STAGE ONE EDUCATION

HELICOPTER

ELECTRONICS AND CODING

WORKSHOP

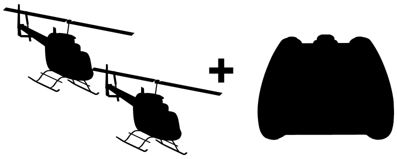
with

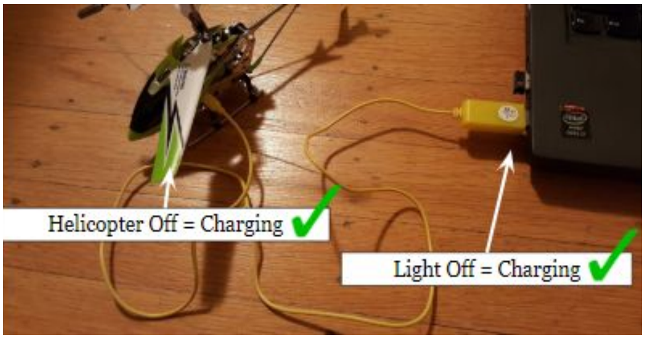
ARDUINO

Step 1: Set Up

1 Make sure your laptops turn on and are charging

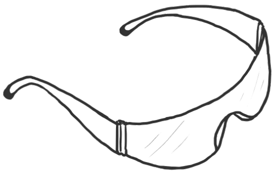
2 Plug the helicopters into the USB cables to start charging!



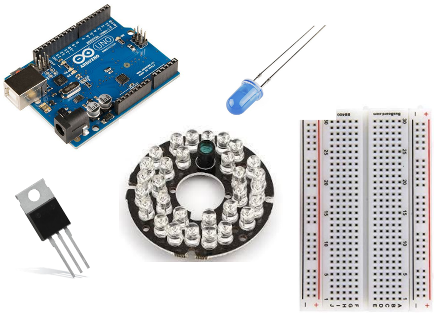


The helicopter should be off, and if it is charging, the light on the USB will be off as well!

3 EVERYONE get a pair of safety glasses



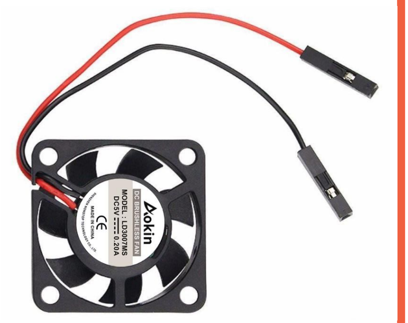
4 Make sure that you have <u>ALL</u> of the parts listed on the next page - and know their names!



Parts that We'll Use Today

IR Light Ring

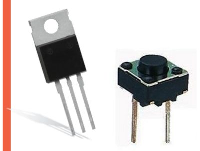
Fan



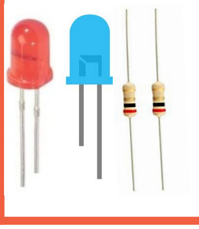
Helicopter Charging Cables (2)

"Knob"

Transistor & Button

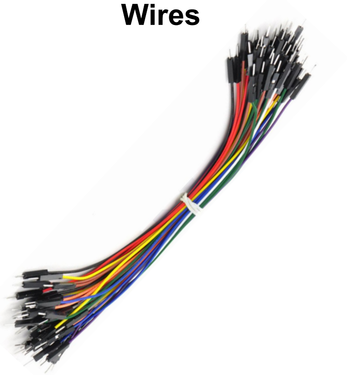


LEDs & Resistors



Battery Pack

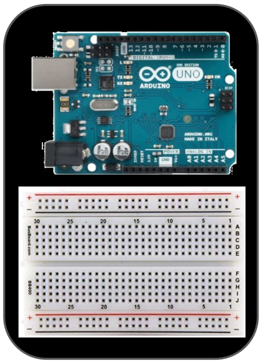
Blue, Green, & Orange Wires



Laptop

Helicopter Remote

Arduino & Breadboard



USB Cable

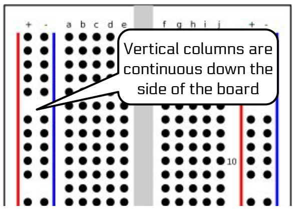
Helicopters (2)

Safety Glasses
(1 pair per person)

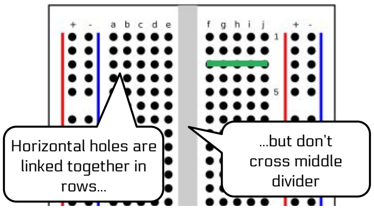
Next, we'll build our first circuit!

BREADBOARDS

NEVER twist wires. **ALWAYS** connect wires using the breadboard!



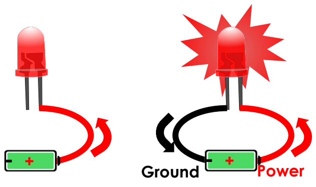
Vertical columns are continuous down the side of the board



Horizontal holes are linked together in rows...

...but don't cross middle divider

Good to Know

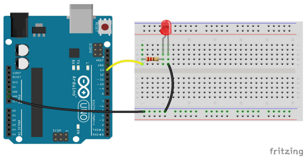


open circuit
=
no light

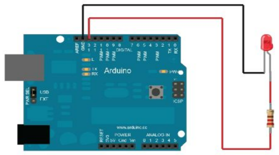
closed circuit
=
LIGHT!

In a **circuit**, voltage moves in a **circle** from a power source back to ground..

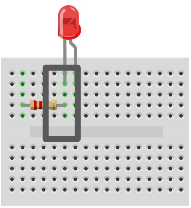
We'll start off with diagrams like this to show you how to set up your circuit:



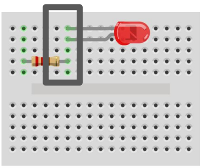
...but we'll move on to circuit diagrams, like the one below:



Diagnosing Problems

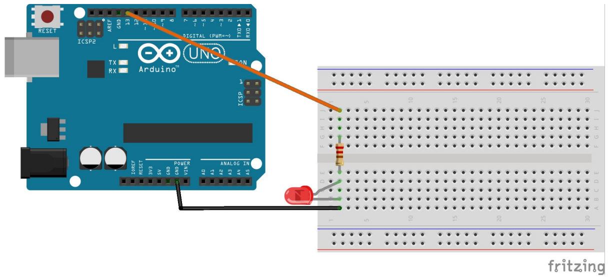


OK

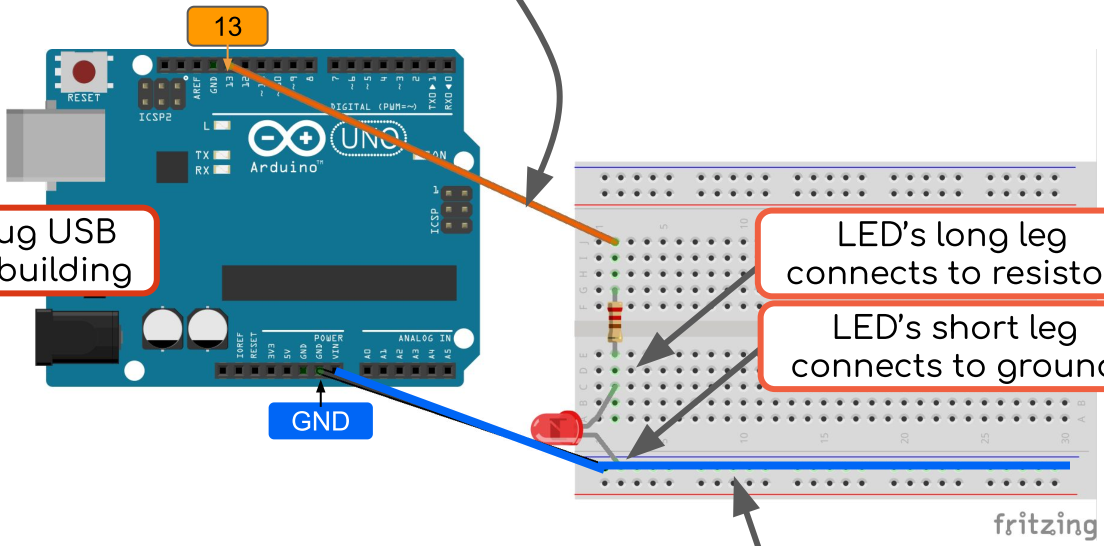


wrong: LED shorted out

what's wrong with the breadboard below?



Step 2: Our First Circuit: “Blink”



Let’s use an orange wire to link pin 13 of the Arduino to the resistor.

unplug USB while building

LED’s long leg connects to resistor

LED’s short leg connects to ground

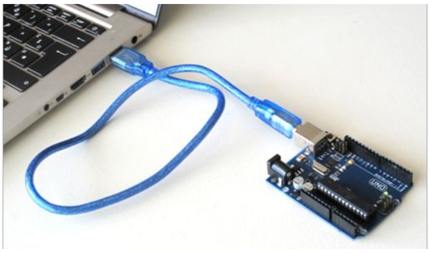
Ground (GND) wires are best to connect on one of the long blue edges of the breadboard. We will use blue wires to match (or black if available**).

LED color will not matter for this circuit.

**Wire color doesn’t affect how the circuit works, but it *does* affect how well you can understand what you just made! Black, and sometimes blue, are often associated with “ground” in circuits.

Step 3: Upload Your Code!

1

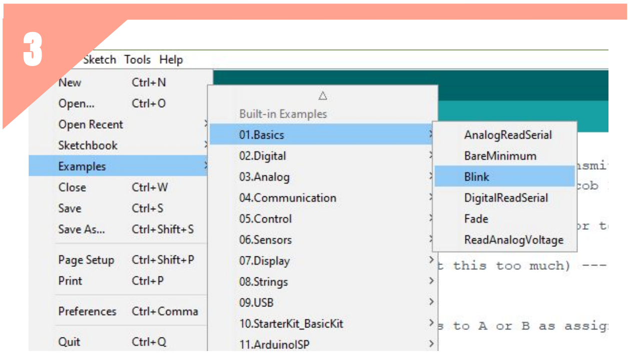


Connect the Arduino board to the laptop via USB

2

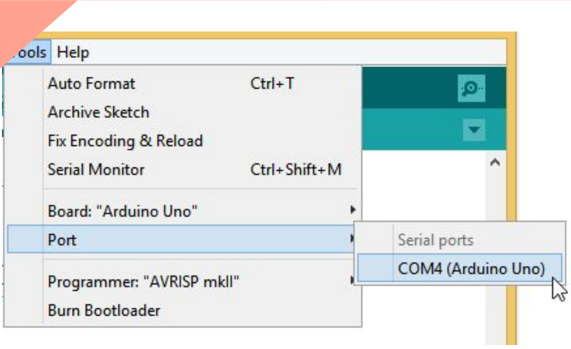
Open the Arduino app on your laptop

3



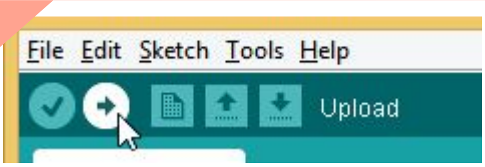
Open File->Examples->Basics->Blink

4



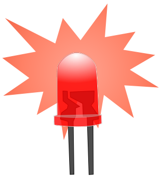
Select the Port for the Arduino (Tools -> Port -> COM)

5



Upload the example code

6



Is it blinking?!? If so, it worked!

Step 4: Basic Code Changes

Change your blink code delays so the light is on for 0.250 seconds and off for 0.750 seconds:

milliseconds on (high) → delay(250);

milliseconds off (low) → delay(750);

```
digitalWrite(LED_BUILTIN, HIGH);
delay(250);
digitalWrite(LED_BUILTIN, LOW);
delay(750);
```

Don’t forget to upload after every code change!

Step 5: Code Change Challenges

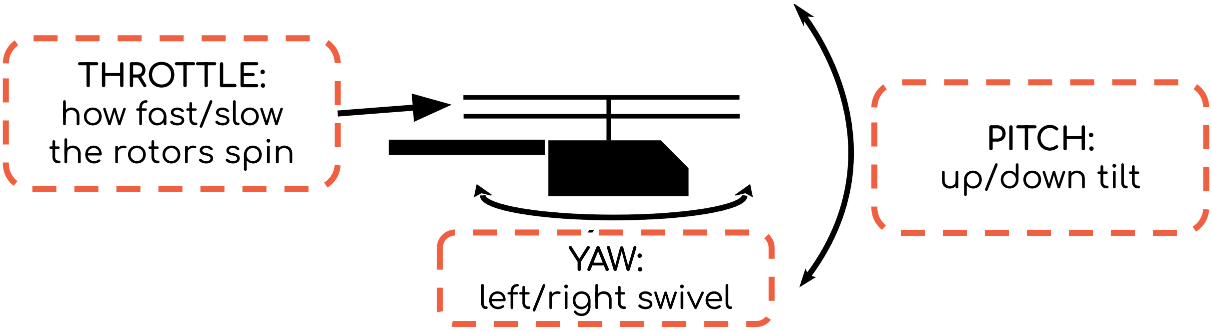
- □ change the delay numbers so that the light stays on for half a second and off for half a second (1 second = 1000 milliseconds)
- □ change the delays so that they are so short, you can’t even tell that the light is blinking anymore – show an instructor!
- □ make the light stay on all of the time (100%!)

Step 6: Get Ready To FLY!

EVERYONE:
Tie up long hair
Put on safety
glasses.

Turn on ONE of
your helicopters
(keep the other
one charging)

Get your remote from
your instructor.
Make sure your
channel is the one
assigned to you.



THROTTLE:
how fast/slow
the rotors spin

PITCH:
up/down tilt

YAW:
left/right swivel

Step 7: Take Your 3-Minute Flight!

❏ Learn the controls for THROTTLE (speed), YAW (turn), and PITCH (tilt) on the remote.

❏ Can you make the helicopter hover? what do you have to do for this to happen?

❏ **EASY:** make a straight line out and back

❏ **MEDIUM:** fly in a perfect square

❏ **HARD:** make the shape of a letter

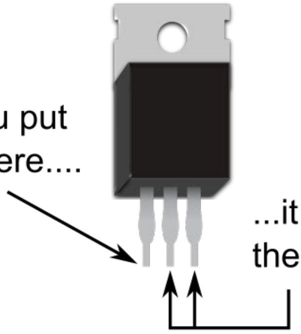
❏ **IMPOSSIBLE:** have a partner tell you how to move (foward, back, etc.) in order to spell a word that they DON'T tell you up front!

❏ Trade to the next group member after 3 minutes!

High Voltage Motors

THINGS TO KNOW:

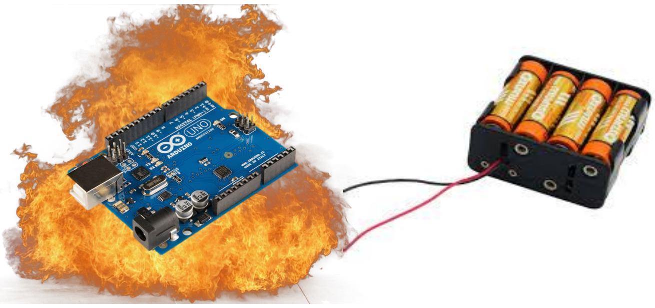
How does our transistor work?



When you put voltage here....

...it connects these two pins

In other words, it's an electronic switch.

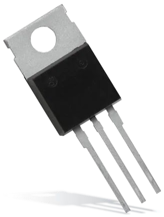


Without the switch, we have a problem.

our transistor is what allows us to control a fan using a 12V battery pack using a 5V Arduino...

...otherwise a high voltage power source would fry a low voltage circuit board

In this step, you'll need:



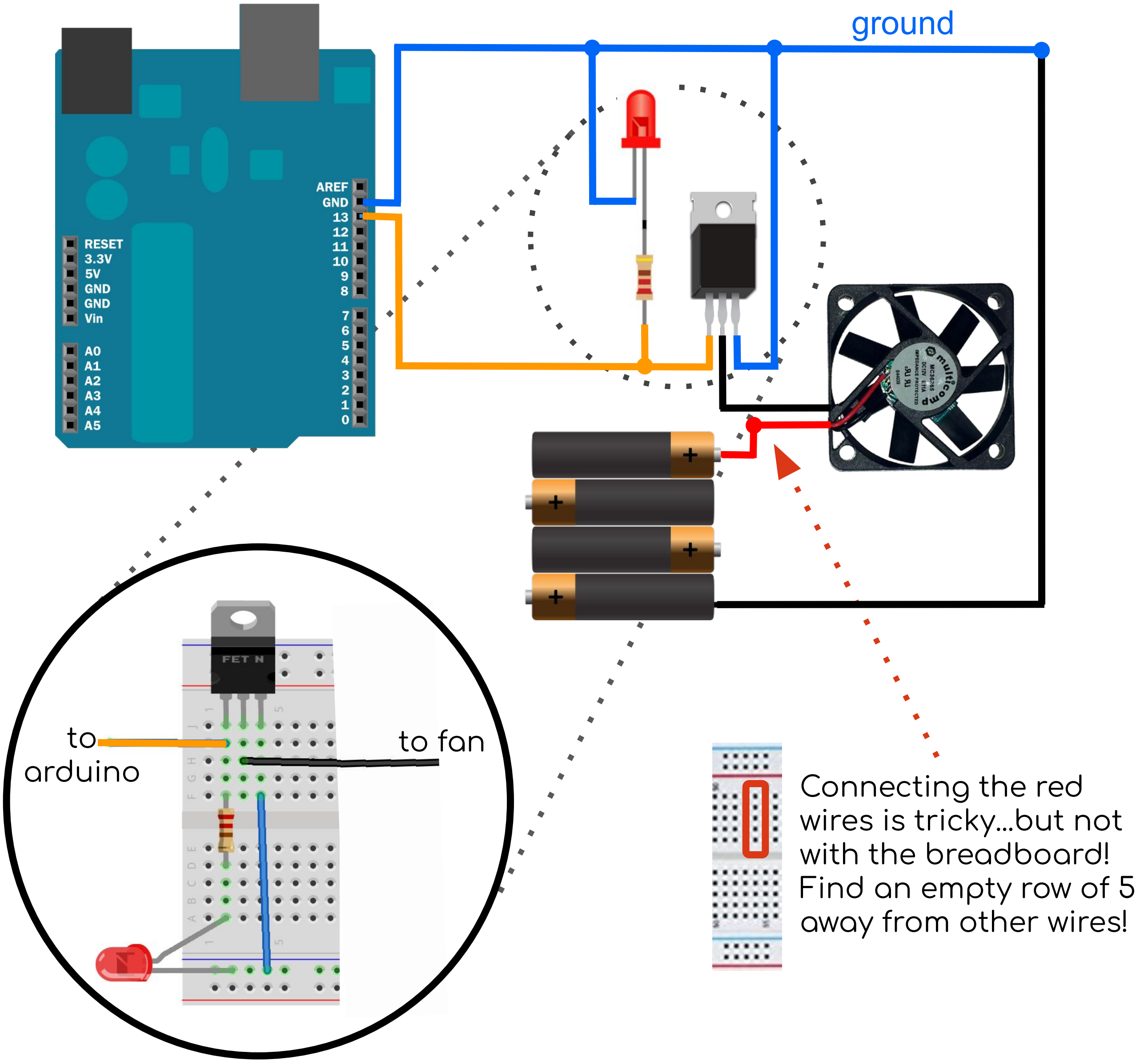
1 transistor

battery pack
with 7 batteries

1 fan

High Voltage Motors

Step 8: Build The Fan Circuilt



Connecting the red wires is tricky...but not with the breadboard! Find an empty row of 5 away from other wires!

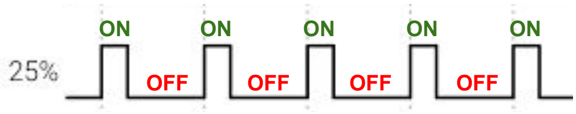
get your circuit checked to get your last battery!

Step 9: Modify the Blink Code

THINGS TO KNOW:

It’s all about ratios!

We are turning the motor on and off.
We can change how long it stays on or off by setting an amount in the *delay.*



Just like we did earlier, change your blink code delays to match the description above, this time to achieve “25% power” for the fan.

milliseconds on (high) → delay(250);

milliseconds off (low) → delay(750);

```
digitalWrite(LED_BUILTIN, HIGH);
delay(250);
digitalWrite(LED_BUILTIN, LOW);
delay(750);
```

Step 10: Blink/Fan Code Challenges

- □ make the fan run at full power all of the time (100%!)
- □ make the fan run at half speed (half on, half off)
- □ can you make it go ¾ speed? 1/10 speed?
- □ can you make it run more smoothly? At how small of a of delay does the fan simply not run?

Input Control

THINGS TO KNOW:

Control the circuit directly

So far, everything we built was pre-programmed, not something that could respond directly to human input.

In this section, we will add a button as direct input that can be used to tell the program to take an action.

Initially, we will make it control the fan. Later, we will use it to start the helicopter control code.

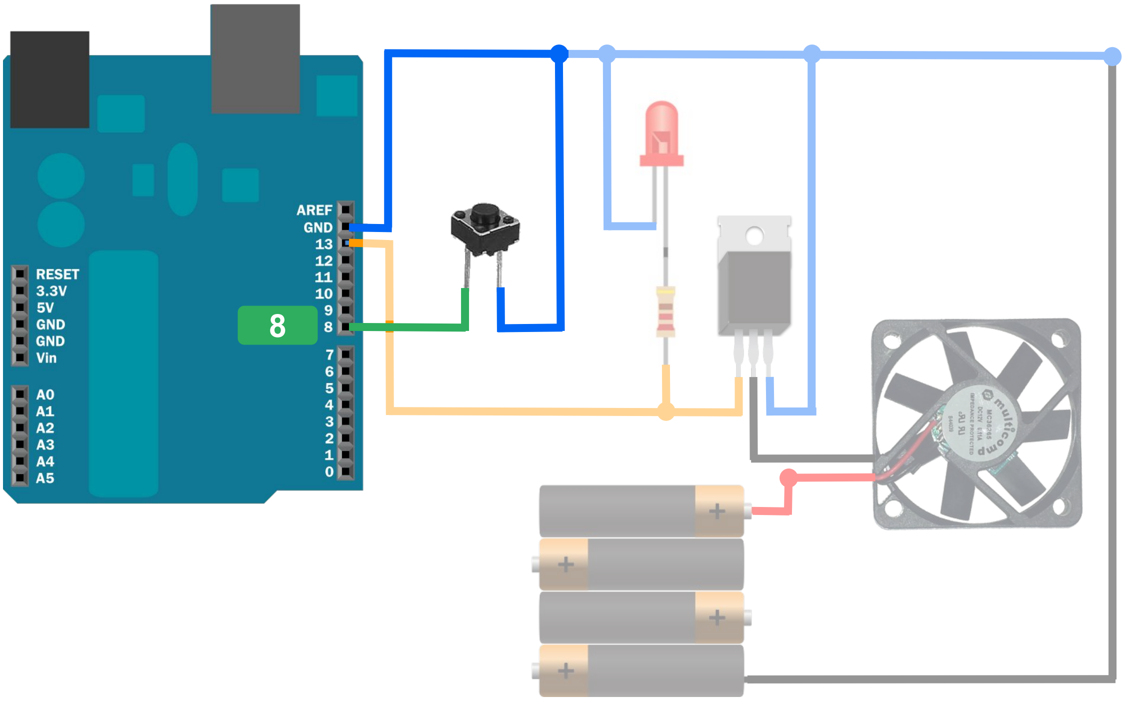
We also include an optional analog input to make fine adjustments possible for the helicopter control code without having to re-write the code.

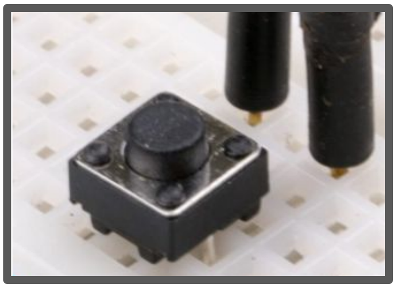
Input Control

Step 11: Add a Button

One pin of the button connects to Arduino pin #8 – use a green wire

The other pin connects to ground – use a black or blue wire





Remember that ALL connections go through the breadboard – don't twist wires directly!

Input Control

Step 12: Add to Setup Code

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(8, INPUT_PULLUP);
}
```

Add this line to make pin 8 an input that starts HIGH, but gets pulled LOW when pushed in.

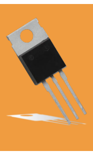
Step 13: Modify Loop Code

```
void loop() {
  if (digitalRead(8) == LOW) {
    digitalWrite(LED_BUILTIN, HIGH);
  } else {
  ~~delay(250);~~
    digitalWrite(LED_BUILTIN, LOW);
  }
  ~~delay(750);~~
}
```

These 3 lines form our **conditional logic**. We start by checking if the button is pushed (if the signal to pin 8 is LOW, then the button must be connecting the circuit's LOW (ground) to the pin 8 input). If it is, then we turn on the fan. If it is *not* low (*else*), then we turn the fan off.

Remove the delays -- we will now manually control on and off with the button, not timing.

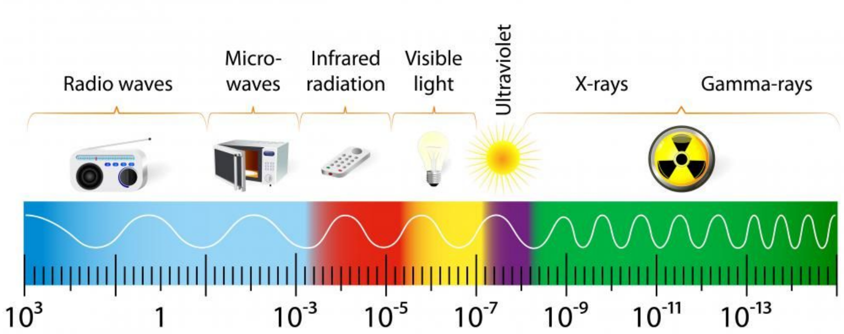
High Voltage Signal



THINGS TO KNOW:

Why are we using IR? What is IR?

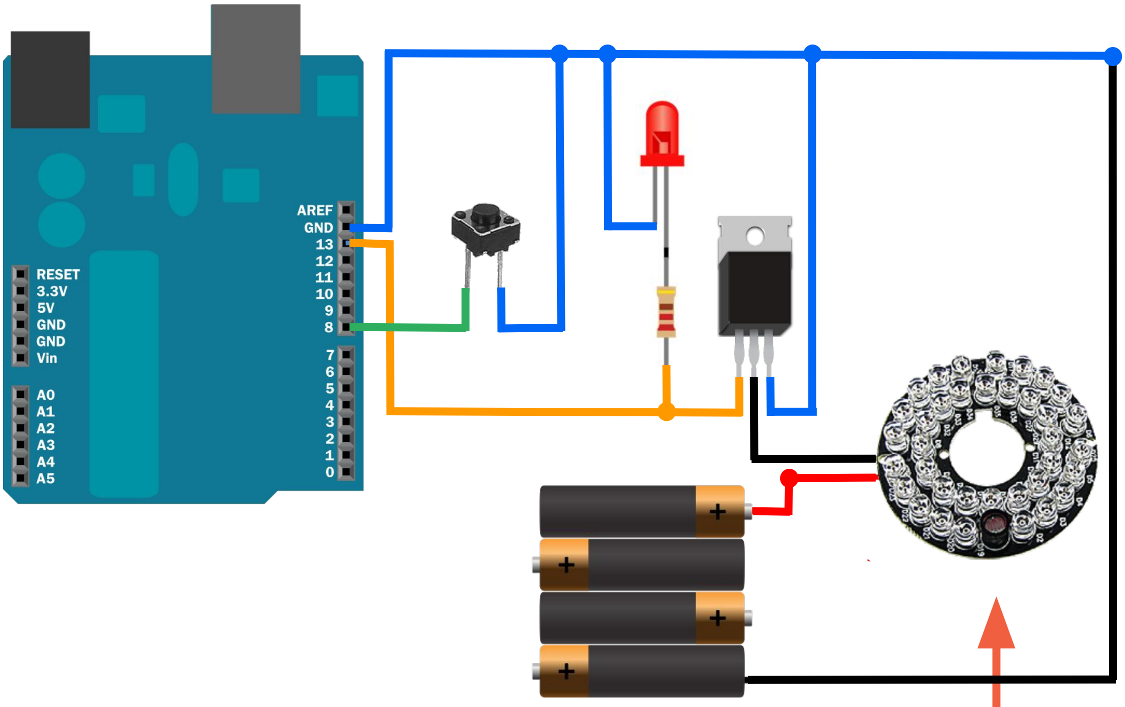
Infrared, or IR, is a specific wavelength of light (~1 mm- 700nm). It's mostly invisible to the human eye but does come up against the very edge of the red spectrum!



IR light can be used to transmit signals to an IR receiver by sending pulses of IR light. Notice how the helicopter controller has an IR light at the front! It also has an IR receiver at the bottom/rear. This is the same technology used for most TV remotes at home.

Step 14: Swap In the IR Array

<u>Do NOT take apart your circuit!!</u>
Remove the fan only.



Do a direct swap of our fan for the IR LED array.
Don't change anything else - it's that easy!

Testing

Use the code from the previous stage to test out the IR LED array. It should blink!

Note that "IR" stands for "infrared." This is just on the edge of the visible spectrum, so you can only see the lights if you shield out other light and look directly at it. you can also see the light with some older smartphones (sorry, not most iPhones though!)

Keyboard Flight!

Step 15: Prep and Upload the Code

READY

- ❏ Open the helicopter folder on the desktop
- ❏ Double-click the file inside

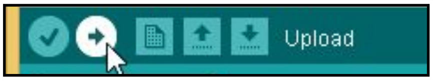
SET

In the code, modify line 1 to match your assigned brand and channel::

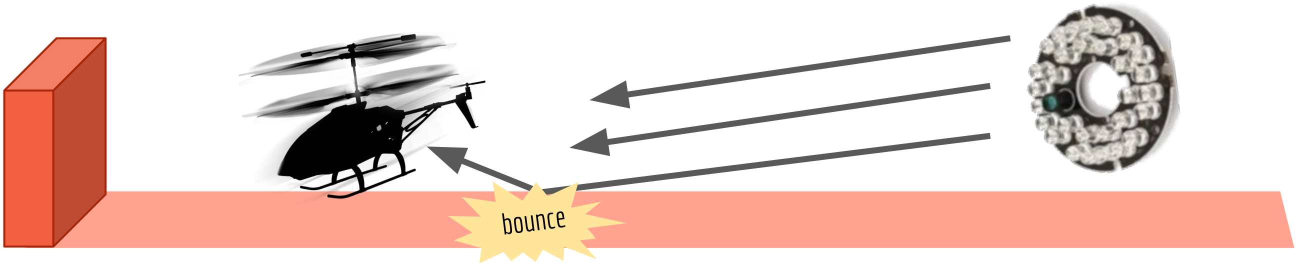
```
byte channel = 5;
```

GO

Upload the code without any other changes to the Arduino -- it is ready to go!



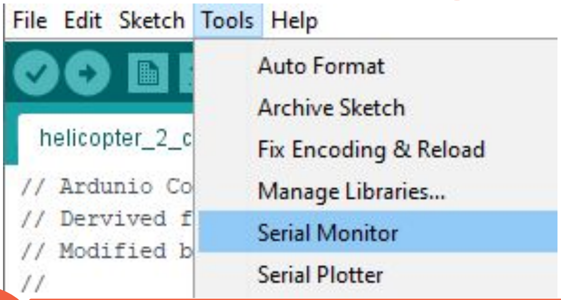
Step 16: Sync the Helicopter



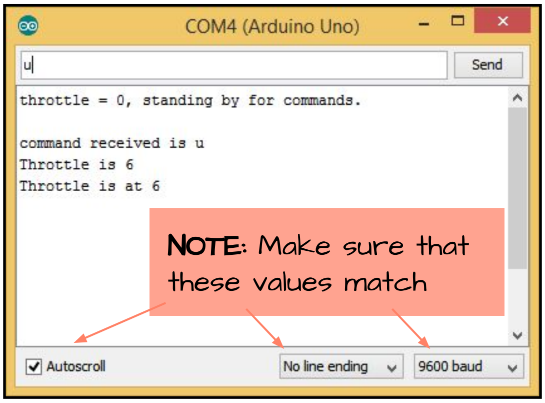
1. Fly pointing your remote at the wall, NOT the center of the room!

2. Point the IR LED ring at the bottom of the helicopter to maintain signal.

3. Turn the helicopter on and set it down. It should sync with your channel



4. Click tools → serial monitor and the window to the right should pop up!



```
throttle = 0, standing by for commands.

command received is u
Throttle is 6
Throttle is at 6
```

NOTE: Make sure that these values match

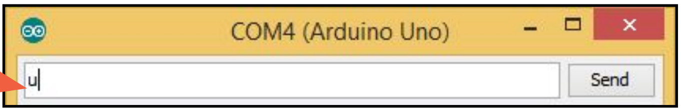
TROUBLESHOOTING

- You know you're synced when the helicopter lights blink with a new pattern.
- Point at the floor under the helicopter to bounce to the sensor beneath.
- If syncing seems impossible, ask for a remote on the correct channel.
- If your helicopter is flying, but not using your code, someone else may have accidentally synced up with it. Turn it off and re-sync.
- Did you lose signal? Make sure to keep the IR LED ring pointed at it!

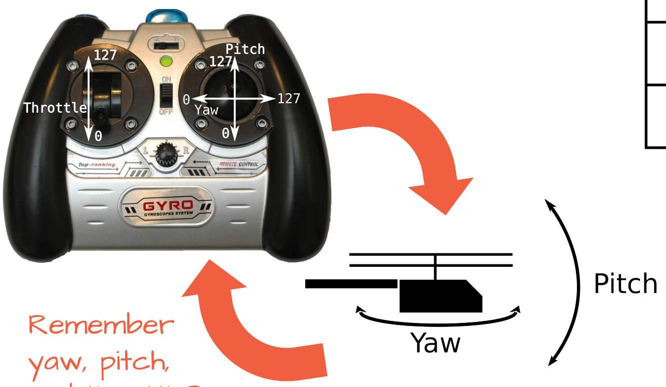
Step 17: Everybody Flies!

In the text box on the serial monitor, enter one letter at a time

NOTE: You must push "Enter" after EVERY character!



5	start at medium throttle		
u	increase throttle	**j**	decrease throttle
w	forwards	**s**	backwards
a	rotate left	**d**	rotate right
0	shutdown	**r**	recenter



Remember yaw, pitch, and throttle?

- **When in doubt, type 0 then Enter**
- Some helicopters may NOT respond to signals until the throttle gets to 30+.
- High is usually around 60, but depends on the battery

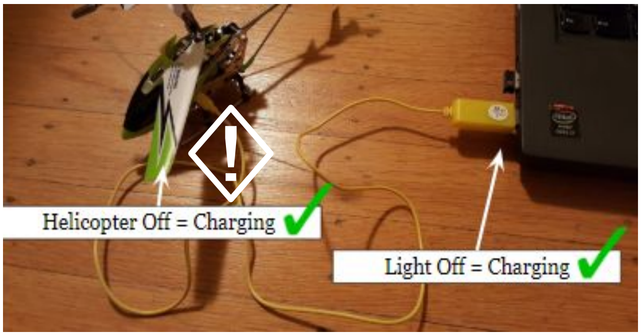
Step 18: Flight Challenges

- ❑ **BASIC:** can you make the helicopter hover? what do you have to do for this to happen?
- ❑ **EASY:** make a straight line out and back
- ❑ **MEDIUM:** fly in a perfect square
- ❑ **HARD:** make the shape of a letter
- ❑ **IMPOSSIBLE:** have a partner tell you how to move (foward, back, etc.) in order to spell a secret word

AUTONOMOUS FLIGHT

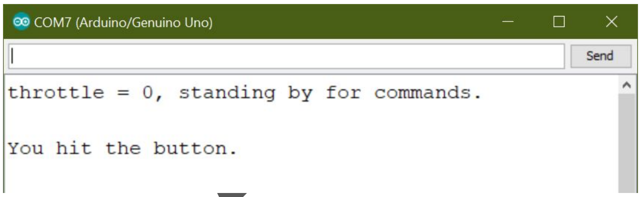
code your helicopter to fly in designed paths on its own!

Are your helicopters charging?



Step 19: Test the Button

1. Get a **new group member** to complete the final step.
2. Test by pushing the button and looking in the serial monitor



```
throttle = 0, standing by for commands.

You hit the button.
```

It should say "you hit the button"

Debugging:

if it says it repeatedly without a new button press, check that the two button pins are on separate rows

if it doesn't work at all, try other commands in the serial prompter (like '**u**') and verify that the correct COM port is selected

CODE

OUR FIRST AUTONOMOUS TAKEOFF

THINGS TO KNOW:

Leave setup/loop alone. This code task is DIFFERENT from previous ones! We will only change a small section.

Remember: Ctrl + / or // to comment/uncomment

Wear eye protection and don't stand over your helicopter.

Step 20: Code Autonomous Flight

PREP

1. Scroll down a short way near line 13-16 inside of the void ButtonPressed() function.
2. Review the example lines for the four `HoldCommands()`
3. Upload the code to the Arduino.

FLY

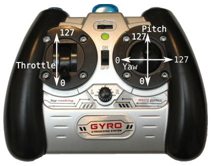
4. Turn on your helicopter to be ready to fly.
5. Press the button! It should take off quickly for a moment, hover, then gently land (FYI - there is no "stop" button!).
    a. You need to maintain signal -- assign one person in the group at a time to point the infrared LED ring at the helicopter.

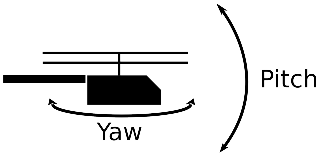
CUSTOMIZE

6. Now, start designing your own commands and flight patterns and try the challenges on the next page! To do this, **you only need to add, remove, and modify HoldCommand() lines. Don't add other code to the file as this will likely cause it to stop working.**
7. With each change of code, simply upload, push the button, observe, improve the code, and repeat!

```
HoldCommand(yaw, pitch, throttle, time in milliseconds)
```

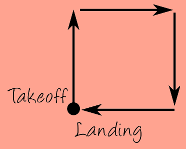
Yaw: `0 - 127,` **`63`** is neutral
Pitch: `0 - 127,` **`63`** is neutral
Throttle: `0 - 127,` **`127`** is slam into ceiling speed





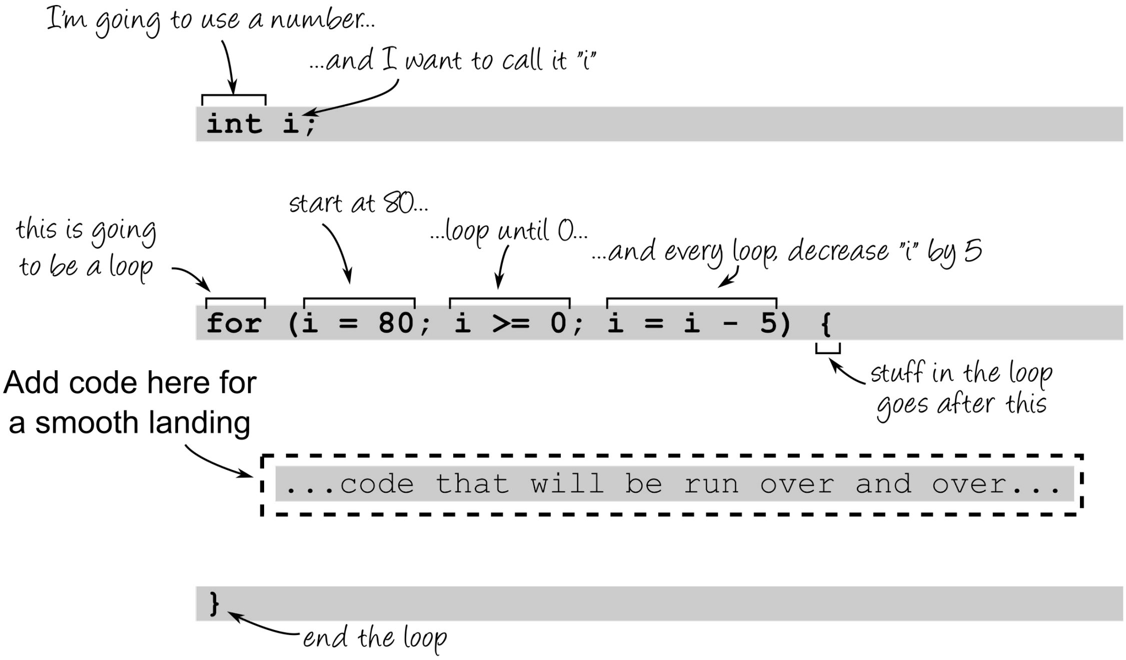
Step 21: Invent YOUR Autonomous Flight!

- ❏ can you make the helicopter hover? what do you have to do for this to happen?
- ❏ **EASY:** make a straight line out and back
- ❏ **MEDIUM:** fly in a perfect square (forward, right,
- ❏ forward, right...)
- ❏ **HARD:** make the shape of a letter (bird's eye view or vertical)
- ❏ **IMPOSSIBLE:** spell out a word or make a shape



OPTIONAL: AUTONOMOUS CODE LOOPS

A smooth landing with a `for` loop (real helicopters slowly spin down their propellers) **type** the code **in, add the "//" back on the previous lines:**

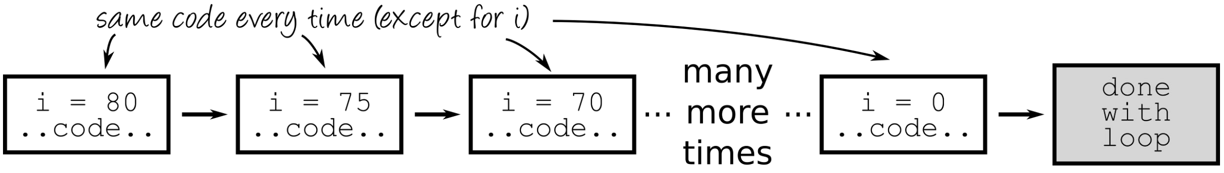


```
int i;

for (i = 80; i >= 0; i = i - 5) {

    ...code that will be run over and over...

}
```

I'm going to use a number...
...and I want to call it "i"
this is going to be a loop
start at 80...
...loop until 0...
...and every loop, decrease "i" by 5
stuff in the loop goes after this
end the loop

Add code here for a smooth landing

how to think about a for loop:



IMPROVE THE SYSTEM:

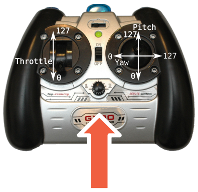
add a throttle trim knob

THINGS TO KNOW:

What does a knob DO?

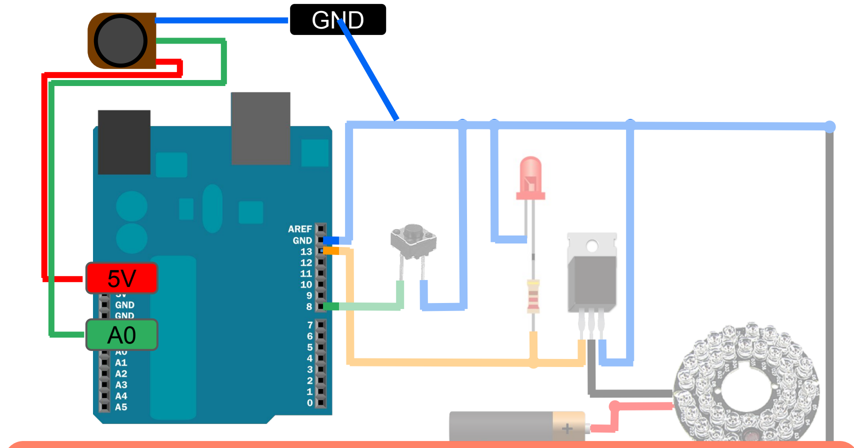
A knob is a kind of **potentiometer**. It takes in 0 volts and 5 volts, and as you turn it, it adjusts the output between these two voltages.

This is just like the trim knobs on the remote, but we are using it to adjust the throttle, giving the helicopter an extra boost of power as the battery slowly dies.



ADD TO YOUR CIRCUIT

ADD a knob to your circuit!



REMEMBER: all ground wires should go to the BLUE row at the side of the breadboard!

Update your code to recognize the input from the trim knob on the next page

IMPROVE THE SYSTEM:

add a throttle trim knob

Update your code to recognize the input from the trim knob:

First, declare the analog input (A0) as an input.
This line goes in the setup() function. However, since this is such a large program, you will need to search (Ctrl+F or Command+F) for **void setup**.

```
pinMode(A0, INPUT);
```

Insert something similar to the code below right above the HoldCommand lines:

```
int raw_knob = analogRead(A0);
int scaled_knob = raw_knob / 103;
Serial.print("Knob value: ");
Serial.println(scaled_knob);
```

Read the knob's input, a number between 0-1023

Shrink the range of values

Display the number in the serial monitor so you can debug.

Now use the knob value in your HoldCommand to tweak throttle without re-uploading your code!

Note: The value will update once at the time you push the start button on your breadboard (when the code above runs), NOT continuously during the program

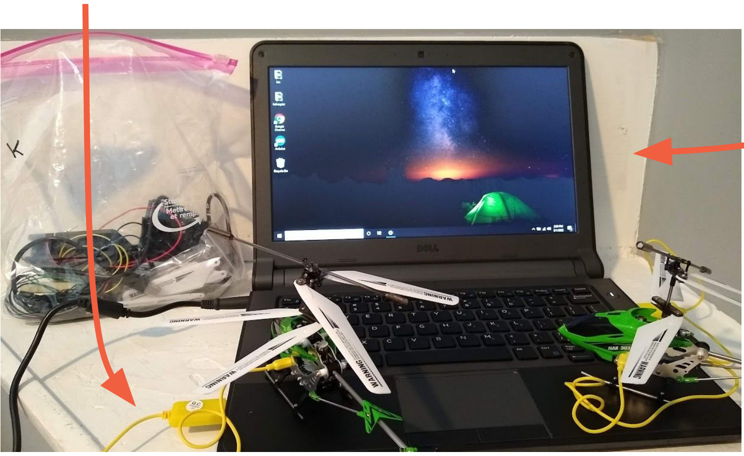
```
HoldCommand(63, 63, 60+scaled_knob, 2000);
```

EXIT CHECKLIST

Please leave your space ready for the next group!

1 **Plug in both helicopters to charge from your laptop.**
Make sure that all lights are off!

2 **Laptop is open, all programs are closed.**



3 PACK UP YOUR PARTS

- Use the chart from the beginning to check to make sure that everything is there.
- **Be sure to turn one battery in separate from the bag** - it should only have 7 batteries in it!
- **Ask your instructor for replacement parts** if any of the parts are broken/missing.

EACH person in your group needs to complete the workshop survey.
Keep it on your phone so that your teacher can see it!

tinyurl.com/arduinohelicopter

Parts Cleanup

Pack the plastic bin as shown below. Other parts stay loose.

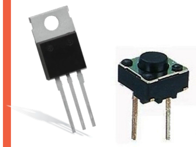
IR Light Ring

Fan

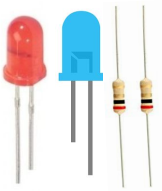
Helicopter Charging Cables (2)

“Knob”

Transistor & Button



LEDs & Resistors



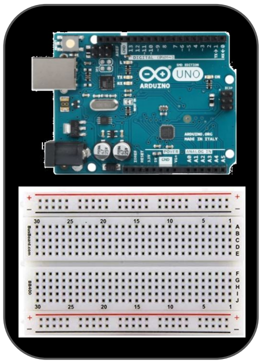
Battery Pack

Blue, Green, & Orange Wires

Laptop

Helicopter Remote

Arduino & Breadboard



USB Cable

Helicopters (2)

Safety Glasses (1 pair per person)