

**HECK YEA, IT'S
CCLAB!**

Oh, hello
Arduino

Arduino

WHAT IS IT?

Arduino

- Arduino is a single-board microcontroller.
- It was designed to make the application of interactive objects or environments more accessible.
- The hardware consists of an open-source hardware board designed around an 8-bit Atmel AVR microcontroller, or a 32-bit Atmel ARM.

Arduino

Long Story Short

- microcontroller
- open source
- accessible
- rapid prototyping

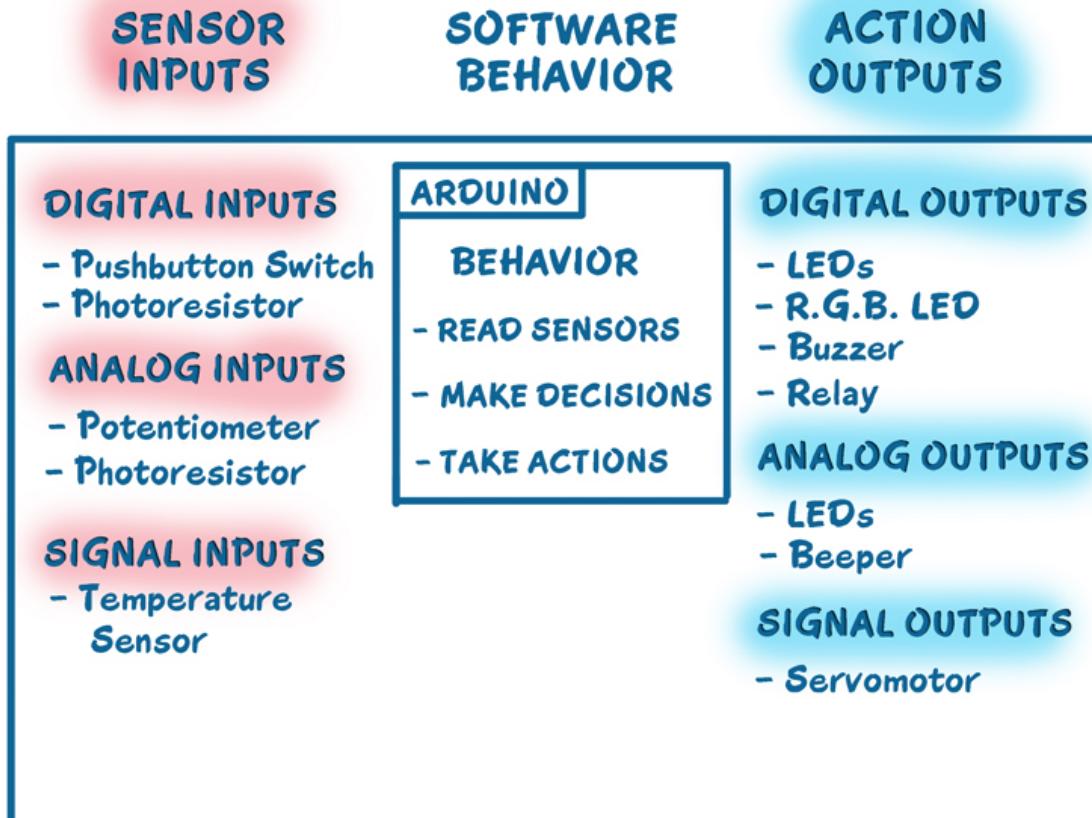
Arduino

input - output machine

Arduino

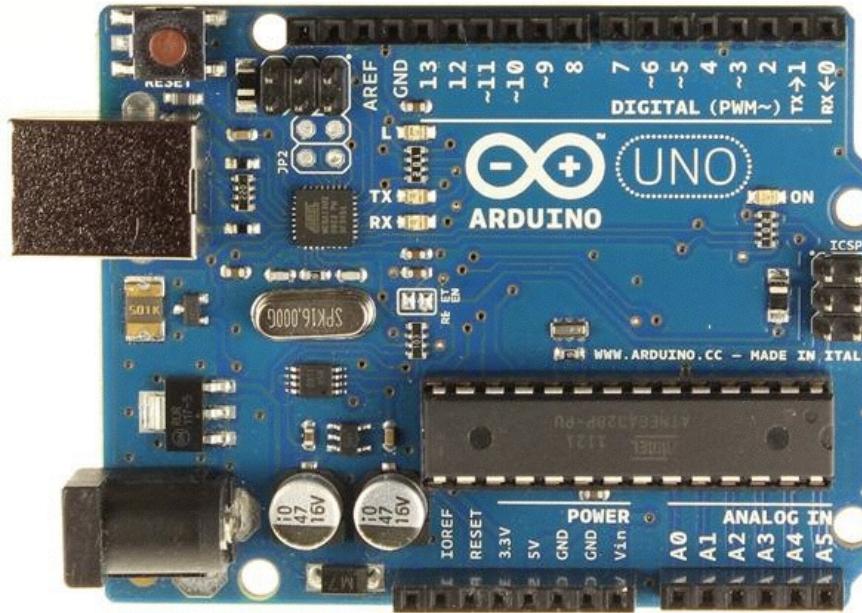
WHAT DOES IT DO...?

Arduino



Arduino

Know Your Board



Arduino

the brain ATmega 328p chip



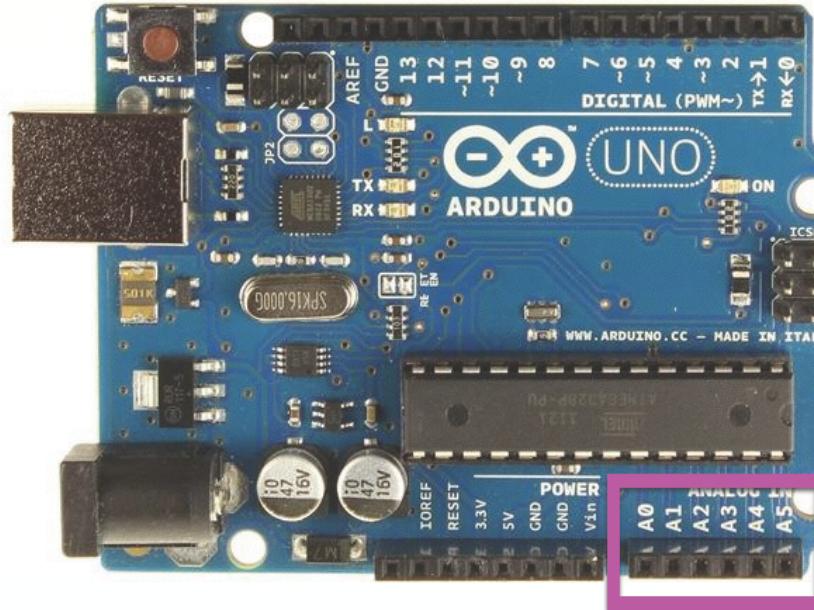
Arduino

digital pins



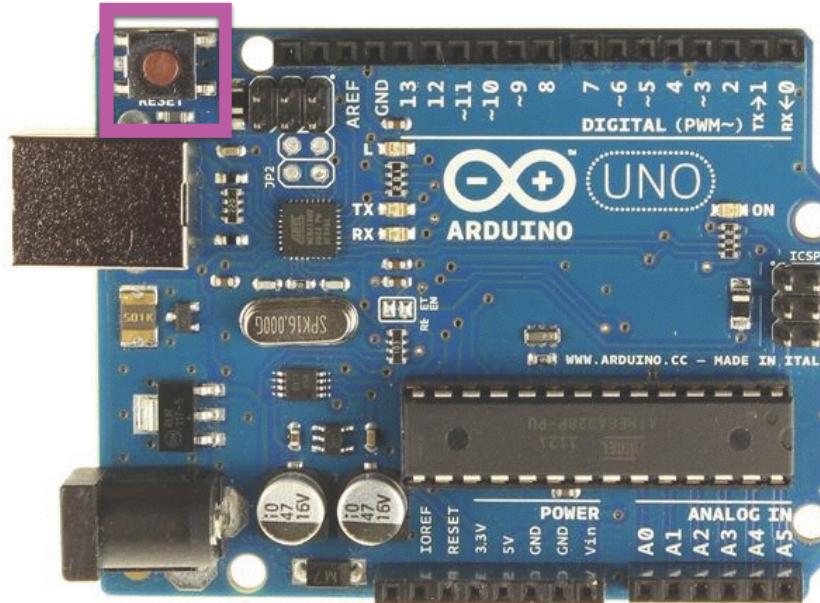
Arduino

analog pins



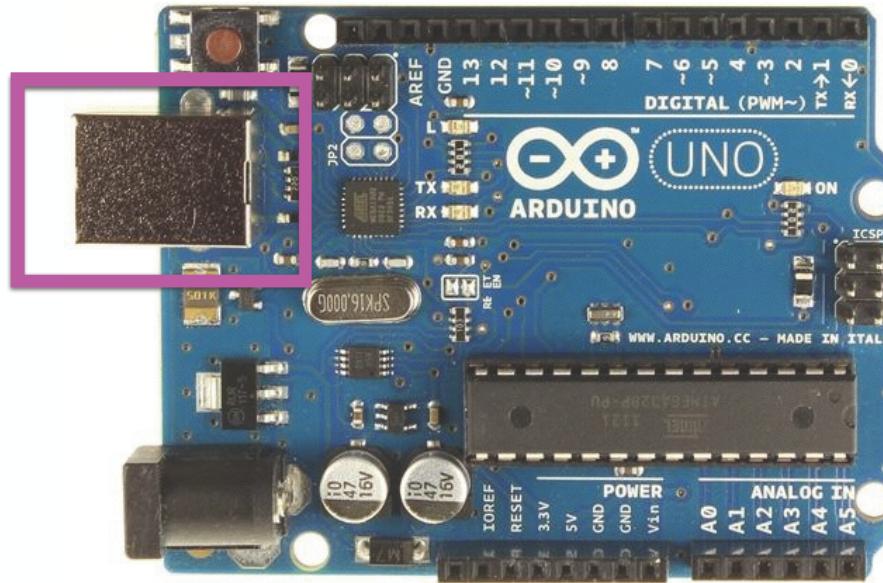
Arduino

reset button



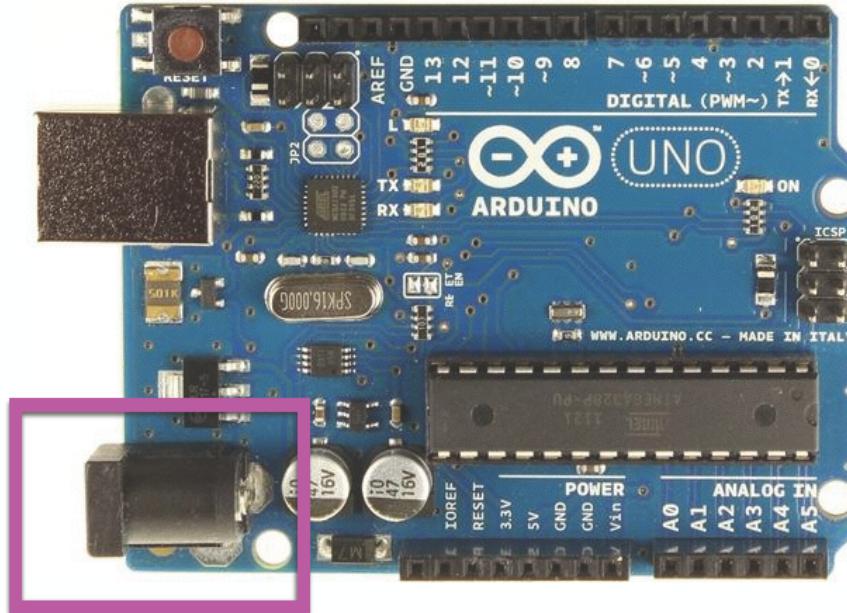
Arduino

usb port



Arduino

power jack



Arduino

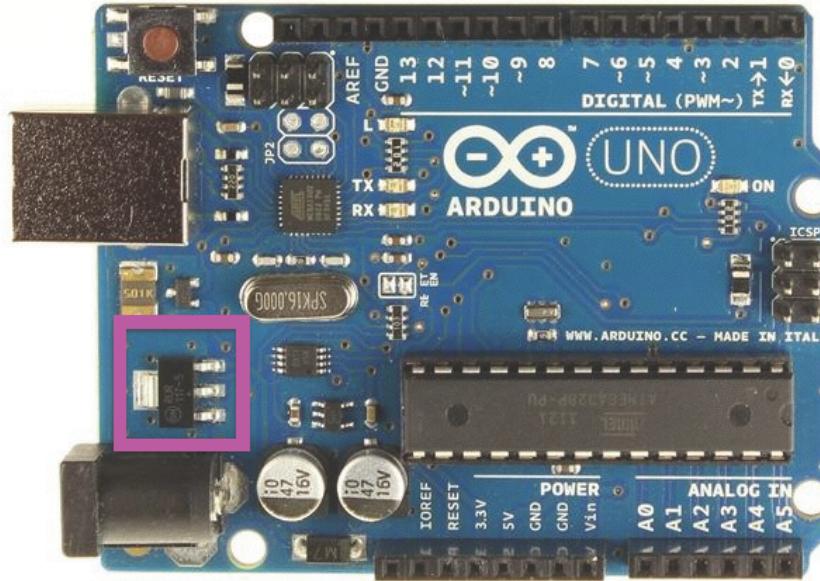
pro tip: buy this



(9 Volt Wall Adapter)

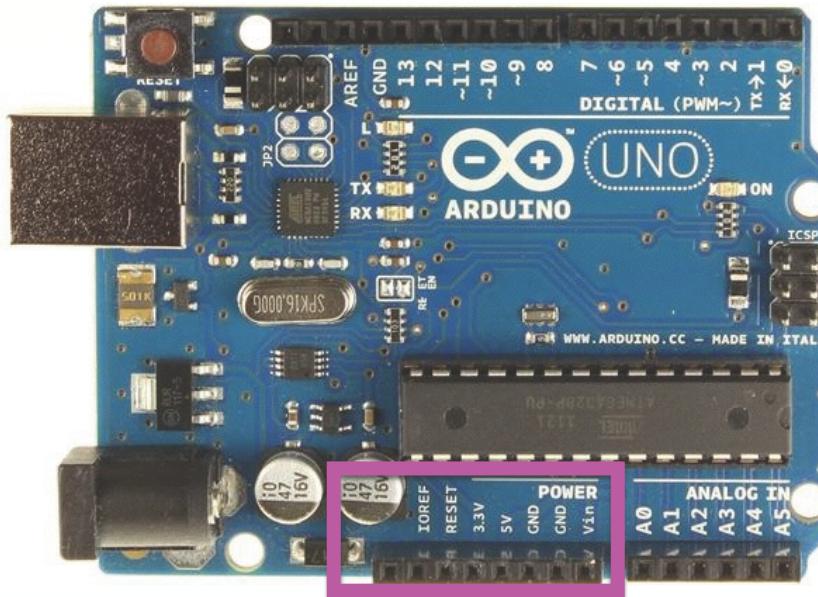
Arduino

voltage regulator

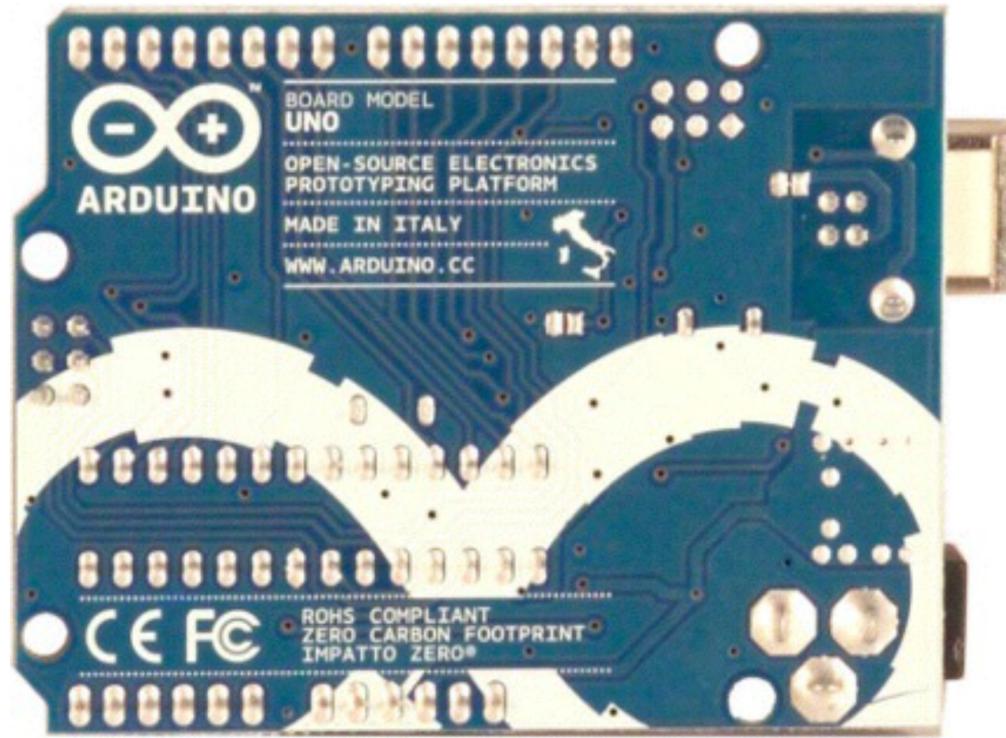


Arduino

Vin, 5V, 3.3V, GND + Reset Pins

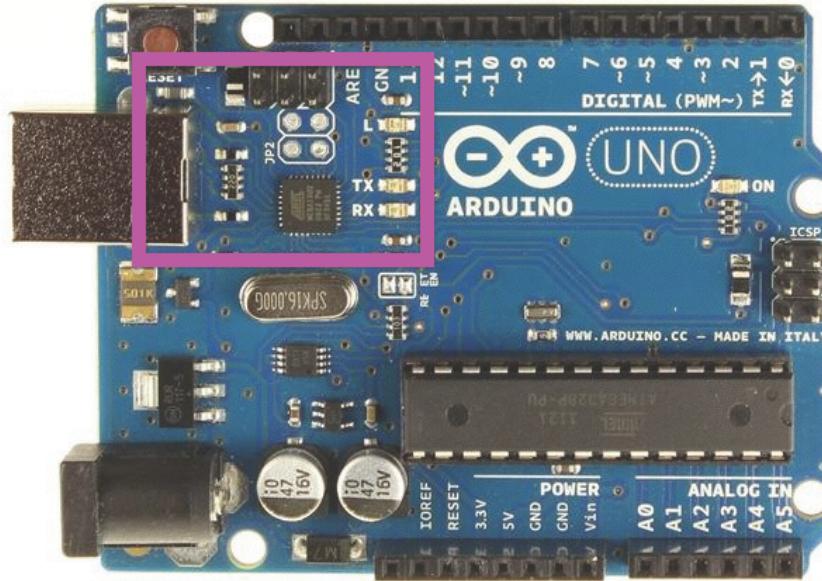


Arduino

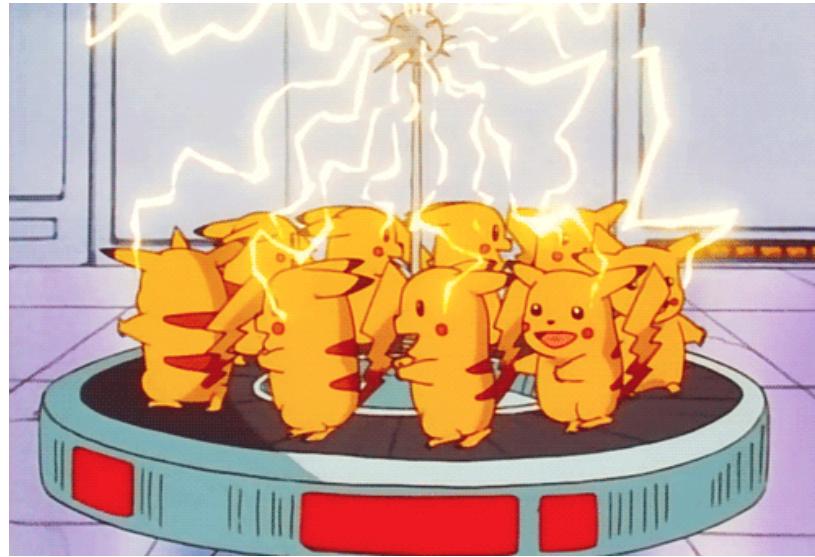


Arduino

internal LED



ELECTRICITY

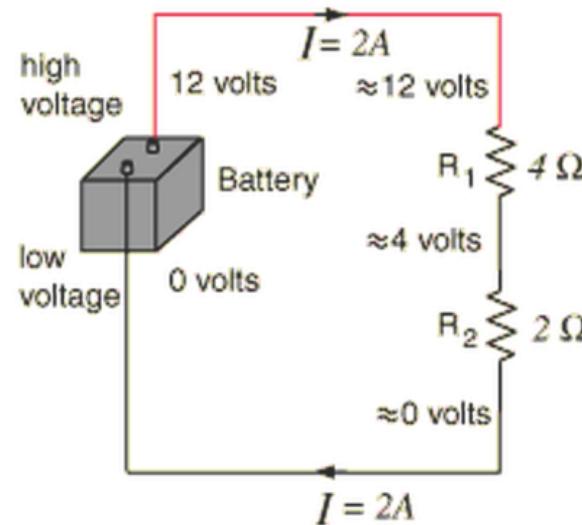
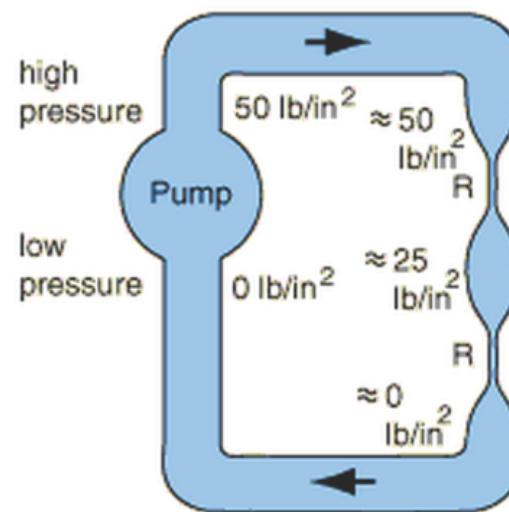


Electricity

**WHAT'S HAPPENING
IN ALL THAT P-COMP?**

Electricity

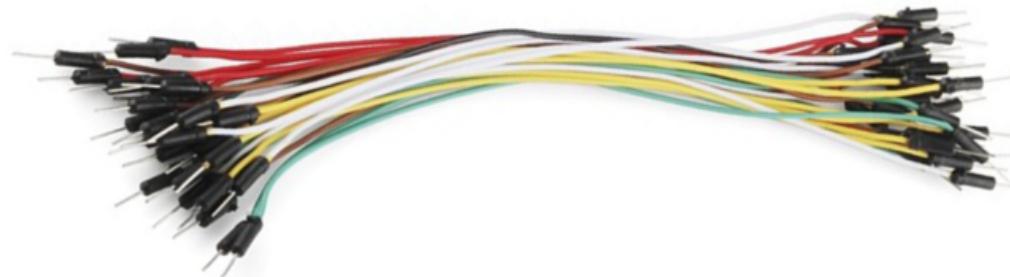
electrical engineering is kinda like plumbing



Electricity

WIRES!

think of them like pipes



Electricity

BATTERY!

It's the “pump” in our system



Electricity

Voltage (V)

The **force** of which electrons are being pushed through the wire.

in other words, “the water pressure”

Electricity

Current (I)

The **amount of electrons** moving through
the wire at any given moment

in other words, “how much water is in the pipes”

Electricity

Think of how a water gun works



current vs voltage

Electricity

Resistance (Ω)

The **opposition** to the passage
of electrons through a wire.

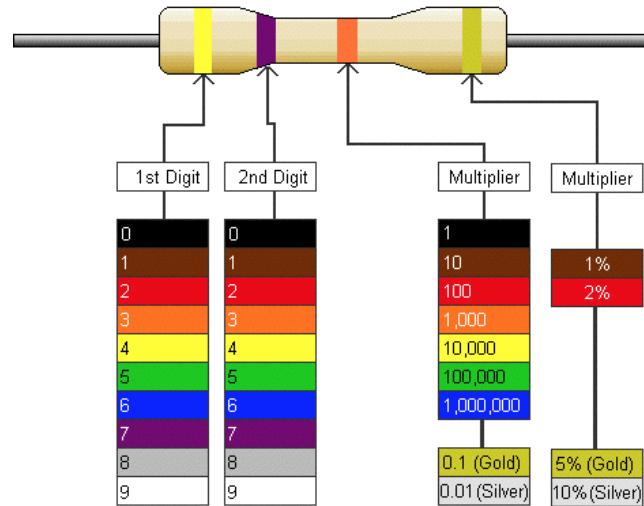
in other words, “how pinched is the pipe”

Electricity



Electricity

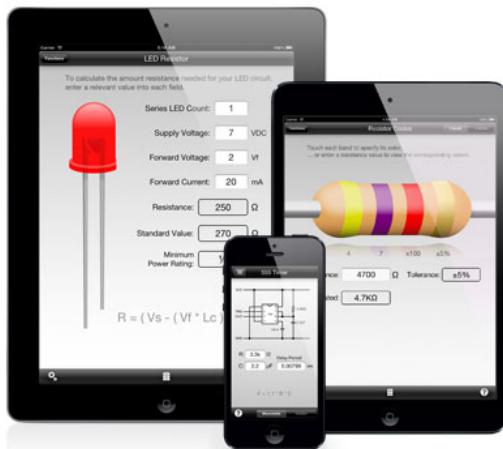
RESISTORS!



the colors aren't just to make them pretty

Electricity

This stuff is confusing but there's tools to help



CIRCUIT PLAYGROUND

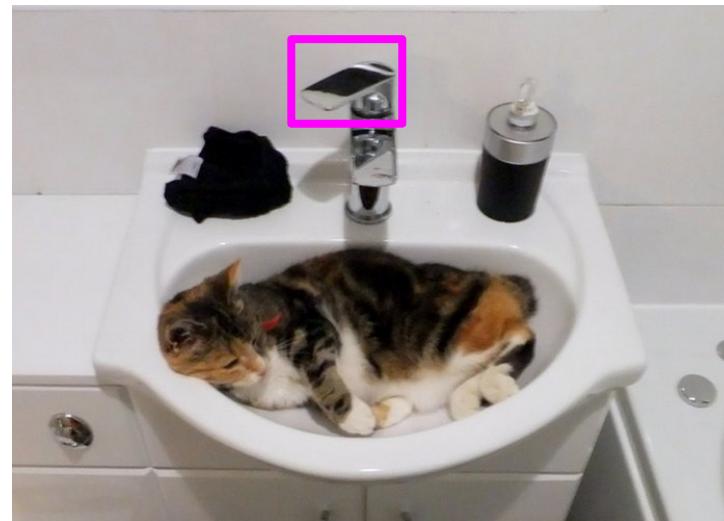


Electricity

IN CONCLUSION:

Electricity

Input



the handle

Electricity

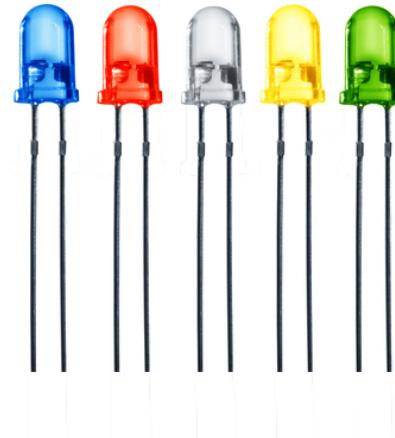
Output



the Tap

Electricity

LED's



Everyone's favorite output

Electricity

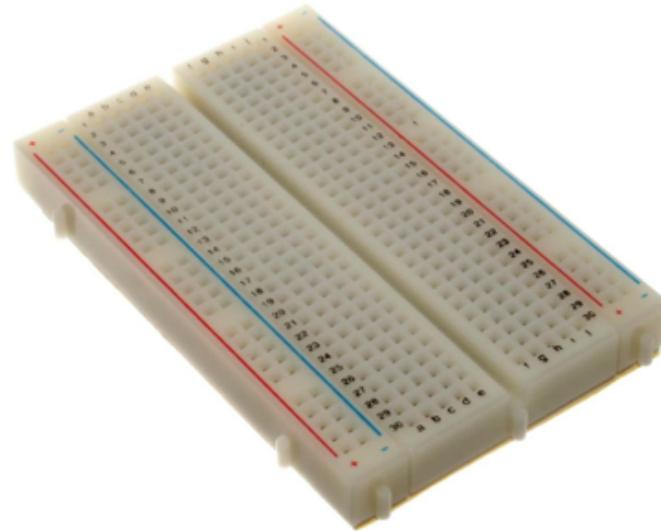
BTW, Breadboards



sadly they have nothing to do with bread

Electricity

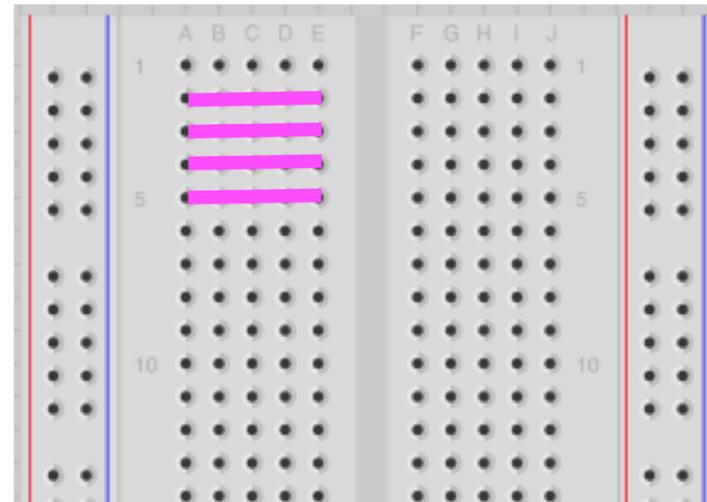
They look like this:



*note the lack of bread

Electricity

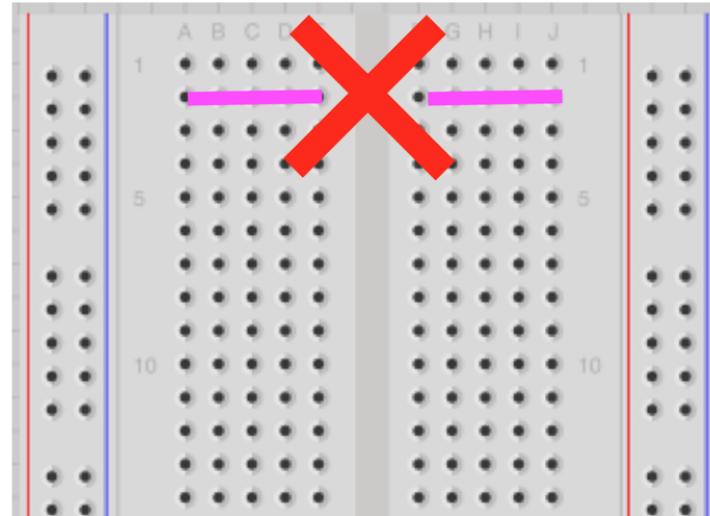
Plug in wires



to make circuits

Electricity

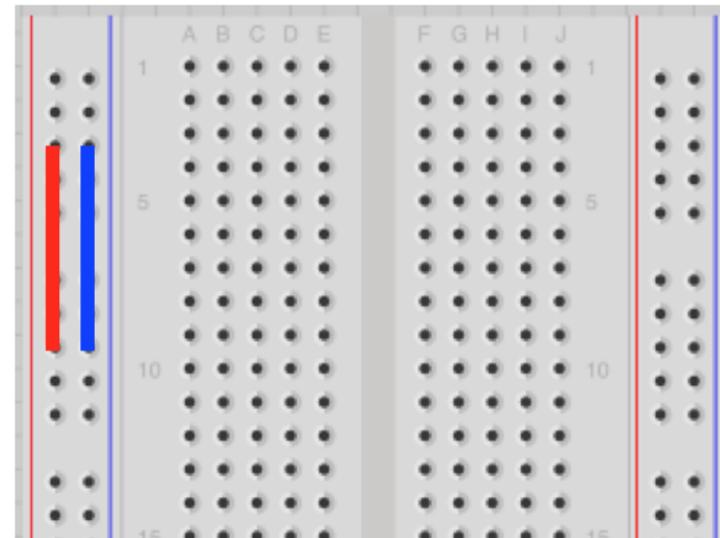
Rows connect inside the board



but don't cross the middle

Electricity

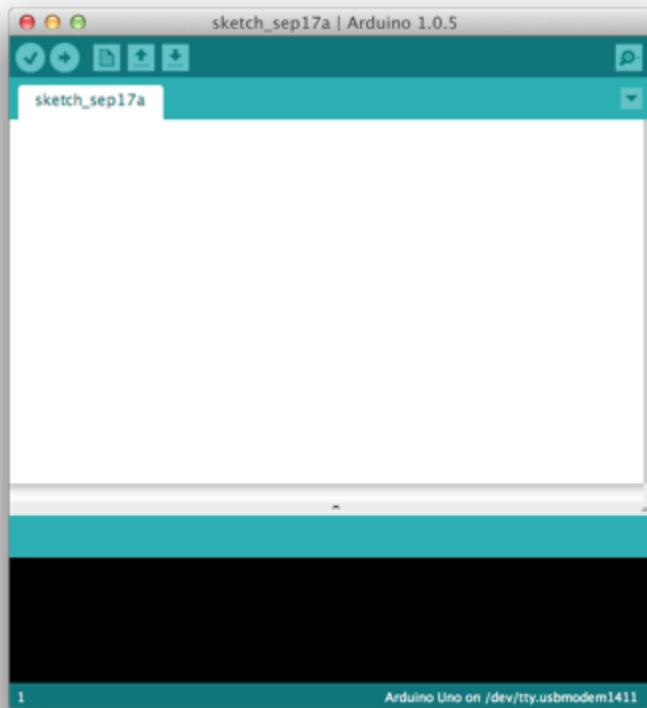
Rails connect inside the board



use them for ground and power

LET'S CODE ALREADY!

Arduino IDE



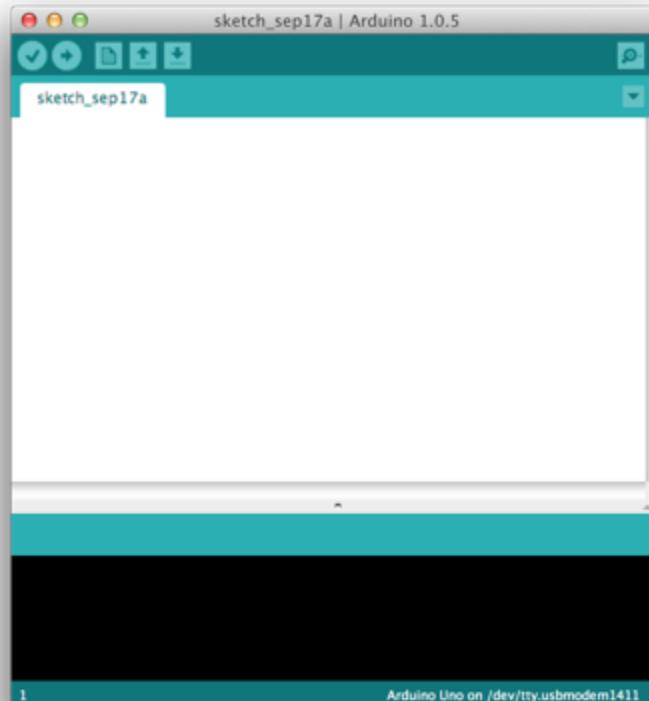
Arduino IDE

Wait... da fuq is an IDE?

(Integrated Development Environment)

Arduino IDE

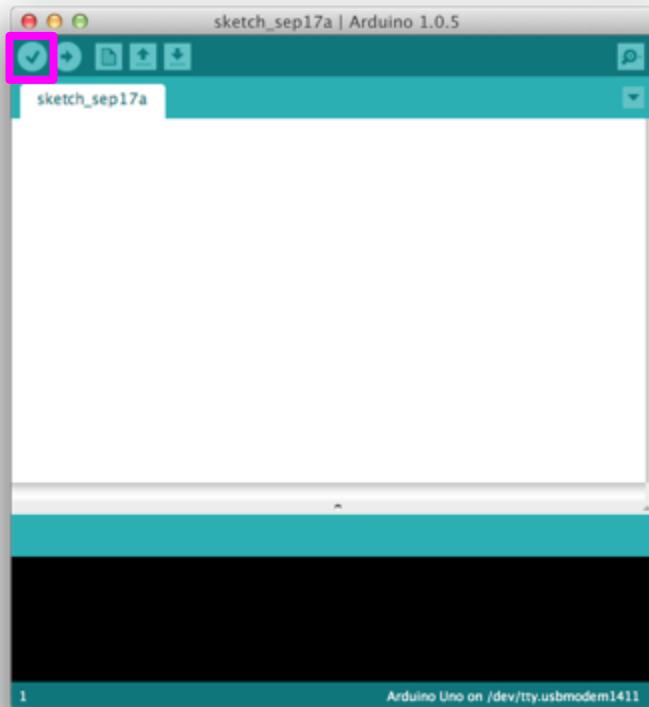
it looks like processing



except green...

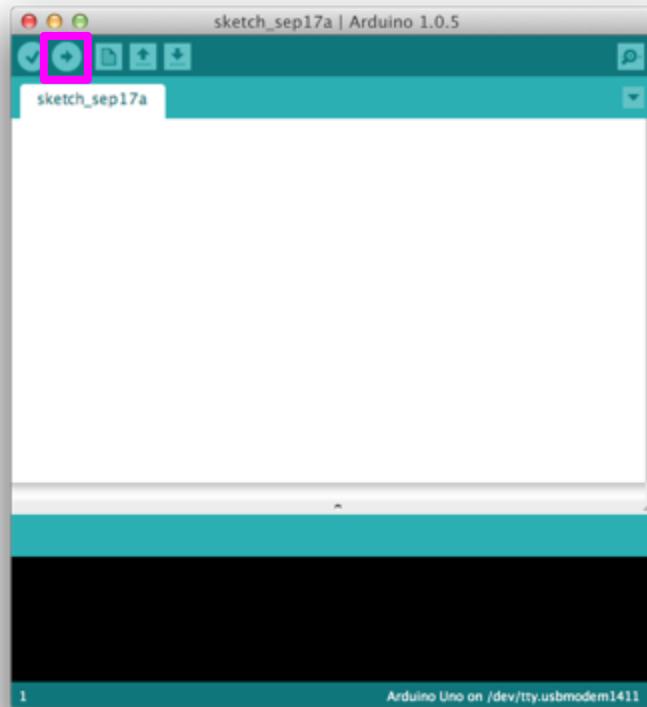
Arduino IDE

Verify code



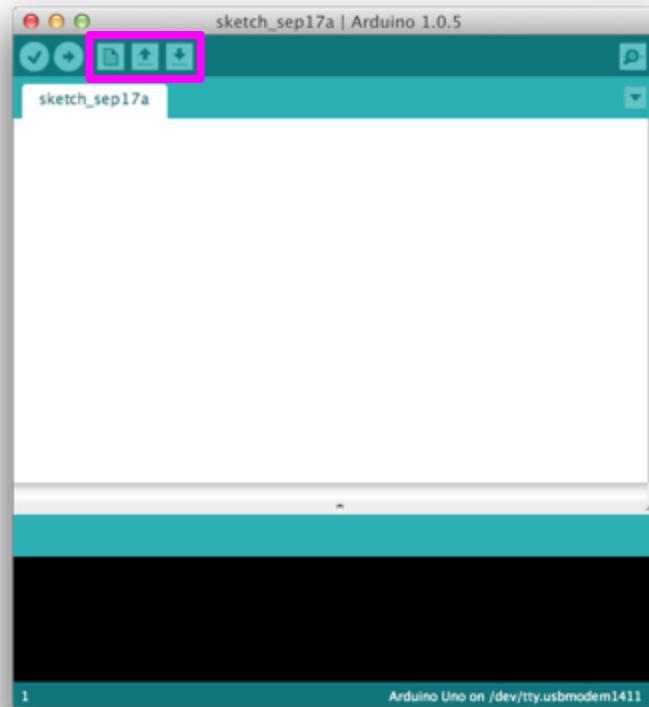
Arduino IDE

Upload code to board



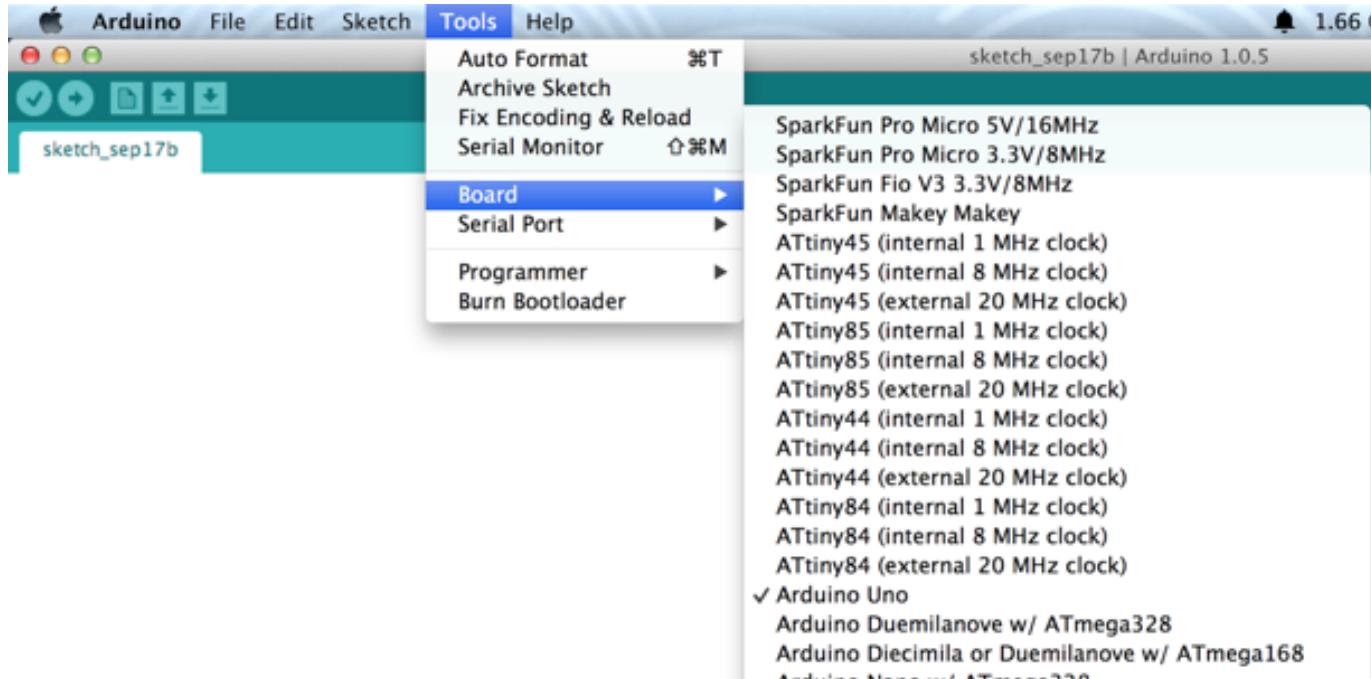
Arduino IDE

New | Open | Save



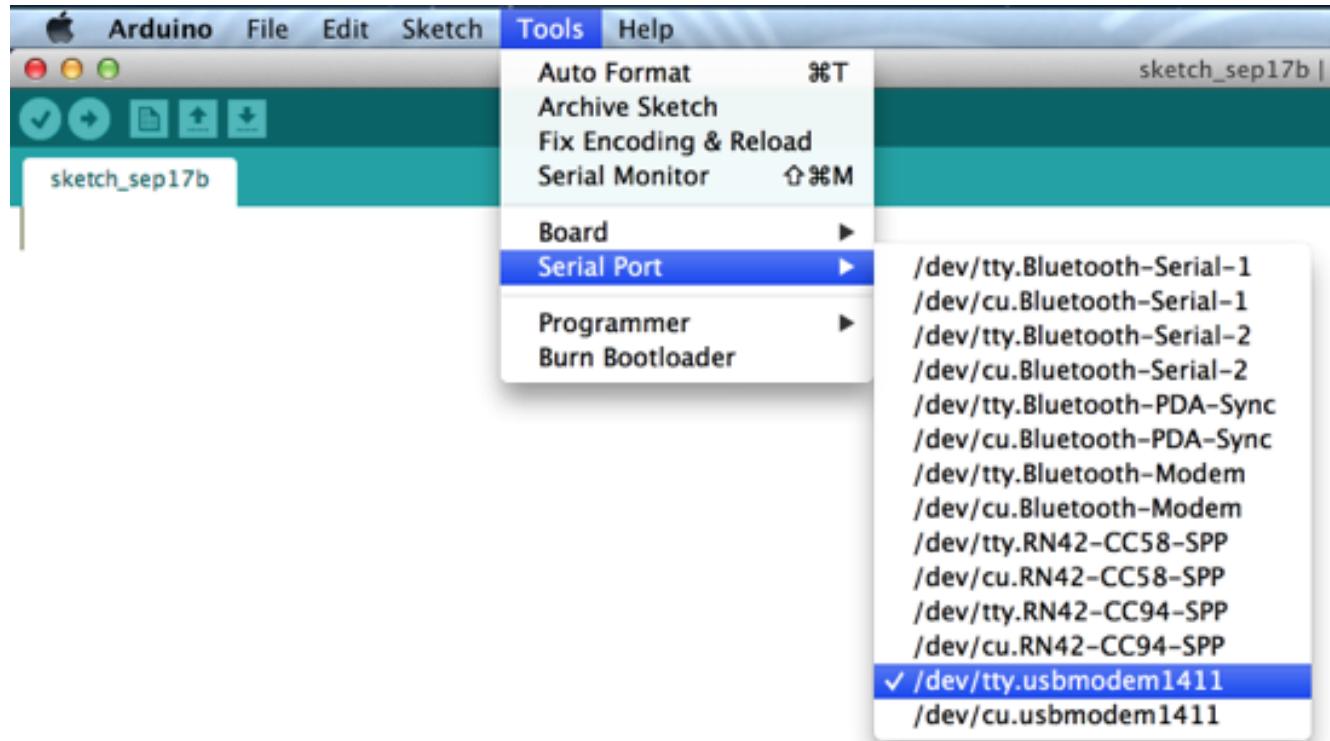
Arduino IDE

tell it what board you're using



Arduino IDE

tell it what serial port it's plugged into



Arduino IDE

Open up the Blink sketch.

(FILE > EXAMPLES > BASICS > Blink Sketch)

Arduino IDE

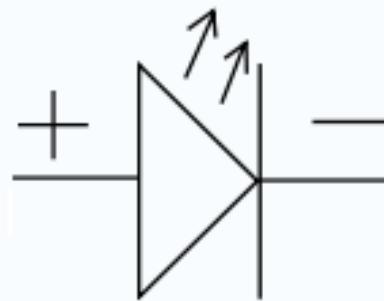
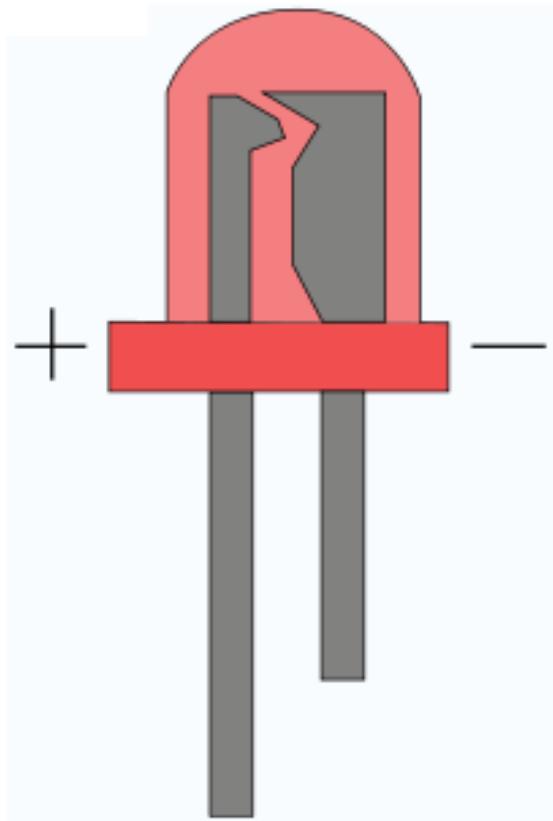
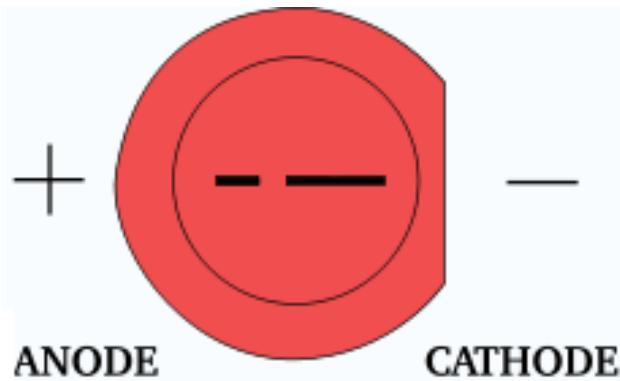
pinMode: Configures the specified pin to behave either as an input or an output.

```
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
    // initialize the digital pin as an output.  
    pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
    digitalWrite(led, HIGH);      // turn the LED on (HIGH is the voltage level)  
    delay(1000);                // wait for a second  
    digitalWrite(led, LOW);       // turn the LED off by making the voltage LOW  
    delay(1000);                // wait for a second  
}
```

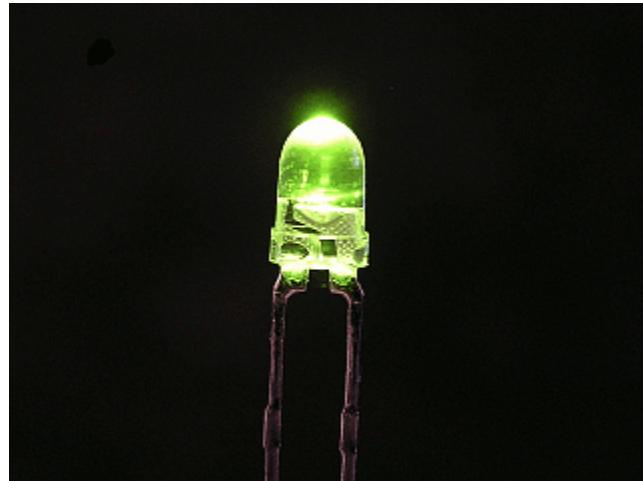
Arduino IDE

digitalWrite: Write a **HIGH** or a **LOW** value to a digital pin.

```
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
    // initialize the digital pin as an output.  
    pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
    digitalWrite(led, HIGH);          // turn the LED on (HIGH is the voltage level)  
    delay(1000);                  // wait for a second  
    digitalWrite(led, LOW);         // turn the LED off by making the voltage LOW  
    delay(1000);                  // wait for a second  
}
```

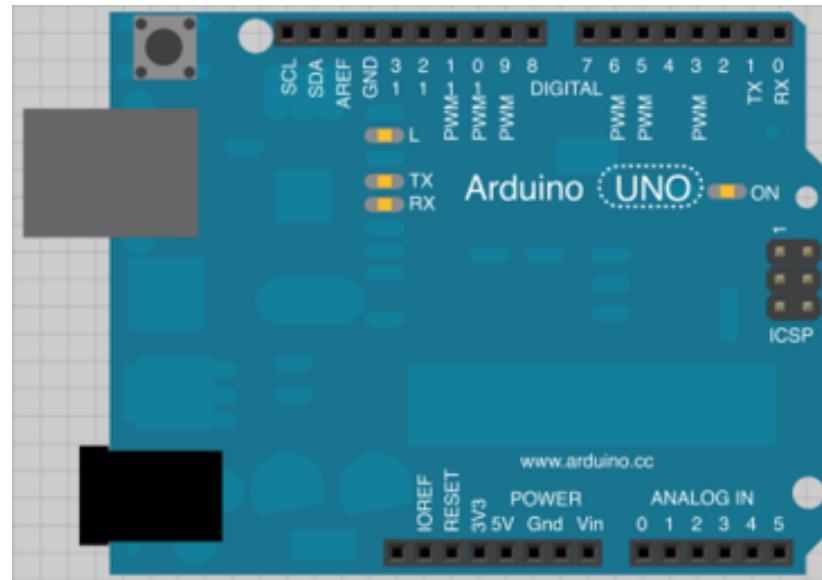


GO TIME!



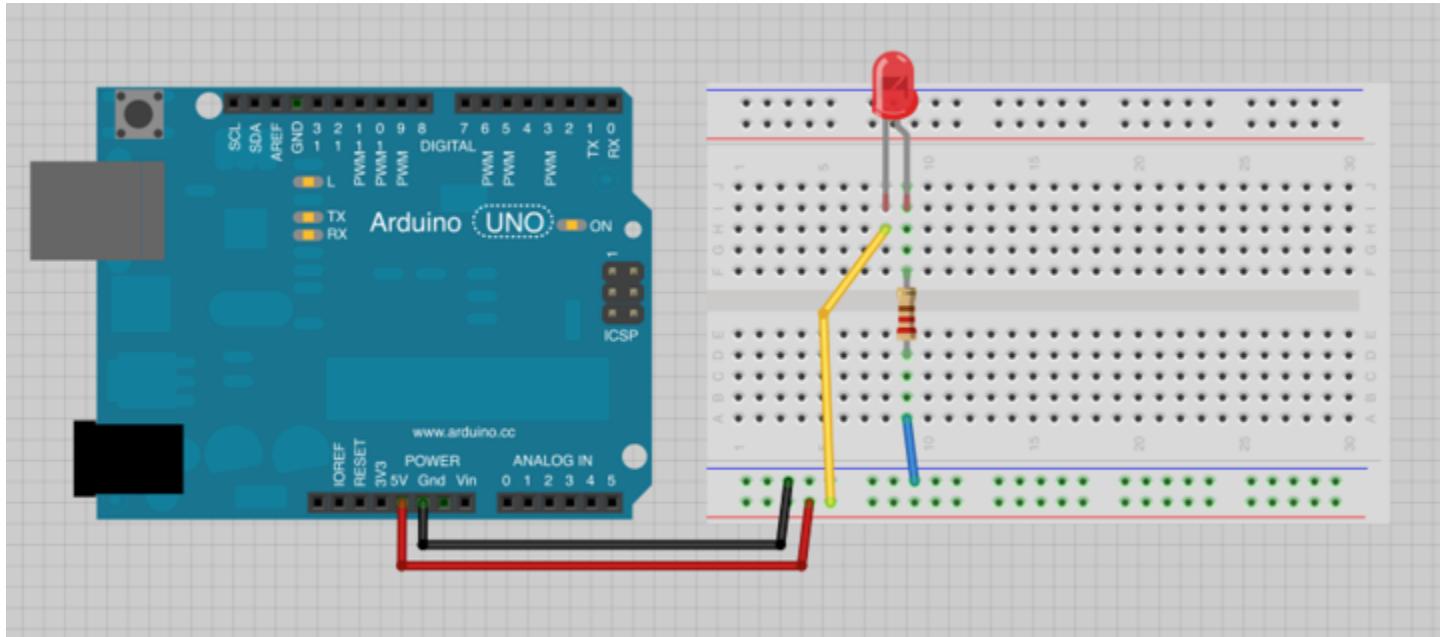
Connecting an LED

STEP 1: Board?



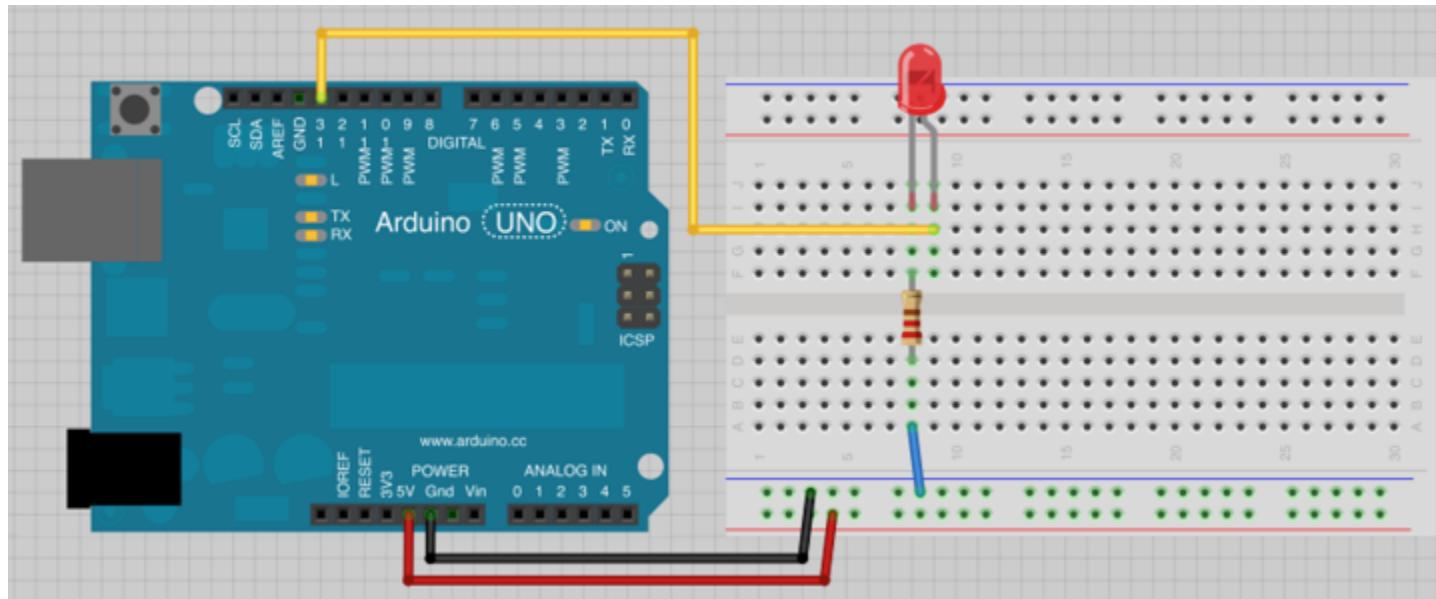
Connecting an LED

STEP 2: BreadBoard + LED + Wires + Resistor



Connecting an LED

STEP 3: Wire Pin 13 as our Digital Output



Arduino IDE

LET'S EDIT THIS JUNK!

```
void setup() {  
    //start the serial connection from Arduino back to computer  
    Serial.begin(9600); ←  
  
    // initialize the digital pin as an output.  
    pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
    digitalWrite(led, HIGH);    // turn the LED on (HIGH is the voltage level)  
    Serial.println("LED is On"); ←  
    delay(1000);              // wait for a second  
    digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW  
    Serial.println("LED is Off"); ←  
    delay(1000);              // wait for a second  
}
```

Arduino IDE

Sets the data rate in bits per second, for serial data transmission.

```
void setup() {  
    //start the serial connection from Arduino back to computer  
    Serial.begin(9600); ←  
  
    // initialize the digital pin as an output.  
    pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
    digitalWrite(led, HIGH);    // turn the LED on (HIGH is the voltage level)  
    Serial.println("LED is On");  
    delay(1000);                // wait for a second  
    digitalWrite(led, LOW);     // turn the LED off by making the voltage LOW  
    Serial.println("LED is Off");  
    delay(1000);                // wait for a second  
}
```

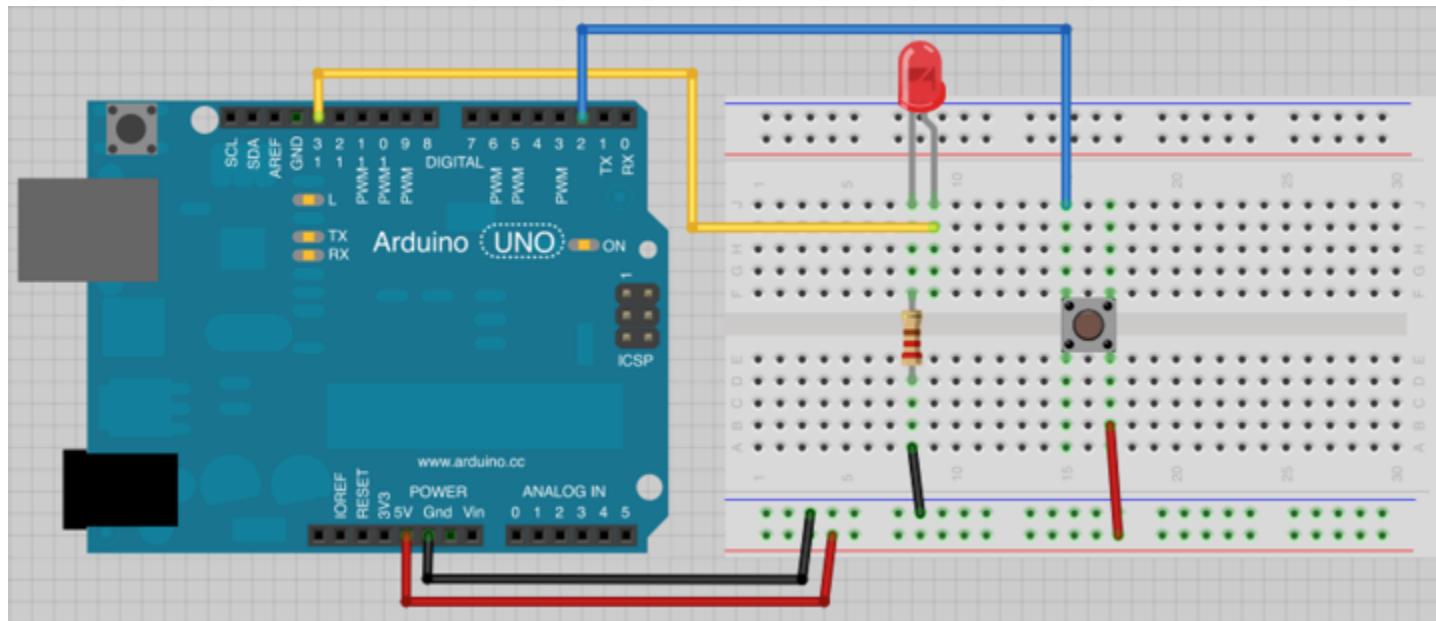
Arduino IDE

Prints data to the serial port as human-readable text

```
void setup() {  
    //start the serial connection from Arduino back to computer  
    Serial.begin(9600);  
  
    // initialize the digital pin as an output.  
    pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
    digitalWrite(led, HIGH);    // turn the LED on (HIGH is the voltage level)  
    Serial.println("LED is On"); ←  
    delay(1000);              // wait for a second  
    digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW  
    Serial.println("LED is Off"); ←  
    delay(1000);              // wait for a second  
}
```

Connecting an LED

STEP 4: Add Button and Wire Pin 2 as our Digital Input



Connecting an LED

Read the button input in our code

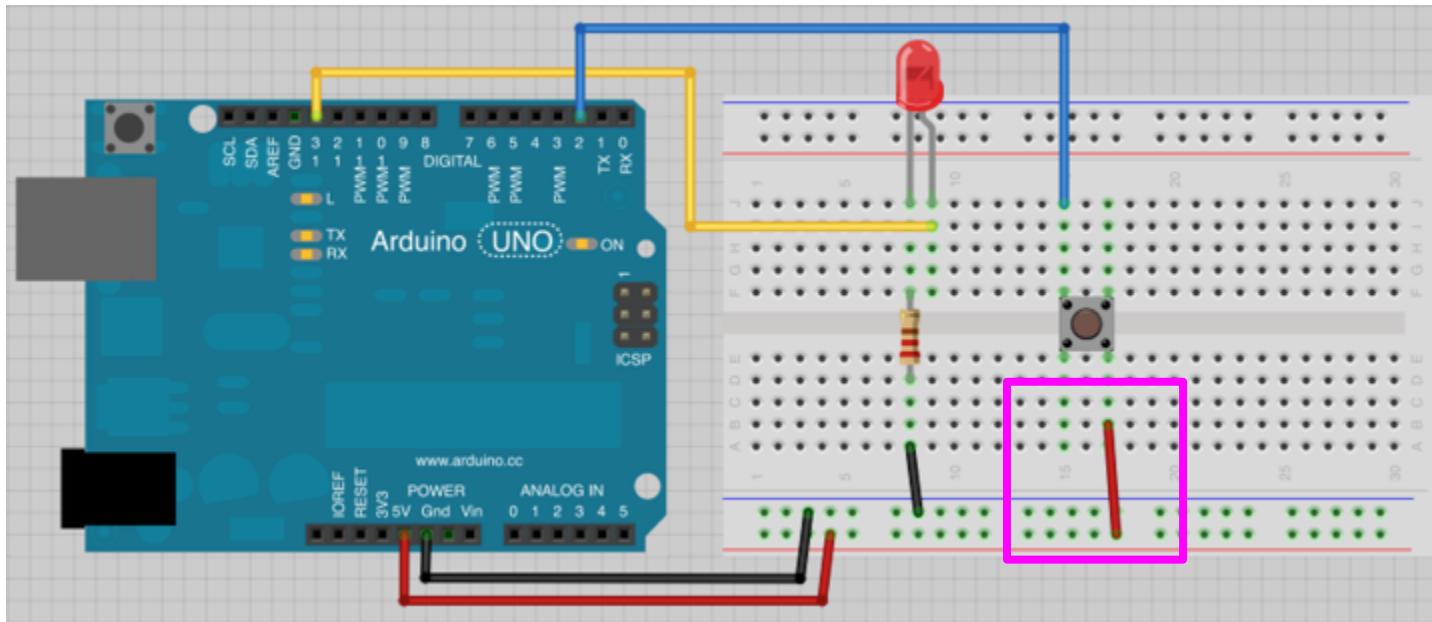
```
int led = 13;
int buttonPin = 2;
int buttonState = 0; ←

// the setup routine runs once when you press reset:
void setup() {
    // initialize the pinModes
    pinMode(led, OUTPUT);
    pinMode(buttonPin, INPUT); ←
}

// the loop routine runs over and over again forever:
void loop() {
    //read the button
    buttonState = digitalRead(buttonPin); ←
    //Perform different actions depending on the state of the button
    if(buttonState == HIGH){ ←
        digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
        delay(1000);           // wait for a second
    } else {
        digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
        delay(1000);           // wait for a second
    }
}
```

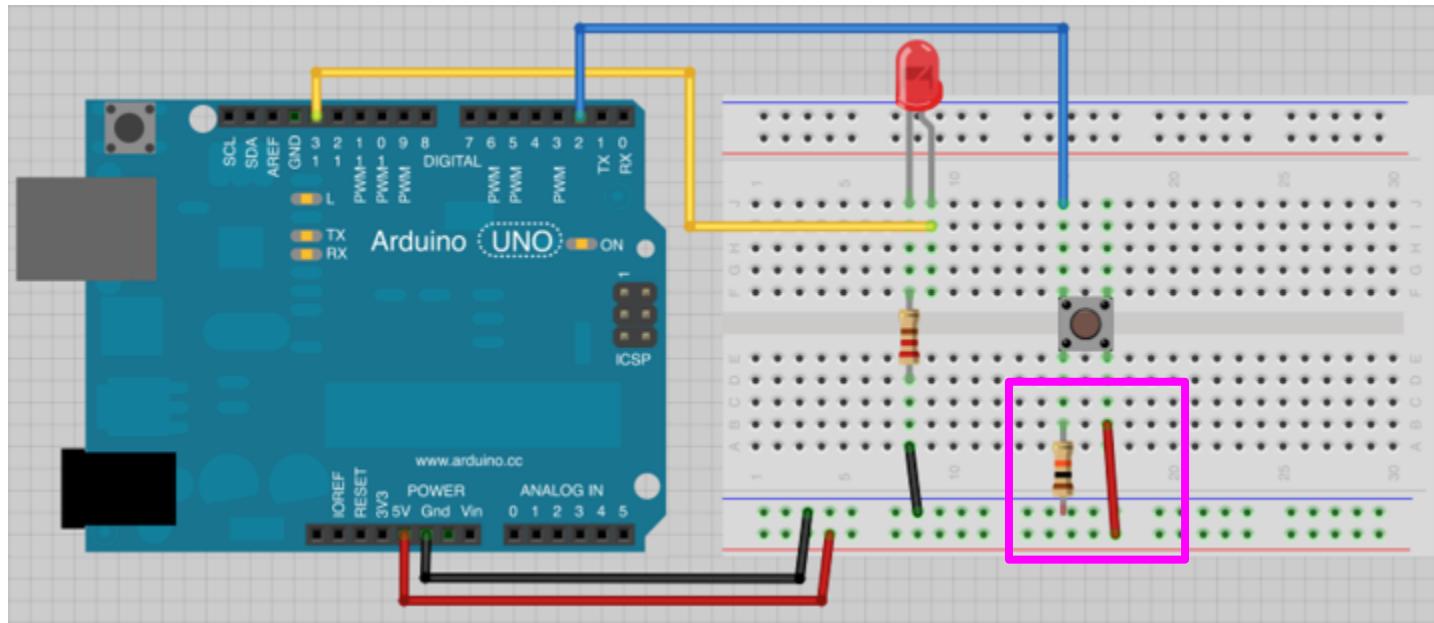
Connecting an LED

STEP 5: Connect Button



Connecting an LED

STEP 5: Connect Button



**Yup, there's
Homework**

Homework

Update your code so the button triggers a state change

For example: the LED stays on when you push it and turns off when the button is pressed again

Homework

Take a video with your phone or webcam and post it to tumblr along with the code