Today:

- Extra credit for being here tonight
- Goal to be done by 8 PM
- Need to get through SD card write and Excel
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)</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>$V_s = 3$ V</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensitivity Change Due to Temperature</td>
<td>$V_s = 3$ V</td>
<td></td>
<td></td>
<td>±0.01</td>
<td>%/ºC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ZERO $g$ BIAS LEVEL (RATIOMETRIC)</th>
<th></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>$V_s = 3$ V</td>
<td></td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>0 $g$ Voltage at $Z_{OUT}$</td>
<td>$V_s = 3$ 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></td>
<td>1.2</td>
<td>1.5</td>
<td>1.8</td>
<td>V</td>
</tr>
</tbody>
</table>

| NOISE PERFORMANCE | |
|-------------------| | | | | |
| 1.5 | | | | | |
| 1.65 | | | | | | |
| 1.8 | | | | | | | |
| 1.85 | | | | | | | | |
| 1.9 | | | | | | | | |
**Accelerometer:**

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

- Then 330 mV for every G so…

\[ G_s = (\text{Accelvoltage} - 1.65 \text{ V}) / (0.330 \text{ V}) \]
Accelerometer:

- Code Changes...

```cpp
// Definitions
int accelX;
float accelXVolt;
float accelXG;
int accelZ;
float accelZVolt;
float accelZG;

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

    accelX = analogRead(A4);
    accelXVolt = accelX*(5.0/1023);
    accelXG = (accelXVolt - (3.3/2))/(0.330);
    accelZ = analogRead(A5);
    accelZVolt = accelZ*(5.0/1023);
    accelZG = (accelZVolt - (3.3/2))/(0.330);
    Serial.print("accelXG ");
    Serial.print(accelXG);
    Serial.print("\t accelZG ");
    Serial.println(accelZG);
}```
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 some additional items to your Balloon Shield
- Two 2 pin headers (LEDs 4 and 3)
- One 3 pin header (Temp2 External)
- Two resistors
Balloon Shield Build Part 6:

- Add Resistor first (same procedure as earlier)
Balloon Shield Build Part 6:

- Add the 2 Pin headers
- Use pliers or team member to hold while soldering
- Short end into the board
- Solder from bottom
Balloon Shield Build Part 6:

- Add the 3 Pin header
- Use pliers or team member to hold while soldering
- Short end into the board
- Solder from bottom
Balloon Shield Build Part 6:

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

```c
// Definitions
int temp2;
float temp2Volt;
float temp2C;
float temp2F;

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

  temp2 = analogRead(A1);
  temp2Volt = temp2*(5.0/1023);
  temp2C = (temp2Volt - 0.5)/(0.01);
  temp2F = (temp2C*(9.0/5.0) + 32);
  Serial.print(temp2);
  Serial.print("\t Temp2Volt ");
  Serial.print(temp2Volt);
  Serial.print("\t Temp2C ");
  Serial.print(temp2C);
  Serial.print("\t Temp2F ");
  Serial.println(temp2F);
}
```
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:

```c
void setup() {
   // put your setup code here, to run once:
   Serial.begin(9600);

   // setup the LED Visual Display
   pinMode(3, OUTPUT); //SD Card writing
   pinMode(4, OUTPUT); //Arduino on
   pinMode(5, OUTPUT); //Sensors/sketch running
   pinMode(6, OUTPUT); //Sensors/sketch running
   pinMode(7, OUTPUT); //Sensors/sketch running
   pinMode(9, OUTPUT); //Sensors/sketch running
}
```
Full Sensor Code Testing:

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

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

  delay(500);

  // Turn script running leds ON at begining of loop
  digitalWrite(5, HIGH);
  digitalWrite(6, HIGH);
  digitalWrite(7, HIGH);
  digitalWrite(9, HIGH);
}
```
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("Temp1 ");
Serial.print(temp1F, 2);

temp2 = analogRead(A1);
temp2Volt = temp2*(5.0/1023);
temp2C = (temp2Volt - 0.5)/(0.01);
temp2F = (temp2C*(9.0/5.0) + 32);
Serial.print("\t Temp2 ");
Serial.print(temp2F, 2);
Full Sensor Code Testing:

humidity = analogRead(A2);
humidityVolt = humidity*(5.0/1023);
RH = (((humidityVolt/5.0)-0.16)/0.0062);
Serial.print("\t RH ");
Serial.print(RH, 2);

pressure = analogRead(A3);
presureVolt = pressure*(5.0/1023);
psi = (pressureVolt-0.5)*(15.0/4.0);
Serial.print("\t PSI ");
Serial.print(psi, 2);
Full Sensor Code Testing:

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

accelZ = analogRead(A5);
accelZVolt = accelZ*(5.0/1023);
accelZG = (accelZVolt - (3.3/2))/(0.330);
Serial.print("\t ZG ");
Serial.print(accelZG,3);

Serial.println();
```
Full Sensor Code Testing:

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

- Should look like this
Part 4 – Arduino Race Track

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

- LEDs
- Arduino Uno
- Monitor
- PC/Mac
- LEDs
- Temp1
- Temp 2
- Pressure
- Humidity
- AccelX
- AccelZ
Part 4 – Arduino Race Track

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

- Arduino Uno
- LEDs
- MicroSD Card Shield
- Monitor
- PC/Mac
- LEDs
- AccelZ
- AccelX
- Pressure
- Humidity
- Temp 2
- Temp 1
Micro SD Card Shield:

- First need to build MicroSD Card Shield
**Micro SD Card Shield:**

- Insert headers as shown (outside holes)
Micro SD Card Shield:

- Solder from bottom and keep pins/headers perpendicular to the board
Micro SD Card Shield:

- Insert microSD card
Micro SD Card Shield:

- Insert microSD card
**Micro SD Card Shield:**

- Align pins to Arduino headers and connect
Micro SD Card Shield:

- Connect Balloon Shield to stack
**Micro SD Card Shield:**

- Reconnect USB and rerun same code

```plaintext
<table>
<thead>
<tr>
<th>Temp1</th>
<th>Temp2</th>
<th>RH</th>
<th>PSI</th>
<th>XG</th>
<th>ZG</th>
</tr>
</thead>
<tbody>
<tr>
<td>73.09</td>
<td>75.72</td>
<td>13.14</td>
<td>12.15</td>
<td>-0.009</td>
<td>1.191</td>
</tr>
<tr>
<td>73.09</td>
<td>76.60</td>
<td>13.61</td>
<td>12.15</td>
<td>0.006</td>
<td>1.191</td>
</tr>
<tr>
<td>72.21</td>
<td>76.60</td>
<td>13.14</td>
<td>12.15</td>
<td>0.006</td>
<td>1.176</td>
</tr>
<tr>
<td>73.09</td>
<td>76.60</td>
<td>13.77</td>
<td>12.15</td>
<td>-0.009</td>
<td>1.176</td>
</tr>
<tr>
<td>73.09</td>
<td>76.60</td>
<td>13.77</td>
<td>12.15</td>
<td>0.006</td>
<td>1.176</td>
</tr>
<tr>
<td>72.21</td>
<td>76.60</td>
<td>13.29</td>
<td>12.15</td>
<td>0.006</td>
<td>1.176</td>
</tr>
<tr>
<td>72.21</td>
<td>75.72</td>
<td>13.29</td>
<td>12.15</td>
<td>-0.009</td>
<td>1.176</td>
</tr>
<tr>
<td>73.09</td>
<td>76.60</td>
<td>13.45</td>
<td>12.15</td>
<td>-0.009</td>
<td>1.176</td>
</tr>
<tr>
<td>73.96</td>
<td>75.72</td>
<td>13.29</td>
<td>12.16</td>
<td>0.006</td>
<td>1.191</td>
</tr>
<tr>
<td>72.21</td>
<td>76.60</td>
<td>13.29</td>
<td>12.16</td>
<td>-0.009</td>
<td>1.176</td>
</tr>
<tr>
<td>72.21</td>
<td>76.60</td>
<td>13.45</td>
<td>12.15</td>
<td>0.006</td>
<td>1.191</td>
</tr>
<tr>
<td>73.09</td>
<td>76.60</td>
<td>13.45</td>
<td>12.15</td>
<td>-0.009</td>
<td>1.176</td>
</tr>
<tr>
<td>73.96</td>
<td>76.60</td>
<td>13.61</td>
<td>12.15</td>
<td>-0.009</td>
<td>1.176</td>
</tr>
<tr>
<td>72.09</td>
<td>76.60</td>
<td>13.61</td>
<td>12.15</td>
<td>-0.009</td>
<td>1.176</td>
</tr>
<tr>
<td>73.09</td>
<td>76.60</td>
<td>13.77</td>
<td>12.15</td>
<td>0.006</td>
<td>1.176</td>
</tr>
<tr>
<td>73.09</td>
<td>76.60</td>
<td>13.77</td>
<td>12.15</td>
<td>-0.009</td>
<td>1.176</td>
</tr>
<tr>
<td>72.21</td>
<td>76.60</td>
<td>13.29</td>
<td>12.15</td>
<td>-0.009</td>
<td>1.176</td>
</tr>
</tbody>
</table>
```

Part 4 – Arduino Race Track

A. SHIELD Integration
B. SD Card Code Integration
C. Data Retrieval
Now let’s explore the code needed to record this data to an SD Card.
**SD Card Code:**

- SD card writing uses libraries built into the Arduino software

- Code gets a little more complicated at this point but relatively easy compared to other languages that write to SD cards
SD Card Code:

- New Code to Add...

```c
#include <SD.h>

// Controls for the data logging system
const int LOG_INTERVAL = 1000;  // milli seconds between entries

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

const int chipSelect = 8;      // This is set to 8 for the SparkFun uSDCard

// Variable for file name
char logFileName[16];

// Use this LED for your writing indicator
int ledState = 0;
```
SD Card Code:

- Define character variables for data storage technique

```c
// Definitions
// Temperature Sensor #1
int temp1;
float temp1Volt;
float temp1C;
float temp1F;
char temp1FString[7];

// Temperature Sensor #2
int temp2;
float temp2Volt;
float temp2C;
float temp2F;
char temp2FString[7];

// Humidity Sensor
int humidity;
float humidityVolt;
float RH;
char RHString[7];
```
**SD Card Code:**

- Define character variables for data storage technique

```c
// Pressure Sensor
int pressure;
float pressureVolt;
float psi;
char psiString[7];

// Accelerometer X
int accelX;
float accelXVolt;
float accelXG;
char accelXGString[7];

// Accelerometer Z
int accelZ;
float accelZVolt;
float accelZG;
char accelZGString[7];
```
SD Card Code:

- Define a string variable to store character data storage

```java
// This variable will hold the data
String dataString = "";
```
- Define a string variable to contain data headers

String sensorNames = "Time stamp (ms), Temp1 (F), Temp2 (F), RH (%) Pres (psi), x5 (g), z5 (g)";
**SD Card Code:**

- New Void Setup code…

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

    Serial.begin(9600);

    // setup the LED Visual Display
    pinMode(3, OUTPUT); // SD Card
    pinMode(4, OUTPUT); // Arduino
digitalWrite(4, HIGH); // Leave

    pinMode(5, OUTPUT); // Sensors
    pinMode(6, OUTPUT); // Sensors
    pinMode(7, OUTPUT); // Sensors
    pinMode(9, OUTPUT); // Sensors

    // turn on Arduino LED
    digitalWrite(4, HIGH); // Leave

    // SD Card setup
    // make sure that the default chip select
    pinMode(10, OUTPUT); // To make
    pinMode(chipSelect, OUTPUT);

    // This function will set up the SD card
    // Also the header of the log file will
    SDCardInit();
}
```
SD Card Code:

- New Void Loop
stuff…

```c
void loop() {
  // put your main code here, to run...  
  // Turn script running leds OFF at
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  digitalWrite(9, LOW);

  // Delay your time interval
  delay(LOG_INTERVAL);

  // Turn script running leds ON at
  digitalWrite(5, HIGH);
  digitalWrite(6, HIGH);
  digitalWrite(7, HIGH);
  digitalWrite(9, HIGH);
}
```
SD Card Code:

- Record the time and print it
- Convert sensor value to character and store in string

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

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);
dtostrf(temp1F, 6, 2, temp1FString); // Convert
dataString = dataString +"," + temp1FString;
```
SD Card Code:

- Think of writing a sentence…

- Sentence = dataString

- Think of the words of the sentence are your sensors values converted from numbers to characters to strings

- Word = temp1FString (for example)
**SD Card Code:**

- Full Example…

> temp1F = 82.1

Convert to string - `dtostrf` does this
> temp1fString = “--82.1-”

Add to the sentence
> dataString = dataString + “,” + temp1FString;
SD Card Code:

- Convert sensor value to character and store in string

```c
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);
Serial.print(temp2F, 2);
 dtostrf(temp2F, 6, 2, temp2FString); // 0.001251
 dataString = dataString +""," + temp2FString;

humidity = analogRead(A2);
humidityVolt = humidity*(5.0/1023);
RH = (((humidityVolt/5.0)-0.16)/0.0062);
Serial.print(',');
Serial.print(RH, 2);
dtostrf(RH, 6, 2, RHString);
dataString = dataString +""," + RHString;
```
**SD Card Code:**

- Convert sensor value to character and store in string

- Include a `Serial.println()` at the end for a new line

```cpp
pressure = analogRead(A3);
pressureVolt = pressure*(5.0/1023);
psi = (pressureVolt-0.5)*(15.0/4.0);
Serial.print(' ,');
Serial.print(psi, 2);
dtostrf(psi, 6, 2, psiString); //
dataString = dataString +" ," + psiString; //

accelX = analogRead(A4);
accelXVolt = accelX*(5.0/1023);
accelXG = (accelXVolt - (3.3/2))/(0.330);
Serial.print(' ,');
Serial.print(accelXG,3);
dtostrf(accelXG, 6, 2, accelXGString);
dataString = dataString +" ," + accelXGString;

accelZ = analogRead(A5);
accelZVolt = accelZ*(5.0/1023);
accelZG = (accelZVolt - (3.3/2))/(0.330);
Serial.print(' ,');
Serial.print(accelZG,3);
dtostrf(accelZG, 6, 2, accelZGString);
dataString = dataString +" ," + accelZGString;
Serial.println();
```
SD Card Code:

- Call the write to sd magic
- If successful, blink led
- Loop

```c
// Write to the SD card and if a write to SD is successful then
// The function writeDataToSD returns TRUE if a write was successful
if (writeDataToSD())
{
  // Change the State of the LED from OFF->ON, or ON->OFF
  ledState++;
  // Handle the case where the LED state is now 2 and set it back
  if (ledState > 1) {
    ledState = 0;
  }
}
// Update the activity LED to show a successful write cycle
digitalWrite(3, ledState);  // Change the LED state
```
**SD Card Code:**

- If you were to upload the code, would it work?
- No because functions are not defined yet…

```cpp
pinMode(10, OUTPUT); //
pinMode(chipSelect, OUTPUT)

// This function will set
// Also the header of the
SDCardInit();

// Write to the SD card and if
// The function writeDataToSD is
if(writeDataToSD())
{
    // Change the State of the LE
```
SD Card Code:

- If you were to upload the code, would it work?
**SD Card Code:**

- Here is the code for SDCardInit();

```java
// This function is called when the system starts or after a power reset to:
// initialize the SD Card if it is present.

boolean SDCardInit()
{
    Serial.println("Initializing SD card...");

    // see if the card is present and can be initialized:
    if (!SD.begin(chipSelect))
    {
        Serial.println("Card failed, or not present");
        delay(100);
        // Can't do anything more. Try another time....
        return 0;
    }

    // The Card is present so find an unused file name

    Serial.println("searching for an unused file name...");
    // Start with LOG0.CSV and count up until an unused file name is found.
```
**SD Card Code:**

- Here is the code for `SDCardInit();`

- But wait there is more!

- We still have the other function

```cpp
for (long i=0; i < 1000; i++)
{
    sprintf(logFileName, "LOG%d.CSV", i);
    if (!SD.exists(logFileName))
    {
        break;
    }
}

Serial.println("The Log filename is: ");
Serial.println(logFileName);

File dataFile = SD.open(logFileName, FILE_WRITE);

// Write the header including sensor names to the newly opened file
dataString = sensorNames;

if (dataFile)
{
    dataFile.println(dataString);
dataFile.close();
    Serial.println("SD Card initialized and data written.");
    return(1); // Able to write to SD card
}
else
{
    Serial.println("SD card present but unable to write to file");
    sprintf(logFileName, "LOG0.CSV"); // Clear out the file name
    return(0); // Unable to write to SD card
}
```
**SD Card Code:**

- Here is the code for `writeDataToSD`;

- So where does all this code go?

- TABS...

```java
boolean writeDataToSD ()
{
    File dataFile = SD.open(logFileName, FILE_WRITE);

    // if the file is available, write to it:
    if (dataFile)
    {
        dataFile.println(dataString);
        dataFile.close();
        return(1);
    }

    // if the file isn't open, notify that there was an error
    else {
        Serial.println();
        Serial.print("error writing to file: ");
        Serial.println(logFileName);
        dataFile.close();
        delay(100);
        Serial.println("Re - Initializing SD card...");
        return(SDCardInit());
    }
}
```
**SD Card Code:**

- Creating tabs is a great to organize your code
**SD Card Code:**

- Creating one and press OK

- Make sure to type it exactly as the function
SD Card Code:

- Brand new real estate to type code!
**SD Card Code:**

- Create another one
**SD Card Code:**

- Every tab you create, creates a new *.ino file in the folder for your main tab or sketch you were working
- Must keep all *.ino files in the folder for the code to work properly
- Can only open the main *.ino file directly
**SD Card Code:**

- Now start typing
**SD Card Code:**

- No. Just kidding. Go to the website and download **SD Code Complete** (zip file)
SD Card Code:

- Unzip and use directly or open tabs, copy and paste content from one tab to your tab

- Build and check for errors

- Upload and watch LED blink

- Raise hand for help
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
- Record some easy to recognize data
- Breath on your humidity sensor twice
- Use solder sucker to give pressure sensor data
- Touch the temp sensor
- Orient your accelerometer
- Wait for 2 minutes
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 LOGGER file to your desktop

- Open this file with Excel
Data Retrieval:

- Graph all data minus the time stamp
Data Retrieval:

- Graph all data minus the time stamp
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