ARDUINO
OPEN-SOURCE
COMMUNITY

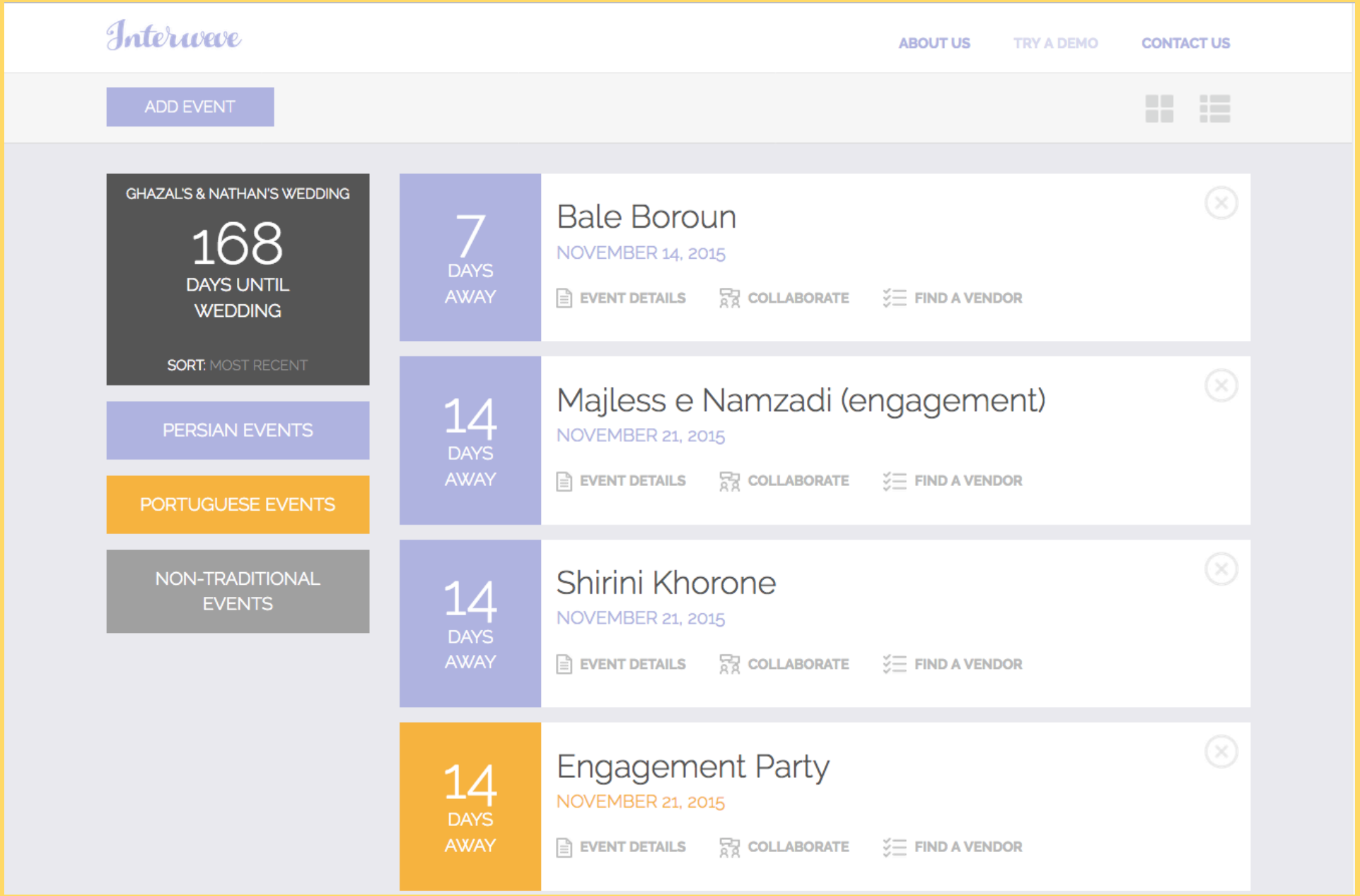
INTERACTING WITH ARDUINO

Agenda

- Last Week
- IoT and Arduino
- Serial Port
- Reading Data
- Writing Data
- In-Class Challenge

What did we cover two weeks ago?

What was the significance?


Interweave
ABOUT US
TRY A DEMO
CONTACT US
ADD EVENT
GHAZAL'S & NATHAN'S WEDDING
168
DAYS UNTIL WEDDING
SORT: MOST RECENT
PERSIAN EVENTS
PORTUGUESE EVENTS
NON-TRADITIONAL EVENTS
7 DAYS AWAY
Bale Boroun
NOVEMBER 14, 2015
EVENT DETAILS
COLLABORATE
FIND A VENDOR
14 DAYS AWAY
Majless e Namzadi (engagement)
NOVEMBER 21, 2015
EVENT DETAILS
COLLABORATE
FIND A VENDOR
14 DAYS AWAY
Shirini Khorone
NOVEMBER 21, 2015
EVENT DETAILS
COLLABORATE
FIND A VENDOR
14 DAYS AWAY
Engagement Party
NOVEMBER 21, 2015
EVENT DETAILS
COLLABORATE
FIND A VENDOR

Inheriting Code – A Learning Experience?

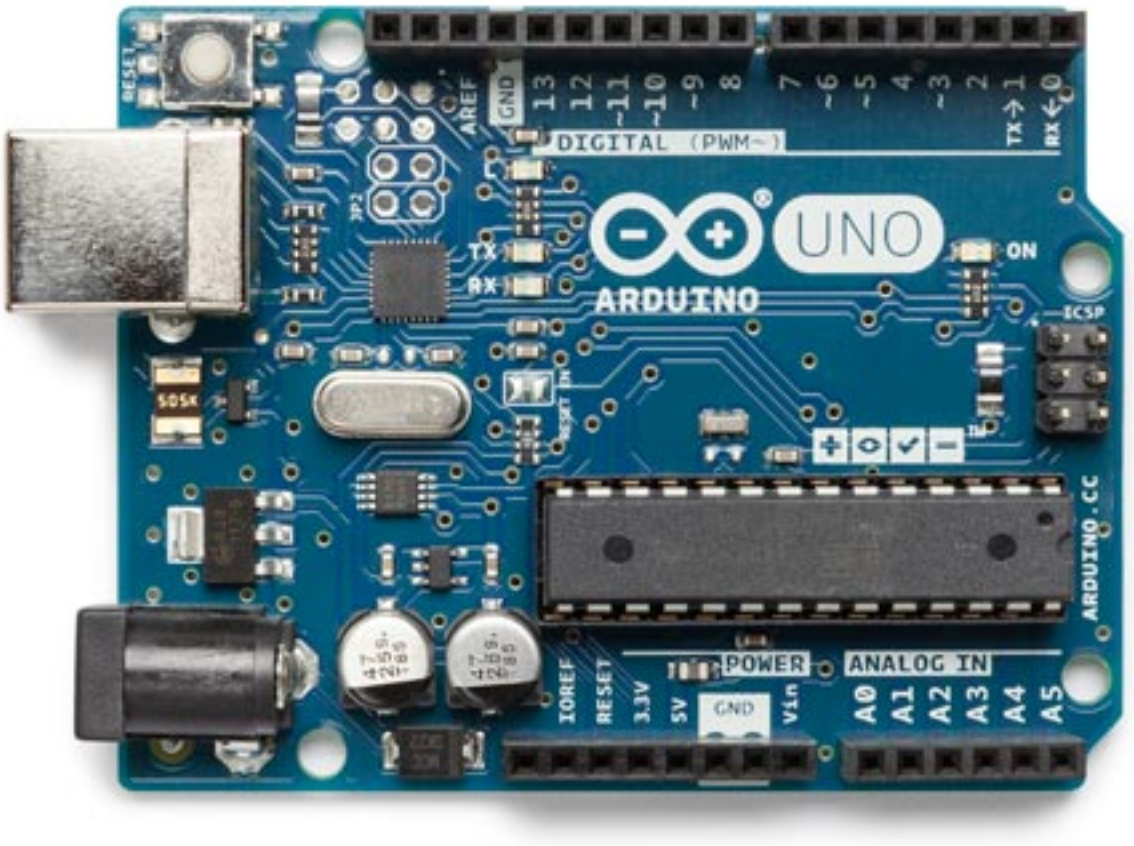
It is inevitable that you will inherit code. This can be challenging, but a very good way to learn.

Beyond screens

Talking to an Arduino using Node.js

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.



Install the serialport module

Use NPM to install the module.

```
> npm install serialport
```

Run the SerialPort.list app

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

```
var SerialPort = require('serialport');

SerialPort.list(function (err, ports) {
    ports.forEach(function (port) {
        console.log(port.comName);
    });
});
```

wk7-01/app.js

Running the program will show this...

Note the names; this is the address of your Arduino. You will need this for the next step.

```
/dev/tty.Bluetooth-Incoming-Port
/dev/tty.usbmodem11
```

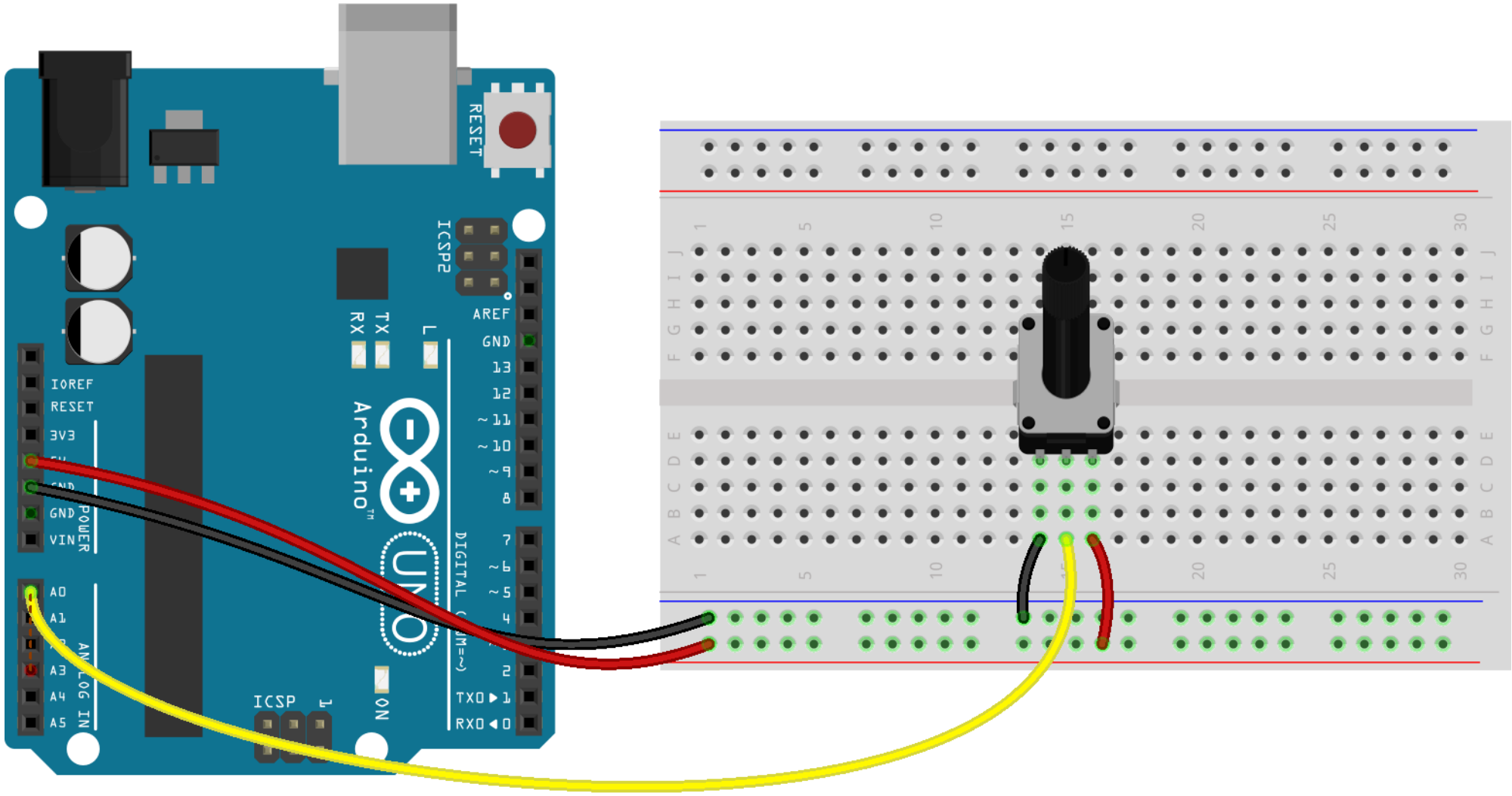
wk7-01/app.js

PC & Mac will have different addresses for the Arduino

PC will probably list COM5 (or something similar)

Build the following circuit on your breadboard

Connect the Potentiometer to A0



Load the following code to your Arduino

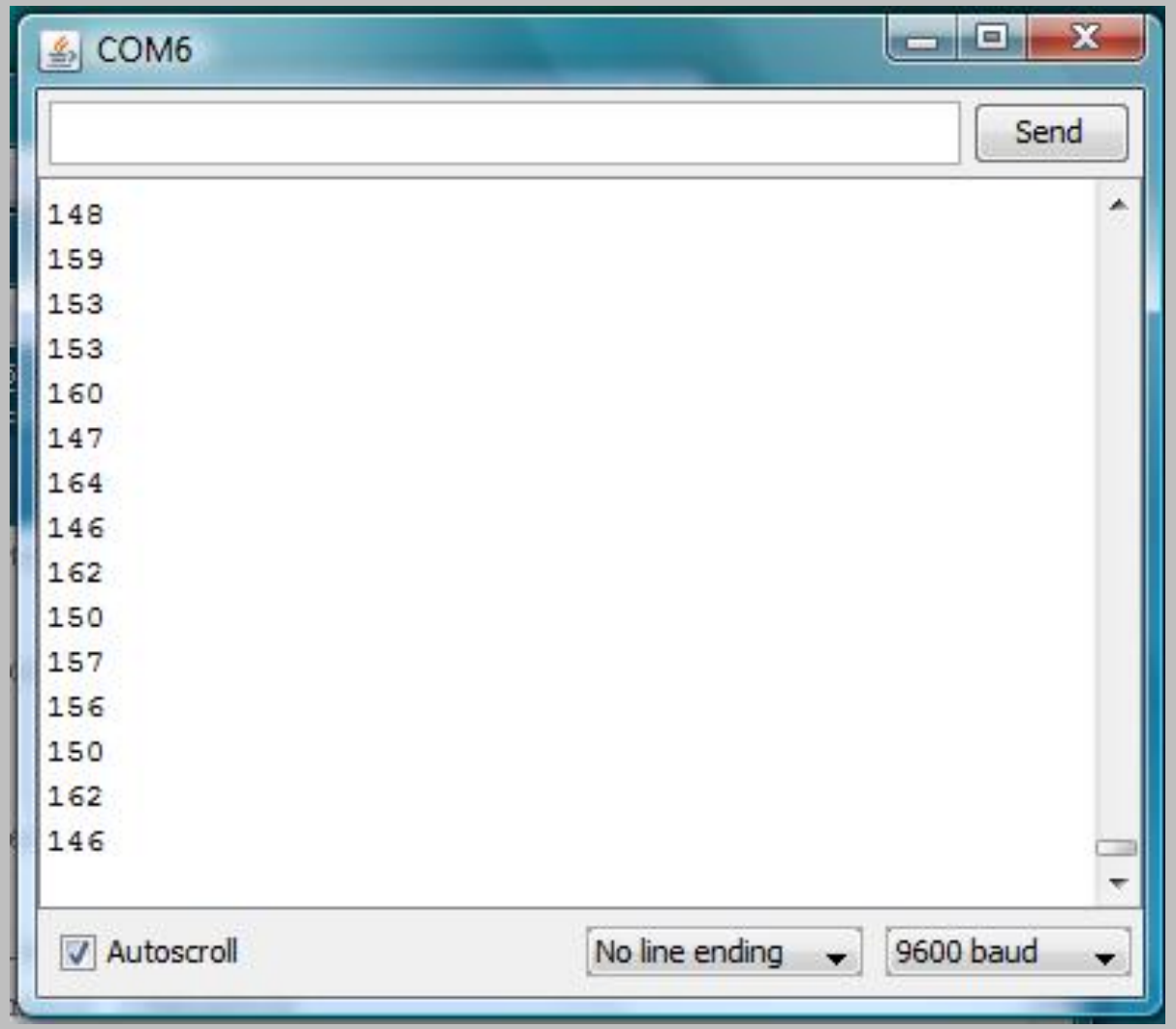
Open the corresponding file and upload it to your board.

```
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(0);
  // map the value between 0 to 100 :
  float val = map(sensorValue,0,1023,0,100); // note the map function
  // print out the value
  Serial.println(val);
}
```

arduino_code/analogInput/analogInput.ino

Check that your circuit works in the Serial Monitor

When you turn the dial to the left the value should approach 0. When you turn the dial to the right it should approach 100. If it is reversed, unplug your Arduino and swap the GND and 5V wires to the potentiometer.



arduino_code/analogInput/analogInput.ino

Add the following code to the wk7-02/app.js app

```
var SerialPort = require('serialport');
var port = new SerialPort('/dev/tty.usbmodem11', {
    baudRate: 9600,
    dataBits: 8,
    parity: 'none',
    stopBits: 1,
    flowControl: false,
    parser: SerialPort.parsers.readline("\r\n")
});

port.on('data', function (data) {
    console.log('Data: ' + data);
});
```

wk7-02/app.js

Run the “wk7-02/app.js”

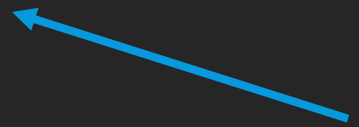
You should see values being logged to the console.

```
> node app.js
```

If you didn't see the console logging values...

Check the name of the SerialPort you are trying to connect to.

```
var SerialPort = require('serialport');
var port = new SerialPort('/dev/tty.usbmodem11', {
    baudRate: 9600,
    dataBits: 8,
    parity: 'none',
    stopBits: 1,
    flowControl: false,
    parser: SerialPort.parsers.readline("\r\n")
});

port.on('data', function (data) {
    console.log('Data: ' + data);
});
```



Make sure this matches the name from the wk7-01/app.js

wk7-02/app.js

Let’s make some changes to our code

In case you have problems connecting with future code, here are some tips

Make Edits to your Arduino Code

Change the baud rate and upload it to your board.

```
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(115200);
}

void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(0);
  // map the value between 0 to 100 :
  float val = map(sensorValue,0,1023,0,100); // note the map function
  // print out the value
  Serial.println(val);
}
```

arduino_code/analogInput/analogInput.ino

The wk7-02/app.js doesn't work anymore...

Because the baud rates aren't the same.

```
var SerialPort = require('serialport');
var port = new SerialPort('/dev/tty.usbmodem11', {
    baudRate: 9600,
    dataBits: 8,
    parity: 'none',
    stopBits: 1,
    flowControl: false,
    parser: SerialPort.parsers.readline("\r\n")
});

port.on('data', function (data) {
    console.log('Data: ' + data);
});
```

Make sure this matches the baud in your Arduino sketch!

wk7-02/app.js

Let’s make some changes to our code

In case you have problems connecting with future code, here are some tips

Make Edits to your Arduino Code

Add a serial.print, serial.println, and delay to the sketch and upload it to your board.

```
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(0);
  // map the value between 0 to 100 :
  float val = map(sensorValue,0,1023,0,100); // note the map function
  // print out the value
  Serial.println(val);
  Serial.print("Hello");
  Serial.println("There");
  delay(2000);
}
```

arduino_code/analogInput/analogInput.ino

What happens now?

The data is sent similar to how it is displayed in the Serial Monitor

Load the following code to your Arduino

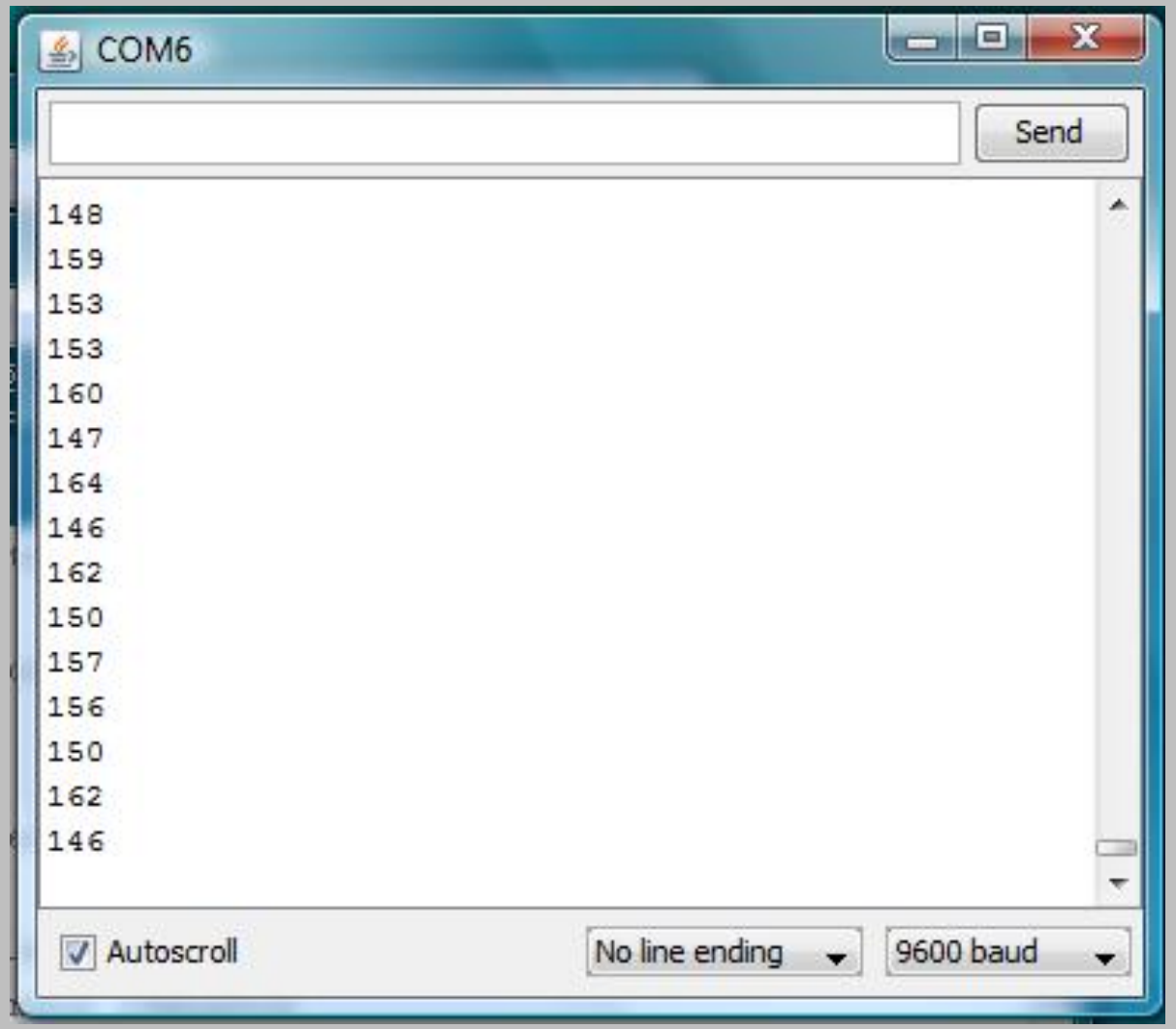
Open the corresponding file and upload it to your board.

```
int sensorValue = 0;        // value read from the potentiometer
int prevValue = 0;          // previous value from the potentiometer

void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

void loop() {
  // read the input on analog pin 0:
  sensorValue = analogRead(0);
  if (prevValue != sensorValue) {
    // map the value between 0 to 100 :
    int val = map(sensorValue, 0, 1023, 0, 100); // note the map function
    // print out the value :
    Serial.println(val);
    prevValue = sensorValue;
  }
  delay(100);
}
```

arduino_code/analogInput/analogInput_better.ino

Check that your circuit works in the Serial Monitor

The values should only be printed in the Serial Monitor when you turn the potentiometer.



arduino_code/analogInput/analogInput.ino

Now we only send data through the serial port if the value changes

You can use delay in the Arduino sketch to set how often the sketch checks the sensor

Add the following code to the wk7-03/app.js app

```
// CODE FOR SERIAL COMMUNICATION BELOW HERE

// Open the connection to the serial port
port.on('open', function () {
    console.log("Open Serial Communication");
    // Handles incoming data
    port.on('data', function (data) {
        console.log('Data: ' + data);
        io.emit('updateData', data);
    });
});

// CODE FOR SERIAL COMMUNICATION ABOVE HERE
```

wk7-03/app.js

Run the “wk7-03/app.js”

You should see values being logged to the console.

```
> node app.js
```

Open your browser to localhost:3000

Turn your potentiometer and you should see the knob on the website moving! Yas!

Make Edits to Your Arduino Code

Adjust the mapped values of the potentiometer and upload it to your board. Also try 0 and 360.

```
int sensorValue = 0;        // value read from the potentiometer
int prevValue = 0;          // previous value from the potentiometer

void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

void loop() {
  // read the input on analog pin 0:
  sensorValue = analogRead(0);
  if (prevValue != sensorValue) {
    // map the value between 0 to 100 :
    int val = map(sensorValue, 0, 1023, -100, 100); // note the map function
    // print out the value :
    Serial.println(val);
    prevValue = sensorValue;
  }
  delay(100);
}
```

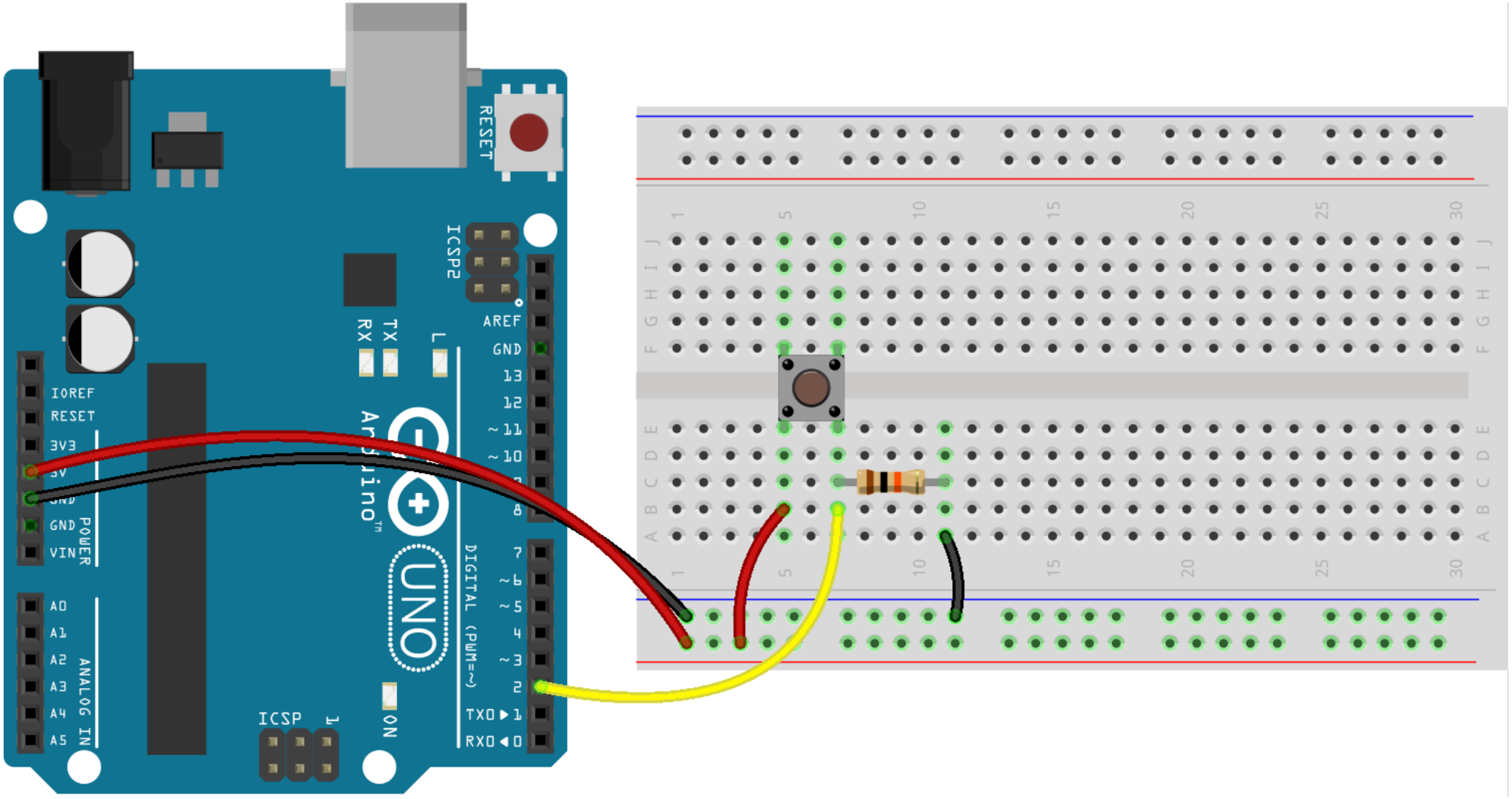
arduino_code/analogInput/analogInput_better.ino

Now let’s look at sending and receiving data

We will make a circuit with a button and an LED.

Build the following circuit on your breadboard

Connect the pushbutton to Pin 2.



Load the following code to your Arduino

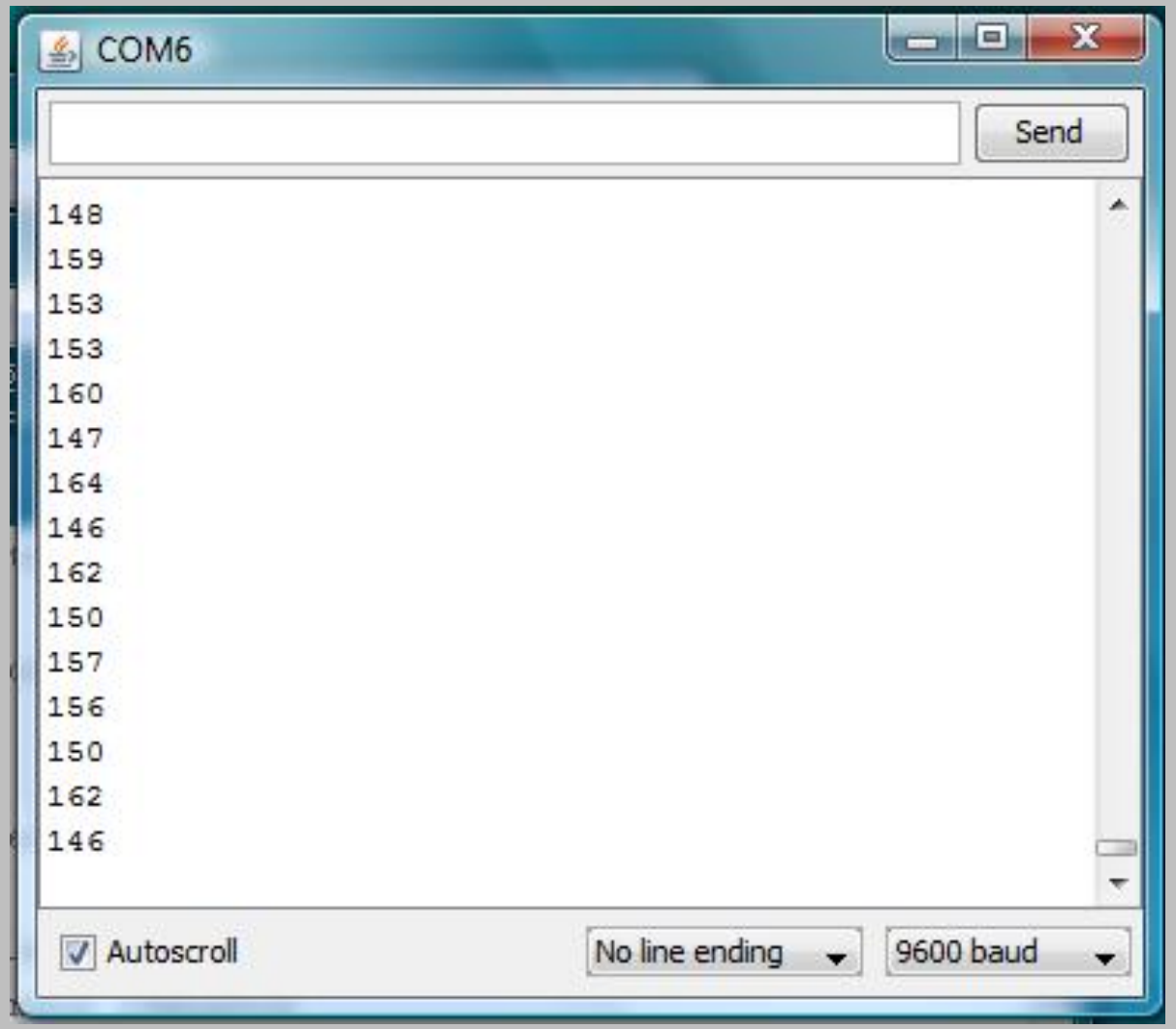
Open the corresponding file and upload it to your board.

```
int buttonState = 0;  // value read from the button
int prevState = 0;    // previous value from the button
int pushButton = 2;   // push button connected to pin 2

void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  pinMode(pushButton, INPUT);
}

void loop() {
  buttonState = digitalRead(pushButton);
  // check previous value vs current value
  if (prevState != buttonState) {
    if (buttonState == HIGH) {
      Serial.println(1);
    } else {
      Serial.println(0);
    }
    prevState = buttonState;
  }
  delay(100);
}
```

arduino_code/analogInput/pushButton.ino

Check that your circuit works in the Serial Monitor

When you press the button, you should see a 1 in the Serial Monitor. When you release the button, a 0 will appear in the Serial Monitor.



arduino_code/analogInput/analogInput.ino

Add the following code to the index.html page

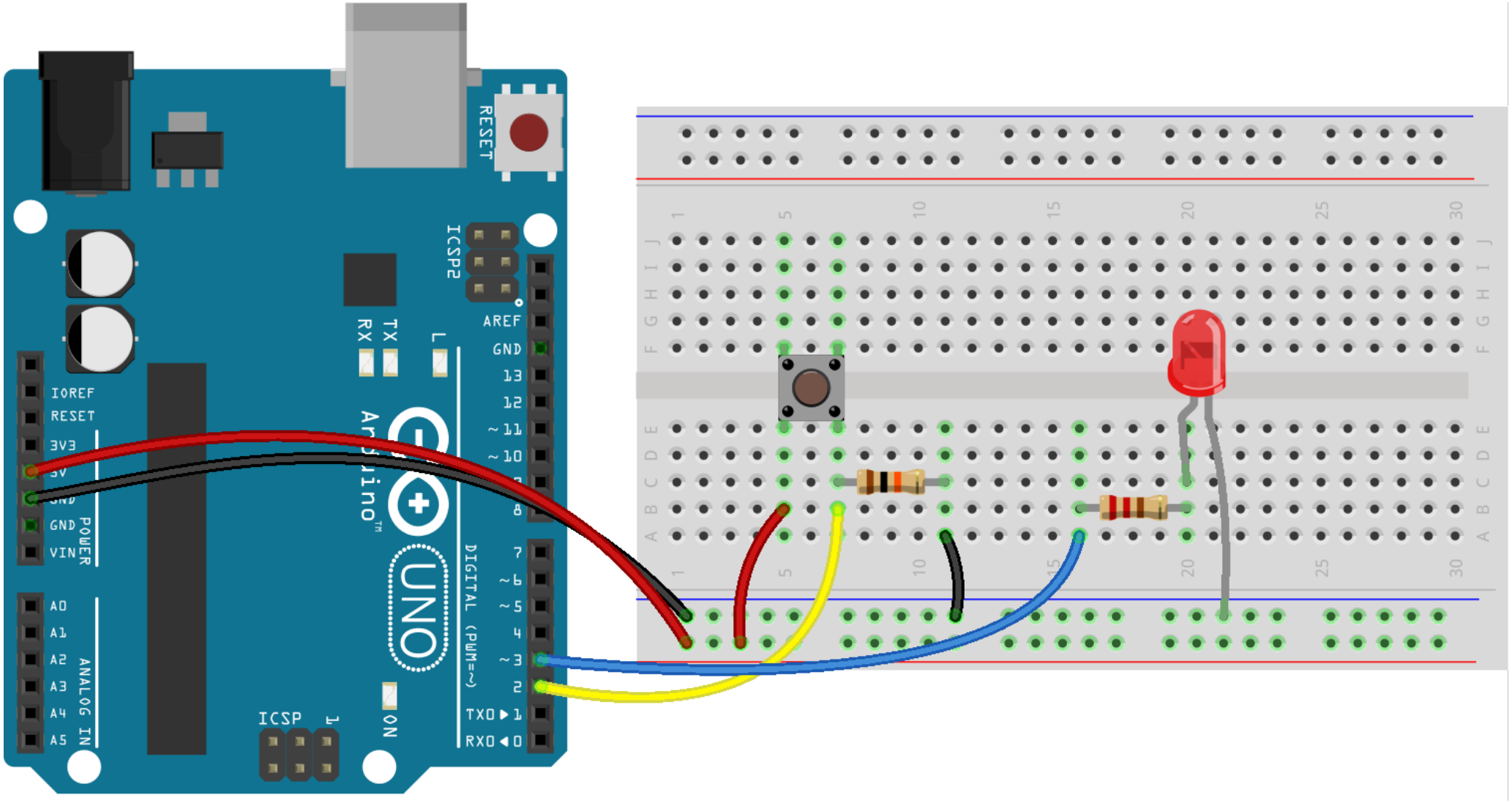
Open the corresponding file and add the following code (shown in blue).

```
// CODE FOR DATA FROM THE SERVER BELOW HERE

socket.on('updateData', function (receivedData) {
   console.log(receivedData);
   if(receivedData == 1){
      $( "#myonoffswitch" ).prop( "checked", true );
   } else {
      $( "#myonoffswitch" ).prop( "checked", false );
   }
});

// CODE FOR DATA FROM THE SERVER ABOVE HERE
```

wk7-04/files/index.html

Build the following circuit on your breadboard

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



Load the following code to your Arduino – 1 of 3

Adds code that reads data from the serial port.

```
// Variables for the pushButton
int buttonState = 0;         // value read from the button
int prevState = 0;           // previous value from the button
int pushButton = 2;          // Pin for the pushbutton

// LED Values
int ledPin = 3;
int lightValue = 0;          // What is the value of the light

// LED read vars
String inData = "";          // a string to hold incoming data

// 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);
  pinMode(ledPin, OUTPUT);
}
```

arduino_code/analogInput/pushButton_LED.ino

Load the following code to your Arduino – 2 of 3

Adds code that reads data from the serial port.

```
// the loop routine runs over and over again forever:
void loop() {

  // Will run code when data is being sent to the arduino
  while (Serial.available() > 0) {
    char received = Serial.read();
    inData += received;

    // Process message when new line character is received
    if (received == 'E')
    {
      lightValue = inData.toInt();
      inData = ""; // Clear buffer of received data
    }
  }
```

arduino_code/analogInput/pushButton_LED.ino

Load the following code to your Arduino – 3 of 3

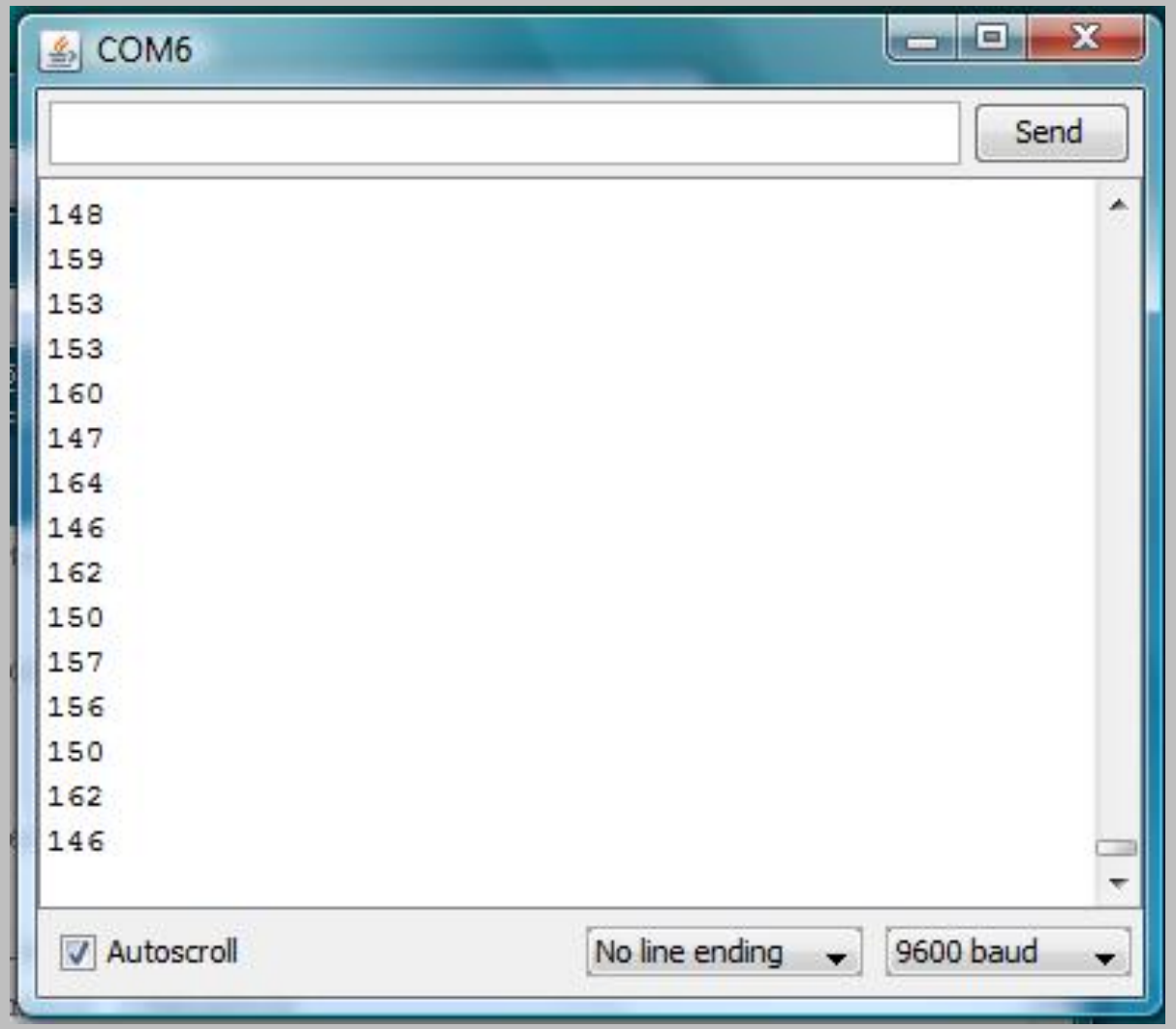
Adds code that reads data from the serial port.

```
  buttonState = digitalRead(pushButton);
  // print out the state of the button into the serial port:
  if (prevState != buttonState) {
    if (buttonState == HIGH) {
      Serial.println(1);
    } else {
      Serial.println(0);
    }
    prevState = buttonState;
  }

  //Will write the light value to the LED
  digitalWrite(ledPin, lightValue);
  // delay in between reads for stability
  delay(100);
}
```

arduino_code/analogInput/pushButton_LED.ino

Check that your circuit works in the Serial Monitor

Type “1E” and click SEND in the Serial Monitor. The LED should turn on. Turn it off by typing “0E” and click SEND.



arduino_code/analogInput/analogInput.ino

Add the following code to the the wk7-05/app.js app

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

```
// CODE FOR DATA TO SEND FROM SERVER TO SERIAL PORT BELOW

io.on('connection', function (socket) {
    console.log("user connected to server");
    socket.on('buttonval', function (data) {
        console.log(data);
        port.write(data + 'E');
    });
});

// CODE FOR DATA TO SEND FROM SERVER TO SERIAL PORT ABOVE
```

wk7-05/app.js

BREAK

See you in 10-15 minutes

In-Class Challenge (Work in Your Groups)

Combine all of the elements together into a single app.

1. A potentiometer on your breadboard controls a dial on the webpage.
2. A button on your breadboard controls a button on the webpage.
3. A dial on the webpage controls the brightness of an LED (analogWrite).
4. A button on the webpage turns and LED on or off.
5. Must create a single webpage.

Show me you have the website working before the end of class.

Build the following circuit on your breadboard

Connect the pushbutton to Pin 2, one LED to Pin 3, one LED to Pin 4, and the Potentiometer to A0.