# The Scientific Ballooning Course

Fall Semester, Units 1 -17

The LaACES Website


Dana Browne

Department of Physics & Astronomy

LSU

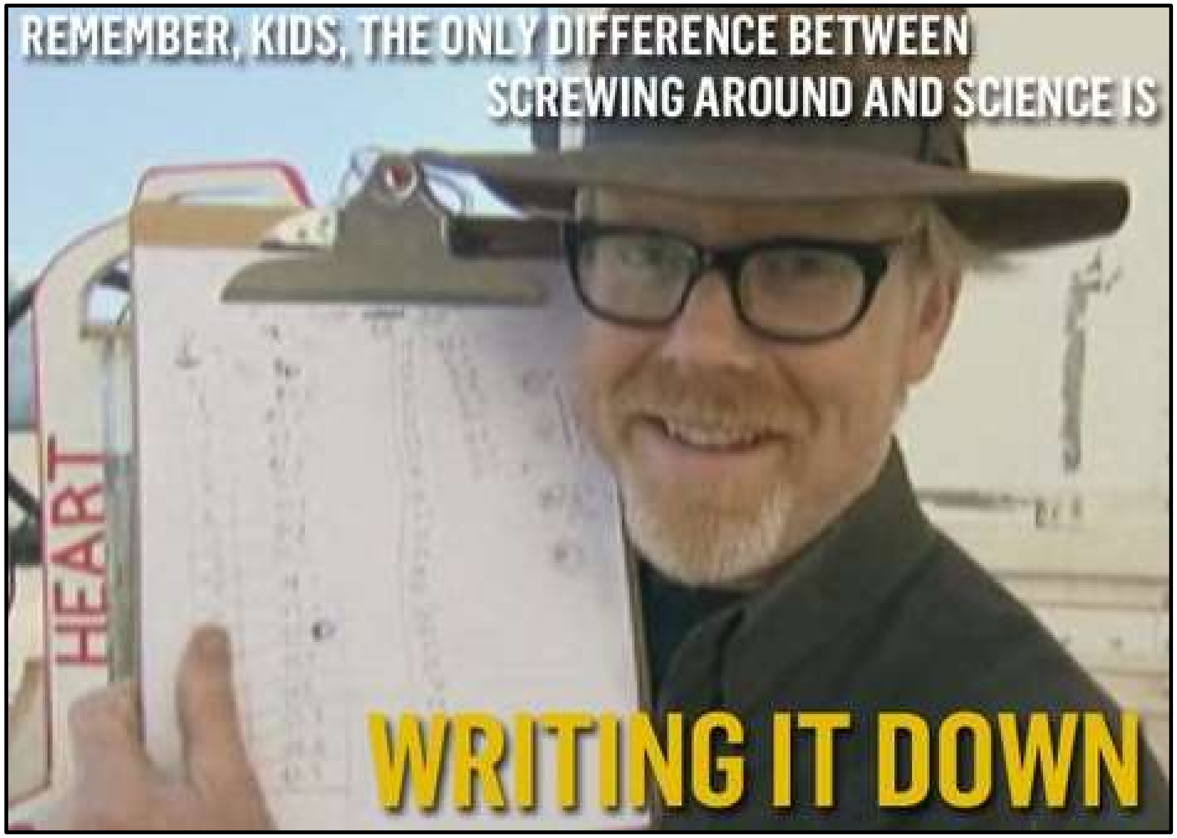
# The goal of this course

To provide the knowledge and practical skills to successfully design and build a payload to make scientific measurements while being carried to 100,000 ft with a sounding balloon and then returned to earth.

# What Students Will Learn

- Recording, writing about and presenting technical projects
- Electronics background and practical electronics skills
- Microcontroller programming
- Analog sensors and signal processing
- Digital Data acquisition, digital communications
- Handling GPS data
- Sensor Interfacing
- Data storage
- Payload mechanical and thermal design (spring semester)

# Real Science: Document It!




Adam Savage, cohost of Mythbusters

# Communicating Results

- Lab Notebooks
  - How to keep them, what goes in them
- Technical Reports
  - Formats (templates provided)
  - Draft -> Feedback -> Revise -> Feedback -> Revise
- Documents delivered to LaACES management
- Documents “graded” Pass / Fail
- Scientific presentations (in spring)

# Electronics

- Basic concepts
- Common components
- DC measurements
- Using Datasheets
- Schematics, Prototyping and Circuit Layout
- Assembly Techniques
- Thru-hole and Surface Mount Soldering
- Using an oscilloscope to view signals

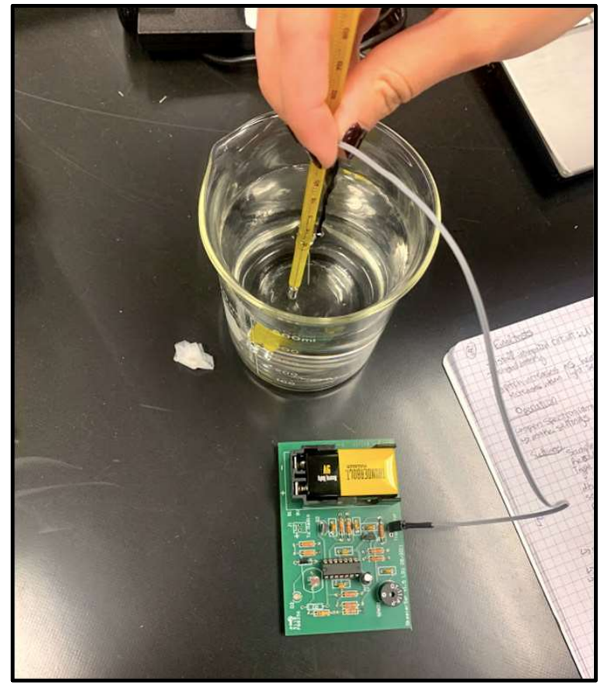
# First Project: SkeeterSat

- Produces audible beeps
- Beep rate and pitch encode data about ambient light and temperature conditions
- Skills:
  - Identifying components
  - Thru-hole Soldering
  - Circuit Testing


The SkeeterSat board.

# SkeeterSat Calibration

- Every instrument must be calibrated to be useful
- Compare SkeeterSat to standard thermometer over a temperature range
- Skills:
  - Precision of measurements
  - Data Fitting Techniques



Calibrating the SkeeterSat

# SkeeterSat Calibration Report

- First foray into technical report writing
- Template will be provided
- Create draft, get feedback, revise
- Repeat until satisfactory
- Faculty advisor should review work
- Recommend internal deadline of 10/9/20 for faculty advisor review and feedback
- Report is due to LaACES Management October 16, 2020
- Report will be reviewed by LaACES Management

# Programming

- Algorithms and Flowcharts
- Programming Concepts
  - Variables, control statements, repetitive operations, data reading and writing
- Features of the Arduino Mega
- Programming the Mega

The Arduino Mega microcontroller will handle data processing for the payload

# Programming Project

- Convert tasks into Arduino code
- Start simple and add new features
- Version control
- Learn to document code
- Measuring and displaying analog data

```
// These constants are used to give names to the pins used:
const int analogInPin = A0;  // Analog input pin that the potentiometer is attached to
const int analogOutPin = 9; // Analog output pin that the LED is attached to

int sensorValue = 0;        // value read from the pot
int outputValue = 0;        // value output to the PWM (analog out)

void setup() {
  Serial.begin(9600);    // initialize serial communications at 9600 bps:
}

void loop() {

  sensorValue = analogRead(analogInPin);   // read the analog in value:
  outputValue = map(sensorValue, 0, 1023, 0, 255);   // map it to the range of the
analog out:
  analogWrite(analogOutPin, outputValue);  // change the analog out value:

  // print the results to the serial monitor:
  Serial.print("sensor = ");
  Serial.print(sensorValue);
  Serial.print("\t output = ");
  Serial.println(outputValue);

  // wait 2 milliseconds before the next loop
  // for the analog-to-digital converter to settle
  // after the last reading:
  delay(2);
}
```

# Digital Data

- Signal waveforms
- Digital signals, logic levels
- Pulse Width Modulation (PWM)
- Digital Input and Output
- Serial communications, (I2C, SPI, ...)
- Handling digital data with the Mega

# Data Acquisition

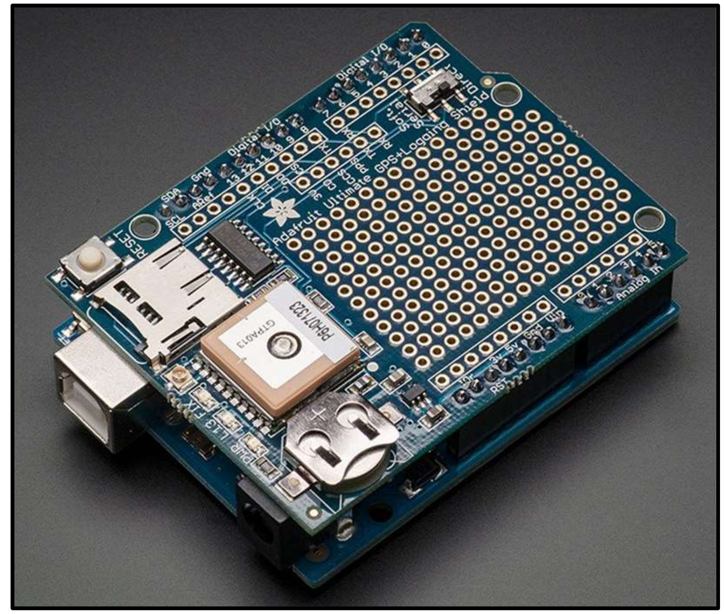
- “How an instrument makes a reading”
- Types of analog sensors
- Signal conditioning, amplification, filtering
- Analog/Digital (A/D) conversion
- Analog data handling on the Mega
- Building an analog sensor interface

# Analog & Digital Practice

- Write Arduino code to
  - Read analog data
  - Digital I/O
  - Pulsewidth Modulation
  - Serial Communications
- Viewing signals with oscilloscope

# Enhancing the Arduino

- Add-on circuits “shields”
- Plug into Mega board
- Provide extra functions
  - Extra sensors
  - Bluetooth or Wifi
  - GPS data
  - Data Storage (EEPROM, SD)



GPS Logger shield with GPS receiver, real time clock, and SD card data logger plugged into the Mega

# Where is my payload?

- Global Positioning System (GPS)

  How it works

- GPS receivers
- GPS data streams
- Decoding GPS data


External and handheld GPS receivers

# GPS & Data Logging

- Write code to
  - control GPS receiver
  - receive GPS data
  - parse GPS data
  - Read/write to memory
    - EEPROM
    - Flash memory/SD

```
// Check if a new NEMA sentence has been received
if (GPS.newNMEAreceived()) {
  NMEAsentence = GPS.lastNMEA();

  if (NMEAsentence.substring(0,6) == "$GPVTG") {    // Is the sentence VTG?
    parseVTG(NMEAsentence);       // Parse the sentence

    // Print sentence and it's parsed information
    Serial.println(""); Serial.print(NMEAsentence);
    Serial.print("Degrees True: "); Serial.println(degreesTrue);
    Serial.print("Degrees Magnetic: "); Serial.println(degreesMagnetic);
    Serial.print("Speed (knots) "); Serial.println(speedKnots);
    Serial.print("Speed (km/hr) "); Serial.println(speedKmHr);
    Serial.print("Mode: "); Serial.println(modeVTG);
    Serial.print("Checksum: "); Serial.println(checksumVTG);
  }

  if (NMEAsentence.substring(0,6) == "$GPGSA") {    // Is the sentence GSA?
    parseGSA(NMEAsentence);       // Parse the sentence

    // Print sentence and it's parsed information
    Serial.print(NMEAsentence);
    Serial.print("Mode: "); Serial.println(modeGSA);
    Serial.print("Fix mode: "); Serial.println(modeFixGSA);
    Serial.print("PRNs: "); Serial.print(PRNs_GSA[0]);
    for (int i=1; i<12; i++) {
      Serial.print(","); Serial.print(PRNs_GSA[i]);
    }
    Serial.println(" ");
```

Code to Read and Interpret GPS data

# Capstone Project

- Bring all the knowledge and skills together
- Mega plus GPS shield
- Interface an analog sensor to the Mega
- Produce sensor readings with time-stamps
- Data stored on SD card
- Written project report submitted to LaACE Management

# Data Logger Report

- Another opportunity in technical writing
- Template will be available
- Process: draft->feedback->revise->...
- Advisors must review and see revisions before submitting
- Due to LaACES Management by Dec 18th
- Report will be reviewed

# What the students will learn

- Electronics, components, schematics, datasheets
- Circuit assembly and testing
- Programming basics, flowcharts, Arduino code development
- Analog and Digital Signals, signal conditioning and sensor interfacing
- Serial communications, digital I/O with the Arduino
- Adding a GPS shield
- Project documentation, technical report writing
- Putting it all together:  A GPS timestamped data logger

# Required Deliverables

- SkeeterSat Calibration Report

  Suggest internal deadline 10/9

  Submit to LaACES Mgmt 10/16

- Sensor Interface Report

  Suggest internal deadline 12/16

  Submit to LaACES Mgmt 12/18

- Both reports reviewed by LaACES Management