Johnny-Five

Javascript robotics programming framework

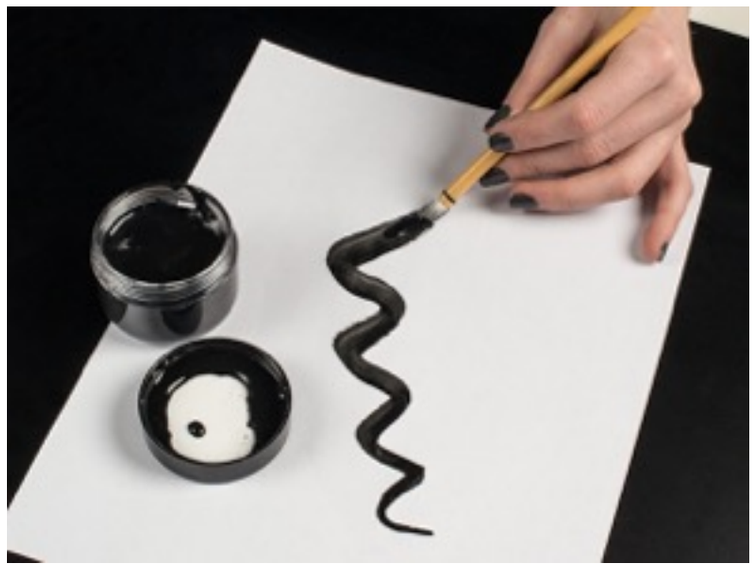
Who ?

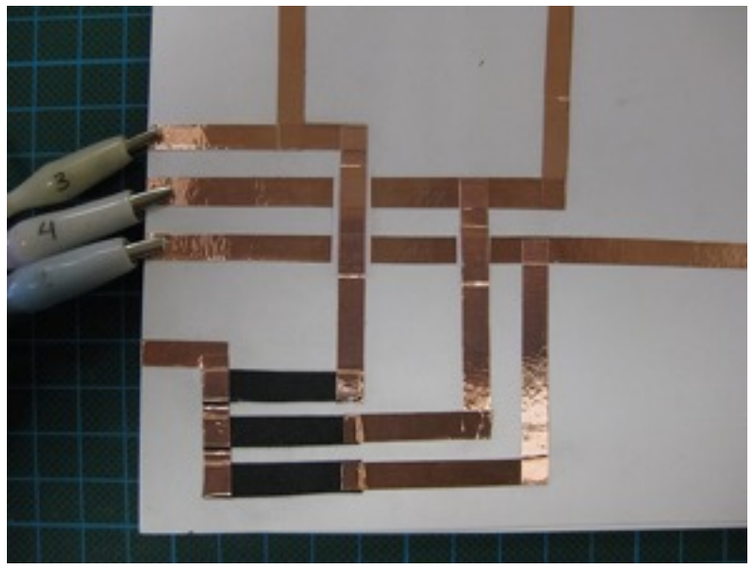
Sara

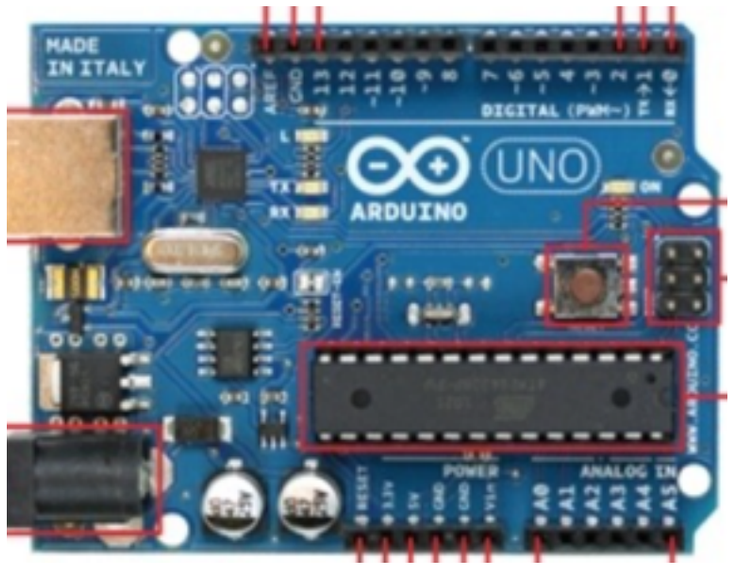
Stef

Divy

What ?



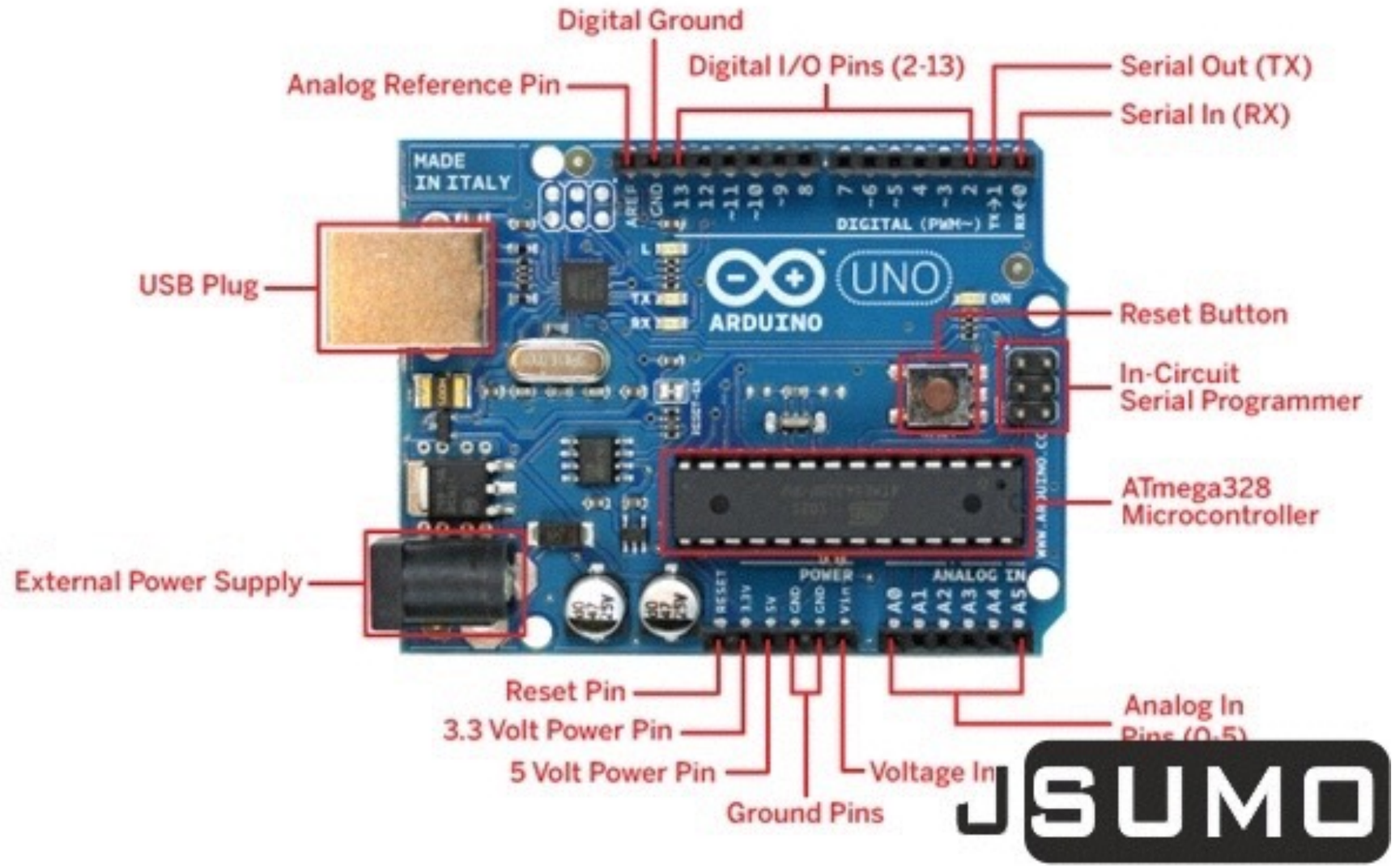




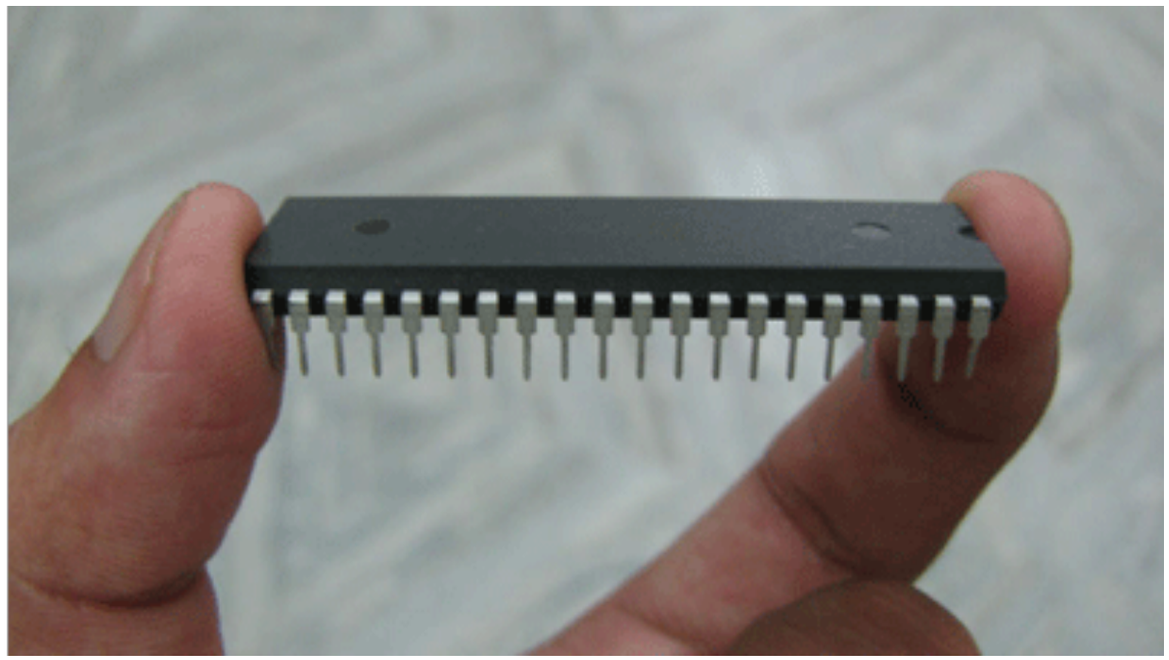
Plan

- Learn what Arduino/Johnny Five is and complete setup(15’)
- Form groups and follow workshop steps(60’)
- Demo (15’)

Arduino

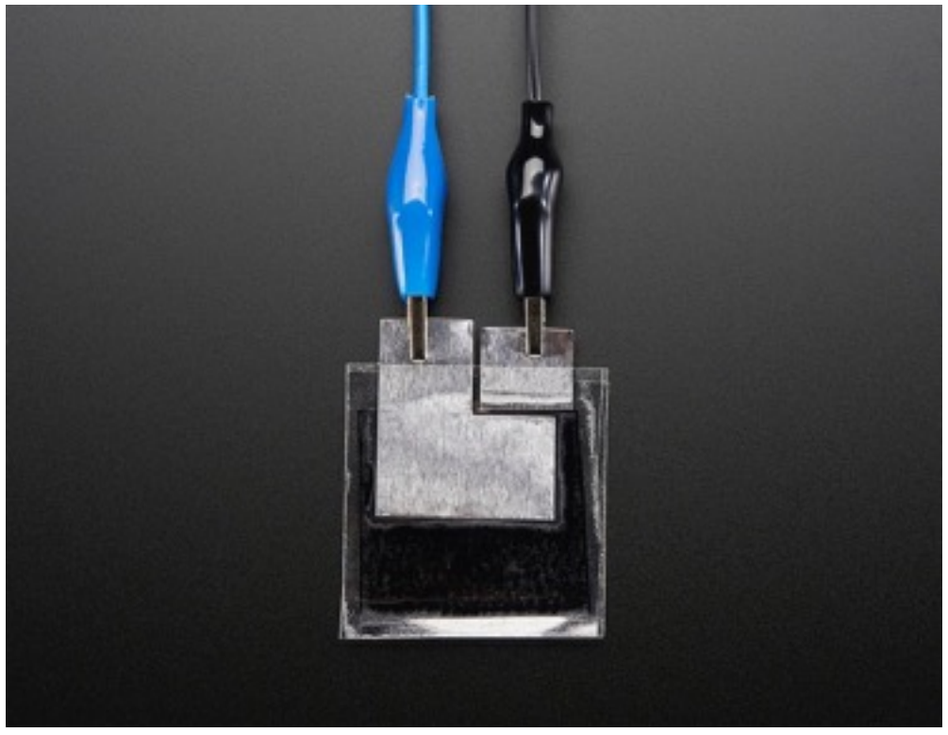

Digital Ground
Analog Reference Pin
Digital I/O Pins (2-13)
Serial Out (TX)
Serial In (RX)
USB Plug
Reset Button
In-Circuit
Serial Programmer
ATmega328
Microcontroller
External Power Supply
Reset Pin
3.3 Volt Power Pin
5 Volt Power Pin
Ground Pins
Voltage In
Analog In
MADE
IN ITALY
DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN
JSUMO

Microcontroller



A microcontroller is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals.

Examples Input

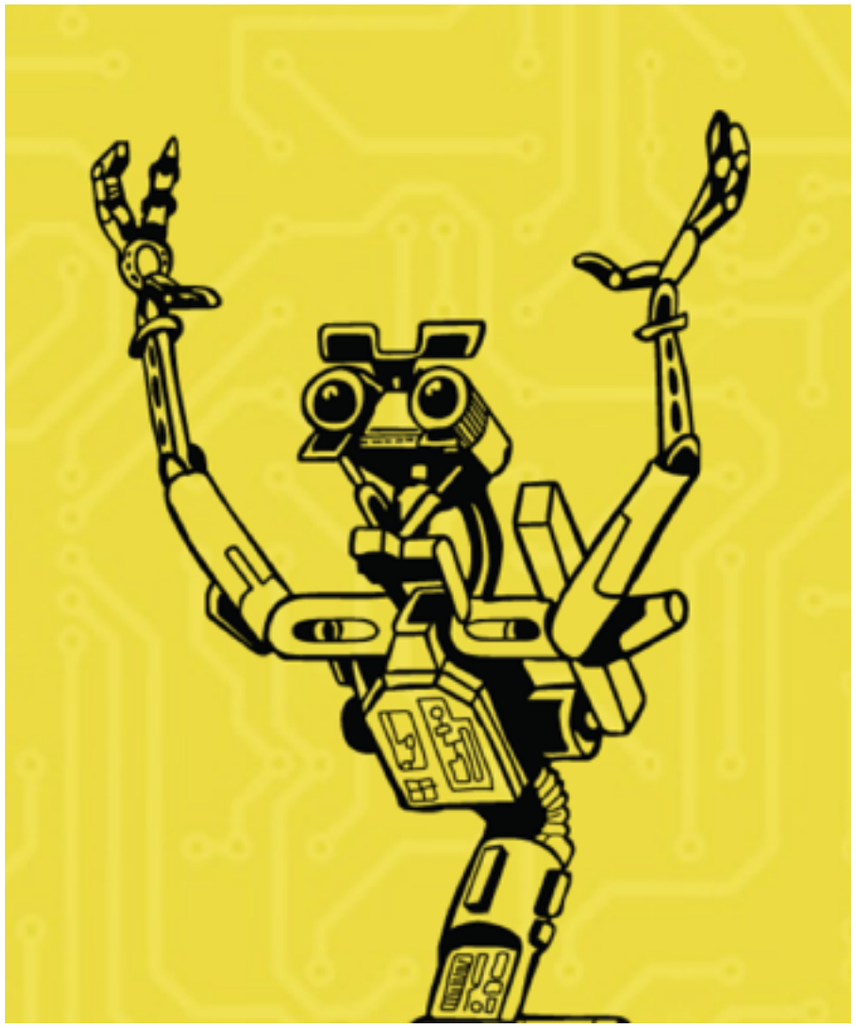


- Conductive tape
- Conductive thread
- Anything that conducts electricity

Examples output

- Sound (buzzer)
- Light (Led)
- LCD (text)

Johnny Five



Javascript vs C

Typical Arduino Sketch

```
void loop()
   {
    //do some stuff
    delay(1000);
}
```

Using johnny-five

```
board.on("ready", function()
{
   sensor.on("change",
function() {
      console.log(this.value);
   });

});
```

Johnny-Five API

- 36 component classes: `Board, Led, Piezo, Button, Pin, Accelerometer, Servo, Button`….
- Each component class comes with their own events and methods.

Led class has methods: `blink()`, `on()`, `off()`, `toggle()`, `strobe()`, `fade()`…

Button class has events: `"hold", "down"`/`"press"`, `"up"`/`"release"`

Handling Board Events

- Just like JQuery

```
var five = require("johnny-five");
var board = new five.Board();
```

```
board.on("ready", function() {
  var led = new five.Led(13);
  led.blink(500);
});
```

- Other events likes: “connect”, “warn”, “fail”, “message”

Firmata

- Johnny-Five uses a protocol called Firmata to communicate with the microcontroller over USB (Universal Serial Bus).

- Firmata is a protocol for communicating from microcontrollers to software. (i.e from our Arduinos to our node!)

- Basically this describes how the bits traveling through the wire are encoded

- You must upload the firmata protocol on your Arduino if you want Johnny-five to work ☺

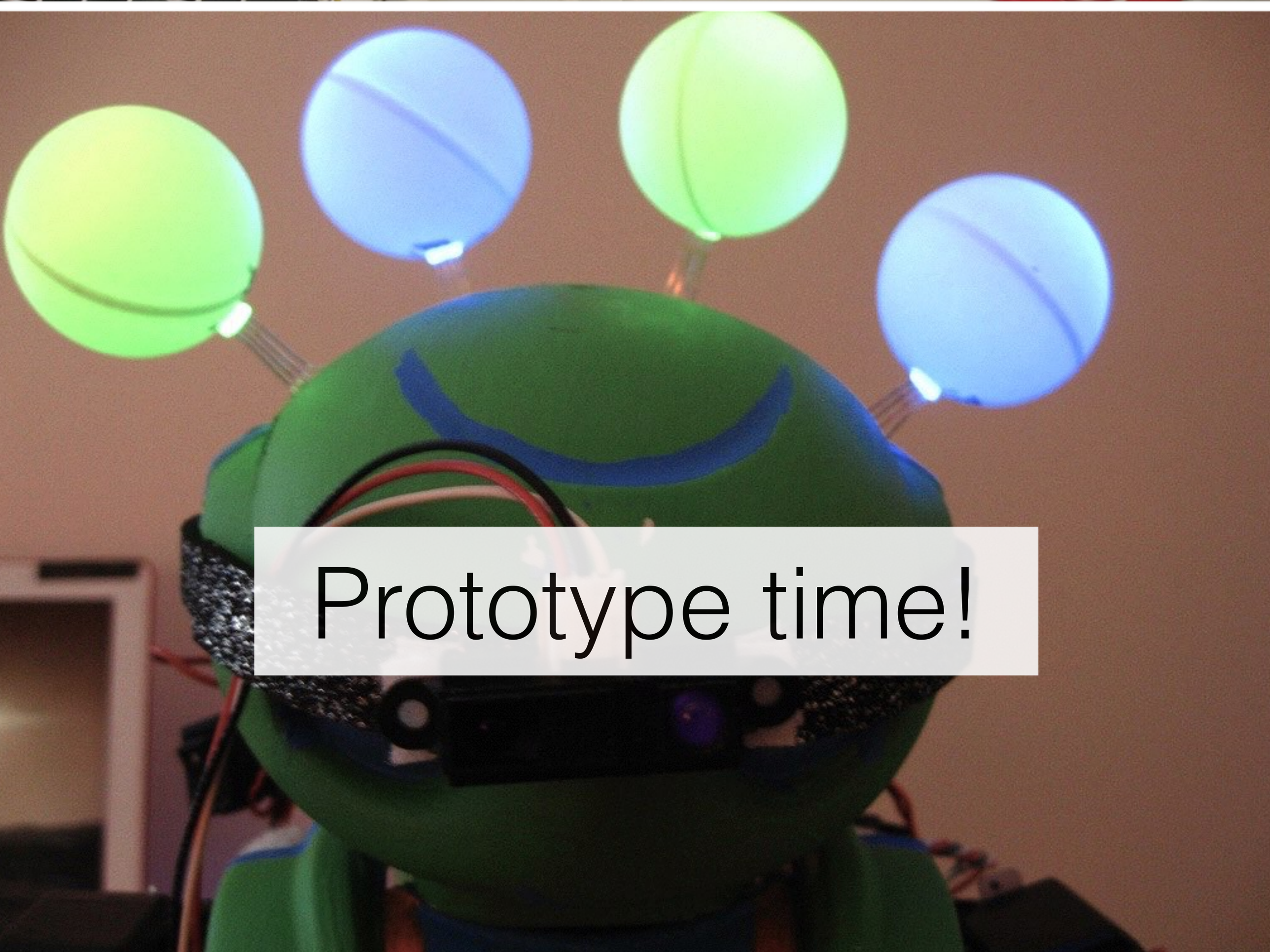
REPL

- Read, Eval, Print, Loop
- Allows us to type commands that control the hardware while the program is running

```
board.on("ready", function() {
    var led = new five.Led(13);
    this.repl.inject({
      on: function() {
        led.on();
      } });
  });
```

Resources

- http://johnny-five.io/api/
- http://node-ardx.org/
- https://github.com/rwaldron/johnny-five



Prototype time!

Prototype time

- Let’s make teams
- **1st step**: Use an Arduino to build an LED light that will change its color by sending a tweet! (15’)
- **2nd step**: Use or design a sensor of your choice to gather some data. (15’)
- **3rd step**: Combine stage one and stage two to create an alert system that changes the LED color when a critical data metric has occurred.(20’)
- Demo time (20’)

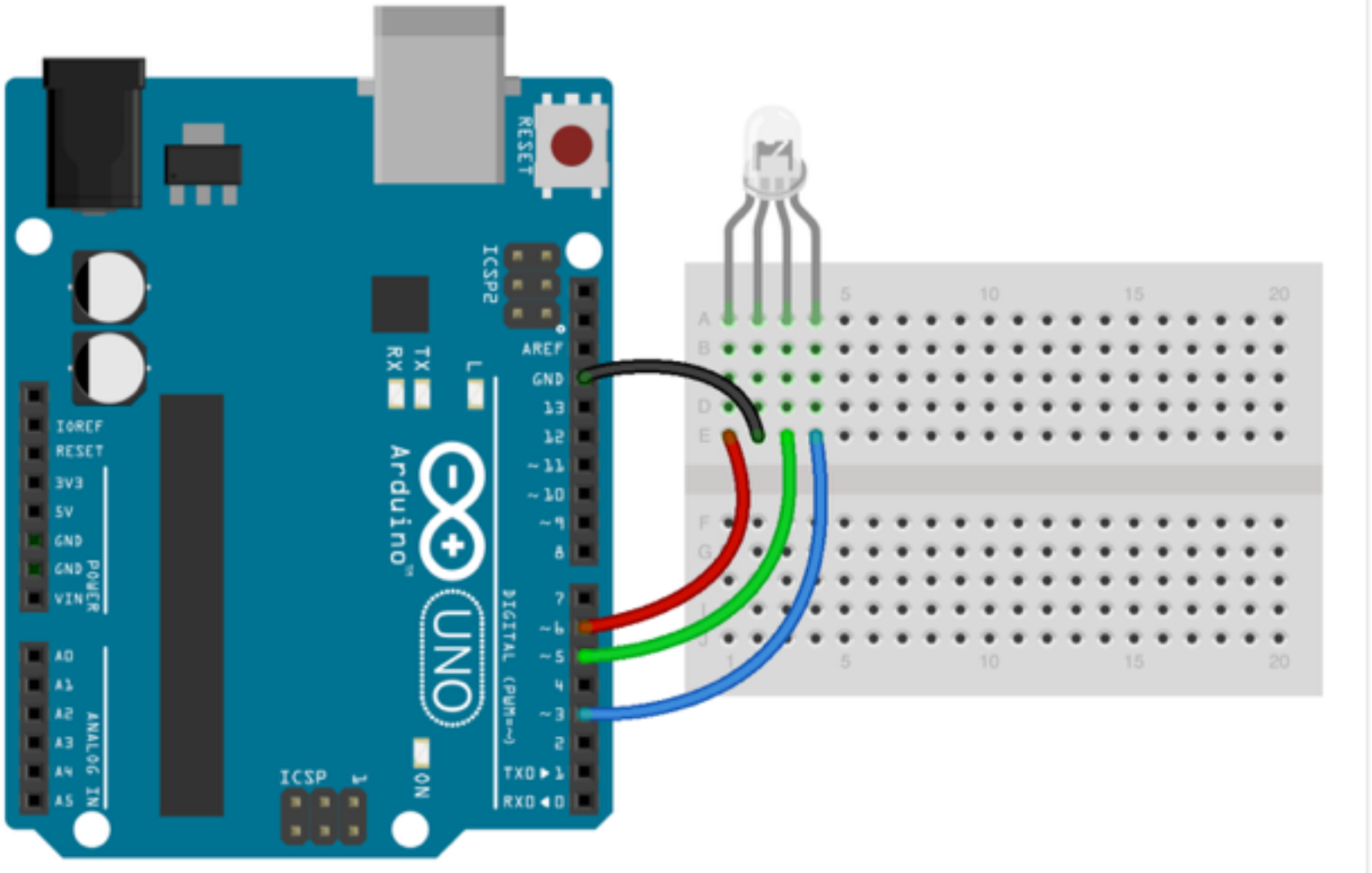
Setup

- 1. Install Node
- 2. Install NPM library Johnny-Five
- 3. Download & install Arduino ide
- Link to detailed instructions for setup
- https://goo.gl/OGNX5x

Github repo
code examples

- https://github.com/hackidemia/nodebots_nyc

1 step: Hello World (Blink)



1 step: Hello World (Blink)

```
var five = require("johnny-five");
var myBoard = new five.Board();


myBoard.on("ready", function() {
  var redLED = new five.Led(11);
  redLED.blink();
});
```

1 step: RGB

Open rgb_eg.js file in Step1 folder

```
var led = new five.Led.RGB({
  pins: {
    red: 11,
    green: 10,
    blue: 9
  }
});
```

1 step: RGB

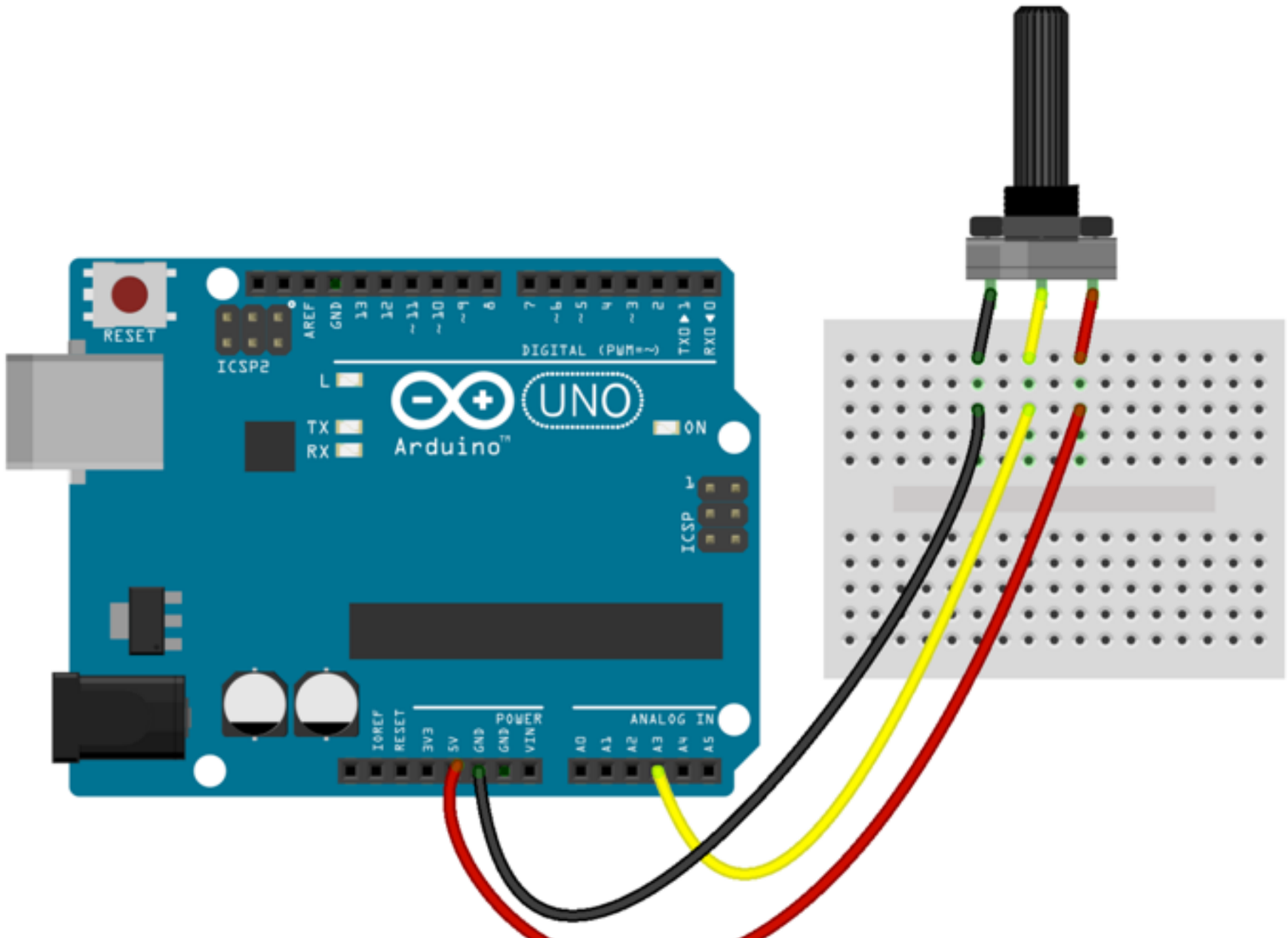
Open rgb_eg.js file in Step1 folder

```
  // this allows you to control board from terminal after
  this.repl.inject({
    led: led
  });

  // Turn it on and set the initial color
  led.on();
  led.color("#FF0000");
  //change the color to pink
  led.blink(500);
```

2nd Step: Sensor

Open pmeter.js file in Step2 folder



2nd Step: Sensor

Open pmeter.js file in Step2 folder

```
var pmeter = new five.Sensor("A0");

pmeter.on("change", function() {
    console.log("The value of the potentiometer is:" +this.value );
});
```

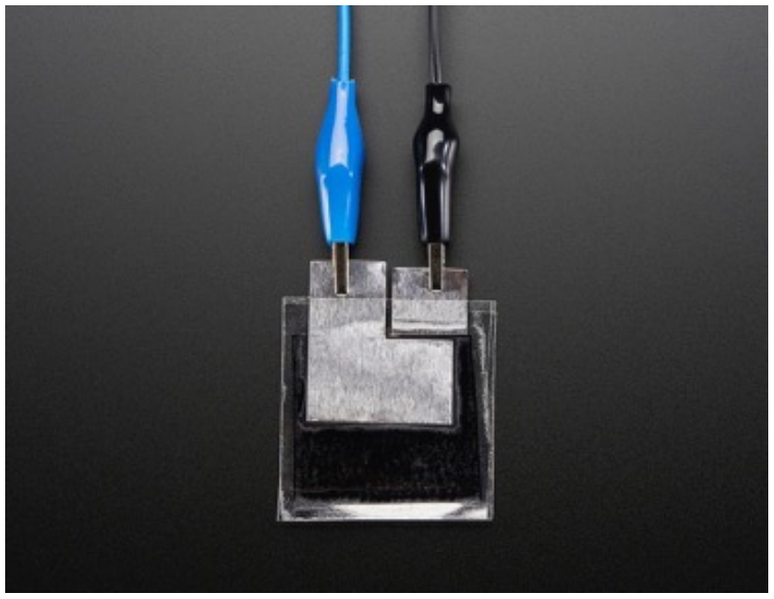
3rd Step: Input + Output

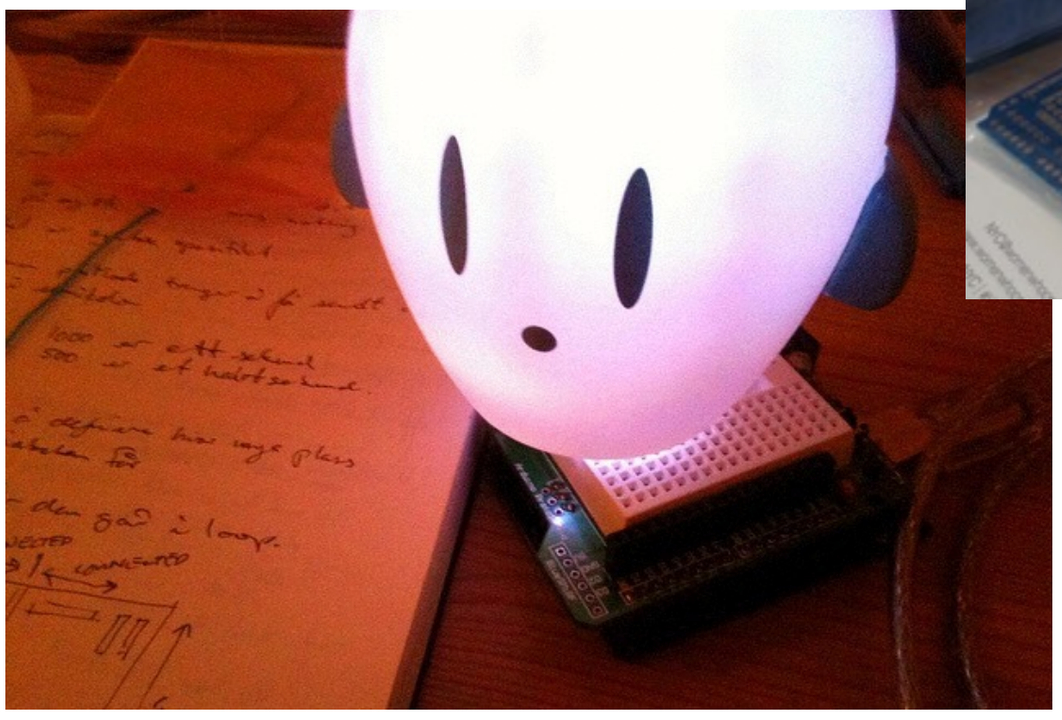
Open control_led.js file in Step3 folder

```
pmeter.on("change", function() {
    console.log("The value of the potentiometer is:" +this.value );
    //Add code here to make the RBG sensor turn green when potentiome
    //Use APIS--> http://johnny-five.io/api/led.rgb/ and http://johnn

    if (this.value<200 ){
        rgb.color("#ff00ff");
    }else if(this.value>200 && this.value<600 ){
        rgb.color("#a020f0");
    }else if(this.value>600){
        rgb.color("#ffff00");
    }else{
        rgb.color("#0000ff");
    }
});
```

3rd Step: improvise:)





Demo time!

Bonus

- https://www.codetutorial.io/nodejs-socket-io-and-jhonny-five-to-control-arduino/