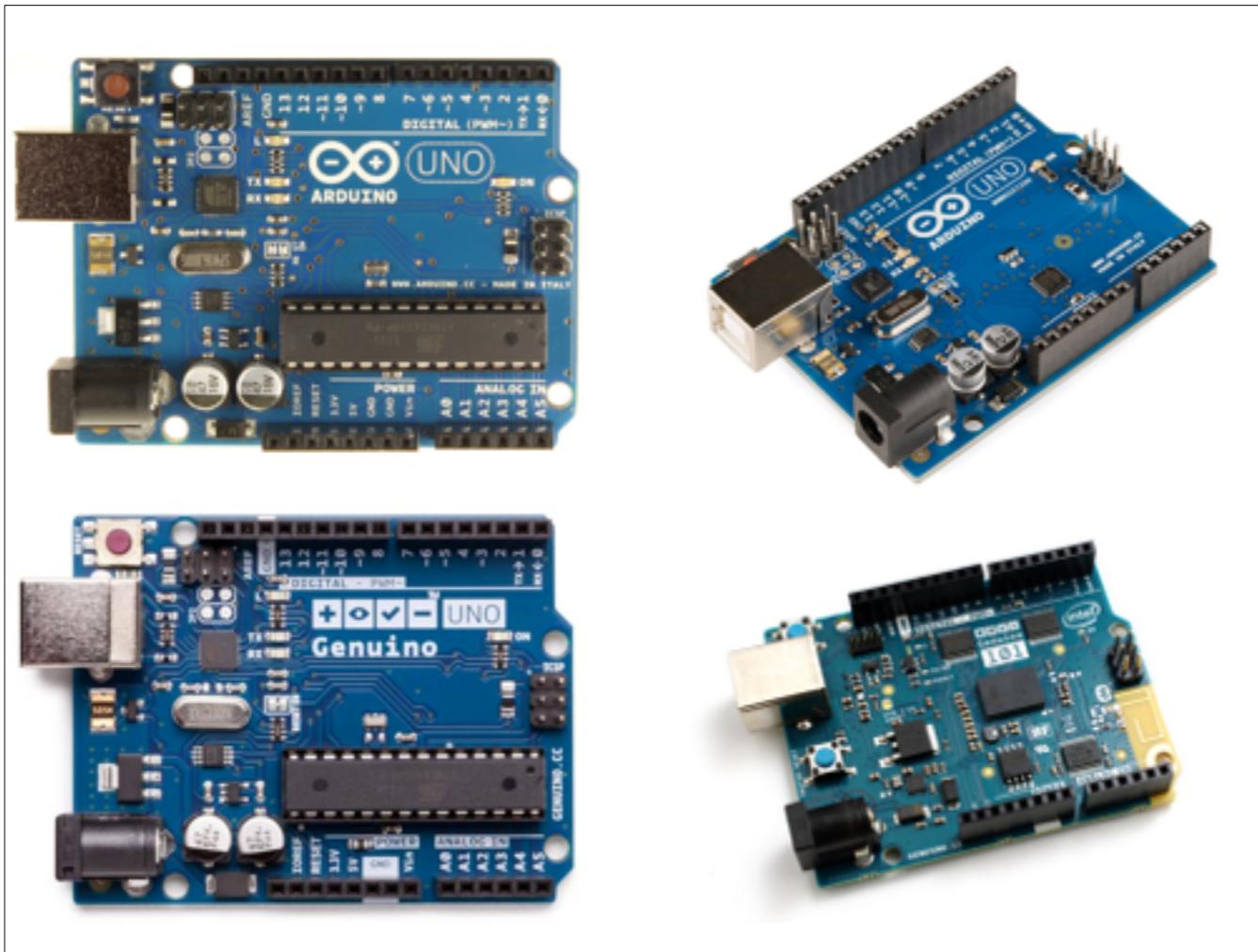




EVOLUTION OF ARDUINO

In 2005, a group at Italy's Interaction Design Institute Ircam developed Arduino as a low-cost, easy-to-use electronics platform for students and artists. It borrows its name from nearby waterfalls. Bar di Rio Ardeatino. Since exploding onto the maker scene, Arduino has cultivated a flourishing community of inventors, engineers, and hackers dedicated to sharing code and developing hardware under an open source banner.



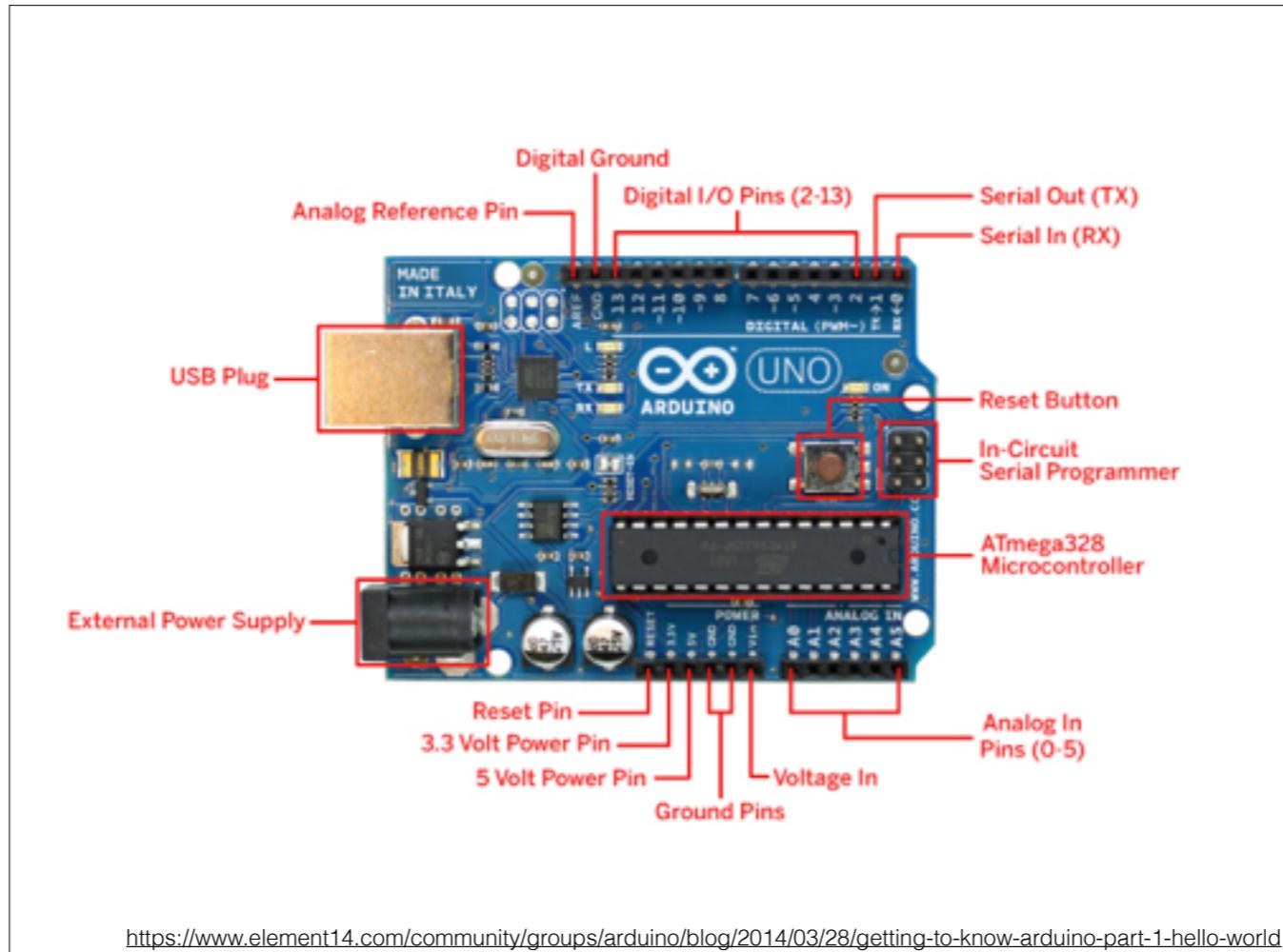


Arduino & Genuino

Arduino open-source, label Arduino-compatible vs Arduino



Meet the arduino



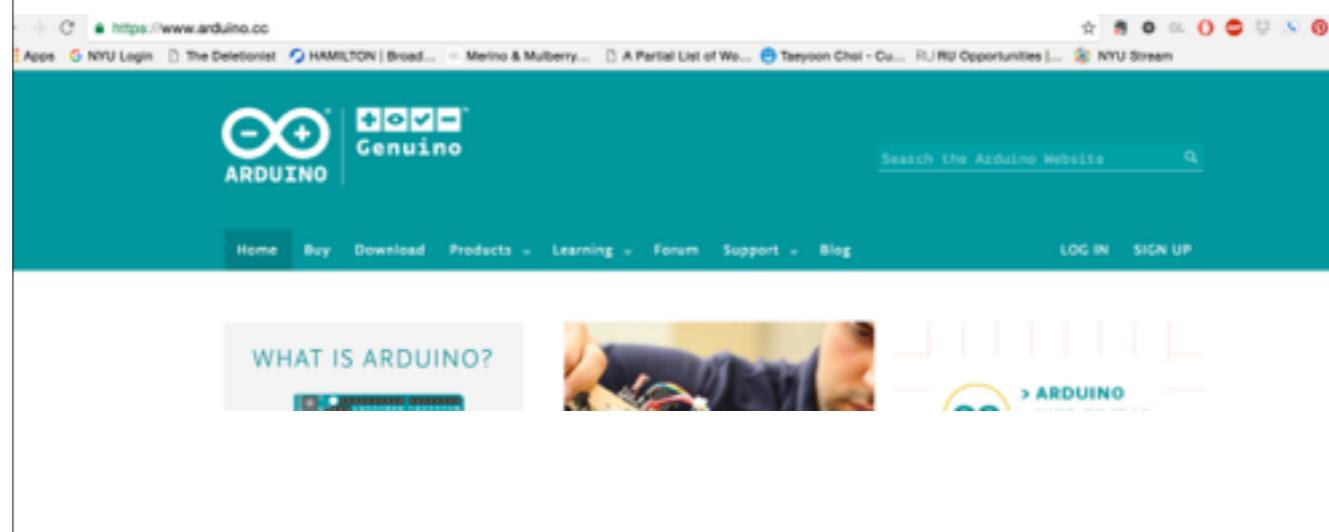
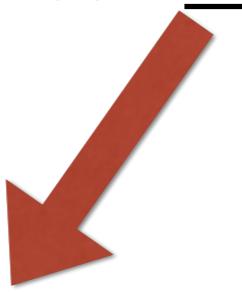
the size of a credit card, it has 14 digital IO pins and 5 analog, and runs off 5v. It can be powered either by the USB cable or up to 12v DC via a barrel jack.

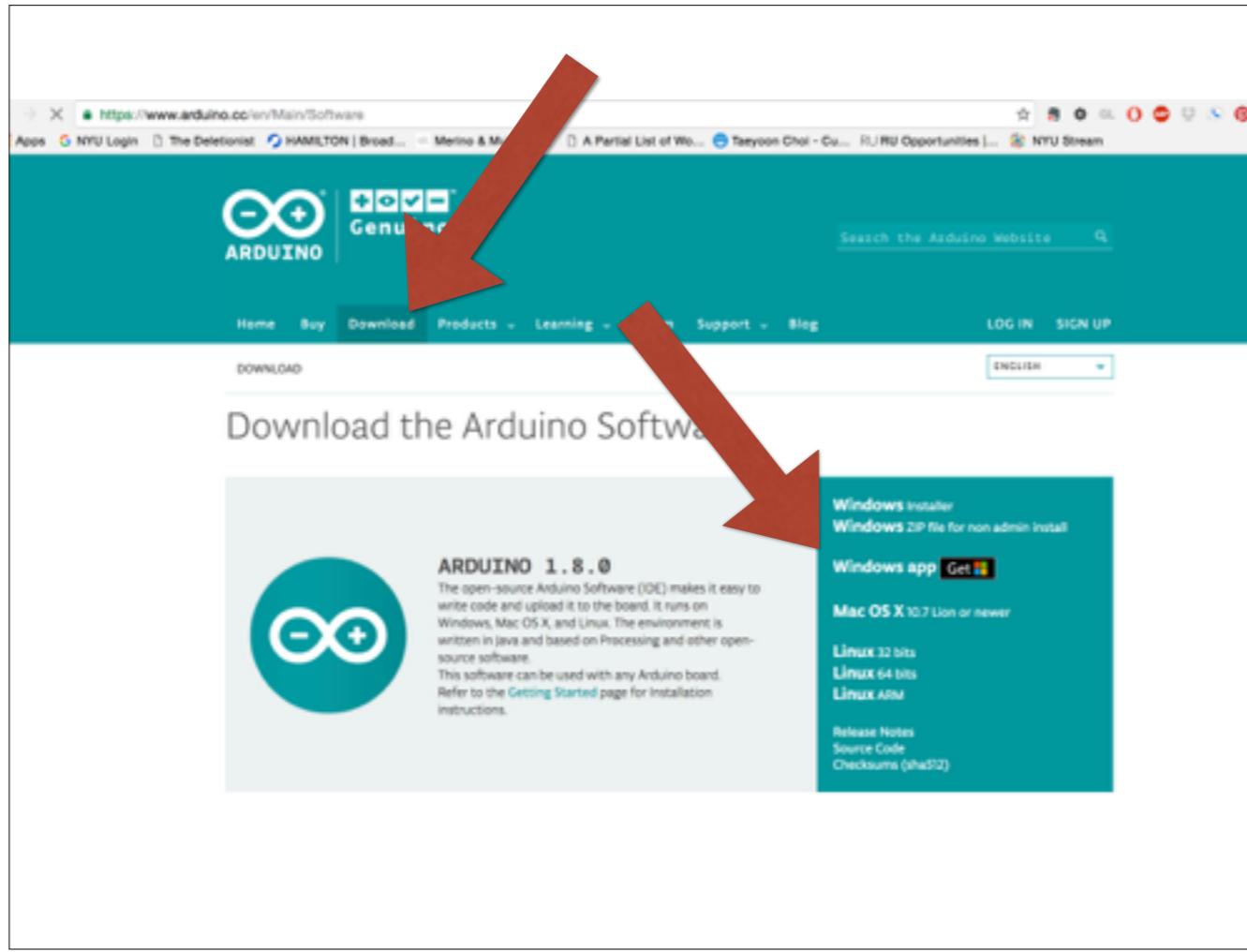
ATMEGA328P, with 32Kb of program memory, which is small by today's standards.

At some point you'll probably hit that limit, but as a starter device to learn from, the Uno is perfect.

<https://www.element14.com/community/groups/arduino/blog/2014/03/28/getting-to-know-arduino-part-1-hello-world>

Go to: www.arduino.cc





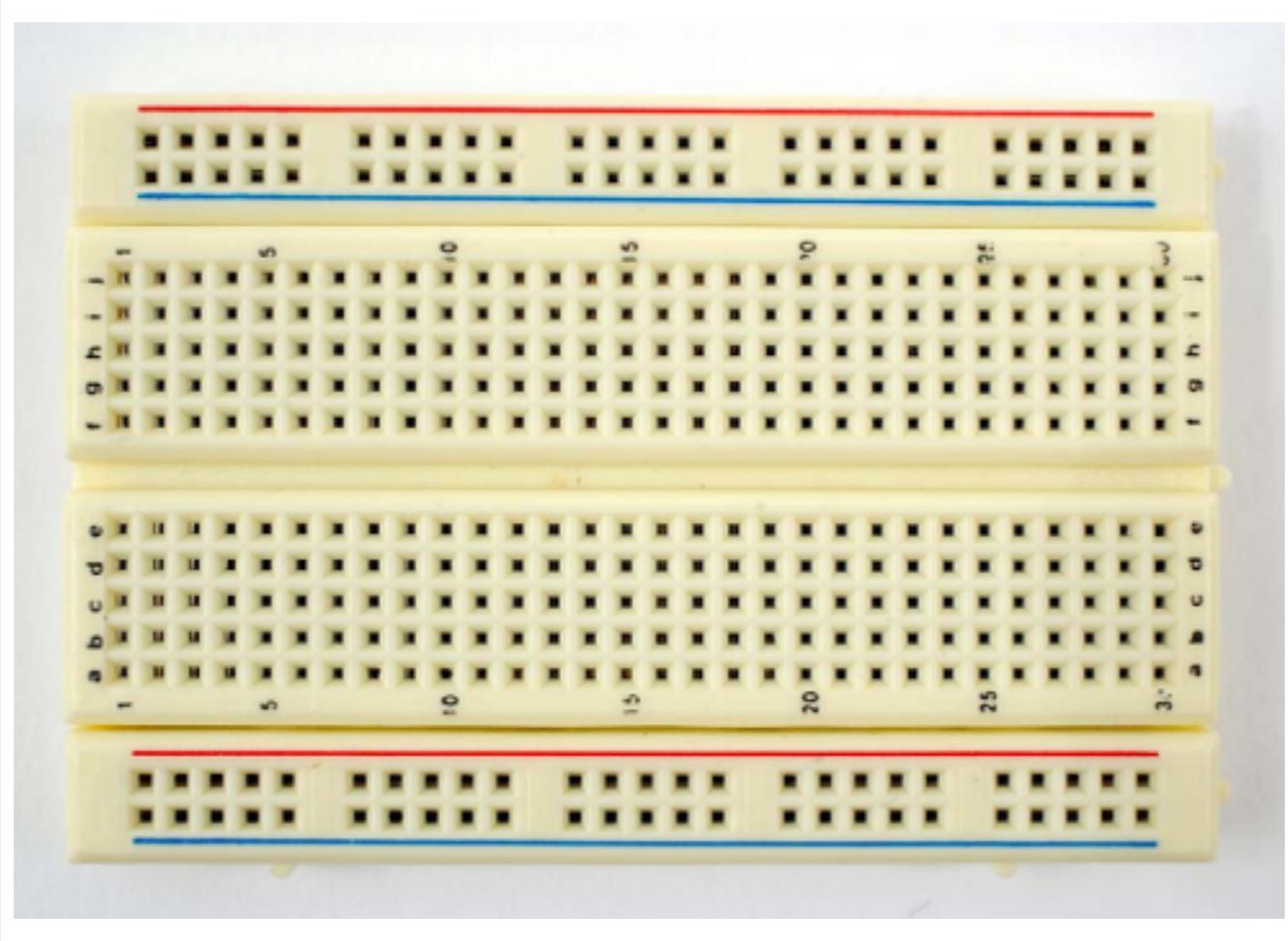
Install the software and open it

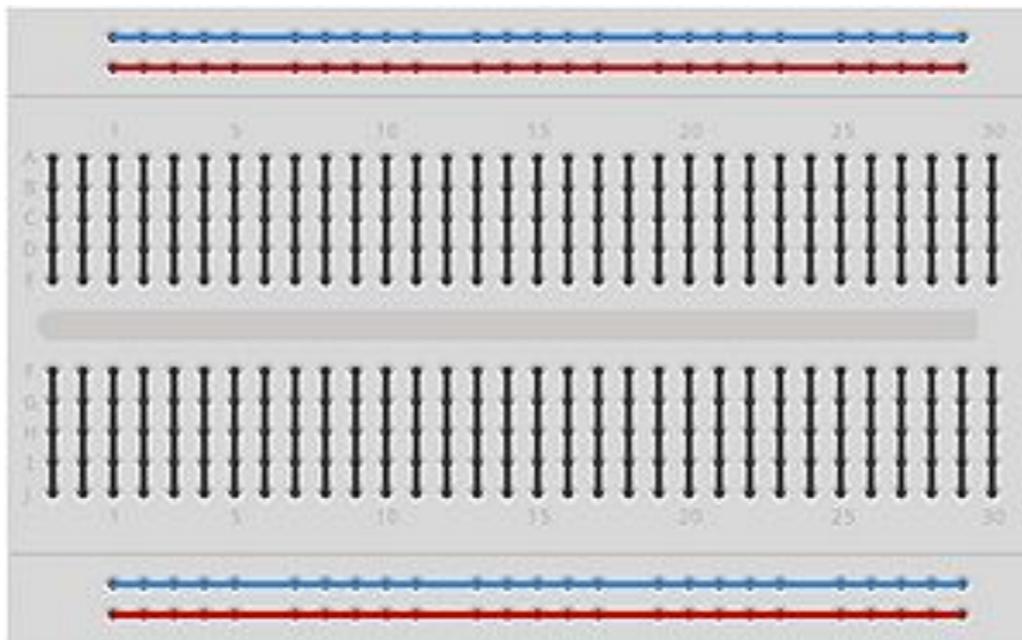
Download the p5 libraries

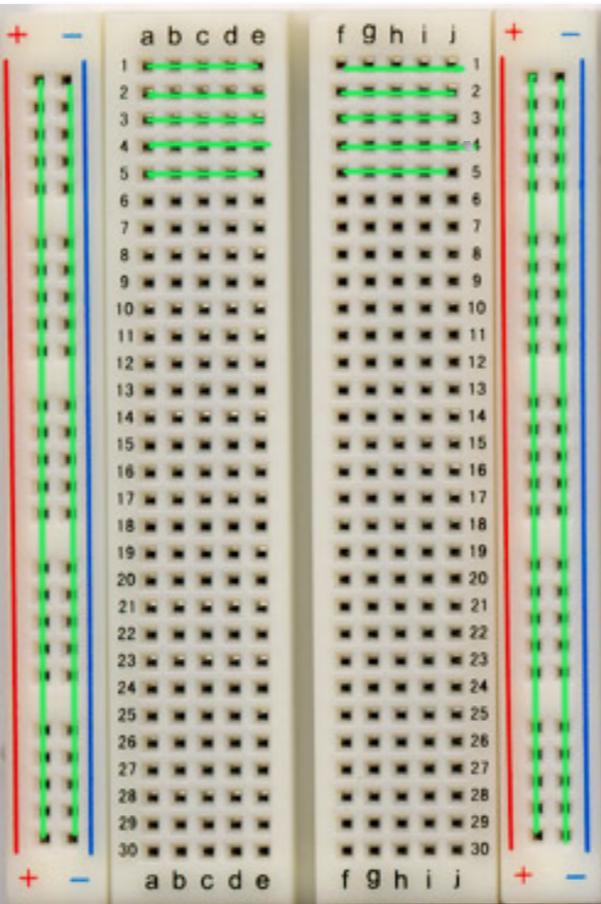
- <https://github.com/vanevery/p5.serialport>
- <https://github.com/vanevery/p5.serialcontrol/releases>



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



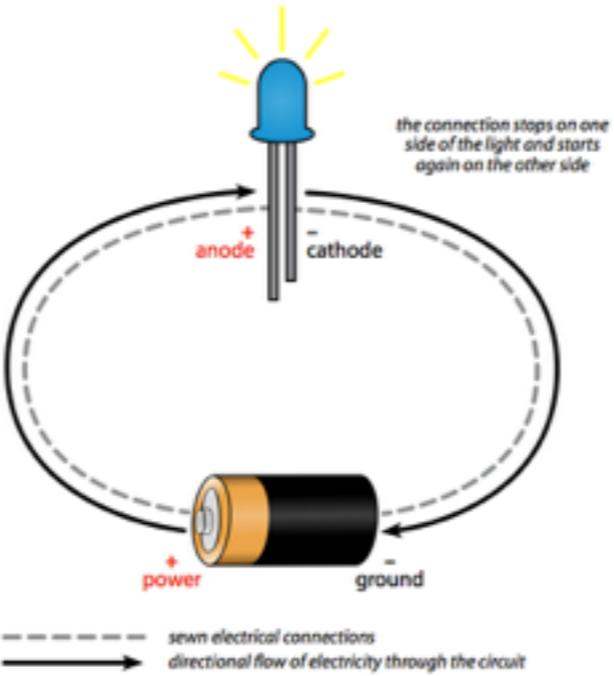


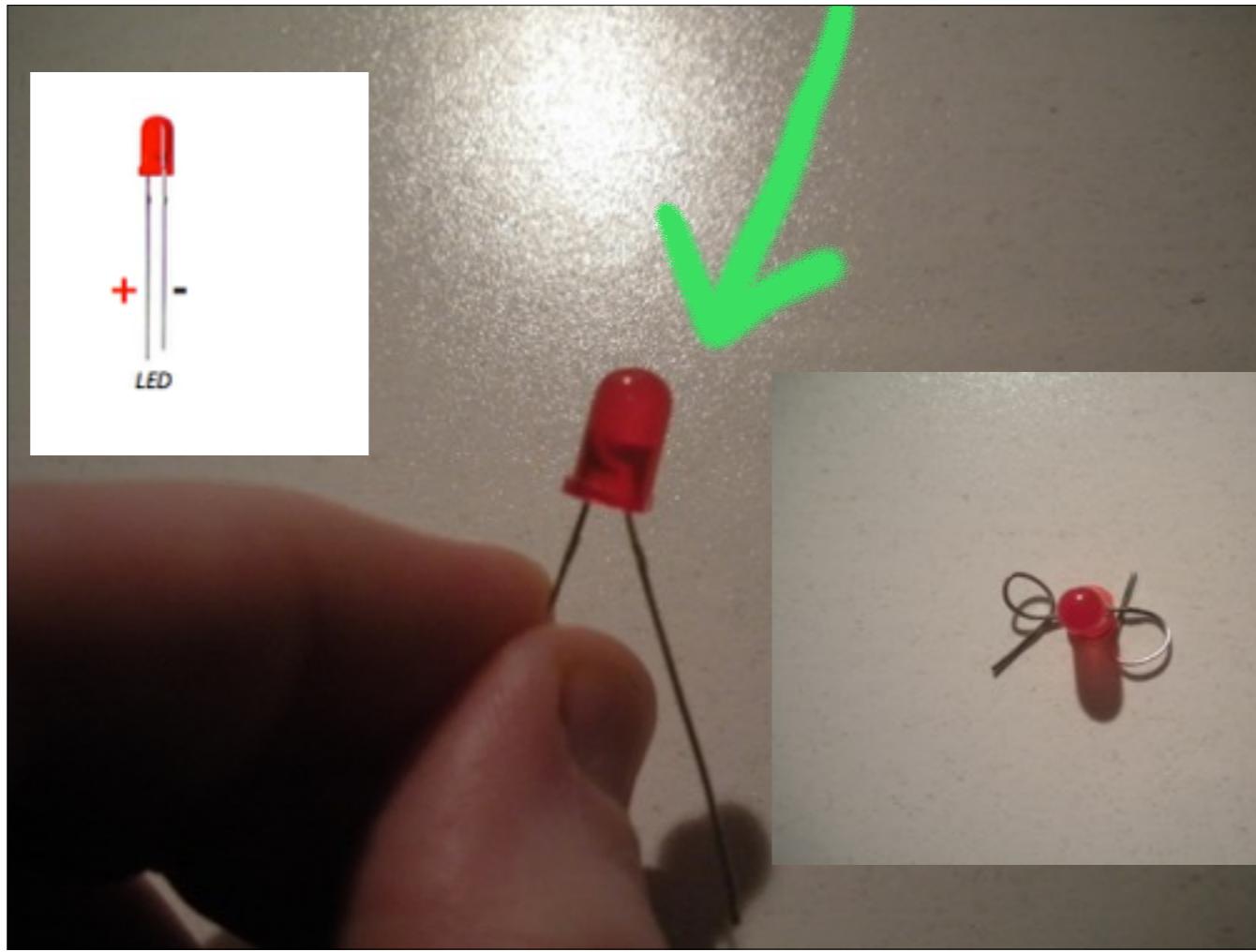


Video and Website © 2004 ClarkZanner.net

simple circuits

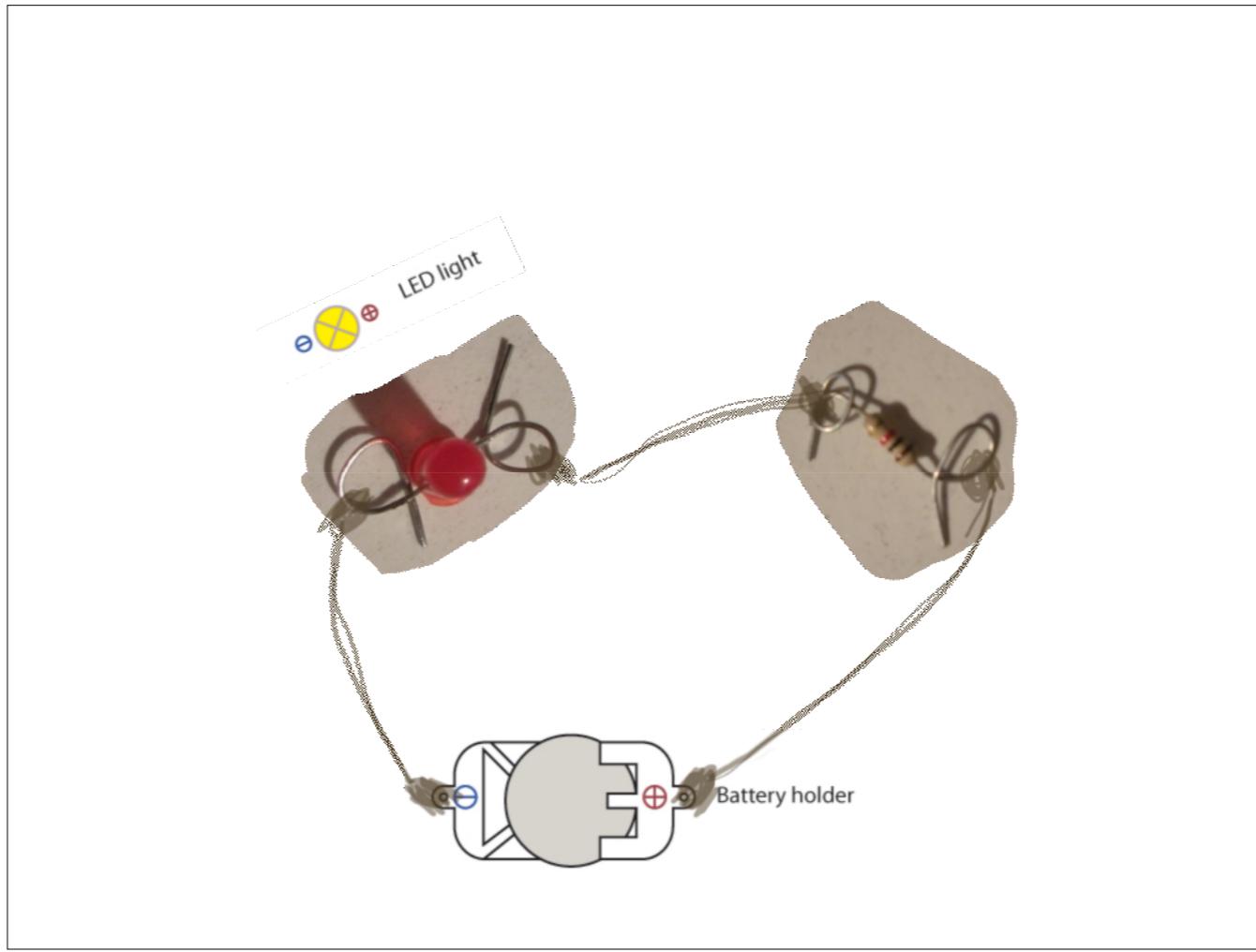
A Simple Circuit Schematic





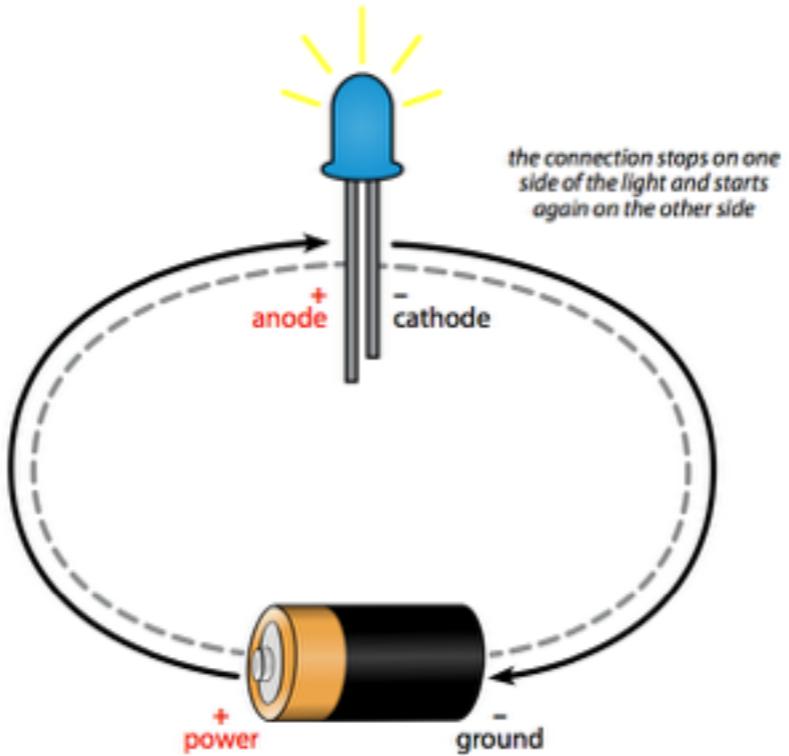
LED's are polarized.

This means that current can only flow through in one direction. As a result, we need to pay attention to which end is which, when we are connecting them.



Notice that the positive side of the LED, is facing the direction of the positive side of the battery. The negative side of the LED is facing the direction of the negative side of the battery.

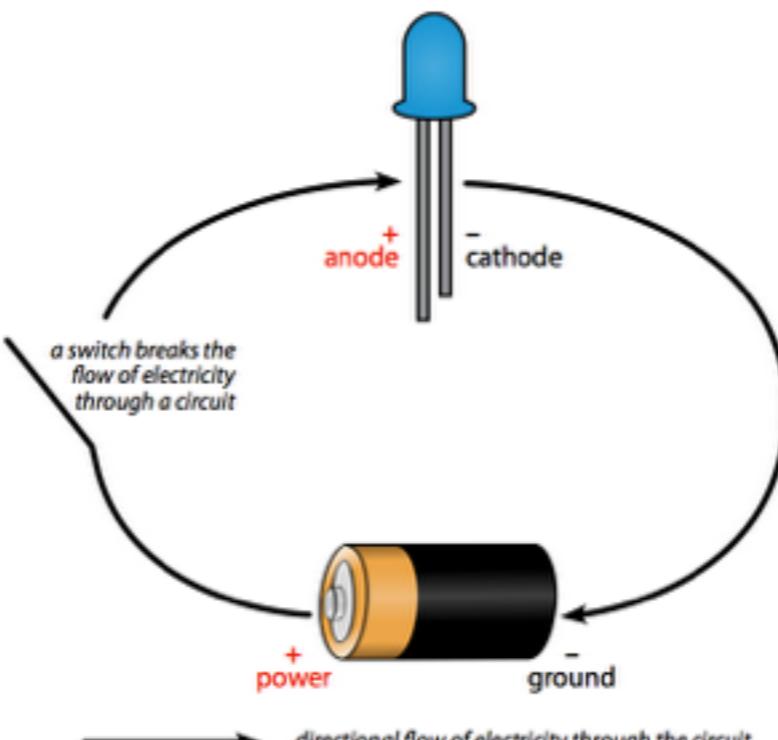
A Simple Circuit Schematic



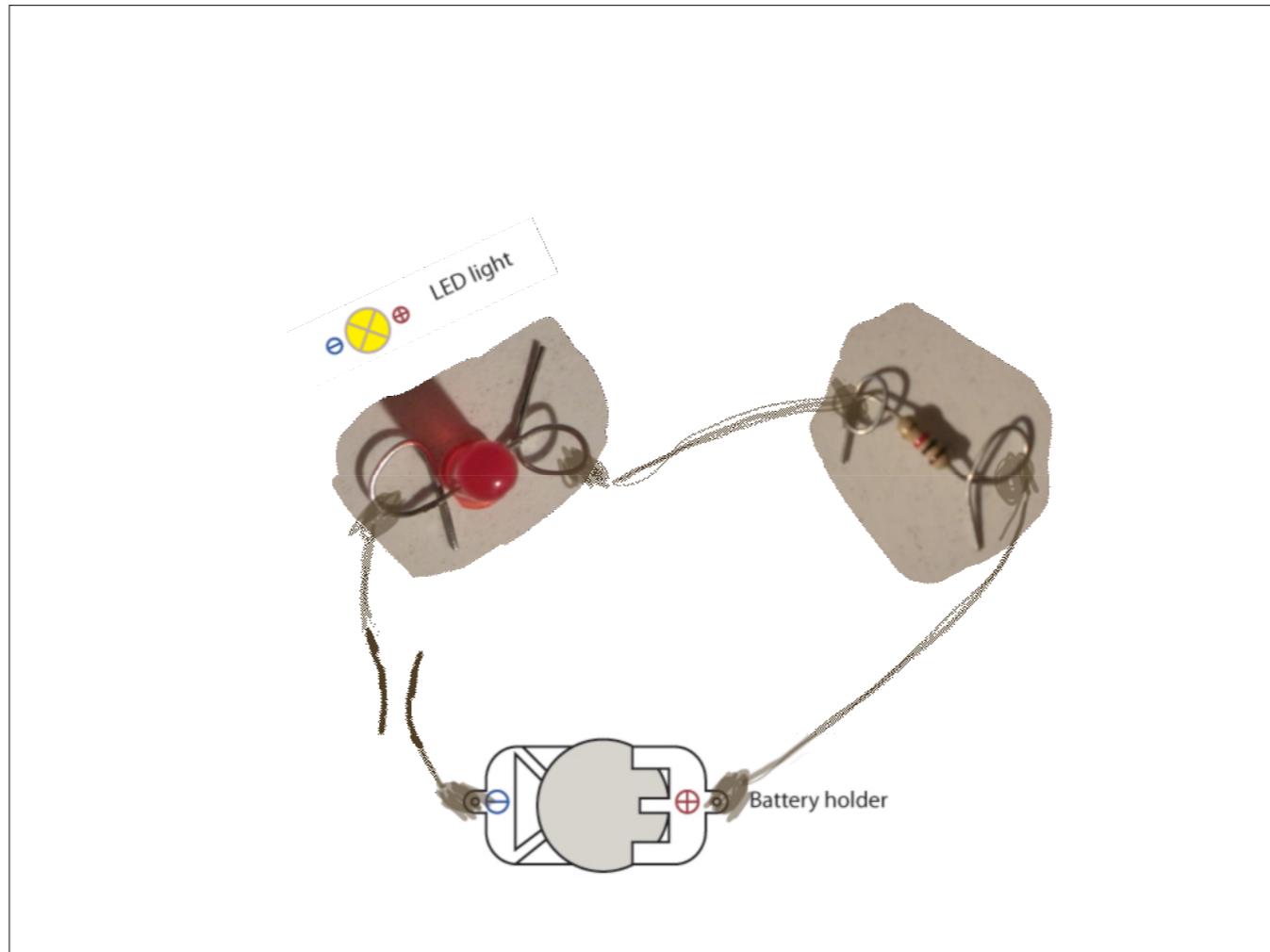
— — — — — *sewn electrical connections*

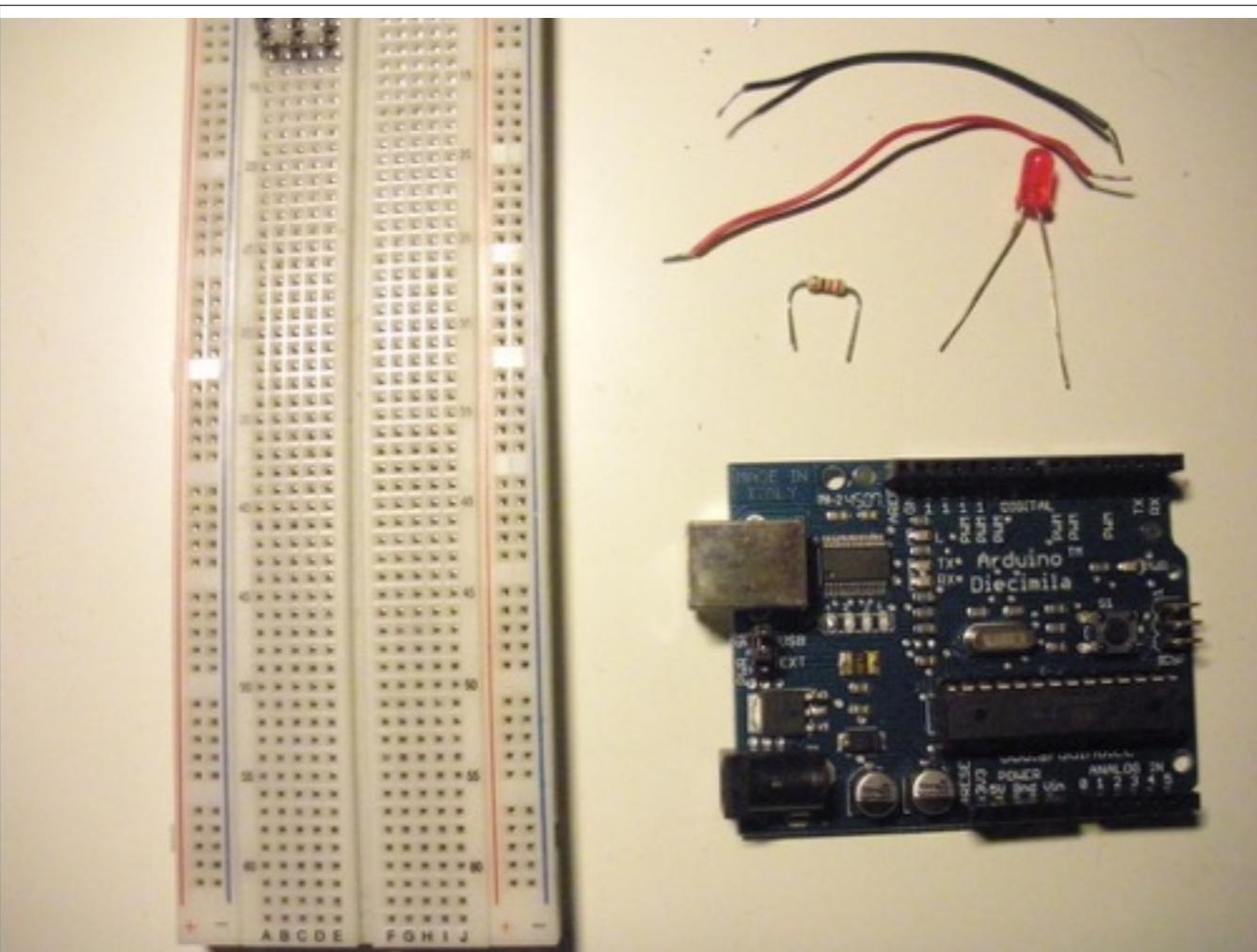
Graphic by Emily Lovell

Simple Circuit Schematic for a Switch

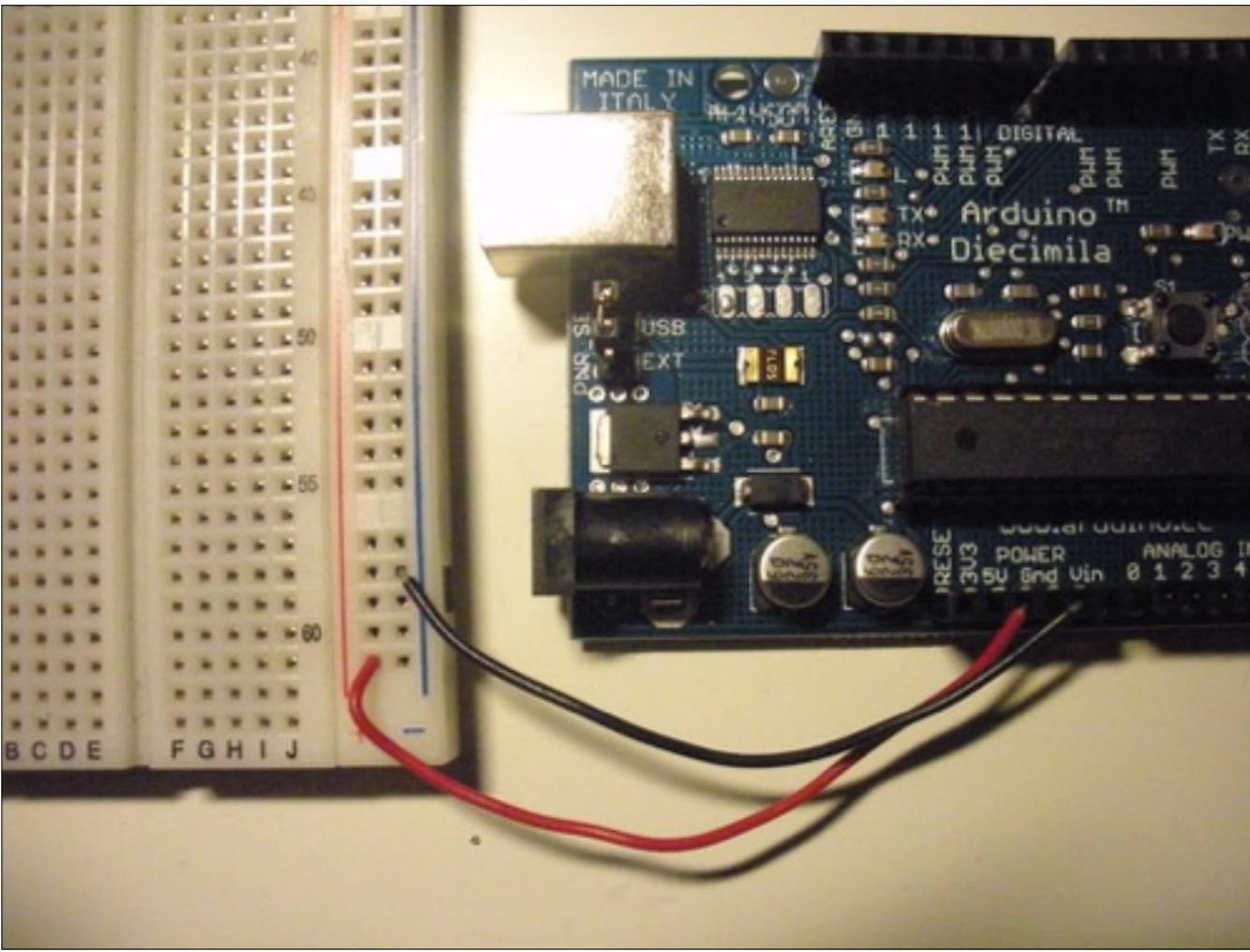


Graphic by Emily Lovell





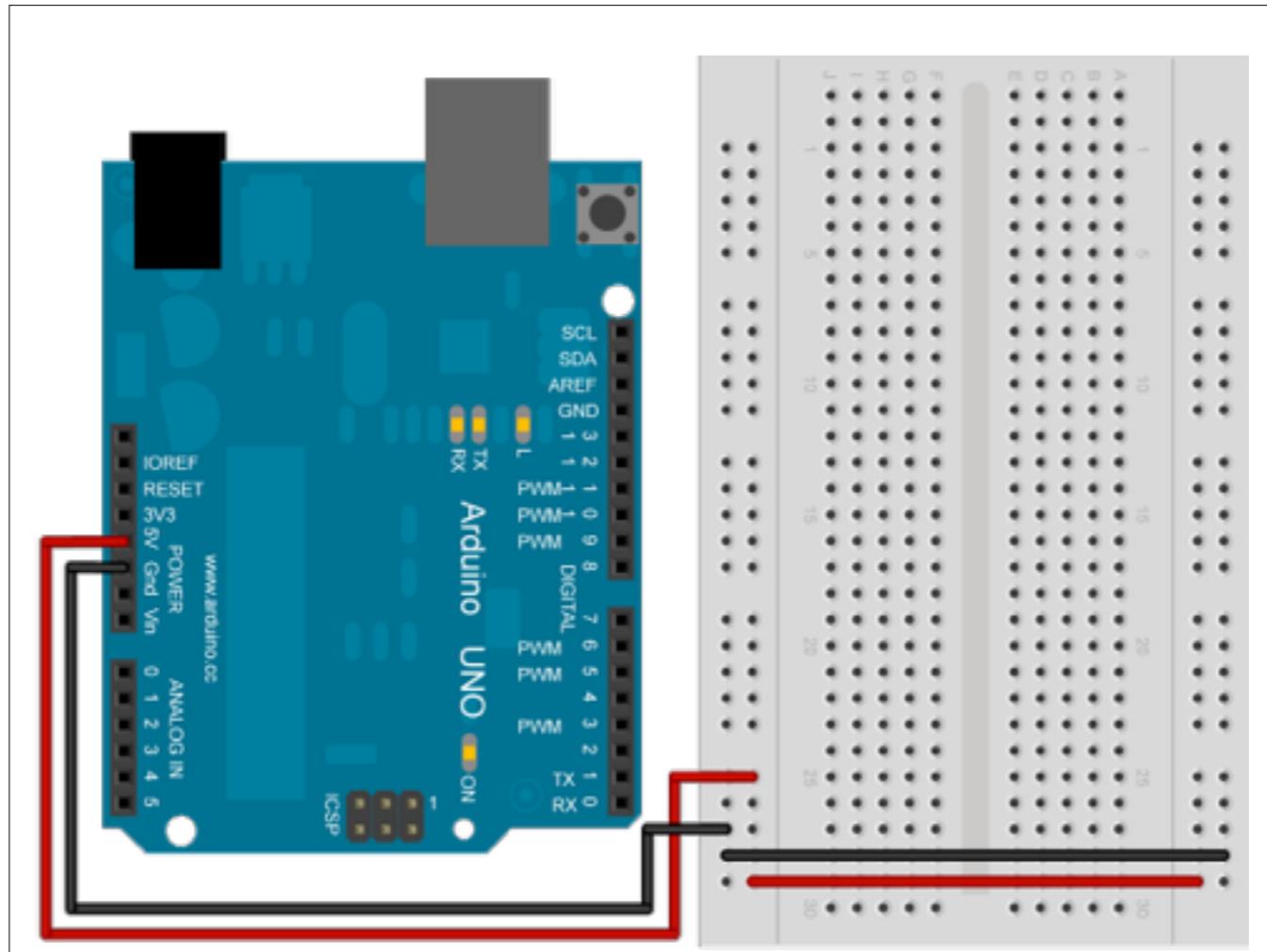
supplies

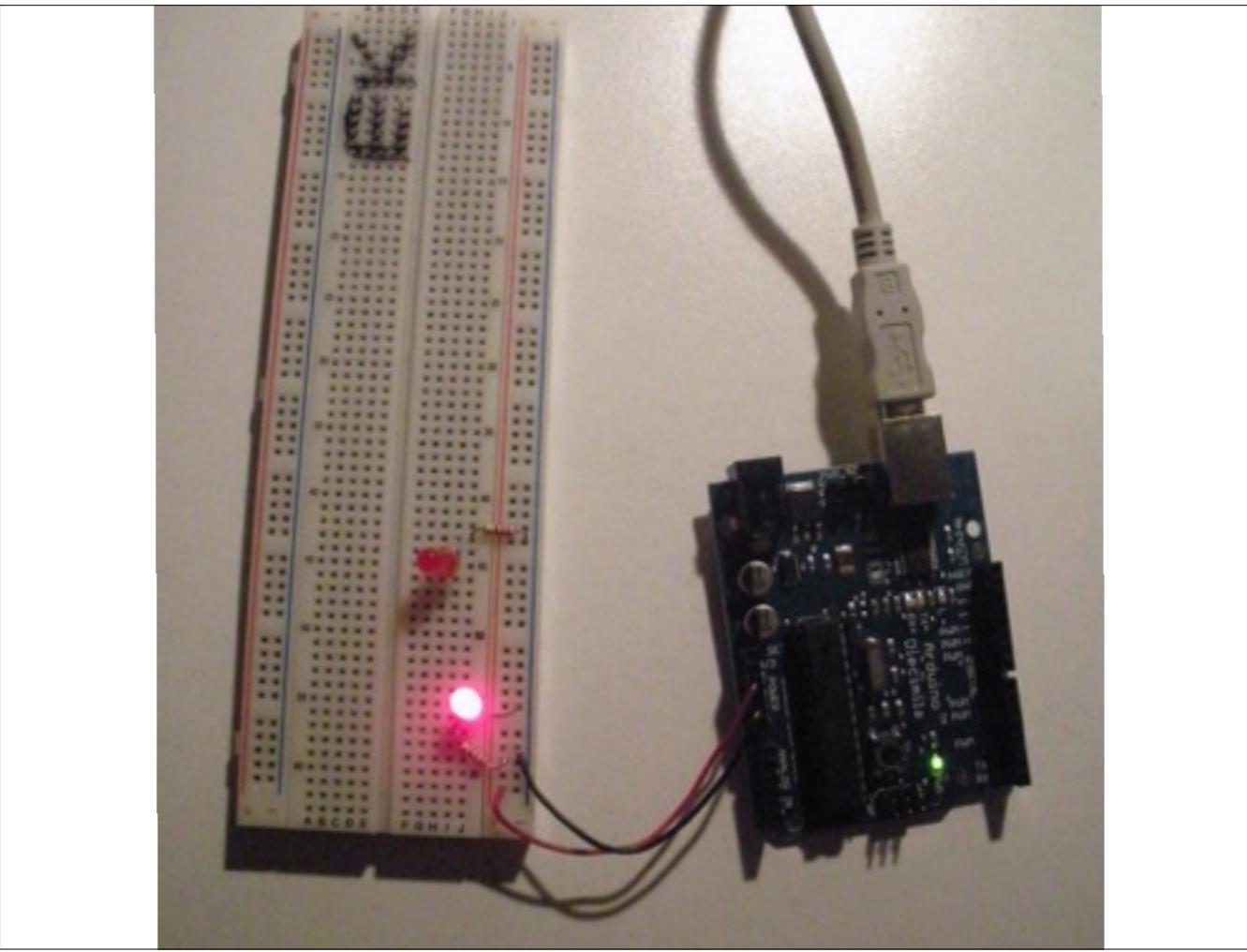


Jumper those lines over to your breadboard.

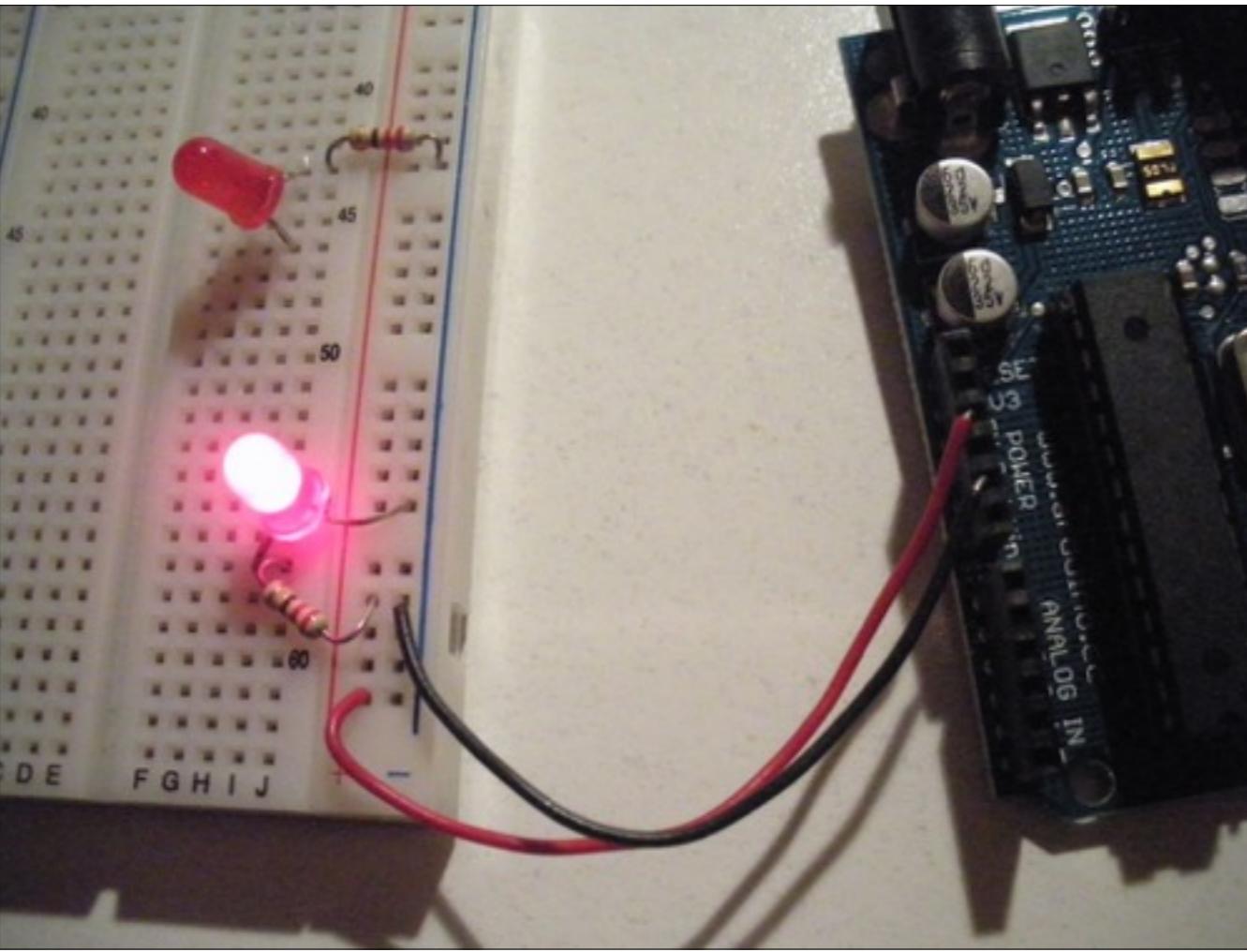
Place the black ground wire into the blue (-) side of your breadboard.

Place the red 5V wire into the red(+) side of your breadboard

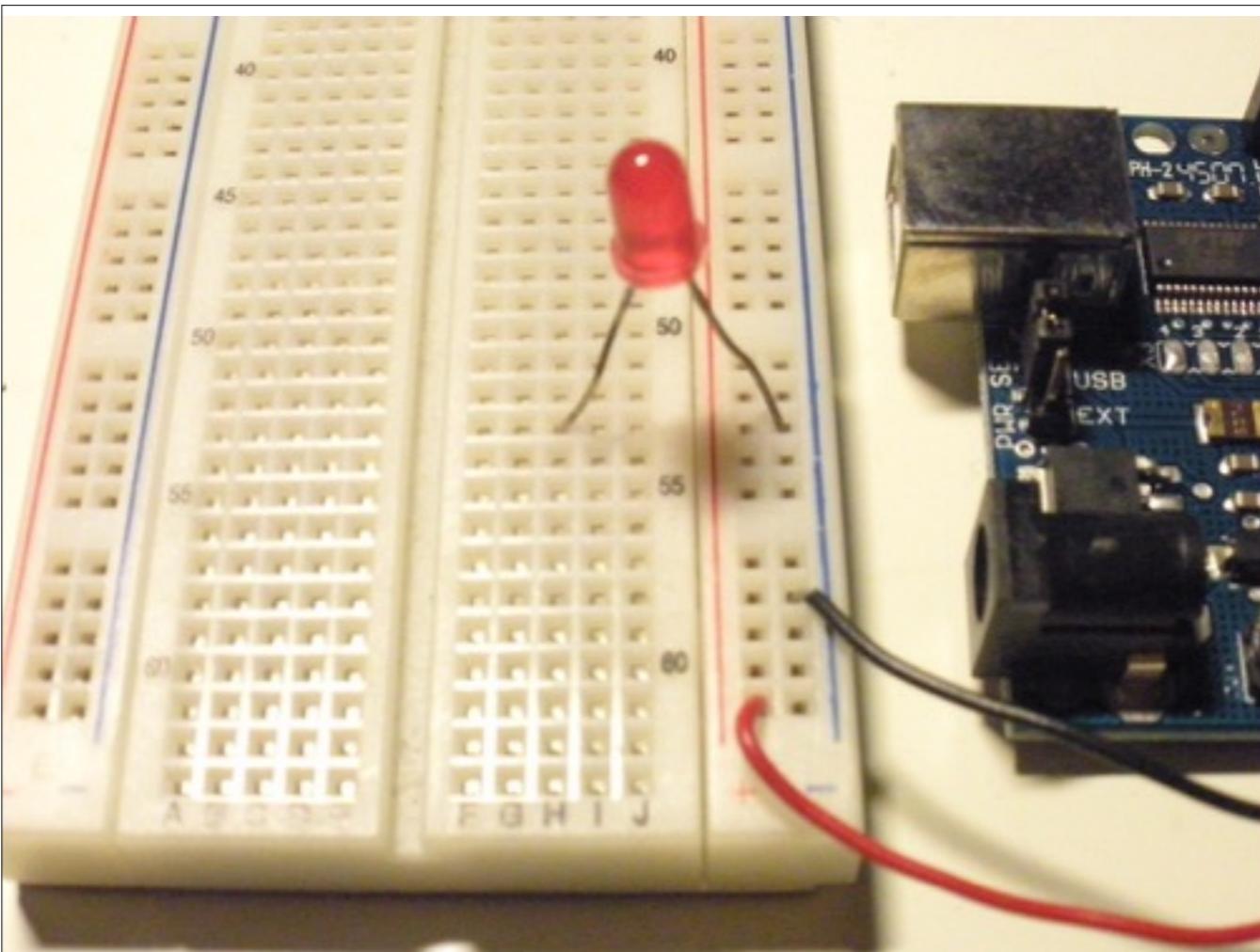




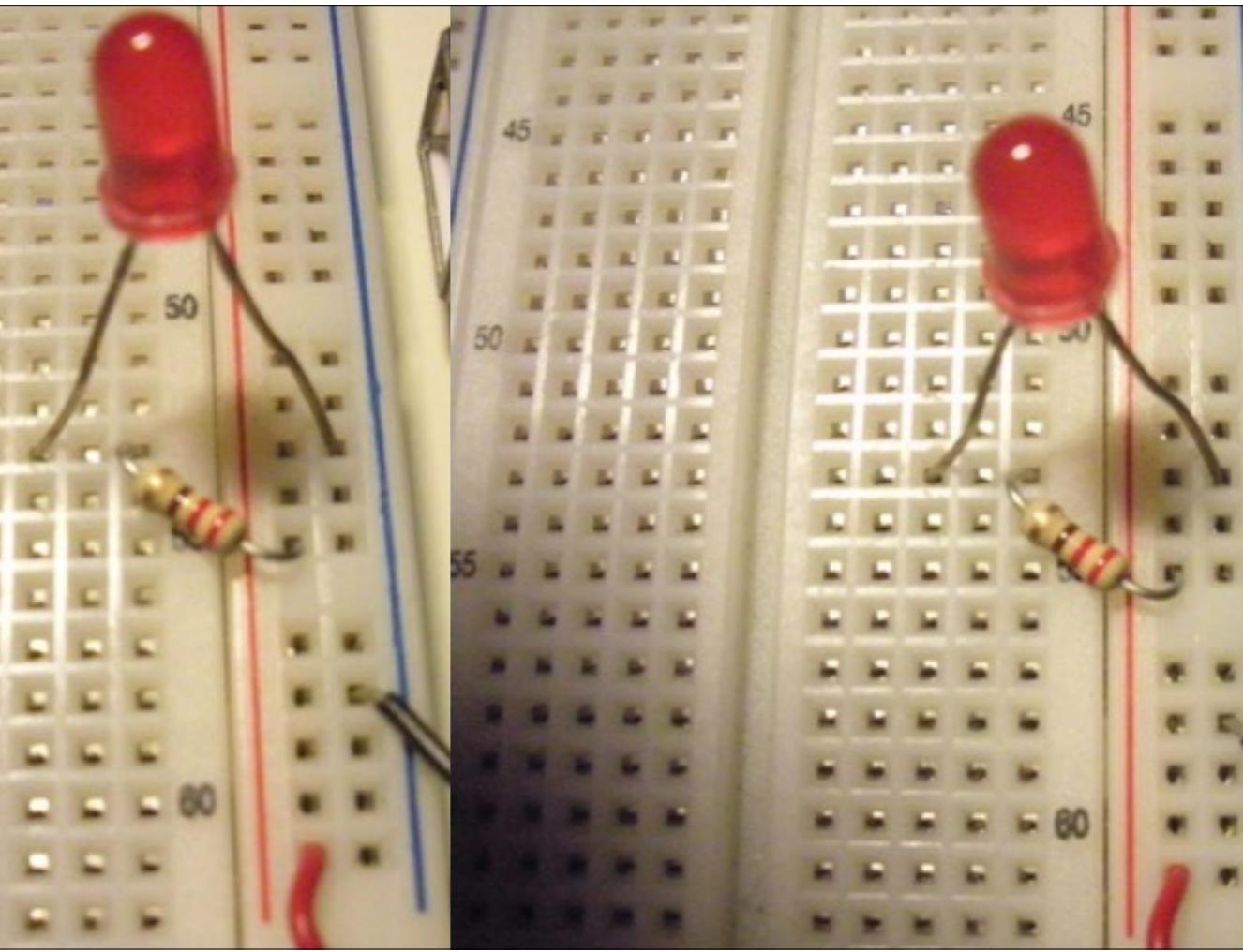
Plug the usb cable into the Arduino board and your laptop



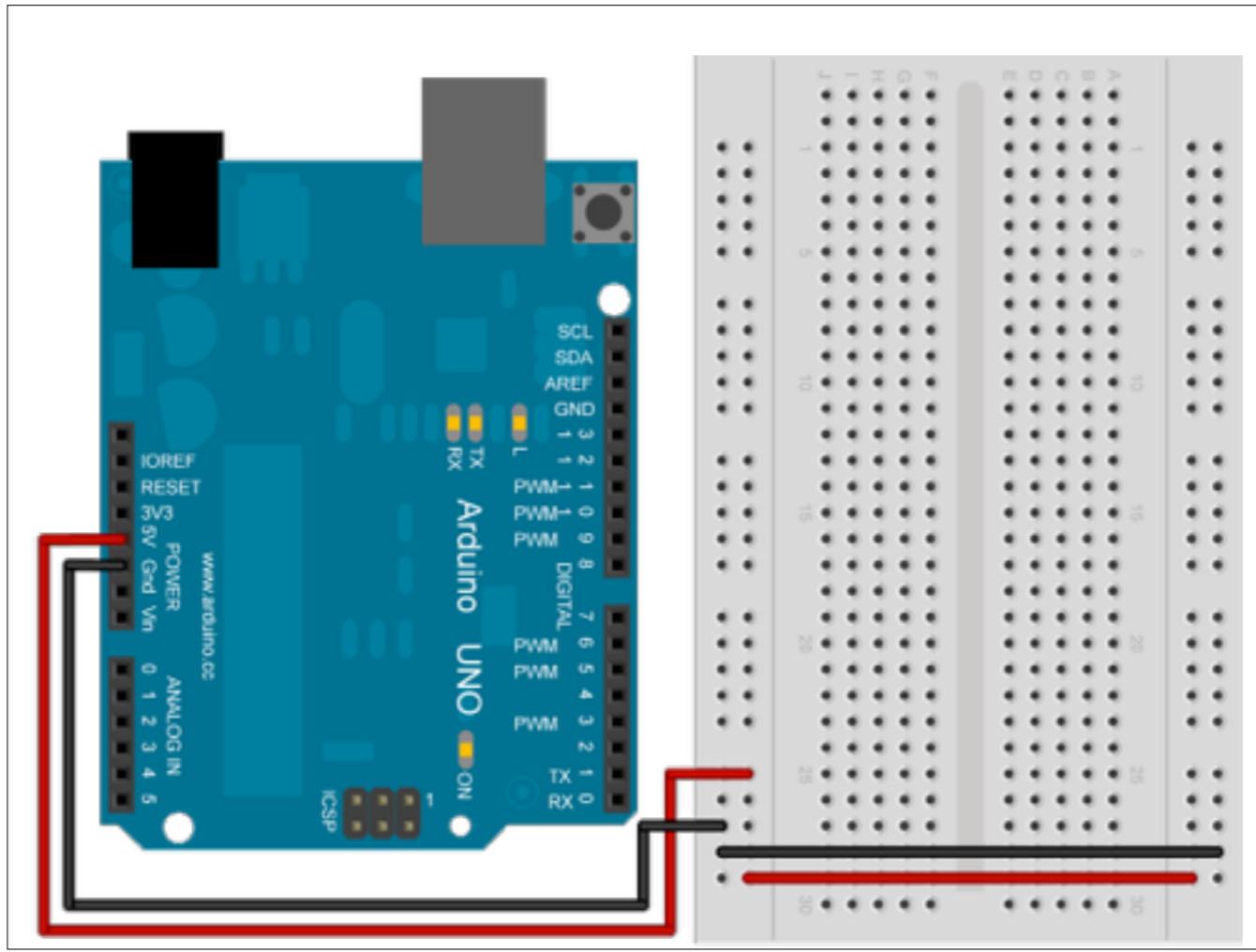
Status check



Place a test led on your breadboard. This will let us know that our breadboard is receiving power. Make sure your led is in properly – ground must go to the blue (ground) side of your board.



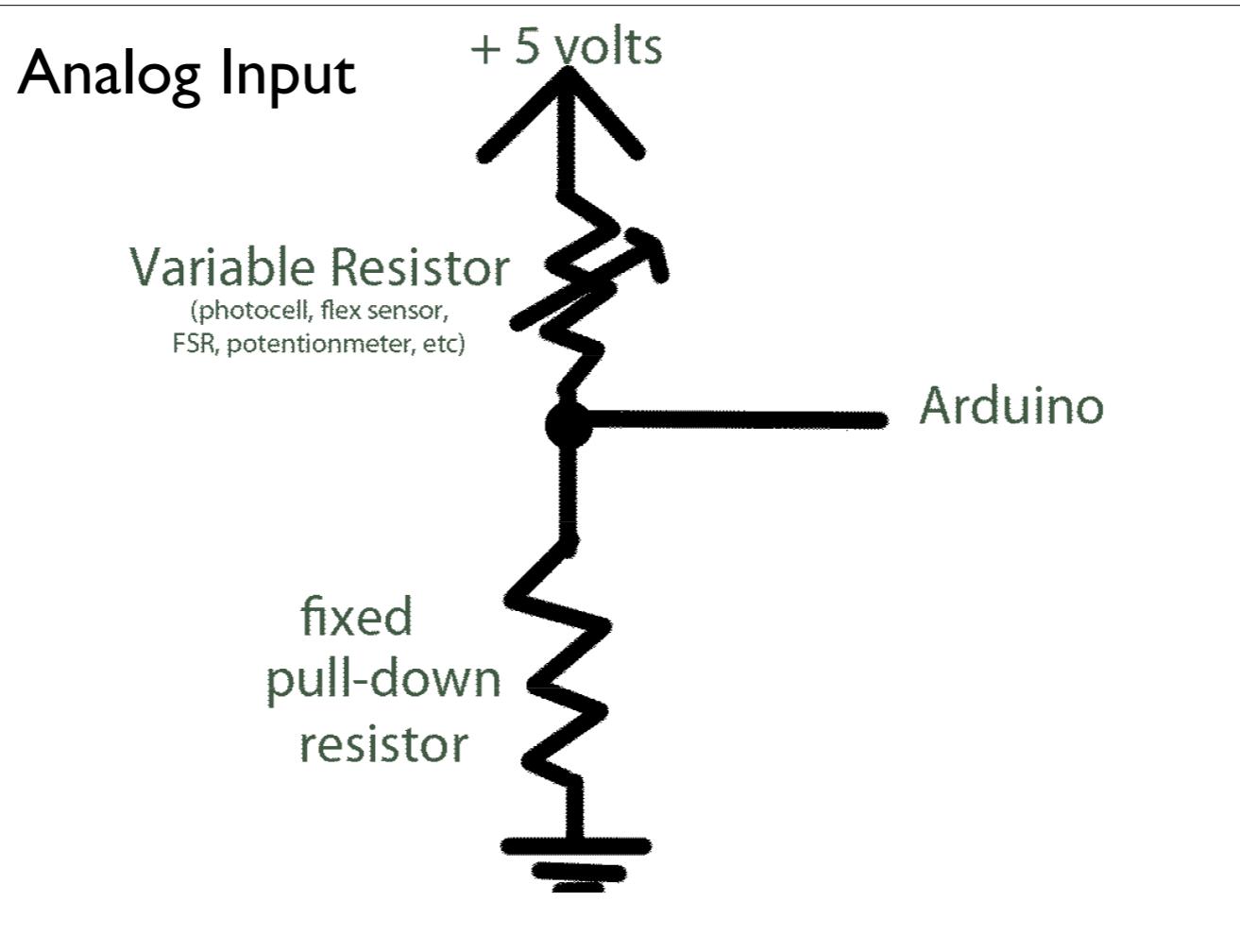
Insert 1 leg of your resistor into the same row of the led is in.
Plug the other leg of the resistor to the red (5V) side of your breadboard.



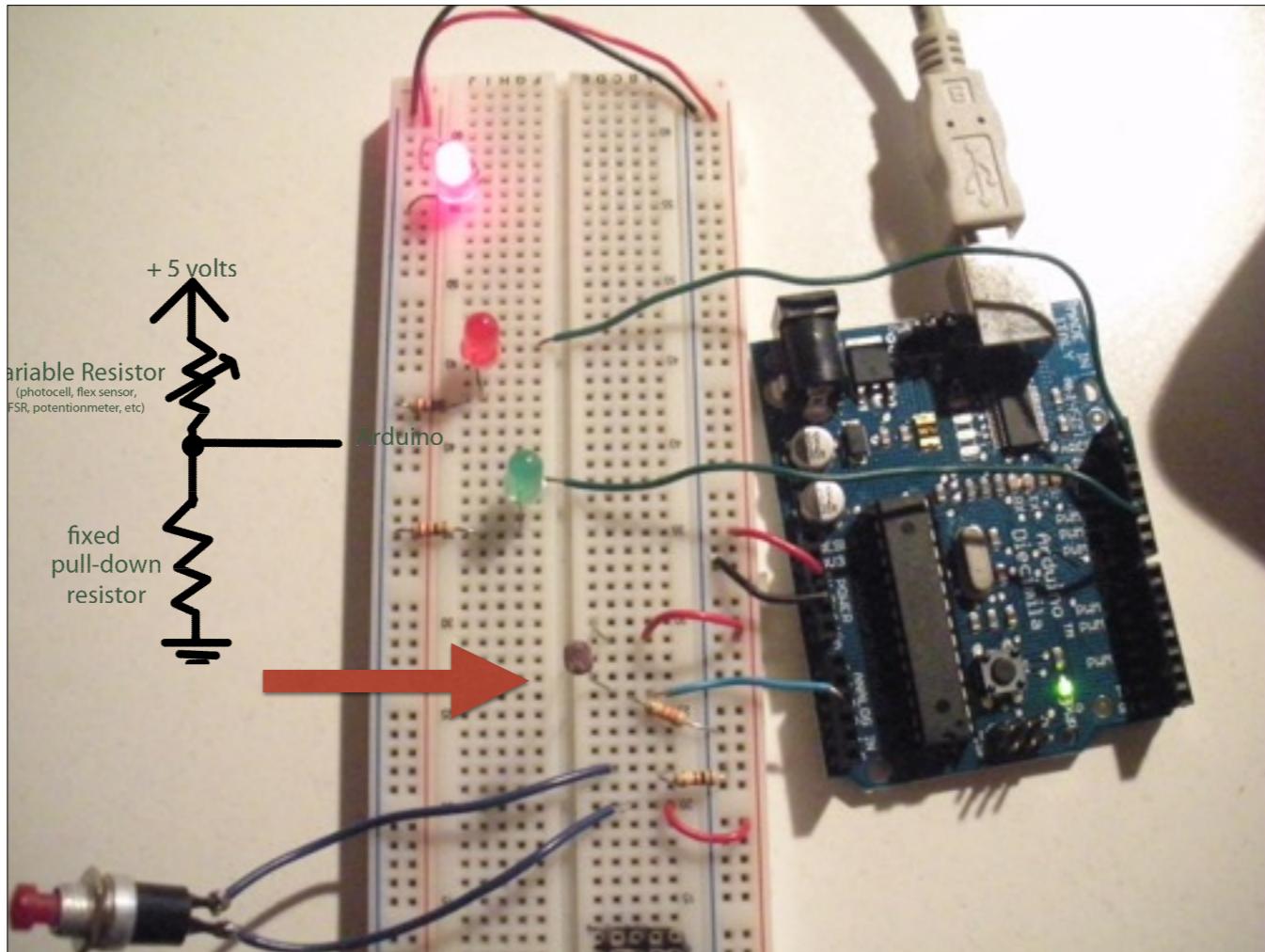
make sure your test LED is on your board as well

Input

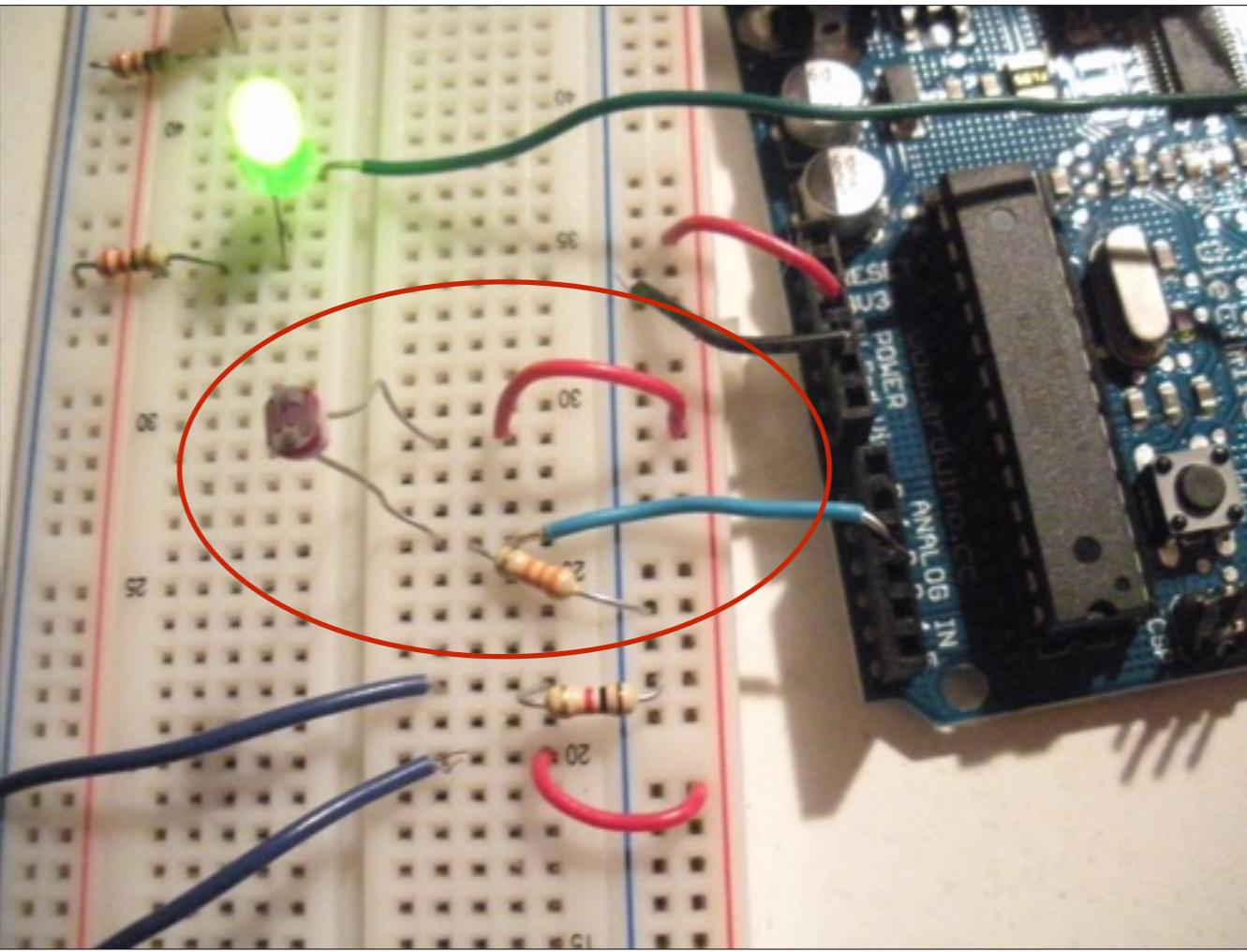
- Analog Input



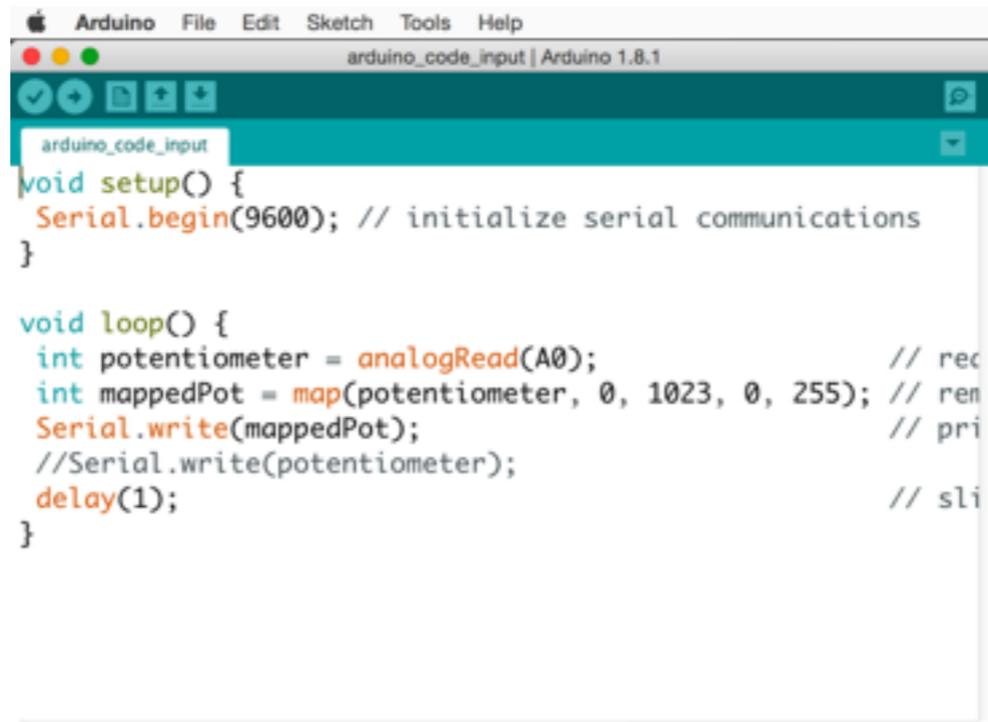
Voltage Divider



Analog In



Burn the code onto the arduino



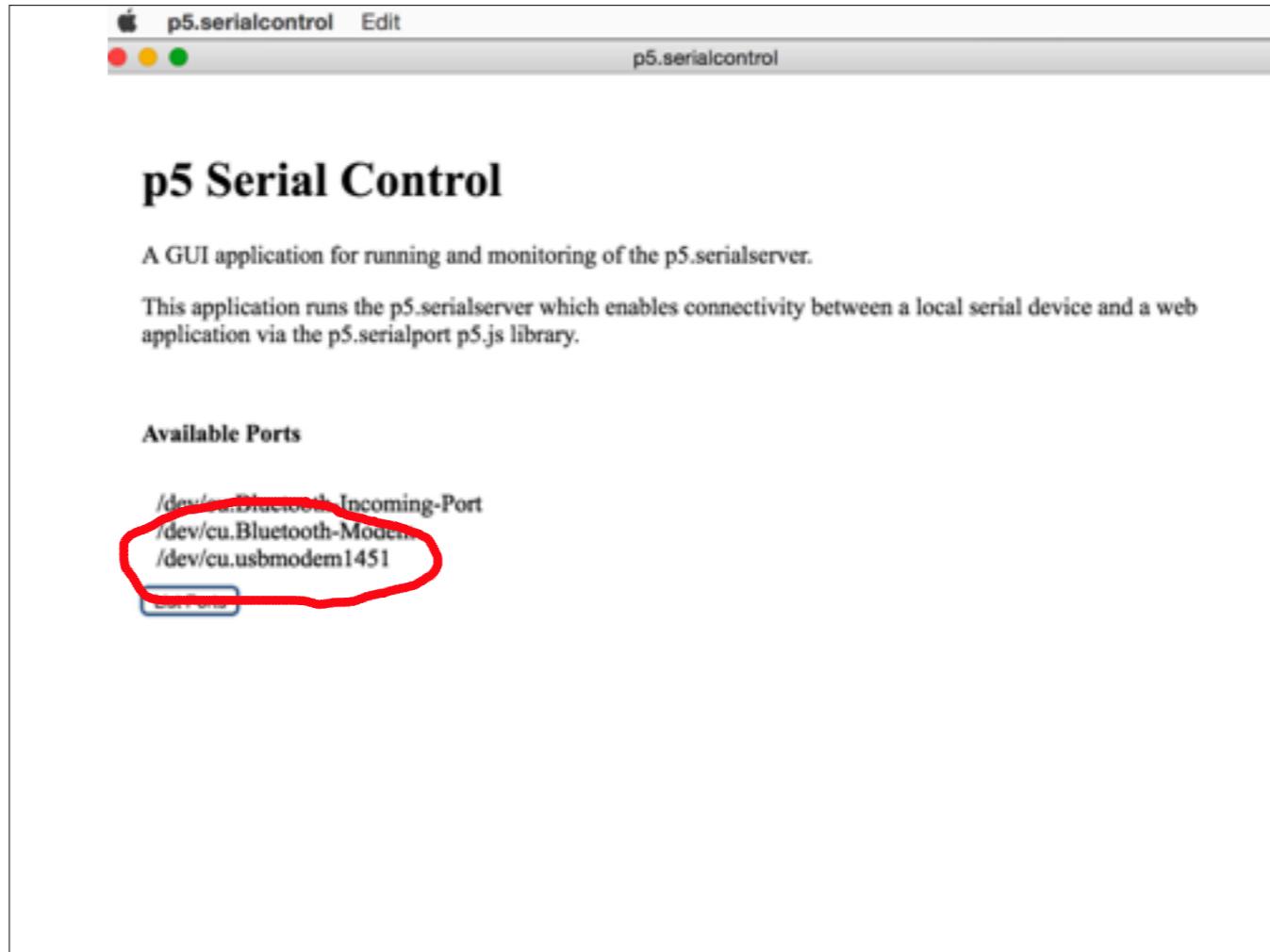
The screenshot shows the Arduino IDE interface with the following code:

```
Arduino  File  Edit  Sketch  Tools  Help
arduino_code_input | Arduino 1.8.1
arduino_code_input
void setup() {
  Serial.begin(9600); // initialize serial communications
}

void loop() {
  int potentiometer = analogRead(A0);          // read
  int mappedPot = map(potentiometer, 0, 1023, 0, 255); // ren
  Serial.write(mappedPot);                     // pri
  //Serial.write(potentiometer);
  delay(1);                                     // sli
}
```

Start the p5 Serial Control App





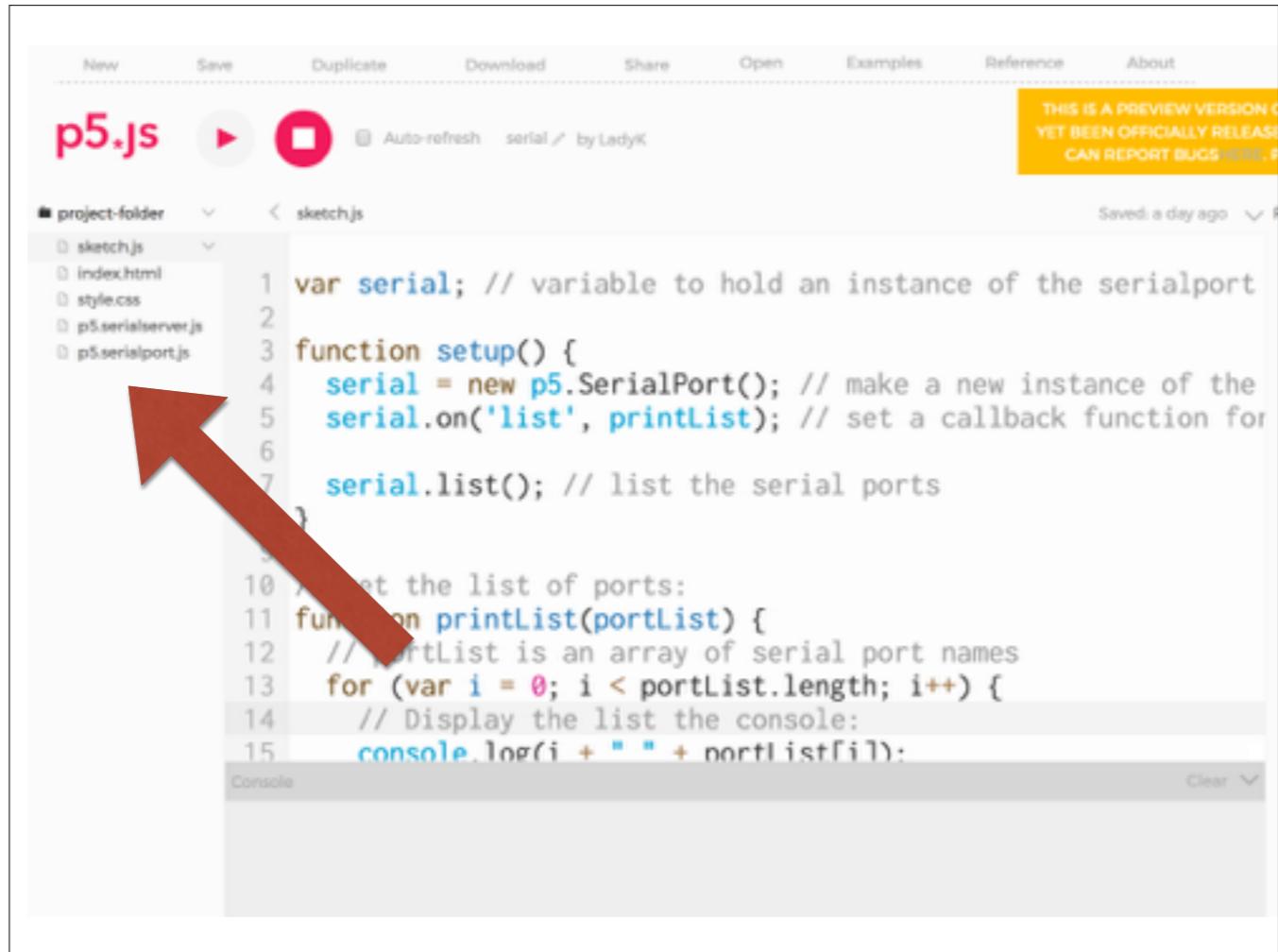
The screenshot shows the p5.js web editor interface. At the top, there is a menu bar with options: New, Save, Open, Examples, Reference, and About. Below the menu is a toolbar with icons for play/pause, stop, and refresh, followed by checkboxes for Auto-refresh and Casual part, and a preview dropdown.

The main workspace shows a file structure on the left: project-folder (sketch.js, index.html, style.css). The sketch.js file is open, displaying the following code:

```
function setup() {
  createCanvas(400, 400);
}

function draw() {
  background(220);
}
```

At the bottom of the workspace is a console tab labeled "Console" with a "Clear" button.



New Save Duplicate Download Share Open Examples Reference About

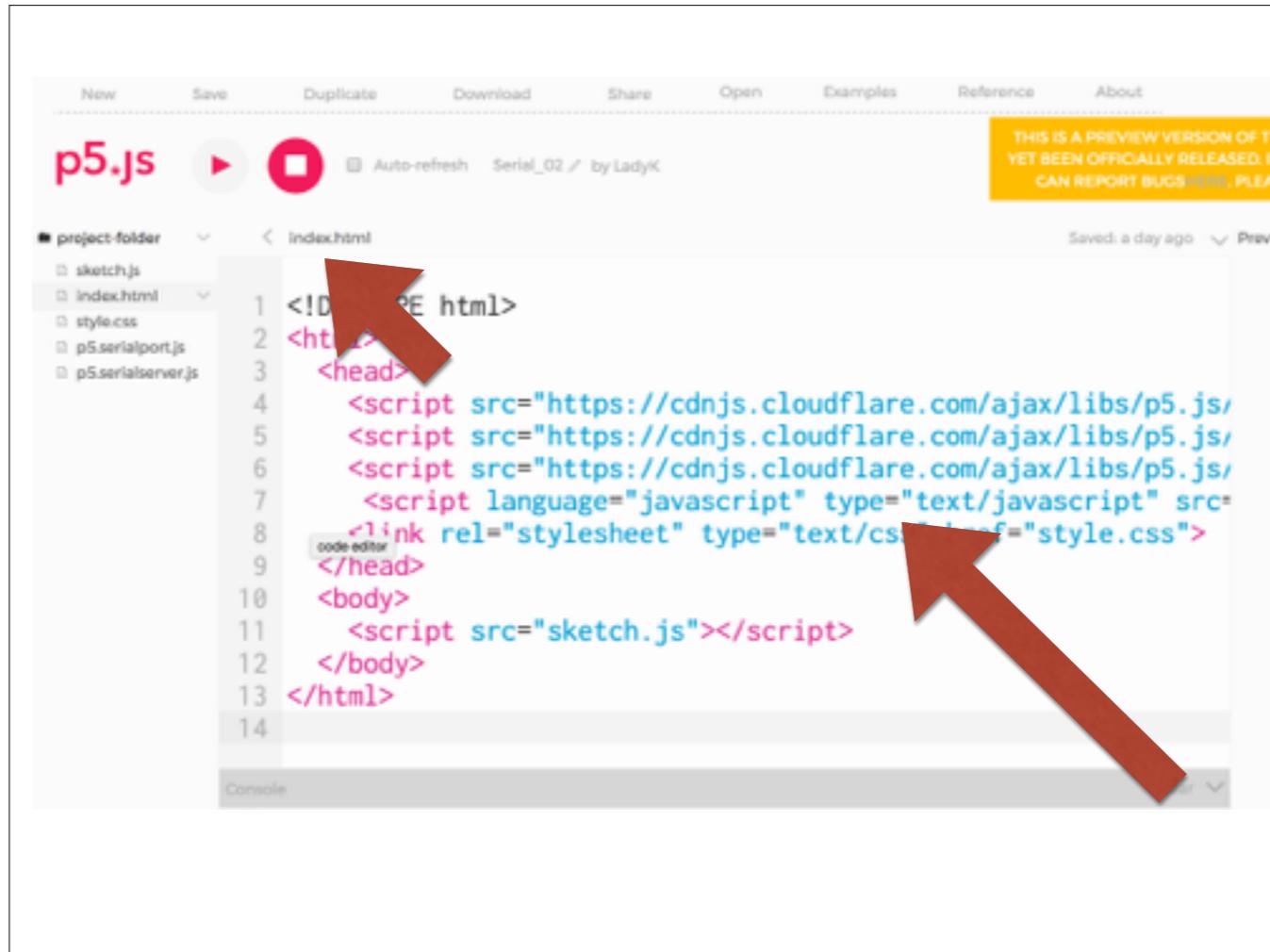
THIS IS A PREVIEW VERSION OF p5.js. IT HAS NOT YET BEEN OFFICIALLY RELEASED. PLEASE REPORT BUGS HERE.

project-folder < sketch.js

```
1 var serial; // variable to hold an instance of the serialport
2
3 function setup() {
4   serial = new p5.SerialPort(); // make a new instance of the
5   serial.on('list', printList); // set a callback function for
6   serial.list(); // list the serial ports
7 }
8
9 // Get the list of ports:
10 function printList(portList) {
11   // portList is an array of serial port names
12   for (var i = 0; i < portList.length; i++) {
13     // Display the list the console:
14     console.log(i + " " + portList[i]);
15   }
}
```

Console Clear

Serial example



The screenshot shows the p5.js code editor interface. At the top, there's a menu bar with options like New, Save, Duplicate, Download, Share, Open, Examples, Reference, and About. Below the menu, the title "p5.js" is displayed next to a play button icon. To the right of the title, it says "Auto-refresh Serial_02 / by LadyK". A yellow banner at the top right states: "THIS IS A PREVIEW VERSION OF THE LIBRARY. IT HAS NOT YET BEEN OFFICIALLY RELEASED. IT CAN REPORT BUGS HERE, PLEASE". The main area shows a file tree under "project-folder" containing sketch.js, index.html, style.css, p5.serialport.js, and p5.serialserver.js. The "index.html" file is open, showing the following code:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.4.4/p5.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.4.4/p5.dom.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.4.4/p5.serialport.js"></script>
    <script language="javascript" type="text/javascript" src="sketch.js"></script>
    <link rel="stylesheet" type="text/css" href="style.css">
  </head>
  <body>
    <script src="sketch.js"></script>
  </body>
</html>
```

Two large red arrows point from the text above the image to the opening tags of the head and body sections of the HTML code.

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

New Save Duplicate Download Share Open Examples Reference About

p5.js Auto-refresh serial by LadyK

THIS IS A PREVIEW VERSION OF
YET BEEN OFFICIALLY RELEASED
CAN REPORT BUGS HERE, PLU

Saved: a day ago Pre

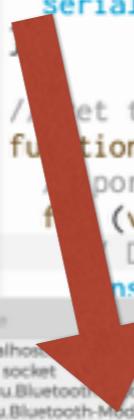
ide editor project folder < sketch.js

sketch.js Index.html style.css p5serialserver.js p5serialport.js

```
1 var serial; // variable to hold an instance of the serialport
2
3 function setup() {
4   serial = new p5.SerialPort(); // make a new instance of the
5   serial.on('list', printList); // set a callback function for
6
7   serial.list(); // list the serial ports
8 }
9
10 // Set the list of ports:
11 function printList(portList) {
12   // portList is an array of serial port names
13   for (var i = 0; i < portList.length; i++) {
14     // Display the list the console:
15     console.log(i + " " + portList[i]);
16   }
17 }
```

Console Clear

ws://localhost:3000 opened socket
0 /dev/cu.Bluetooth-Incoming-Port
1 /dev/cu.Bluetooth-Modem
2 /dev/cu.usbmodem1451
0 /dev/cu.Bluetooth-Incoming-Port
1 /dev/cu.Bluetooth-Modem
2 /dev/cu.usbmodem1451



The screenshot shows the p5.js editor interface. At the top, there's a menu bar with options like New, Save, Duplicate, Download, Share, Open, Examples, Reference, and About. Below the menu is a toolbar with icons for play, stop, and refresh, followed by status indicators for Auto-refresh and the current sketch name, "Serial_02" by LadyK. A yellow banner at the top right states, "THIS IS A PREVIEW VERSION OF p5.js, WHICH HAS NOT YET BEEN OFFICIALLY RELEASED. PLEASE DO NOT USE IN PRODUCTION. YOU CAN REPORT BUGS HERE." The main area is a code editor titled "sketch.js". On the left, a sidebar shows the project folder structure: "project-folder" containing "sketch.js", "index.html", "style.css", "p5.serialport.js", and "p5.serialserver.js". The "sketch.js" file contains the following code:

```
1 var serial; // variable to hold an instance of the serialport
2 var portName = '/dev/cu.usbmodem1411'; // fill in your serial
3 var inData;
4
5 function setup() {
6   serial = new p5.SerialPort(); // make a new instance of the
7   serial.on('list', printList); // set a callback function for
8   serial.on('connected', serverConnected); // callback for cor
9   serial.on('open', portOpen); // callback for the port openir
10  serial.on('data', serialEvent); // callback for when new dat
11  serial.on('error', serialError); // callback for errors
12  serial.on('close', portClose); // callback for the port clos
13
14  //serial.list(); // list the serial ports
15
```

Below the code editor is a "Console" window showing the output of the serial port listing command:

```
ws://localhost:8081
opened socket
0 /dev/cu.Bluetooth-Incoming-Port
1 /dev/cu.Bluetooth-Modem
2 /dev/cu.usbmodem1451
0 /dev/cu.Bluetooth-Incoming-Port
1 /dev/cu.Bluetooth-Modem
2 /dev/cu.usbmodem1451
```

change your port name

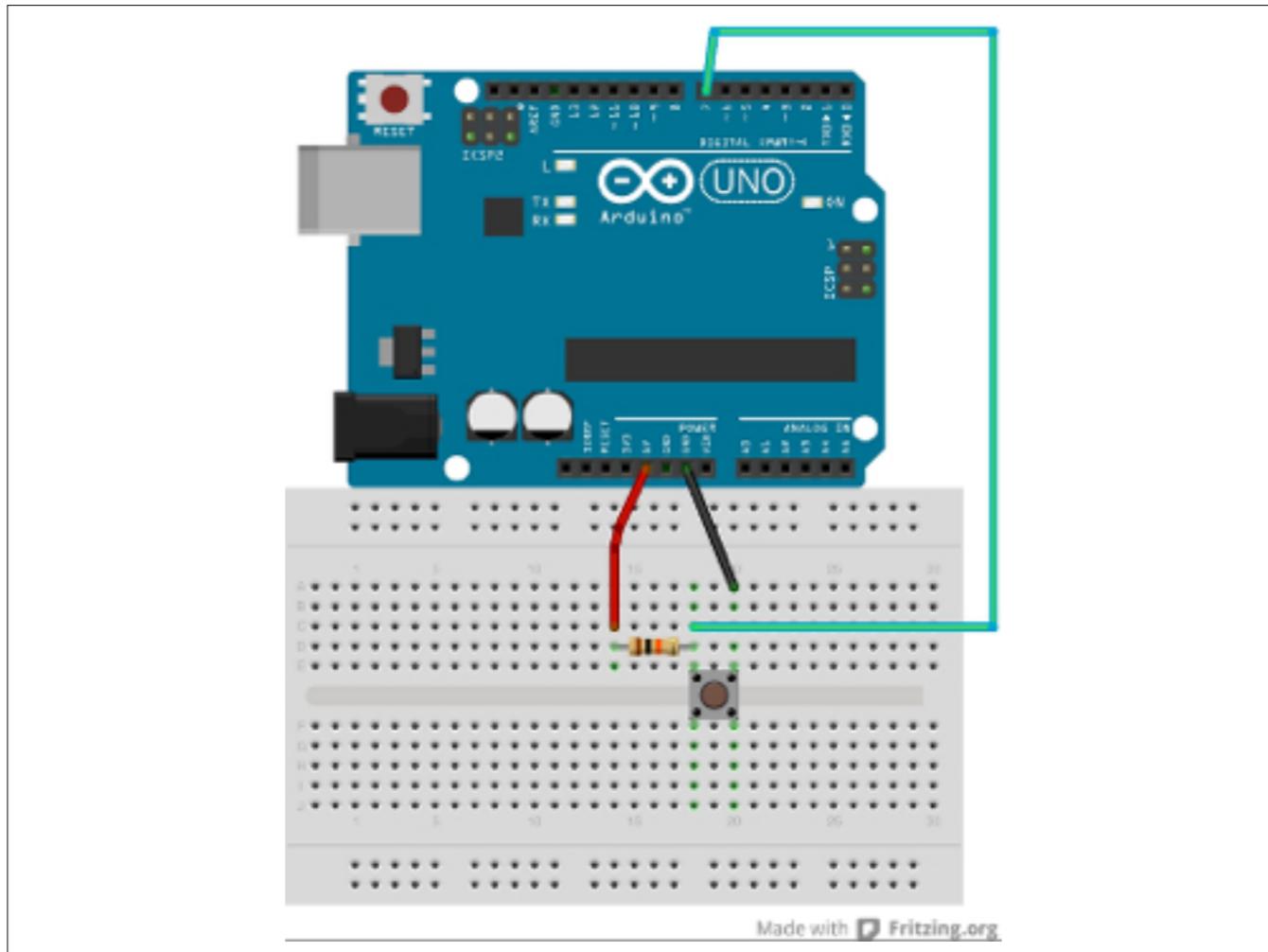
Input

- Digital switch

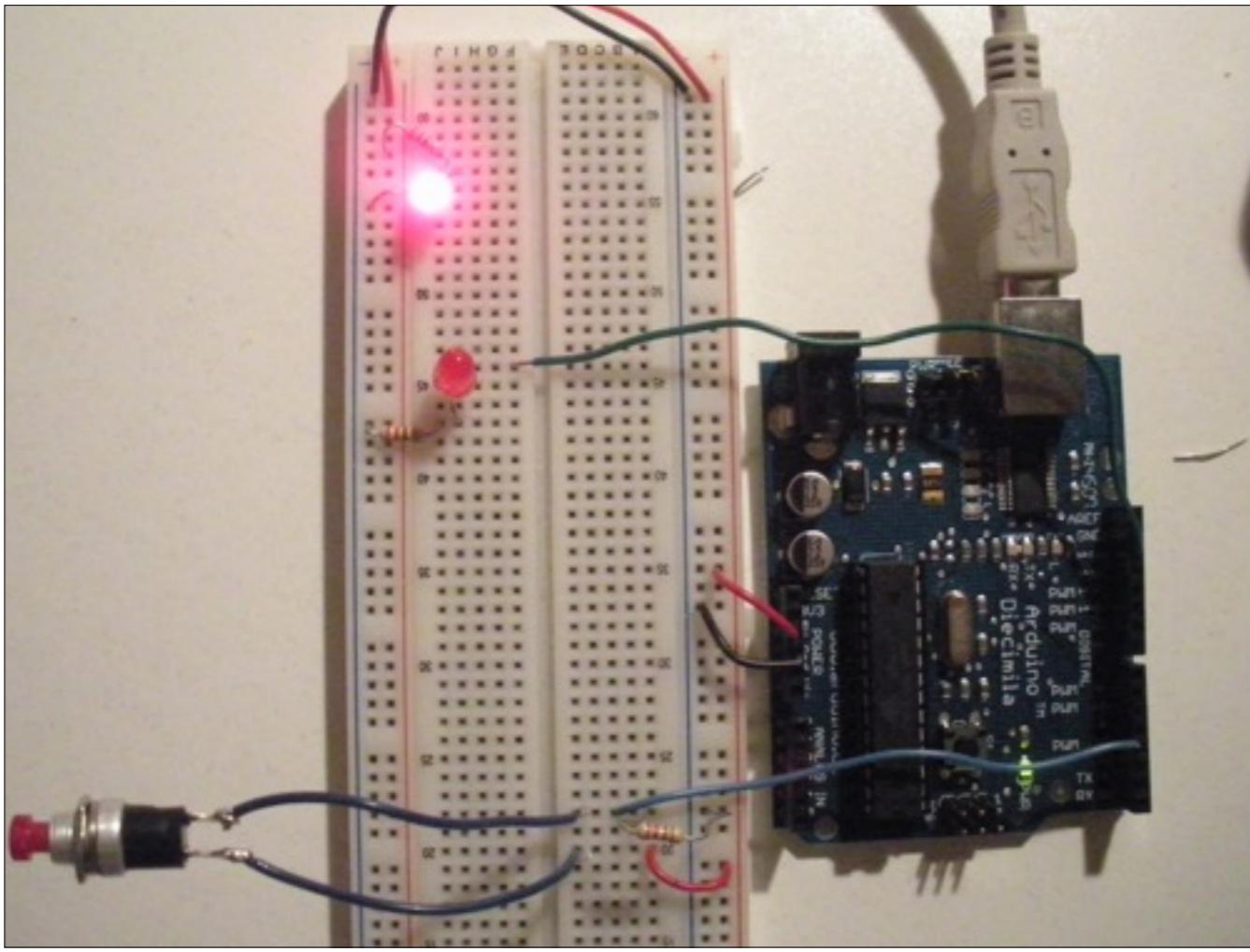
Burn the Digital Output code
on the Arduino

Start the p5 Serial Control App

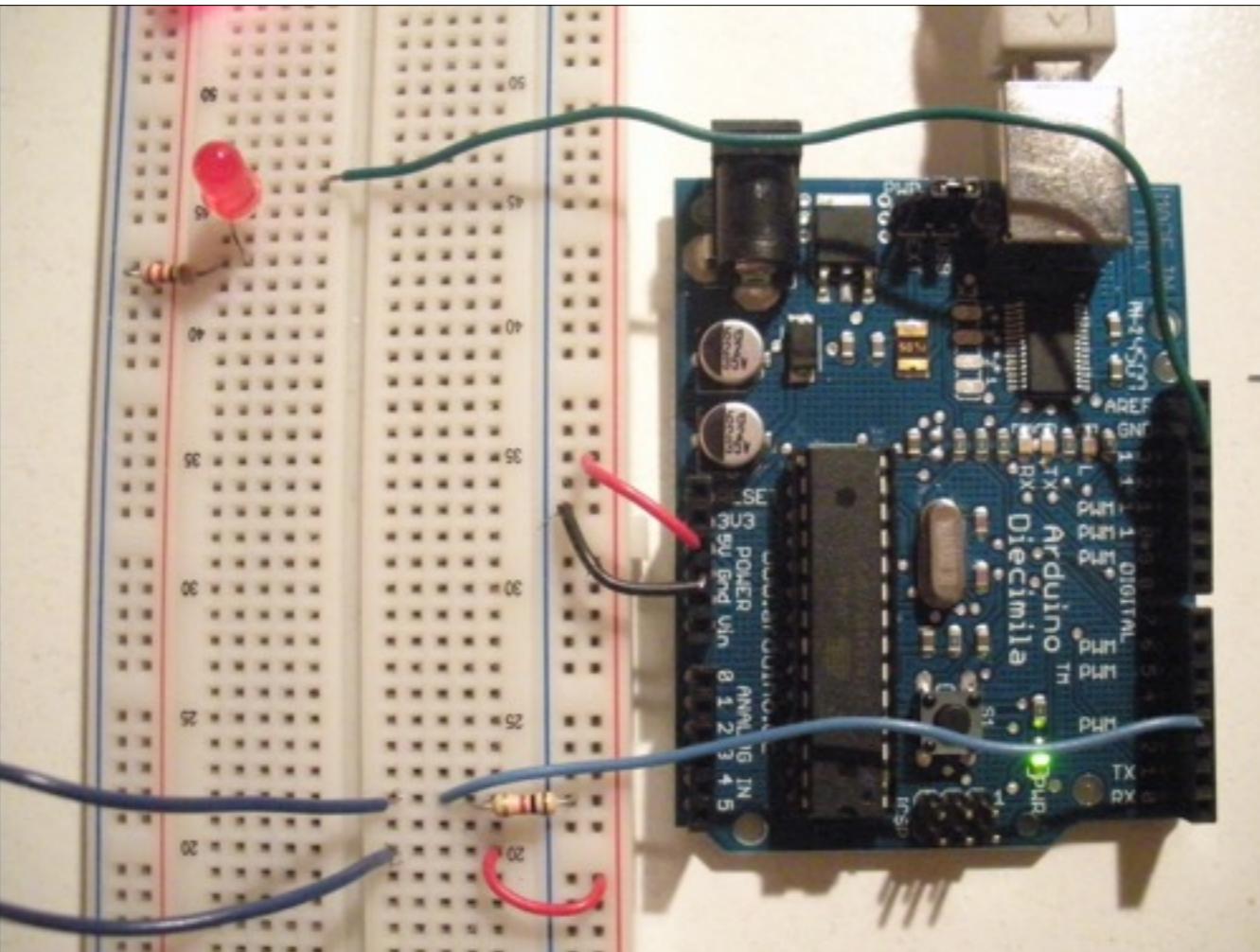


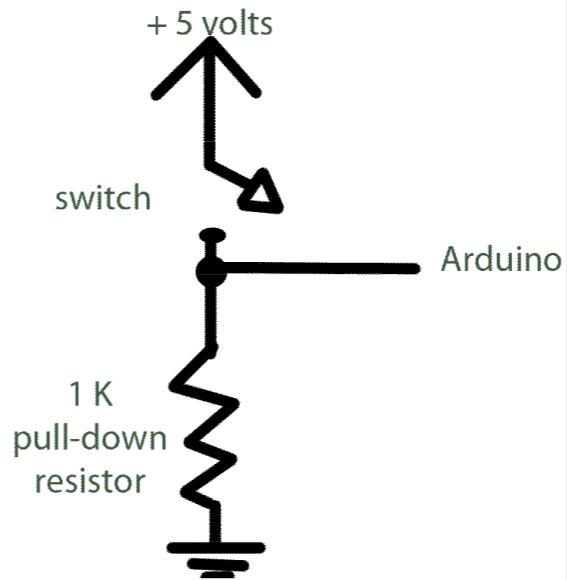
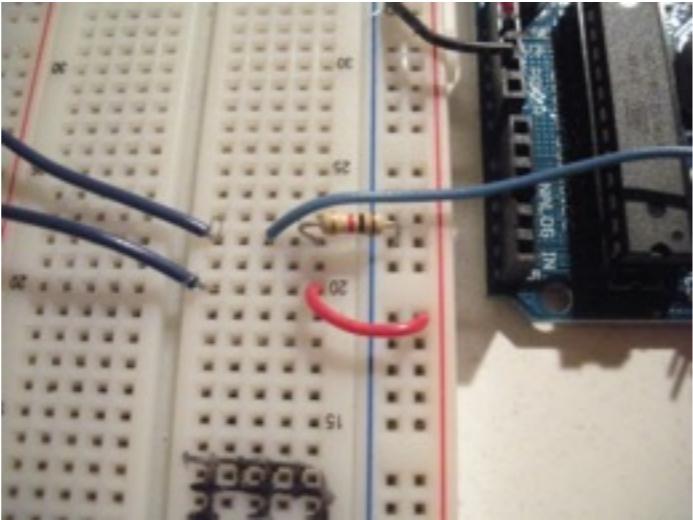


Made with Fritzing.org



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





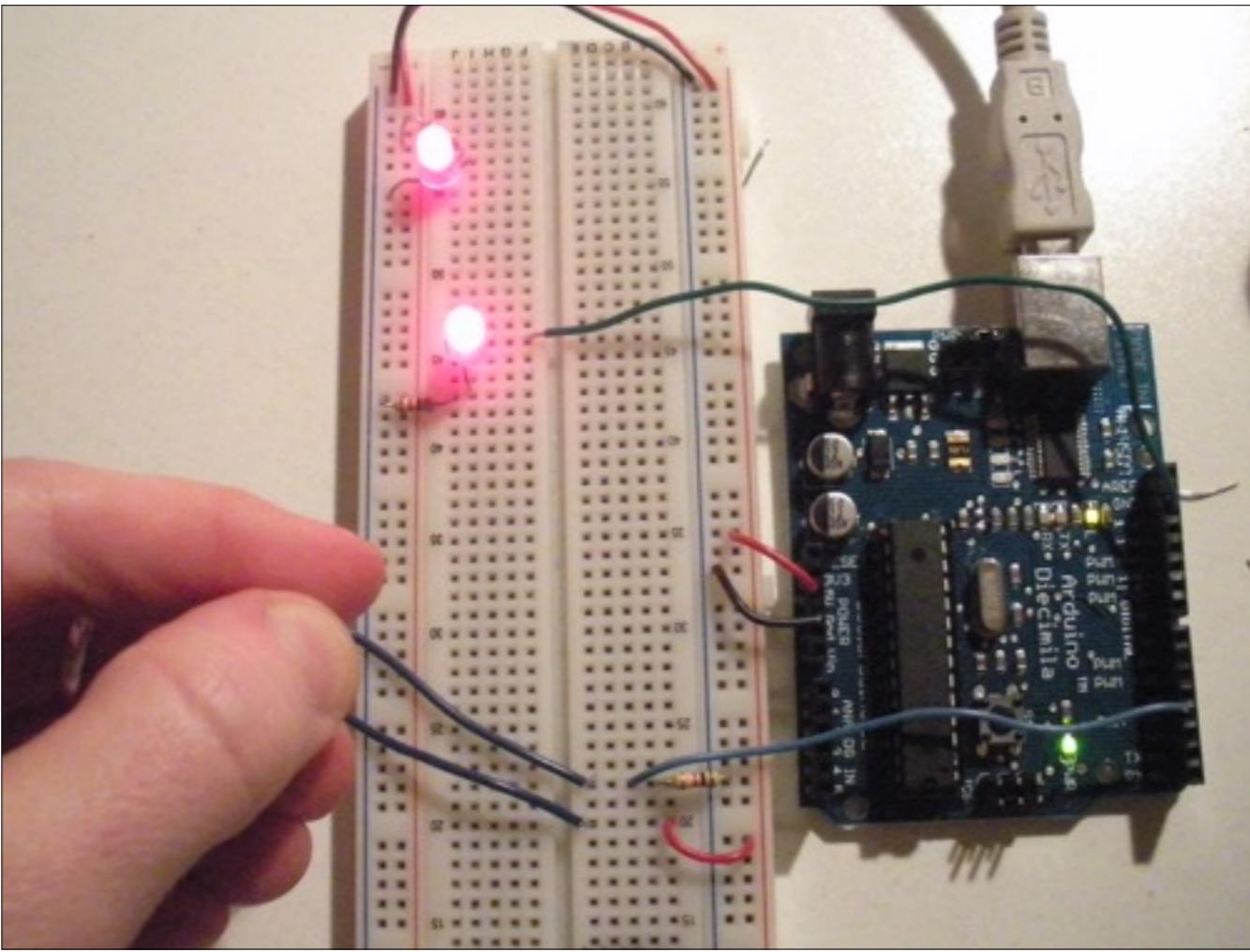
A Switch

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.

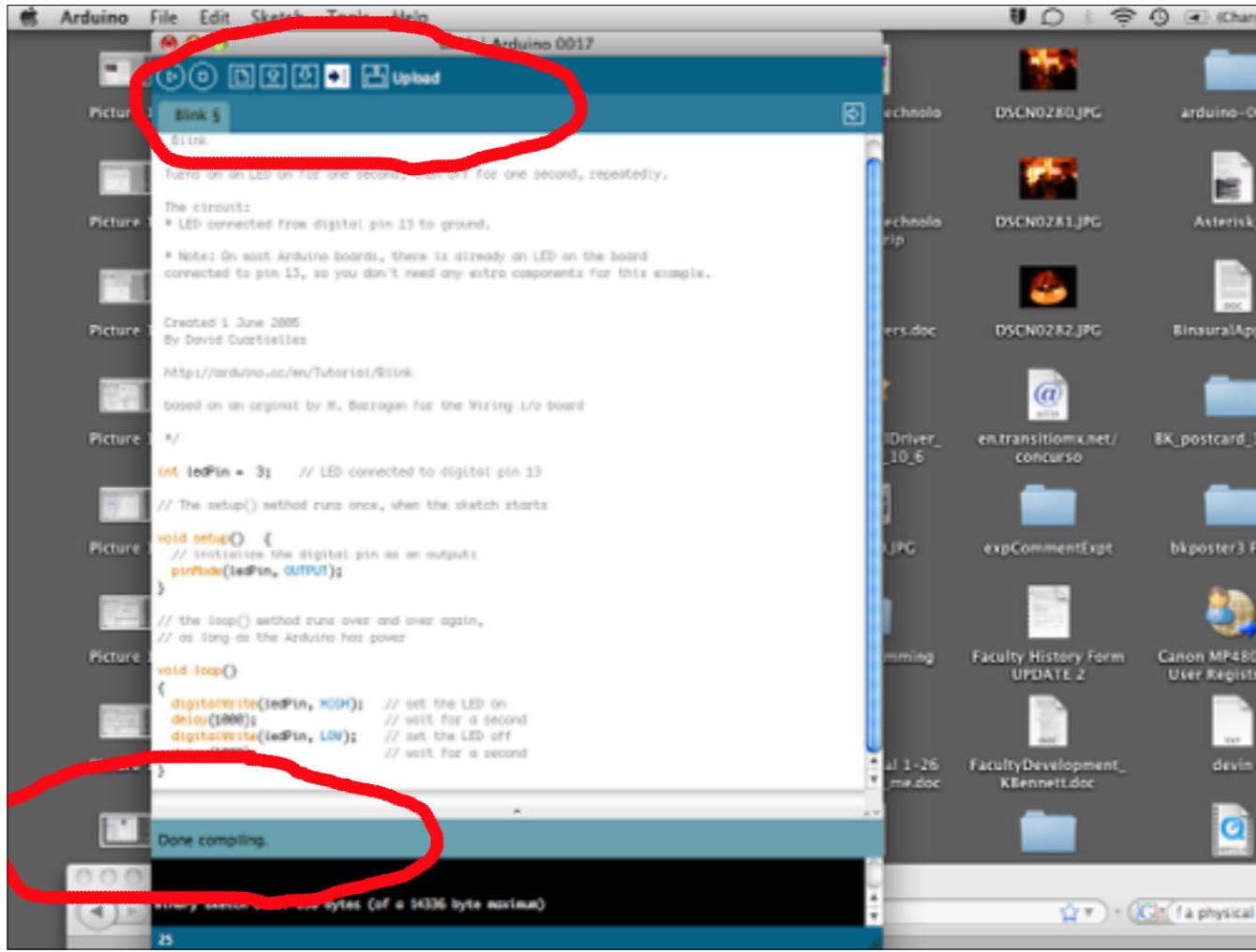
The resistor here prevents the Arduino from reading low voltages in the atmosphere.

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.

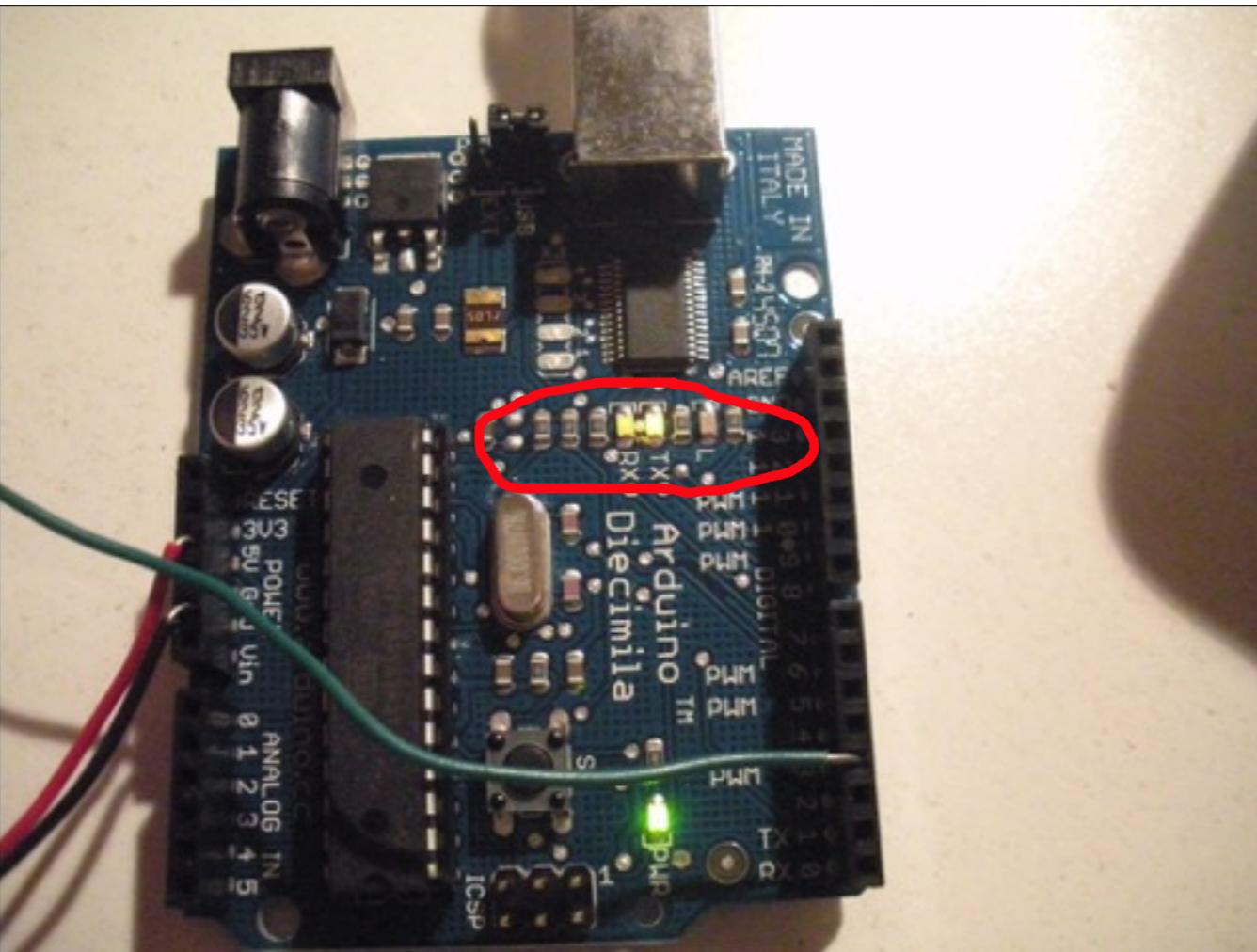


Make sure you have the correct ports and chip selected in the Arduino Program.

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.

Custom DIY Switches



EXAMPLE PROJECTS

WORKSHOPS

ACTUATORS

CONNECTIONS

POWER

SENSORS

TRACES

CIRCUITS AND CODE

WIRELESS

CONDUCTIVE

MATERIALS

NON-CONDUCTIVE MATERIALS

TOOLS

TECHNIQUES

THINKING OUT LOUD

SENSORS

3D PRINTED SENSORS
ANALOG PIN STROKE SENSOR
ANTI-STATIC FOAM PRESSURE MATRIX
BALLOON SENSOR
CAPACITIVE FABRIC SLIDERS AND WHEELS
CIRCULAR KNIT INFLATION SENSOR
CIRCULAR KNIT STRETCH SENSORS
CONDUCTIVE POMPOM
CONSTRUCTED STRETCH SENSORS
CROCHET BUTTON
CROCHET CONDUCTIVE BEAD
CROCHET FINGER SENSOR
CROCHET PRESSURE SENSOR
CROCHET TILT POTENTIOMETER
CROCHET/KNIT PRESSURE SENSORS
CROCHET/KNIT SQUEEZE SENSORS
DANISH KROWN SLIDE SWITCH
RESISTIVE SENSORS
ELASTIC BUTTON FABRIC
EMBROIDERED POTENTIOMETERS
FABRIC BUTTON
FABRIC POTENTIOMETER
FABRIC STRETCH SENSORS

WELCOME TO THE KOBAKANT DIY WEARABLE TECHNOLOGY DOCUMENTATION

This website aims to be a comprehensible, accessible and maintainable reference resource, as well as a basis for further exploration and contribution.

MOST RECENT POSTS

Sensors

CAPACITIVE FABRIC SLIDERS AND WHEELS

Guidelines CAPACITIVE SENSING MADE EASY!, Part 2 DesignGuidelines: >> <http://www.cypress.com/file/114086/download> Capacitive Touch Sensing Layout Guidelines: >> <http://www.mouser.com/pdfdocs/semitech-capacitive-touch-sensing-layout-guidelines.pdf> Hardware Design for Capacitive Touch: >> <https://www.silabs.com/documents/public/application-notes/AN0040.pdf> CHMAT publication An Accessible Platform for Exploring Haptic Interactions with Co-located Capacitive and Piezoresistive Sensors >> <http://dl.acm.org/citation.cfm?id=2680571> TI capsenslibrary >> <http://www.ti.com/tool/capsenslibrary> Pressure and Distance Sensing by admanchoonen More in depth theory [...]

Workshops

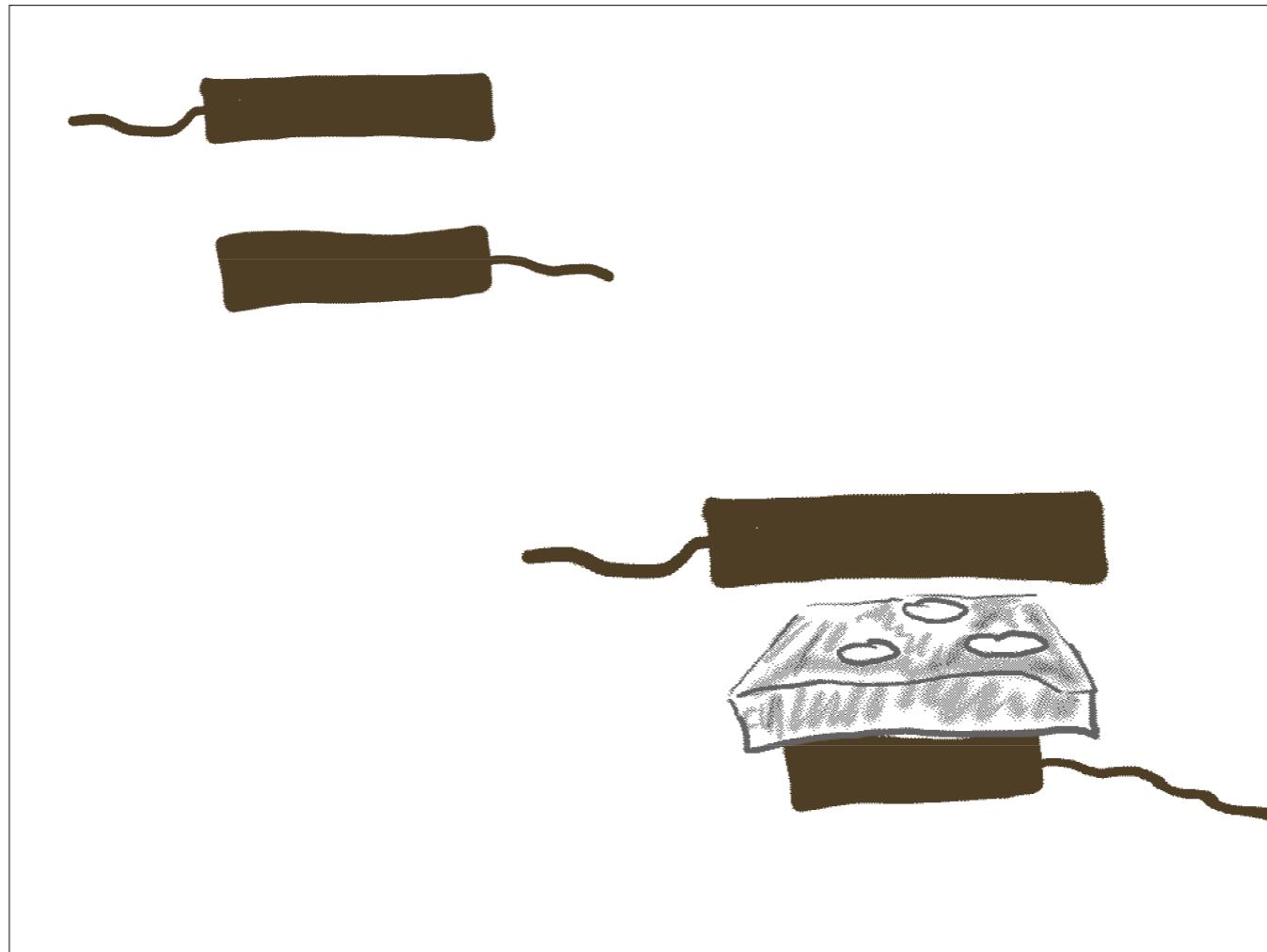
TRANSPARENT AND DANGEROUS WORKSHOP

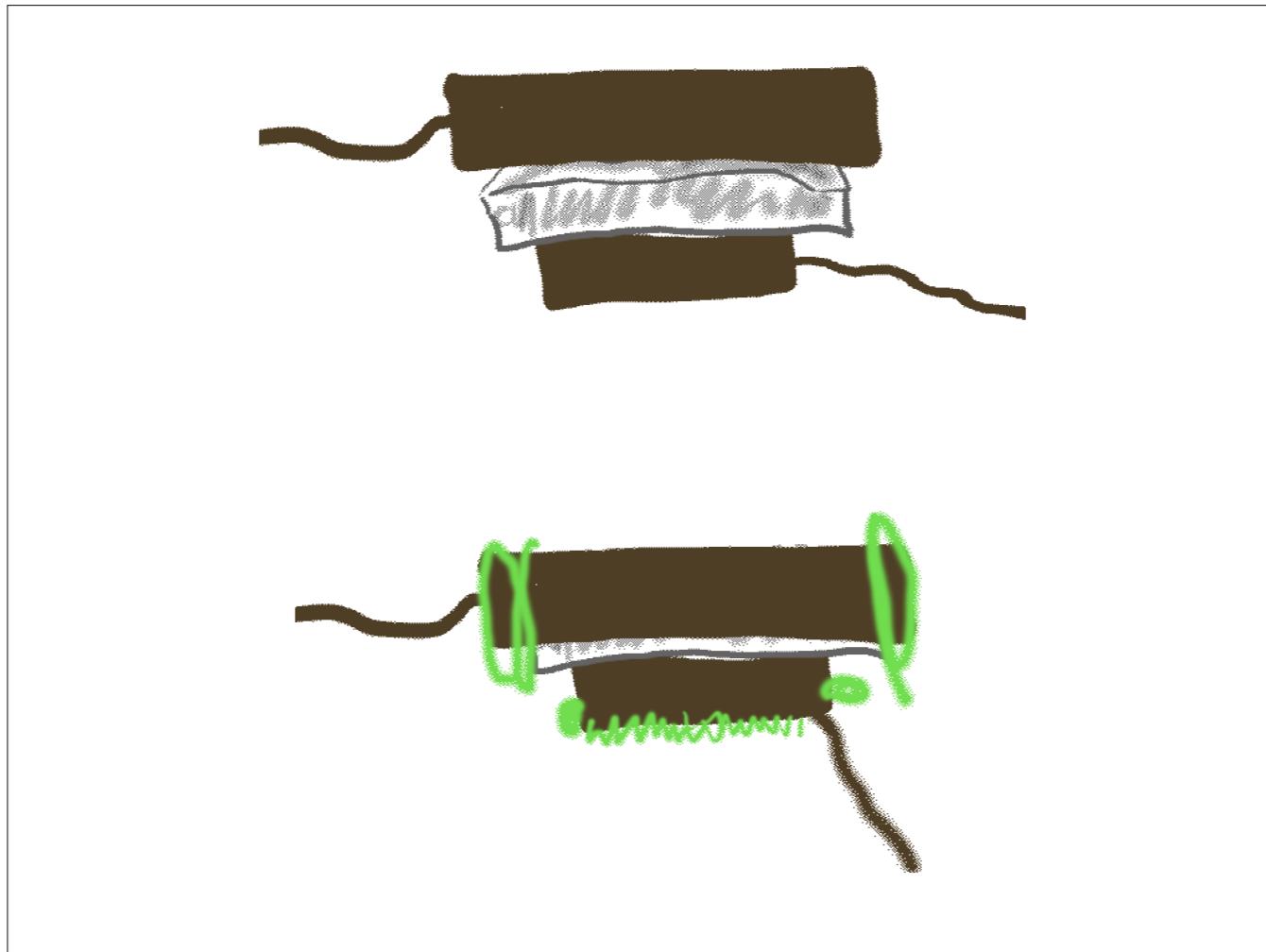


Monday 24 April 2017, 12:30 - 15:20 at the Center for Digital Arts and Experimental Media (CDARTS) at the University of Washington in Seattle, USA. This workshop will only be open to

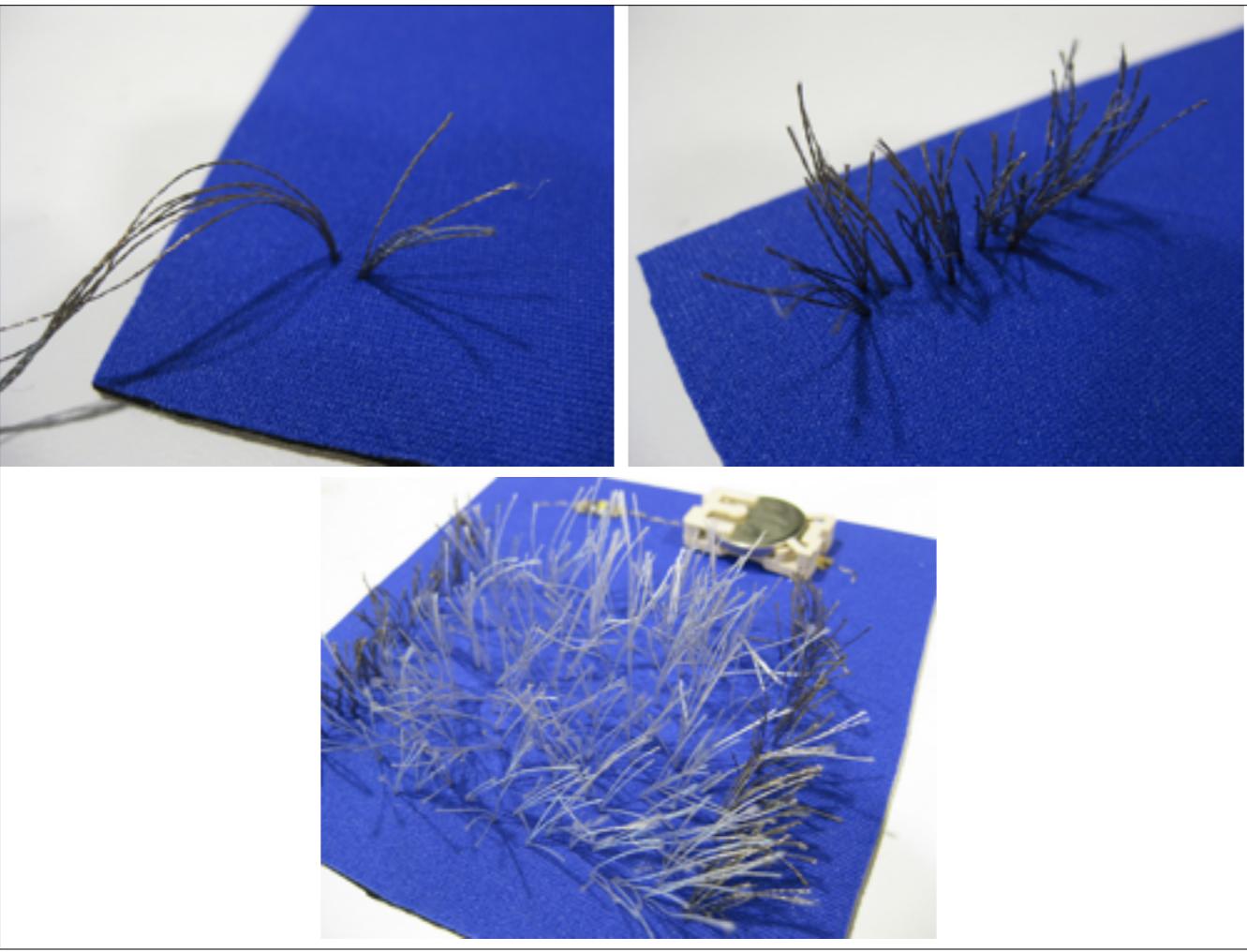
Custom DY Switches
conductive materials

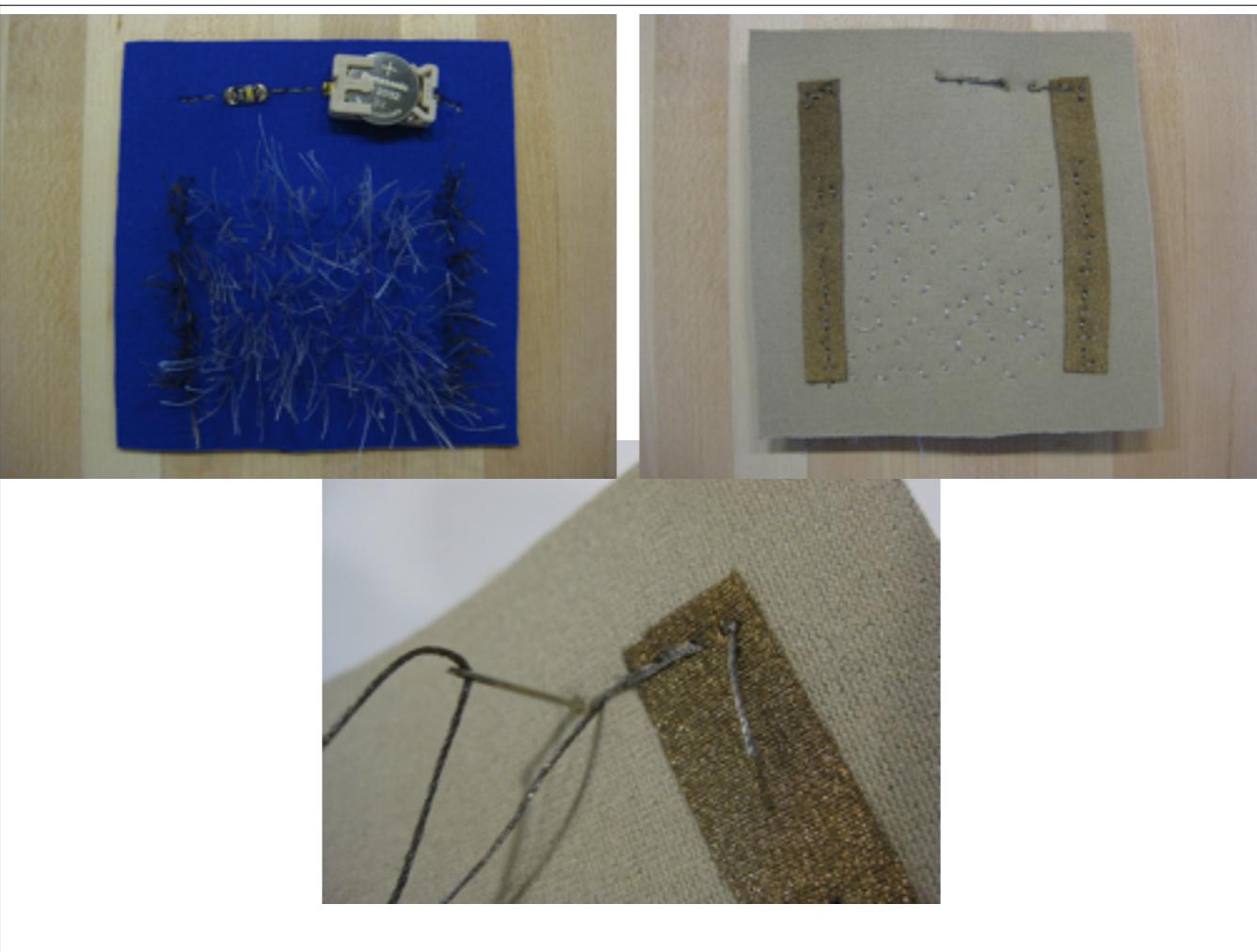
<http://www.kobakant.at/DIY/>

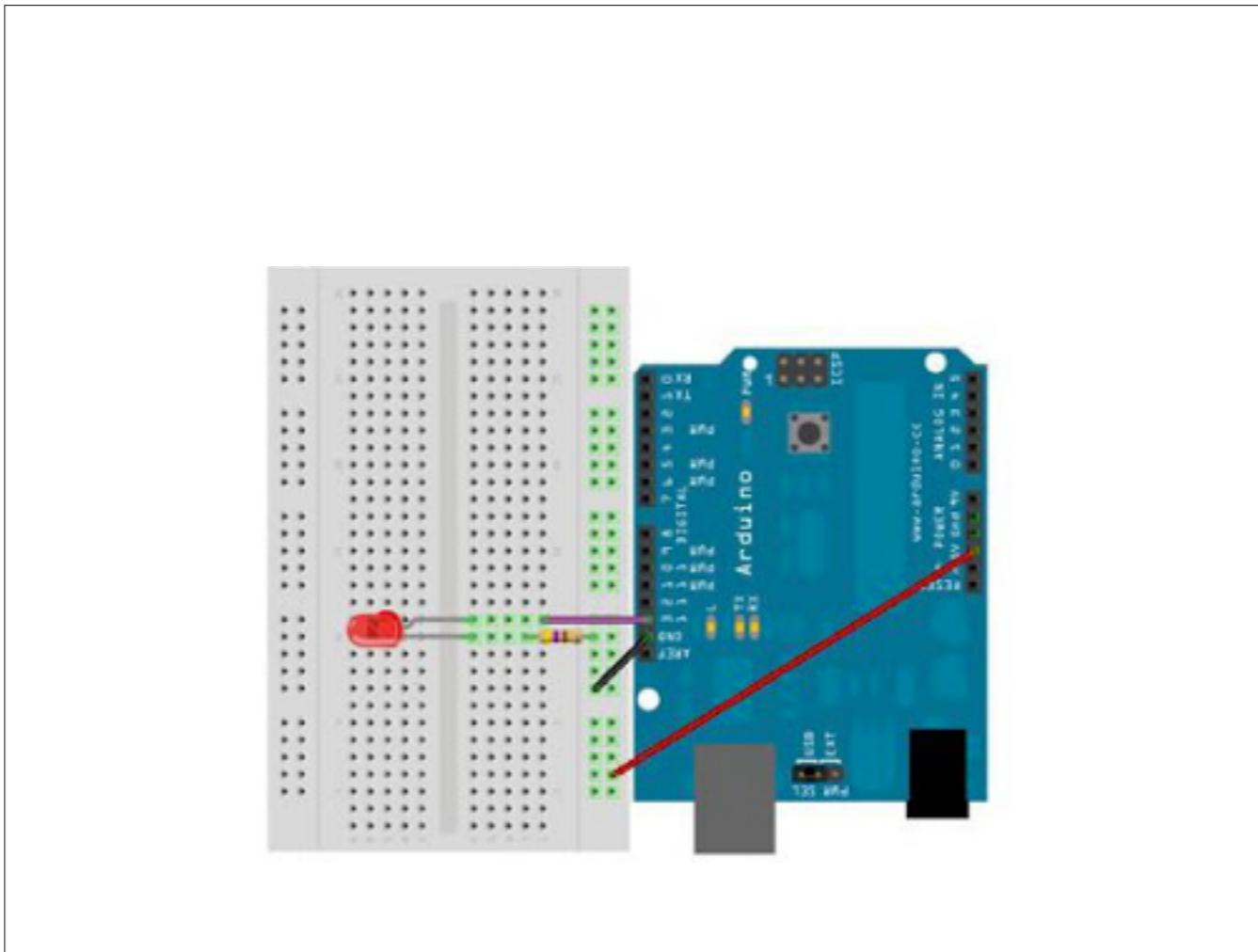




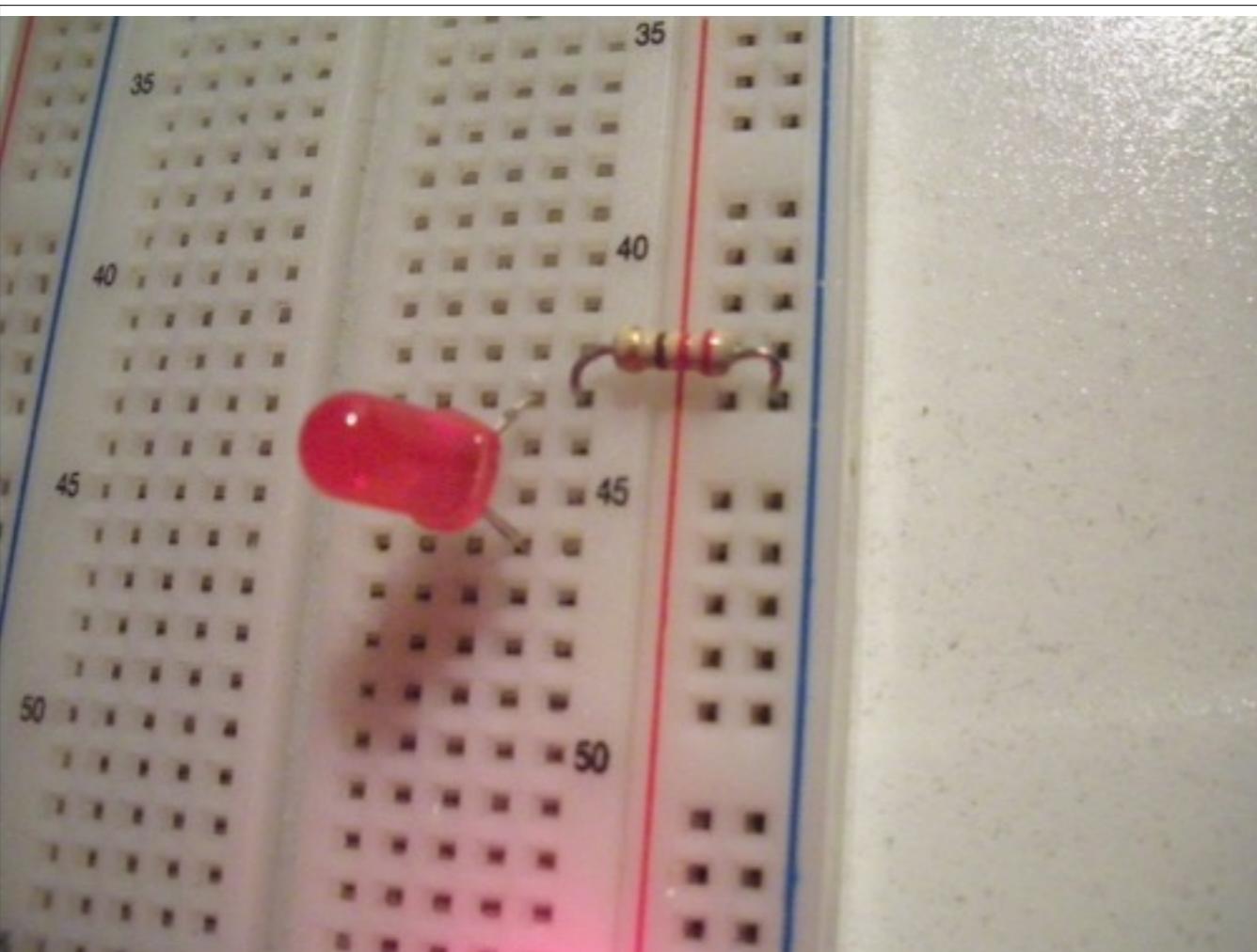




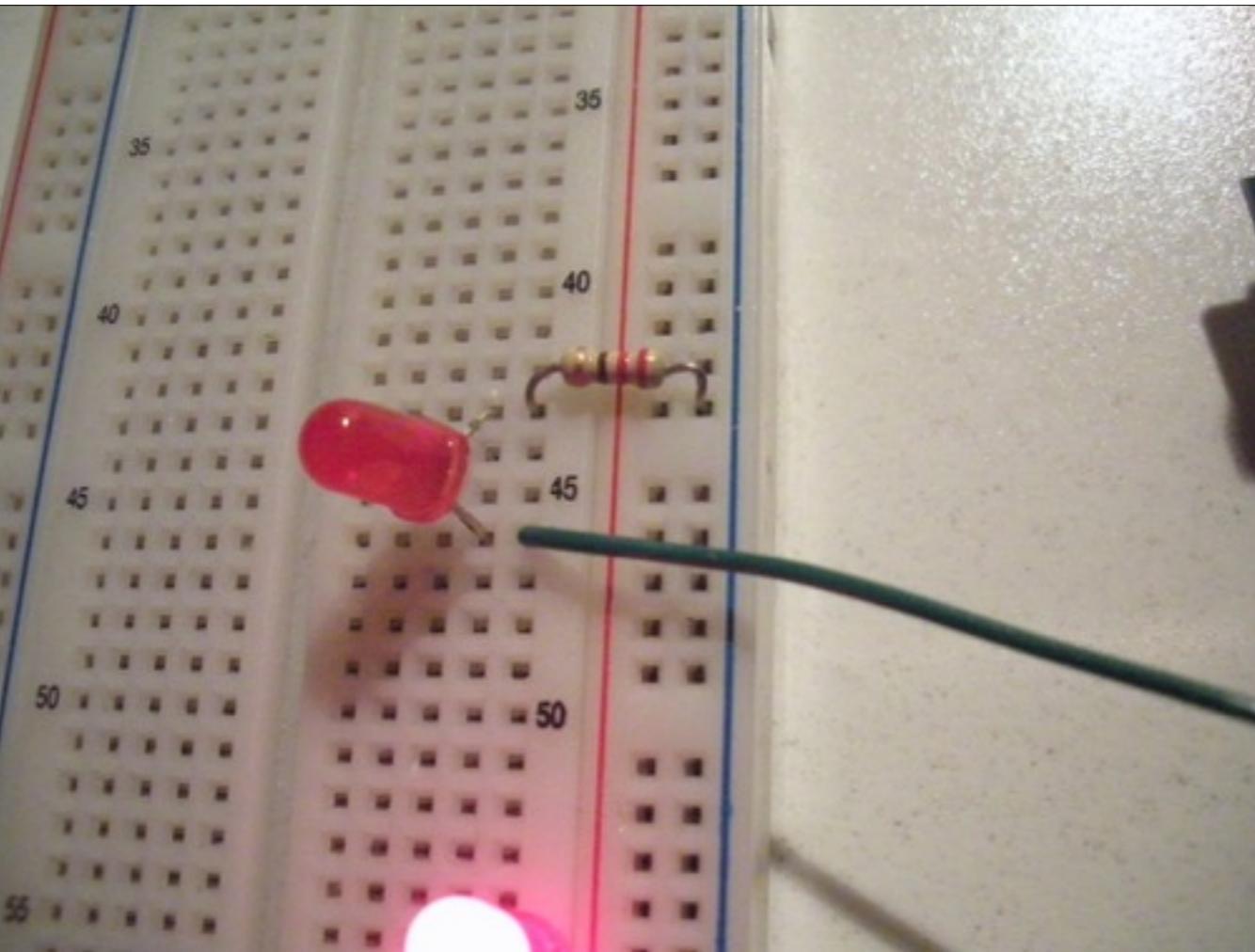




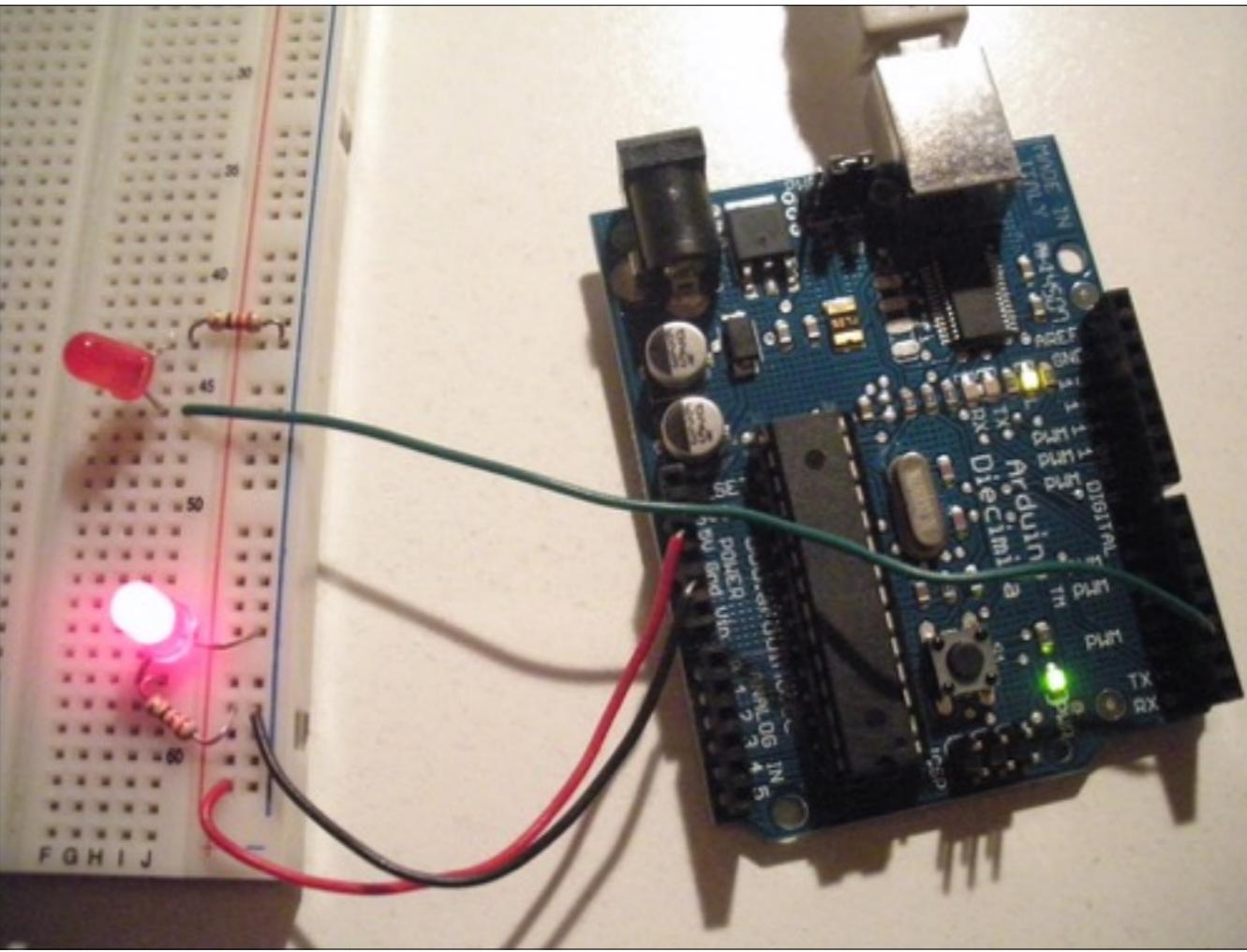
Wired for output



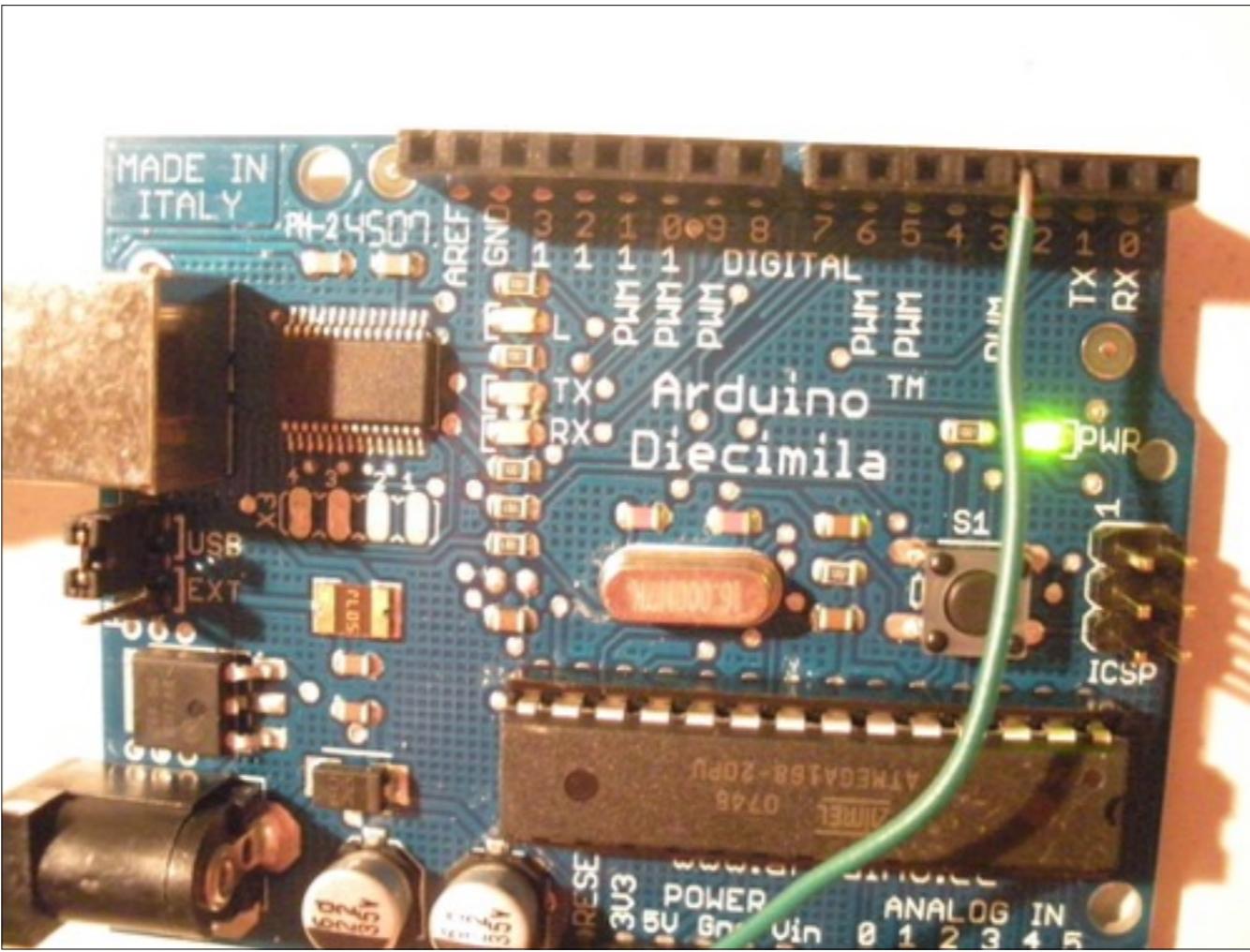
Straddle another led across a few rows of your breadboard. Doing so, will ensure that the two sides of the led aren't connected to each other. Place 1 leg of a resistor into the same row as your led. Plug the other leg of that resistor into the ground side of your breadboard



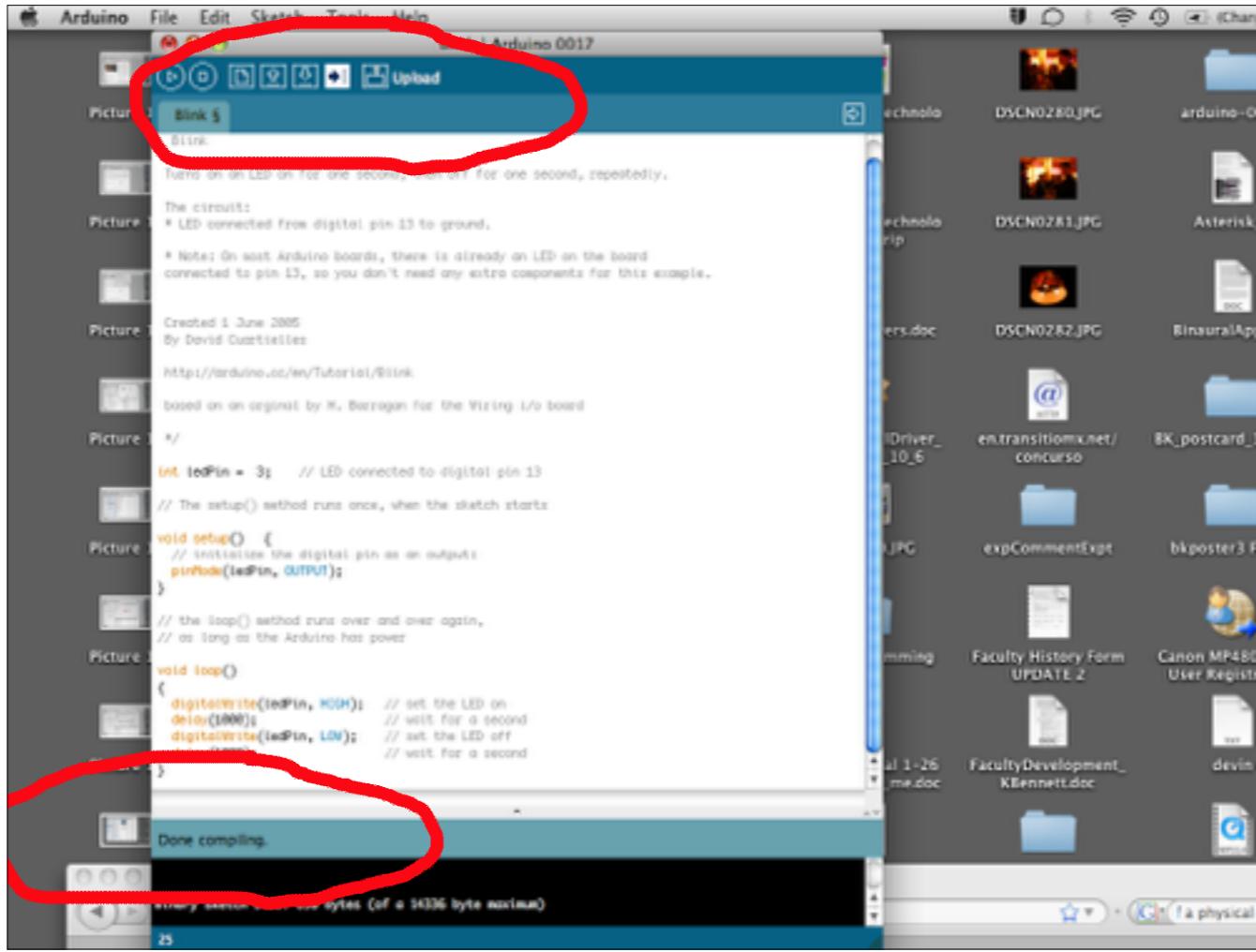
Plug a wire that is not red or black, into the side of the led that is unconnected.



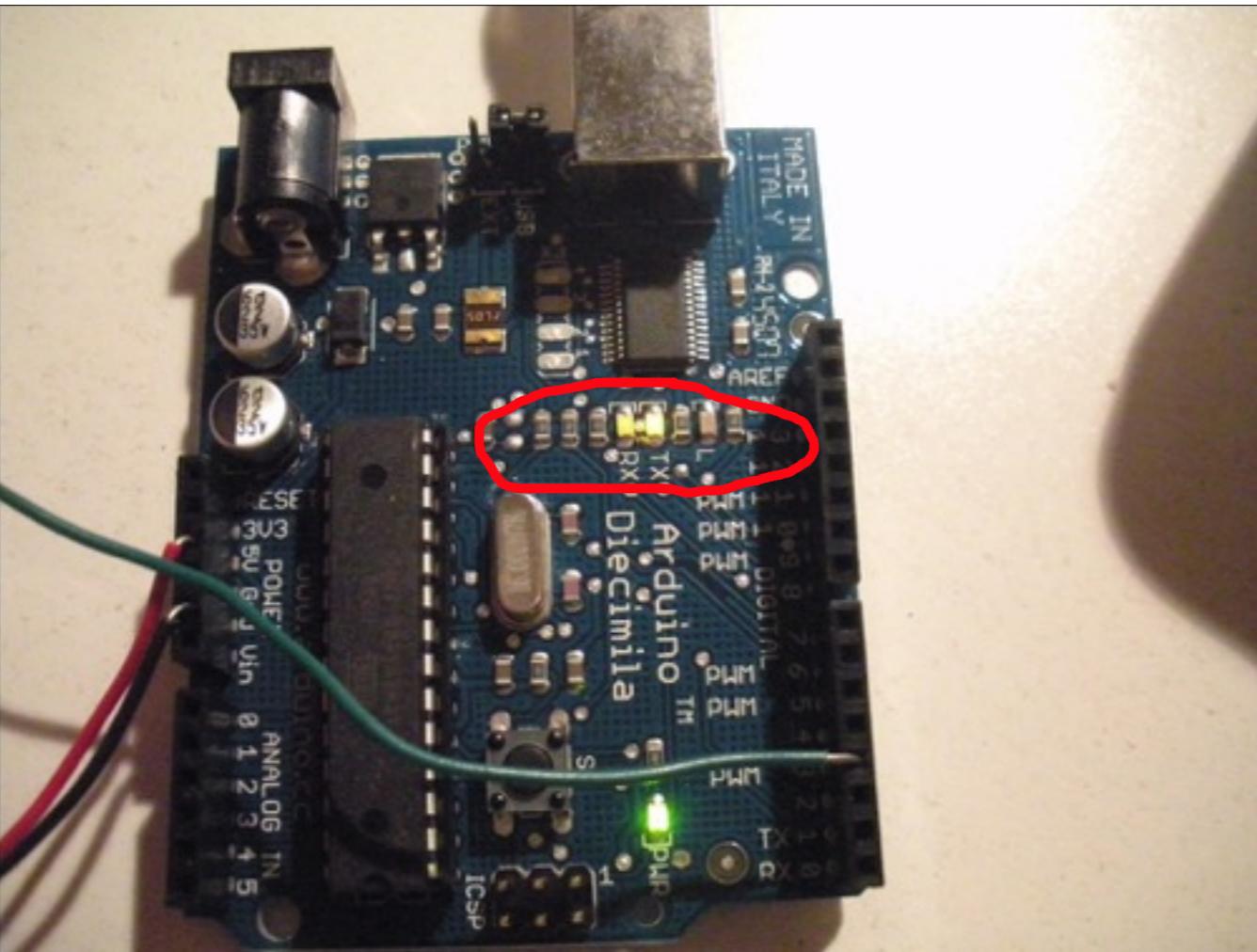
Plug the other side of that colored wire, into a DIGITAL pin on your Arduino board.
Do not choose digital pins 0 or 1. I chose digital pin 3.



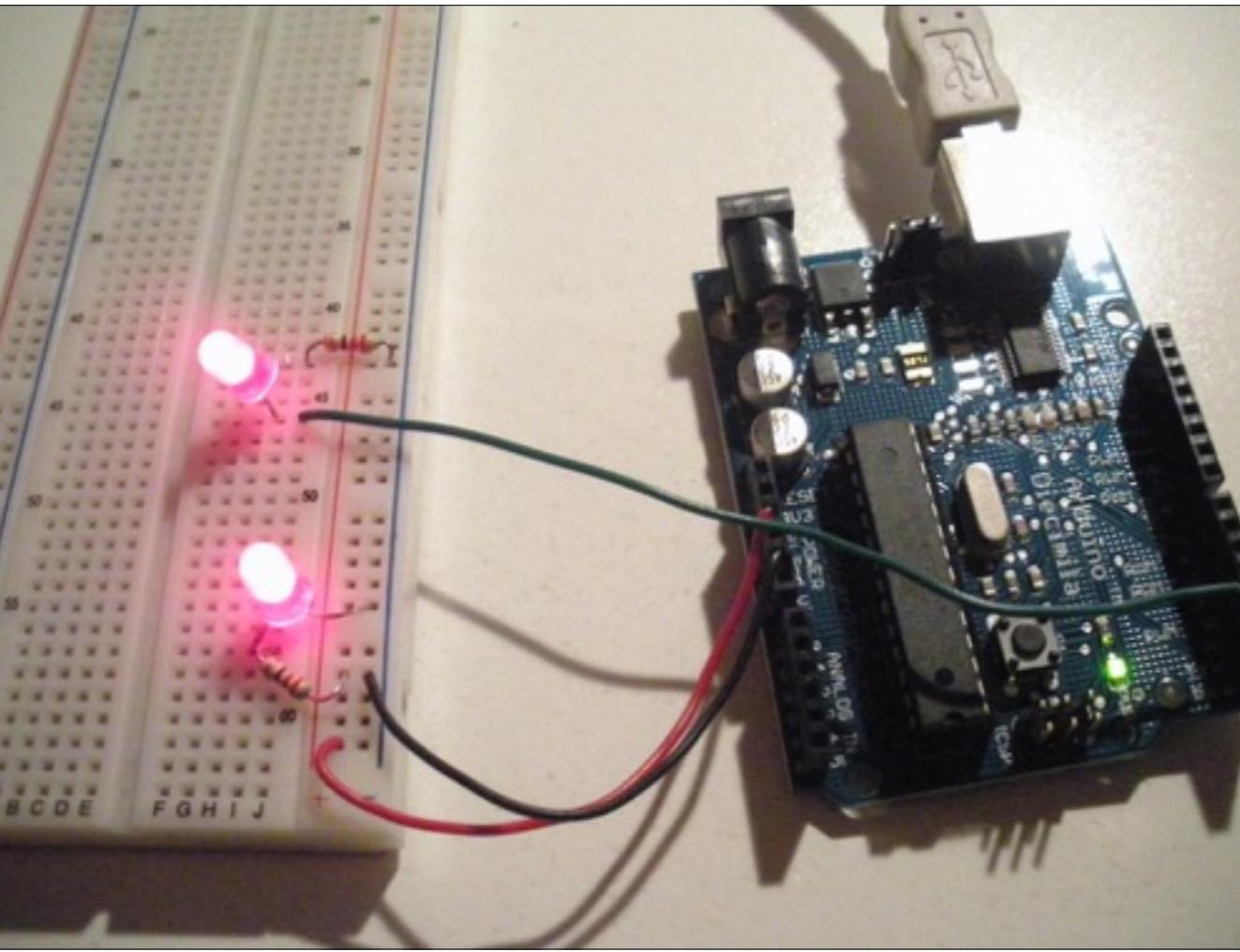
Plug the other side of that colored wire, into a DIGITAL pin on your Arduino board.
Do not choose digital pins 0 or 1. I chose digital pin 3.



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.



Now, your led, on your breadboard, should be flashing in the pattern that you programmed it.

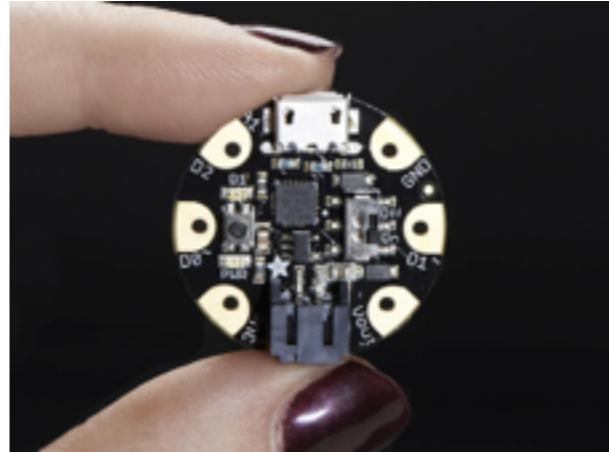
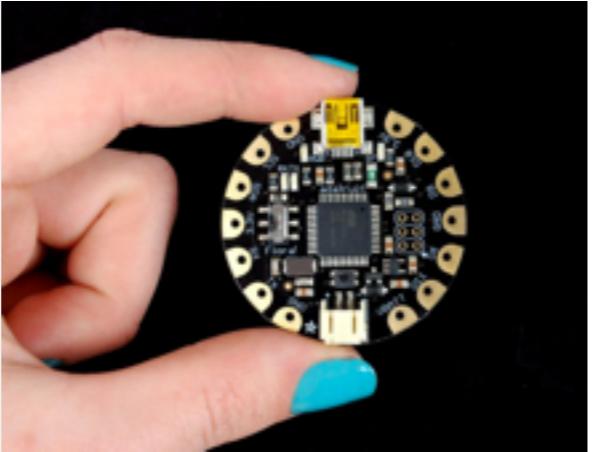
Review the code and try to have it flash in a sequence of 3 different patterns

Next workshop, we'll add sensors to the arduino board, giving it ears.
And will give the circuit it's own power supply, so it doesn't have to use your computer's power.

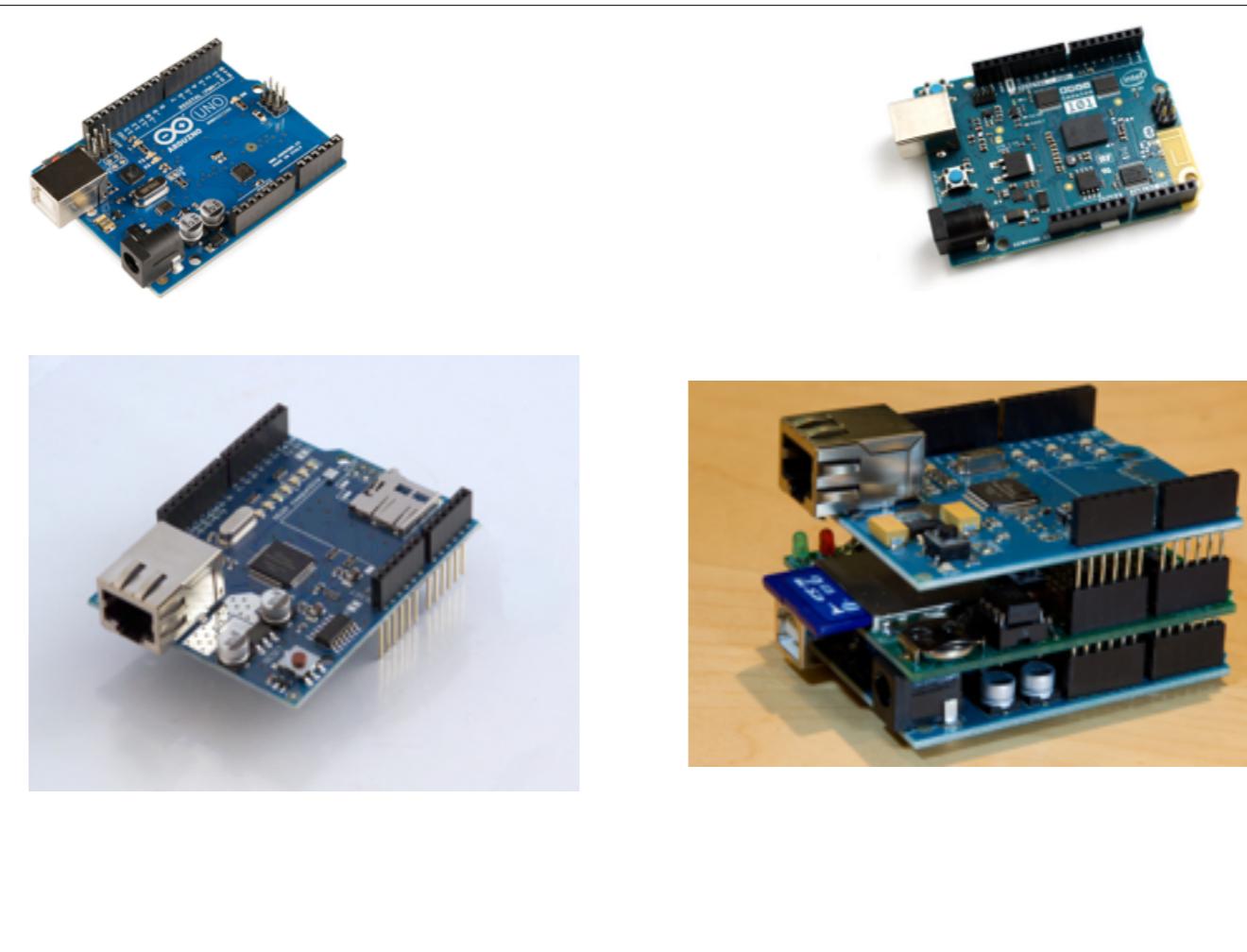
More Info

Arduino comes in many different flavors...

Such as the Gema



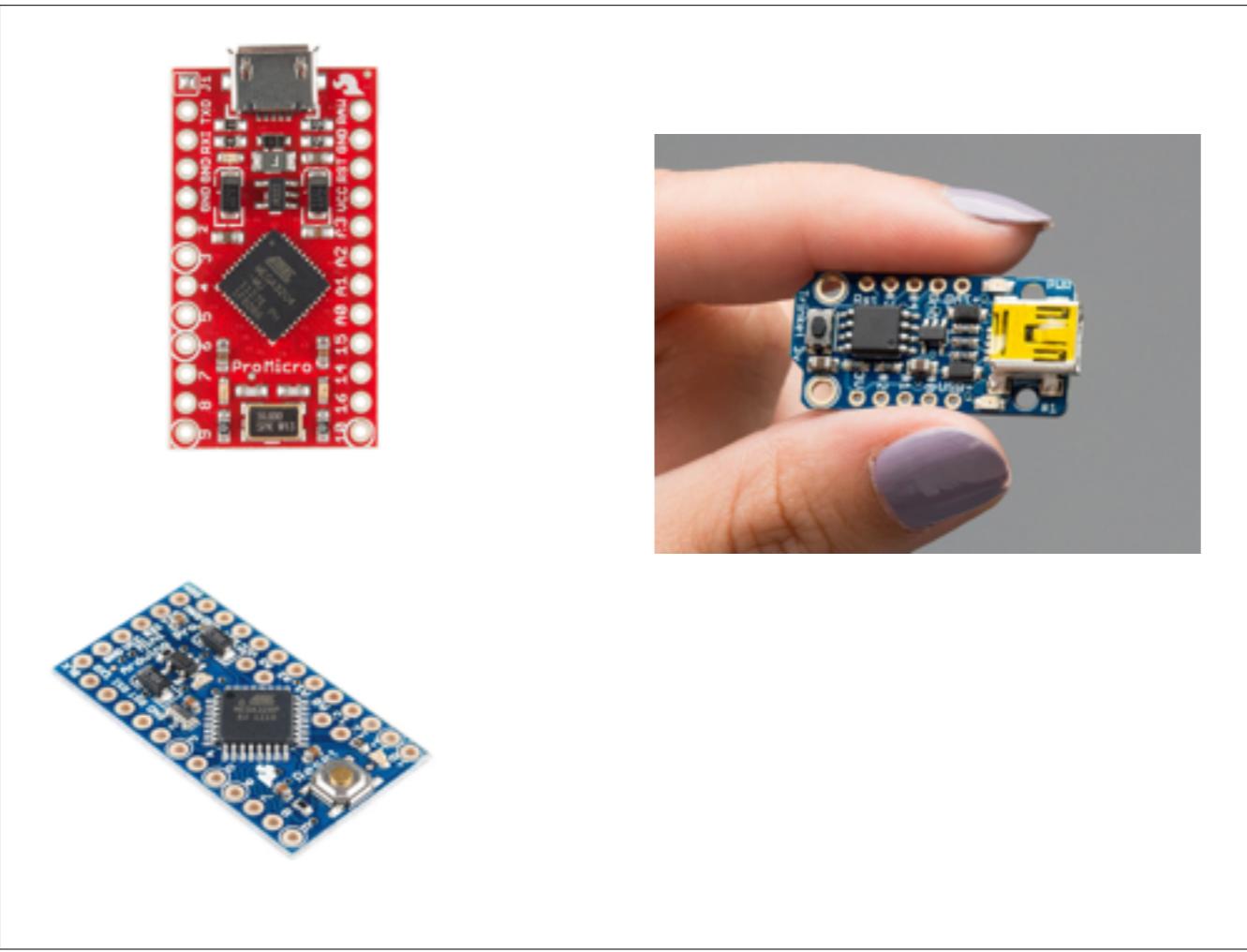
Gemma - The Attiny85 is a fun processor because despite being so small, it has 8K of flash, and 5 I/O pins, including analog inputs and PWM 'analog' outputs.
“Washable” & wearable



Shields

Extend functionality of the arduino

<http://ece.jntuhceh.ac.in/openhardware/sites/default/files/shield%20arduino.jpg>



Pro Micro

ATmega32U4 running at 5V/16MHz

ATMega 32U4 running at 3.3V/8MHz

4 x 10-bit ADC pins

12 x Digital I/Os (5 are PWM capable)

Rx and Tx Hardware Serial Connections

+ Trinket + ProTrinket

IO pins to play with; a total of 18, some of which can be analog, and only 32Kb of memory.

USB Device

5v and 3.3v versions,

ATmega328P onboard chip in QFN package

16MHz clock rate, 28K FLASH available

18 GPIO, 2 extra analog inputs, 28K of flash, and 2K of RAM.

Pro Mini + more

RESOURCES

Web

Craft:
<http://www.craftzine.com>



Fashioning Technology
<http://www.fashioningtech.com>



How To Get What You Want
<http://www.kobakant.at/DIY/>



Instructables
<http://www.instructables.com>



LilyPond
<http://lilypond.media.mit.edu>



Make:
<http://www.makezine.com>



Soft Circuit Saturdays
<http://softcircuitsaturdays.com/>



talk2myShirt
<http://www.talk2myshirt.com/>



Graphic by Emily Lovell