# UAVs@Berkeley

## Bootcamp 1
## Fall 2018

# Intro

# Bootcamp Materials

- All slides and documentation is available in the Google Drive
  - Club Documentation > Technical Documentation > 1_Bootcamp Curriculum

# Intro

- Block 1 - Trey
  - Digital Fabrication, Slicing, and 3D Printing
- Block 2 - Darius
  - Quad Anatomy, wired and wireless communication, Lipos
- Block 3 - Alex
  - Arduino Intro

# Digital Fabrication, Slicing, and 3D Printing

## Block 1, Trey Fortmuller

# What is digital fabrication?

Digital fabrication is a production process involving automated additive and subtractive manufacturing. These processes tend to be faster, more accurate, and more consistent than manual fabrication processes.

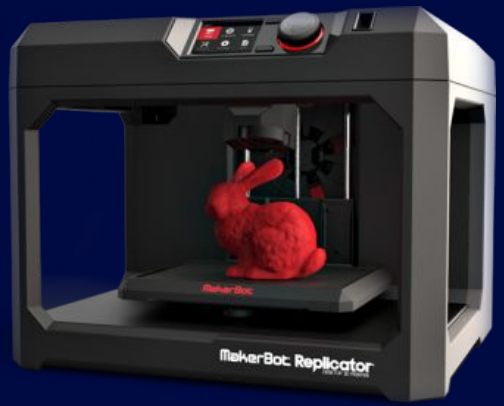

3D Printing (additive)


Laser Cutting


CNC (Computer Numerical Controlled) Milling

# 3D Printing Technologies

- The most accessible, cheap process of the automated manufacturing processes
- Extremely useful for prototyping, visualization, or use in an end-product
- Different technologies exist, they all have their own advantages and disadvantages



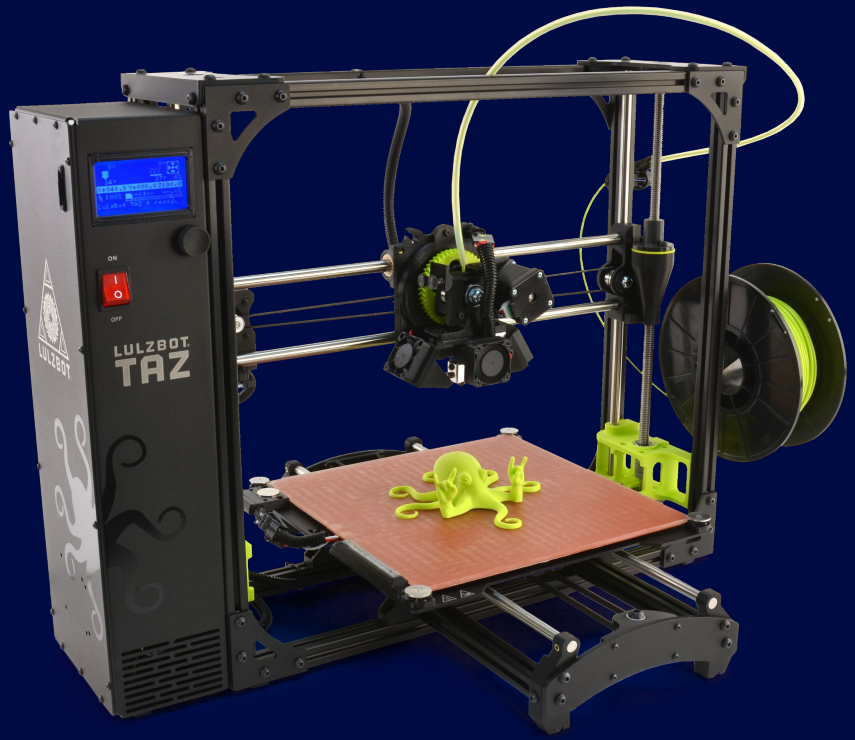

Desktop FDM 3D Printer

SLA (stereolithography apparatus) 3D Printer


Metal Sintering 3D Printer

# 3D Printing Technologies

- Desktop FDM machine anatomy
  - Frame
    - Spool holder
  - Build plate
    - Can be heated
  - Hotend
    - Extruder
    - Fans
  - Sensors and endstops
  - Stepper motors
  - GT2 timing belts
  - Microcontroller
  - Motor drivers
  - Power Supply Unit
  - Interface



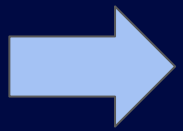

Desktop FDM 3D Printer

UAVs@Berkeley

# Design Workflow for a 3D Printed Part

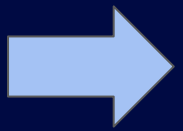
## CAD

- Obtain an STL from online
- Design the part using CAD software
  - Autodesk Fusion 360
  - SolidWorks
  - Inventor
  - Catia
  - NX
  - Others...



## Slicing

- Throw the STL in your slicer and set its parameters
  - Cura
  - Slic3r
  - Simplify3D
  - Others...
- Output Gcode for the printer

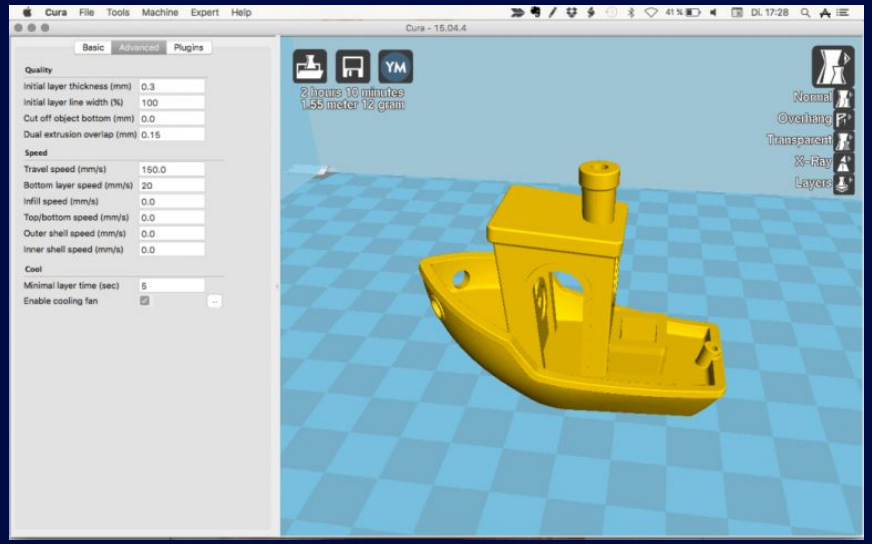


## Printing

- Load the Gcode
- Wait for a failure to occur
  - Back to step 2
- Get your part, realize it doesn't work
  - Back to step 1

# Slicing Pointers

- We'll go through several slicing examples but generally, there are 5 main things to consider:
  - Material
  - Strength
  - Time
  - Adhesion
  - Supports
- The importance of failure
- Slicers will help you, but only you can be logical about the way you orient a part and create the printing file
- Don't use 3D printing where it isn't needed




Cura Slicing Interface

# Slicing Demos

- Intro 3D benchy
- An STL exported from CAD
- An STL downloaded from Thingiverse
- GrabCAD

# Quadcopter Anatomy

Block 2, Darius Dastur

# Parts Overview

Core:

- Frame
- Flight Controller (FC)
- Motors
- Electronic Speed Controllers(ESC)
- Power Distribution Board(PDB)
- RF Receiver
- Propellers
- Battery

Extras:

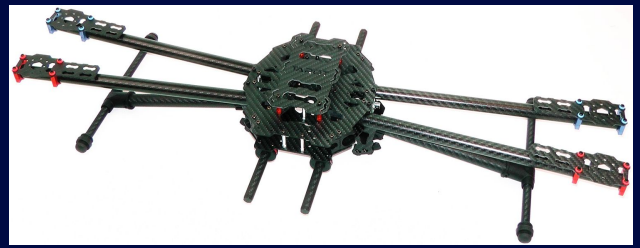
- Camera
- Video Transmitter
- Video Transmission Antenna
- Sensors (obstacle detection)
  - GPS
  - Barometer
  - Magnetometer

# Frame

Highly dependent on the desired function (payload, sensors, flight time, etc.)

- What impact does the number of arms have on the drone?
- How significantly does weight effect the flight time and other factors?

- Size
  - Measured motor to motor(diagonally)
- Thickness
  - 3mm, 4mm, 5mm
- Types of carbon fiber
  - Unibody vs Parts
  - Twill Weave
    - 1k vs 3k vs 12k
- Hardware
  - Steel, Aluminum, and Titanium

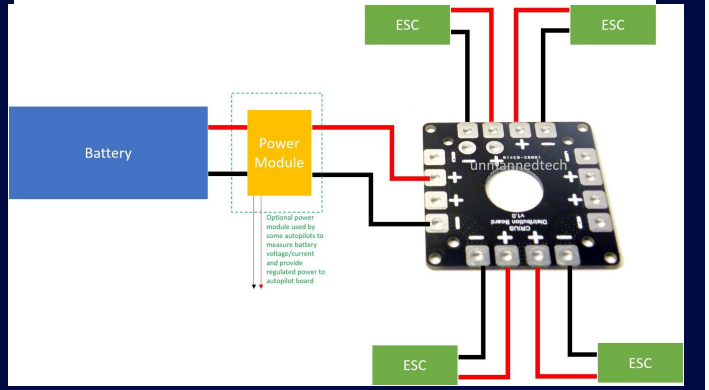


# Flight Controllers and Power Distribution Boards

- Main Purpose is to keep UAV level and control the drone from user input

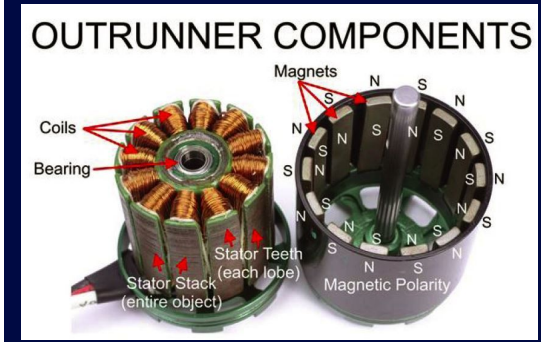
---

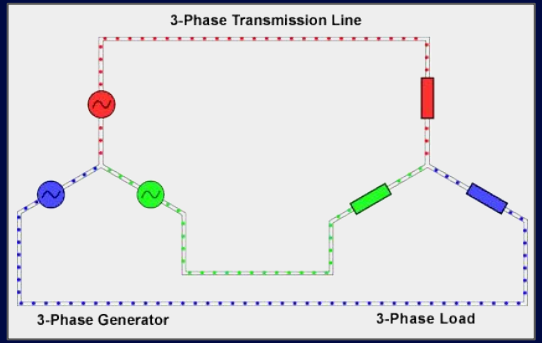
- Racing Flight Controllers
  - Betaflight, Cleanflight, KISS, Raceflight
- Videography
  - Naza
- Engineering
  - Arducopter (Ardupilot), Pixhawk (PX4)

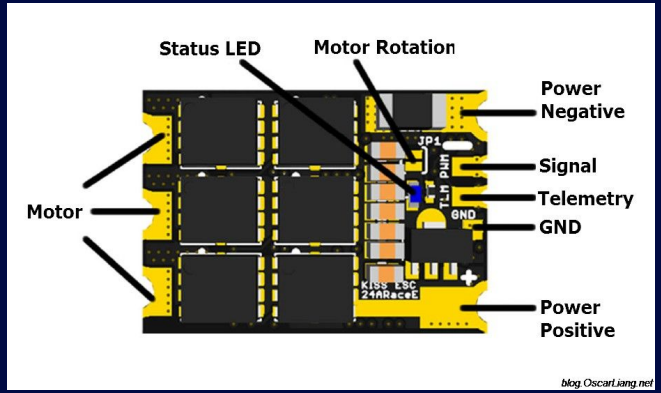
# Brushless Motors and Electronic Speed Controllers

- Motor specifications
  - Size XXYY
    - stator diameter and height in mm
    - Ex. 2205 = 22mm diameter stator with 5mm height
  - KV rotations per minute per volt
- How does a motor work?
  - 3 Phases
  - AC current through each phase



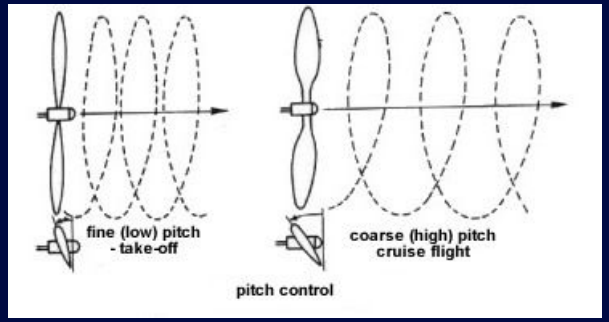




- How does an ESC work?
- Important specifications for an ESC
  - Rated Voltage
  - Max Current
  - Burst Current



# Propellers and Batteries

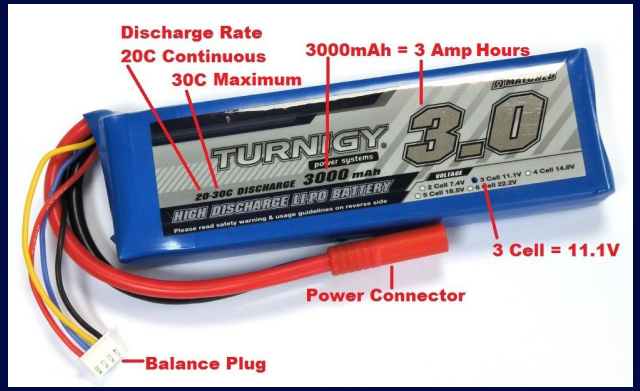
- Naming convention
  - 5x4x3(5040x3), 5040BN, 9.4x5
- Materials
  - Polycarbonate, ABS, Carbon Fiber, Glass Fiber
- Vortex Ring State(prop wash)
  - Descending too quickly results in turbulent air disrupting normal airflow


# Batteries

- Cell specification
  - Series - S (3S)
  - Parallel - P (3P)
- Capacity
- Discharge rates
- Chemistries
  - Li-ion
    - Lithium Polymer
  - NiMH
  - Lead Acid Batteries

- A typical LiPo battery rated at **3S1P 3,000mAh 20C** can be broken down as follows:
  - **3S1P** means 3 cells in series none in parallel, since individual cells are 3.7 volts, the battery is 11 volts.
  - **3,000mAh** means since the cells are only in series each 3.7 volt cell has 3,000mAh or 3 Amp Hours of capacity.
  - the 20C rating means the Battery can be discharged at a rate 20 times its 3 Amp Hour capacity up to 60 Amps continuously. 3.0Ah x 20 = 60Amps



# Lipo Safety

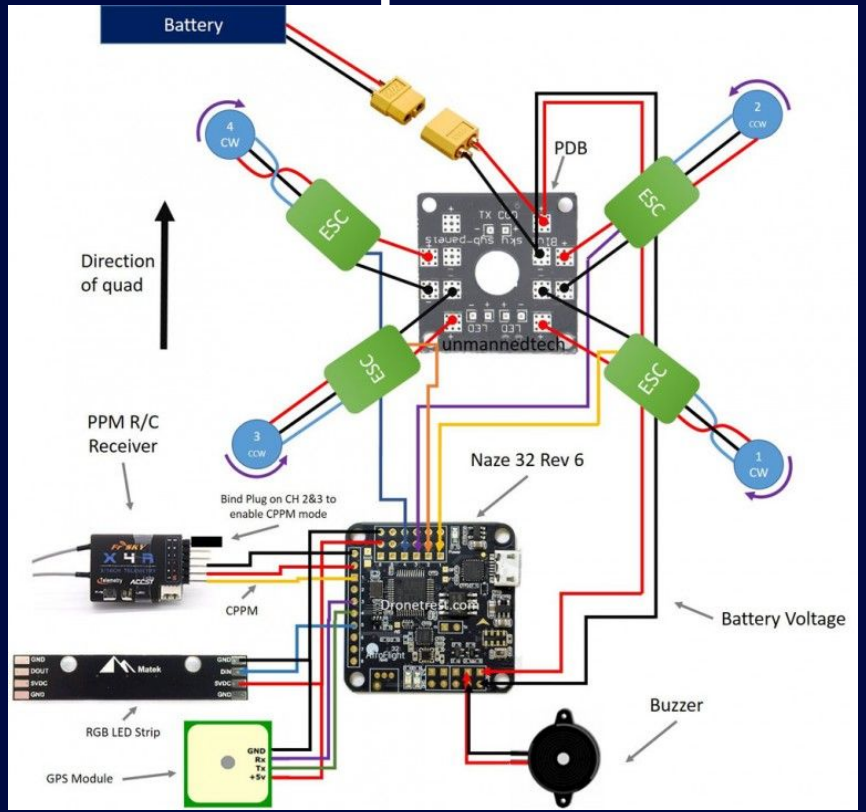
Lithium Polymer batteries (LiPos) are dangerous...

Guidelines:

- Discharge
  - Do not discharge batteries below 3.7V per cell
- Charge
  - Never charge a battery above 1C
  - Never leave charging batteries unattended
- Storage
  - Store Lipos at 3.8V per cell
  - Do not leave batteries charged for more than 2 days

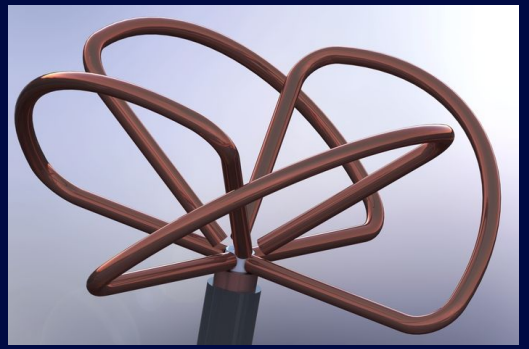
# Communication Between components

- RF Reciever to Flight Controller
  - Sbus, pwm, ppm,
- Flight Controller to ESC
  - Pwm, oneshot, multishot, dshot
- Flight Controller to Video Transmitter
  - Smartaudio



# Radio Waves

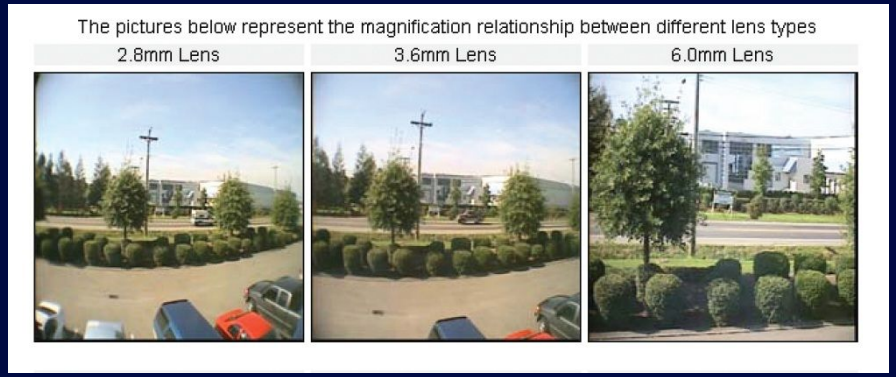
- Frequencies used by consumers
  - 900MHz, 1.3GHz, 2.4GHz, 5.8GHz
- How to receive RF signal
- Antennas
  - Polarization
    - Linear
    - Circular Polarized (Left & Right)
  - Shapes
    - Dipole, clover leaf, helical, crosshair, patch
- Ham Radio



# FPV equipment

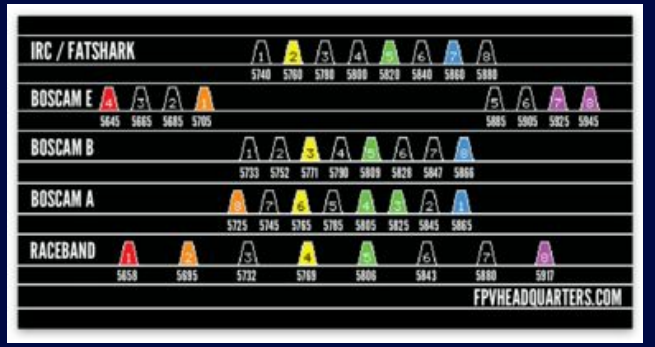
Camera

- Complementary Metal Oxide Semiconductor(CMOS) vs Charge Coupled Device(CCD)
- Global vs Rolling shutter
- Lens sizes



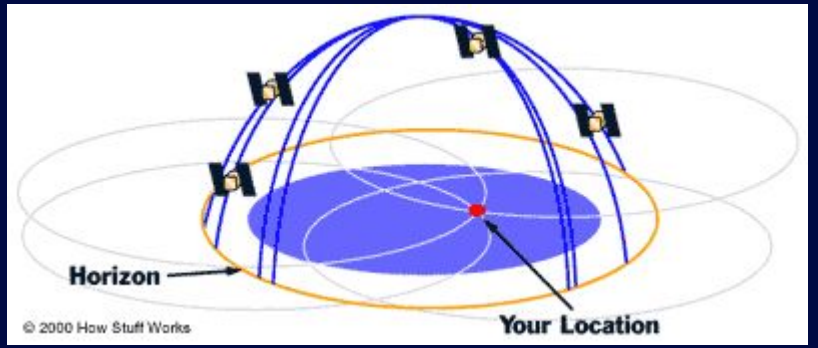

Video Transmitter(VTX)

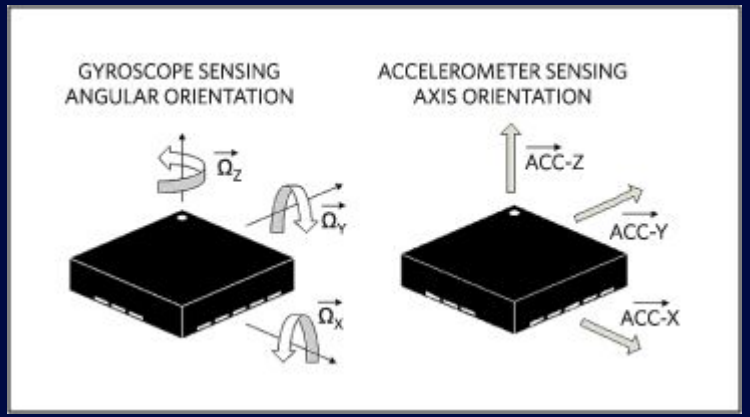
- Output Power
- Frequency Bands
  - A, B, E, F, R

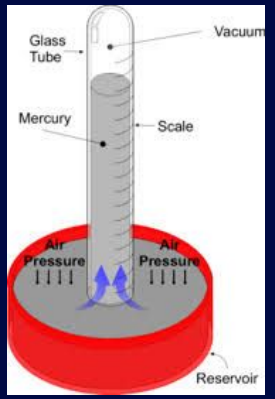


# Sensors

- Global Positioning System
  - Location
  - Trilateration
- Barometer
  - Pressure -> altitude
- Magnetometer
  - Orientation
  - Senses magnetic field
- Accelerometer
- Gyroscope



# Cameras and Gimbals

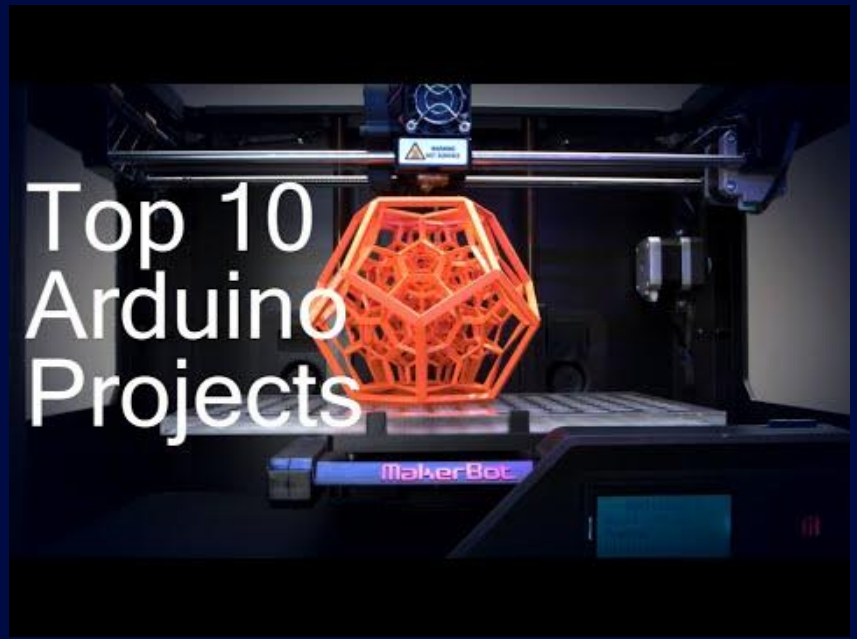
- Allows mounted cameras to be inertial
- Stabilize shots by isolation of vibrations

# Arduino Intro

Block 3, Alex Chan

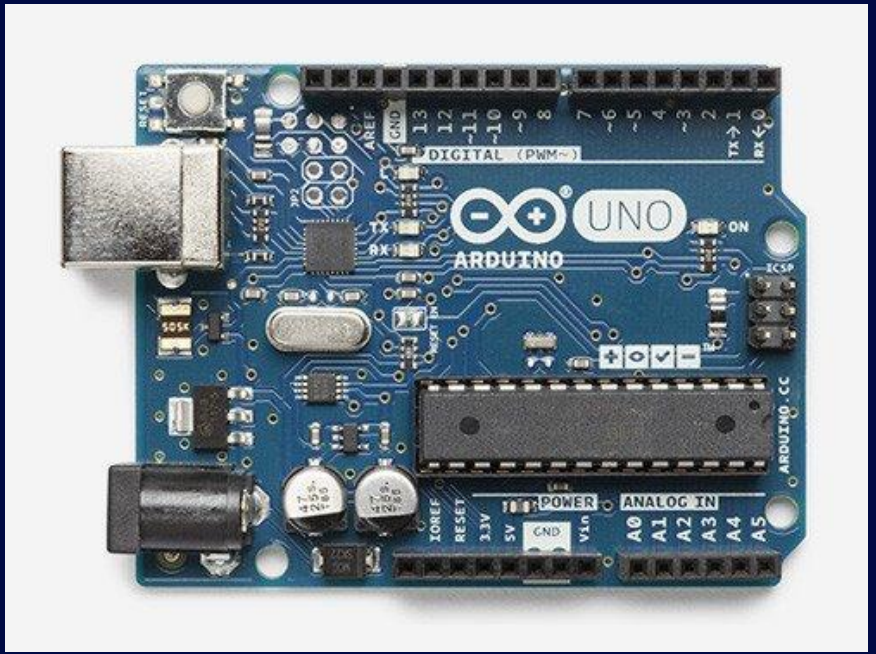
# Follow Along!

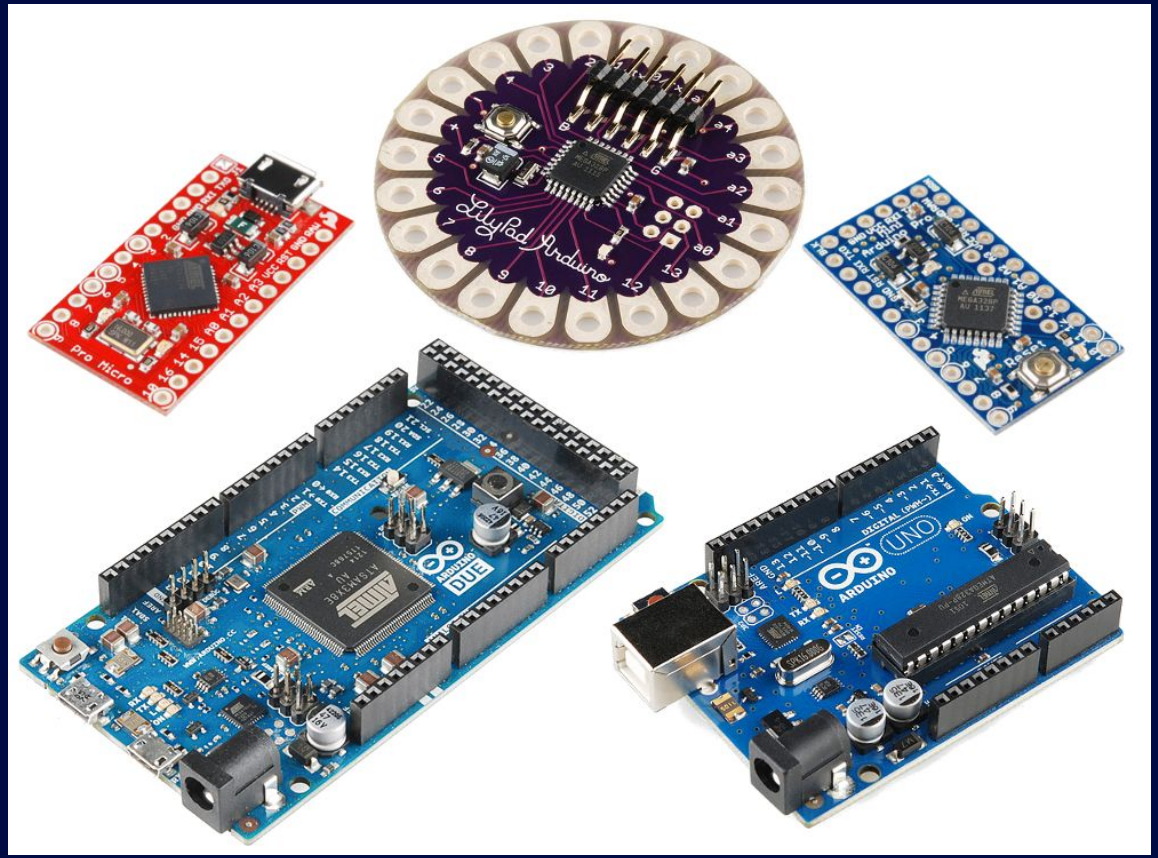
https://www.arduino.cc/en/Main/Software

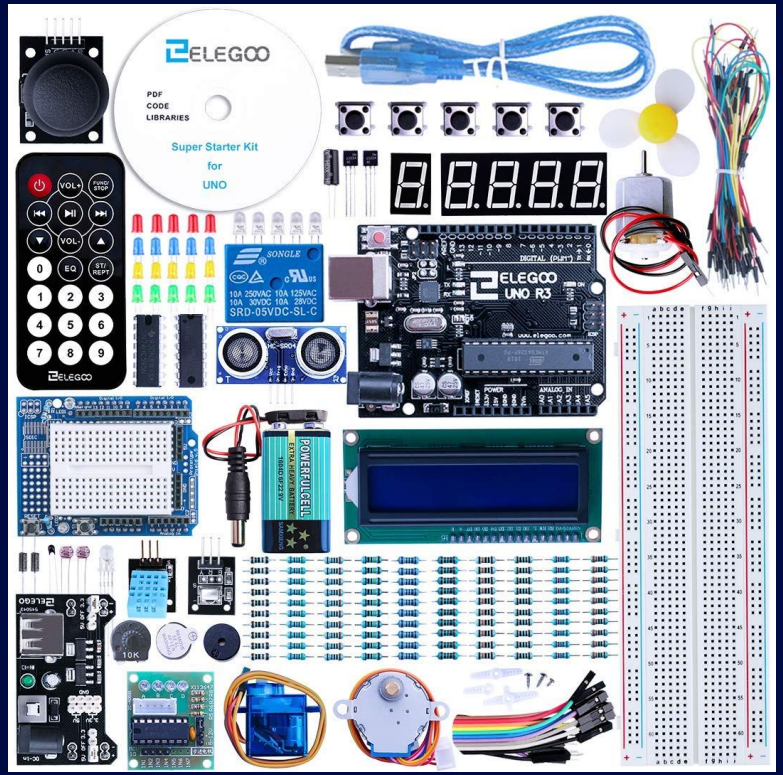

Top 10
Arduino
Projects
MakerBot

# What is Arduino?

- Circuit board w/ micro-controller
- Open source
- Easy to code customized circuits

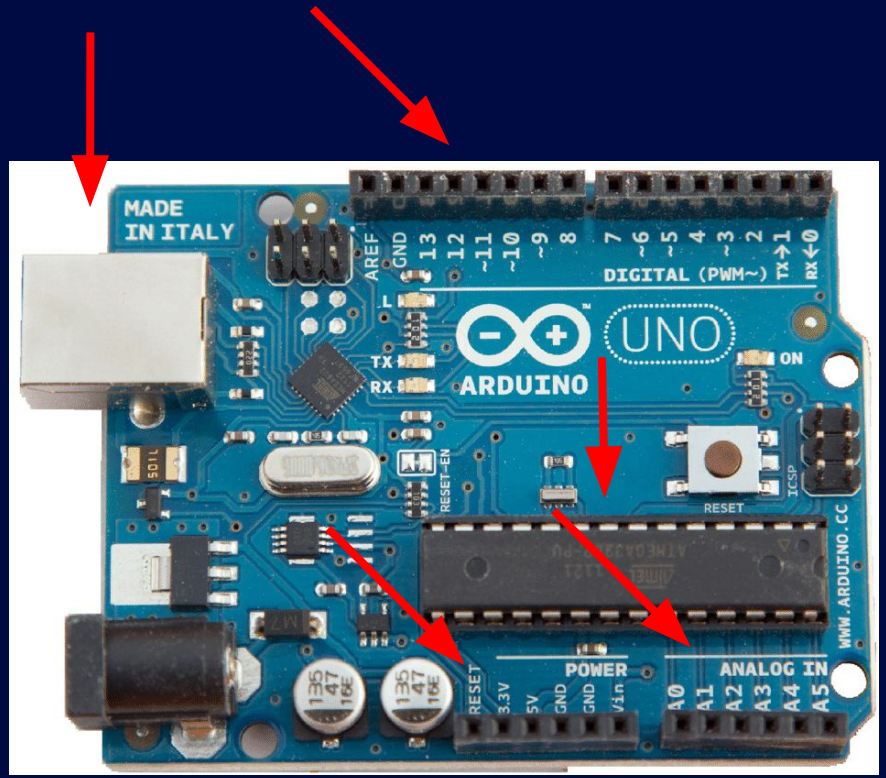



Pro Micro
LilyPad Arduino
ARDUINO
DUE
ARDUINO
UNO


ELEGOO
PDF
CODE
LIBRARIES
Super Starter Kit
for
UNO
SONGLE
SRD-05VDC-SL-C
ELEGOO
UNO R3
POWERFULCELL
10K

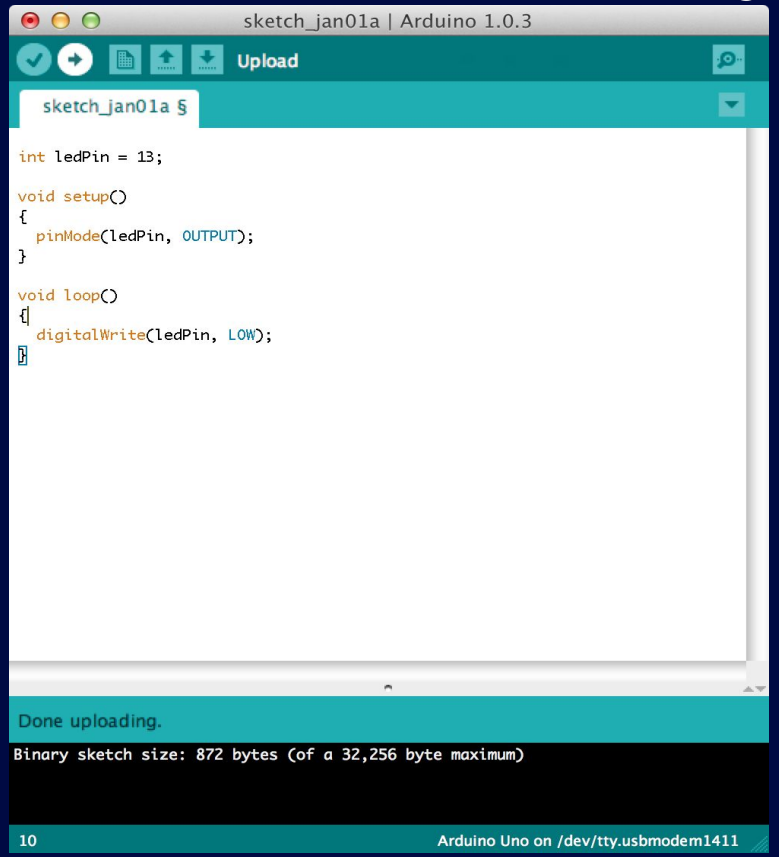
# Hardware Components

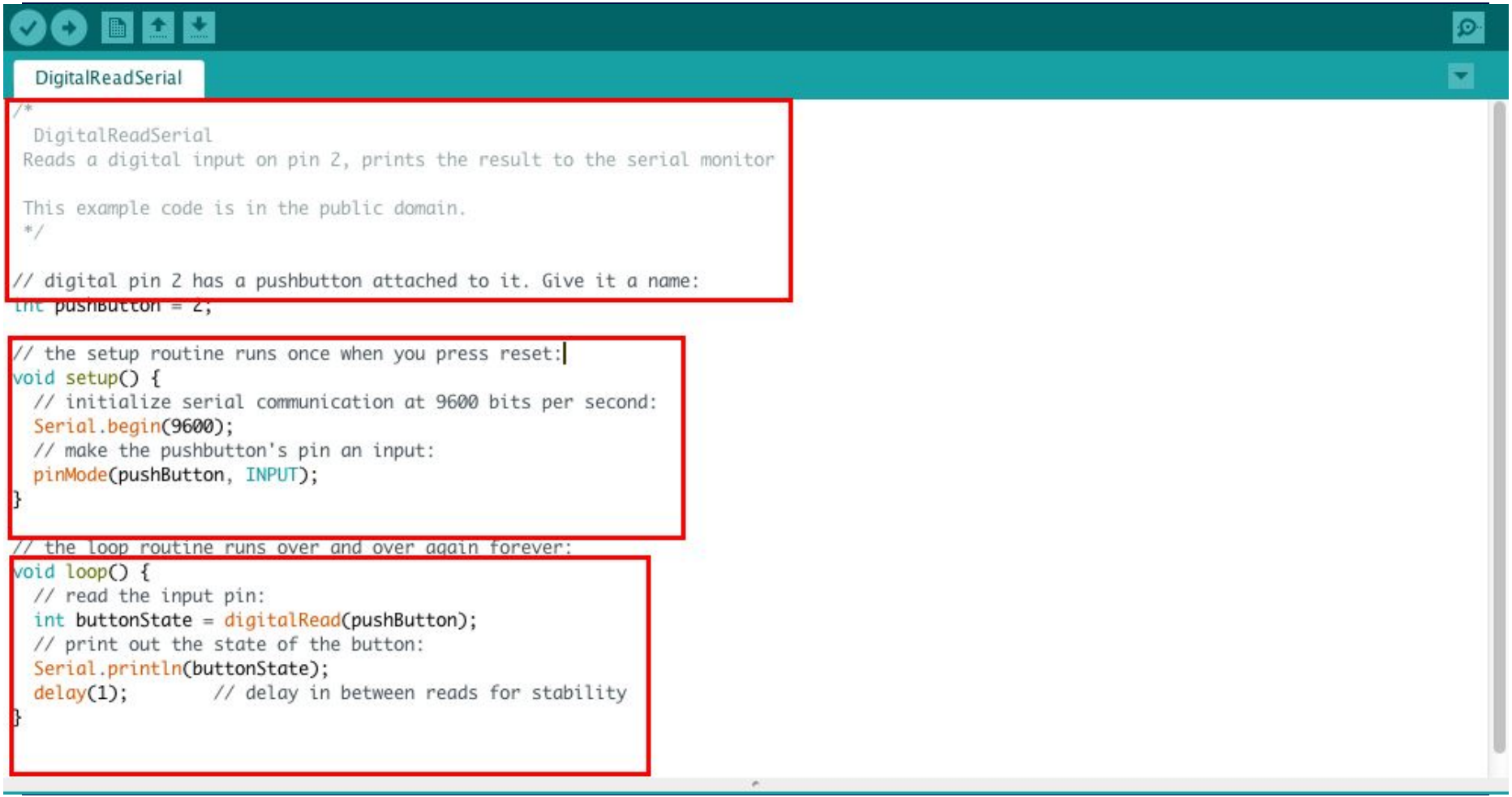
- Integrated circuit
- USB connection
- Digital inputs/outputs
- Analog inputs
- Variable power sources



# Software

- Interfaces with micro-controller
- Basically C++ (Don't need to be an expert at C++ to use!)
- Certain variables are mapped to different inputs/outputs of the Arduino




DigitalReadSerial
/*
DigitalReadSerial
Reads a digital input on pin 2, prints the result to the serial monitor
This example code is in the public domain.
*/
// digital pin 2 has a pushbutton attached to it. Give it a name:
int pushButton = 2;
// the setup routine runs once when you press reset:
void setup() {
// initialize serial communication at 9600 bits per second:
Serial.begin(9600);
// make the pushbutton's pin an input:
pinMode(pushButton, INPUT);
}
// the loop routine runs over and over again forever:
void loop() {
// read the input pin:
int buttonState = digitalRead(pushButton);
// print out the state of the button:
Serial.println(buttonState);
delay(1); // delay in between reads for stability
}

# Functions

```
type functionName(parameters) {
        statement;
        ...
}
```

# Data Types

# Integer

Keyword: int

Whole numbers from -32,768 to 32,767

Examples:

```
int myNumber = 15;

int addTwo(int x) {

    return x + 2;

}
```

# Float

Keyword: float

Numbers from $-3.4028235*10^{38}$ to $3.4028235*10^{38}$

Examples:

```
float pi = 3.14;

float addTwo(float x) {

    return x + 2.0;

}
```

# Boolean

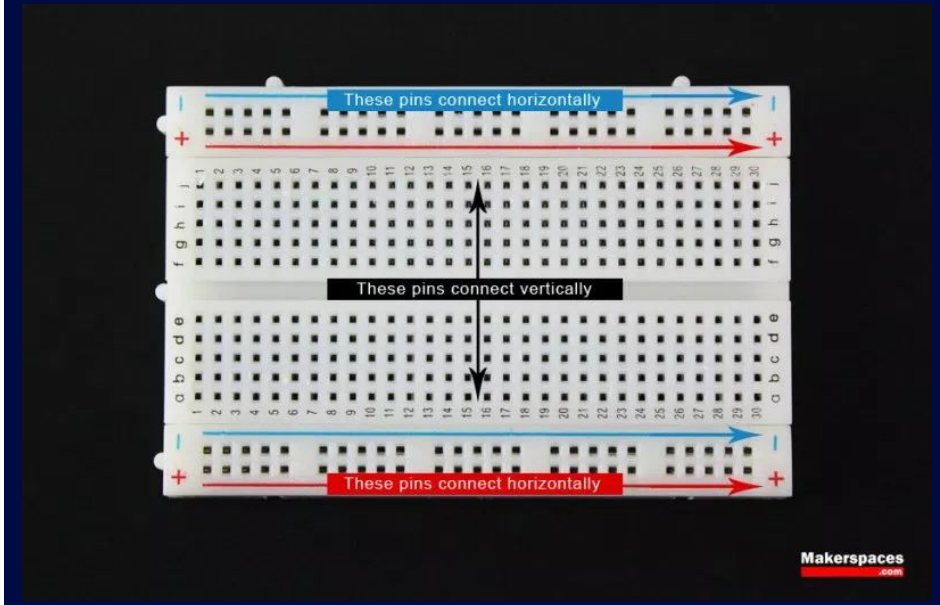
Keyword: boolean

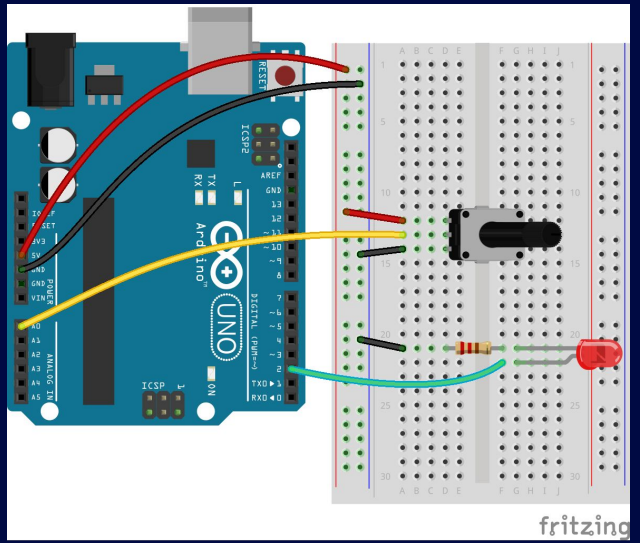
Value that either evaluates to true or false

Examples:

```
boolean myBoolean = true;

boolean isBiggerThan5(int x) {

    return x > 5;

}
```

# Breadboard - the canvas of your circuit design

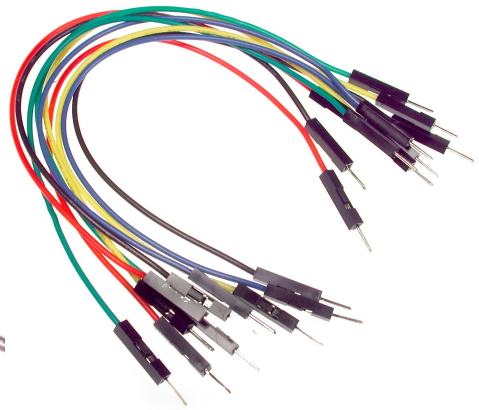


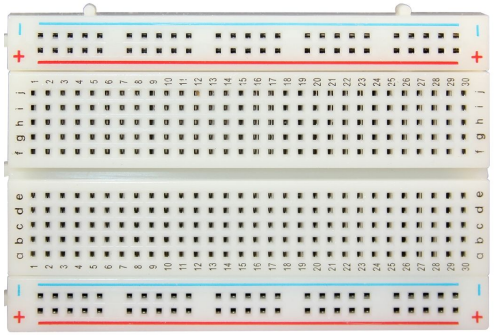



# Blink LED

UAVs@Berkeley

# What you will need

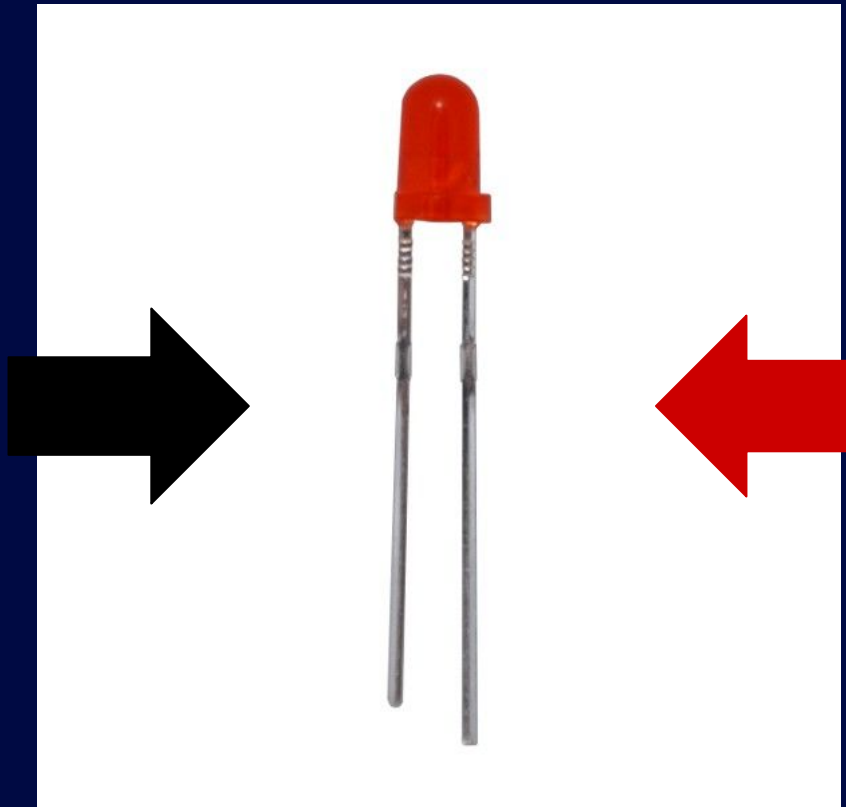




LED 220Ω resistor Jumper wires Breadboard

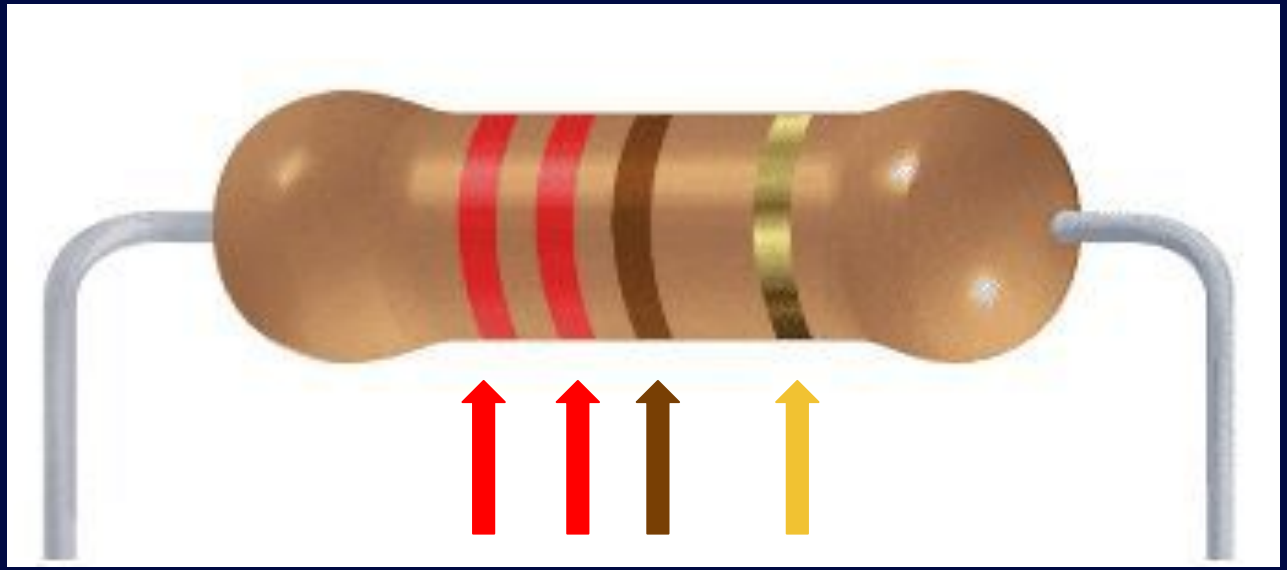
# Light Emitting Diode

Cathode - connects to ground



Anode- connects to positive voltage

# Resistors

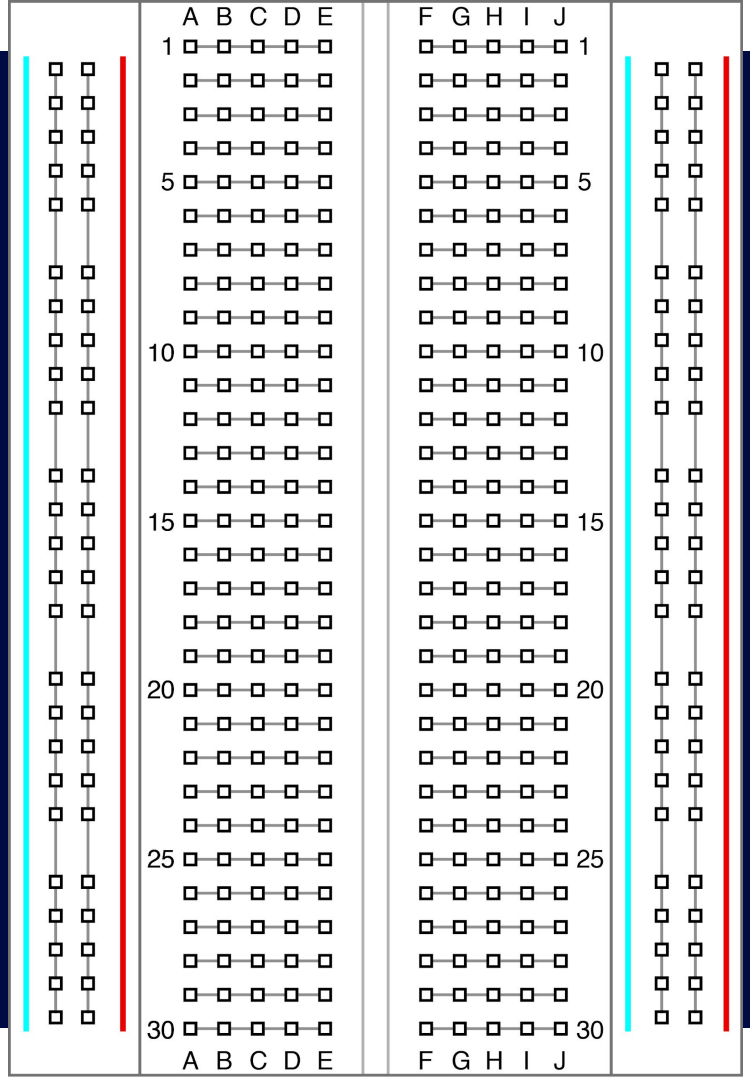


**2 2 0 ± 5%**

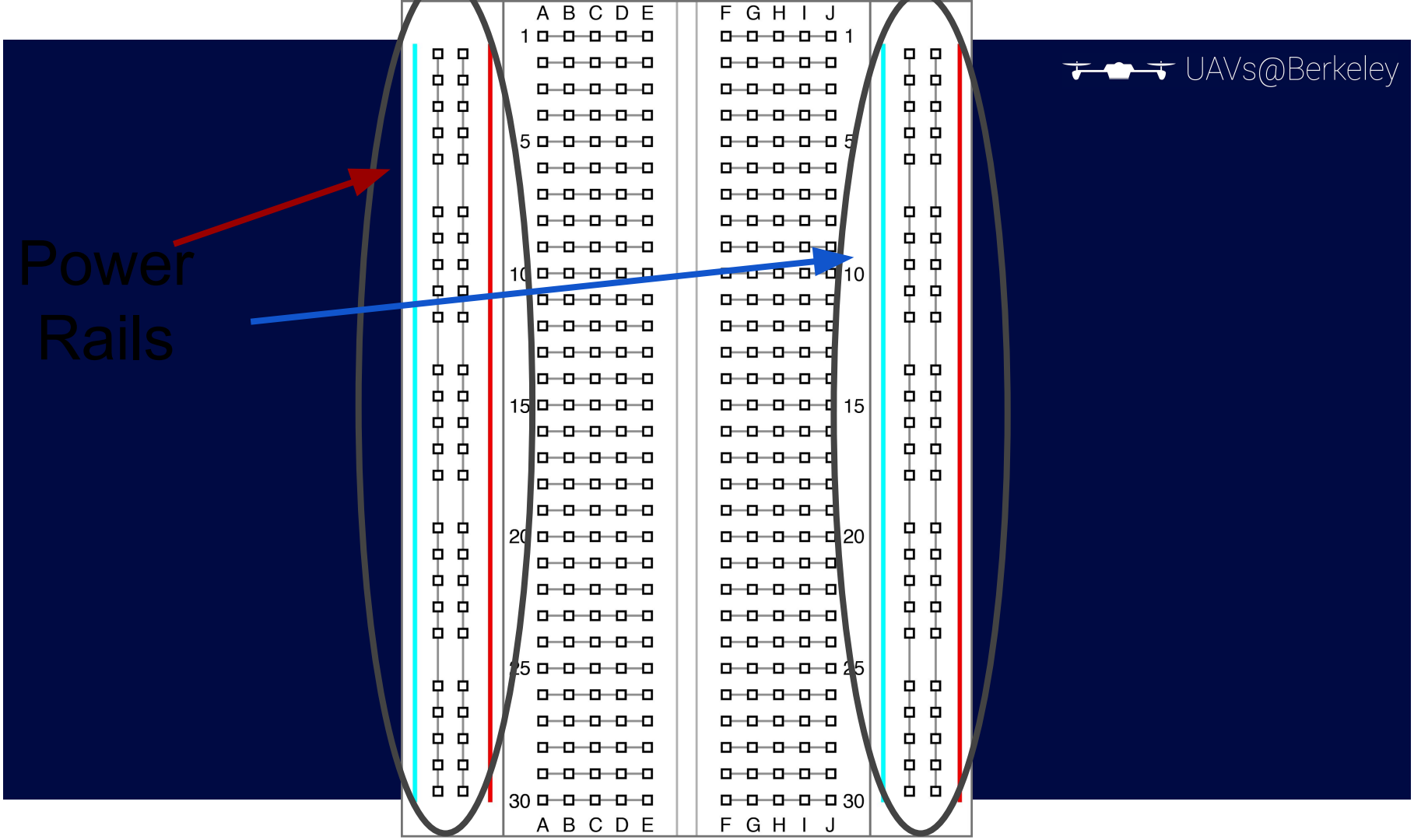
# Instructions

1. Put one end of the resistor in digital pin 13
2. Put the short lead on the LED in the GND pin next to pin 13 and bend the other lead toward the resistor
3. With the alligator clip connect the resistor and LED
4. Plug your Arduino into your computer
5. Open the Arduino program
6. Go to File -> Examples -> Basics -> Blink

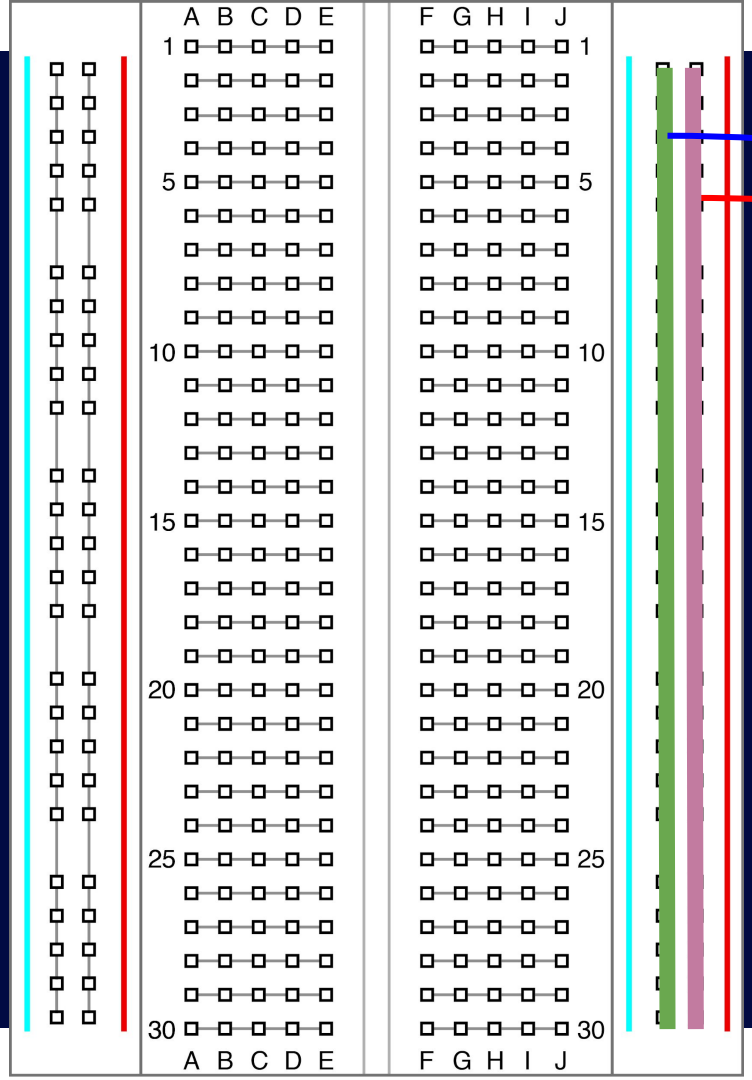
# Breadboards



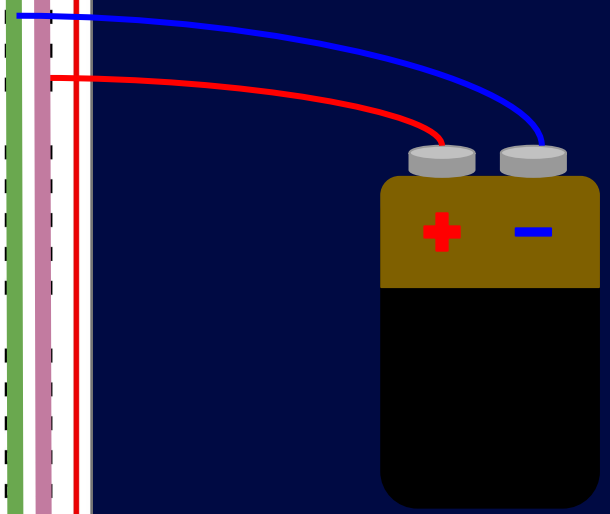

UAVs@Berkeley


Power
Rails
UAVs@Berkeley
A B C D E
F G H I J
1
5
10
15
20
25
30

The entire power rail or column shares the same voltage

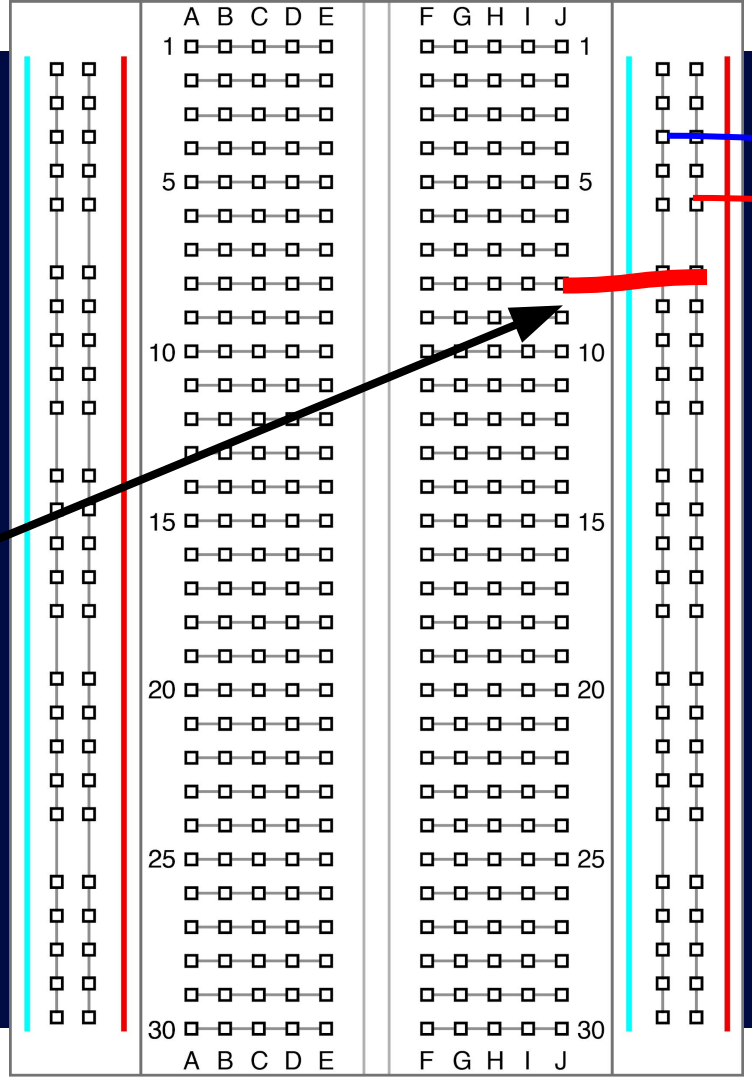



UAVs@Berkeley

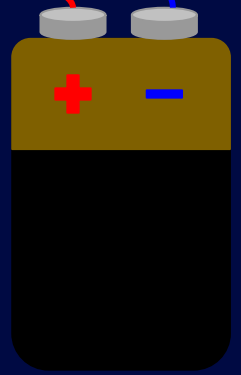


Battery or Power Supply

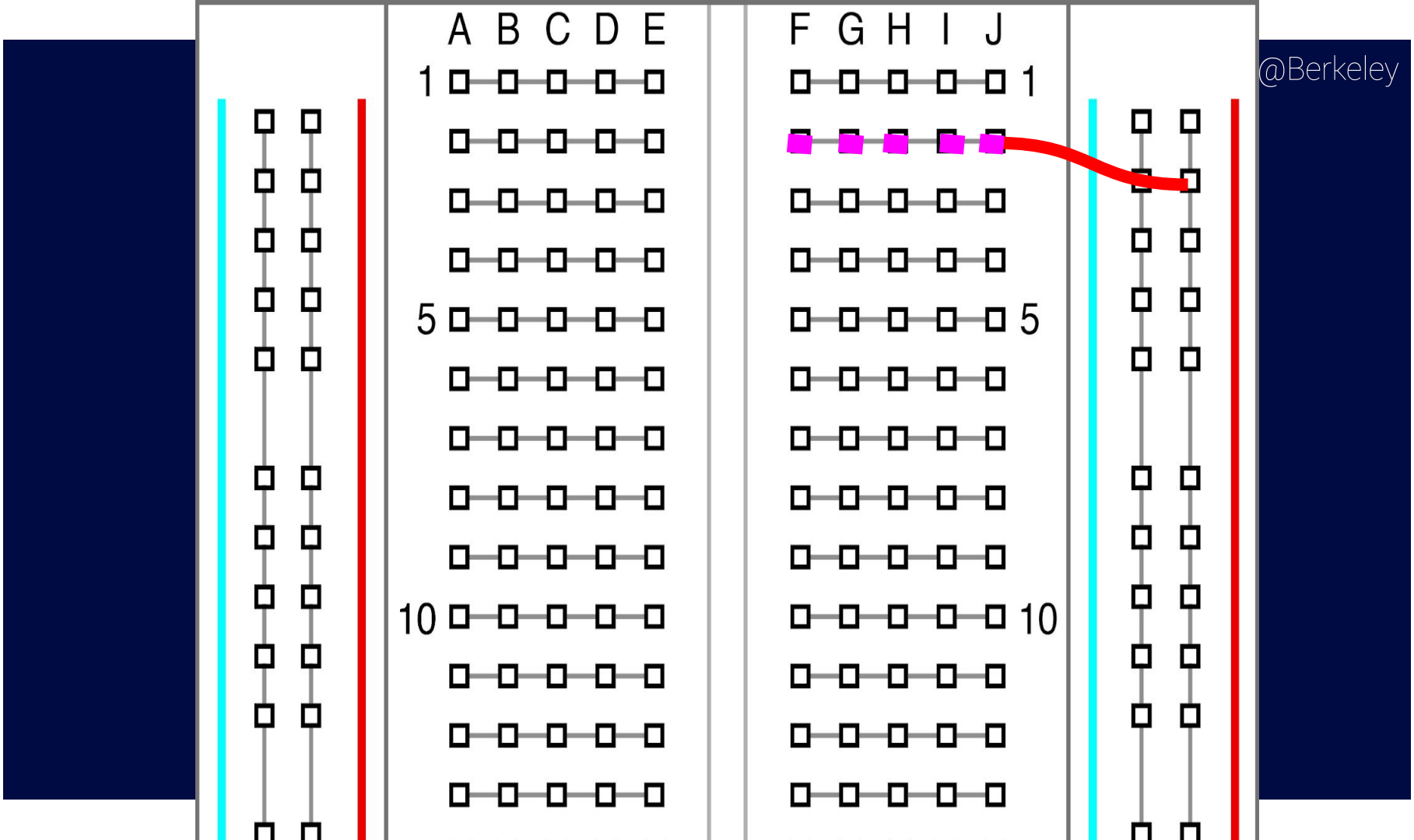
How do we get power to the rest of the board?

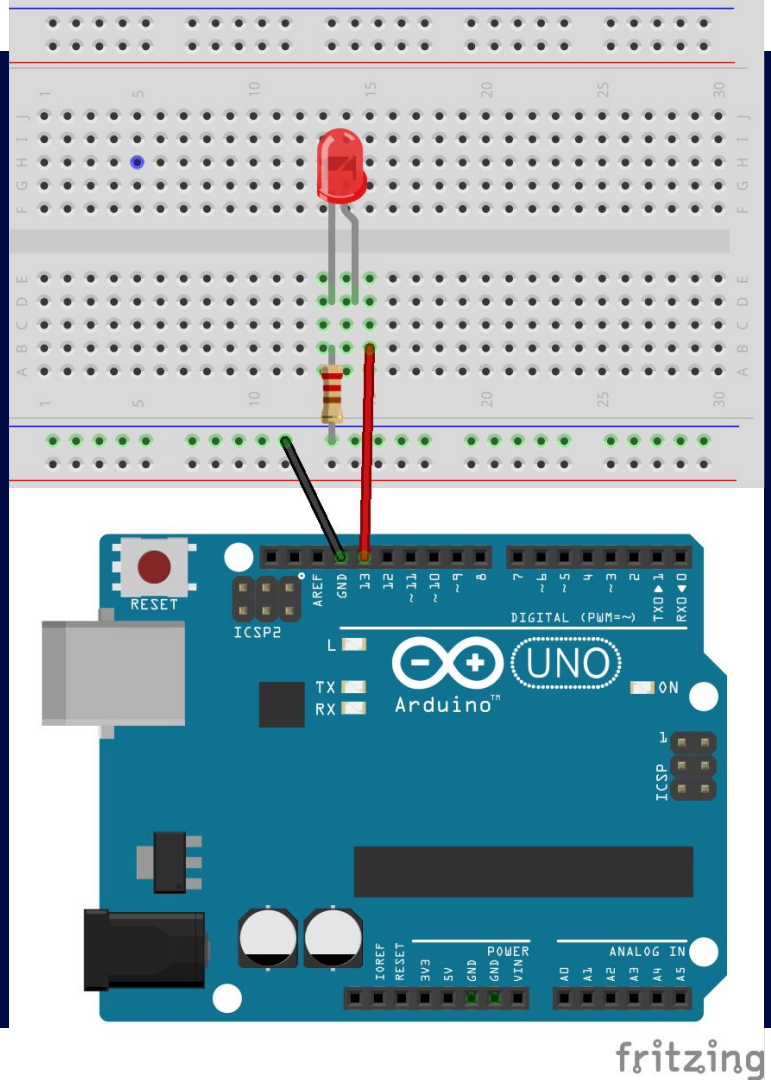



UAVs@Berkeley



Battery or Power Supply


A B C D E
F G H I J
1
5
10
1
5
10
@Berkeley


RESET
ICSP2
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM=~)
L
TX
RX
UNO
Arduino™
ON
ICSP
POWER
ANALOG IN
IOREF
RESET
3V3
5V
GND
GND
VIN
A0
A1
A2
A3
A4
A5

fritzing

```
/*  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the Uno and
  Leonardo, it is attached to digital pin 13. If you're unsure what
  pin the on-board LED is connected to on your Arduino model, check
  the documentation at http://www.arduino.cc
  This example code is in the public domain.
  modified 8 May 2014
  by Scott Fitzgerald
 */
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

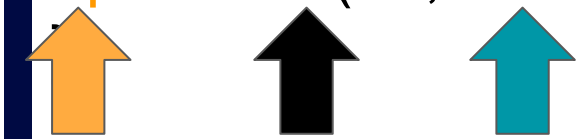
Name of the sketch

```
/*  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.
```

Description of the code

```
  Most Arduinos have an on-board LED you can control. On the Uno and
  Leonardo, it is attached to digital pin 13. If you're unsure what
  pin the on-board LED is connected to on your Arduino model, check
  the documentation at http://www.arduino.cc
  This example code is in the public domain.
  modified 8 May 2014
  by Scott Fitzgerald
*/
```

```
// the setup function runs once when you press reset or
power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}
```



Function  pin #  keyword: INPUT or OUTPUT

```
// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);   // turn the LED on (HIGH is the voltage level)
```

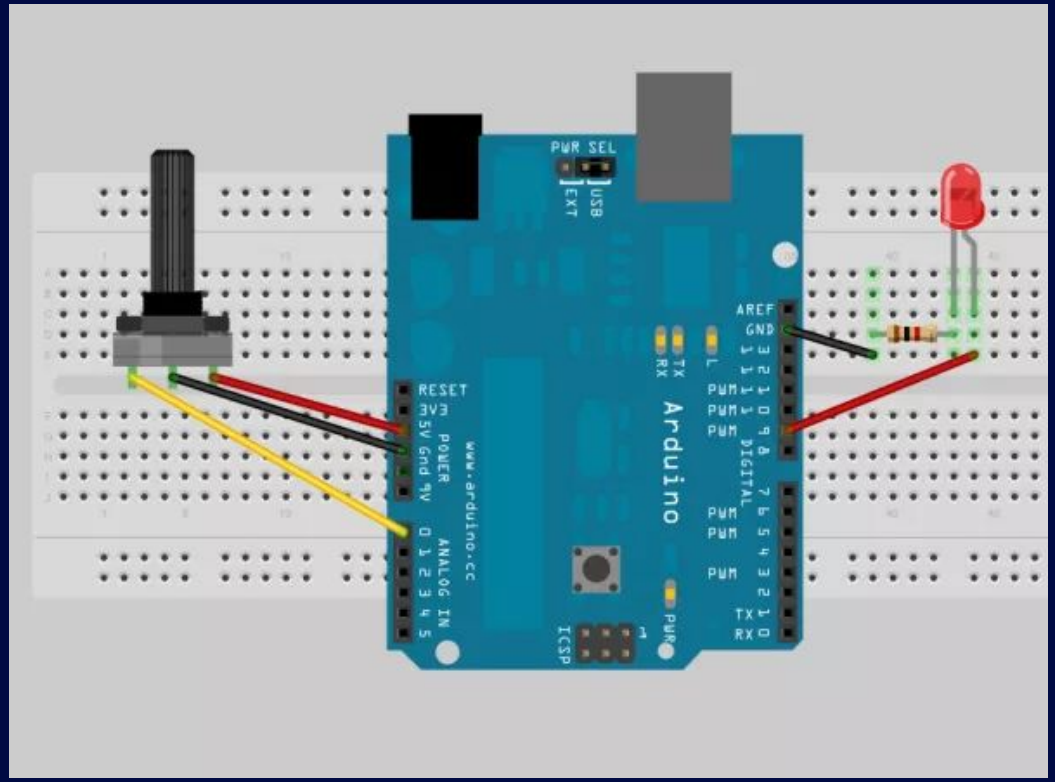
pin # type of voltage: high or low

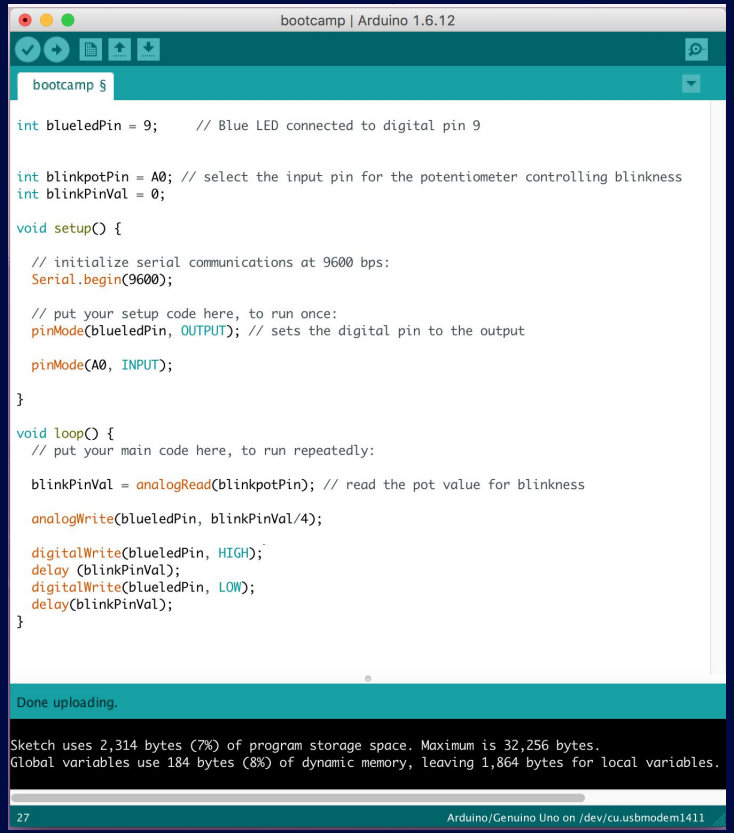
```
  delay(1000);              // wait for a second
```

number of milliseconds

```
  digitalWrite(13, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

# Demo


PWR SEL
USB
EXT
AREF
GND
DIGITAL
TX
RX
Arduino
RESET
3V3
5V
Gnd
Vin
POWER
ANALOG IN
www.arduino.cc
ICSP
PWR


bootcamp | Arduino 1.6.12
bootcamp §
int blueledPin = 9;     // Blue LED connected to digital pin 9
int blinkpotPin = A0; // select the input pin for the potentiometer controlling blinkness
int blinkPinVal = 0;
void setup() {
// initialize serial communications at 9600 bps:
Serial.begin(9600);
// put your setup code here, to run once:
pinMode(blueledPin, OUTPUT); // sets the digital pin to the output
pinMode(A0, INPUT);
}
void loop() {
// put your main code here, to run repeatedly:
blinkPinVal = analogRead(blinkpotPin); // read the pot value for blinkness
analogWrite(blueledPin, blinkPinVal/4);
digitalWrite(blueledPin, HIGH);
delay (blinkPinVal);
digitalWrite(blueledPin, LOW);
delay(blinkPinVal);
}
Done uploading.
Sketch uses 2,314 bytes (7%) of program storage space. Maximum is 32,256 bytes.
Global variables use 184 bytes (8%) of dynamic memory, leaving 1,864 bytes for local variables.
27
Arduino/Genuino Uno on /dev/cu.usbmodem1411

# Step Through This Code

```
const int buzzerPin = 8;                // buzzzer signal pin attached to digital pin 8
const int trigPin = 11;                 // ultrasonic triggerPin connected to digital pin 11
const int echoPin = 10;                 // ultrasonic echoPin connected to digital pin 10
int ledPin = 13;                        // LED positive connecteed to digital pin 13 (no need for resistor)
long duration;
float distanceMeters, distanceCm, distanceInch;
```

```
void setup() {                          // define all digital and analog pins
  pinMode(13, OUTPUT);                  // LED pin as output
  pinMode(trigPin, OUTPUT);             // trigger pin as output because it sends signal
  pinMode(8, OUTPUT);                   // buzzer pin as output
  pinMode(echoPin, INPUT);               // echo pin as input becaue it recievs signals
  Serial.begin(9600);                    // initiate serial communication
}
```

```
void loop() {
  digitalWrite(trigPin, HIGH);               // initiate the ultrasonic sensor for sending and receiving
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);

  distanceMeters = duration * 0.00034 / 2;      // speed of sound per second; divide by 2 due to then sound coming back to the ech
  distanceCm = duration * 0.034 / 2;
  distanceInch = duration * 0.0133 / 2;



  Serial.print("Distance: ");                        // print distance (cm) in serial monitor
  Serial.print(distanceCm);
  Serial.println("  Centimeters");
  delay(300);
  Serial.println("----------------------------");     // seperation
```

```
if (distanceCm <= 20 )    {
  digitalWrite(13, HIGH);
  tone(buzzerPin,1500);
  delay(200);
  digitalWrite(13,LOW);
  noTone(buzzerPin);

}
else if (distanceCm <=60) {
  digitalWrite(13,HIGH);

}
else {
  digitalWrite(13,LOW);
}
```

# Communication & Project Management

(not presented, included for reference)

# Project Management

- “Project management is the discipline of initiating, planning, executing, controlling, and closing the work of a team to achieve specific goals and meet specific success criteria.” - Wikipedia
- We standardize how we communicate, delegate tasks, and track progress so that **expectations are clear, team members are held accountable, and deliverables are met** within the time and financial constraints
- Work in UAVs@Berkeley project teams are highly parallelized so interfacing with your fellow team members needs to be quick, easy, and effective
- This block serves as an introduction to the tools we use at UAVs@B to facilitate project management

# Project Management Tools

Google Drive

Slack

Trello

- Project Outlines
- Design Reviews
- Milestones

# Google Drive

- UC Berkeley students get unlimited Google Drive storage with their Berkeley account
- UAVs@B uses it for file storage including media, active and archived
- Tour of the Drive
- You should be able to follow along and have access to it!

# Slack

- Used for chat, event planning, project communication, and personal messaging within the club
- Understand the channel naming convention
- You should have joined the UAVs@B team when you filled out the General Membership Sign-Up
- We have various apps installed for our team
  - Google Drive
  - Simple Polls
  - Trello
  - Google Calendar
- Tag people/channels, pin items, star channels, initiate a poll, refer to trello cards, share files, create group chats, all within Slack
- Add a photo!

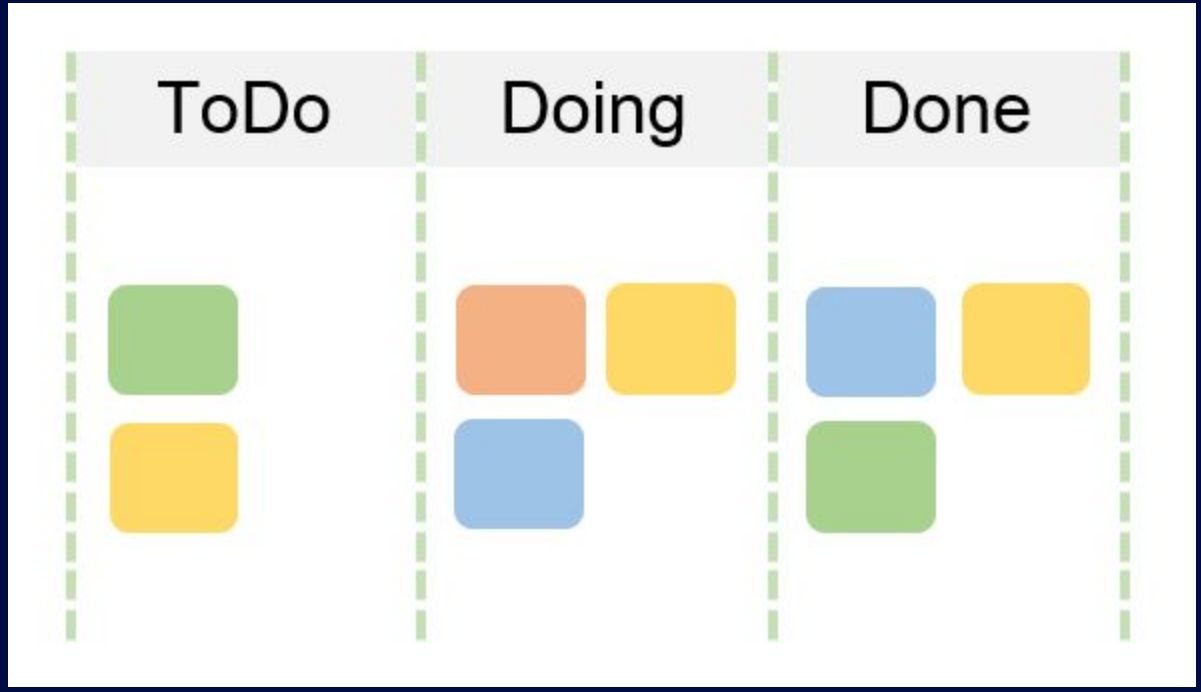
# Trello

- Project management and workflow tracking tool that we use to monitor progress, delegate tasks, and build cool stuff
- How you are given work and report progress within your project team at UAVs@Berkeley
- Meant to be a high detail, task specific tool, **should always be up-to-date**
    - All project team members should know how to use Trello to get assignments and organize work
- An online, collaborative implementation of a “Kanban board”

# Kanban Boards

“The Kanban technique emerged in the late 1940s as Toyota's re-imagined approach to manufacturing and engineering. Line-workers displayed colored kanbans — actual cards — to notify their downstream counterparts that demand existed for parts and assembly work. (Kanban is the Japanese word for "visual signal" or "card.") The system's highly visual nature allowed teams to communicate more easily on what work needed to be done and when. It also standardized cues and refined processes, which helped to reduce waste and maximize value.”

- leankit.com




Basic Kanban Board

# Trello

- A tour of the leadership team's Trello board
- Utilize the full featureset of Trello to maximize transparency and traceability
  - Lists
  - Due dates
  - Labels
  - Member card assignments
  - Descriptions
  - Checklists
  - Attachments
  - Comments
  - Notification subscriptions
  - Slack channel notifications

Make frequent updates, download it on mobile, ensure your notification settings are correct! Add a photo!

# Project Outlines

- Project Outlines completed by your team leaders
  - A “syllabus” for your project, contains a summarized, high level summary of week by week goals
  - Trello cards are typically assigned based on these outlines
  - This is a living document that will update as the semester continues to ensure deliverables are met by the end of the semester
  - You should read yours in your project team’s Google Drive directory now!

# Project Tracking Events

- Design Reviews
  - Project teams present to leadership team in brief design review format every two weeks
- Project Milestones
  - MILESTONE 1: Halfway point
    - Large event presenting project progress, difficulties, design innovations to the rest of the project teams
  - MILESTONE 2: ¾ point
    - Reevaluate approach as semester comes to an end, what will be deliver? No event
  - MILESTONE 3: Project completion
    - Large symposium style event for all project teams including presentations and live demos

# Project Management Summary

Google Drive

Slack

Trello

- Project Outlines
- Design Reviews
- Milestones

# Building Basics

(not presented, included for reference)

# The Soldering Iron & other Soldering Tools


Soldering Station

Solder

# The Soldering Iron & other Soldering Tools

Desoldering Pump


Desoldering Wick

# How to solder safely and efficiently

- Make sure that you minimize the time that the iron is powered
    - It doesn’t take long to heat up, but prolonged heat on the iron can damage it
    - For general soldering, a temp of around 300 degrees Celsius should suffice
- If you want to lessen the chances of burning yourself, use tweezers
- A little solder goes a long way, you can always add more

# Soldering Techniques

UAVs@Berkeley

- “Tin” your surfaces before you solder. This means to add a little bit of solder to the surface (pad, wire, iron) so that the actual solder job doesn’t add too much heat to your components
    - This being said, don’t keep the iron on sensitive components for too long >7 seconds

- Keep the iron tip clean by periodically wiping on a damp sponge or brass wool, the tip should be shiny

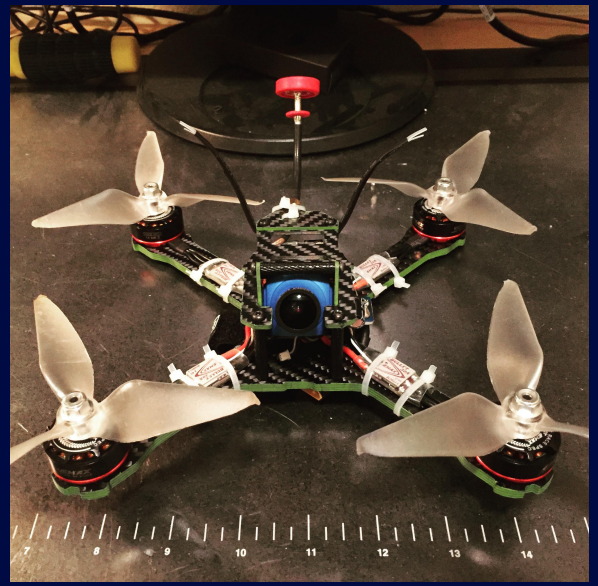
- When soldering add solder to the component, not the iron
- It should not look like this

# Quad Anatomy Review

- Frame
- Motors
- Electronic Speed Controller (ESC)
- Power Distribution Board
- Flight Controller
- Receiver
- Camera & Video Transmitter
- FPV Antenna
- Propellers
- Battery

# The Build

- The actual build process involves incorporating all the previous components in a way that is light, protected, and intuitive
- Builds vary based on purpose of the drone



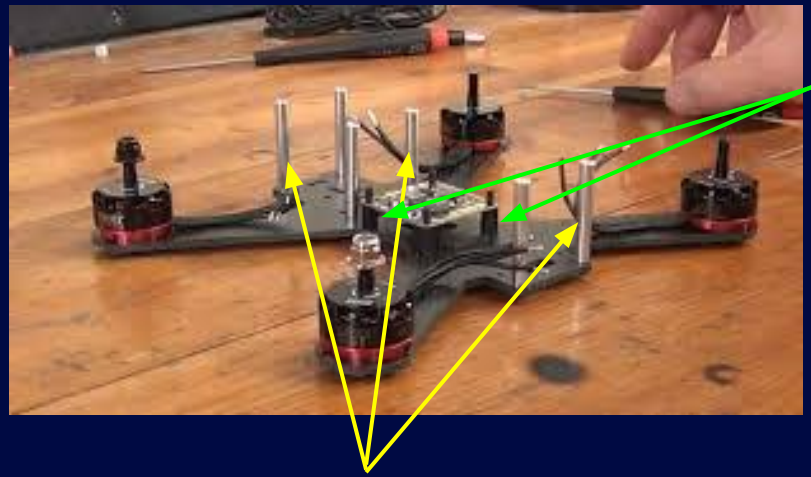
Provides protection and insulation for wires


Different wire thicknesses (gauge) can handle different amounts of current

# Heatshrink and Wires

# Standoffs and Metric Fasteners

These provide the vertical structure for the drone's frame also hold components tightly to it





Metric bolts holds the frame together and components onto the drone

# Adhesives and Waterproofing

~You don't really use too many adhesives during the build as components are bolted on~

- Threadlocker
- Hot Glue
- Double Sided Tape
- Electrical Tape
- Liquid Electrical Tape

To waterproof components use Silicone Conformal Coating, but be careful not to accidentally gum up any physical buttons on your components