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>$V_{supply}$</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:

- 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:
  
  \[ V_{\text{supply}} = 5.0 \text{ V} \]
  
  \[ P_{\text{max}} = 15.0 \text{ psi} \]
  
  \[ P_{\text{min}} = 0.0 \text{ psi} \]
  
  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

\[
Output(V) = \frac{0.8 \times V_{SUPPLY}}{(P_{max} - P_{min})} \times (pressure_{applied} - P_{min}) + 0.10 \times V_{suply}
\]

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

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

\[
\frac{15.0}{4.0} \times (-0.5 + Output(V)) = pressure_{applied}
\]
**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
Balloon Shield Build Part 4:

- 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
Today:

- Extra credit for being here tonight
- Goal to be done by 9 PM
- Need to get through SD card write and Excel
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 beginning of loop
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  digitalWrite(9, LOW);

  delay(500);

  // Turn script running leds ON at beginning of loop
  digitalWrite(3, HIGH);
  digitalWrite(4, HIGH);
  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);
pressureVolt = pressure*(5.0/1023);
psi = (pressureVolt-0.5)*(15.0/4.0);
Serial.print("\t PSI ");
Serial.print(psi, 2);
Full Sensor Code Testing:

```cpp
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 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

Micro SD Card Shield

Monitor

PC/Mac

LEDs

AccelZ

AccelX

Pressure

Humidity

Temp 2

Temp 1
Part 4 – Arduino Race Track

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

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

<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>71.33</td>
<td>72.21</td>
<td>44.98</td>
<td>12.07</td>
<td>0.036</td>
<td>1.132</td>
</tr>
<tr>
<td>71.33</td>
<td>72.21</td>
<td>45.14</td>
<td>12.07</td>
<td>0.036</td>
<td>1.132</td>
</tr>
<tr>
<td>71.33</td>
<td>72.21</td>
<td>45.14</td>
<td>12.07</td>
<td>0.036</td>
<td>1.132</td>
</tr>
<tr>
<td>70.45</td>
<td>72.21</td>
<td>44.51</td>
<td>12.07</td>
<td>0.021</td>
<td>1.132</td>
</tr>
<tr>
<td>71.33</td>
<td>72.21</td>
<td>44.67</td>
<td>12.07</td>
<td>0.036</td>
<td>1.132</td>
</tr>
<tr>
<td>71.33</td>
<td>72.21</td>
<td>44.51</td>
<td>12.07</td>
<td>0.036</td>
<td>1.132</td>
</tr>
<tr>
<td>71.33</td>
<td>72.21</td>
<td>44.67</td>
<td>12.07</td>
<td>0.036</td>
<td>1.132</td>
</tr>
<tr>
<td>69.57</td>
<td>71.33</td>
<td>44.35</td>
<td>12.07</td>
<td>0.036</td>
<td>1.132</td>
</tr>
<tr>
<td>70.45</td>
<td>72.21</td>
<td>44.83</td>
<td>12.07</td>
<td>0.021</td>
<td>1.132</td>
</tr>
<tr>
<td>70.45</td>
<td>71.33</td>
<td>44.83</td>
<td>12.07</td>
<td>0.036</td>
<td>1.132</td>
</tr>
<tr>
<td>71.33</td>
<td>72.21</td>
<td>44.51</td>
<td>12.07</td>
<td>0.036</td>
<td>1.132</td>
</tr>
<tr>
<td>70.45</td>
<td>72.21</td>
<td>44.83</td>
<td>12.07</td>
<td>0.036</td>
<td>1.132</td>
</tr>
<tr>
<td>71.33</td>
<td>71.33</td>
<td>44.67</td>
<td>12.07</td>
<td>0.036</td>
<td>1.132</td>
</tr>
<tr>
<td>70.45</td>
<td>72.21</td>
<td>44.51</td>
<td>12.07</td>
<td>0.036</td>
<td>1.132</td>
</tr>
<tr>
<td>71.33</td>
<td>72.21</td>
<td>44.67</td>
<td>12.07</td>
<td>0.036</td>
<td>1.132</td>
</tr>
<tr>
<td>71.33</td>
<td>71.33</td>
<td>45.14</td>
<td>12.09</td>
<td>0.021</td>
<td>1.132</td>
</tr>
</tbody>
</table>
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 Shield

PC/Mac

Monitor

Temp1

Temp 2

Humidity

Pressure

AccelX

AccelZ

MicroSD Card Shield:
Part 4 – Arduino Race Track

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

- LEDs
- Arduino Uno
- PC/Mac
- Monitor
- MicroSD Card Shield
- Temp 1
- Accel Z
- Accel X
- Pressure
- Humidity
- Temp 2
- Temp 1
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
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:

- `#include` adds libraries

- **LOG_INTERVAL** is our sample rate

- `uint32_t` is our time stamp timer

- `logFileName` is our sd card file name

- `ledState` = is our LED blink variable

- `chipSelect` required by sd
**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 from a read cycle
String dataString = "";
```
SD Card Code:

- Define a string variable to contain data headers

```cpp
// Titles for column headings
String sensorNames = "Time Stamp (ms), Temp1 (F), Temp2 (F), RH (%), Pres (psi), XG (g), ZG (g)";
```
**SD Card Code:**

- New Void Setup code...

```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
```
**SD Card Code:**

- New Void Setup code...

```
// turn on Arduino LED

digitalWrite(4, HIGH); // Leave on while power is on

// SD Card setup
// make sure that the default chip select pin is set to output,
pinMode(10, OUTPUT); // To make the SPI (microSD card) interface
pinMode(chipSelect, OUTPUT); // Set the Chip Select pin

// This function will set up the SD card so we can write to it.
// Also the header of the log file will be written to the file.
SDCardInit();
```
New Void Loop:

```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 your time interval
    delay(LOG_INTERVAL);

    // Turn script running leds ON at begining of loop
    digitalWrite(5, HIGH);
    digitalWrite(6, HIGH);
    digitalWrite(7, HIGH);
    digitalWrite(9, HIGH);
}
```
**New Void Loop:**

- Clear the `dataString` at beginning of loop
- Record the time stamp
- Add to `dataString`

```cpp
// Clear out dataString to start a new record.
dataString = "";

// Log the time
timeStamp = millis();
dataString = String(timeStamp);
Serial.print(timeStamp);
```
SD Card Code:

- Convert sensor value to character and store in string

```cpp
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); //
dataString = dataString + "", " + temp1FString;
```
**SD Card Code:**

- Think of `dataString` as a sentence

- Eventually we will write our sentence to the SD card

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

- **Word** = `temp1FString`
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
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 float to string
dataString = dataString + "", " + temp1FString;
```
SD Card Code:

- Convert sensor value to character and store in string

```cpp
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);

dtostrf(temp2F, 6, 2, temp2FString); // Convert

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;
```

62
**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
// The function writeDataToSD returns
if(writeDataToSD())
{
    // Change the State of the LED from
    ++ledState;
    // Handle the case where the LED state
    if (ledState > 1) {
        ledState = 0;
    }
}
// Update the activity LED to show a
// Change
digitalWrite(3, ledState);
```
**SD Card Code:**

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

- If you were to upload the code, would it work?

- No because functions are not defined yet...

```c
// SD card setup
// make sure that the device is defined
pinMode(10, OUTPUT);
pinMode(chipSelect, OUTPUT);

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

// The function writeData
if (writeDataToSD())
{
    // Change the State of
    ++ledState;
    // Handle the case when
```
- Here is the code for `SDCardInit();`

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

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

- But wait there is more!

- We still have the other function

```cpp
Serial.print("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
    return(0); // Unable to write to SD
}"
**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 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:

- 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 “.”
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

**AND GO HOME**