HiBall Balloon Payload Workshop

Sensors Part 2
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:

- Arduino Uno
- Monitor
- PC/Mac
- LEDs
- Humidity
- Temp1
Humidity Sensor:

- Humidity sensor (or the Darth Vader Sensor)

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

- First need to solder header to sensor
Humidity Sensor:

- 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:

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:**

*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:

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:

- Modify sketch to read new sensor on A2

```cpp
// 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 begin
  digitalWrite(5, LOW);  //Green LED
  }
```
Humidity Sensor:

- 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:

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

- We know \( V_{\text{out}} \) and \( V_{\text{supply}} \) so using algebra

<table>
<thead>
<tr>
<th>Voltage output (1st order curve fit)</th>
<th>( V_{\text{out}} = (V_{\text{supply}})(0.0062(\text{sensor RH}) + 0.16), ) typical at 25 °C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature compensation</td>
<td>True RH = ( (\text{Sensor RH})/(1.0546 - 0.00216T), ) ( T ) in °C</td>
</tr>
</tbody>
</table>
Humidity Sensor:

- % RH is a linear function of voltage

- 100% RH looks like ~3.7 V
Humidity Sensor:

- Here’s the algebra and the equation to code

\[
V_{OUT} = \left( V_{SUPPLY} \right) \left( 0.0062 \times \text{sensorRH} + 0.16 \right)
\]

\[
\frac{V_{OUT}}{\left( V_{SUPPLY} \right)} \times 0.16 \div \frac{0.0062}{\text{sensorRH}} = \text{sensorRH}
\]

\[
\text{sensorRH} = \frac{V_{OUT}}{\left( 5.0 \ V \right)} \times 0.16 \div \frac{0.0062}{0.0062}
\]
Humidity Sensor:

// 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:

- Verify and upload your code

- Launch serial monitor

- Breathe on humidity sensor like Darth Vader

- Watch LEDs on Shield
Humidity Sensor:

- 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:

- Disconnect your Balloon Shield and add the Humidity Sensor
Balloon Shield Build Part 2:

- Reconnect your Balloon Shield to the Arduino
- Connect USB and reload code
- Verify same results

```
<table>
<thead>
<tr>
<th>id</th>
<th>voltage</th>
<th>units</th>
</tr>
</thead>
<tbody>
<tr>
<td>316</td>
<td>1.54</td>
<td>24.02</td>
</tr>
<tr>
<td>318</td>
<td>1.55</td>
<td>24.33</td>
</tr>
<tr>
<td>315</td>
<td>1.54</td>
<td>23.86</td>
</tr>
<tr>
<td>314</td>
<td>1.53</td>
<td>23.70</td>
</tr>
<tr>
<td>316</td>
<td>1.54</td>
<td>24.02</td>
</tr>
<tr>
<td>313</td>
<td>1.53</td>
<td>23.54</td>
</tr>
<tr>
<td>315</td>
<td>1.54</td>
<td>23.86</td>
</tr>
<tr>
<td>316</td>
<td>1.54</td>
<td>24.02</td>
</tr>
<tr>
<td>317</td>
<td>1.55</td>
<td>24.17</td>
</tr>
<tr>
<td>315</td>
<td>1.54</td>
<td>23.86</td>
</tr>
</tbody>
</table>
```
Part 2 – Arduino Road Trip

Sensors

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

- Arduino Uno
- Pressure
- Humidity
- Temp1
- Monitor
- PC/Mac
- LEDs
Pressure Sensor:

- 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:

- Pressure sensor orientation
Pressure Sensor:

- Pressure sensor orientation

<table>
<thead>
<tr>
<th>Pin 1</th>
<th>Pin 2</th>
<th>Pin 3</th>
<th>Pin 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>NC</td>
<td>V_supply</td>
<td>OUTPUT+</td>
<td>GND</td>
</tr>
</tbody>
</table>
Pressure Sensor:

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

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

<table>
<thead>
<tr>
<th>Pin 1</th>
<th>Pin 2</th>
<th>Pin 3</th>
<th>Pin 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>NC</td>
<td>V_{supply}</td>
<td>OUTPUT+</td>
<td>GND</td>
</tr>
</tbody>
</table>
**Pressure Sensor:**

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

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

- Known:
  \[ V_{\text{supply}} = 5.0 \text{ V} \]
  \[ P_{\text{max}} = 15.0 \text{ psi} \]
  \[ P_{\text{min}} = 0.0 \text{ psi} \]
  Output(V) = measured
  Pressure applied = solve

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

- Here’s the algebra and the equation to code

\[
\text{Output}(V) = \frac{(0.8 \times V_{\text{SUPPLY}})}{(P_{\max} - P_{\min})} \times (\text{pressure}_{\text{applied}} - P_{\min}) + 0.10 \times V_{\text{supply}}
\]

\[
\text{Output}(V) = \frac{(0.8 \times 5.0)}{(15.0 - 0.0)} \times (\text{pressure}_{\text{applied}} - 0.0) + 0.10 \times 5.0
\]

\[
\text{Output}(V) = \frac{(4.0)}{(15.0)} \times (\text{pressure}_{\text{applied}}) + 0.5
\]

\[
\frac{15.0}{4.0} \times (0.5 + \text{Output}(V)) = \text{pressure}_{\text{applied}}
\]
Pressure Sensor:

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

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

sensor = analogRead(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);
```
Pressure Sensor:

- Build and Upload

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

- Use solder sucker

---

<table>
<thead>
<tr>
<th>751</th>
<th>voltage 3.67</th>
<th>units 11.89</th>
</tr>
</thead>
<tbody>
<tr>
<td>751</td>
<td>voltage 3.67</td>
<td>units 11.89</td>
</tr>
<tr>
<td>751</td>
<td>voltage 3.67</td>
<td>units 11.89</td>
</tr>
<tr>
<td>751</td>
<td>voltage 3.67</td>
<td>units 11.89</td>
</tr>
<tr>
<td>751</td>
<td>voltage 3.67</td>
<td>units 11.89</td>
</tr>
<tr>
<td>751</td>
<td>voltage 3.67</td>
<td>units 11.89</td>
</tr>
<tr>
<td>751</td>
<td>voltage 3.67</td>
<td>units 11.89</td>
</tr>
<tr>
<td>751</td>
<td>voltage 3.67</td>
<td>units 11.89</td>
</tr>
</tbody>
</table>

PLEASE SAVE YOUR SKETCH FILE
Pressure Sensor:

- 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:

- Install Pressure Sensor into headers
Pressure Sensor:

- Reconnect your Balloon Shield to the Arduino
- Connect USB and reload code
- Verify same results

```
751   voltage 3.67   units 11.89
751   voltage 3.67   units 11.89
751   voltage 3.67   units 11.89
751   voltage 3.67   units 11.89
751   voltage 3.67   units 11.89
751   voltage 3.67   units 11.89
751   voltage 3.67   units 11.89
751   voltage 3.67   units 11.89
751   voltage 3.67   units 11.89
751   voltage 3.67   units 11.89
```
Part 2 – Arduino Road Trip

Sensors

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

- Arduino Uno
- Monitor
- PC/Mac
- LEDs
- Temp1
- Pressure
- Humidity
- AccelX
- AccelZ
**Accelerometer:**

- 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:

- Solder 6 pin header to board
Accelerometer:

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

- Wire accelerometer as shown

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

- Wire accelerometer as shown

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

- Wire accelerometer as shown

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

- Looking at the data sheet...

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.

<table>
<thead>
<tr>
<th>SENSITIVITY (RATIOMETRIC)$^2$</th>
<th>Each axis</th>
<th>$V_s = 3$ V</th>
<th>$V_s = 3$ V</th>
<th>$V_s = 3$ V</th>
<th>mV/g</th>
<th>%/°C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity at $X_{OUT}$, $Y_{OUT}$, $Z_{OUT}$</td>
<td>270</td>
<td>300</td>
<td>330</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensitivity Change Due to Temperature$^3$</td>
<td>±0.01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ZERO g BIAS LEVEL (RATIOMETRIC)</th>
<th>$V_s = 3$ V</th>
<th>$V_s = 3$ V</th>
<th>$V_s = 3$ V</th>
<th>$V_s = 3$ V</th>
<th>V</th>
<th>V</th>
<th>mg/°C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 g Voltage at $X_{OUT}$, $Y_{OUT}$</td>
<td>1.35</td>
<td>1.5</td>
<td>1.65</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 g Voltage at $Z_{OUT}$</td>
<td>1.2</td>
<td>1.5</td>
<td>1.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 g Offset vs. Temperature</td>
<td>±1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NOISE PERFORMANCE
Accelerometer:

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

- Then 330 mV for every G so...

\[ G_s = \frac{\text{Accelvoltagge} - 1.65 \text{ V}}{0.330 \text{ V}} \]

ADXL335

The ADXL335 output is ratiometric, therefore, the output sensitivity (or scale factor) varies proportionally to the supply voltage. At \( V_s = 3.6 \text{ V} \), the output sensitivity is typically 360 mV/g. At \( V_s = 2 \text{ V} \), 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:

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

```c
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);
}

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

- Upload your 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:

- Upload your 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:

- Disconnect your Balloon Shield and add the Accelerometer
- Reconnect your Balloon Shield to the Arduino

- Connect USB and reload code

- Verify same results

Accelerometer:

<table>
<thead>
<tr>
<th>Xg</th>
<th>Zg</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.13</td>
<td>1.07</td>
</tr>
<tr>
<td>-0.13</td>
<td>1.07</td>
</tr>
<tr>
<td>-0.11</td>
<td>1.07</td>
</tr>
<tr>
<td>-0.13</td>
<td>1.07</td>
</tr>
<tr>
<td>-0.11</td>
<td>1.07</td>
</tr>
<tr>
<td>-0.13</td>
<td>1.07</td>
</tr>
<tr>
<td>-0.13</td>
<td>1.07</td>
</tr>
<tr>
<td>-0.13</td>
<td>1.07</td>
</tr>
<tr>
<td>-0.13</td>
<td>1.07</td>
</tr>
<tr>
<td>-0.13</td>
<td>1.07</td>
</tr>
<tr>
<td>-0.13</td>
<td>1.07</td>
</tr>
<tr>
<td>-0.13</td>
<td>1.07</td>
</tr>
<tr>
<td>-0.13</td>
<td>1.07</td>
</tr>
</tbody>
</table>
Part 2 – Arduino Road Trip

Sensors

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

- Add Orange LED to D4
- Red wire to + and Black wire to -
External Temperature Sensor:

- Add Blue LED to D3
- Red wire to + and Black wire to -
External Temperature Sensor:

- Add Temp2 to Temp2
- Note wire colors
**External Temperature Sensor:**

- Open Temp1 Sketch; save as Temp2

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

    sensor = analogRead(A1);  // Blue LED
    sensorVolt = sensor*(5.0/1023);  // Orange LED
    sensorUnitsC = (sensorVolt - 0.5)/(0.01);  // Green LED
    sensorUnits = (sensorUnitsC*(9.0/5.0) + 32);  // Purple LED
    Serial.print(sensor);  // Red LED
    Serial.print(" \t voltage ");
    Serial.print(sensorVolt);  // White LED
    Serial.print(" \t units ");
    Serial.println(sensorUnits);  // Yellow LED

    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);  // Blue LED
    }
    if(sensorUnits > 79.0) {
        digitalWrite(6, HIGH);  // Orange LED
    }
    if(sensorUnits > 80.0) {
        digitalWrite(7, HIGH);  // Green LED
    }
    if(sensorUnits > 81.0) {
        digitalWrite(9, HIGH);  // Yellow LED
    }
    digitalWrite(3, HIGH);  // Blue LED
    digitalWrite(4, HIGH);  // Orange LED
    delay(100);
```
**External Temperature Sensor:**

- Build and upload your sketch

- Temp2 will stick outside your BalloonSat

- LED 3 and 4, will also stick outside your BalloonSat
External Temperature Sensor:

- 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:
Part 2 – Arduino Road Trip

Sensors

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

```c
// 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;

// Pressure Sensor
int pressure;
float pressureVolt;
float psi;

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

// Accelerometer Z
int accelZ;
float accelZVolt;
float accelZG;
```
// Time keeper
// The time stamp used when recording data points
uint32_t timeStamp = 0;
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");
}
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:

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:

```c
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:

```cpp
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:

- Download code or get from desktop and run and verify it works....
Full Sensor Code Testing:

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Last modified</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Directory</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Balloon_Shield_Test_Code_no_SD.ino</td>
<td>2017-01-04 14:33</td>
<td>3.2K</td>
<td></td>
</tr>
<tr>
<td>Balloon_Shield_Test_Code_no_SD.ino.zip</td>
<td>2017-08-16 16:06</td>
<td>1.3K</td>
<td></td>
</tr>
</tbody>
</table>

If .ino file doesn’t work, try downloading the .zip version
Full Sensor Code Testing:

- Should look like this

```
Time,Temp1F,Temp2F,RH,Pres,AccX,AccZ
499,73.09,144.35,24.96,11.87,0.021,1.117
1003,72.21,120.59,24.96,11.87,0.021,1.102
1508,72.21,110.91,24.49,11.87,0.021,1.117
2012,72.21,114.43,24.96,11.87,0.021,1.117
2515,72.21,117.95,24.65,11.87,0.021,1.117
3019,72.21,111.79,24.65,11.89,0.021,1.117
3523,72.21,109.16,25.12,11.89,0.021,1.117
4027,71.33,116.19,24.80,11.89,0.021,1.102
4532,72.21,117.07,24.96,11.87,0.021,1.117
5036,72.21,110.91,24.80,11.87,0.021,1.117
5539,72.21,110.04,24.65,11.89,0.021,1.117
6043,72.21,117.07,24.96,11.87,0.021,1.117
6547,72.21,117.07,24.96,11.87,0.021,1.102
```
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:

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

- 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:

- Insert MicroSD card as shown
Open Log:

Place OpenLog in correct spot on Balloon Shield
Open Log:

- Reconnect USB and rerun same code
Part 2 – Arduino Race Track Sensors

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

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

“This is your last chance ... After this, there is no turning back. You take the blue pill - the story ends, you wake up in your bed, and believe whatever you want to believe. You take the red pill, ... you stay in Wonderland, and I show you how deep the rabbit-hole goes.”

~ Morpheus’ Warning To Neo (From The Film; “The Matrix”) ~
OpenLog Code:

- 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:

- Rotate your accelerometer like...

4. X Down

5. X Up
Sensor Testing:

8. Z Down

9. Z UP
Data Retrieval:

- 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:

- Remove microSD card from Uno and insert into SD card adapter
Data Retrieval:

- Remove microSD card from Uno and insert into SD card adapter
Data Retrieval:

- Insert SD card adapter into your laptop
Data Retrieval:

- Navigate to card and copy last LOG file to your desktop.
- Open this file with Excel.
Data Retrieval:

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

- Mac Users you must change tab name to remove "."
Data Retrieval:

- Do you see your data markers?
Data Retrieval:

- Re-plot just your accel data
Data Retrieval:

- 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:

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

- Do not connect USB and 9V ever
Alternate Power:
Alternate Power:

- Splice red and black wires onto switch terminals and solder taking care not to shrink tubing

- Take care not to overheat the switch
Alternate Power:

- Place heat shrink tube on black and red wiring at switch terminals
- Heat to shrink tubing, taking care not to overheat the switch
Alternate Power:

- Cut Red and Black wire to ~1 foot in length
- Cut black housing on barrel connector to reveal wires, being careful not to cut wires.
- Cut barrel connector black wire in half
- Strip ends of cut wire back ~1/3 inch
Alternate Power:

- Place heat shrink tubing on one side of wires that will be able to cover connections.
- Splice red and black extensions into connector and solder
Alternate Power:

- Place heat shrink tube around solder joint and heat
Alternate Power:

- Should look similar to this at the end
Alternate Power:

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

- 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