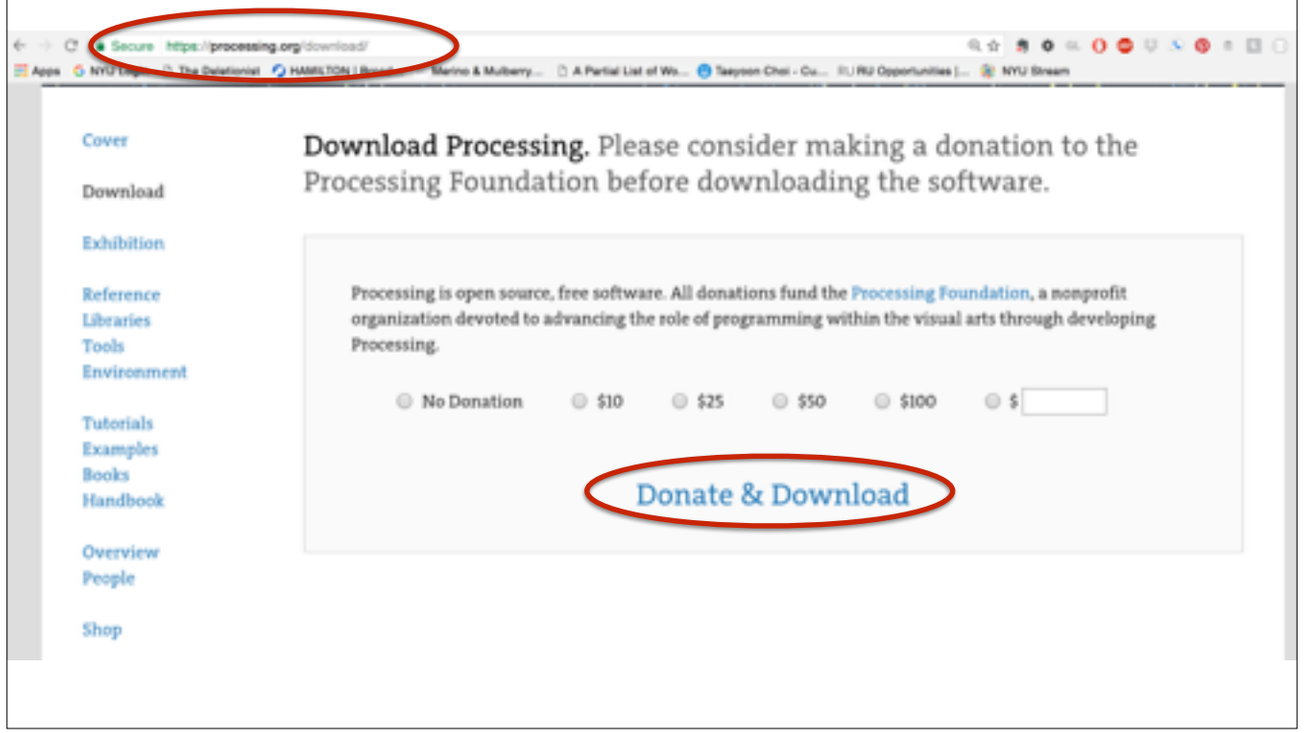
Arduino: Processing & P5.js

Arduino + Processing



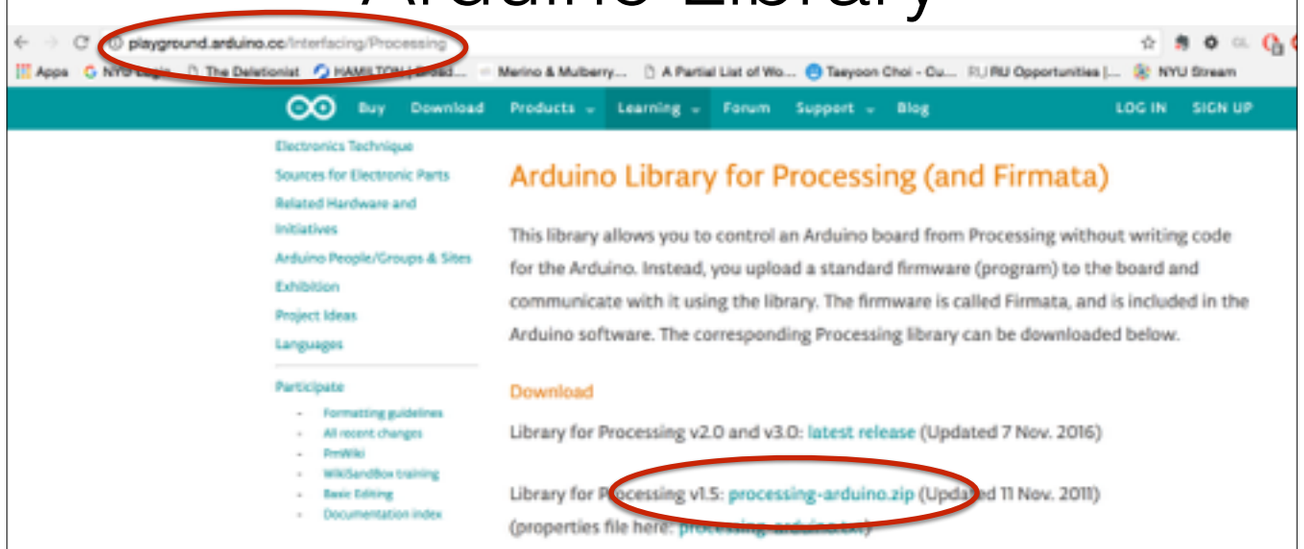
Download Processing for your laptop and Install it

Using the Serial Library

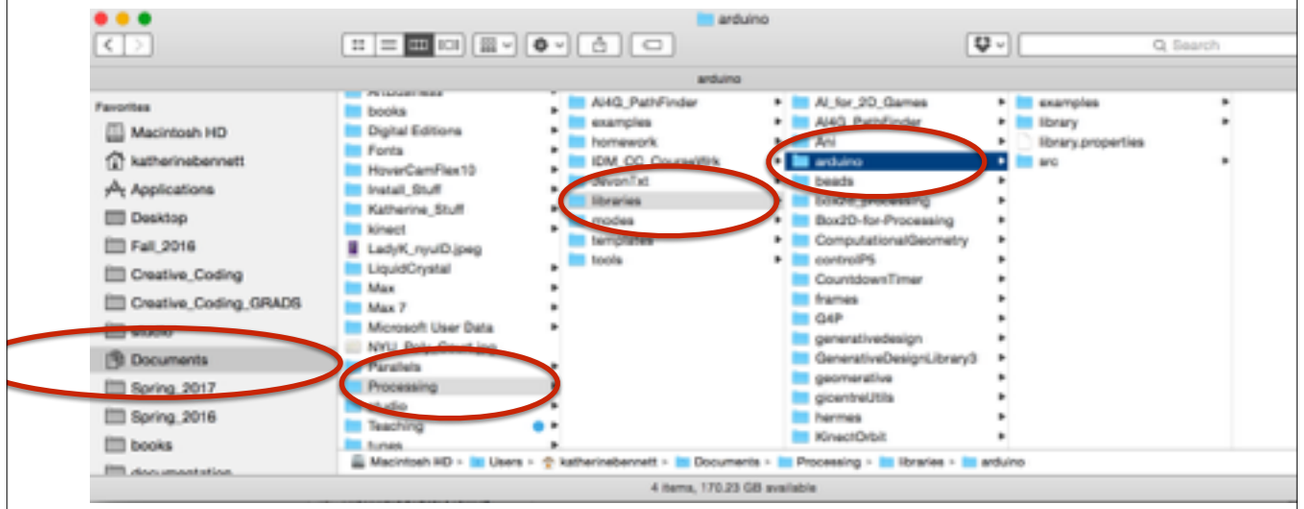
- Import Serial library
- note port name in Serial object
- read and write to the port

Another method is to use the Processing Library for the Arduino within Processing

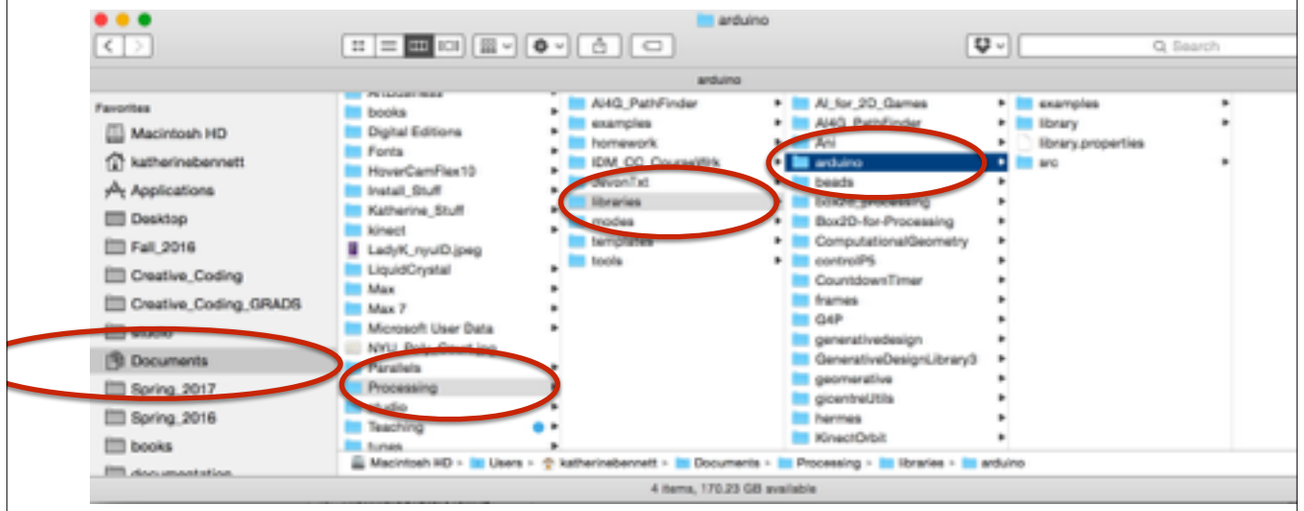
Download the Processing-Arduino Library



Now that you've downloaded Processing + installed it, look in your Documents folder or in your "c" drive for a folder called "Processing"

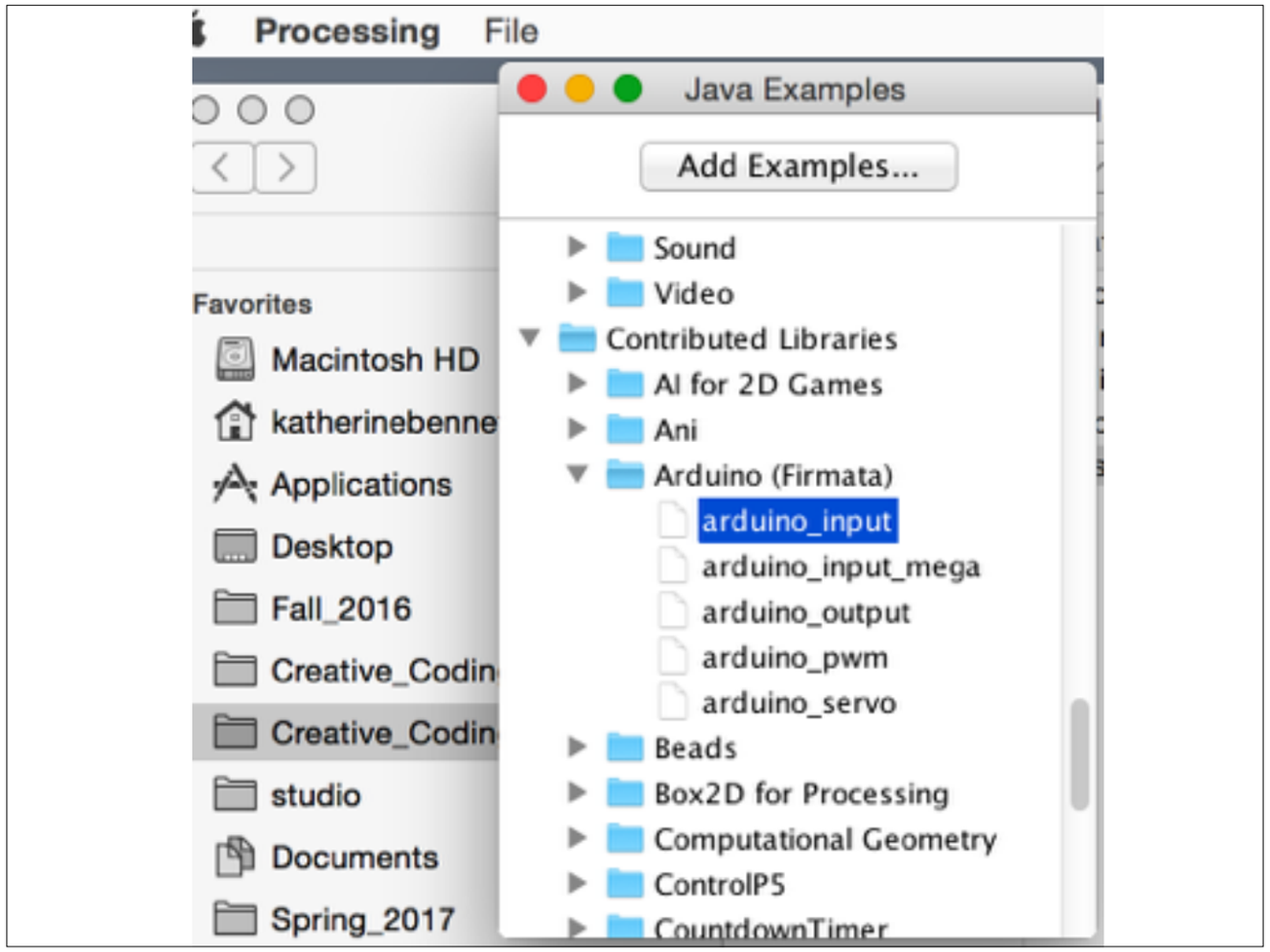


If you don't have a libraries folder, make one and drop the "arduino" file from the Processing-Arduino download inside of it.



- With the Firmata sketch already loaded onto the Arduino board, we are ready to switch to Processing.
- If the Firmata sketch is not on your Arduino, load it on there from the File/Examples/Firmata/ StandardFirmata
- Close the Arduino Application!!!!

Open up the examples that came with the arduino library for the Processing (Application)


Processing
File
Java Examples
Add Examples...
Sound
Video
Contributed Libraries
AI for 2D Games
Ani
Arduino (Firmata)
arduino_input
arduino_input_mega
arduino_output
arduino_pwm
arduino_servo
Beads
Box2D for Processing
Computational Geometry
ControlP5
CountdownTimer
Favorites
Macintosh HD
katherinebenne
Applications
Desktop
Fall_2016
Creative_Codin
Creative_Codin
studio
Documents
Spring_2017

`

- Edit the example code to select the serial port used by Arduino. Specifically, change the [0] in this line

```
arduino = new Arduino(this, Arduino.list()[0], 57600);
```

- To find the correct item in the array, run this code in Processing:

```
import processing.serial.*;
import cc.arduino.*;
println(Arduino.list());
```

- The output window will enumerate your serial ports. Select the number corresponding to the serial port in your Arduino environment found under Tools > Serial Port.

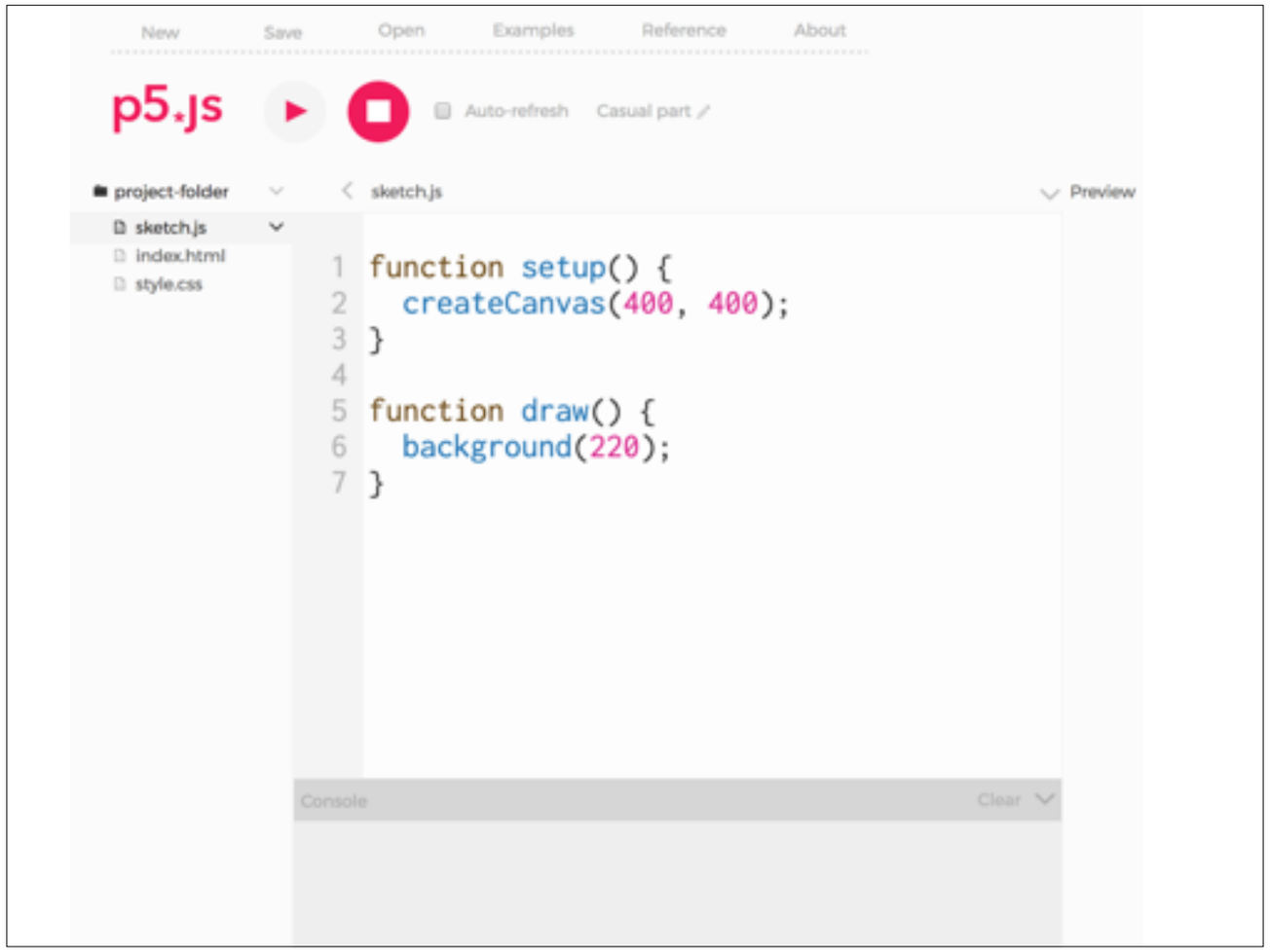
Arduino:
P5.js

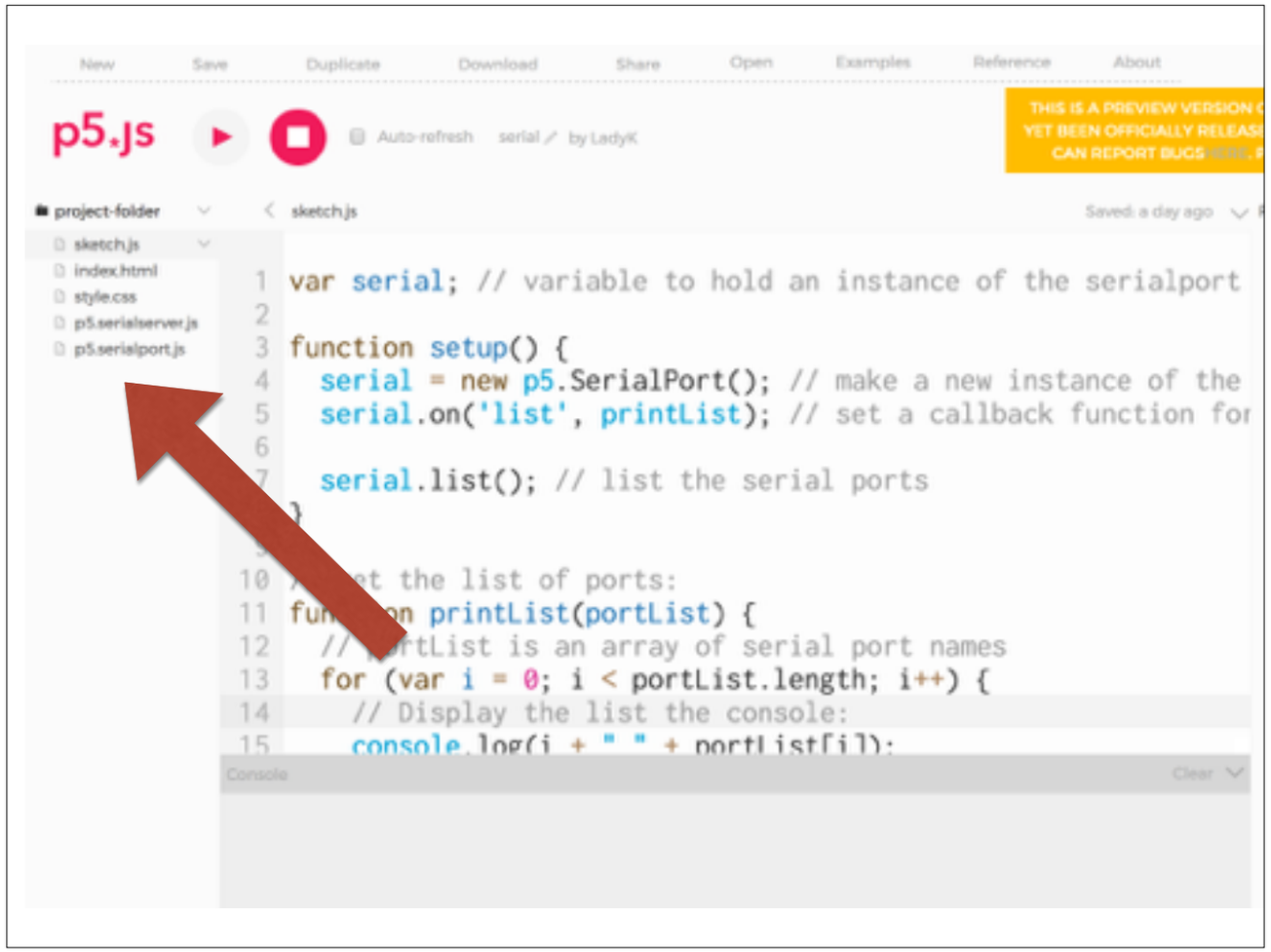
Download the p5 libraries

- https://github.com/vanevery/p5.serialport
- https://github.com/vanevery/p5.serialcontrol/releases

Need 2 things: the library in your p5 project folder + an internal server running (either with node.js installed + node startserver.js from the command line OR the p5.SerialControl (stand alone) application. We are going to use the p5.SerialControl app in conjunction with our p5 sketches

Start the p5 Serial Control App

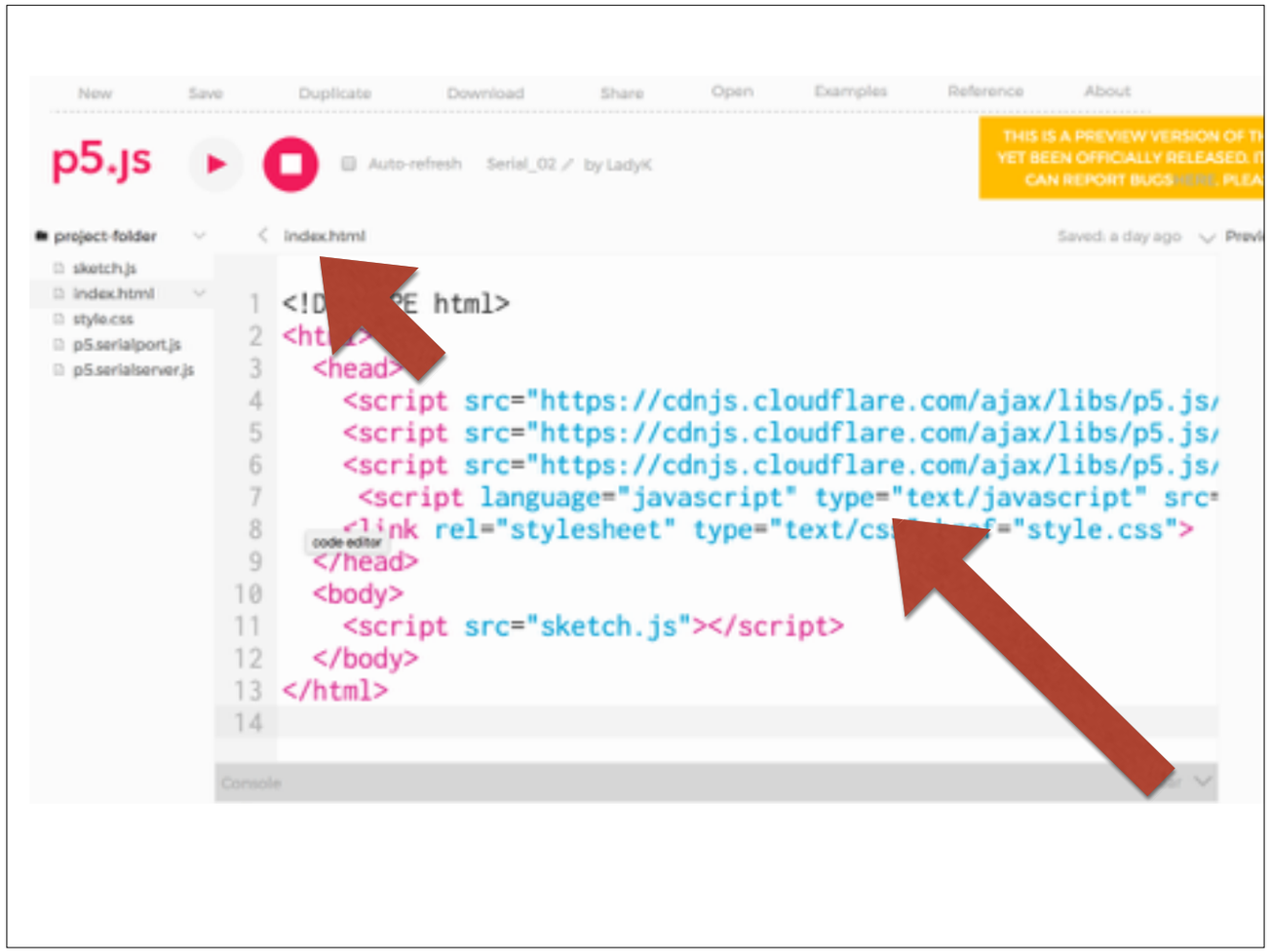

New
Save
Open
Examples
Reference
About
p5.js
Auto-refresh
Casual part
project-folder
sketch.js
index.html
style.css
sketch.js
Preview
function setup() {
  createCanvas(400, 400);
}
function draw() {
  background(220);
}
Console
Clear



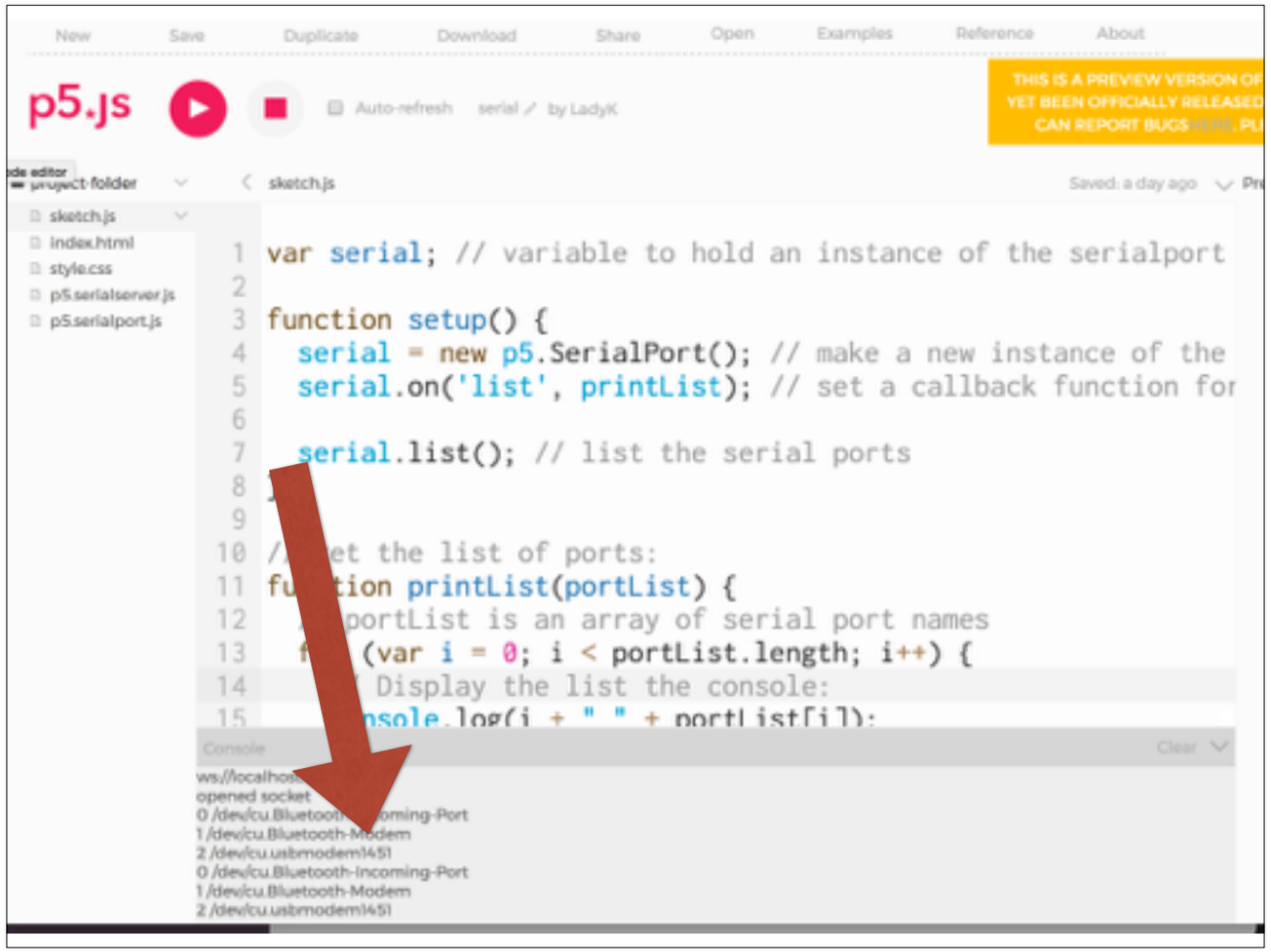
Serial example

move the p5.serialport.js file and add it to your sketch, through the sidebar

you don’t need the p5.serialserver.js file. ignore that file

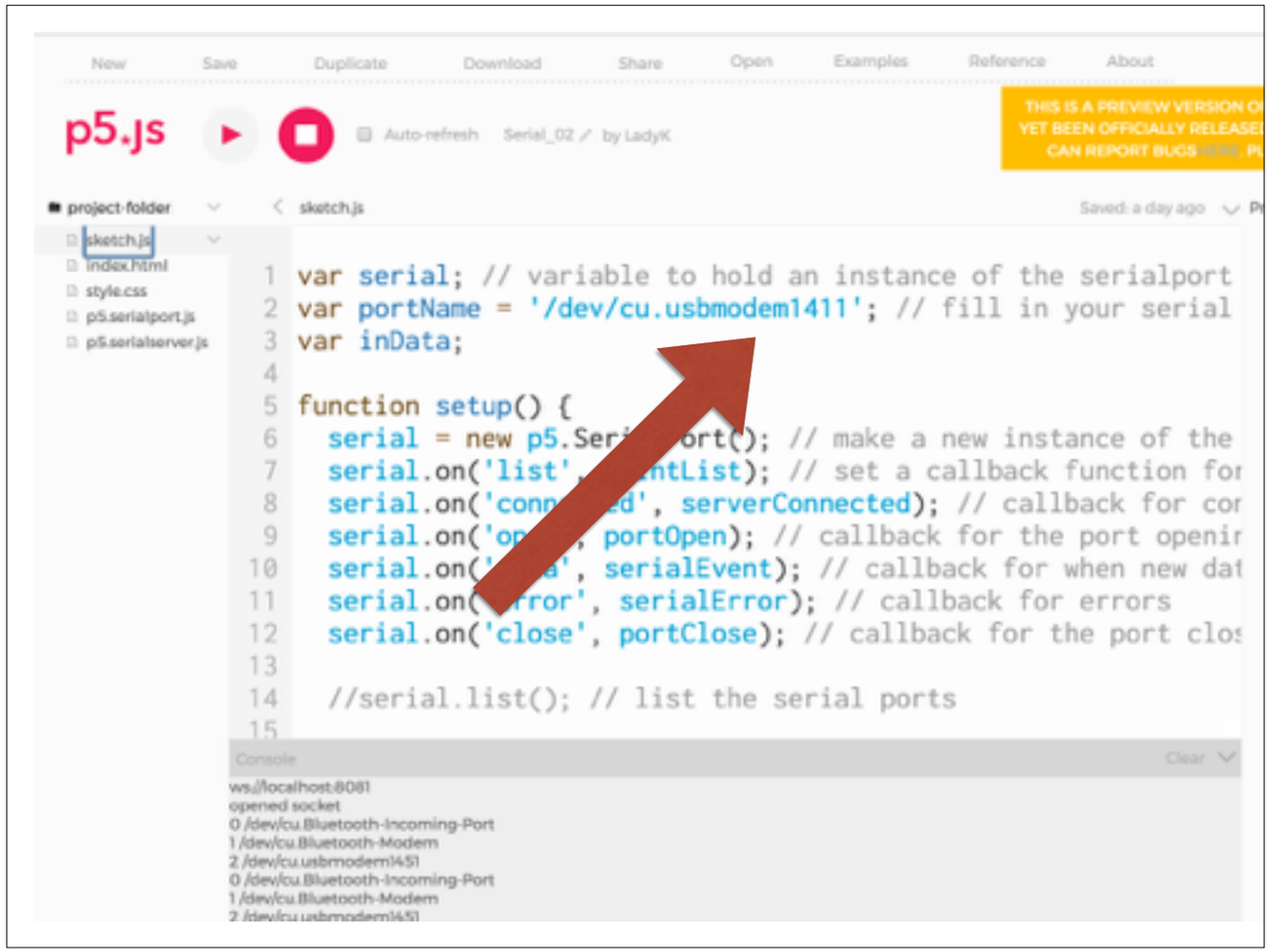


Make a reference in your index.html for the p5.serialPort.js library in your index file



Make sure you are referencing your port number that you are communicating on.

Make note of it in the console log



change your port name

Input

- Digital switch

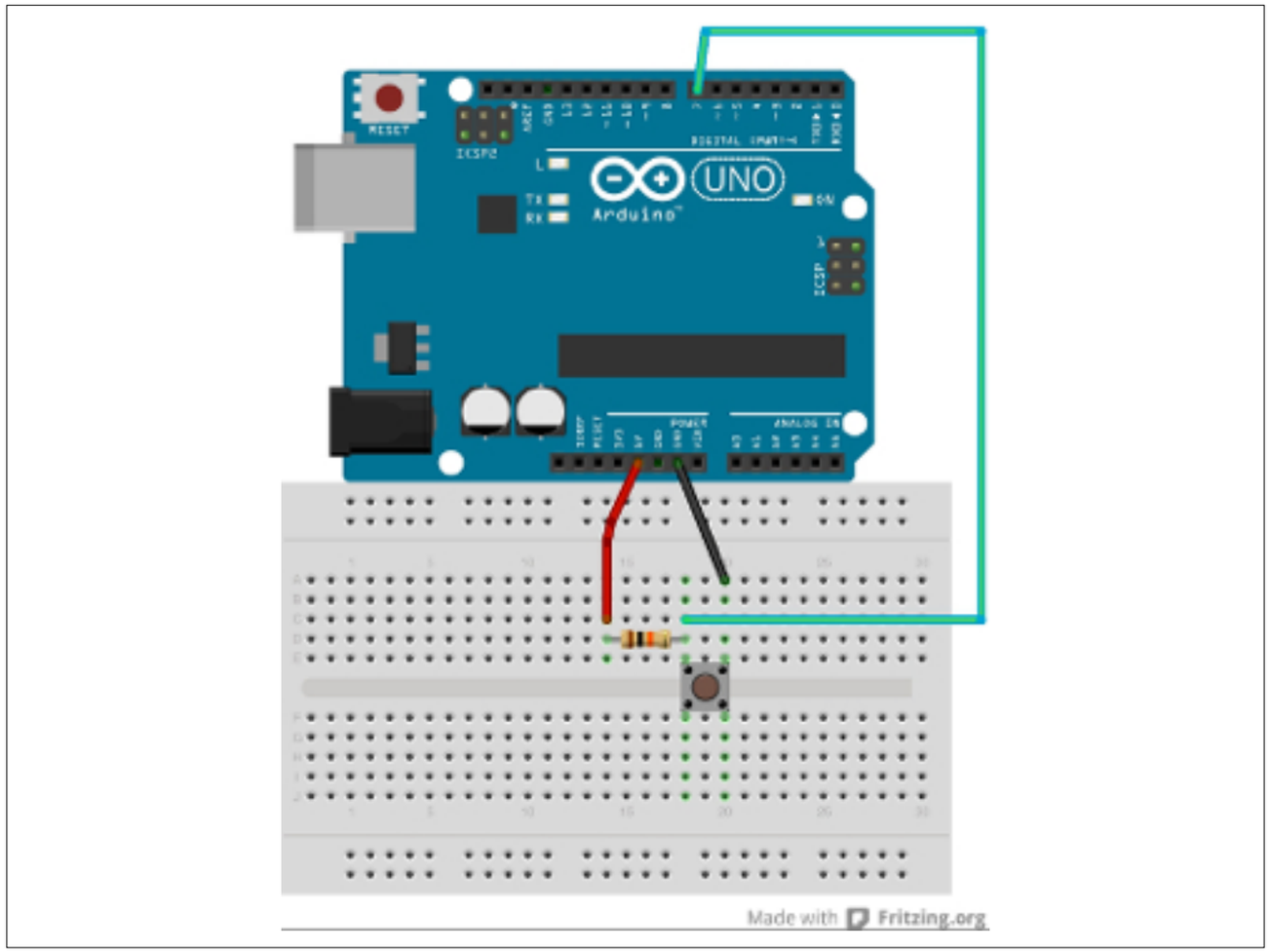
Or you could do analog in. Or output.

But serially write the data in/out to the arduino

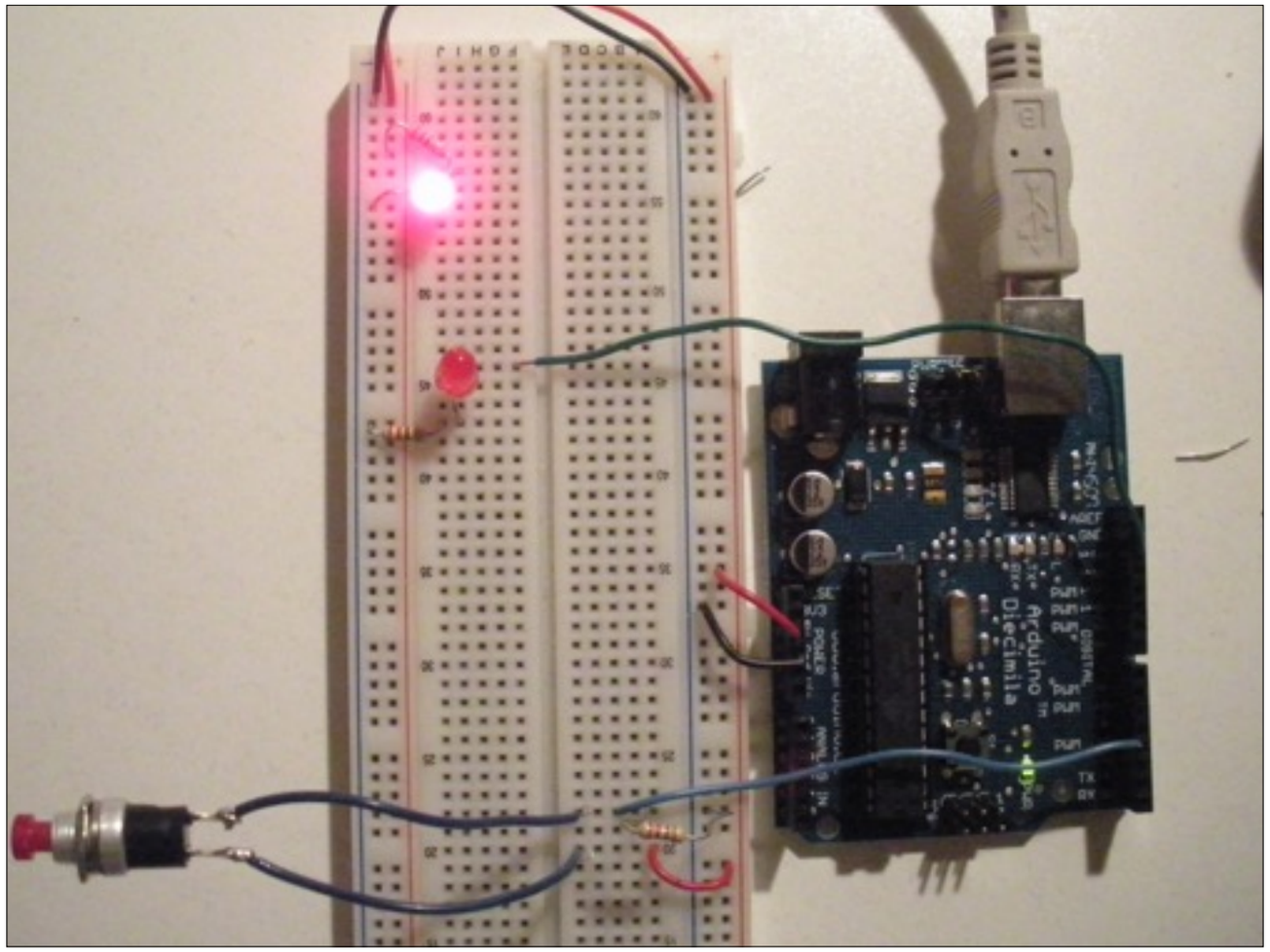
Burn the Digital Output code on the Arduino

Burn the code on the arduino and close the arduino application

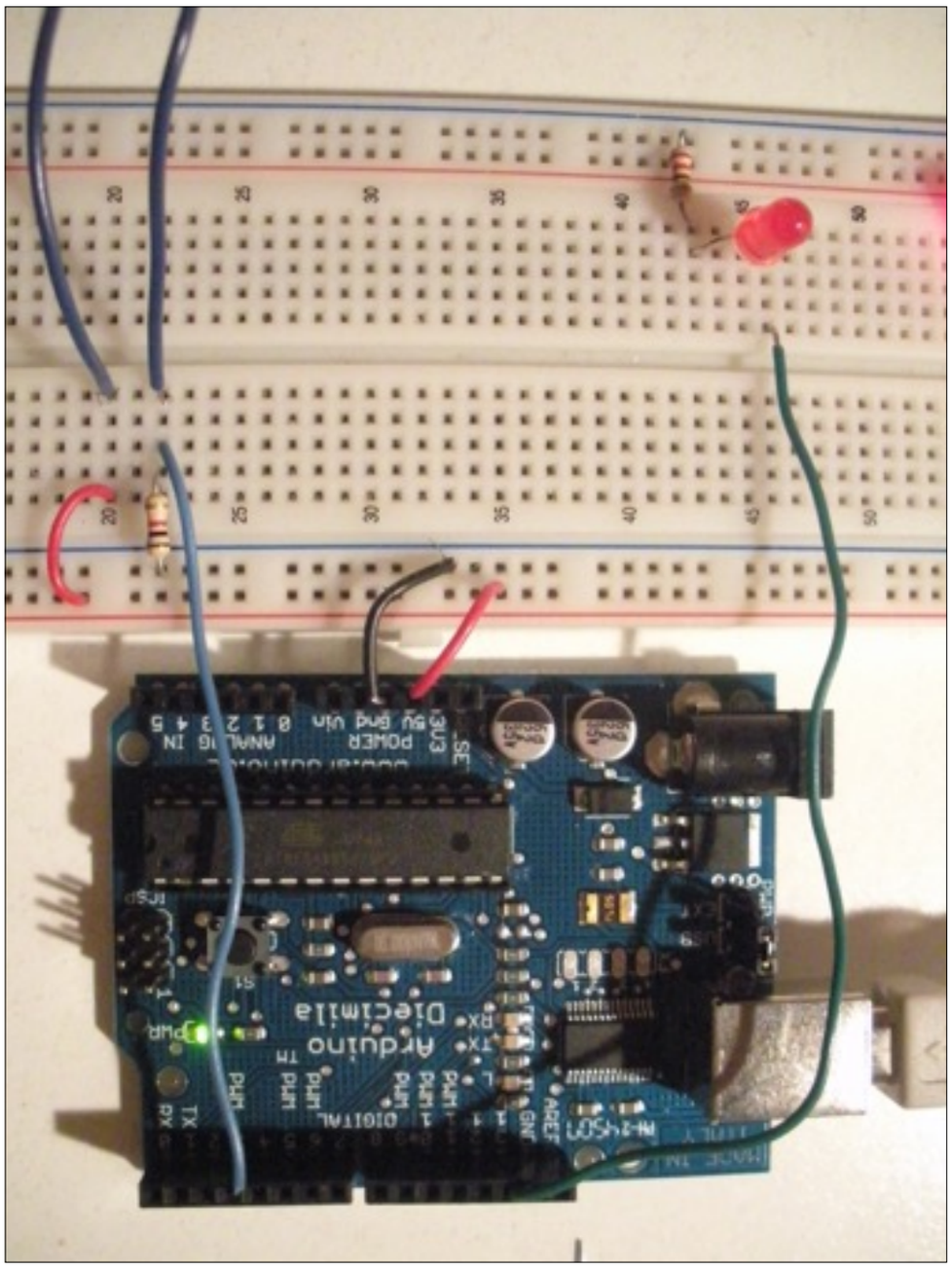
Start the p5 Serial Control App

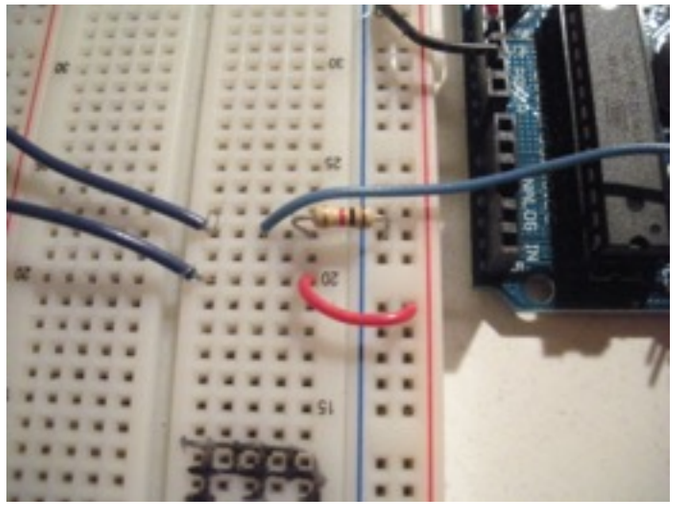


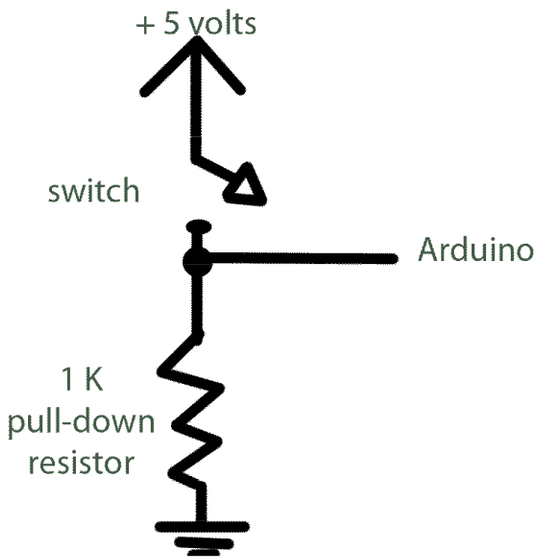
wire up your circuits accordingly



Basic set-up for the switch: switch, 1k resistor, wires.

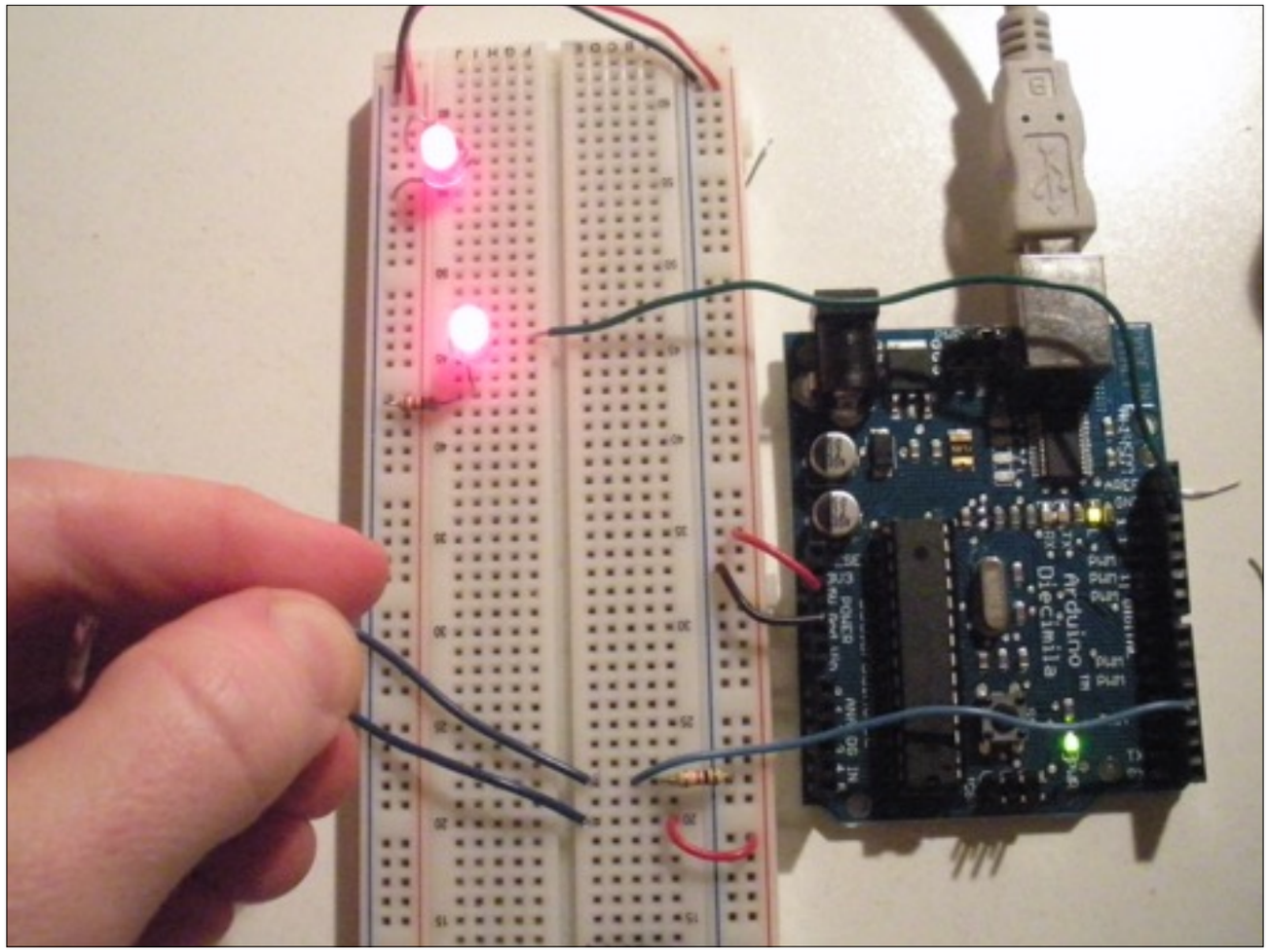

Arduino
Diecimila
DIGITAL
ANALOG IN
POWER
3V3 5V Gnd Vin 0 1 2 3 4 5
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX 1
RX 0
PWR



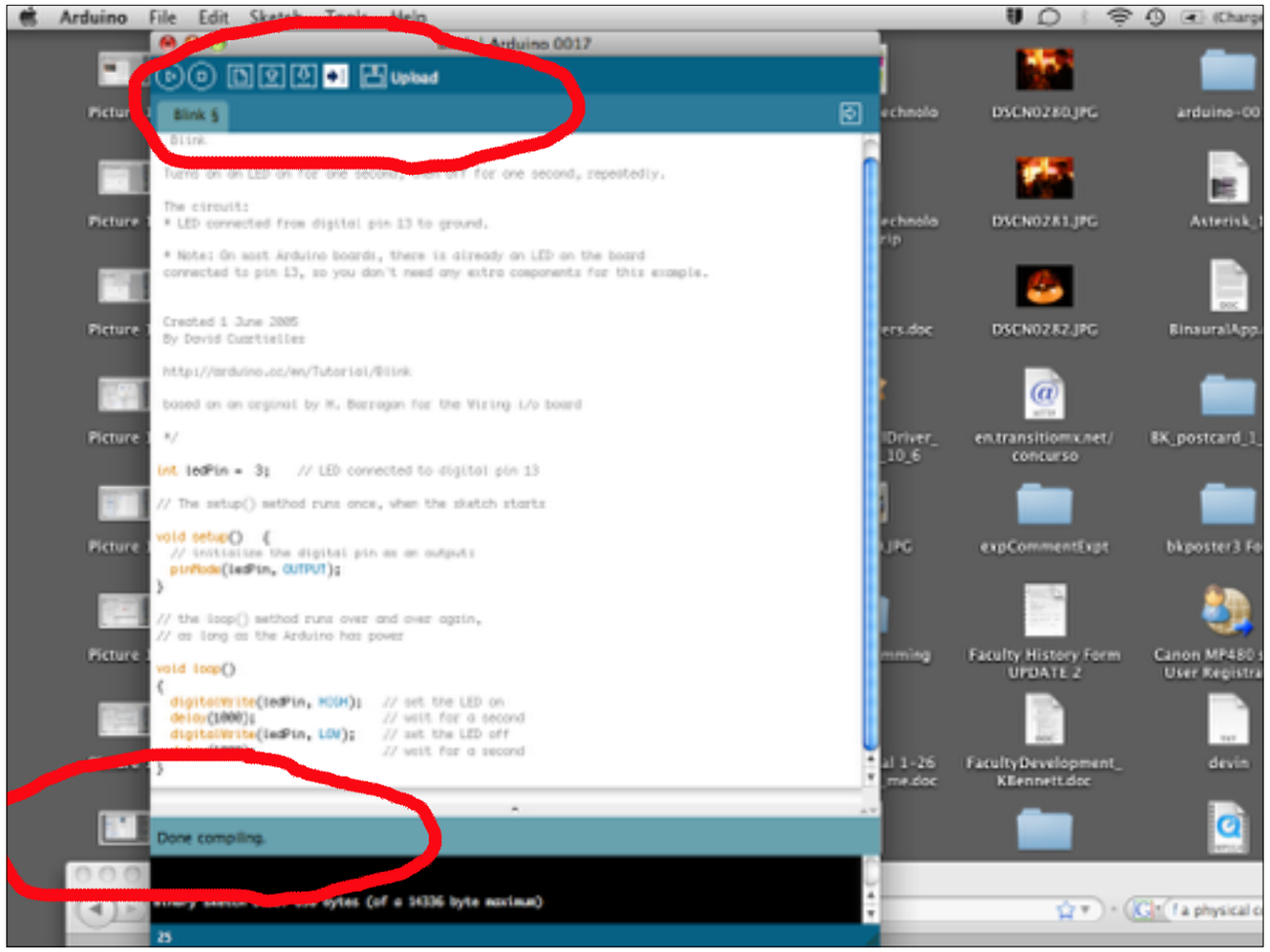


A Switch
Switches ALWAYS need a pull-down resistor. One side of this resistor goes between ground. The other side goes to the “node” where one end of the switch also connects to the wire going into our Arduino Board.
The other end of the switch is connected to the positive 5 volts.

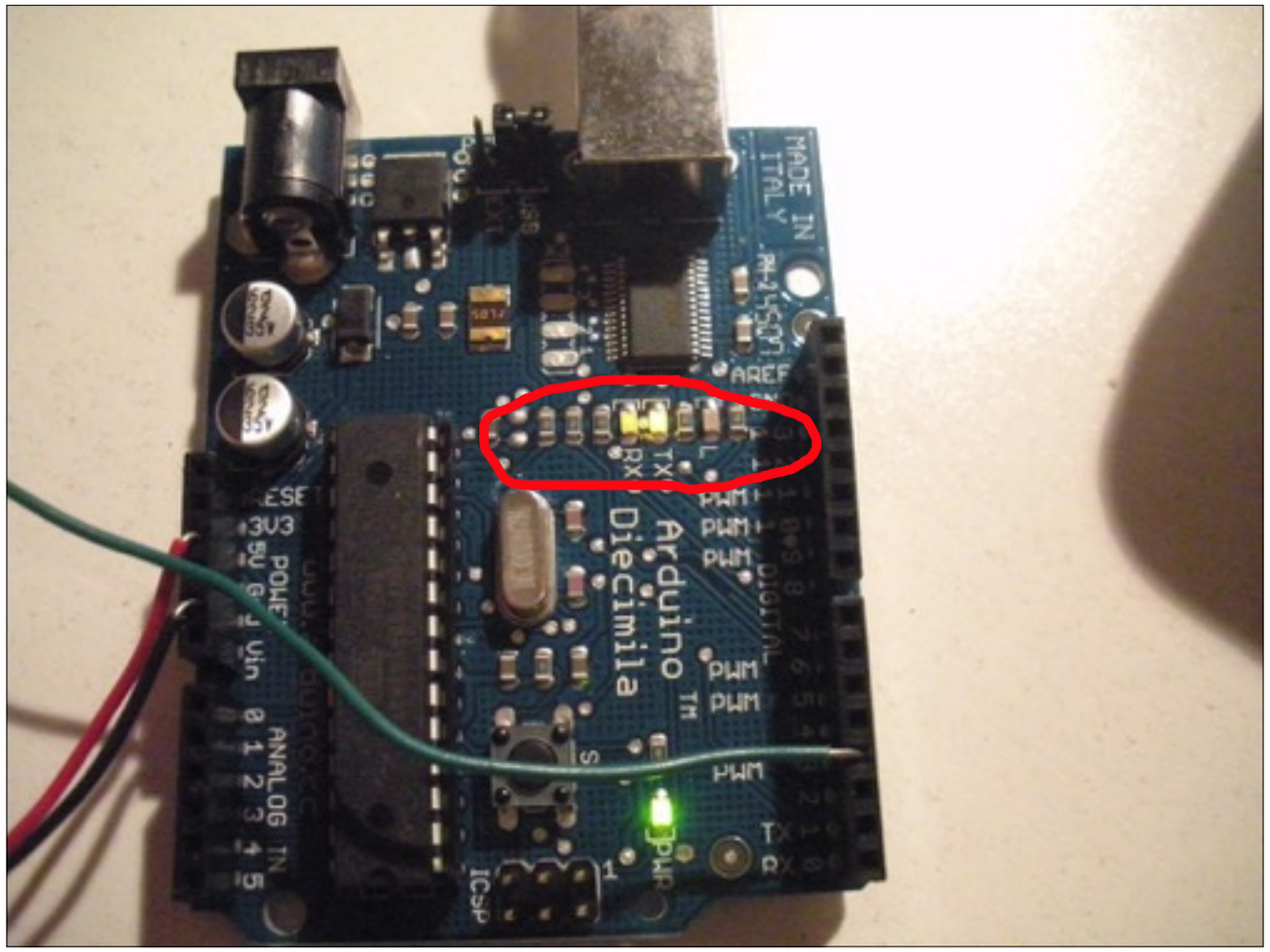
The resistor here prevents the Arduino from reading low voltages in the atmosphere.
Electrical current always flows from the path of least resistance to ground. If you just had a wire connecting the switch and the pin to ground, then, when you close the switch, the path of least resistance would be through the wire, and you’d have a short circuit. The easiest path for the current is through the closed switch. When the switch is open, the resistor offers the only path and the current goes through it.



Make sure you have the correct ports and chip selected in the Arduino Program.
Upload your button code. Make sure the code reflects the correct pins you've wired your button (input- listening) and led(output - talking) to.



Let's upload the code to the chip by pressing the upload button
at the top of your Arduino sketch window, on your computer.
This is now burning the code onto the chip.
Notice the flashing TX and RX lights on the Arduino board while this is happening.



Notice the flashing TX and RX lights on the Arduino board while this is happening.