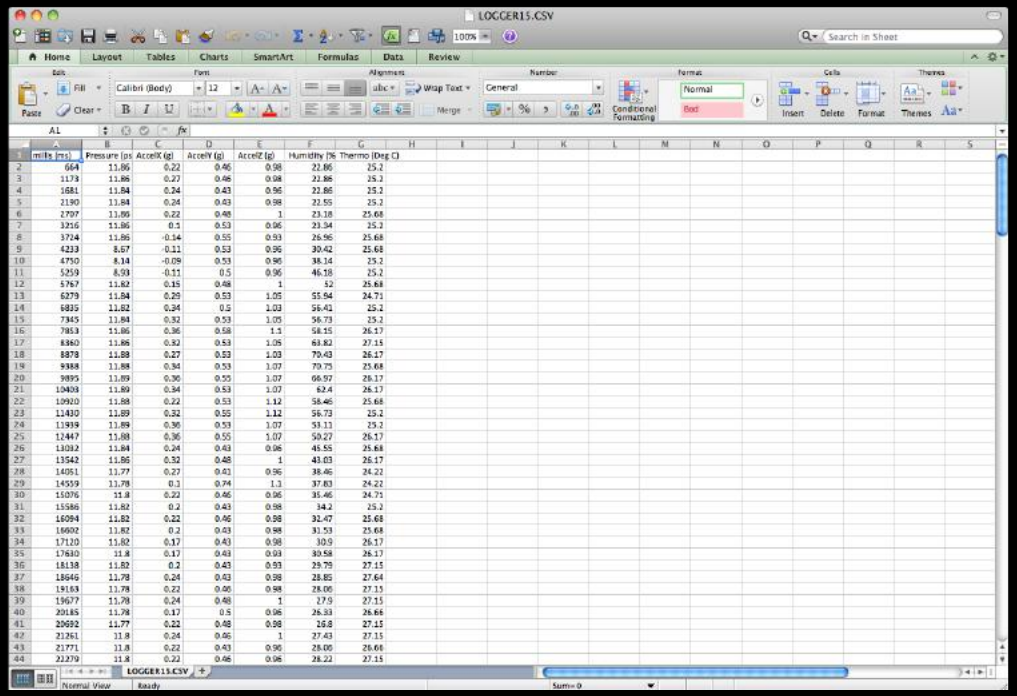


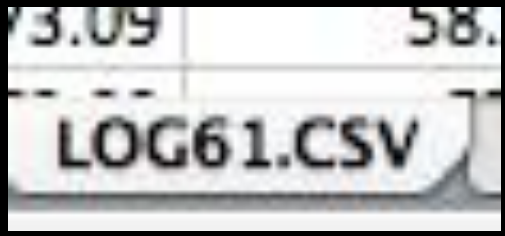
Data Retrieval:

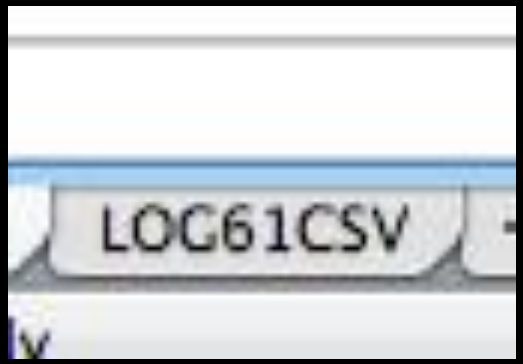
Space Minor
UNIVERSITY OF COLORADO BOULDER

- Graph all data minus the time stamp (Using Excel)

- Mac Users you must change tab name to remove “.”



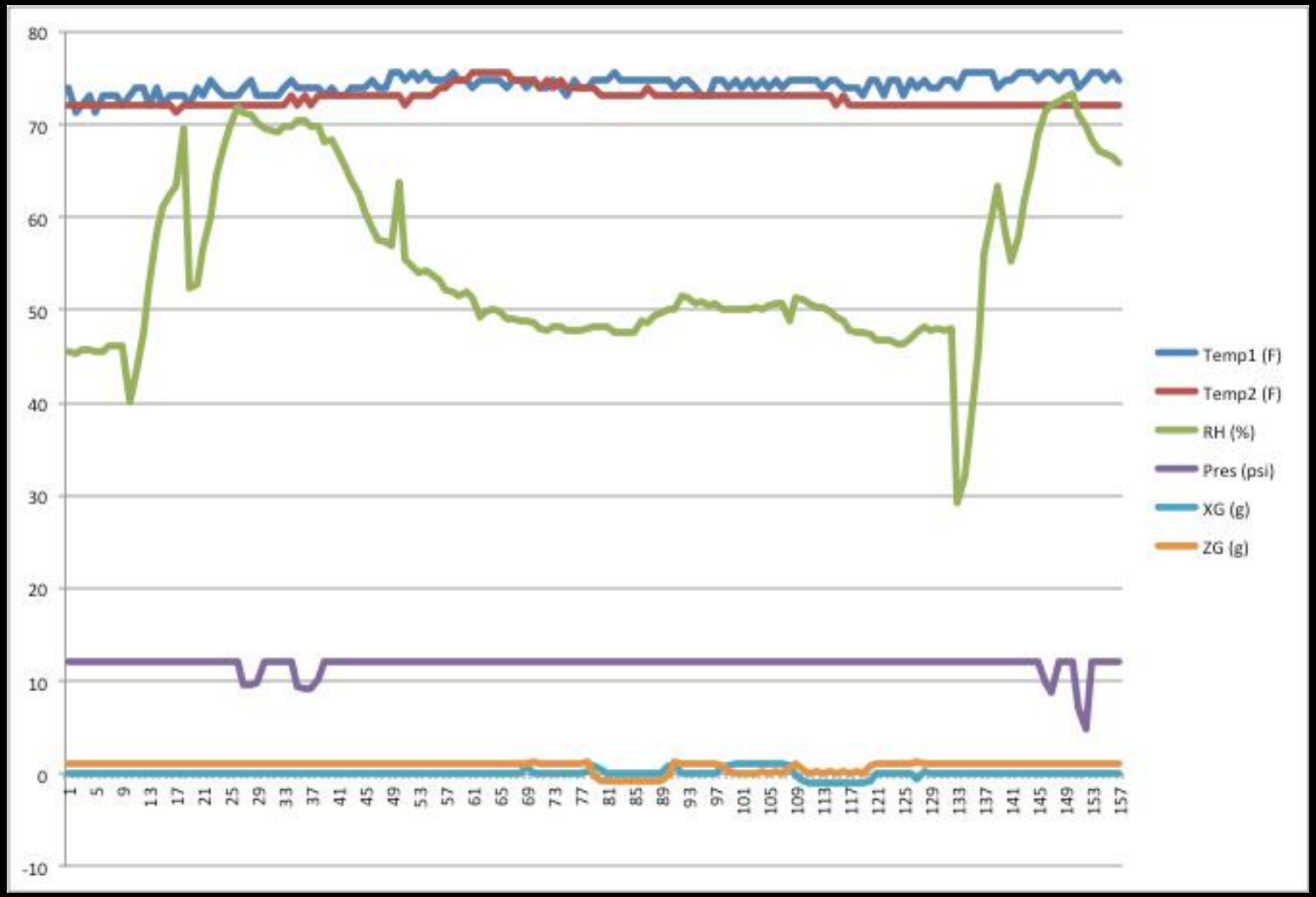




Data Retrieval:

Space Minor
UNIVERSITY OF COLORADO BOULDER

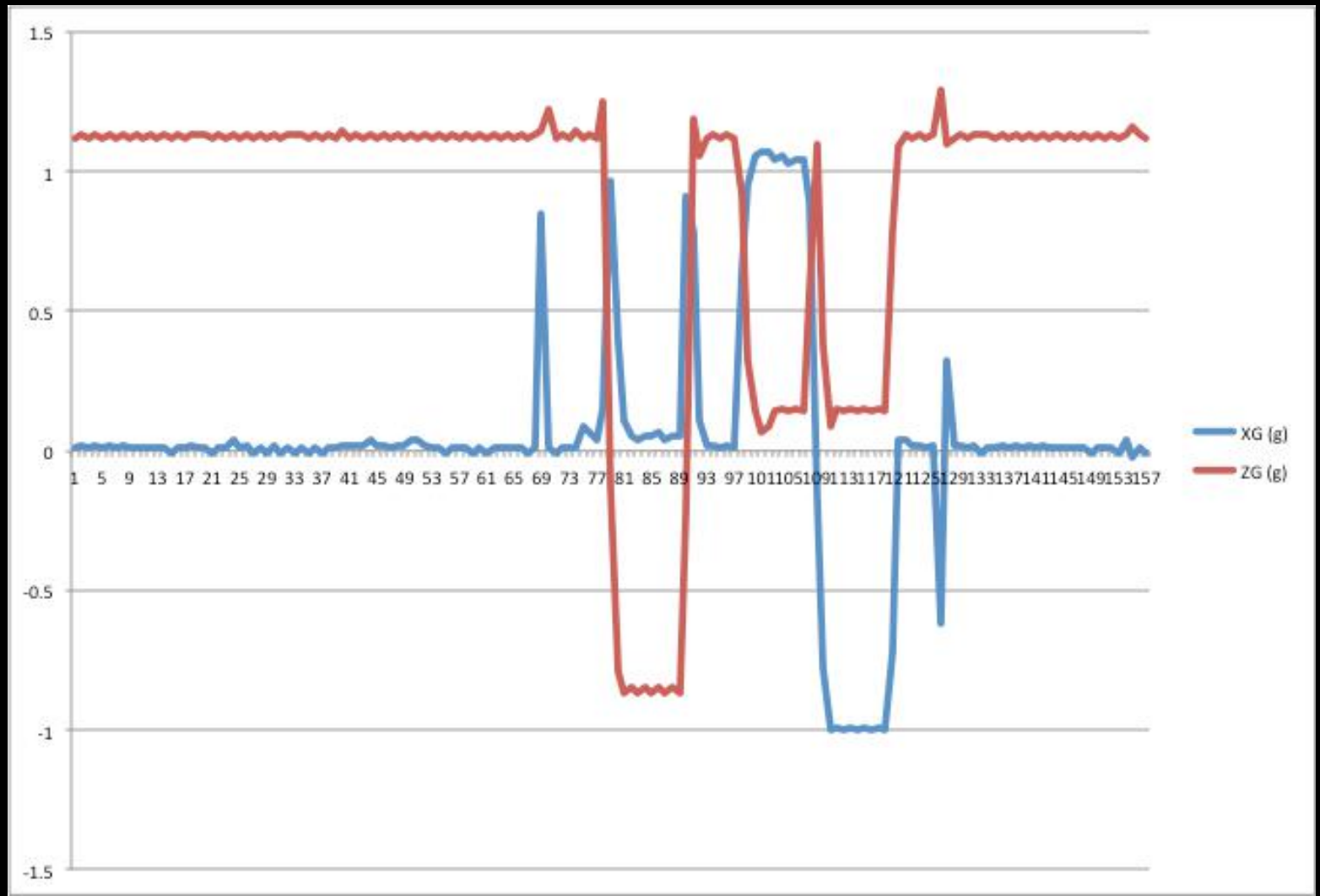
- Do you see your data markers?



Data Retrieval:

Space Minor
UNIVERSITY OF COLORADO BOULDER

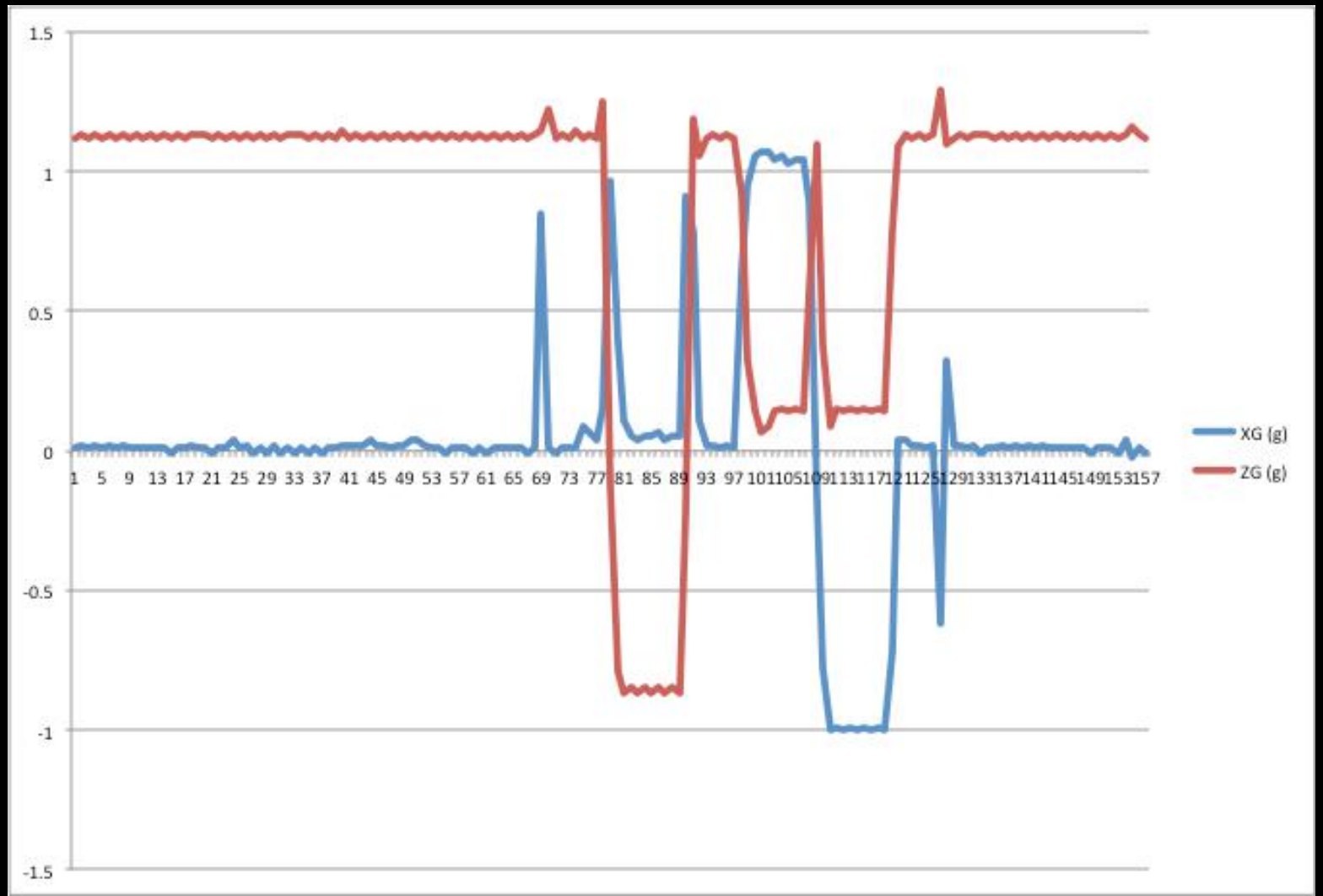
- Re-plot just your accel data



Data Retrieval:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- How can you use this data?



Part 2 – Arduino Race Track Sensors

A. OpenLog Integration
B. OpenLog Code
C. Data Retrieval

SUCCESS

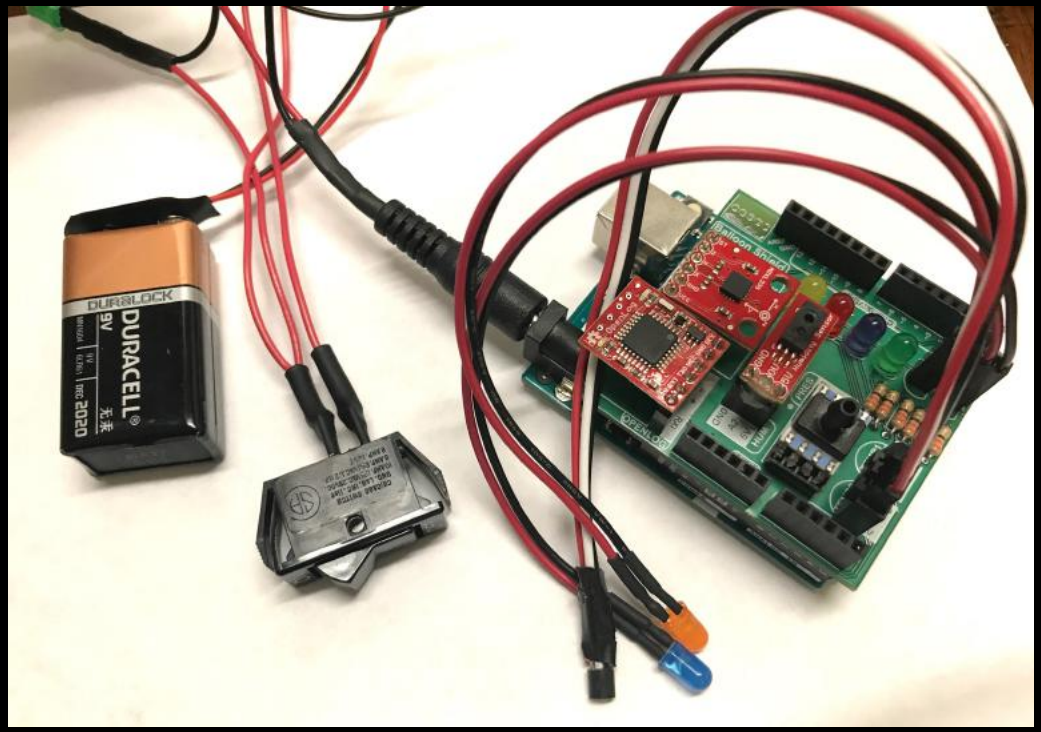
Because you too can own this face of pure accomplishment

Alternate Power:

Space Minor
UNIVERSITY OF COLORADO BOULDER

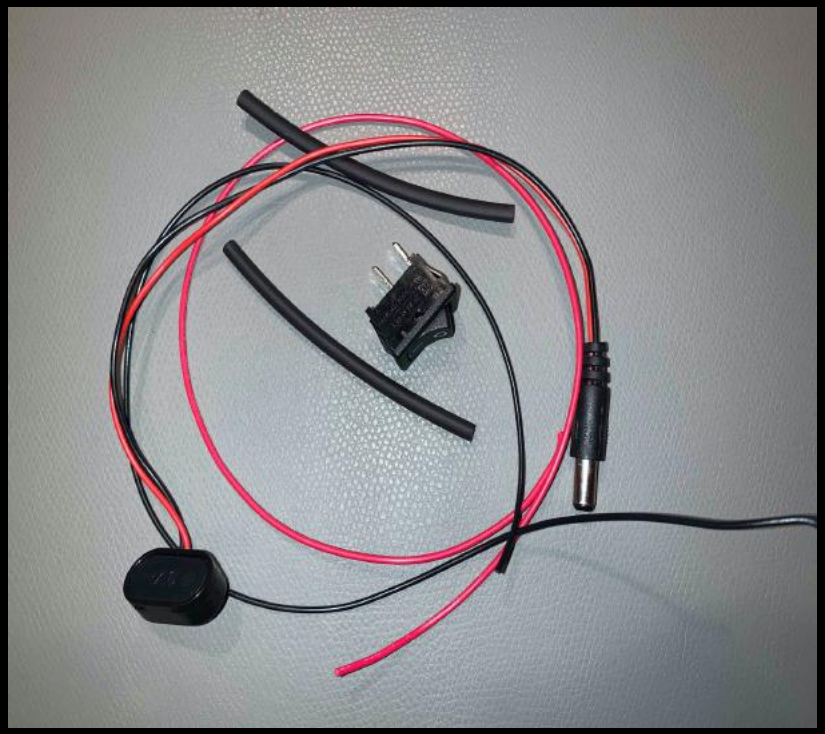
- For balloon flight, need to power Arduino with 9V battery

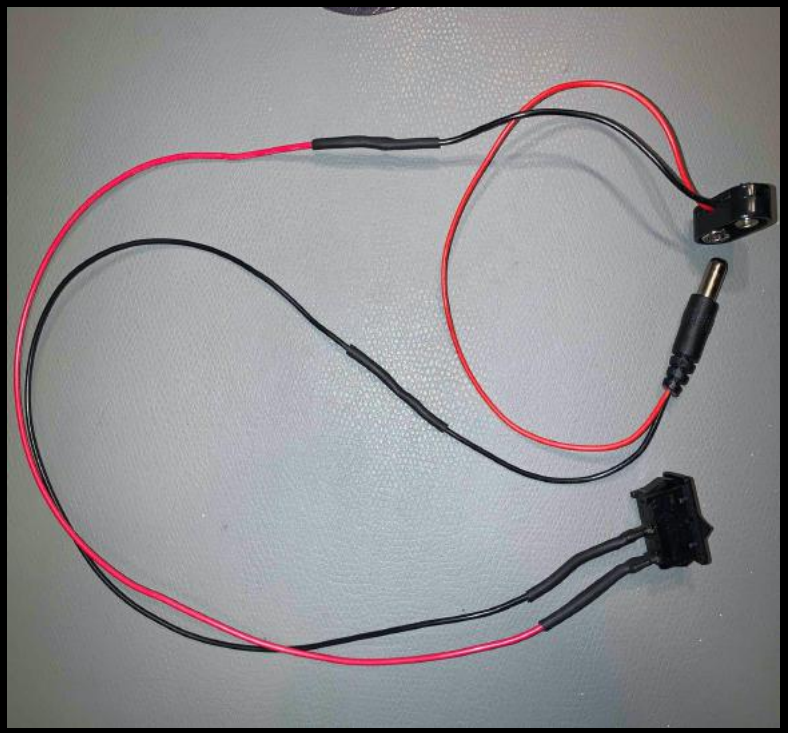
- Do not connect USB and 9V ever



Alternate Power:

Space Minor
UNIVERSITY OF COLORADO **BOULDER**

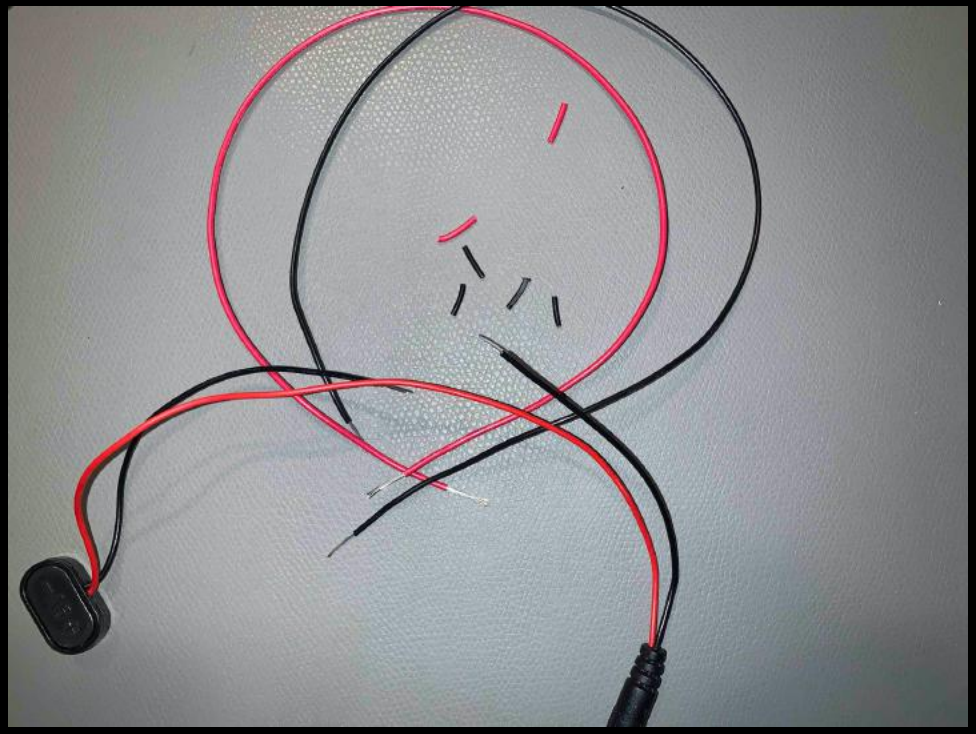




Alternate Power:

Space Minor
UNIVERSITY OF COLORADO BOULDER

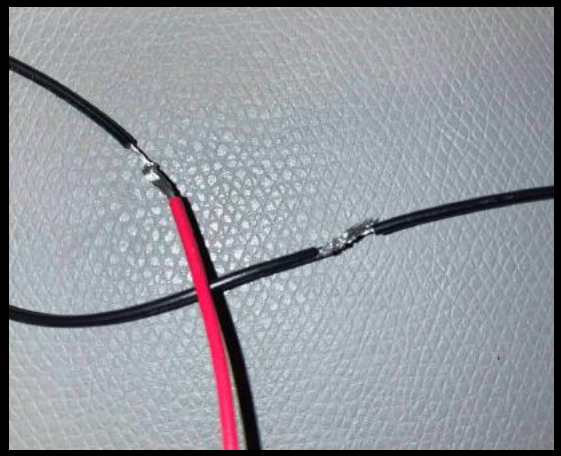
- **Cut Red and Black wire to ~1 foot in length**
- **Cut barrel connector black wire in half**
- **Strip ends of cut wire back ~1/3 inch**

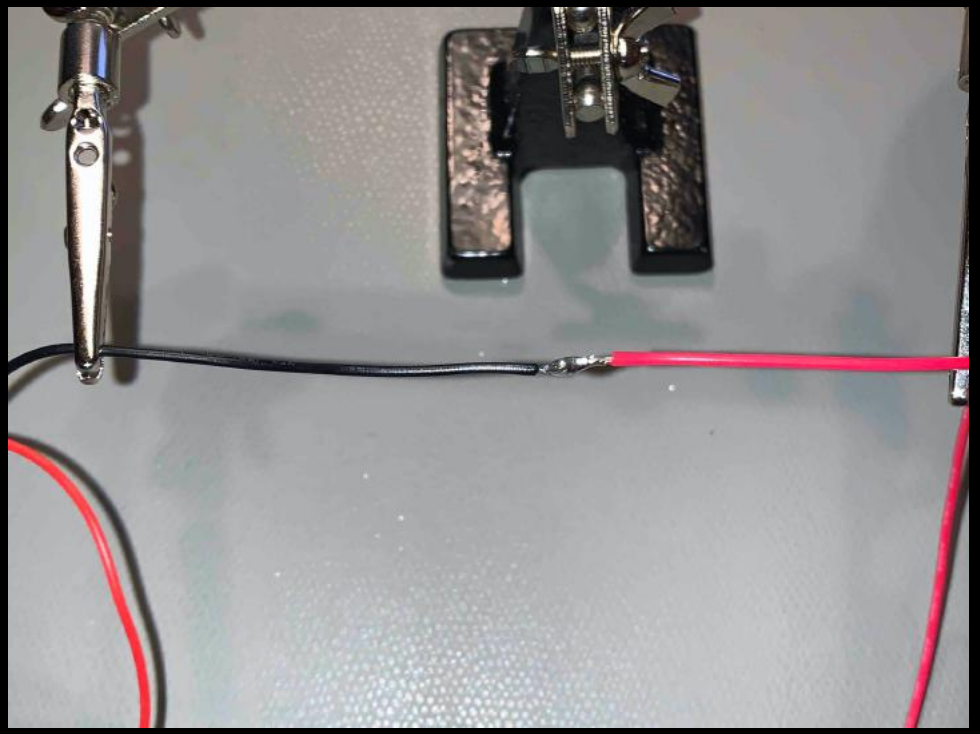


Alternate Power:

Space Minor
UNIVERSITY OF COLORADO **BOULDER**

- **Splice red and black extensions into connector and solder**

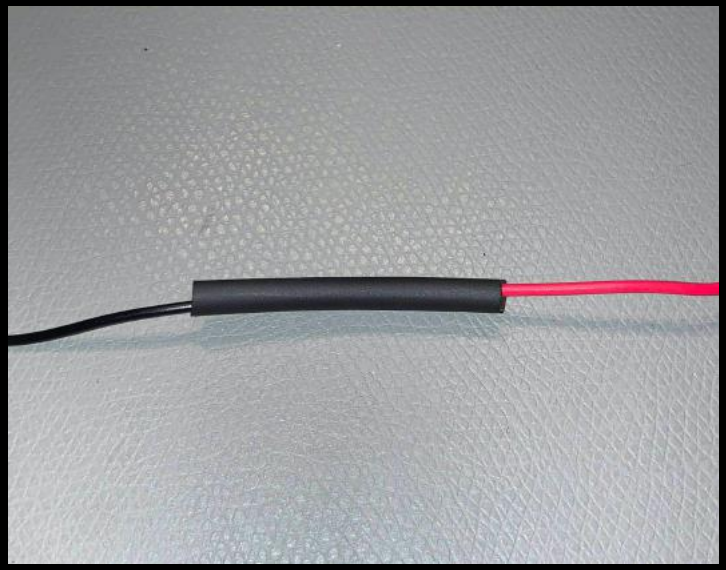




Alternate Power:

Space Minor
UNIVERSITY OF COLORADO **BOULDER**

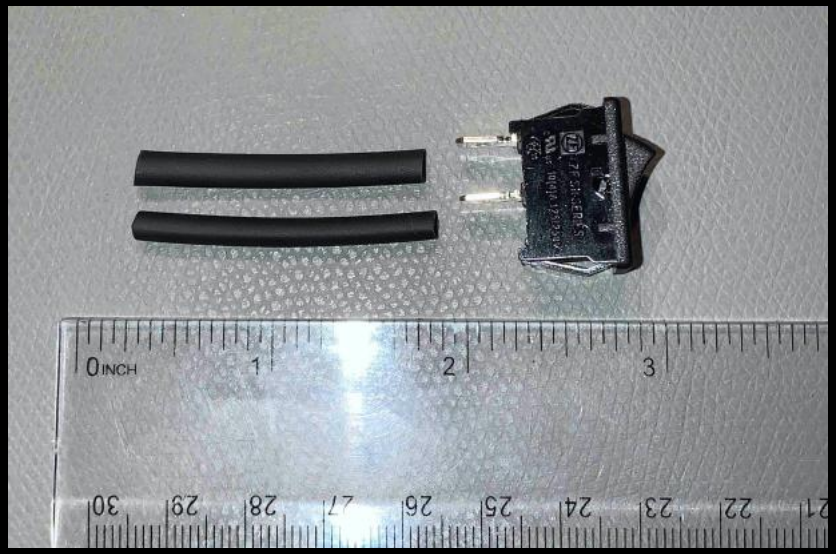
- **Place heatsrink tube around solder joint and heat**

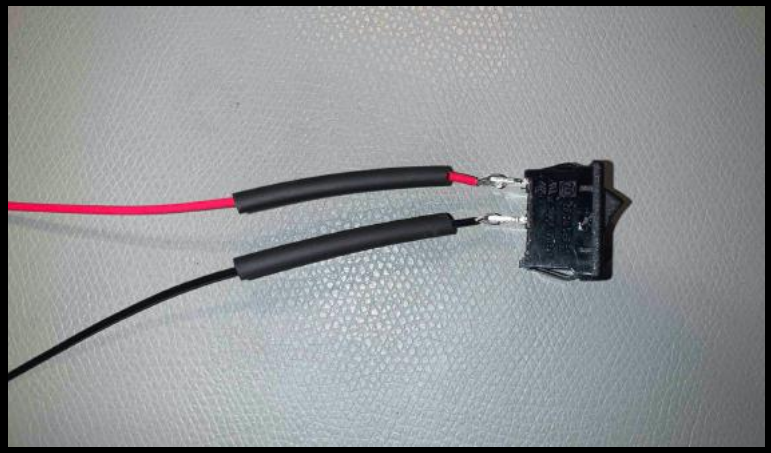


Alternate Power:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- **Place more heatsrink tube on black and red wiring**
  - **Move tubing away from ends of wire**
- **Splice red and black wires onto switch terminals and solder taking care not to shrink tubing**

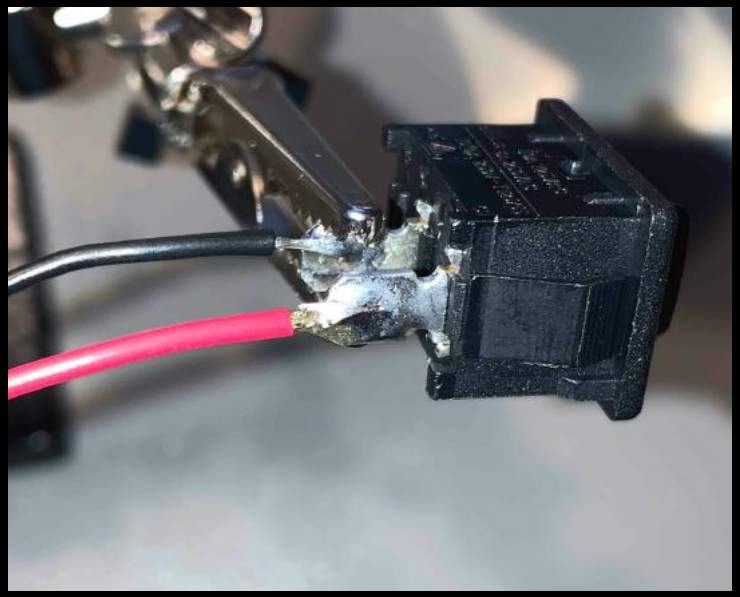


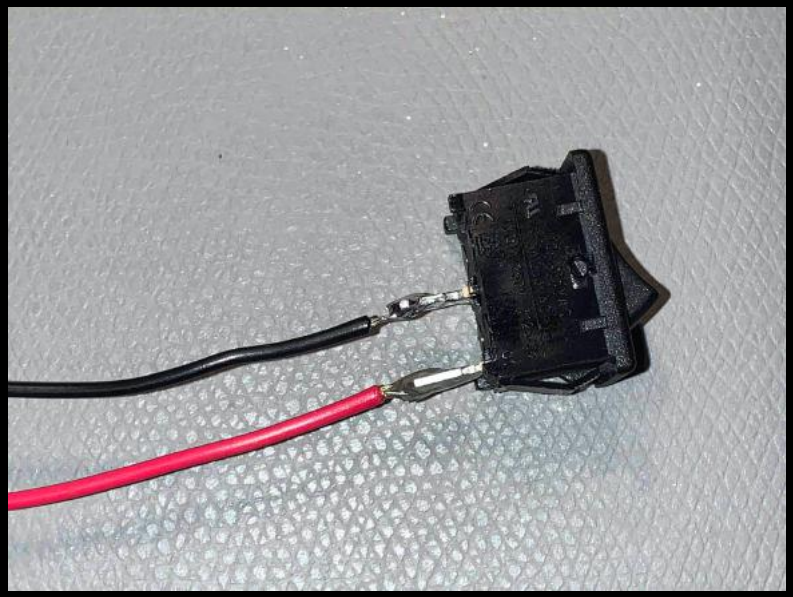


Alternate Power:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- **After soldering…**
- **Take care not to overheat the switch**

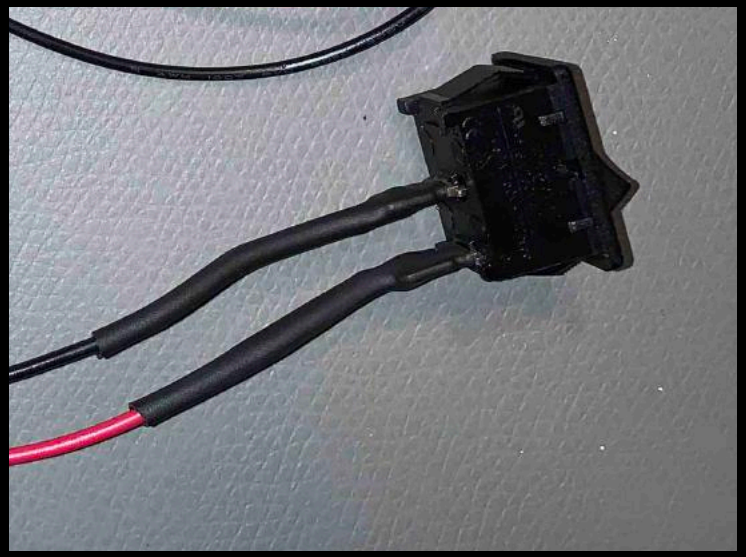


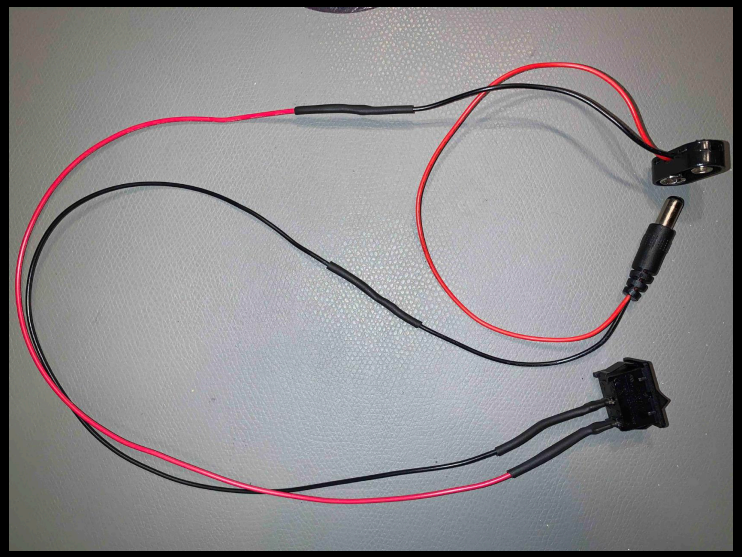


Alternate Power:

Space Minor
UNIVERSITY OF COLORADO BOULDER

- **Move heatshrink tubing over terminals and heat to shrink**
- **Take care not to overheat the switch**

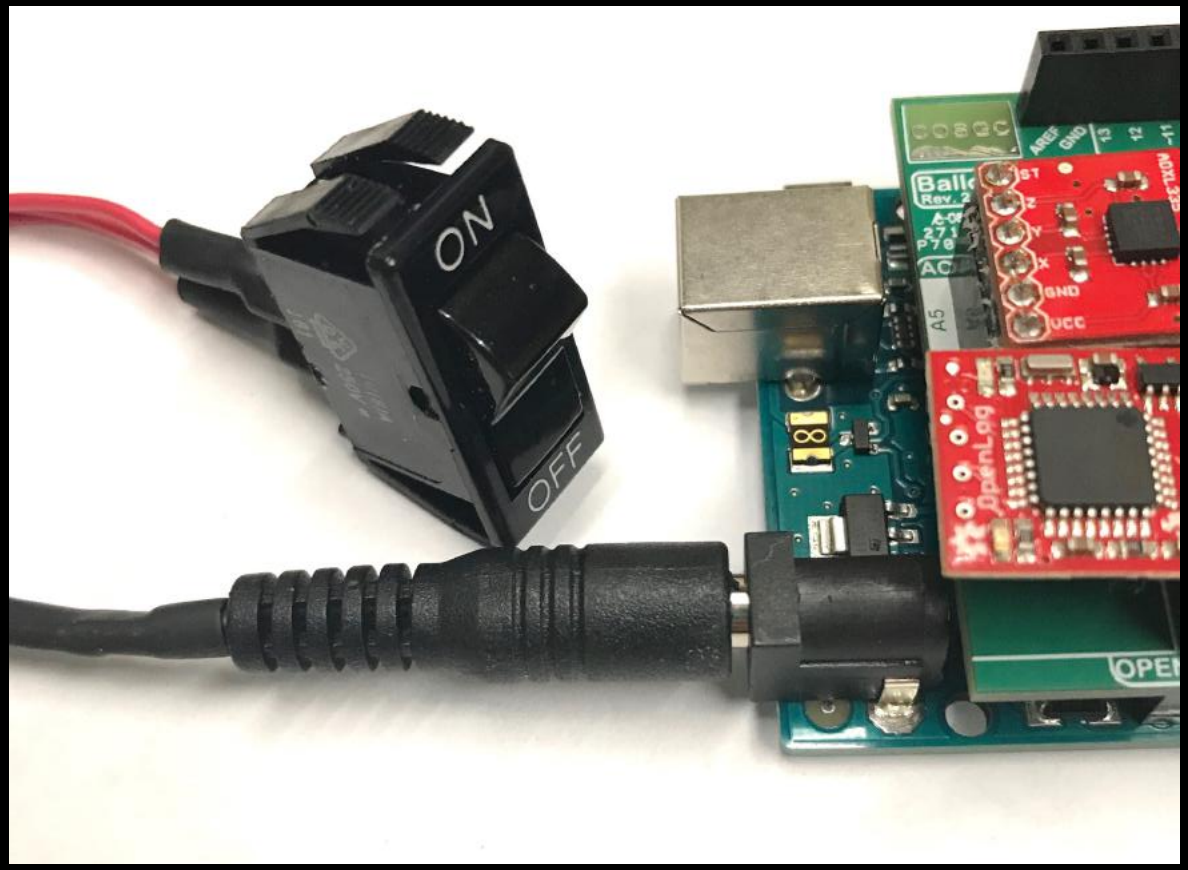




Alternate Power:

Space Minor
UNIVERSITY OF COLORADO BOULDER

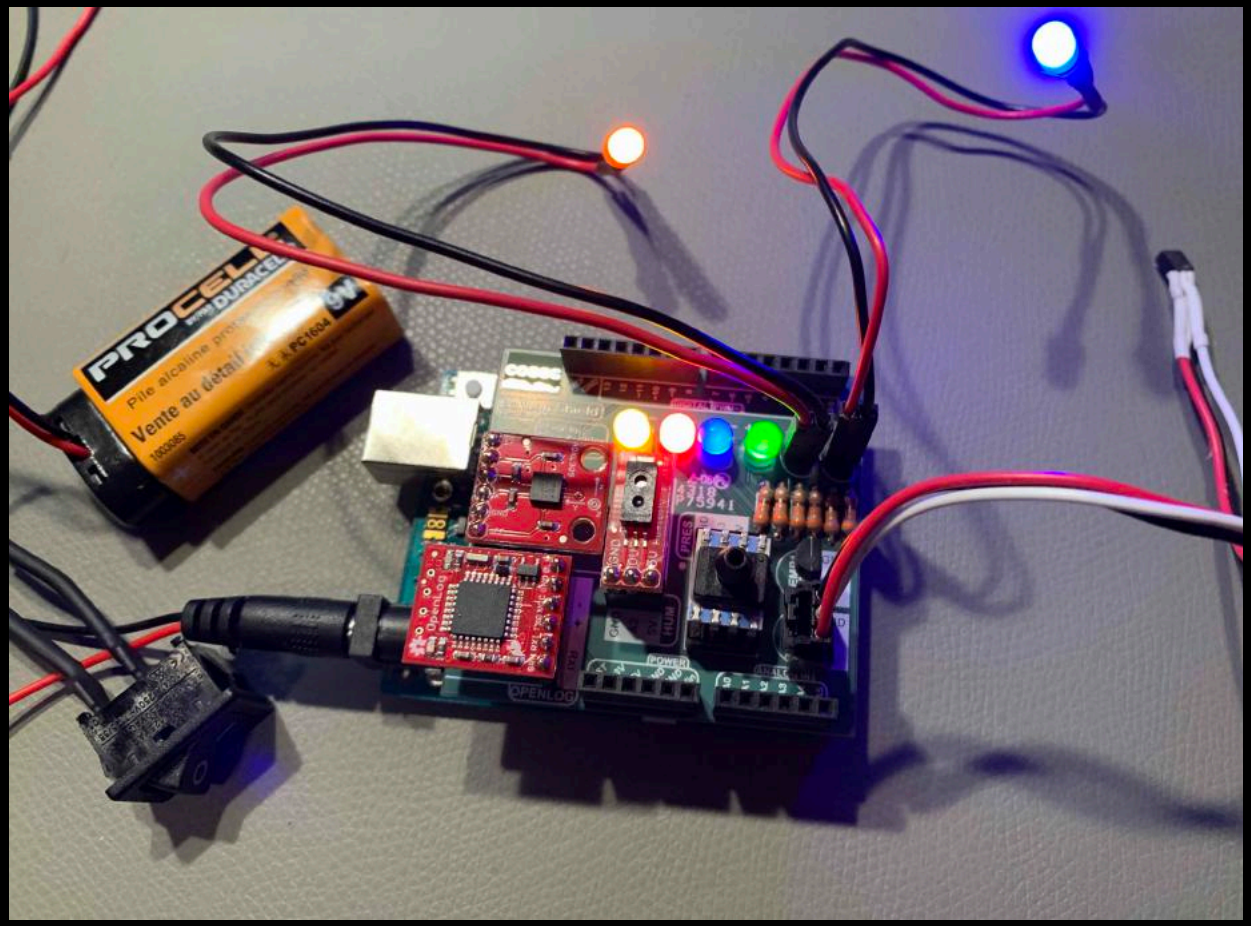
- **Plug battery and switch into Arduino (Remove USB cable)**
- **Flip the switch ON**



Alternate Power:

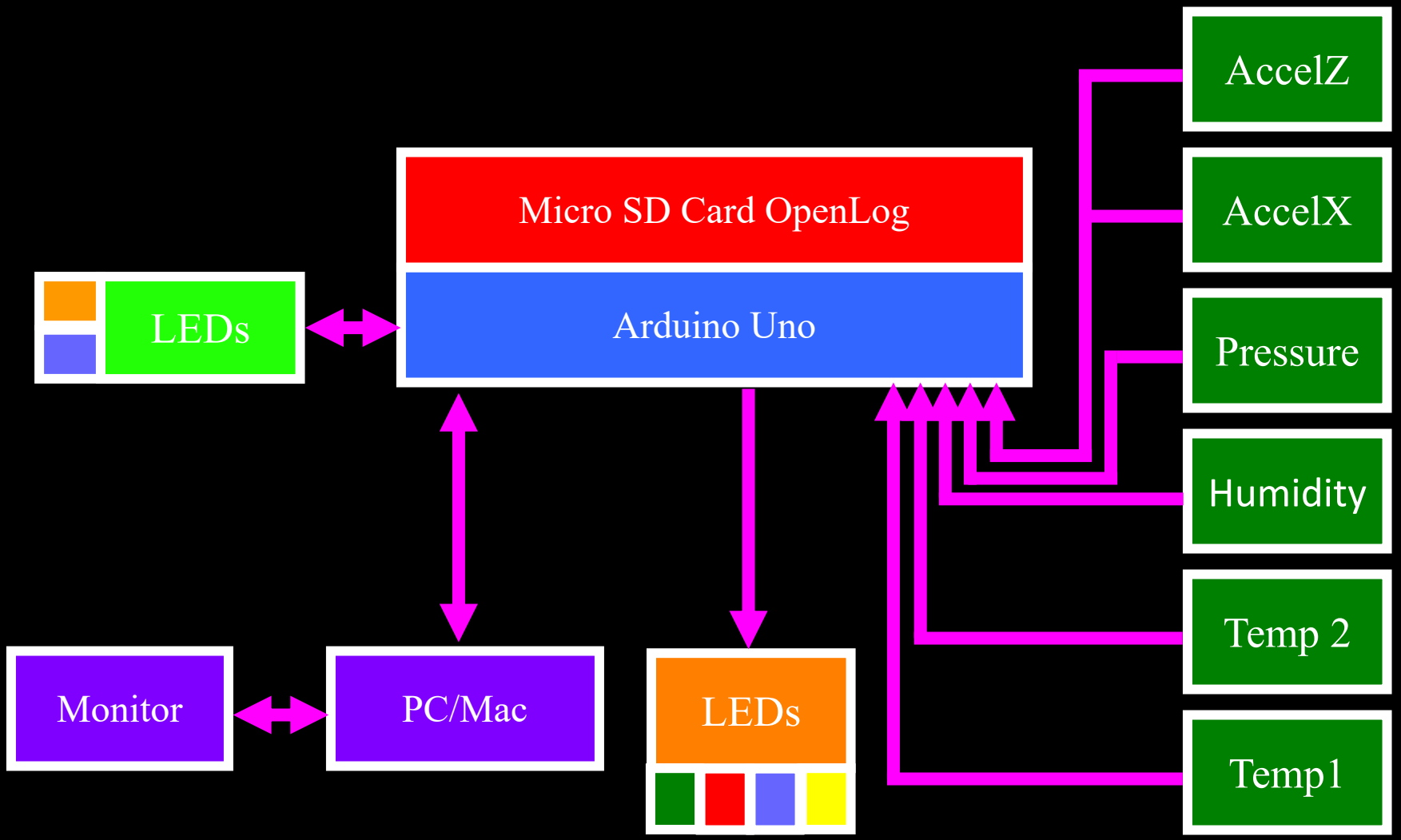
Space Minor
UNIVERSITY OF COLORADO BOULDER

- You are now recording data until power is lost



Part 2 – Arduino Race Track Sensors

A. SHIELD Integration
B. SD Card Code Integration
C. Data Retrieval


AccelZ
AccelX
Pressure
Humidity
Temp 2
Temp1
Micro SD Card OpenLog
Arduino Uno
LEDs
Monitor
PC/Mac
LEDs