Today:

- Announcements

- Next classes

- Arduino Part 3
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:
**Humidity Sensor:**

- Modify sketch to read new sensor on A2

```cpp
sensor = analogRead(A2);
sensorVolt = sensor *(5.0/1023);
Serial.print(sensor);
Serial.print("\t Sensor Voltage ");
Serial.println(sensorVolt);
```

```cpp
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);
```
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(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

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

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

- Modify your code like before

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

if(RH > 10) {
  digitalWrite(5, HIGH);
}
if(RH > 25) {
  digitalWrite(6, HIGH);
}
if(RH > 40) {
  digitalWrite(7, HIGH);
}
if(RH > 55) {
  digitalWrite(9, HIGH);
}
delay(100);
```

```c
sensor = analogRead(A2);
sensorVolt = sensor*(5.0/1023);
RH = (((sensorVolt/5.0)-0.16)/0.0062);
Serial.print(sensor);
Serial.print("\\t Sensor Voltage ");
Serial.print(sensorVolt);
Serial.print("\\t RH% ");
Serial.println(RH);
```
Humidity Sensor:

- Verify and upload your code
- Launch serial monitor
- Breathe on humidity sensor like Darth Vader
- Watch LEDs on Shield
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

Homework
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:

- Code Changes... A0
- Comment out RH

```c
sensor = analogRead(A0);
sensorVolt = sensor*(5.0/1023);
// RH = (((sensorVolt/5.0)-0.16)/0.16);
Serial.print(sensor);
```

```c
if(sensorVolt > 0.76) {
    digitalWrite(5, HIGH);
}
if(sensorVolt > 0.77) {
    digitalWrite(6, HIGH);
}
if(sensorVolt > 0.78) {
    digitalWrite(7, HIGH);
}
if(sensorVolt > 0.79) {
    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

- Next, let’s convert volts to Celsius
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…

\[
TempC = \frac{(0.77 - 0.5)}{0.01}
\]

\[
TempC = \frac{(0.27)}{0.01} = 27 \text{ } ^\circ \text{C}
\]

\[
TempF = TempC \times \frac{9}{5} + 32
\]
Temperature Sensor:  - Modify your sketch

```c
sensor = analogRead(A0);
sensorVolt = sensor*(5.0/1023);
// RH = (((sensorVolt/5.0)-0.16)/0.00
TempC = (sensorVolt - 0.5)/(0.01);
TempF = (TempC*(9.0/5.0) + 32);
```

```c
Serial.print(sensor);
Serial.print("\t Sensor Voltage ");
Serial.print(sensorVolt);
Serial.print("\t Temp C ");
Serial.print(TempC);
Serial.print("\t Temp F ");
Serial.println(TempF);
delay(100);
```

```c
if (TempF > 72.0) {
digitalWrite(2, HIGH);
}
if (TempF > 73.0) {
digitalWrite(3, HIGH);
}
if (TempF > 74.0) {
digitalWrite(4, HIGH);
}
if (TempF > 75.0) {
digitalWrite(5, HIGH);
}
```
**Temperature Sensor:**

- Build and Upload

- Test by touching your temp sensor

Save this sketch as TEMP 2 as we will use this later for Temp 2

PLEASE SAVE YOUR SKETCH FILE
Balloon Shield Build Part 3:

- Disconnect your Balloon Shield and add the Temperature Sensor
- 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

Monitor

PC/Mac

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>$V_{\text{supply}}$</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 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 2 to A4 on the Arduino
Pressure Sensor:

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

- Known:
  V_supply = 5.0 V
  P_{max} = 15.0 psi
  P_{min} = 0.0 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

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

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

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

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

- Modify your sketch as before

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

sensor = analogRead(A3);
sensorVolt = pressure(3.0/1023);
psi = (pressureVolt-0.5)*(15.0/4.0);
Serial.print(sensor);
Serial.print("\\t Sensor Voltage ");
Serial.print(sensorVolt);
Serial.print("\\t psi ");
Serial.println(psi);

if(psi < 12.20) {
    digitalWrite(5, HIGH);
}
if(psi < 10.10) {
    digitalWrite(6, HIGH);
}
if(psi < 8.10) {
    digitalWrite(7, HIGH);
}
if(psi < 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
- Disconnect your Balloon Shield and add the Pressure Sensor

- Bend leads to hold in place

- Solder from the bottom

- Trim leads
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:
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 $V_S = 3.6$ 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>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>3 V</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensitivity Change Due to Temperature³</td>
<td>3 V</td>
<td>±0.01</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

\[
G_s = \frac{\text{Accelvoltage} - 1.65 \text{V}}{0.330 \text{V}}
\]
Accelerometer:

- Code Changes… A4, A5
  change print statements and
  comment out LED ifs

```c
// Definitions
int sensorX;
int sensorZ;
float sensorXVolt;
float sensorZVolt;
float Xg;
float Zg;

sensorX = analogRead(A4);
sensorZ = analogRead(A5);
sensorXVolt = sensorX*(5.0/1023);
sensorZVolt = sensorZ*(5.0/1023);
Xg = (sensorXVolt - (3.3/2))/(0.330);
Zg = (sensorZVolt - (3.3/2))/(0.330);
Serial.print("Xg ");
Serial.print(Xg);
Serial.print(" \t Zg ");
Serial.println(Zg);
```
**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
- Next, let’s convert volts to Gs
Accelerometer:

- 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
Balloon Shield Build Part 5:

- Disconnect you 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:

- Arduino Uno
- Monitor
- PC/Mac
- LEDs
- Temp 1
- Temp 2
- Pressure
- Humidity
- AccelX
- AccelZ

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

Additionally, the diagram shows an external temperature sensor with the following details:

- Temperature 1
- Temperature 2
- Acceleration in the Z-axis (AccelZ)
- Acceleration in the X-axis (AccelX)
- Pressure
- Humidity
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 **Temp2** Sketch; save as Temp2

```c
// Definitions
int sensor;
float sensorVolt;
float RH;
float TempC2;
float TempF2;
```

```c
pinMode(3, OUTPUT); //Blue LED
pinMode(4, OUTPUT); //Orange
pinMode(5, OUTPUT); //Green LED
pinMode(6, OUTPUT); //Purple
pinMode(7, OUTPUT); //Red LED
pinMode(9, OUTPUT); //Yellow
```

```c
if (tempF2 > 78.0) {
digitalWrite(3, HIGH);
}
if (tempF2 > 79.0) {
digitalWrite(4, HIGH);
}
if (tempF2 > 80.0) {
digitalWrite(5, HIGH);
}
if (tempF2 > 81.0) {
digitalWrite(6, HIGH);
}
delay(100);
```

```c
sensor = analogRead(A1);
sensorVolt = sensor*(5.0/1023);
tempC2 = (sensorVolt - 0.5)/(0.01);
tempF2 = (tempC2*[9.0/5.0] + 32);
Serial.print(sensor);
Serial.print(" Sensor Voltage ");
Serial.print(sensorVolt);
Serial.print(" TempC2 ");
Serial.println((tempC2);
Serial.print(" TempF2 ");
Serial.println((tempF2);
```
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
- 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