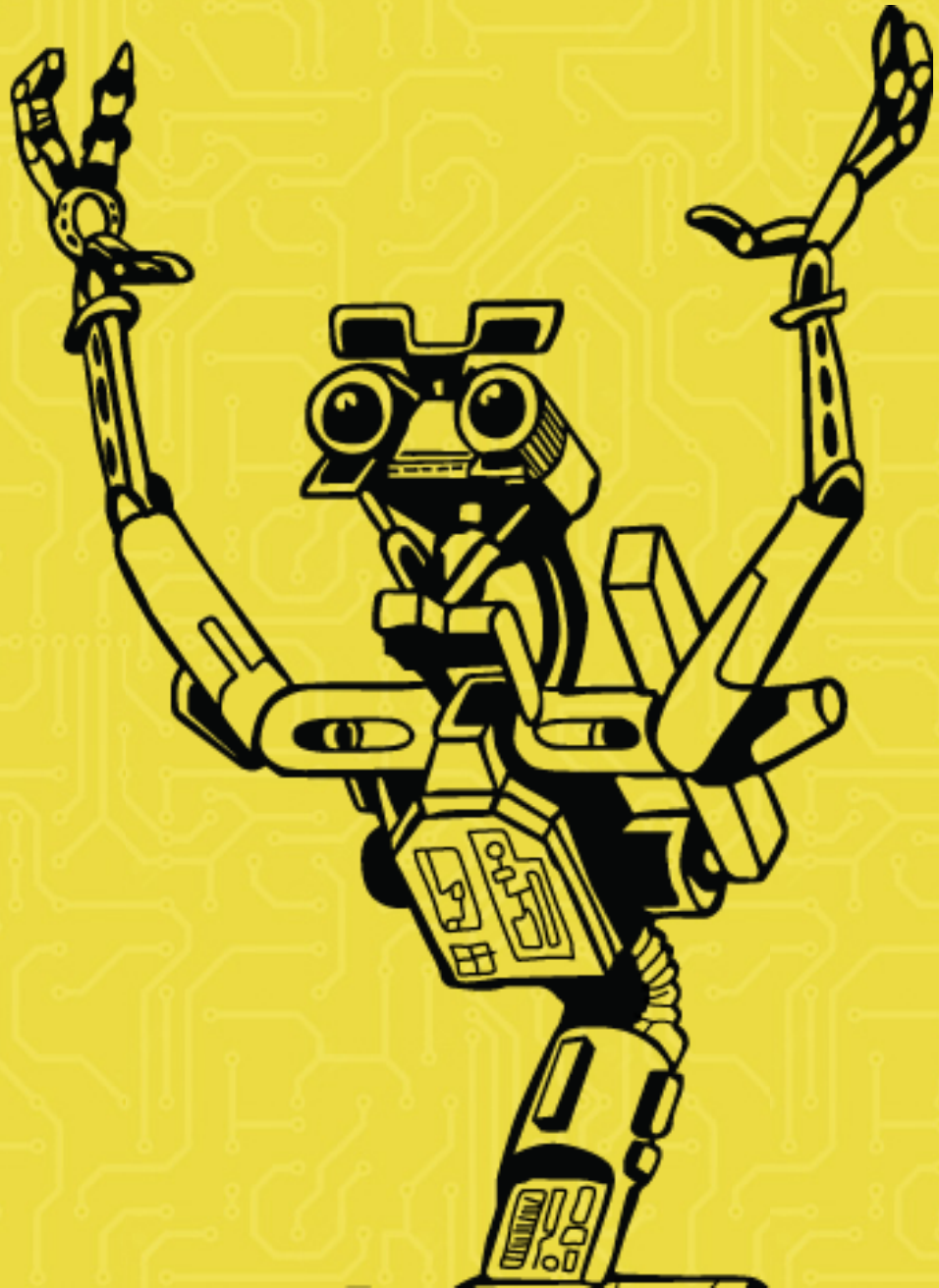


USING JOHNNY FIVE WITH ARDUINO

Agenda

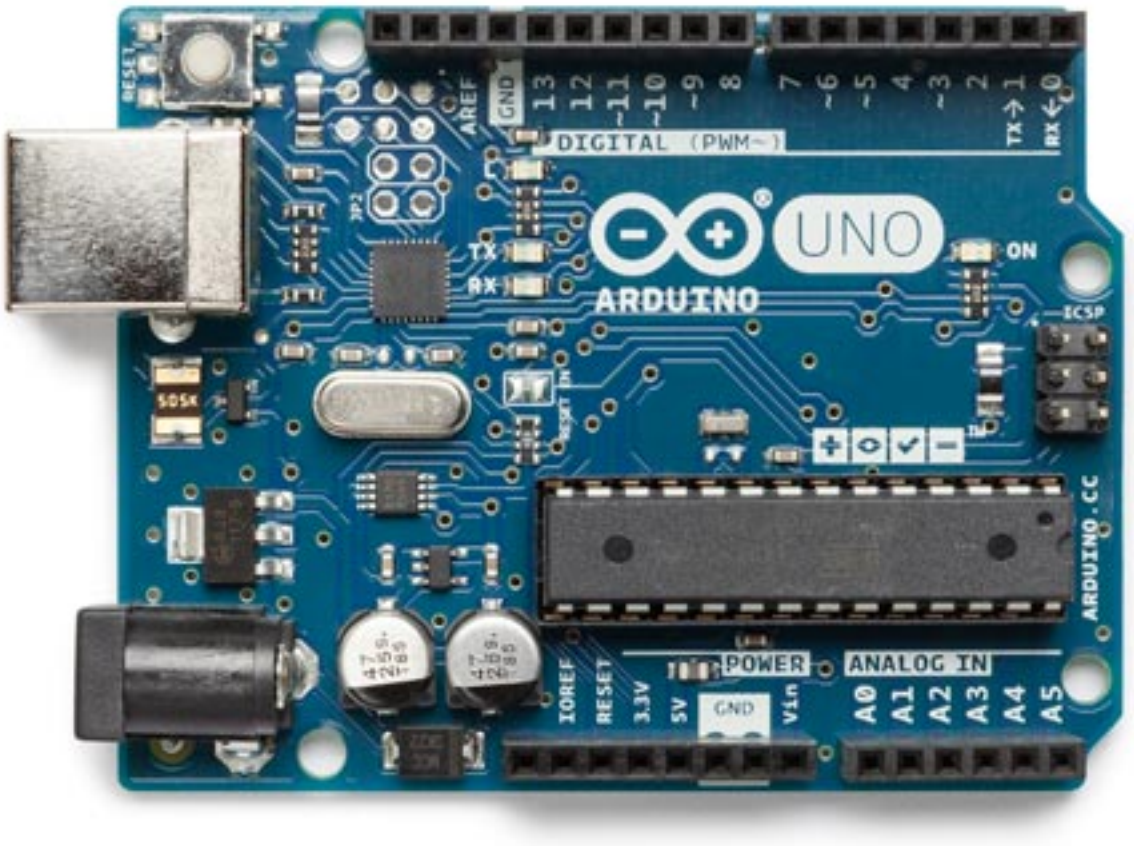
- Last Week
- IoT and Arduino
- Johnny-Five
- In-Class Challenge

What did we cover last week?

What was the significance?

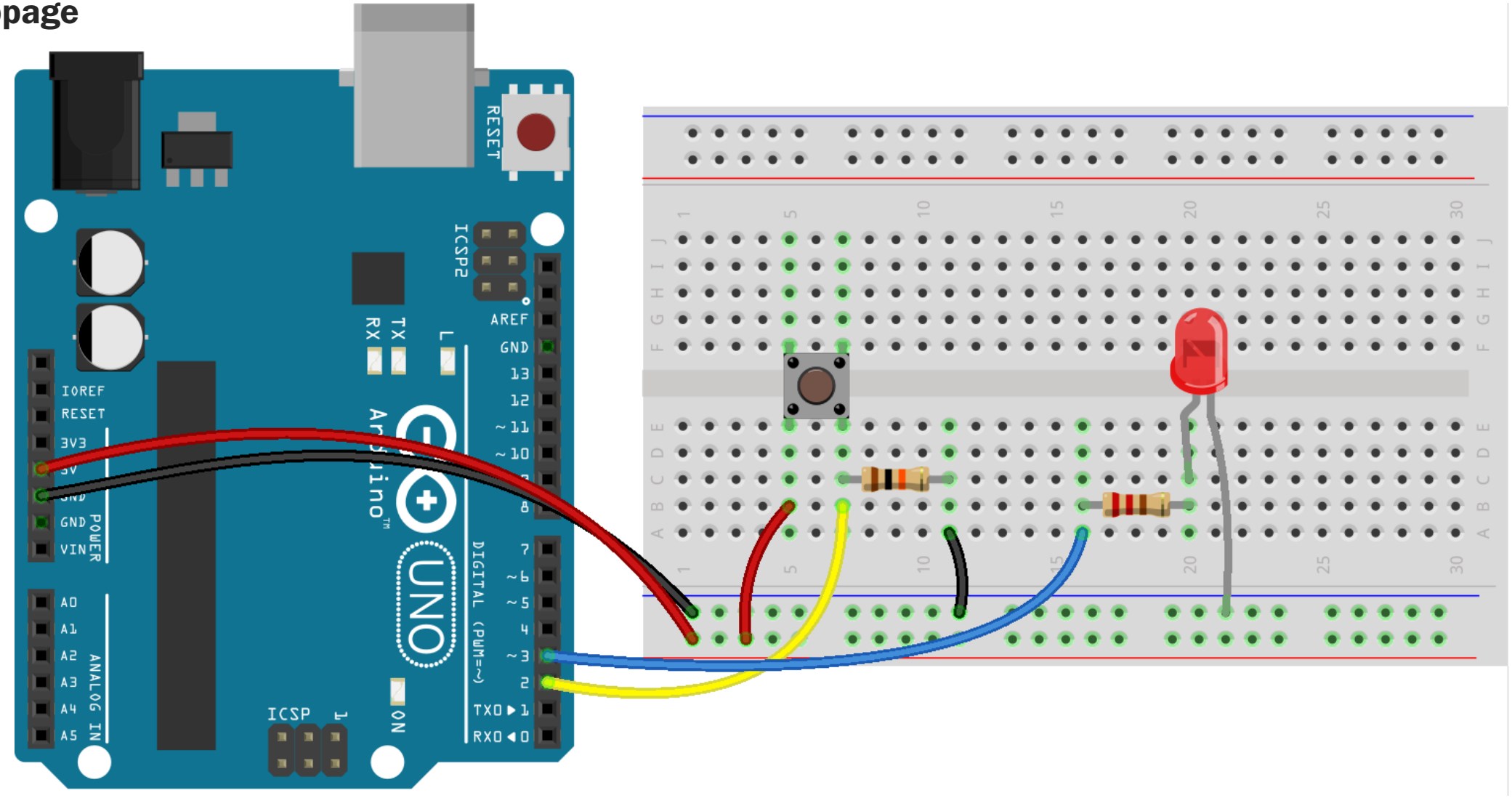
Arduino & Node.js

We will examine how to read and write data from / to an Arduino. Our connection to the microcontroller will be made through a serial (USB) connection. The goal is to create a web interface to interact with the device.



We build the following circuit

And a we used a node.js server to pass information between the arduino and server to communicate with a webpage



Beyond screens – Part II

Talking to an Arduino using Node.js

We examined the serialport module

This was ok, but it wasn't the easiest thing to setup.

```
> npm install serialport
```

Instead We Could Use Johnny-Five (npm)

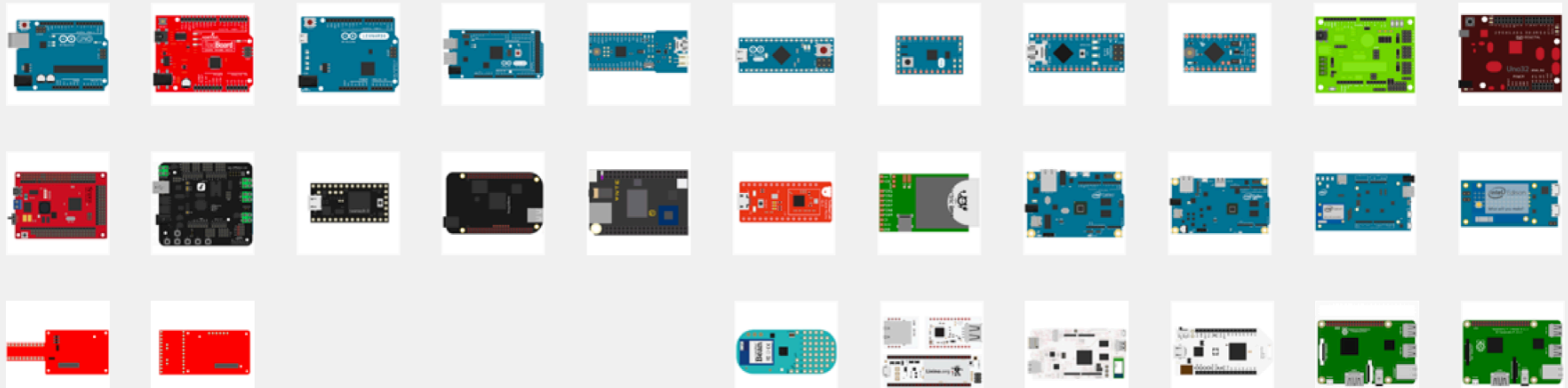
J5

News API Examples Articles Platform Support

Fork me on GitHub

Platform Support

Johnny-Five has been tested with a variety of Arduino-compatible Boards. For non-Arduino based projects, platform-specific IO Plugins are available. IO Plugins allow Johnny-Five code to communicate with any hardware in whatever language that platform speaks!



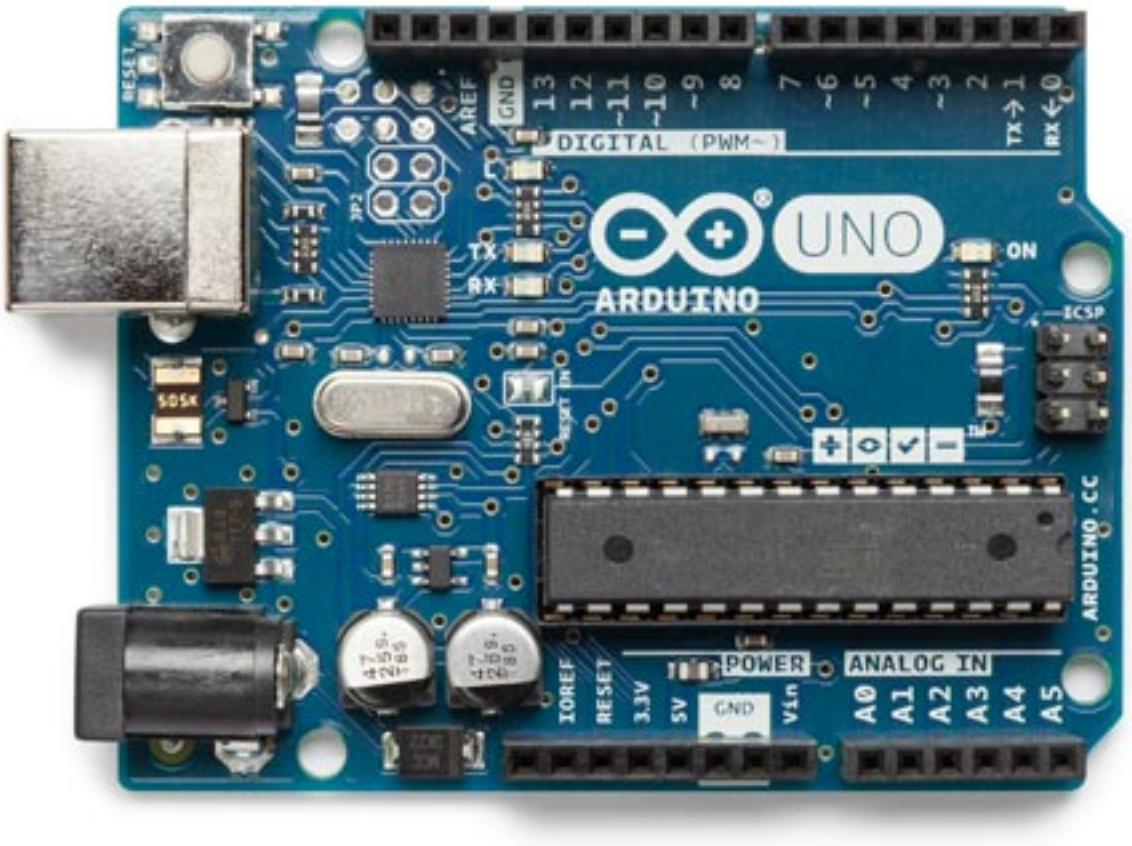
Installing Johnny-Five

https://github.com/rwaldron/johnny-five/wiki/Getting-Started#trouble-shooting

Prepping Your Arduino

To use Johnny-Five, make sure that StandardFirmataPlus is installed on the board:

- Download Arduino IDE
- Plug in your Arduino or Arduino compatible microcontroller via USB
- Open the Arduino IDE, select: File > Examples > Firmata > StandardFirmataPlus
- Click the "Upload" button.
- If the upload was successful, the board is now prepared and you can close the Arduino IDE.



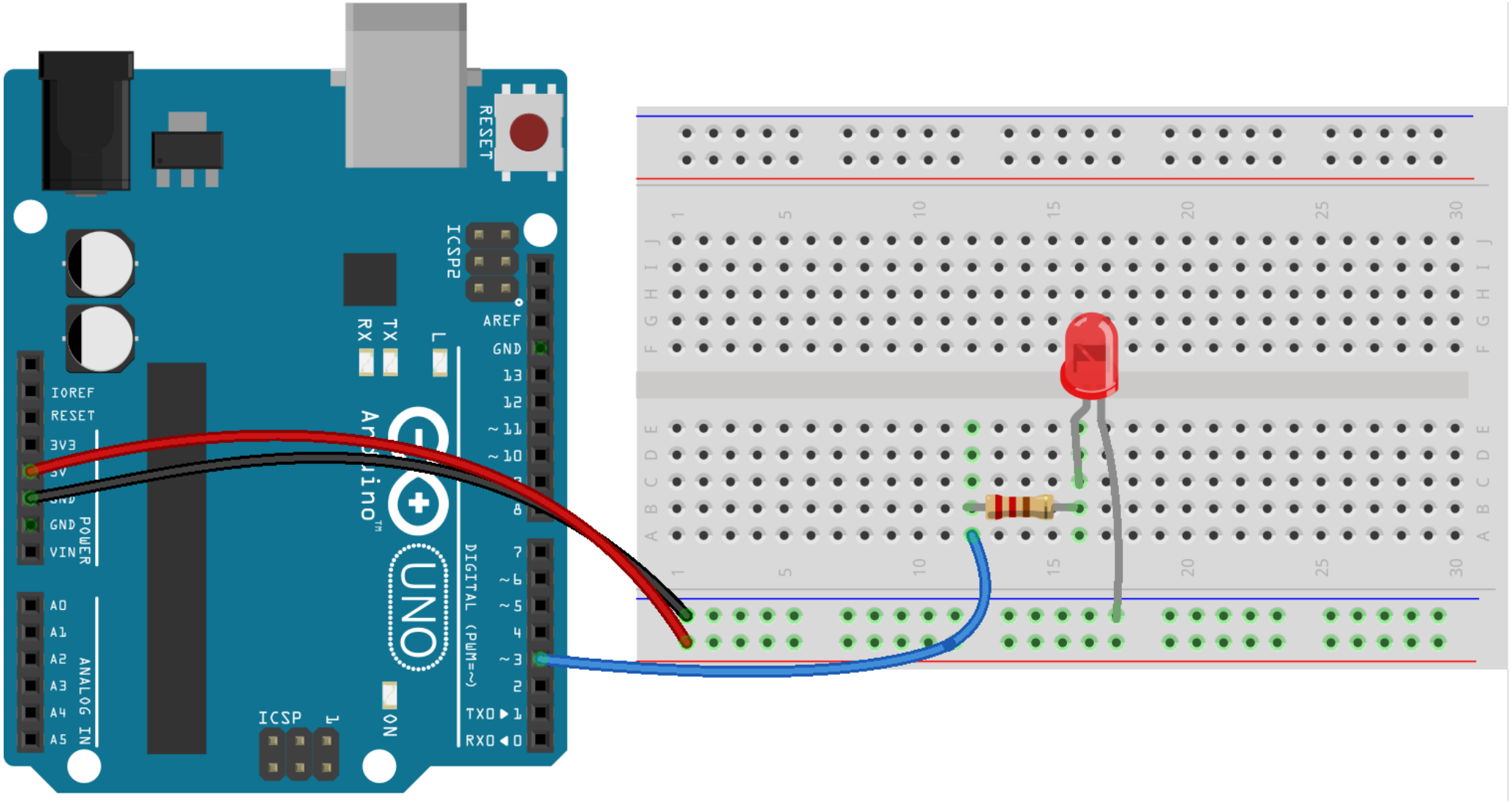
Load the Johnny-Five Package

Use NPM to install the module.

```
> npm install johnny-five
```

Build the following circuit on your breadboard

Connect the LED to Pin 3



Add the Code for the LED

Open the corresponding file and add the following code (shown in blue).

```
var five = require("johnny-five"),
board = new five.Board();
var ledPin = 3;

board.on("ready", function() {
    // Create an Led on pin 3
    var led = new five.Led(ledPin);

    // Strobe the pin on/off, defaults to 100ms phases
    led.strobe();
});
```

wk8/strobe.js

Run the “wk8/strobe.js”

You should see your light flashing.

```
> node strobe.js
```

Run the “wk8/led_demo.js” & “wk8/led_animation.js”

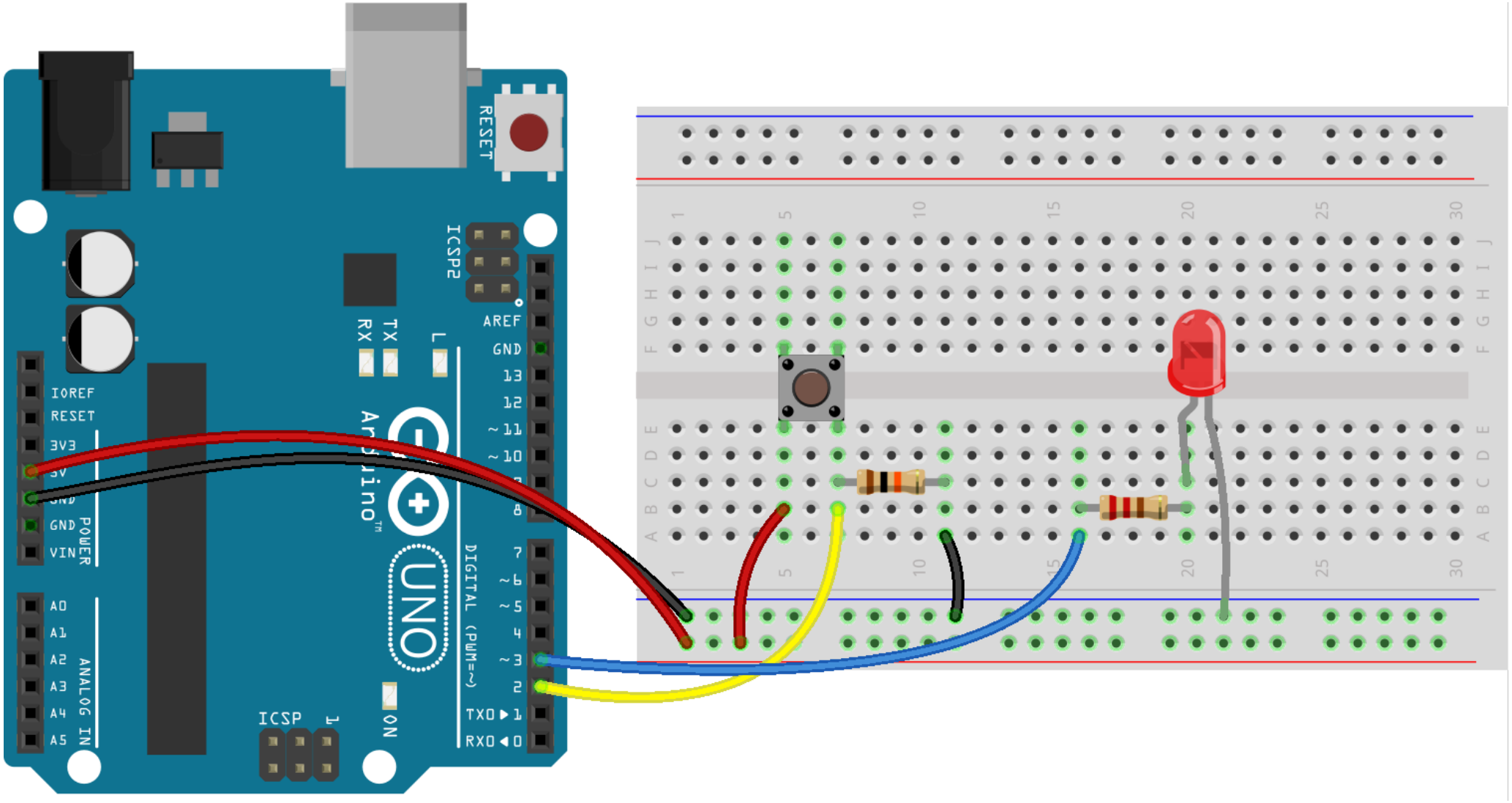
These are from the johnny-five documentation and show some of the LED control with johnny-five.

```
> node led_demo.js
```

```
➢ node led_animation
➢ .js
```

Build the following circuit on your breadboard

Connect the LED to Pin 3 and the Button to Pin 2.



Add the Code for The Push Button and LED

Open the corresponding file and add the following code (shown in blue).

```
var five = require("johnny-five");
var board = new five.Board();
var ledPin = 3;
var btnPin = 2;
board.on("ready", function () {
    var led = new five.Led(ledPin);
    var btn = new five.Button(btnPin);
    // See if the button has been pressed
    btn.on("down", function () {
        led.on();
    });
    btn.on("up", function () {
        led.off();
    });
});
```

wk8/pushbutton.js

Run the “wk8/pushbutton.js”

You should see your light flashing.

```
> node pushbutton.js
```

Add the following code to the the wk8/app.js app

Open the corresponding file and add the following code (shown in blue).

```
// ADD SOCKET.IO CODE BELOW

io.on('connection', function (socket) {
    console.log("user connected to server");
    socket.on('buttonval', function (data) {
        console.log(data);
        if (data == 1) {
            led.on();
        }
        else {
            led.off();
        }
    });
});

// ADD SOCKET.IO CODE ABOVE
```

wk8/app.js

Run the “wk8/app.js”

Open your browser to http://localhost:3000

```
> node app.js
```

In-Class Challenge(s)

Working in your assignment #2 groups, you will need to complete at least 3 challenges by the end of class.

Challenge #1 – Distance Sensor with Indicators.

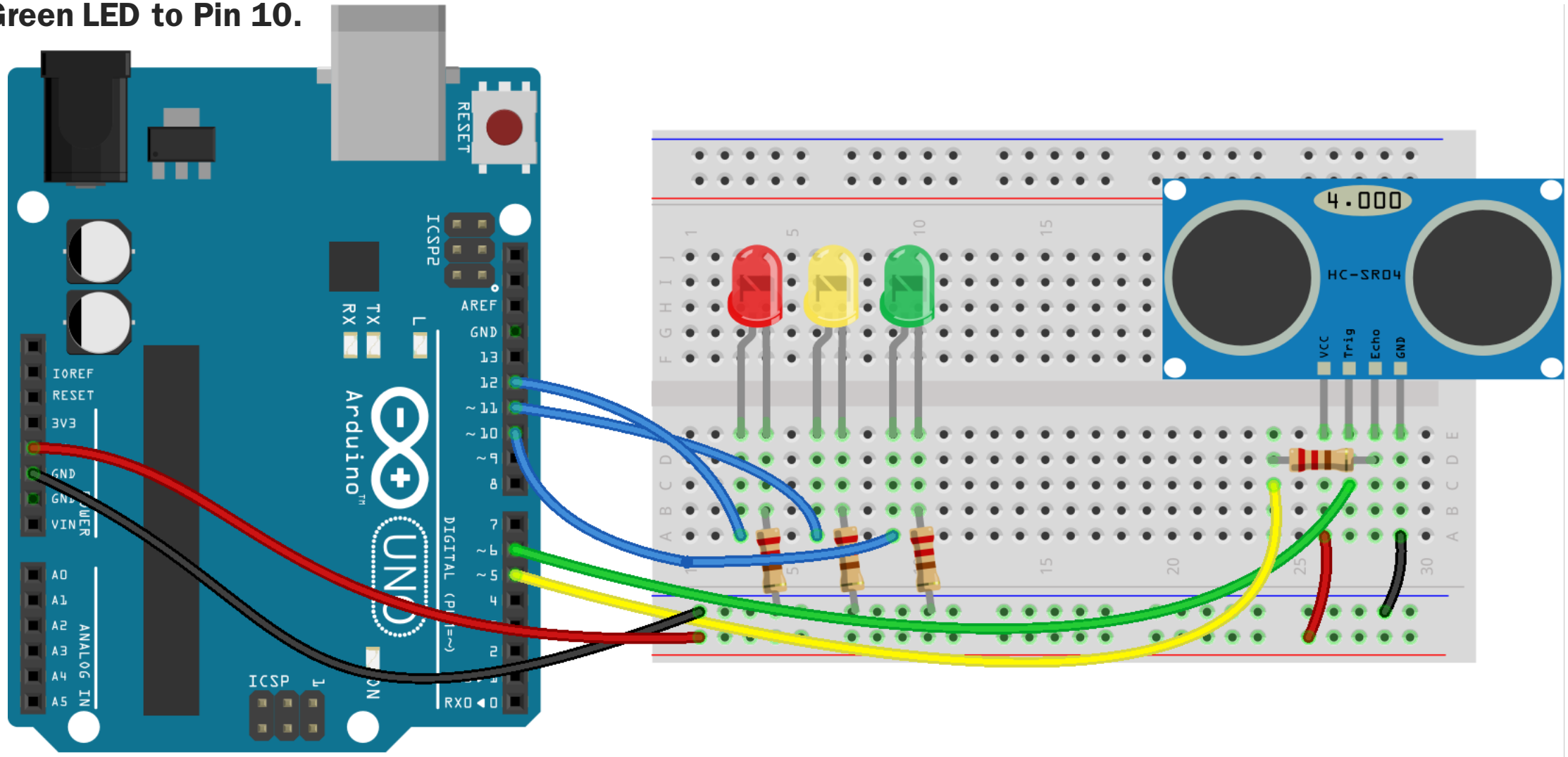
Build the circuit on the next page. You will monitor the distance using the HC-SR04 sensor. You will send the distance value to your webserver and display the distance on a webpage. In addition, your web server will control which LED is on. If the distance is above 30cm, the green LED should be on. If the value is between 30cm and 15cm, the yellow LED should be on. If the distance is below 15cm, the red LED should turn on. Only one LED should be on at any time.

NOTE: It is advised that you use 'serialport'. You will need to load the 'new_ping' library for your Arduino.

Sample Files are available in 'wk8/Challenges/Distance_Sensor_with_Indicator'

Challenge #1 – Build the following circuit

The HC-SR04 sensor is connected to Pins 5 & 6. The Red LED to Pin 12, the Yellow LED to Pin 11, and the Green LED to Pin 10.



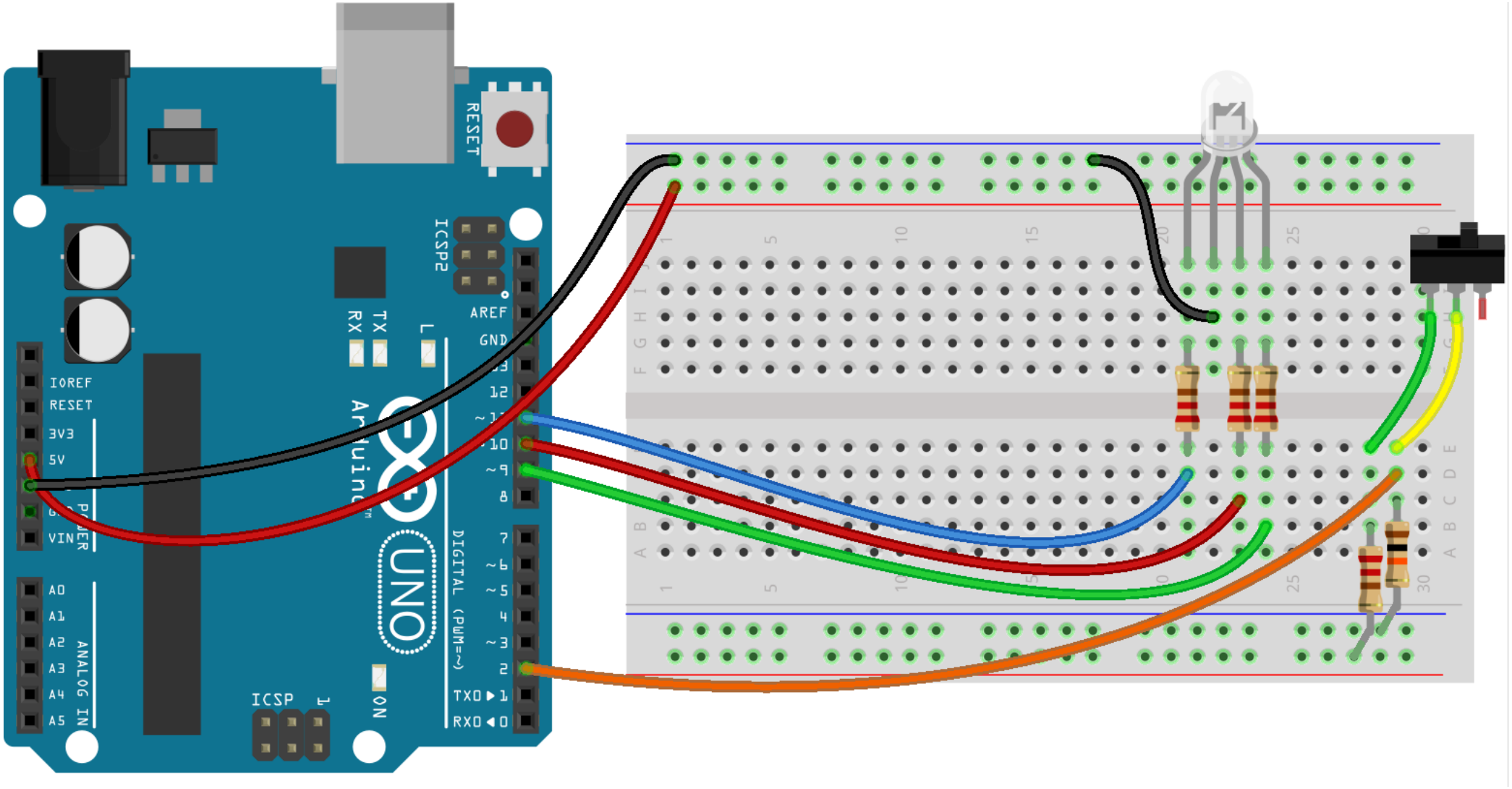
Challenge #2 – Colour Wheel LED.

Build the circuit on the next page. You will create a webpage with red, green & blue sliders. As you make changes to the slider on the webpage, it will change the colour of the LED on your breadboard. This will only work if the switch is set to ON. If the switch is set to OFF, the RGB sliders will be greyed out on your webpage.

NOTE: Make sure you connect your RGB LED pins to Arduino pins with a (~) and are capable of PWM.

Challenge #2 – Colour Wheel LED

NOTE: Each of the pins to the RGB LED must be to an Arduino Pin with a ~



Challenge #3 – Music Box.

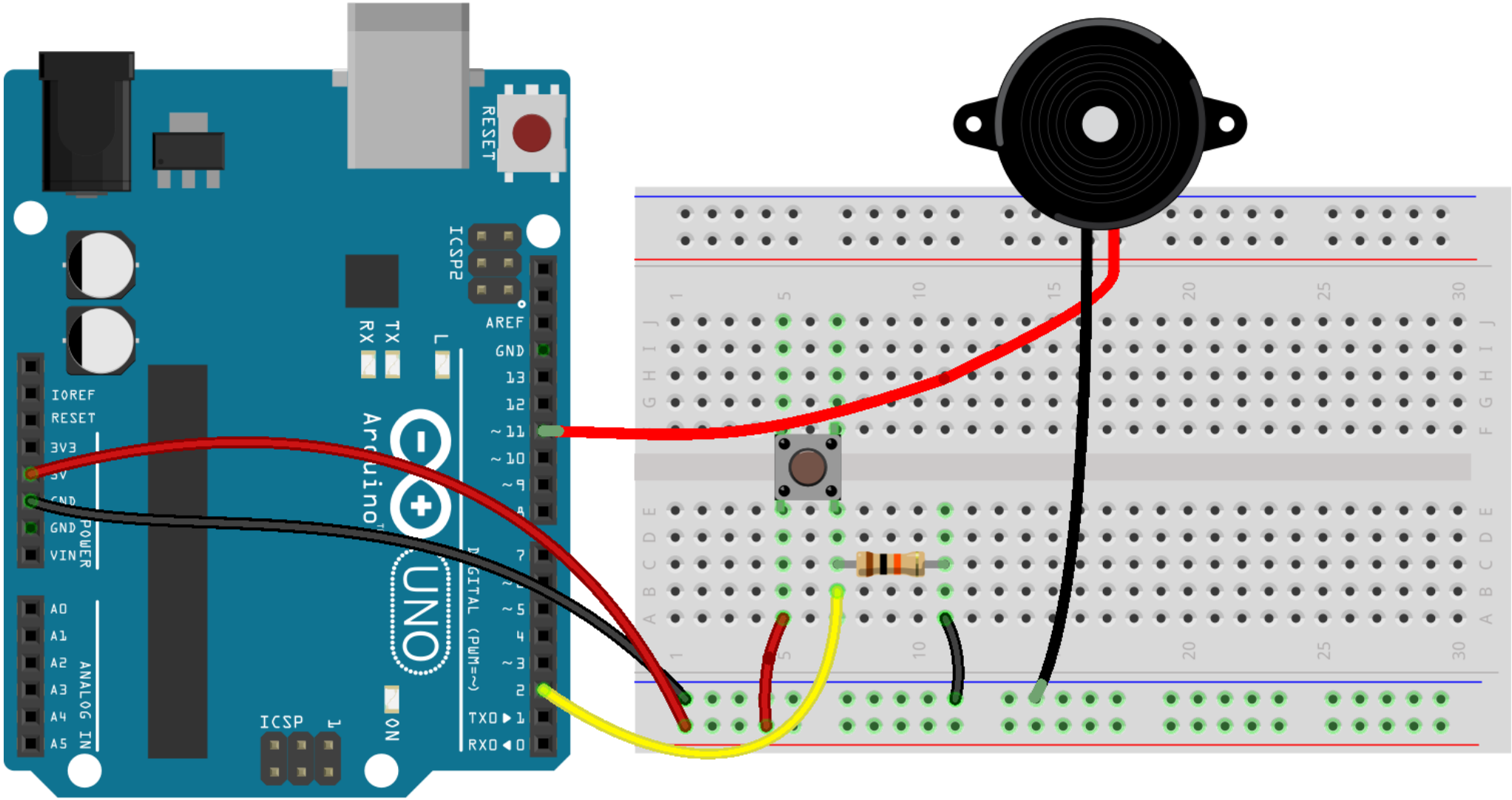
Build the circuit on the next page. You will create a webpage that displays a dropdown with the available songs you can play. The user will select a song with the webpage. When a user presses the button on the breadboard, the song will play.

NOTE: You will need to load the 'j5-songs' module. Look inside this package to find the song titles.

Sample Files are available in 'wk8/Challenges/Music_Box'

Challenge #3 – Music Box

NOTE: Buzzer connected to Pin 11, Push Button connected to Pin 2.

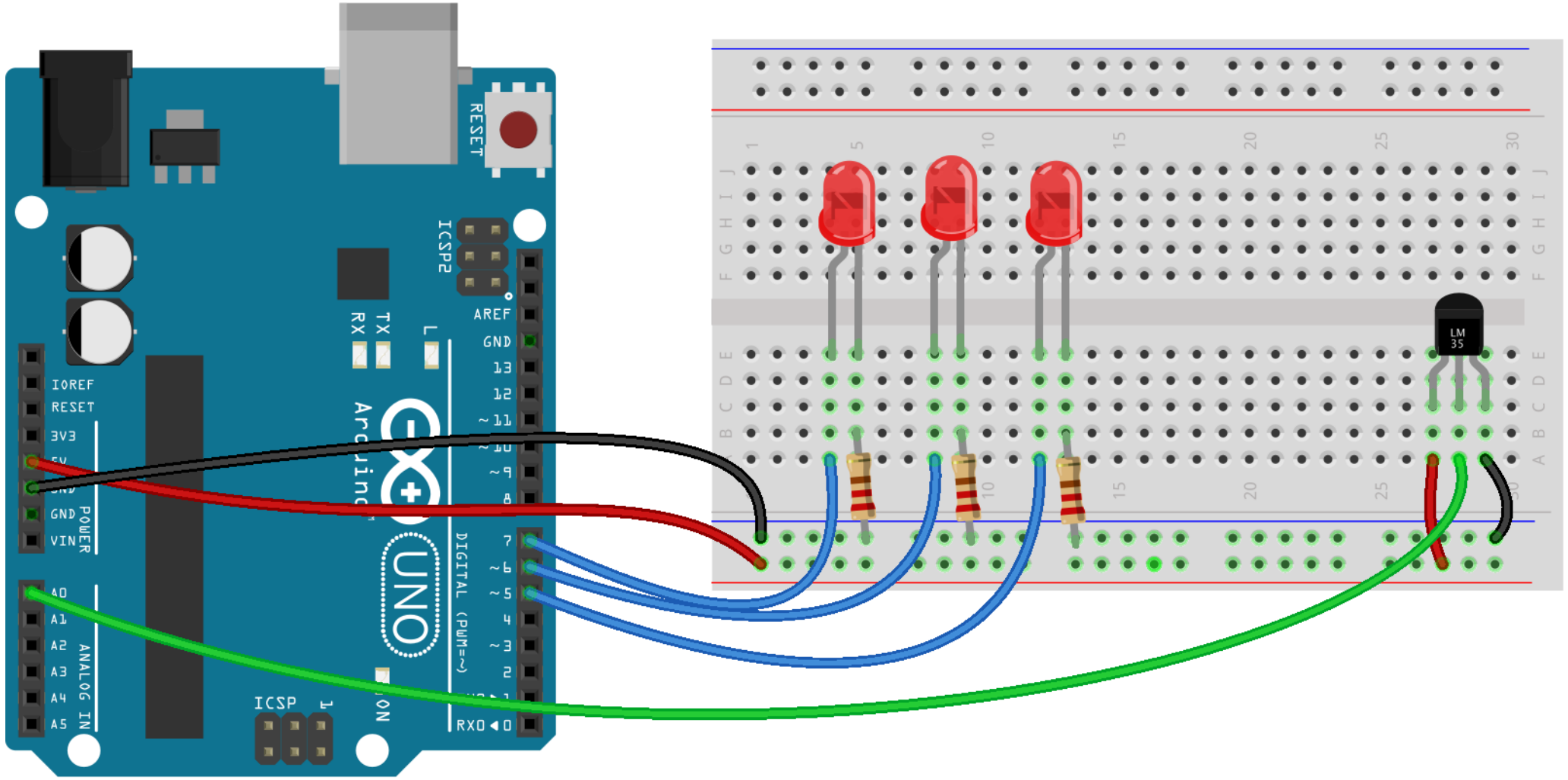


Challenge #4 – Turn Up the Heat.

Build the circuit on the next page. You will create a webpage that reads the temperature from the sensor and displays it on the screen. When the program starts, take the baseline temperature; you will use this for all other calculations. Turn on an LED for every 2 degree increase in temperature.

Challenge #4 – Turn Up the Heat

NOTE: Temperature Sensor connected to Pin A0, LED's to Pins 5,6,7.

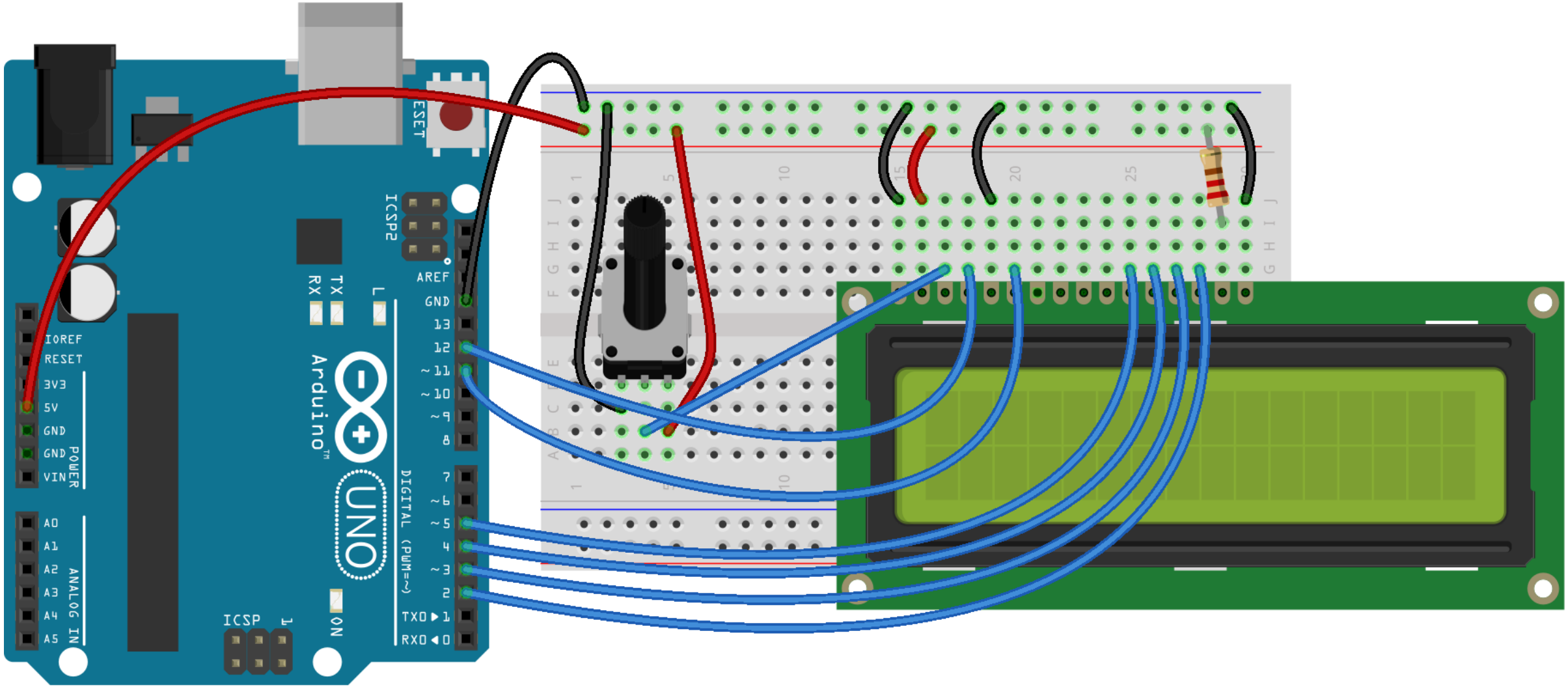


Challenge #5 – Fortune Teller.

Build the circuit on the next page. You will create a website that has one large red button in the middle of the page. A user will ask the website a question and press the button for the answer. After clicking the button, the LCD screen connected to the breadboard will show one of eight customized responses.

Challenge #5 – Fortune Teller

NOTE: Ummmmm…There are a lot of wires…pay attention! POT is connected directly to LCD Brightness.



Challenge #6 – Wheelie.

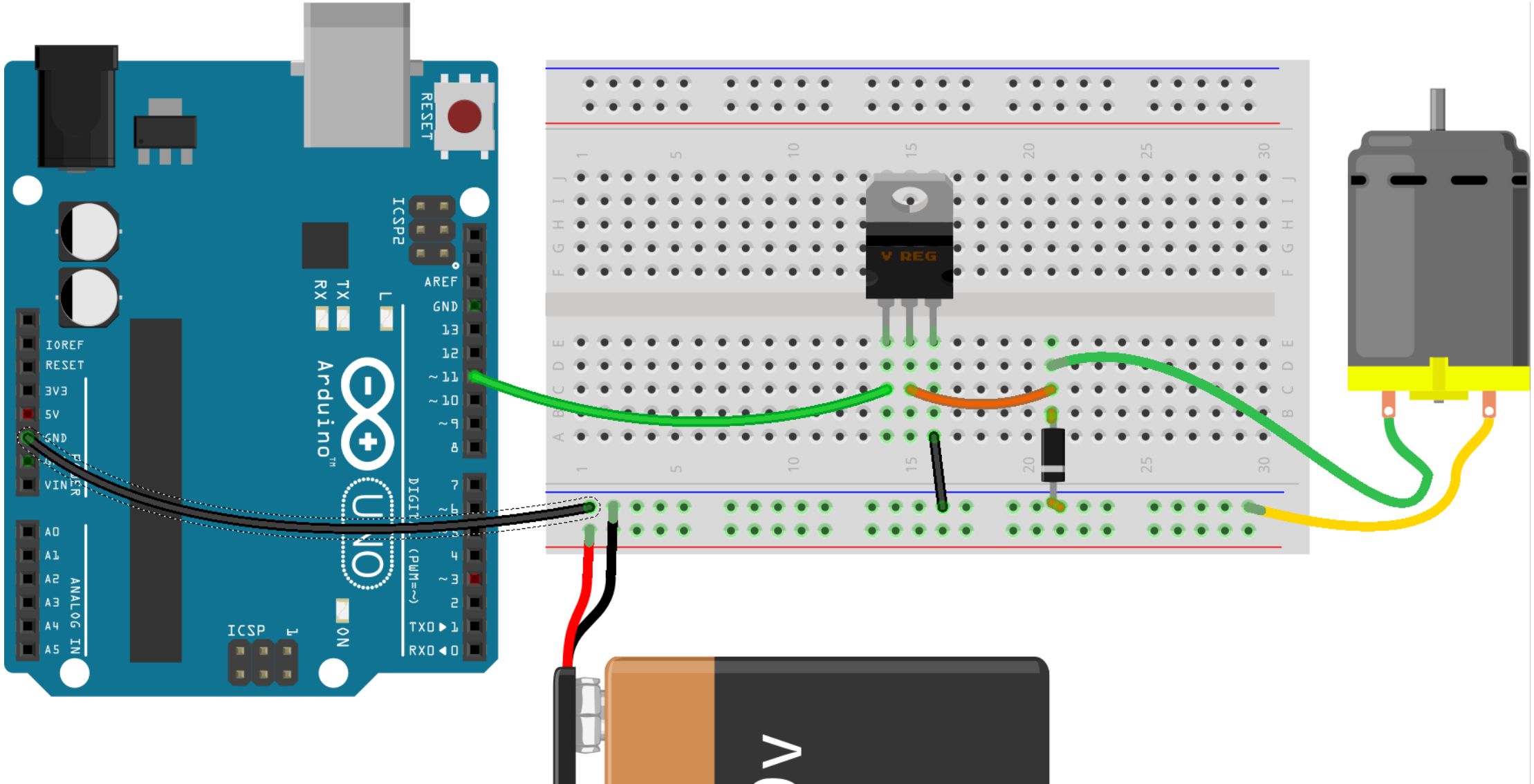
Build the circuit on the next page. You will create a website that has an ON / OFF switch. Changing the switch will turn the motor ON or OFF.

NOTE: It is important to insert the diode in the circuit as shown. Diode's have a + and – terminal.

Sample Files are available in 'wk8/Challenges/Wheelie'

Challenge #6 – Wheelie

NOTE: Make sure the Dioide is positioned exatly as diagram. Diodes have a – and + terminal.



Challenge #7 – Lord of the Light.

Build the circuit on the next page. You will create a website that has an image of the Lord of the Light on it. Depending on the light values detected by the photocell, it will control the opacity of the image on the website. More light means the image has a higher opacity.

Challenge #7 – Lord of the Light

NOTE: This one is pretty easy…