Today:

- Announcements
- Next classes
- Arduino Part 3
Rules for class today:

- All stay through announcements

- Team members not interested in learning more about Arduino after that may leave to work on team project

- 2 members need to remain in class (more preferred)

- Those that leave shall return by end of class to meet up with team and turn in one minute report

- Questions?
Announcements:

- HW #7 Assigned today

- CA Office Hours – Check your emails

- First round of grades next week

- Proposal grading in progress…
  - Should be graded by the end of this week

- Authority to Proceed meetings Thursday and Friday please
Announcements: Scheduling will be done via email – doodle poll

Thursday (11 am – 2 pm)    Friday (8 am – 4 pm)
One Minute Report Questions:

http://www.goatslive.com
Day In Space:

- Juno at Jupiter

NASA / SwRI / MSSS / D. Peach
Day In Space:

Star Wars medal
Tonight...

You are free! We won’t meet this evening.
Thursday...

Requirements and HASP: Miura
Questions?
Part 3 – Arduino Road Trip

A. Humidity Sensor
B. Temperature Sensor
C. Pressure Sensor
D. Accelerometers
E. External Temp Sensor
Part 3 – Arduino Road Trip

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

**Arduino Uno**

- **Monitor**
- **PC/Mac**
- **LEDs**
- **Humidity**
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:

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

  // Turn script running leds OFF at beginning
  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 Vout and Vsupply so using algebra

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

- % RH is linear as function of voltage
- 100% RH looks like \(~3.7\) V
Humidity Sensor:

- Here’s the algebra and the equation to code

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

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

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

```cpp
// 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);
```
**Humidity Sensor:**

- Verify and upload your code
- Launch serial monitor

- Breathe on humidity sensor like Darth Vader

- Watch LEDs on Shield
Humidity Sensor:

- While waiting for the rest of the group, play with your new sensor

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

PLEASE SAVE YOUR SKETCH FILE
Balloon Shield Build Part 2:

- 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

---

```
317  voltage 1.53  units 24.02
316  voltage 1.54  units 24.02
318  voltage 1.55  units 24.33
318  voltage 1.55  units 24.33
315  voltage 1.54  units 23.86
314  voltage 1.53  units 23.70
316  voltage 1.54  units 24.02
313  voltage 1.53  units 23.54
315  voltage 1.54  units 23.86
316  voltage 1.54  units 24.02
317  voltage 1.55  units 24.17
315  voltage 1.54  units 23.86
```
Part 3 – Arduino Road Trip

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

Arduino Uno

Monitor

PC/Mac

LEDs

Humidity

Temp1
Temperature Sensor:

Temperature sensor is the TMP36 - Temperature Sensor

Will use two on balloon flight
- One internal
- One external

Only working with internal now
**Temperature Sensor:**

- Leave Balloon Shield Connected to Arduino

- Use same wiring as humidity sensor except middle wire goes to A0
Temperature Sensor:

- Leave Balloon Shield Connected to Arduino

- Use same wiring as humidity sensor except middle wire goes to A0
Temperature Sensor:

- Leave Balloon Shield Connected to Arduino

- Use same wiring as humidity sensor except middle wire goes to A0
Temperature Sensor:

- Let’s consult the data sheet for the sensor

- 10 mV/C (0.010V/C)

The TMP35 is functionally compatible with the LM35/LM45 and provides a 250 mV output at 25°C. The TMP35 reads temperatures from 10°C to 125°C. The TMP36 is specified from –40°C to +125°C, provides a 750 mV output at 25°C, and operates to 125°C from a single 2.7 V supply. The TMP36 is functionally compatible with the LM50. Both the TMP35 and TMP36 have an output scale factor of 10 mV/°C.
Temperature Sensor:

- Data sheet also says there is an offset

- For TMP36, Offset = 0.5 Volts

Table 4. TMP3x Output Characteristics

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Offset Voltage (V)</th>
<th>Output Voltage Scaling (mV/°C)</th>
<th>Output Voltage @ 25°C (mV)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMP35</td>
<td>0</td>
<td>10</td>
<td>250</td>
</tr>
<tr>
<td>TMP36</td>
<td>0.5</td>
<td>10</td>
<td>750</td>
</tr>
<tr>
<td>TMP37</td>
<td>0</td>
<td>20</td>
<td>500</td>
</tr>
</tbody>
</table>
**Temperature Sensor:**

- So to understand the data, we need to do some math to convert voltage to C

\[
TempC = \frac{(\text{tempVoltage} - 0.5)}{0.01}
\]

Using what we are seeing from our serial monitor, 0.77 Volts, we would get…

\[
TempC = \frac{(0.77 - 0.5)}{0.01} = 27 \, \text{C}
\]

\[
TempF = TempC \times \frac{9}{5} + 32
\]
// Definitions
    int sensor;
    float sensorVolt;
    float sensorUnits;
    float sensorUnitsC;

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

    sensor = analogRead(A0);
    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);
    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);
    }
    delay(100);
**Temperature Sensor:**

- Build and Upload the code and look at serial monitor

- Should see ~0.77 V

- Put your fingers on temp sensor and lightly squeeze

- Look at monitor and LEDs for change

---

*PLEASE SAVE YOUR SKETCH FILE*
Balloon Shield Build Part 3:

- Disconnect your Balloon Shield and add the Temperature Sensor 1

- Note the orientation
Balloon Shield Build Part 3:

- Solder from bottom of board and then trim leads
Balloon Shield Build Part 3:

- Reconnect your Balloon Shield to the Arduino
- Connect USB and reload code
- Verify same results
Part 3 – Arduino Road Trip

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

Arduino Uno

PC/Mac

Monitor

LEDs

Pressure

Humidity

Temp1
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>$V_{sup}$</td>
<td>OUTPUT+</td>
<td>GND</td>
</tr>
</tbody>
</table>
**Pressure Sensor:**

- Connect GND to Pin 4, 5V to Pin 2, and Pin 2 to A4 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 A4 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

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

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

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

\[
\frac{15.0}{4.0} \times (-0.5 + \text{Output}(V)) = \text{pressure}_{\text{applied}}
\]
// Definitions
int sensor;
float sensorVolt;
float sensorUnits;

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

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

- Also use mouth but be careful not to *spit*

Please save your Sketch file
Pressure Sensor:

- While waiting for the rest of the group, play with your new sensor
- Try to get your sensor to zero

PLEASE SAVE YOUR SKETCH FILE
Balloon Shield Build Part 4:

- Disconnect your Balloon Shield and add the Pressure Sensor split header

- Bend leads to hold in place

- Solder from the bottom

- Trim leads
Balloon Shield Build Part 4:

- Install Pressure Sensor into headers
Balloon Shield Build Part 4:

- Reconnect your Balloon Shield to the Arduino
- Connect USB and reload code
- Verify same results
Part 3 – Arduino Road Trip

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

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

- 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...

---

**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.

---

**SENSITIVITY (RAITOMETRIC)$^2$**

<table>
<thead>
<tr>
<th>Each axis</th>
<th>$V_S = 3,\text{V}$</th>
<th>270</th>
<th>300</th>
<th>330</th>
<th>mV/g</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity at $X_{OUT}$, $Y_{OUT}$, $Z_{OUT}$</td>
<td>$V_S = 3,\text{V}$</td>
<td>±0.01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensitivity Change Due to Temperature$^3$</td>
<td>$V_S = 3,\text{V}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ZERO g BIAS LEVEL (RAITOMETRIC)**

| $0\,\text{g Voltage at } X_{OUT}, Y_{OUT}$ | $V_S = 3\,\text{V}$ | 1.35 | 1.5 | 1.65 | V |
| $0\,\text{g Voltage at } Z_{OUT}$ | $V_S = 3\,\text{V}$ | 1.2  | 1.5 | 1.8  | V |
| $0\,\text{g Offset vs. Temperature}$ | $V_S = 3\,\text{V}$ | ±1   |     |     | mg/°C |

---

**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…

\[ G_s = \frac{(\text{Accel voltage} - 1.65 \text{ V})}{0.330 \text{ V}} \]

---

**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, print statements and comment out LED ifs

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

    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:

- 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
Balloon Shield Build Part 5:

- Disconnect your Balloon Shield and add the Accelerometer

- YES, humidity sensor is very close to accel board

- Solder from bottom of board
Balloon Shield Build Part 5:

- Reconnect your Balloon Shield to the Arduino
- Connect USB and reload code
- Verify same results
Part 3 – Arduino Road Trip

A. Humidity Sensor
B. Temperature Sensor
C. Pressure Sensor
D. Accelerometers
E. External Temp Sensor
External Temperature Sensor:
**Balloon Shield Build Part 6:**

- Add Orange LED to D4
- Red wire to + and Black wire to -
Balloon Shield Build Part 6:

- Add Blue LED to D3
- Red wire to + and Black wire to -
Balloon Shield Build Part 6:

- Add Temp2 to Temp2
- Note wire colors
Balloon Shield Build Part 6:

- 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
}
```
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 loop
  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);
Balloon Shield Build Part 6:

- Build and upload your sketch
- Temp2 will stick outside your BalloonSat
- LED 3 and 4, will also stick outside your BalloonSat
Balloon Shield Build Part 6:

- Build and upload your sketch

- Temp2 will stick outside your BalloonSat

- LED 3 and 4, will also stick outside your BalloonSat
Part 3 – Arduino Road Trip

A. Humidity Sensor
B. Temperature Sensor
C. Pressure Sensor
D. Accelerometers
E. External Temp Sensor
Great Job!

My fist....

bump it
Full Sensor Code Testing:

- 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
Full Sensor Code Testing:

// 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;
Full Sensor Code Testing:

// 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");
}
```cpp
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
  timeStmp = millis();
  Serial.print(timeStmp);
}```
Full Sensor Code Testing:

```c
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:

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

celZ = 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....

### Index of /images/TransferStudent/Work

<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/</td>
<td>2017-01-04 14:33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Part_1_4_LED_and_Accels/</td>
<td>2017-01-04 10:46</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Part_1_4_LED_and_POT_no_ifs/</td>
<td>2017-01-04 09:25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Part_1_4_LED_and_Pressure/</td>
<td>2017-01-04 13:24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Part_1_4_LED_and_RH/</td>
<td>2017-01-04 15:23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Part_1_4_LED_and_Temp/</td>
<td>2017-01-04 10:24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Part_1_4_LED_and_Temp_2/</td>
<td>2017-01-04 16:03</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**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 4 – Arduino Race Track

A. SHIELD Integration
B. SD Card Code Integration
C. Data Retrieval
**MicroSD Card Shield:**

- **Arduino Uno**
- **LEDs**
- **PC/Mac**
- **Monitor**
- **Micro SD Card OpenLog**
- **LEDs**
- **Temp 1**
- **Temp 2**
- **Humidity**
- **Pressure**
- **AccelX**
- **AccelZ**
- **OpenLog**

Connections:
- LEDs to Arduino Uno
- Arduino Uno to PC/Mac
- Arduino Uno to LEDs
- Arduino Uno to Temp 1
- Arduino Uno to Temp 2
- Arduino Uno to Pressure
- Arduino Uno to Humidity
- Arduino Uno to AccelX
- Arduino Uno to AccelZ
Part 4 – Arduino Race Track

A. SHIELD Integration
B. SD Card Code Integration
C. Data Retrieval
MicroSD Card Shield:

LEDs

Arduino Uno

Micro SD Card OpenLog

PC/Mac

Monitor

LEDs

AccelZ

AccelX

Pressure

Humidity

Temp 2

Temp 1
Micro SD Card OpenLog:

- Need to test the MicroSD Card OpenLog
**Micro SD Card OpenLog:**

- Insert MicroSD card as shown
Micro SD Card Shield:

- Connect wires to Balloon Shield

> VCC to 3.3 V

> GND to GND

> RXI to TX1
Micro SD Card Shield:

- Connect wires to Balloon Shield

> VCC to 3.3 V

> GND to GND

> RXI to TX1
Micro SD Card Shield:

- Connect wires to Balloon Shield

> VCC to 3.3 V

> GND to GND

> RXI to TX1
Micro SD Card Shield:

- Connect wires to Balloon Shield

> VCC to 3.3 V

> GND to GND

> RXI to TX1
Micro SD Card Shield:

- Connect wires to Balloon Shield

> VCC to 3.3 V

> GND to GND

> RXI to TX1
**Micro SD Card Shield:**

- Reconnect USB and rerun same code
Part 4 – Arduino Race Track

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

- Now let’s explore the code needed to record this data to an SD Card

“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”) ~
SD Card 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 4 – Arduino Race Track

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

- Press the reset button on your microSD card Shield and 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:

- Rotate your accelerometer like…

4. X Down

5. X Up
Data Retrieval:

8. Z Down

9. Z UP
Data Retrieval:

Record end of file markers

Touch your temp sensor for 5 seconds

Suck on your pressure sensor

Breath on the humidity sensor

- Pull USB from Uno
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

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

![CSV File Image]
Data Retrieval:

- Do you see your data markers?
Data Retrieval:

- Re-plot just your accel data
Data Retrieval:

- How can you use this data?
Data Retrieval:

- So that’s the SD card stuff

- Questions?
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:

- Flip the switch ON
Alternate Power:

- You are now recording data until power is lost
Micro SD Card Shield:

- Disconnect USB cable and Balloon Shield
Part 4 – Arduino Race Track

A. SHIELD Integration
B. SD Card Code Integration
C. Data Retrieval
MicroSD Card Shield:

LEDs
Arduino Uno
Micro SD Card OpenLog
PC/Mac
Monitor

LEDs

AccelZ
AccelX
Pressure
Humidity
Temp 2
Temp 1