Day 02 - ICfL Make It at the Library

Basic electronics and Arduino

Nick Raymond

nraymond@makermedia.com

Make:

Overview

- Day 1
  - Electronics and Breadboarding
  - Lunch
  - Arduino Basic Projects
- Day 2
  - Arduino intermediate Projects
  - Lunch
  - Wearable’s and E-textiles

Workshop Files

- Please copy files form one of the flash drives, they will have Fritizing files and Arduino sketches.
- Also, additional software and libraries to install for Day 03

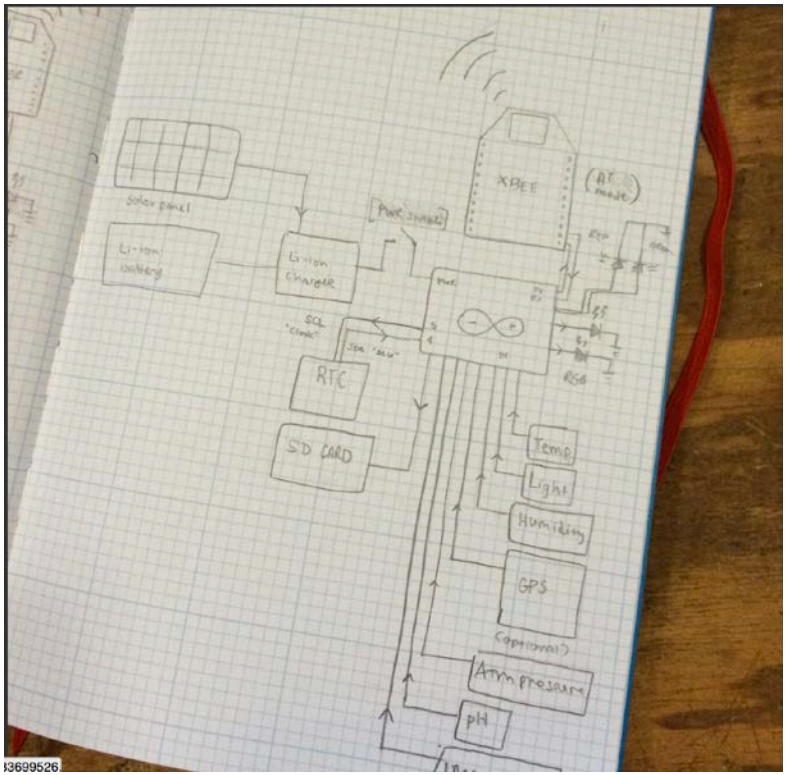
Name	Date Modified
Presentations	Today, 6:49 AM
Project 01 - Breadboard and LED	Yesterday, 11:54 PM
Project 02 - 555 timer touch switch	Today, 6:49 AM
Project 03 - 555 timer noise maker	Today, 6:50 AM
Project 04 - Arduino Blink LED	Today, 6:50 AM
Project 05_RGB LED	Today, 6:52 AM
Project 06_FSR	Today, 6:52 AM
Project 07_Inputs and Output	Today, 6:53 AM
Project 08_Makey Makey	Today, 6:54 AM

Workshop Format

- Workshop presenters
  - Nick Raymond
  - Adam Day
- Hands on workshop
  - Build the circuits
  - There will be problems
  - Ask us for help, thanks why we are here!

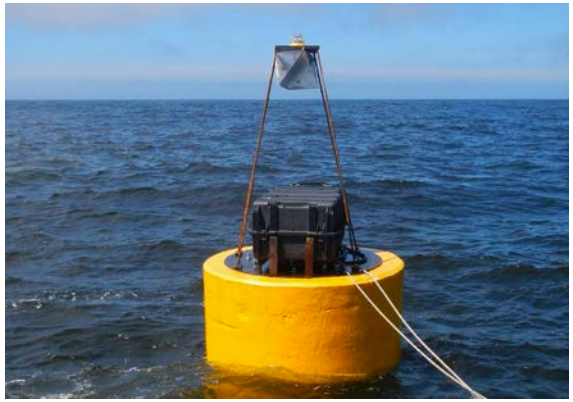
About myself…

- Nick Raymond – Maker Media – ndraymond@makermedia.com
- BS Mechanical Engineering (UD Davis)
- MS Student – emphasis in manufacturing and mechatronics
- Hobbies
  - Beer brewing
  - Surfing
  - 3D Printing
  - CNC machines
  - Wood working
  - Electronics/Arduino

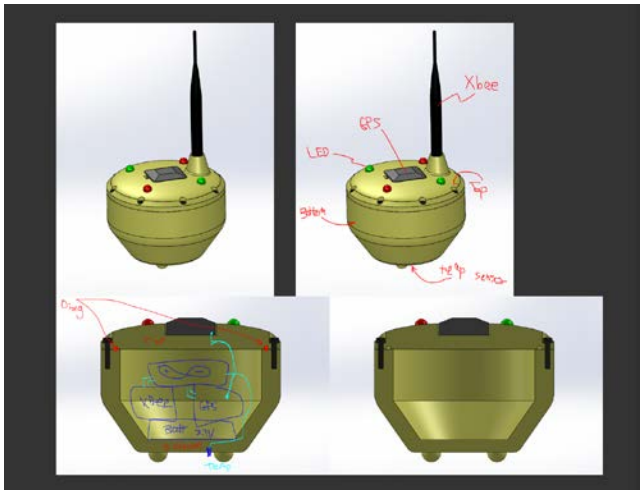


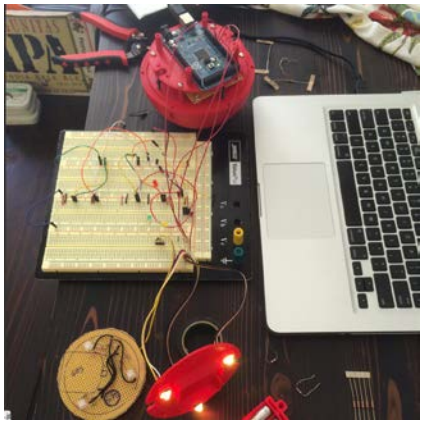
About myself…

- Ocean powered wave energy converter – 2012/2013



- Open source Ocean Wave Buoy Project - ongoing

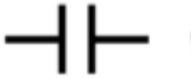




Electrical Component Symbols

- Resistors
- Capacitors
- Inductors
- Diodes
- Transistor
- Voltage source

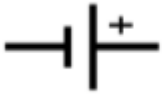

Resistor


Capacitor


Inductor


Diode


Transistor



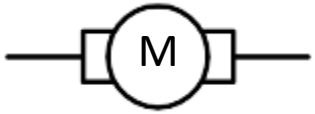
DC voltage source

Electrical components

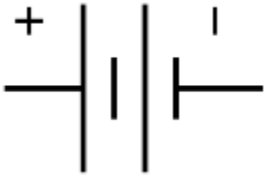
- Potentiometers

- Light Emitting Diode (LED)

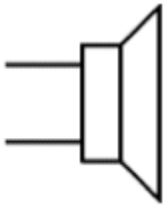
- Motor



- Battery



- Speaker



Voltage – Current - Resistance

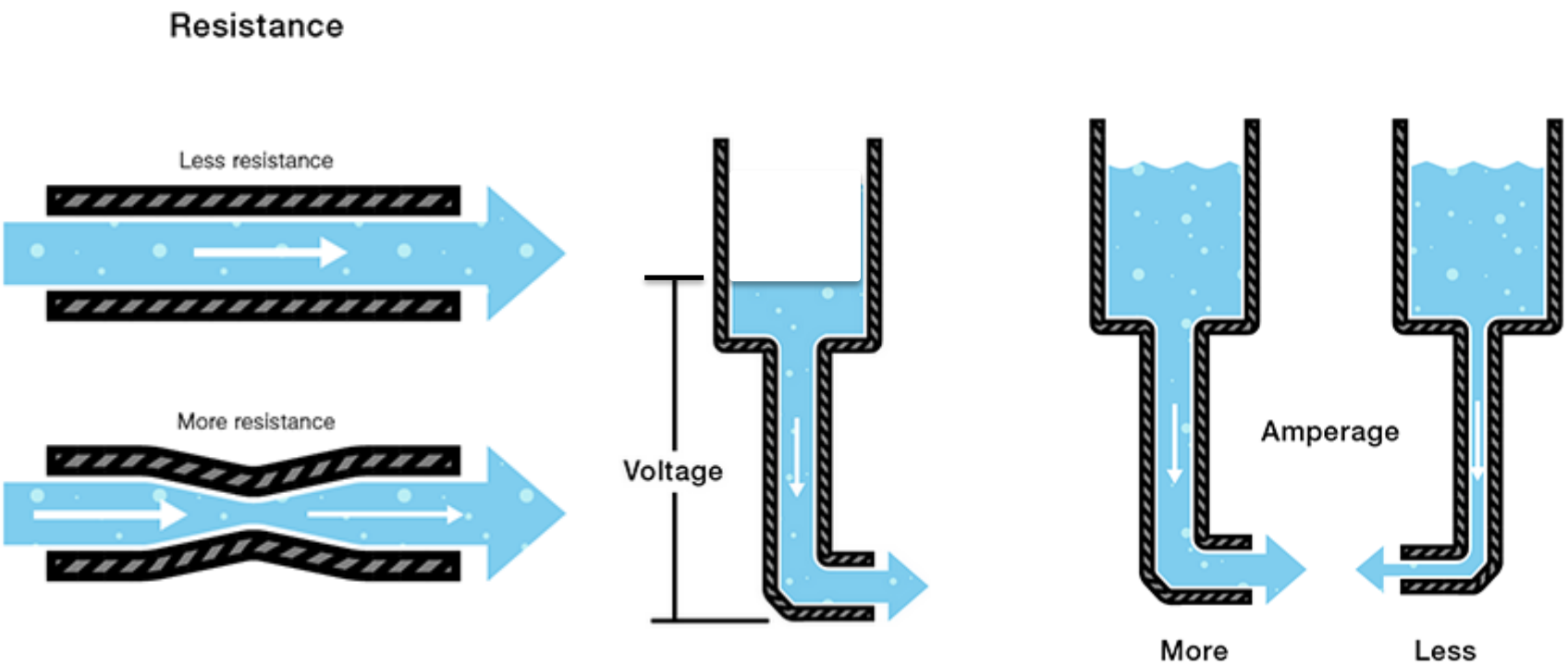


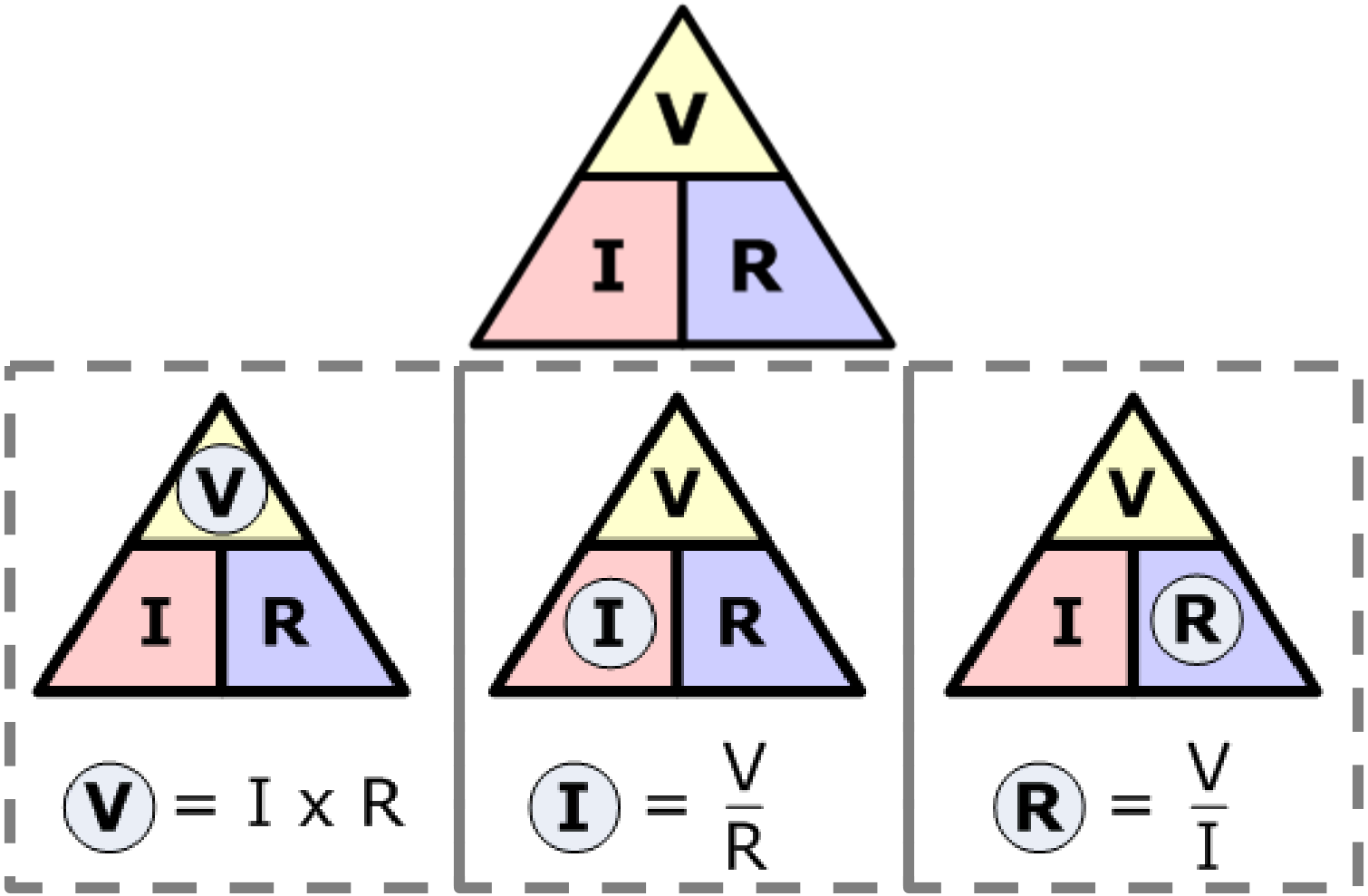
Image: Sparkfun.com

Current vs. Voltage

	Current	Voltage
Definition	Current is the rate at which electric charge flows past a point in a circuit. In other words, current is the rate of flow of electric charge.	Voltage, also called electromotive force, is the potential difference in charge between two points in an electrical field. In other words, voltage is the "energy per unit charge".
Symbol	I	V
Unit	A or amps or amperage	V or volts or voltage
SI Unit	1 ampere =1 coulomb/second.	1 volt = 1 joule/coulomb. (V=W/C)
Measuring Instrument	Ammeter	Voltmeter
Relationship	Current is the effect (voltage being the cause). Current cannot flow without Voltage.	Voltage is the cause and current is its effect. Voltage can exist without current.

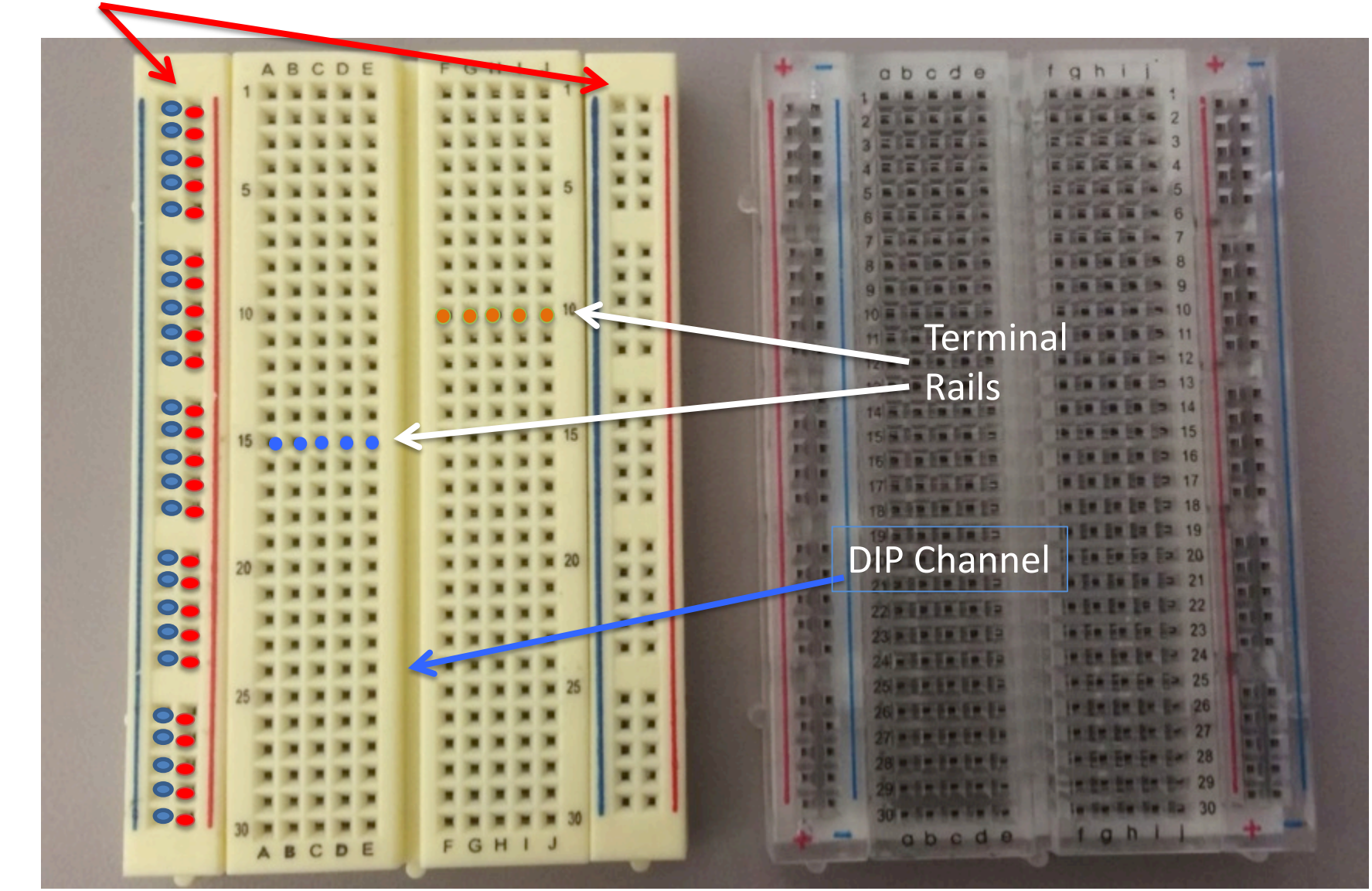
Source: http://www.diffen.com/difference/Current_vs_Voltage

Ohm's Law



Source: http://www.electronics-tutorials.ws/dccircuits/dcp_2.html

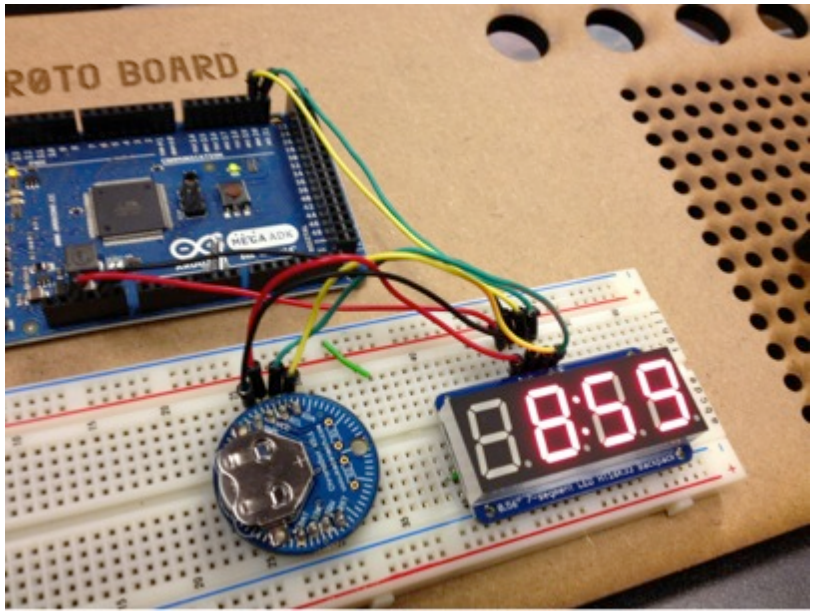
Breadboard Layout

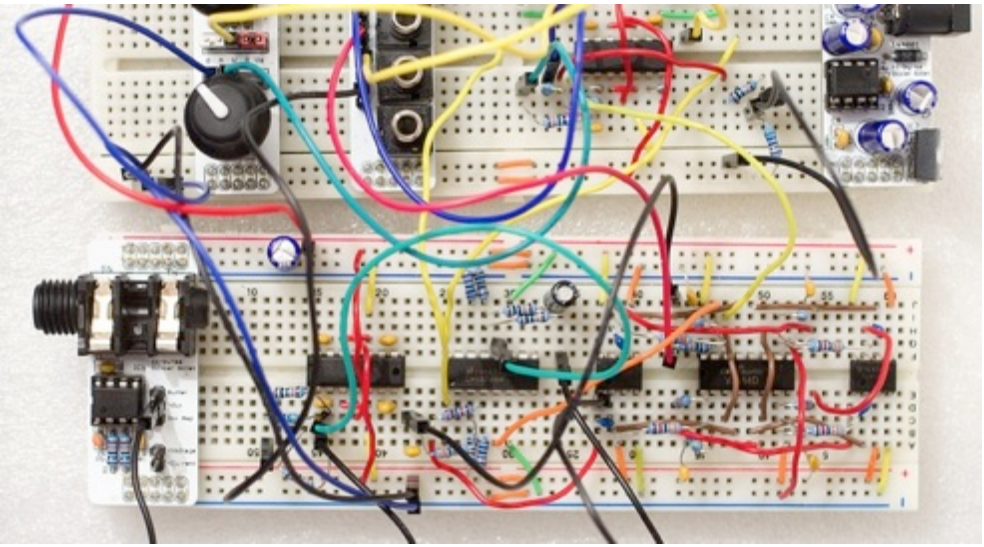


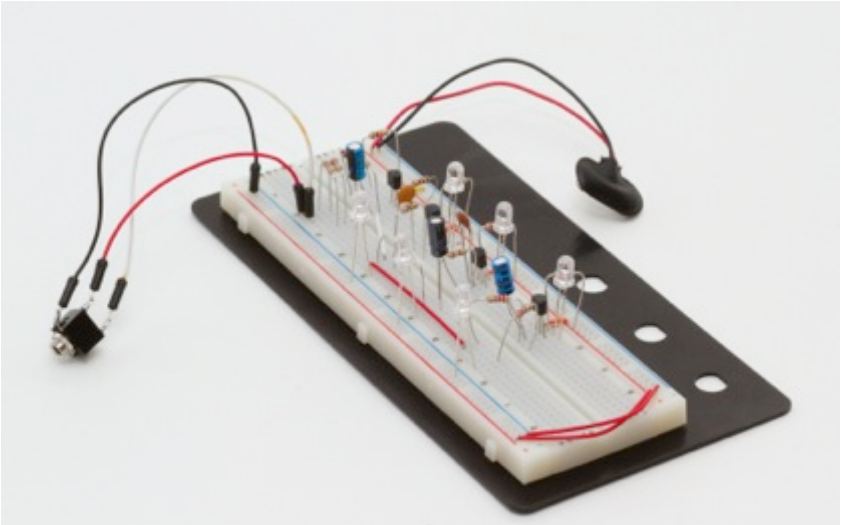
Breadboard “Insides”

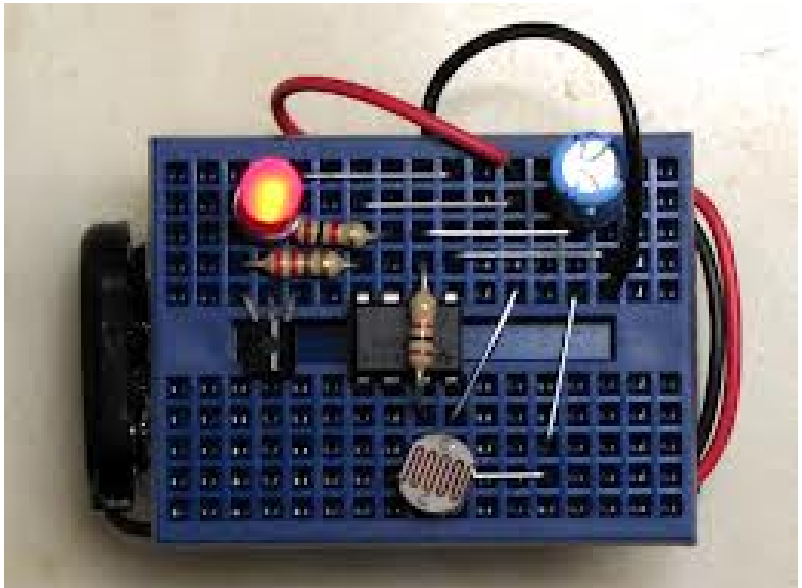
Images – sparkfun.com

Breadboards

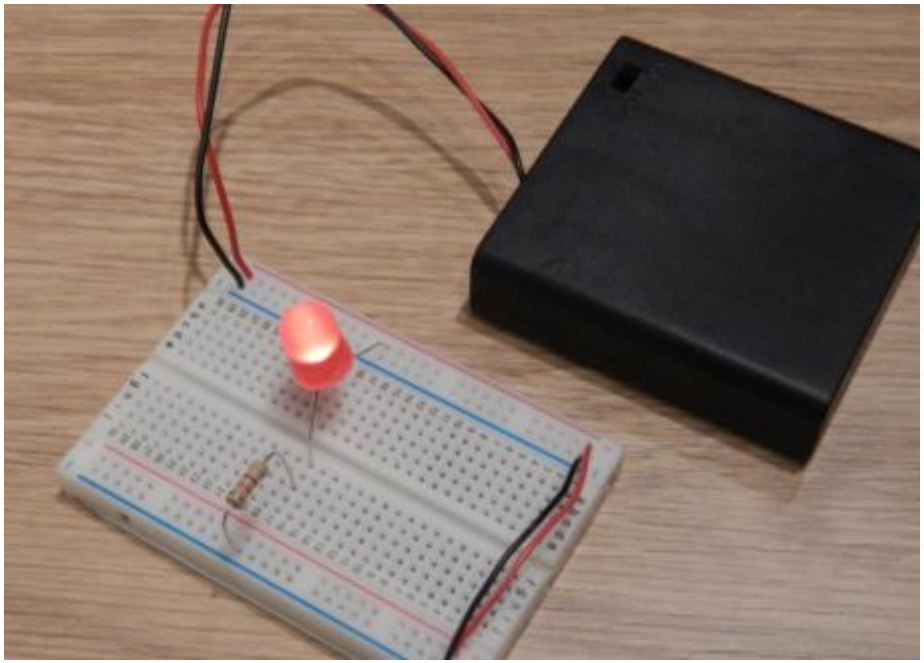


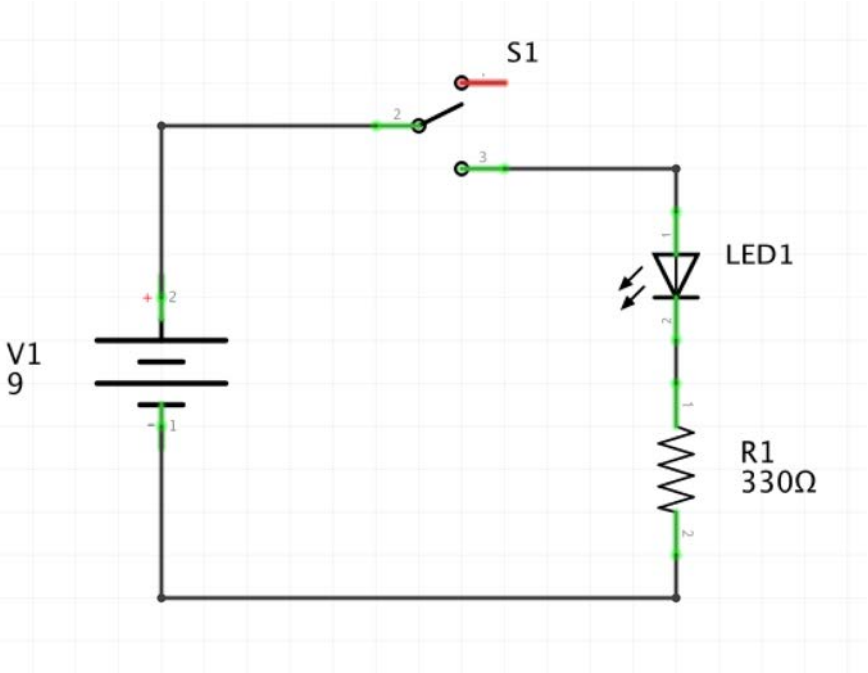




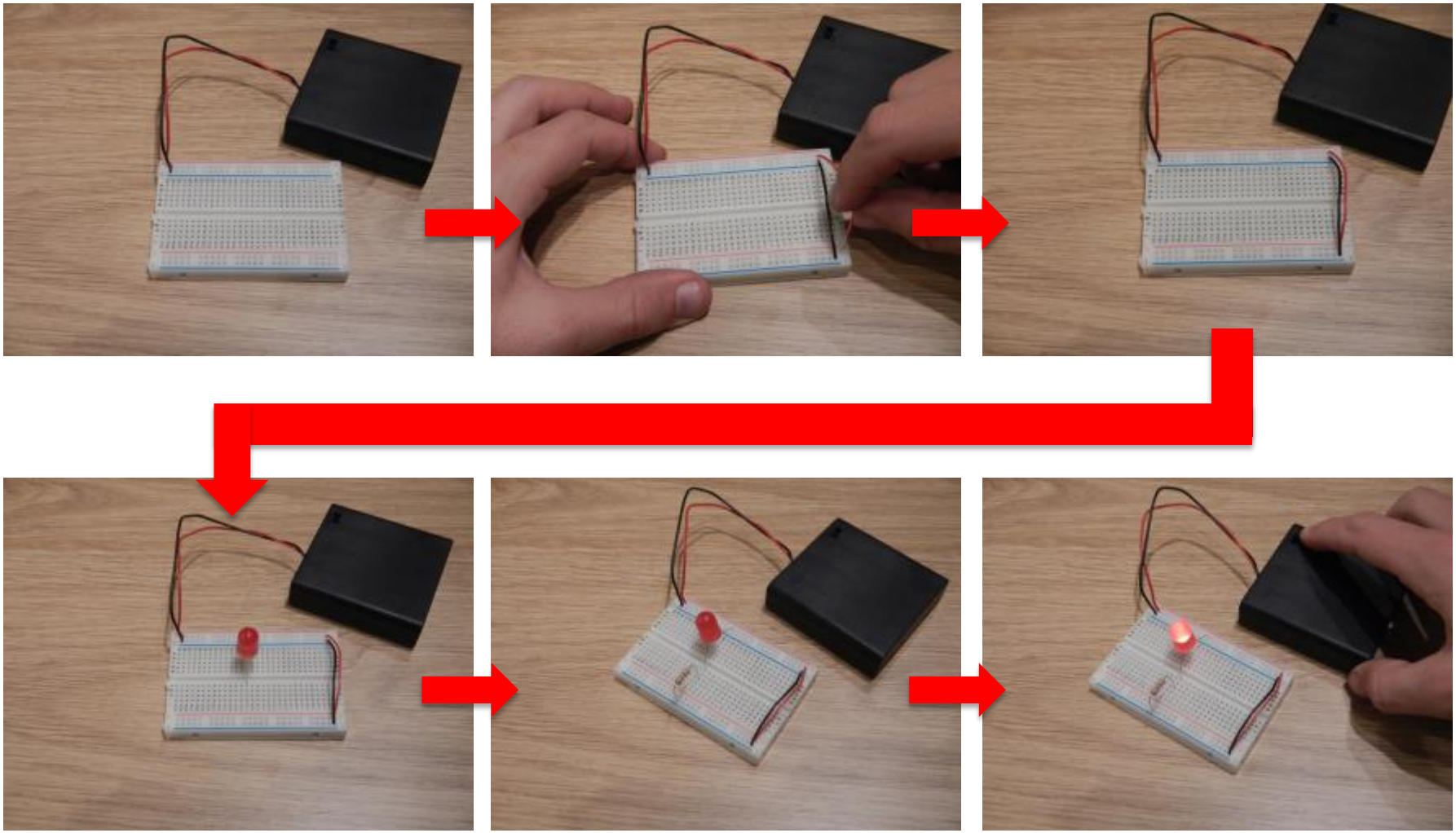


Basic LED + Switch circuit

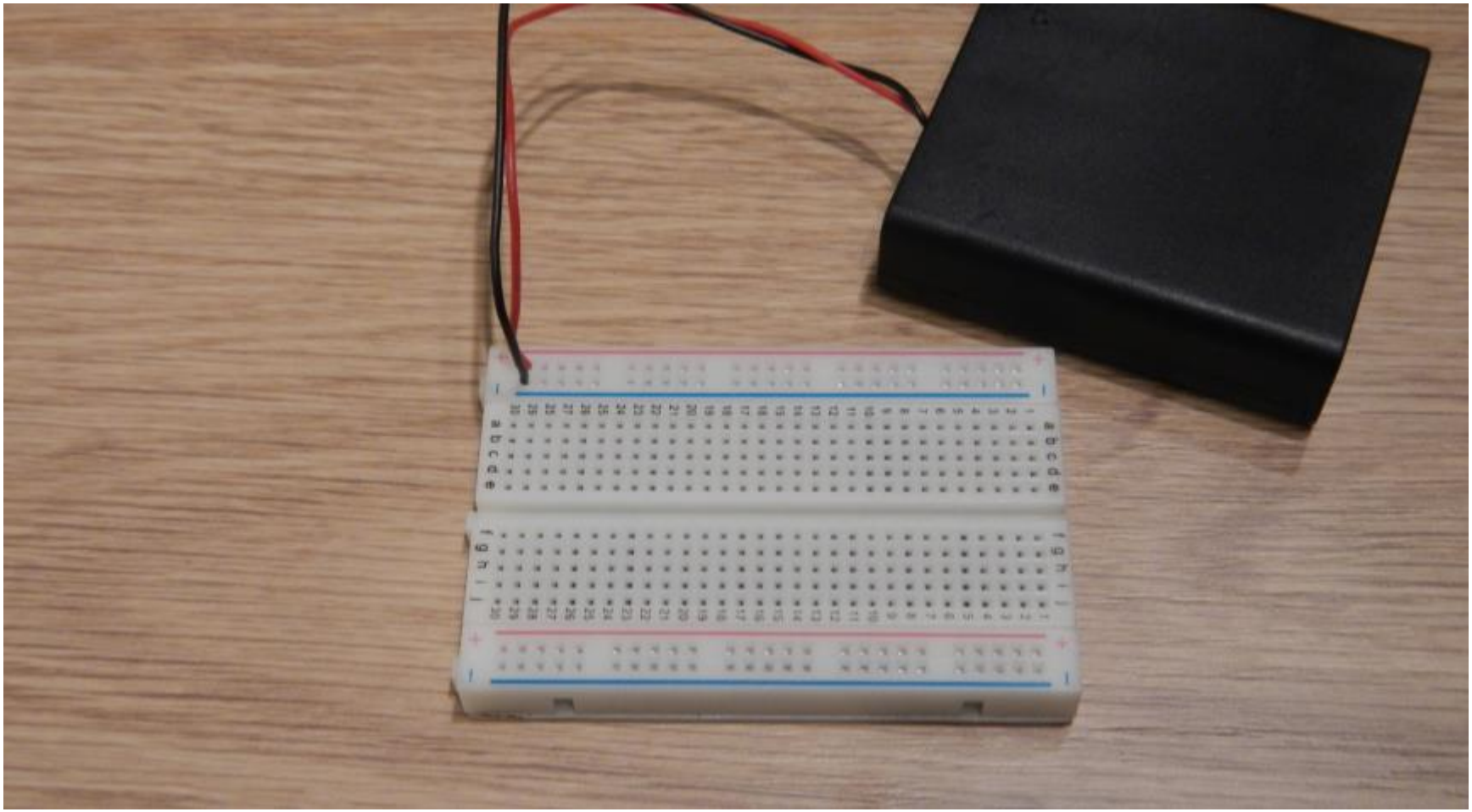




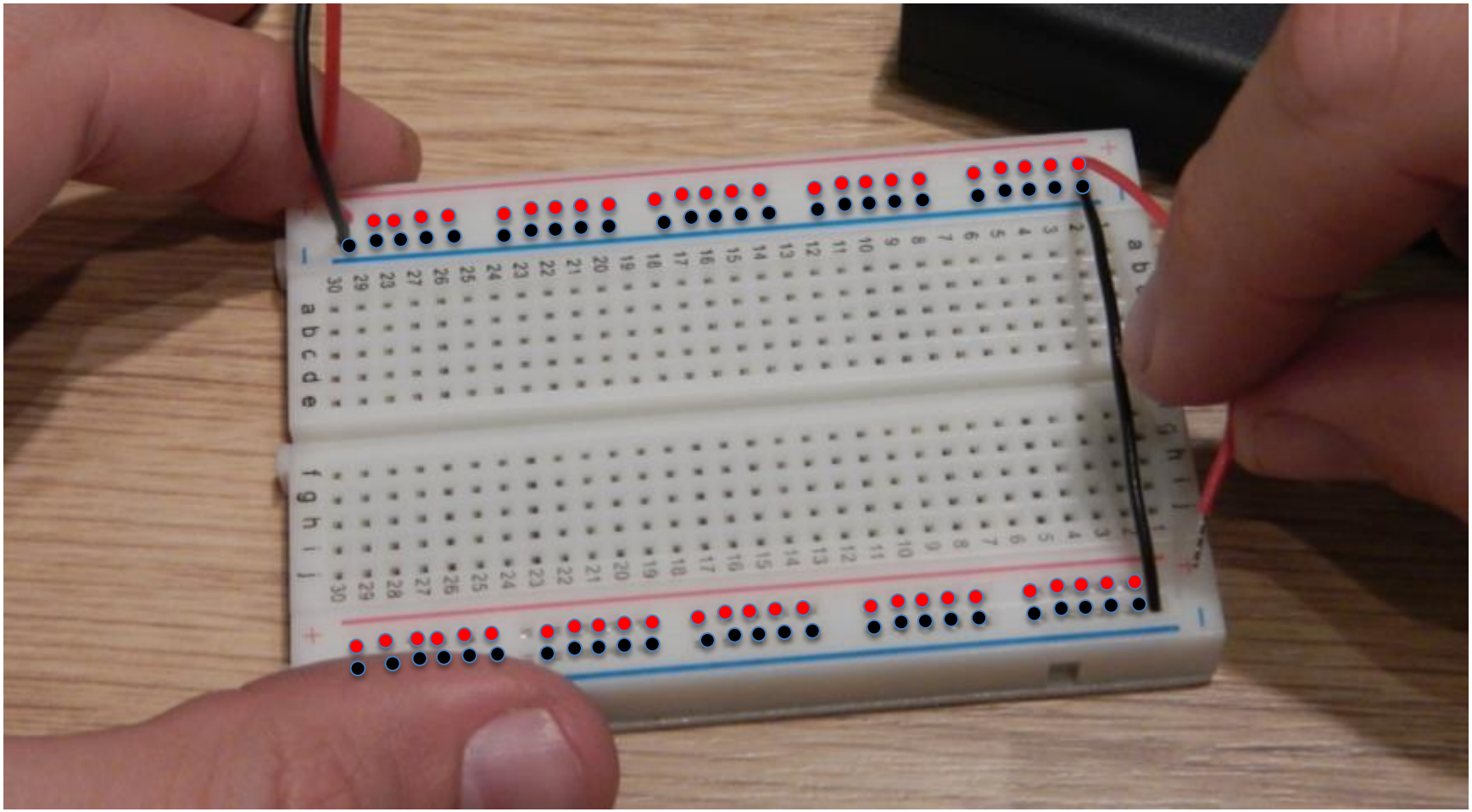
Build the Circuit



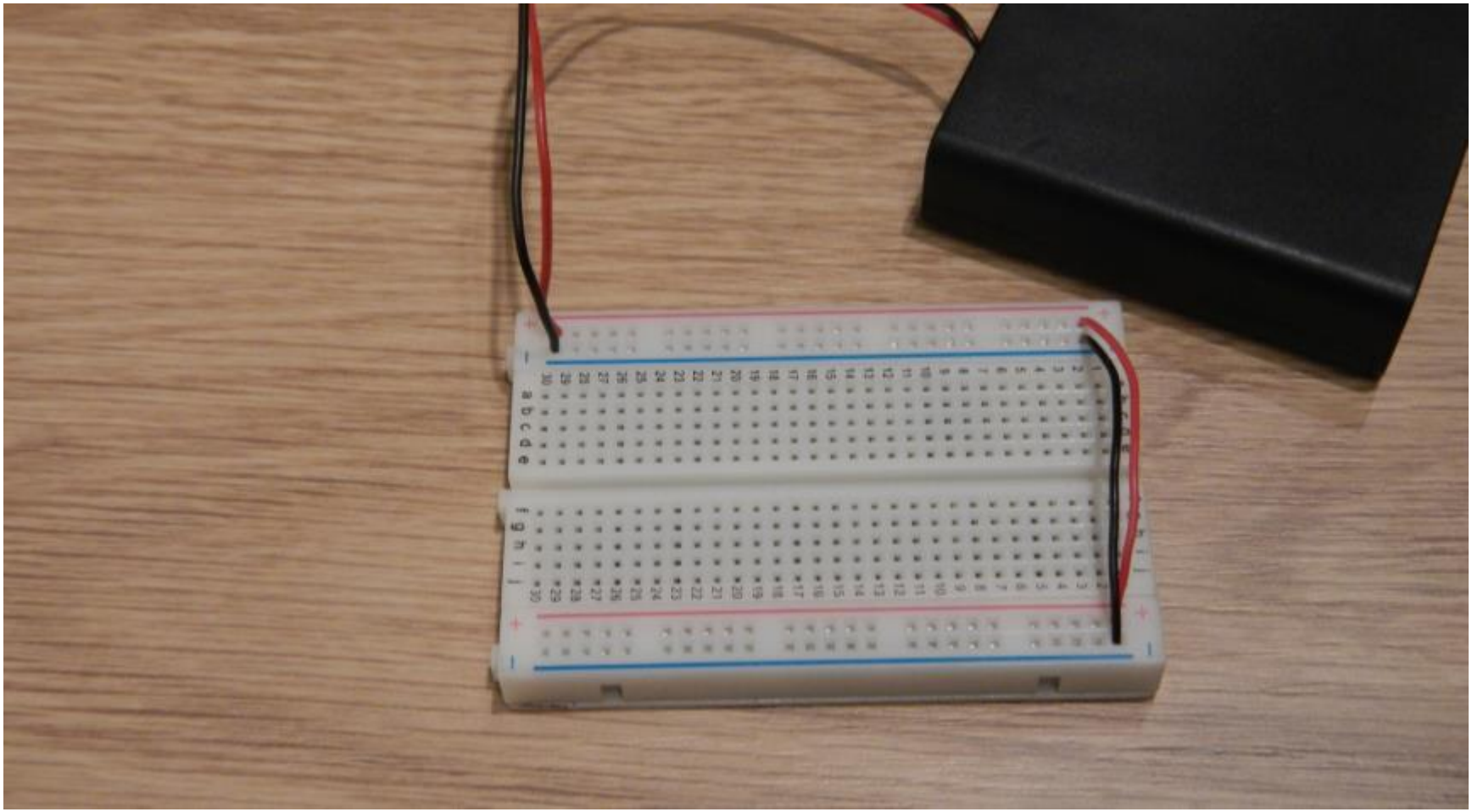
Step 1



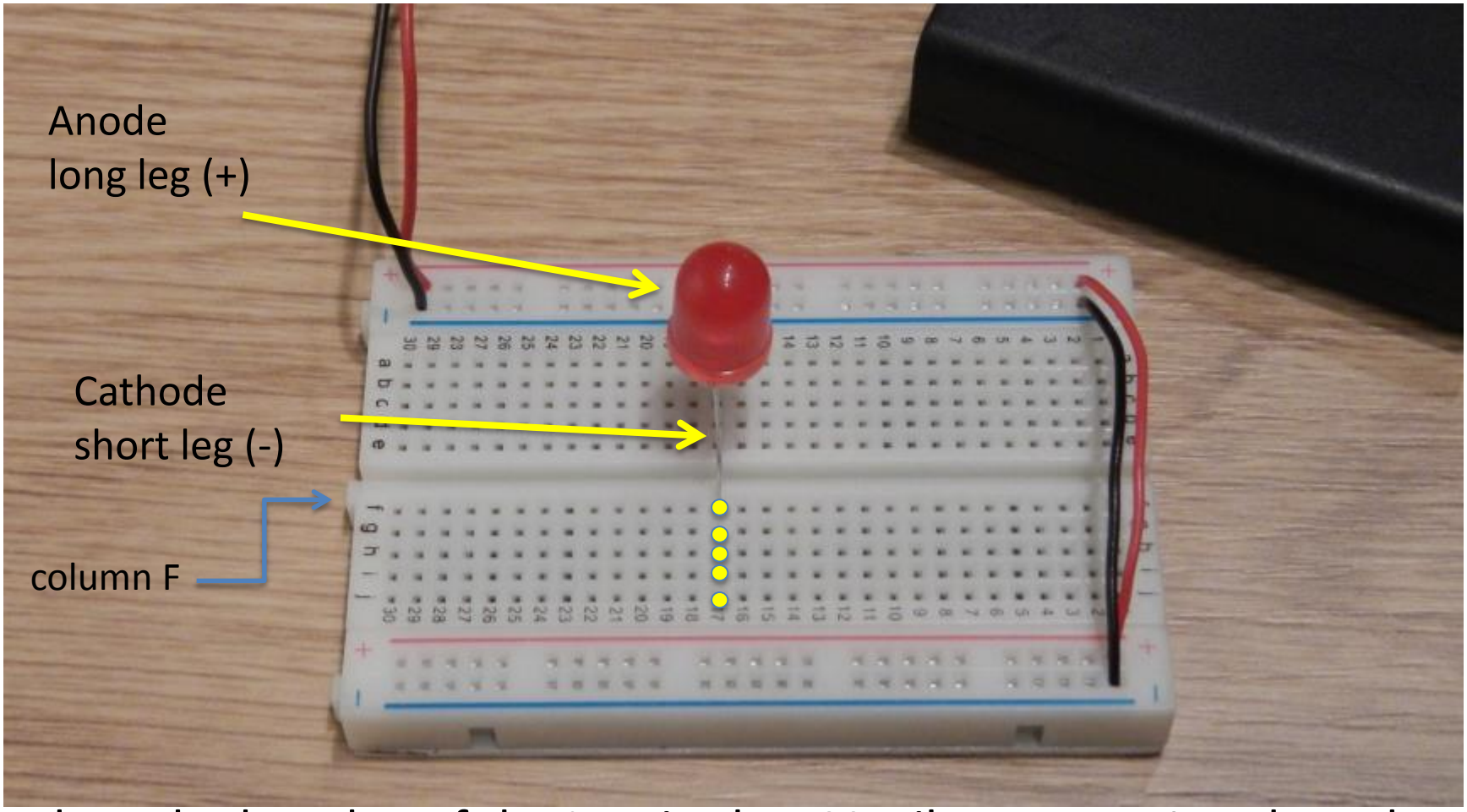
Step 2



Step 3

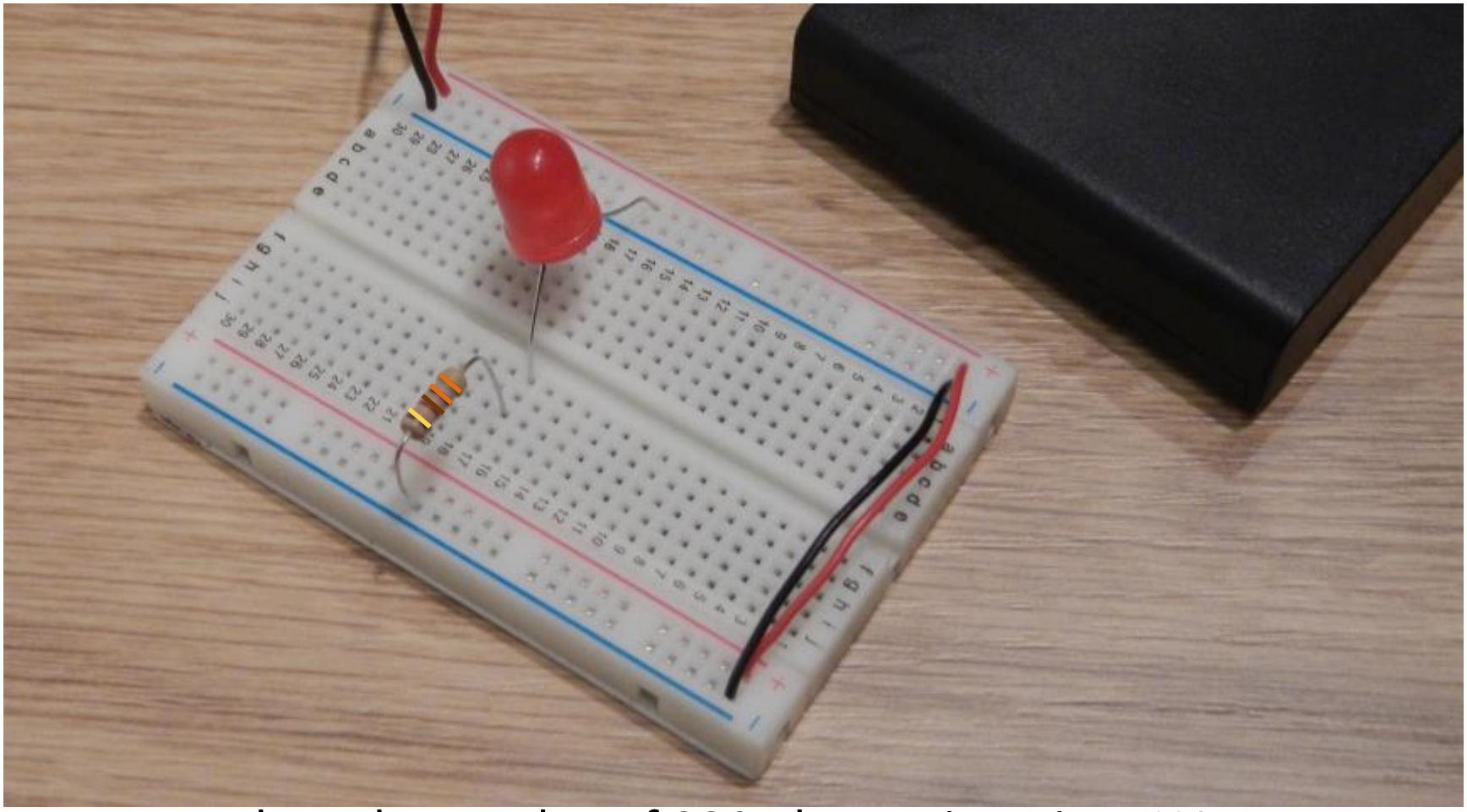


Step 4



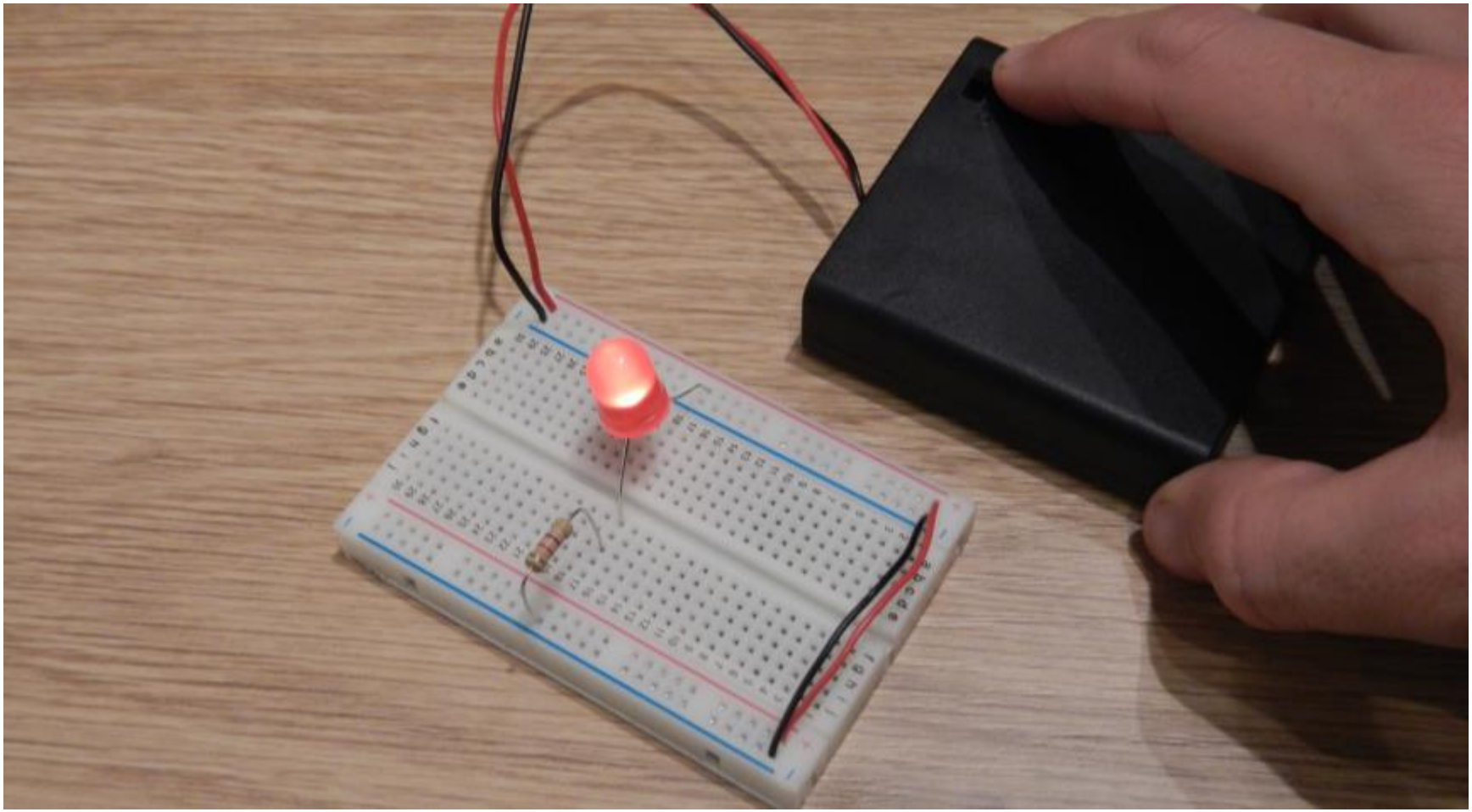
Place the long leg of the LED in the +V rail near row 17, place the short leg in F17.

Step 5



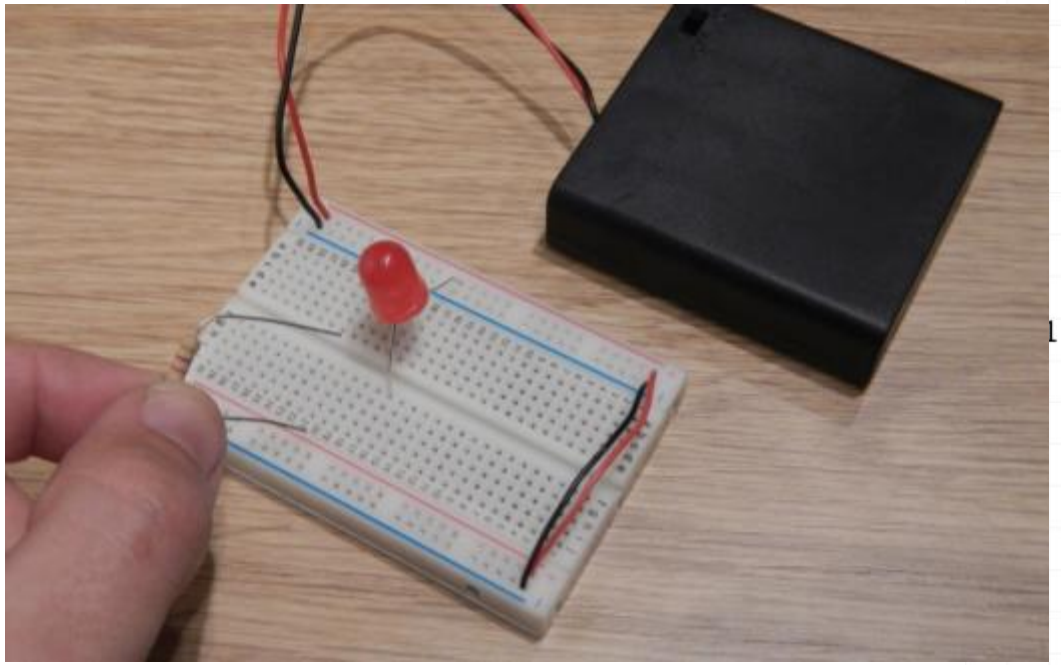
Place the one leg of 330 ohm resistor into H17
(orange, orange, brown, gold)

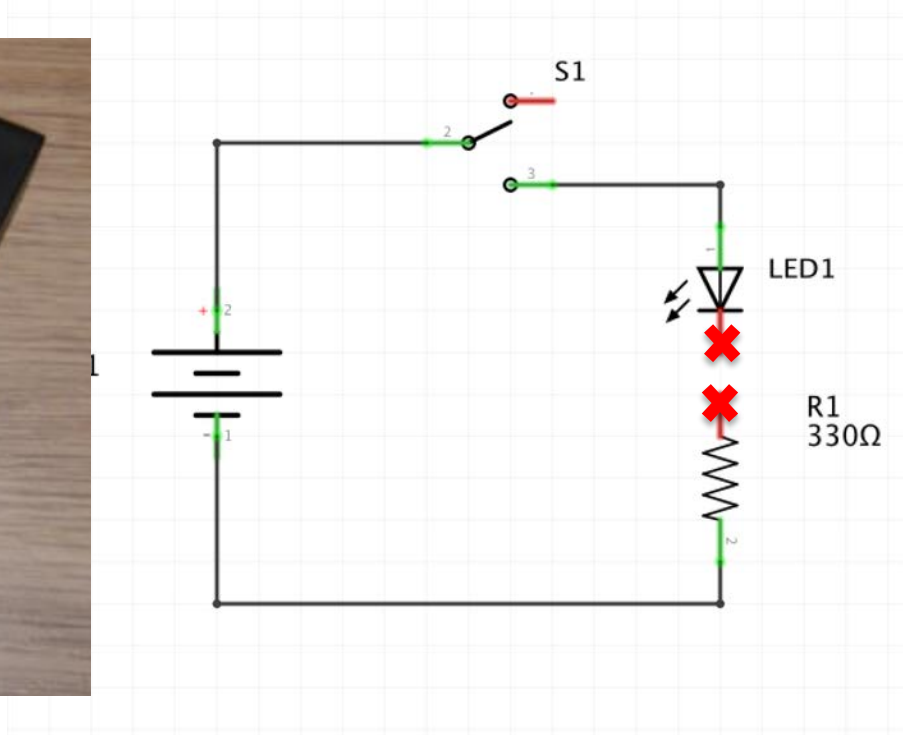
Step 6



Turn on the switch!

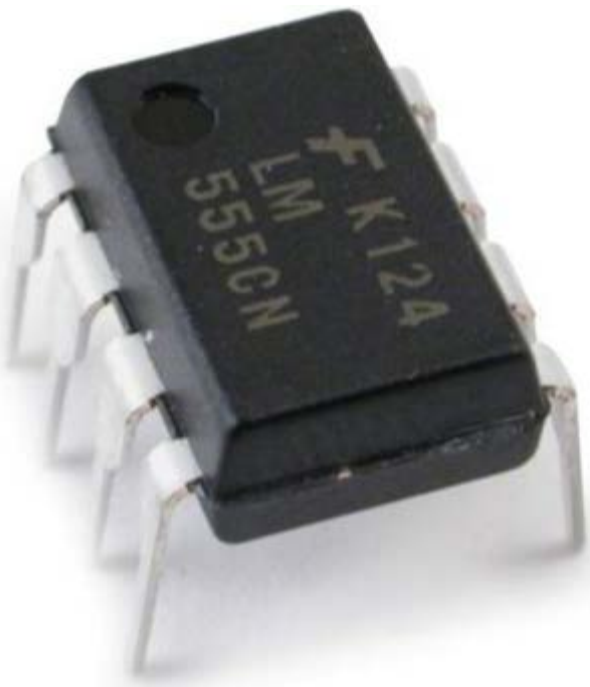
Remove the resistor…

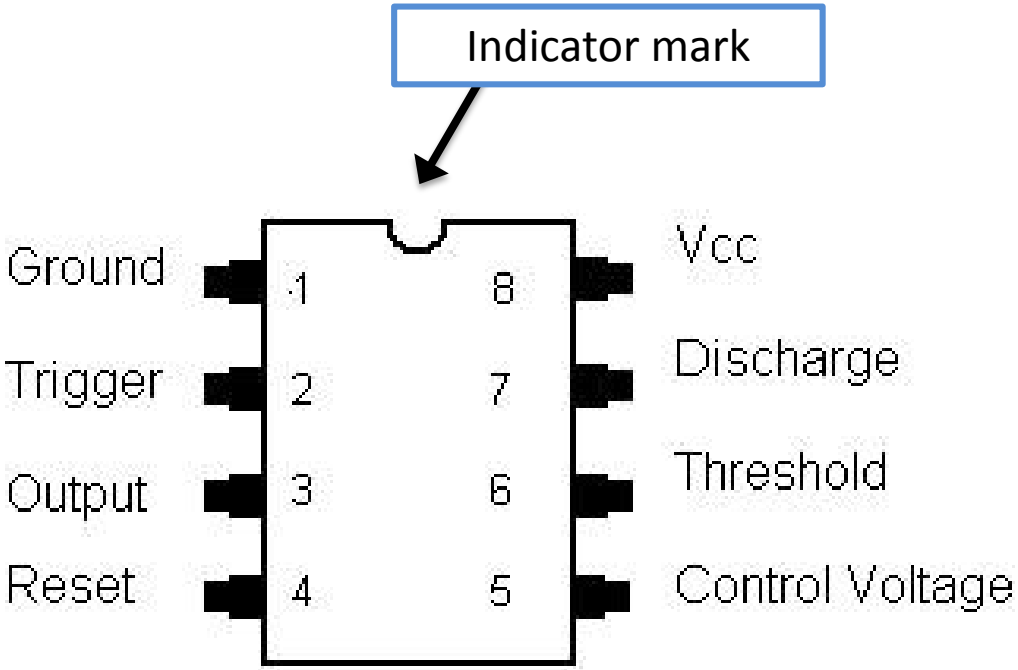




Creates a break in the circuit, electrons cannot flow and LED turns off.

555 Timer IC

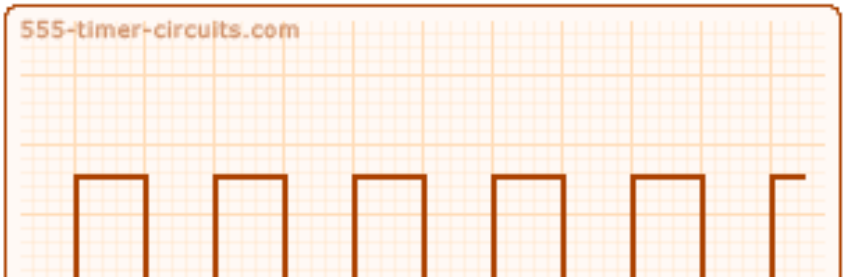




"Integrated Circuit" = IC and "Dual Inline Package" = DIP

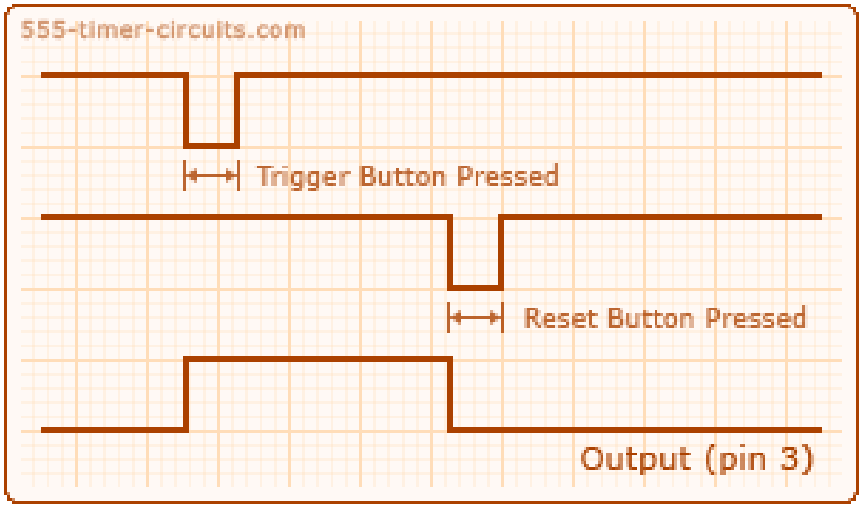
555 Timer IC

Astable – no stable state
- Switches between high and low, operates as an oscillator
- Used for motor control, flashing lamps and LED, or as a clock



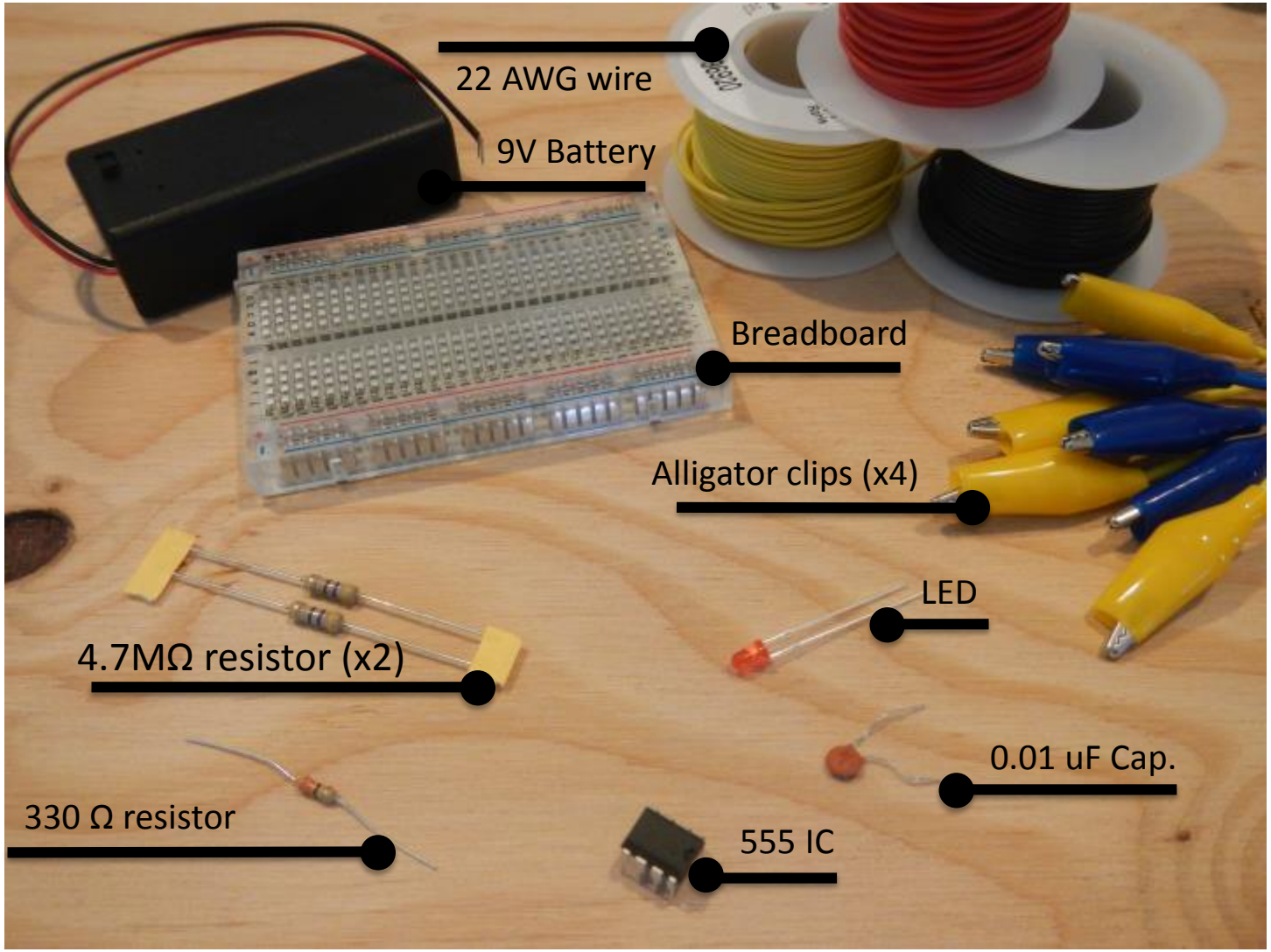
Monostable – one stable state
- One shot pulse of fixed length in response to input signal
-Ideal for "push to start" projects that will move then turn off after certain time.

Bistable – two stable states, high and low
-taking trigger input high makes output low,
- taking trigger input low makes output high
switches between two stable states based on the state of the inputs, great for ON/OFF switch

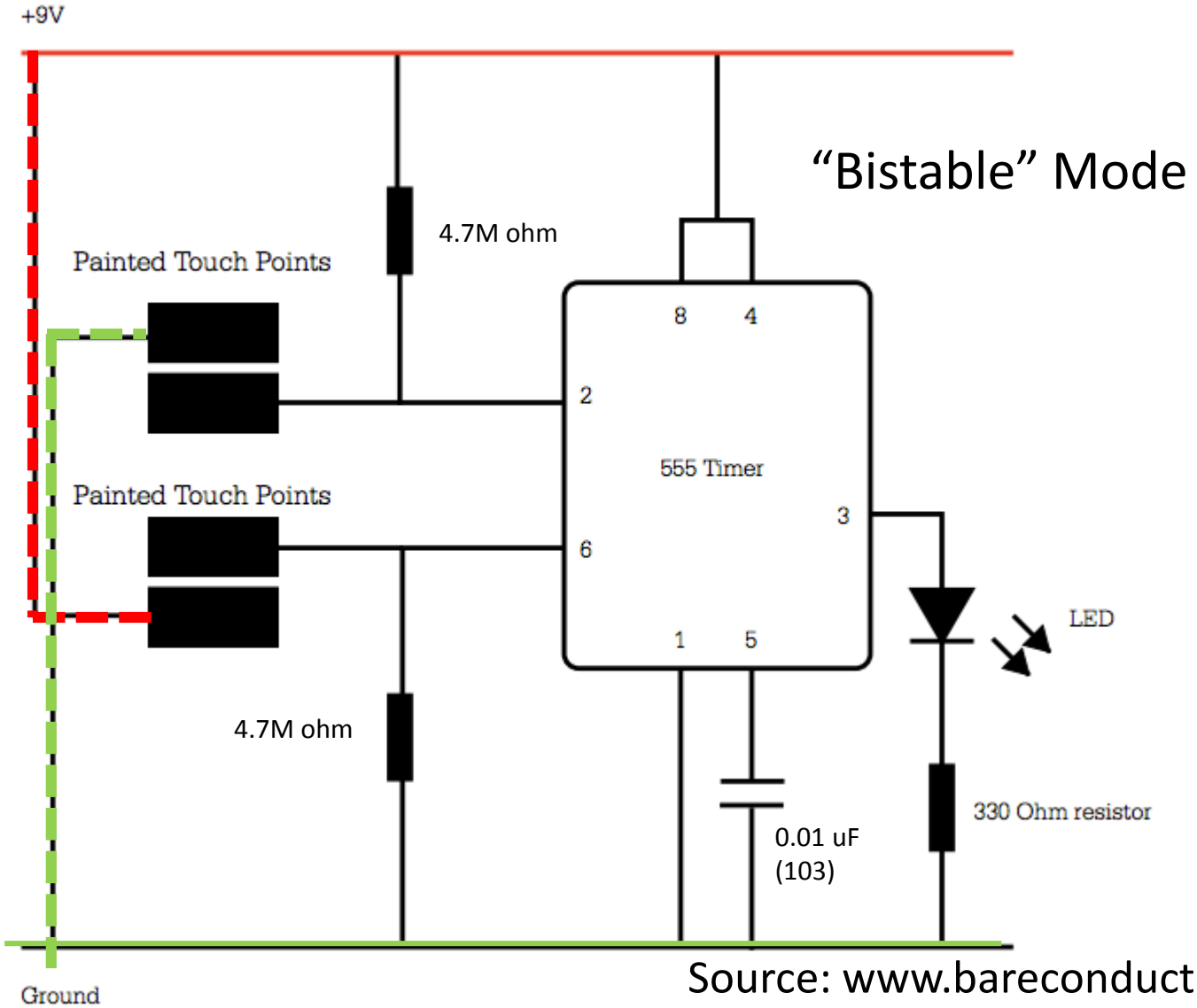


Source: http://www.555-timer-circuits.com/operating-modes.html

Project 2 - Materials

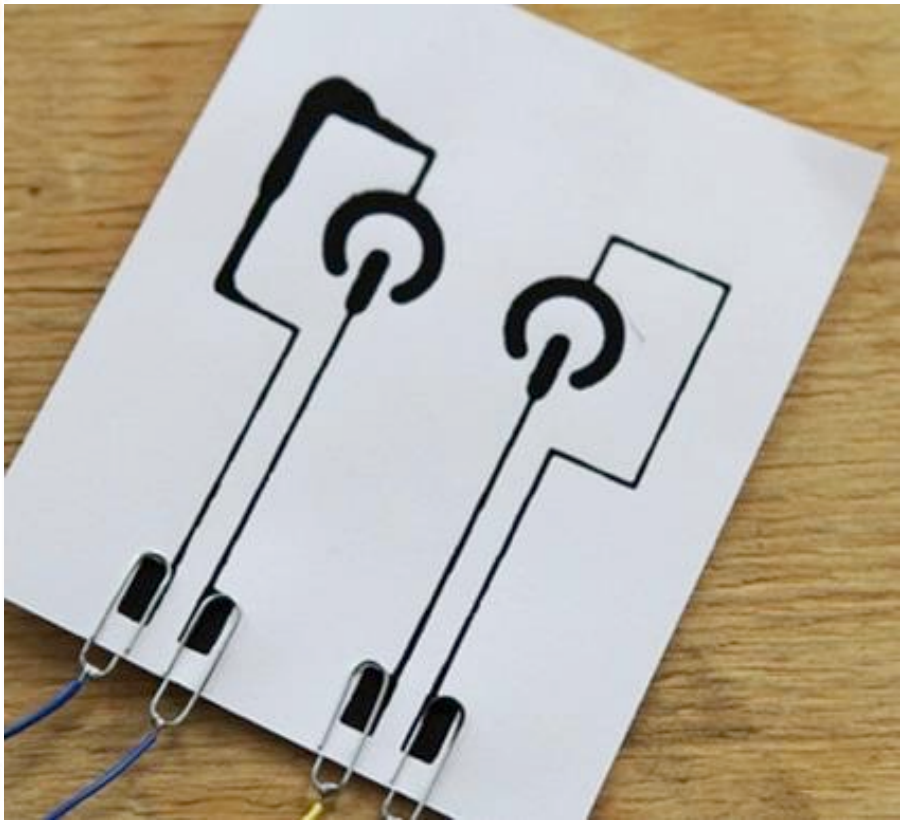


Project 02 - Schematic

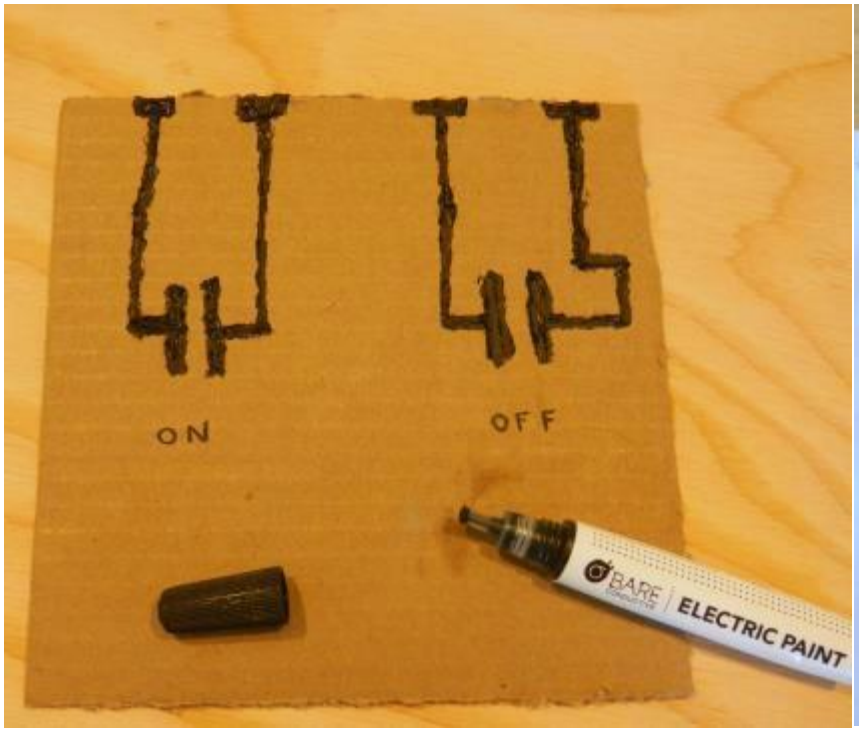


Source: www.bareconductive.com

Paint the touch sensors



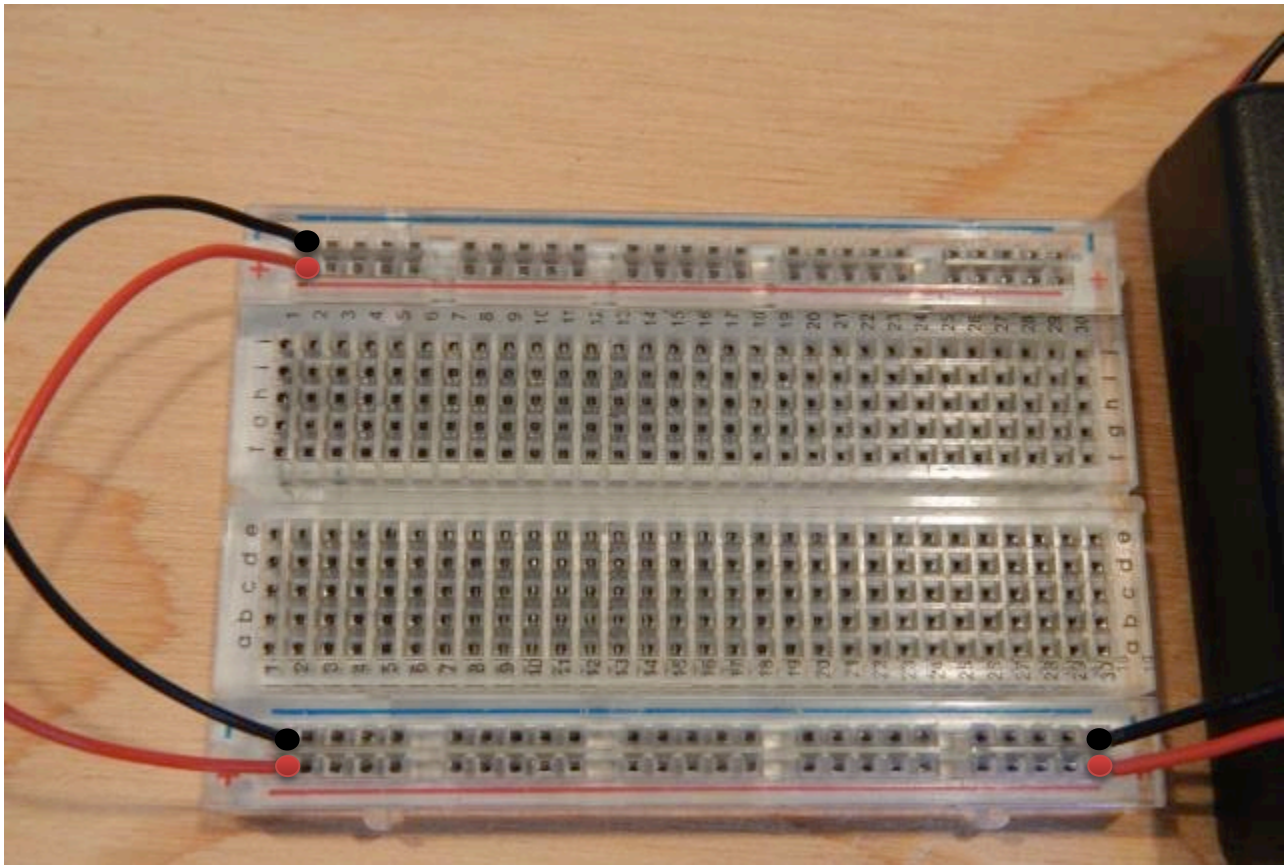
Paint the touch sensors



Conductive Ink Pen

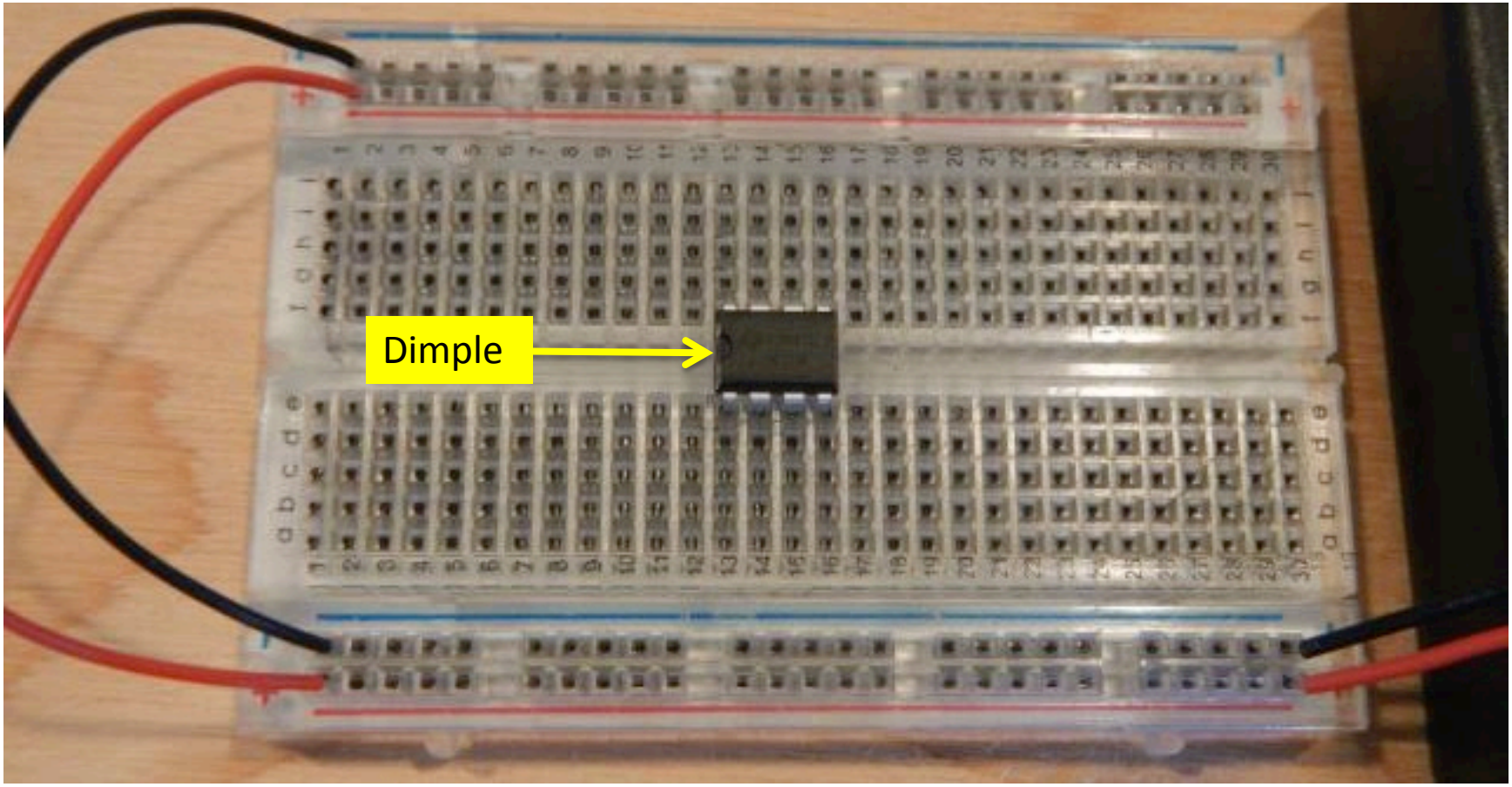
Conductive Ink Pots + Brush

Step 1



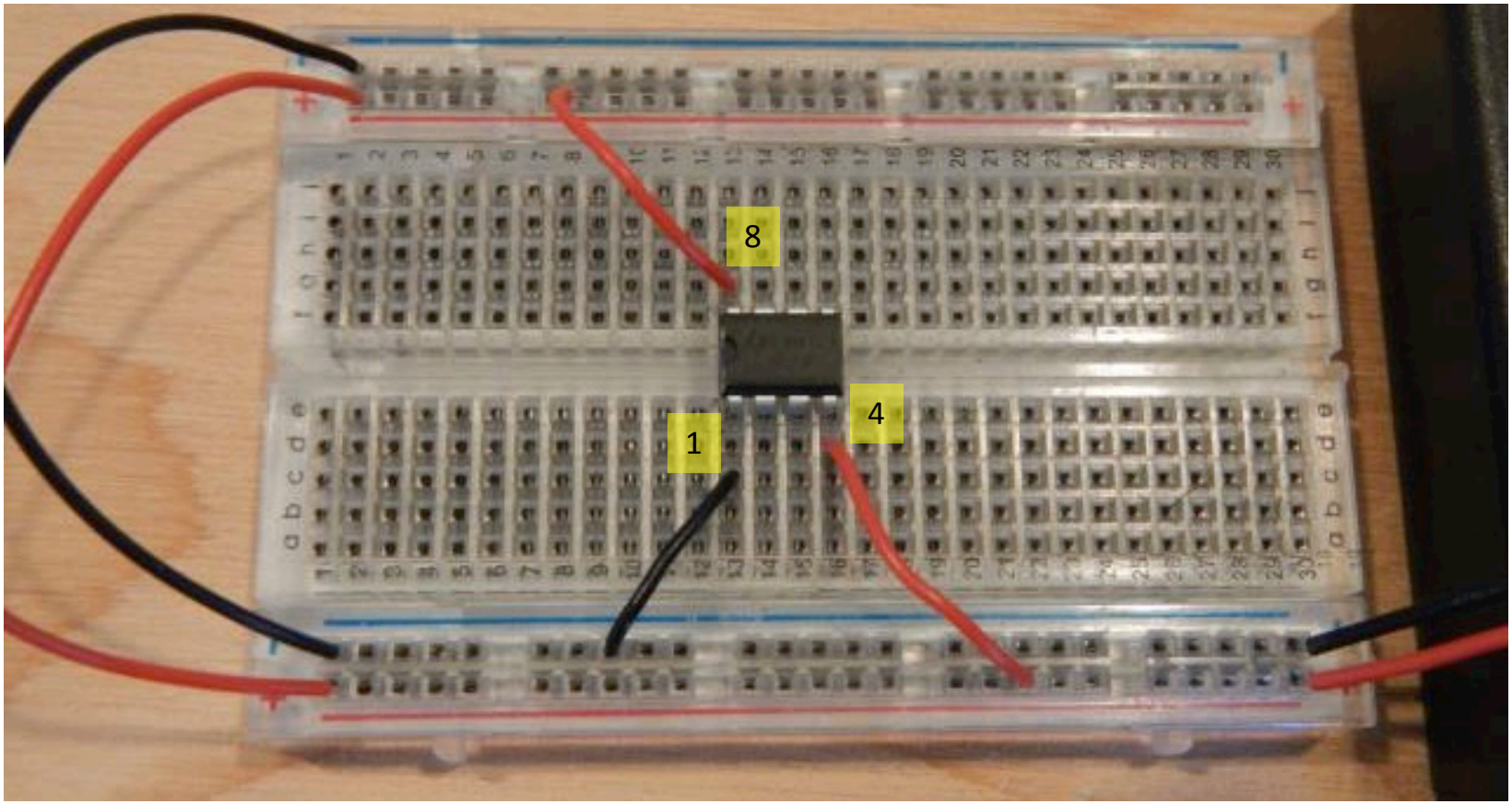
Insert the positive and negative battery wires into the power rails.
Use two more wires to jump power to the opposite side of the board.

Step 2



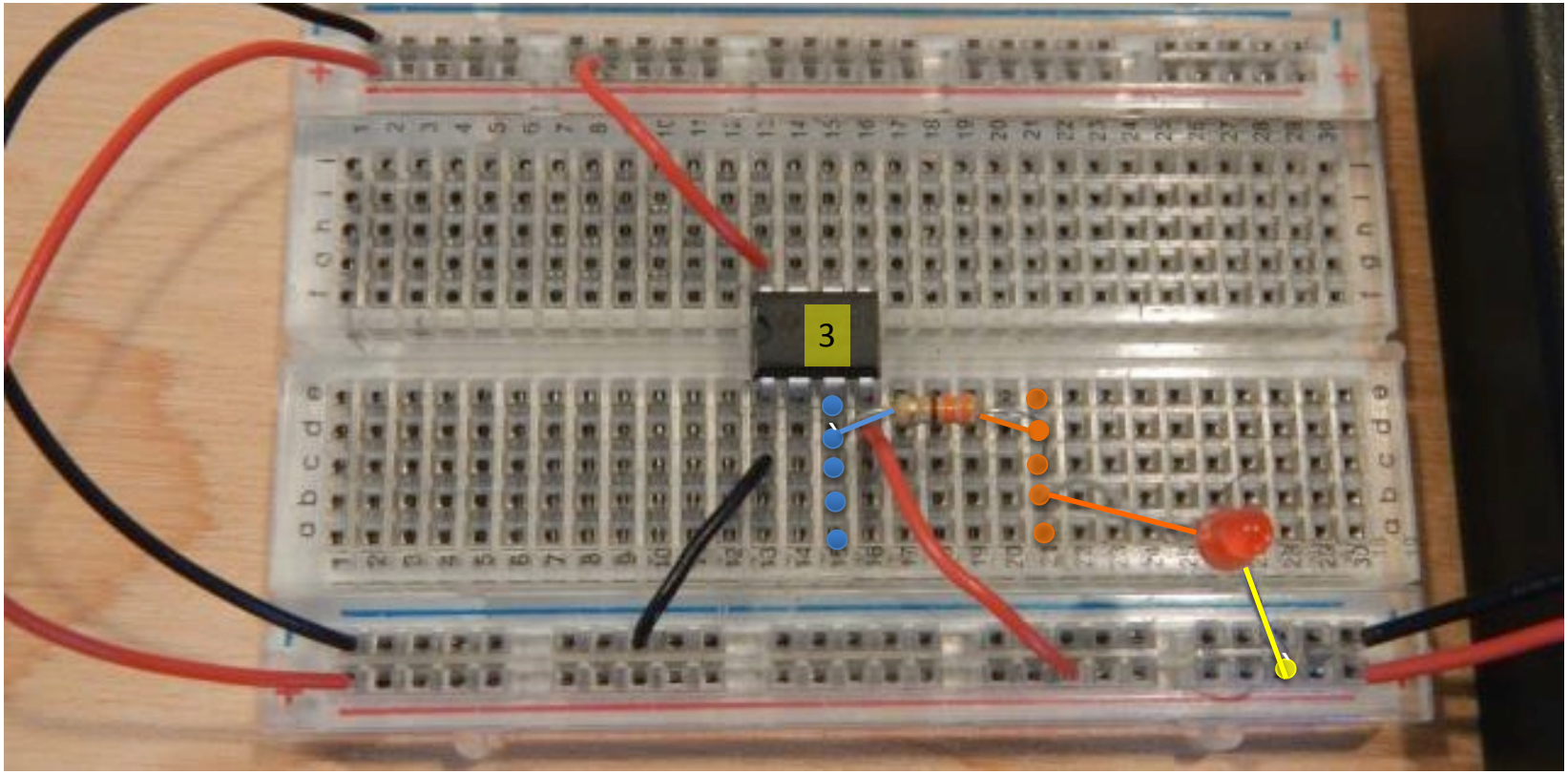
Place the 555 timer in the middle of the breadboard.

Step 3



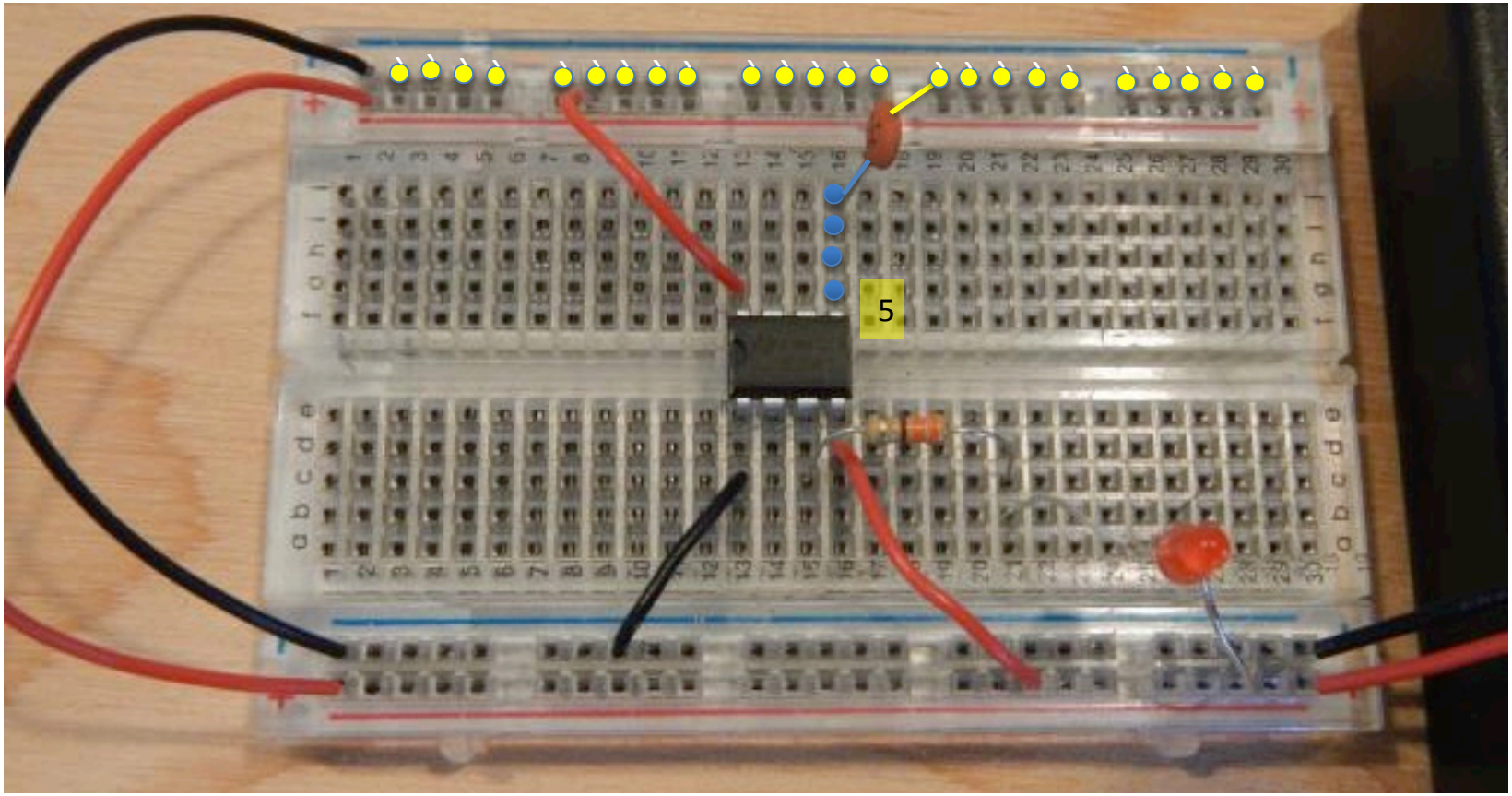
Connect Leg#1 to GND, connect Leg#4 to +V, connect Leg#8 to +V.

Step 4



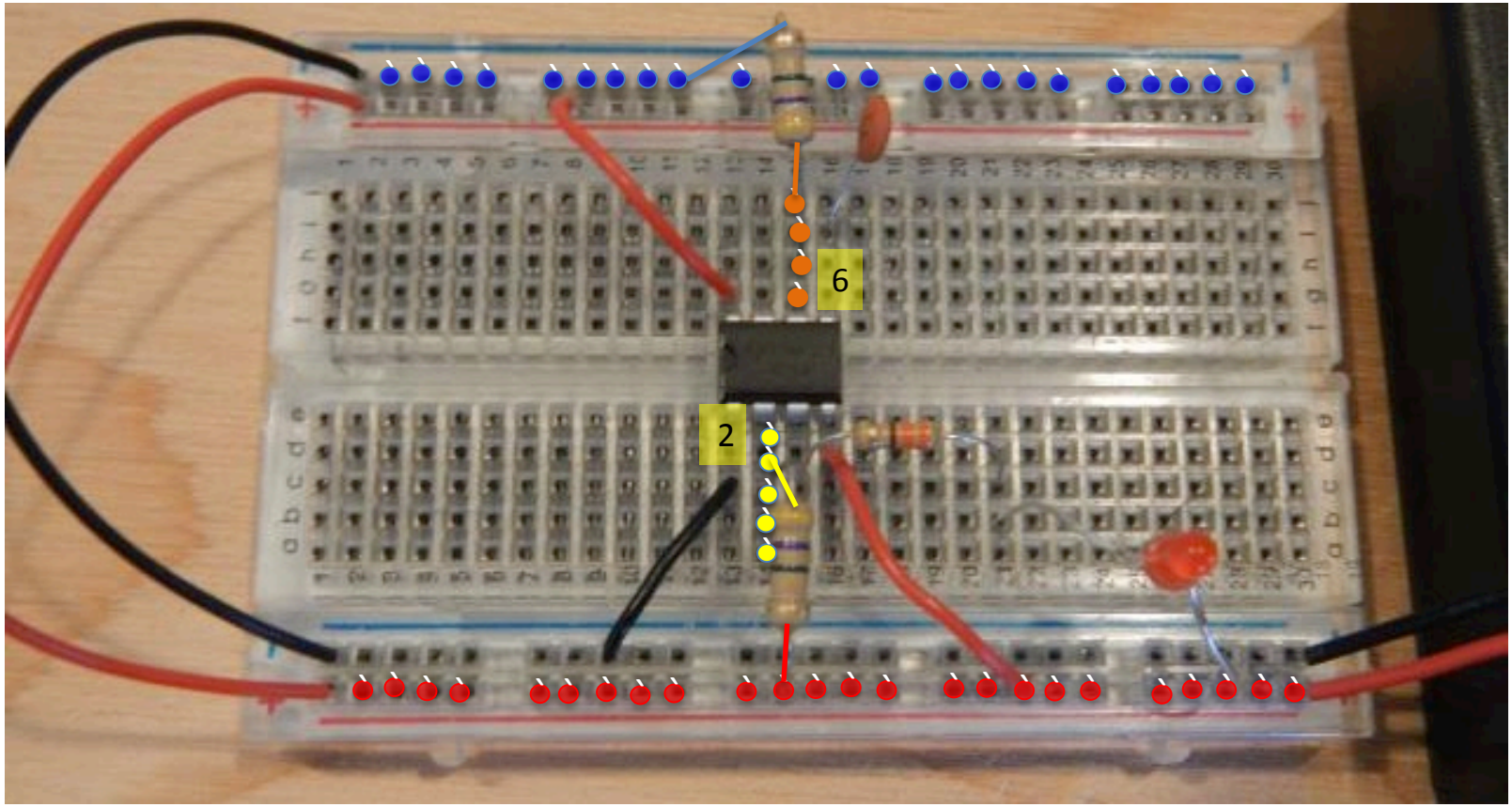
Jump a 330 ohm resistor from Leg#3 to the long leg of the LED.
Then connect the short leg of LED to GND

Step 5



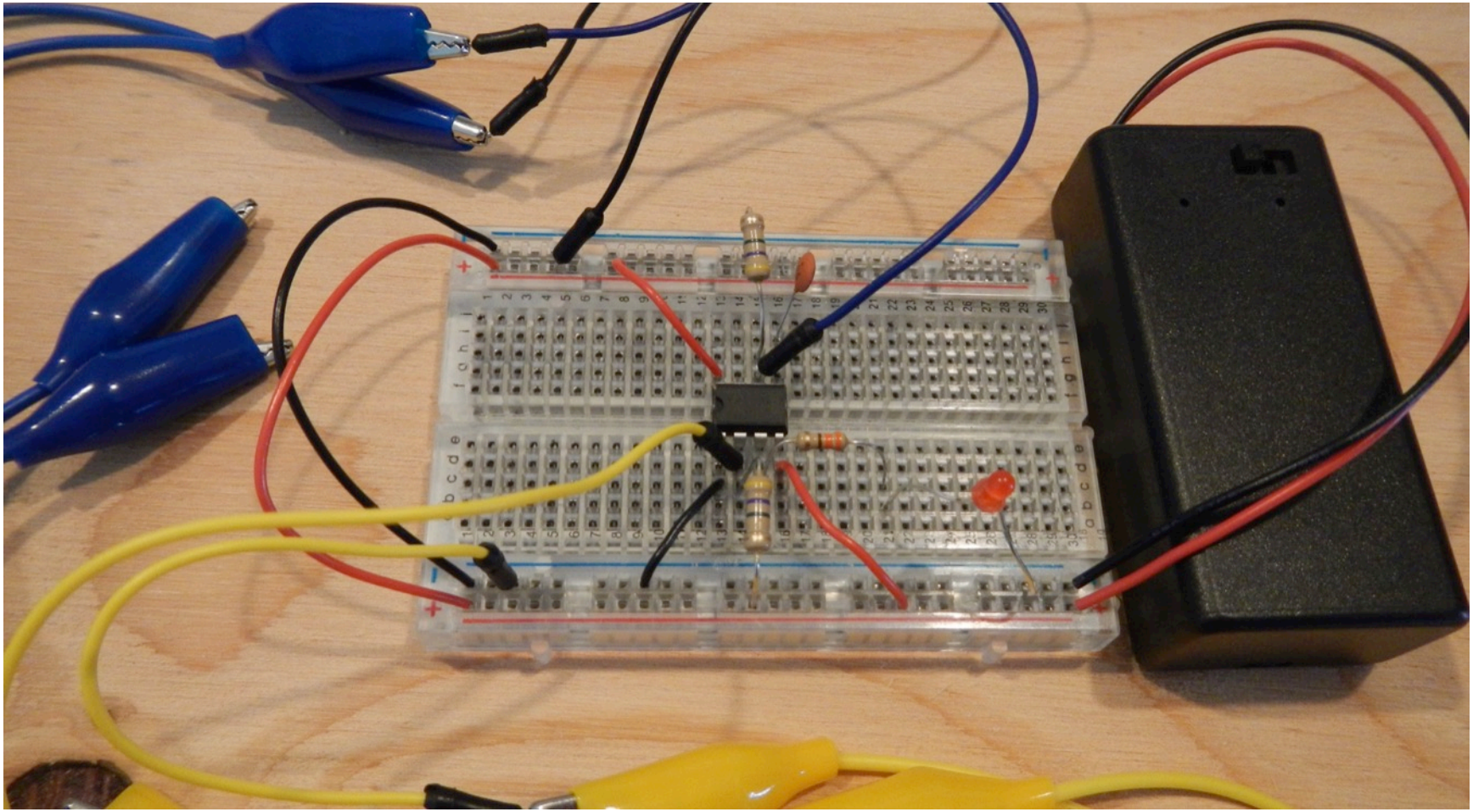
Jump 0.01 uf capacitor (103) from Leg#5 to GND.

Step 6



Jump 4.7M ohm resistor from Leg#2 to +V,
Jump 4.7M ohm resistor from Leg#6 to GND
4.7M ohm = yellow, purple, green, gold.

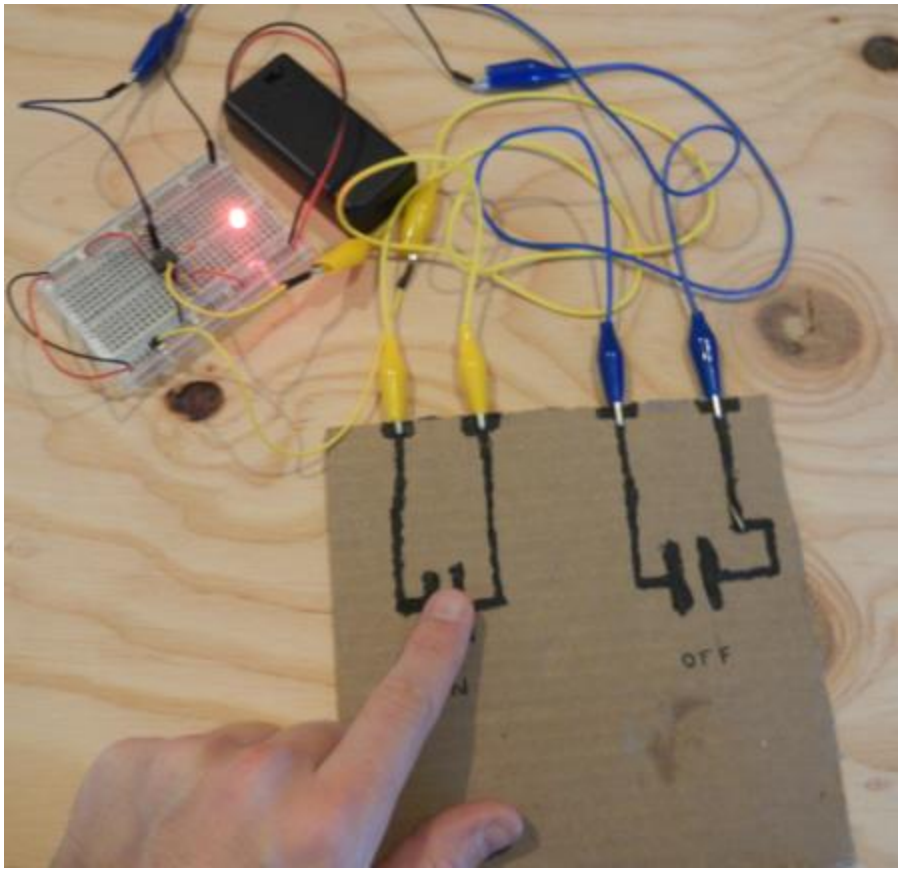
Step 7



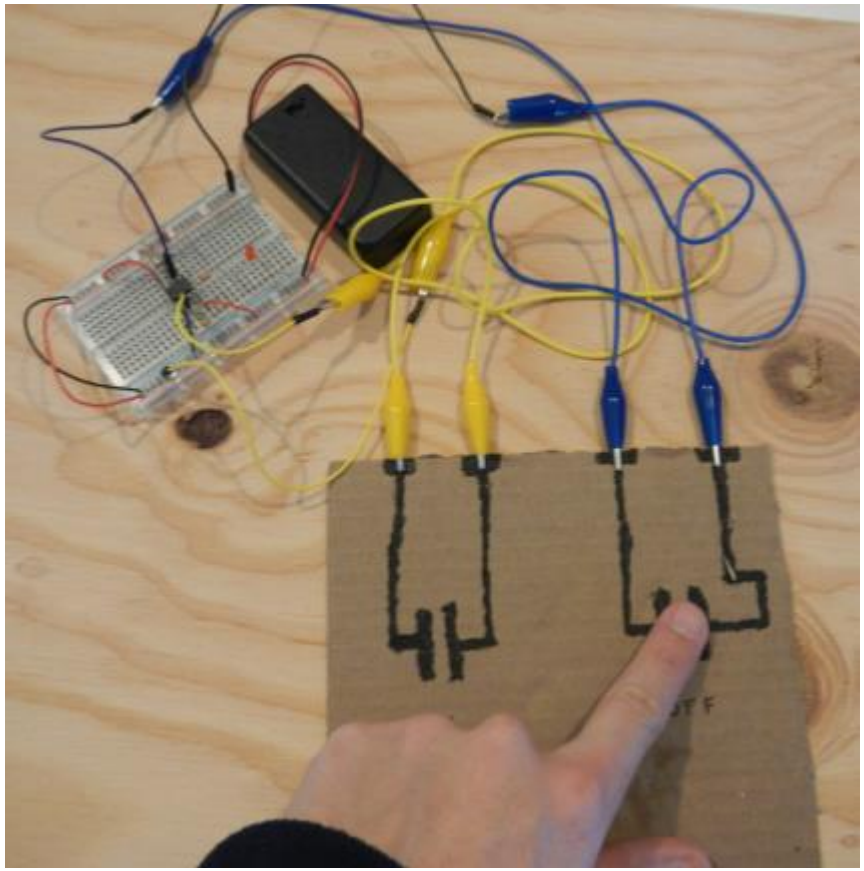
Place a yellow alligator clip on Leg#2 row, and another on GND
Place a blue alligator clip on Leg#6 row, and another on +V rail.`

Attach Conductive Sensors

Turn the LED ON/OFF

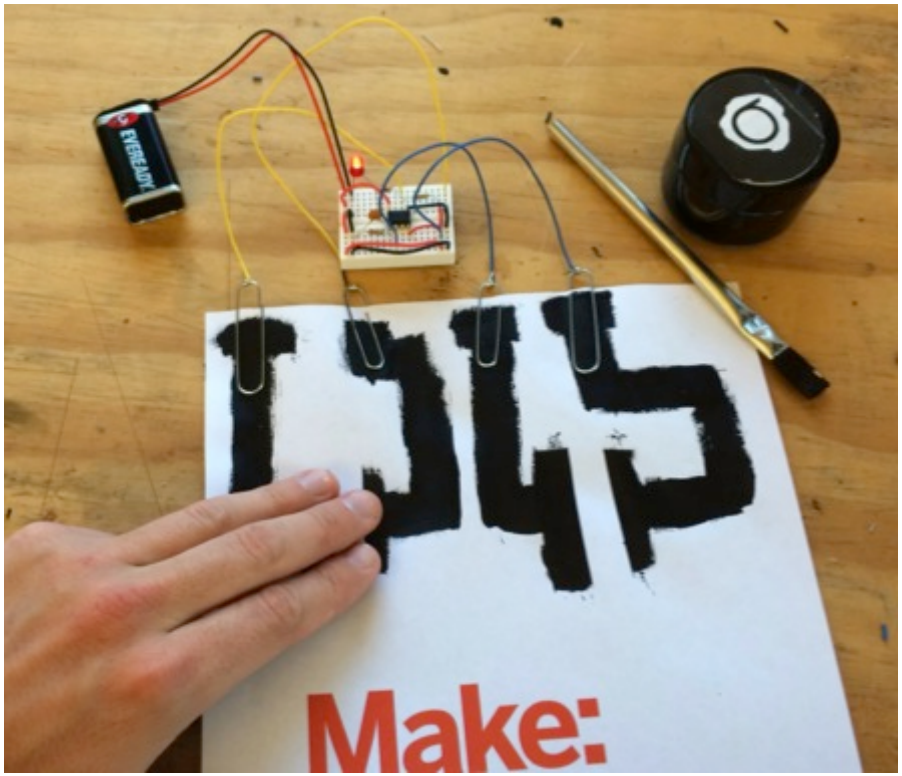


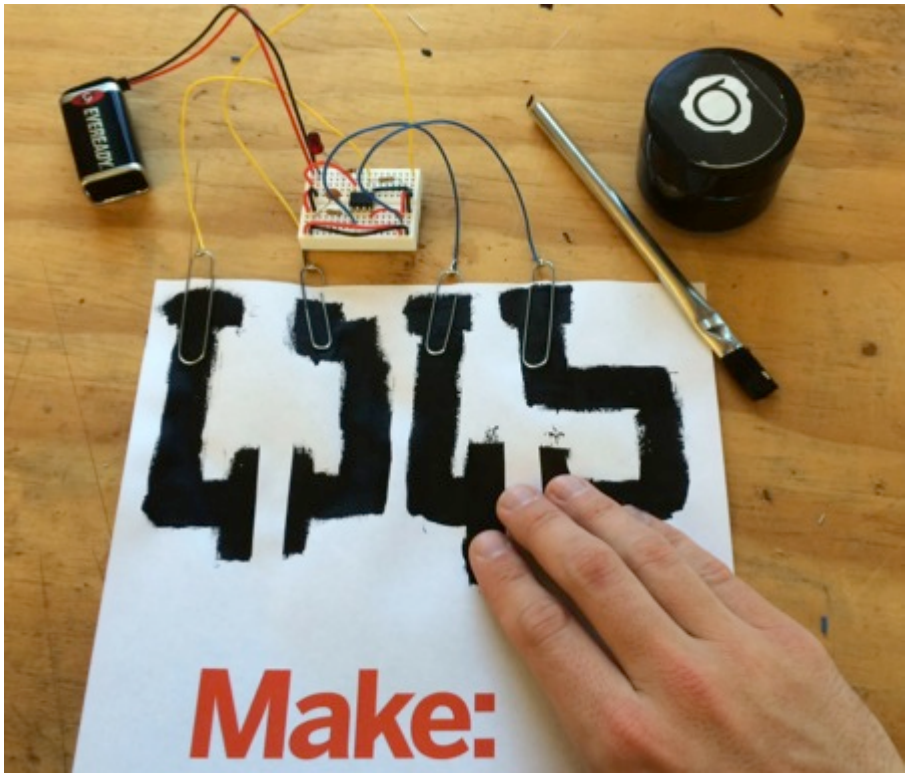
ON



OFF

Turn the LED ON/OFF



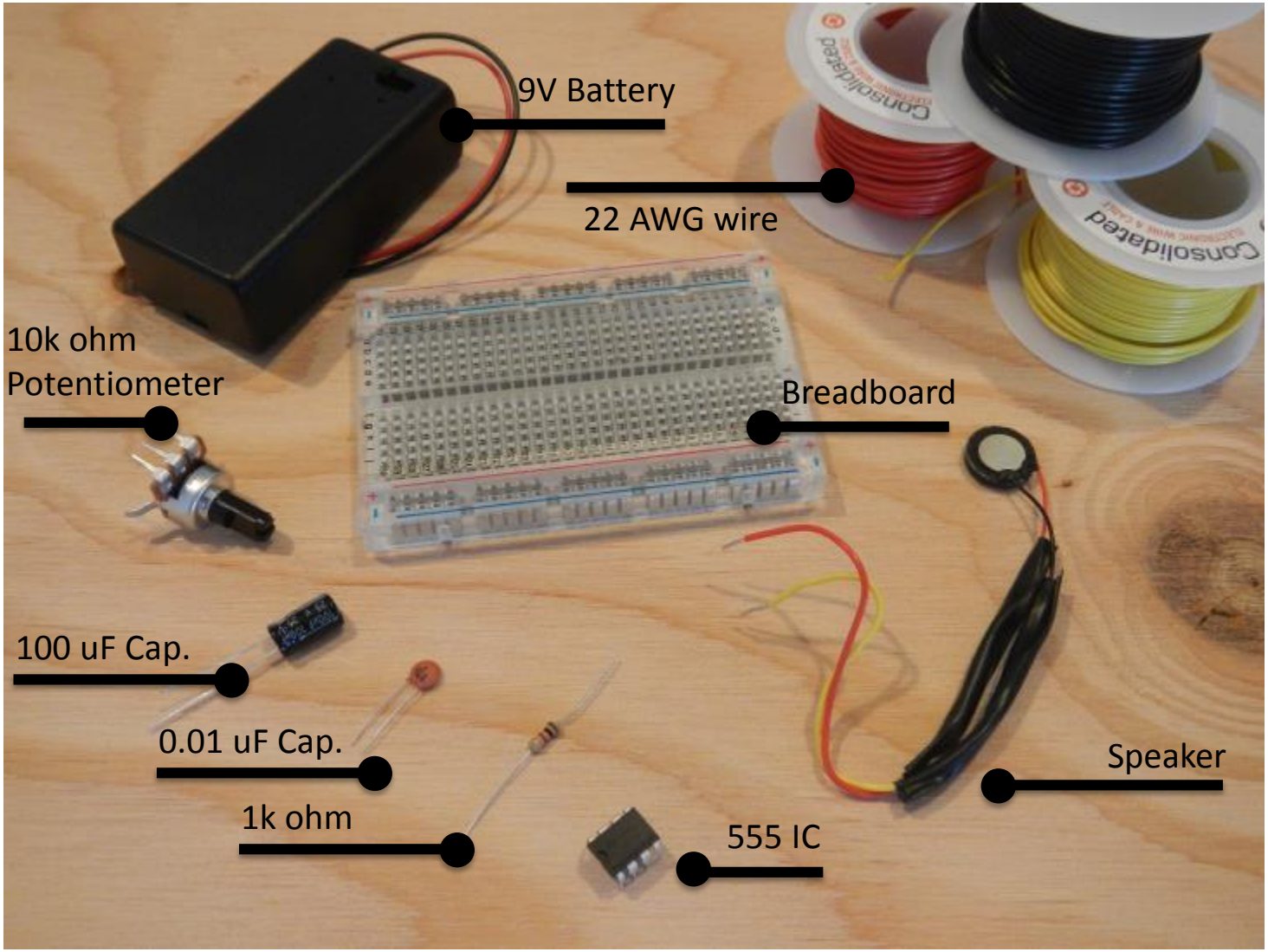


Other Ideas

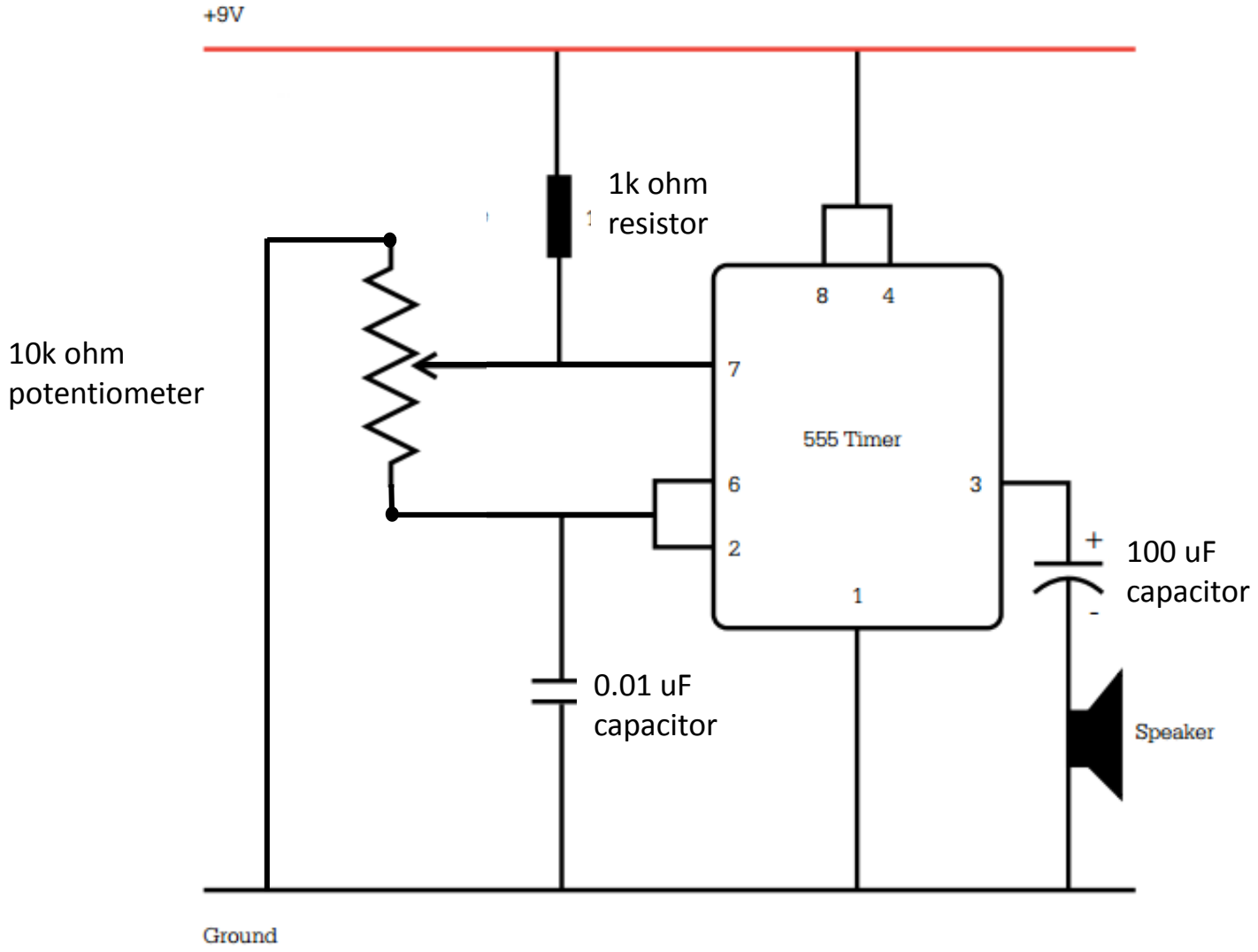
- You can make your own touch sensors
- Draw different contact pads
- Use the conductive paint to make more complicated patterns and circuits
- OR use the same circuit but instead of conductive paint you can use cardboard and aluminum foil to make a pressure sensor

Questions?

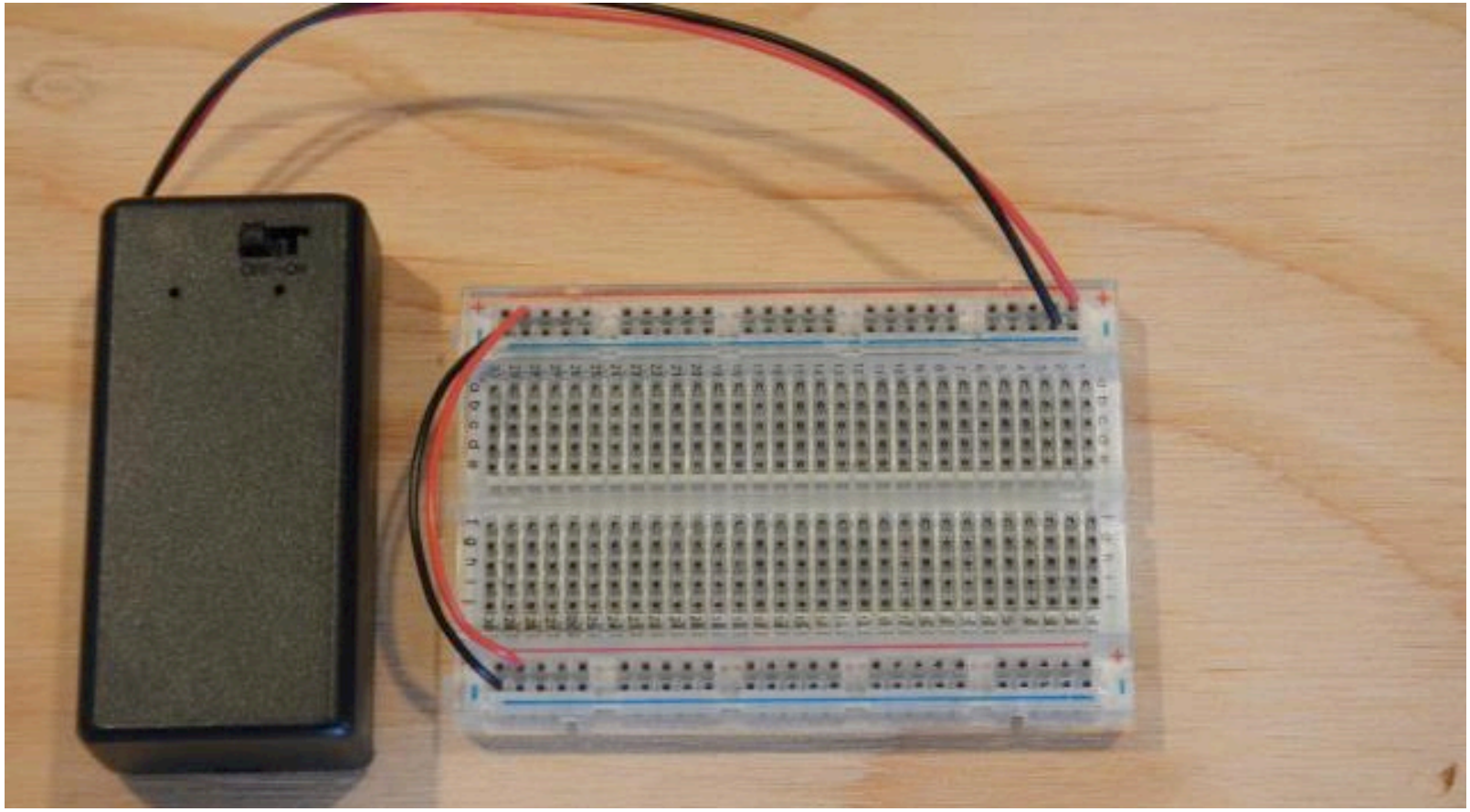
Project 03 – Materials



Project 03 - Schematic

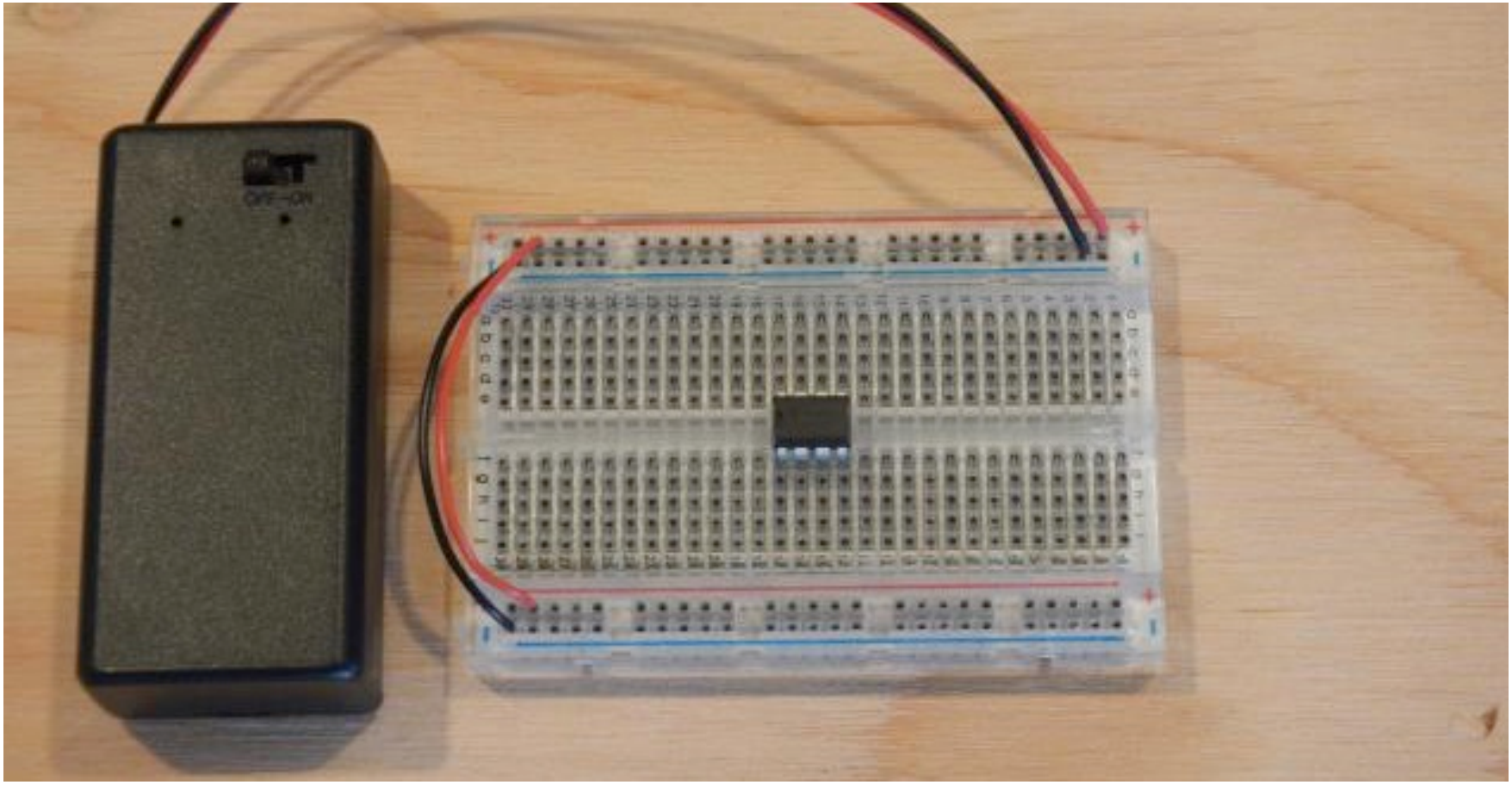


Step 1



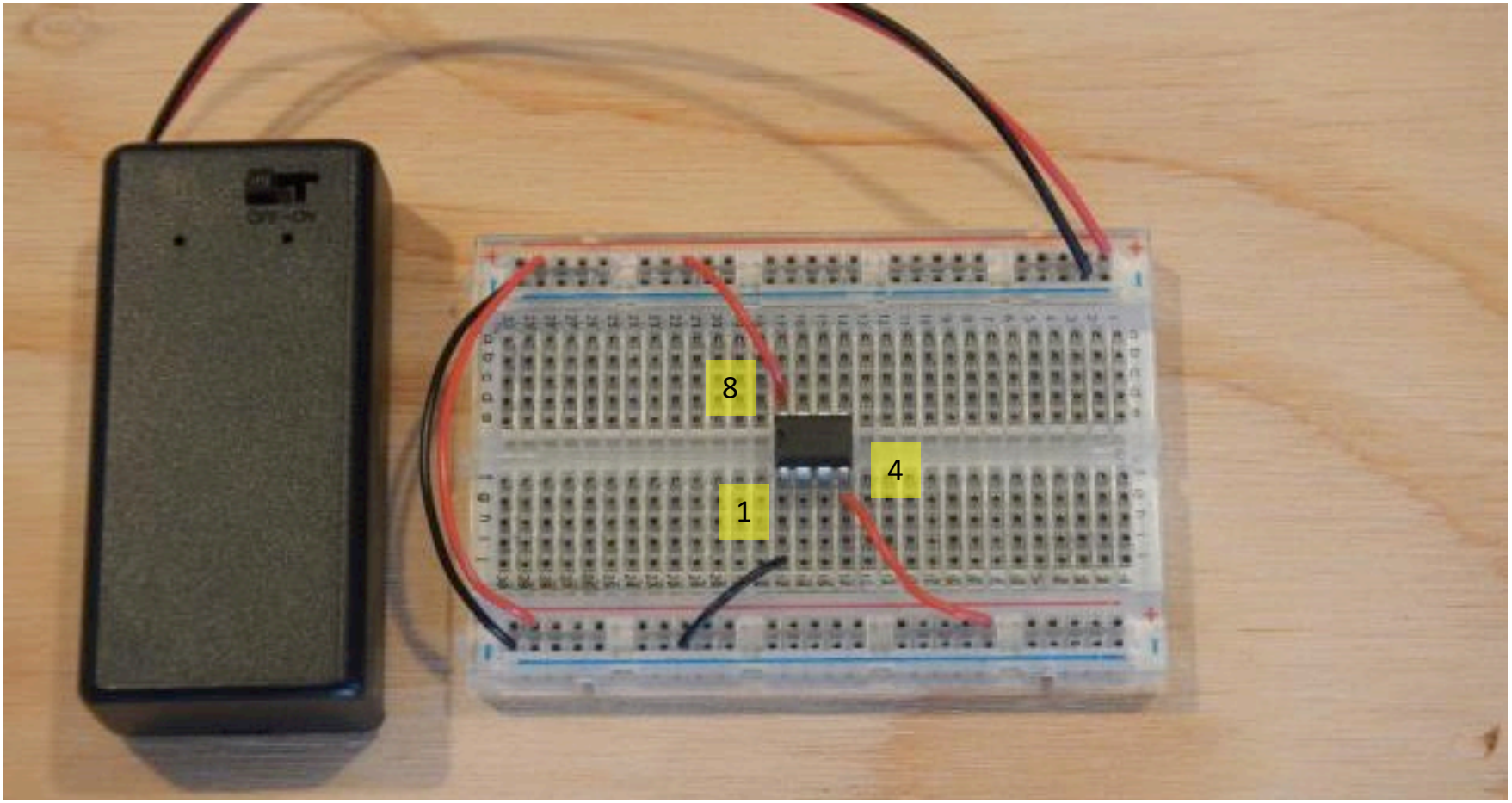
Add +V and GND wires to power rails, and add jumpers across.

Step 2



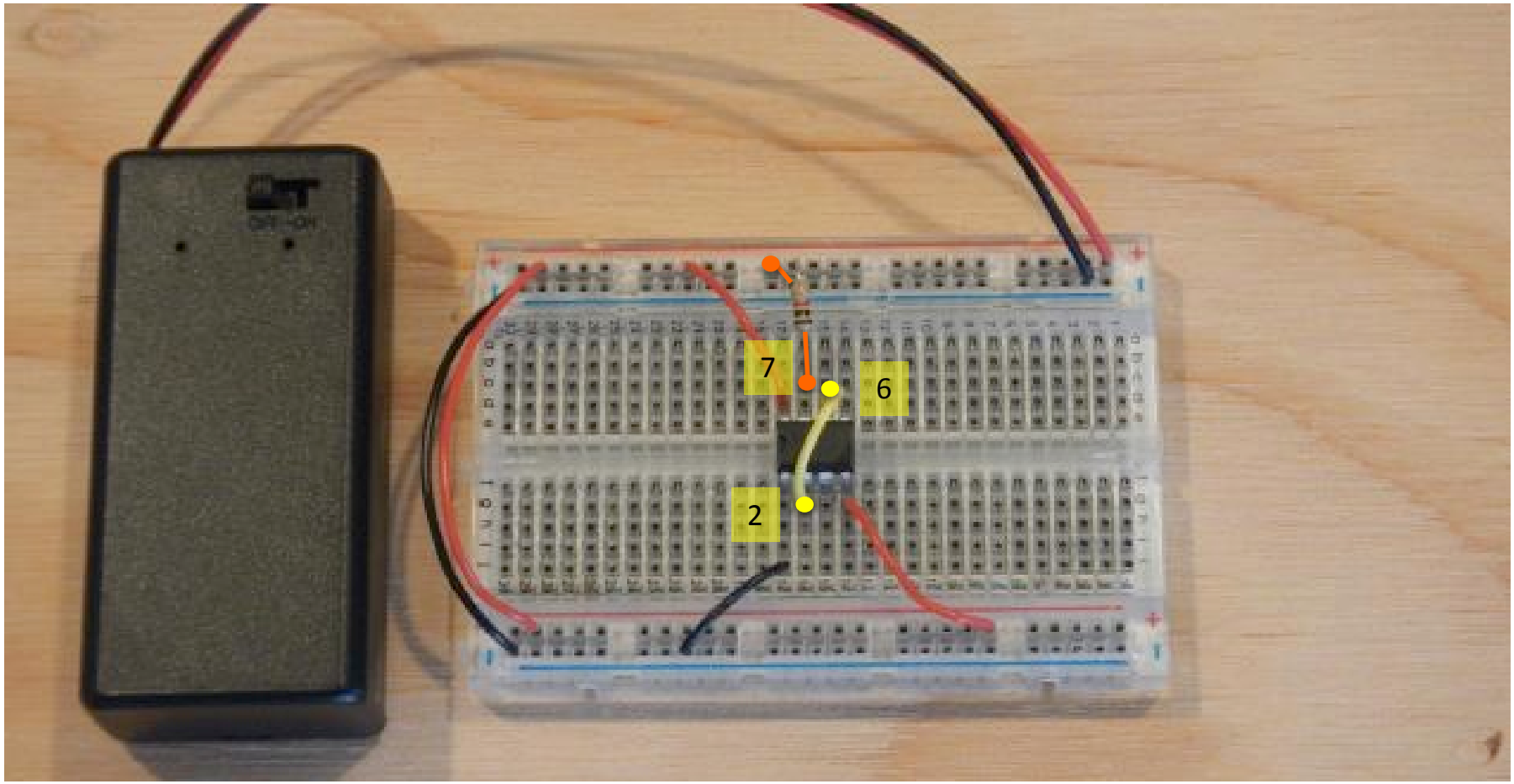
Add 555 timer to the middle of the board, dimple facing LEFT.

Step 3



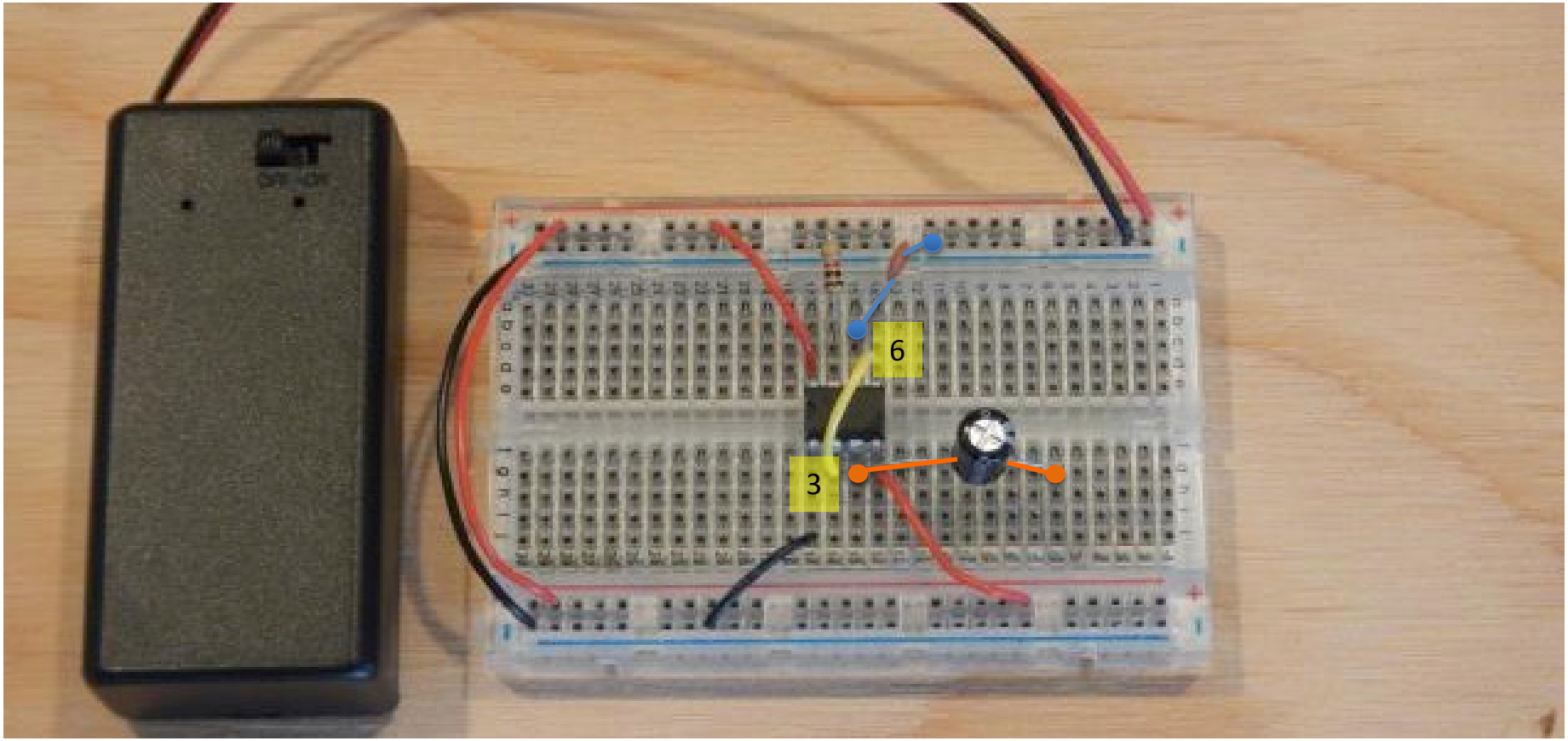
Jump Leg#1 to GND, jump Leg#4 to +V, jump Leg#8 to +V.

Step 4



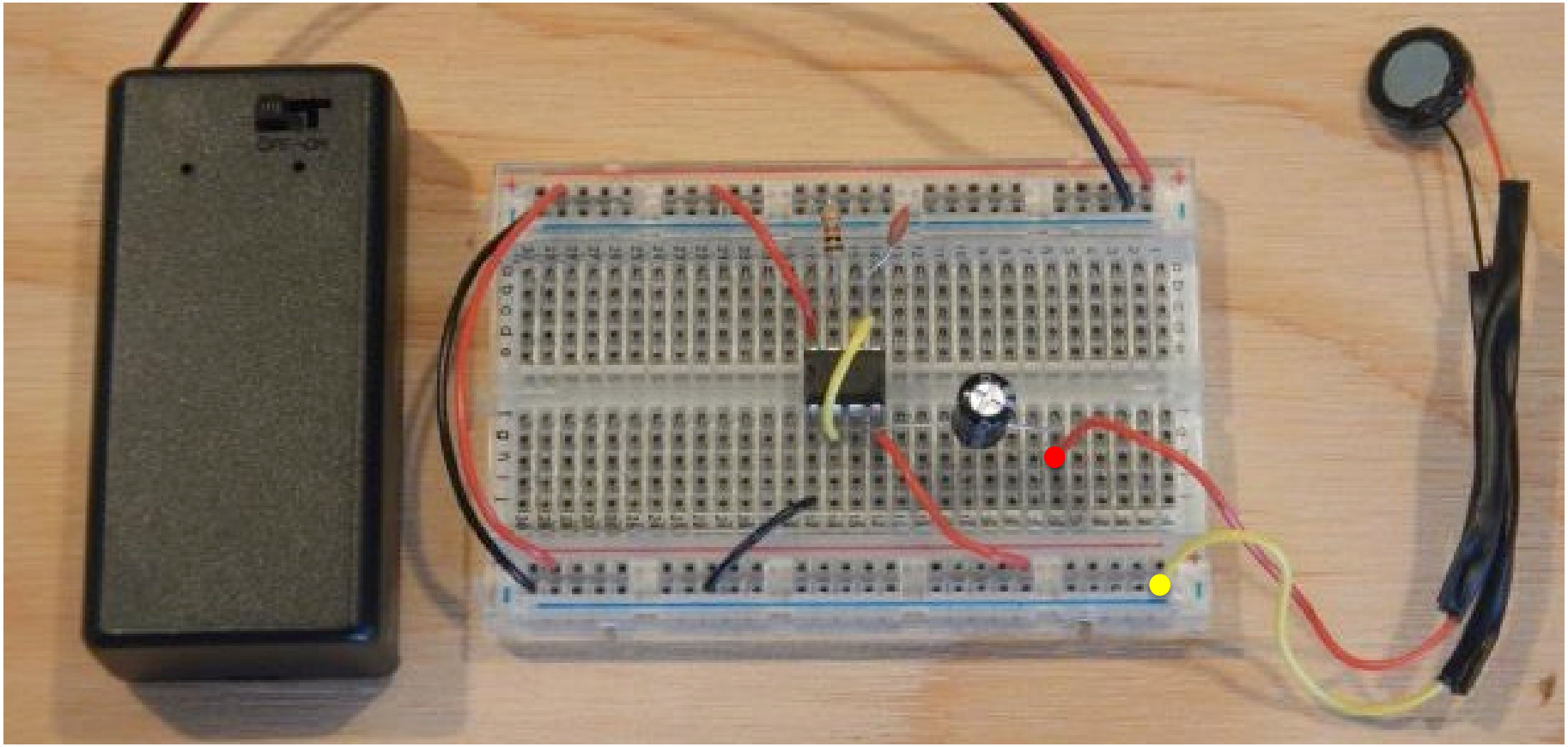
Use a short yellow wire, jump from Leg#2 to Leg#6.
Jump a 1k ohm resistor from Leg#7 to +V. (brown, black, red, gold)

Step 5



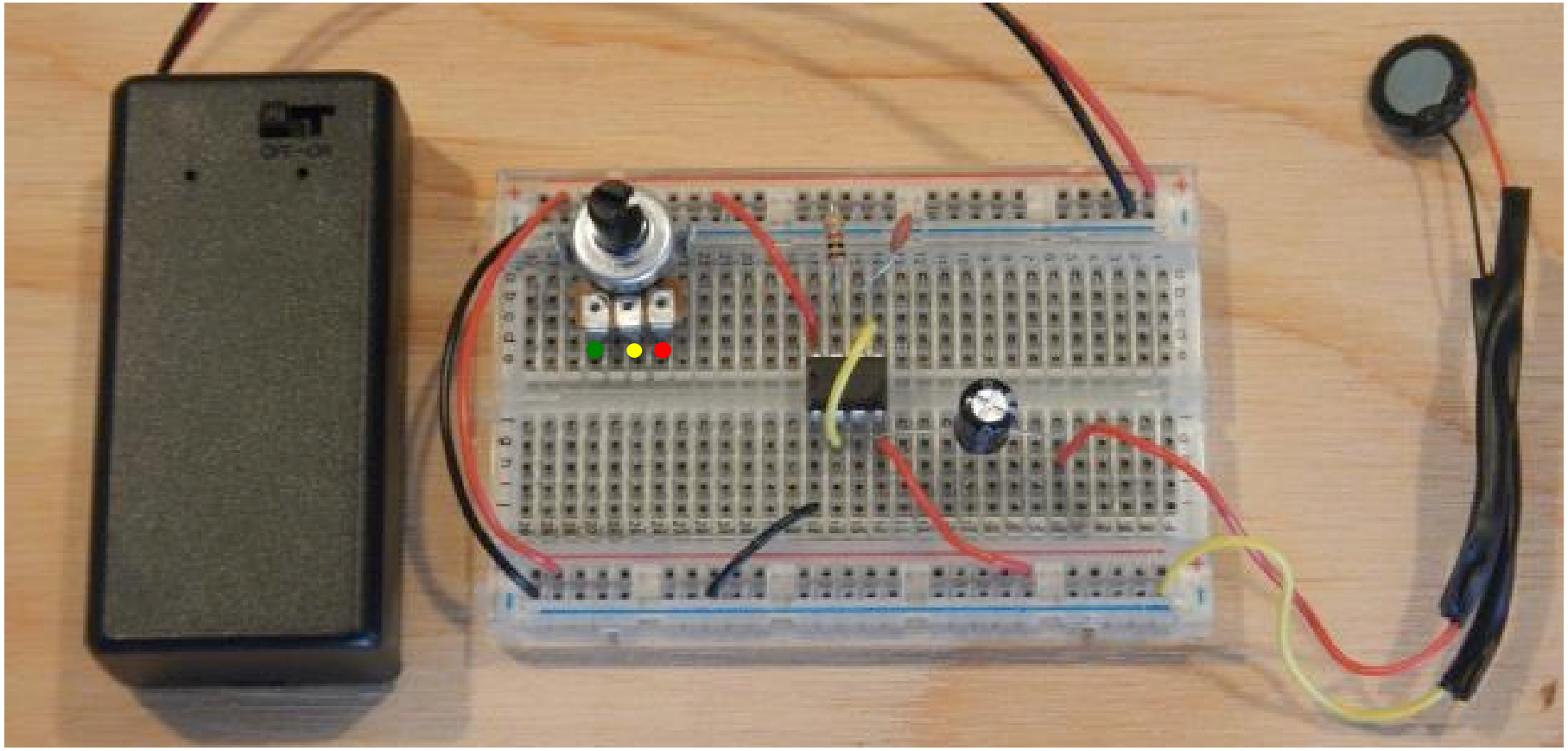
Place long leg of large capacitor, 100 uf, into Leg#3.
Add a small capacitor, 0.01 uf (103), from Leg#6 to GND.

Step 6



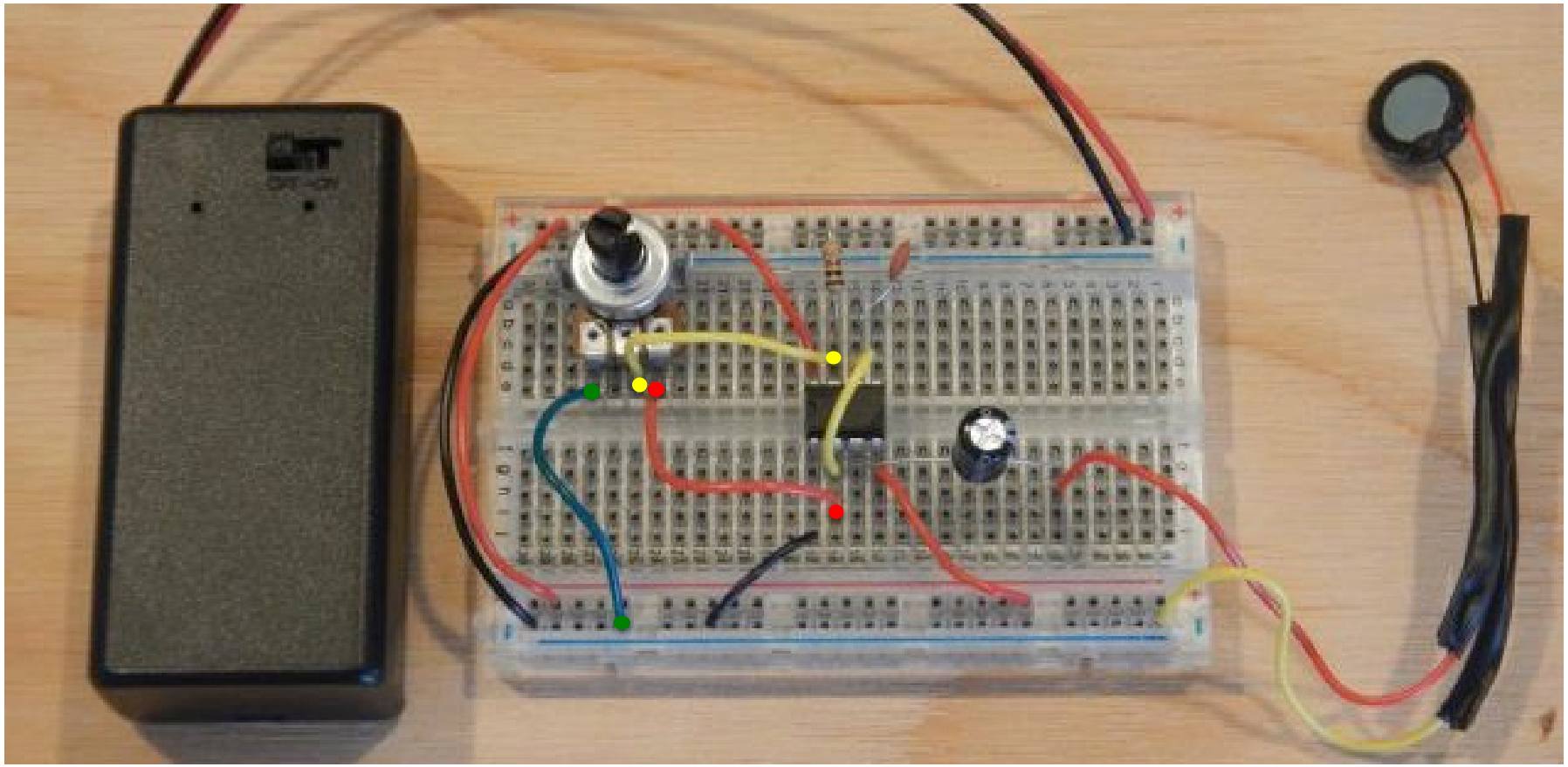
Add the speaker, one wire from speaker to short leg of 100 uf cap,
Add the second wire to GND. Note: Speakers not polarized.

Step 7



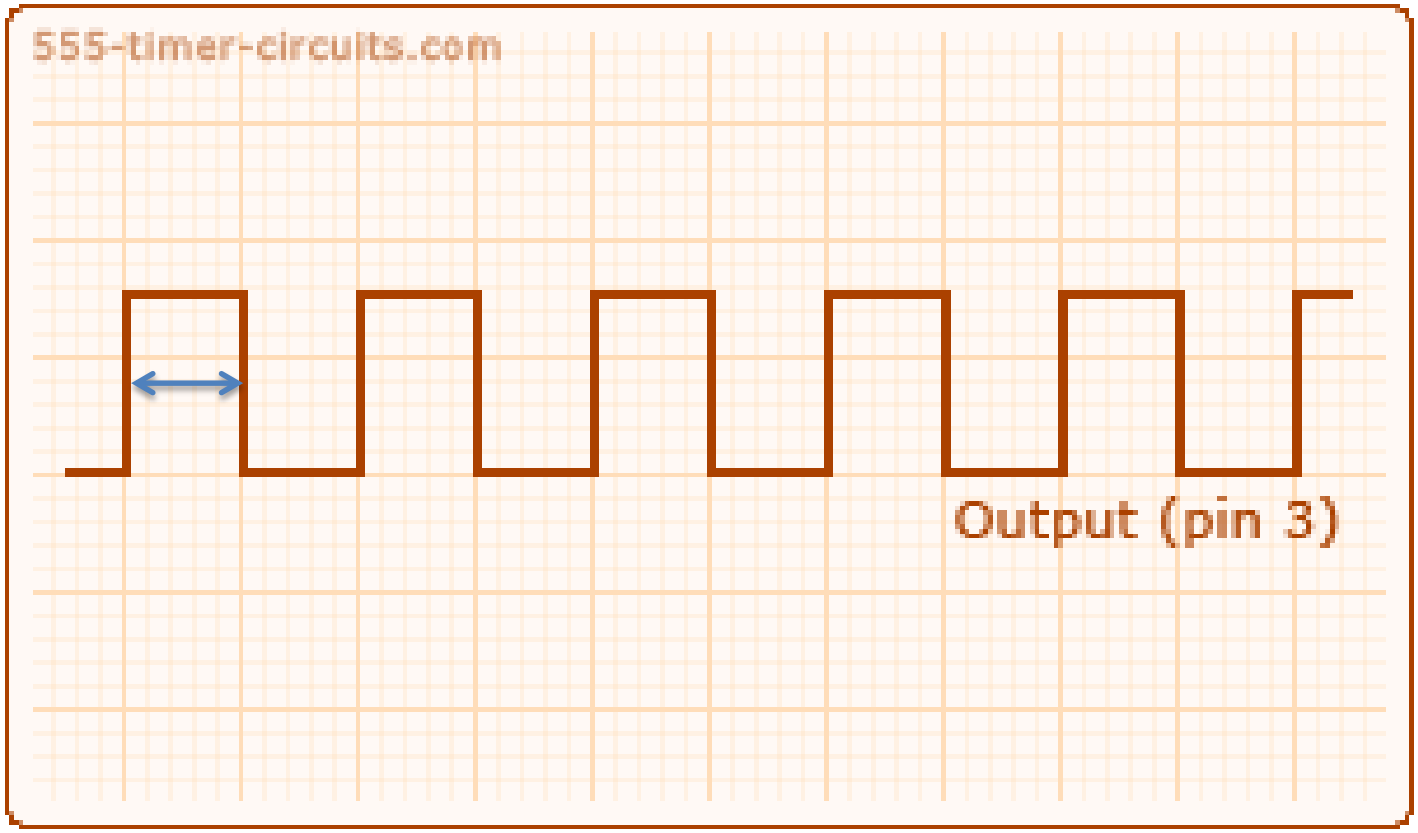
Add the 10k ohm potentiometer to the breadboard, top left.

Step 8



Jump left leg of pot to GND, right leg of pot to Leg#2, Middle leg of pot to Leg#7. Then turn on battery switch!

Astable mode - How it works



As we rotate the potentiometer, we change the input resistance. This change in resistance controls frequency of the output pulses. Short pulses lead to higher frequency noise from the speaker.

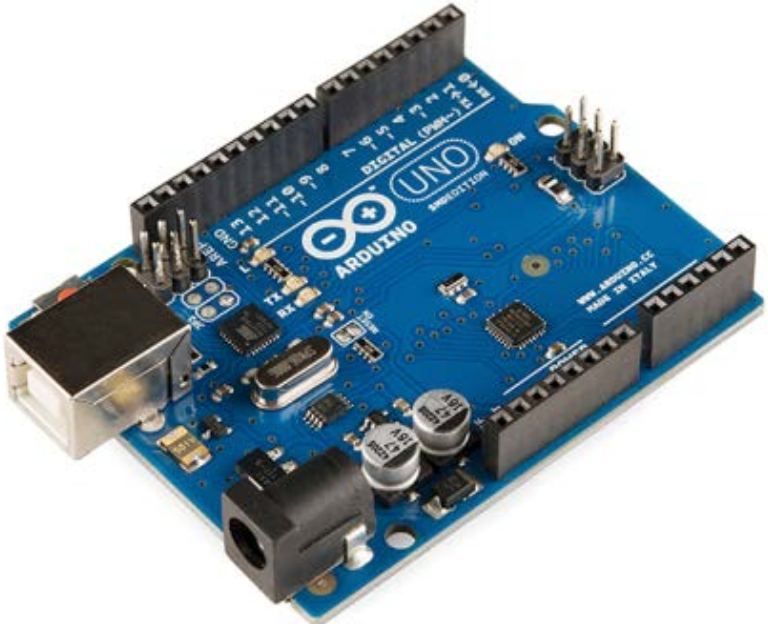
Questions?

LUNCH

Note: If you do not have the Arduino IDE installed, please install during lunch. If you need help, let us know!

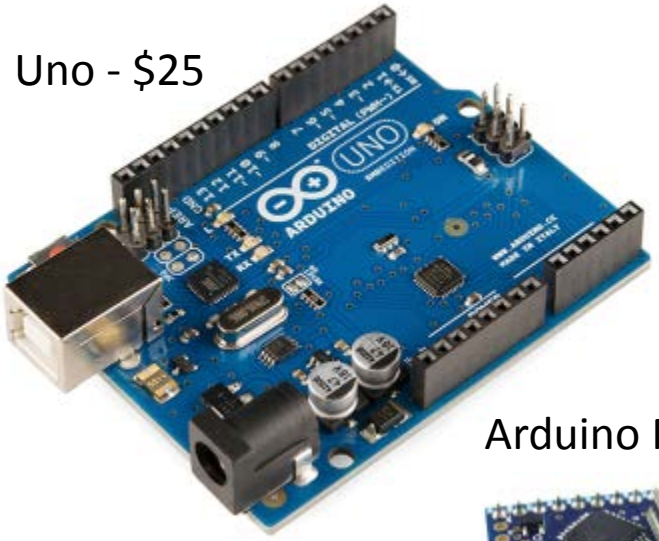
Workshop 1 - Outline

- What is Arduino? (hardware and software)
- Types of Arduino microcontrollers
- Arduino UNO Specifications
- Integrated Programming Environment - IDE
- Programming basics

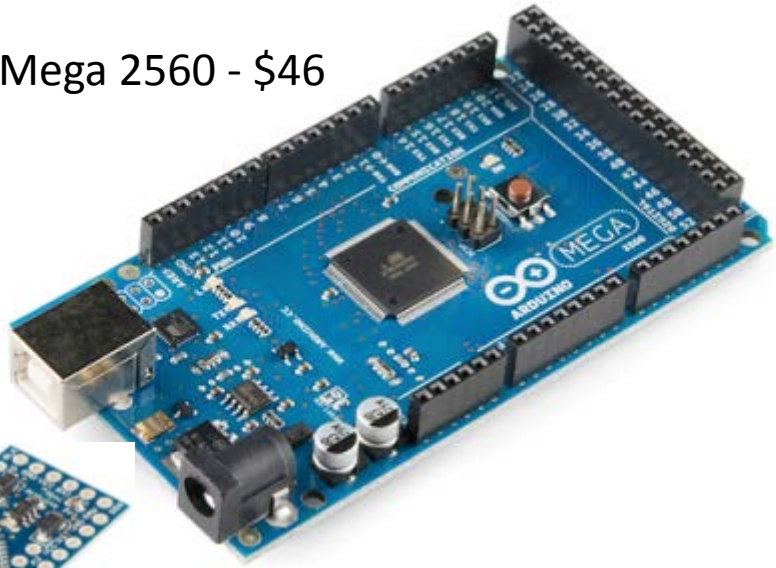


Types of Arduinos

Uno - $25



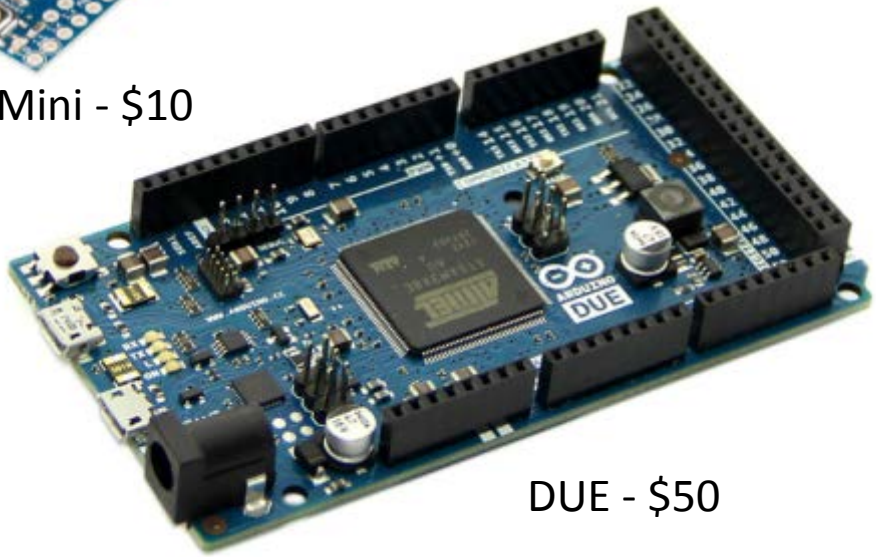
Mega 2560 - $46



Arduino Mini - $20

Pro Mini - $10

Leonardo - $25

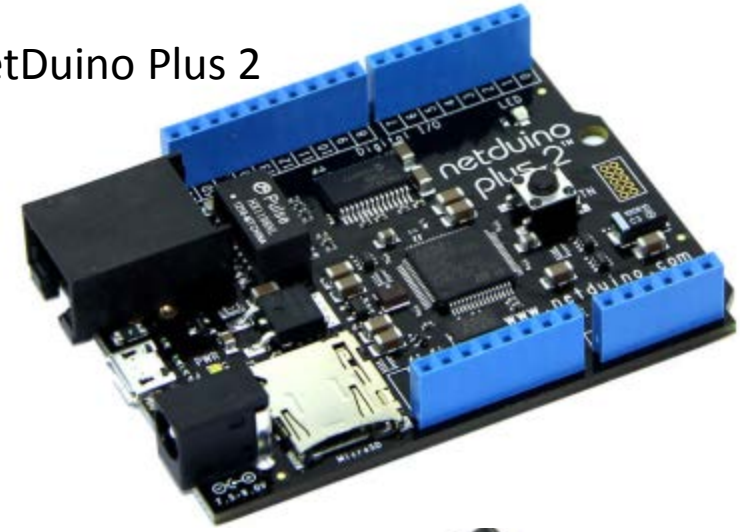


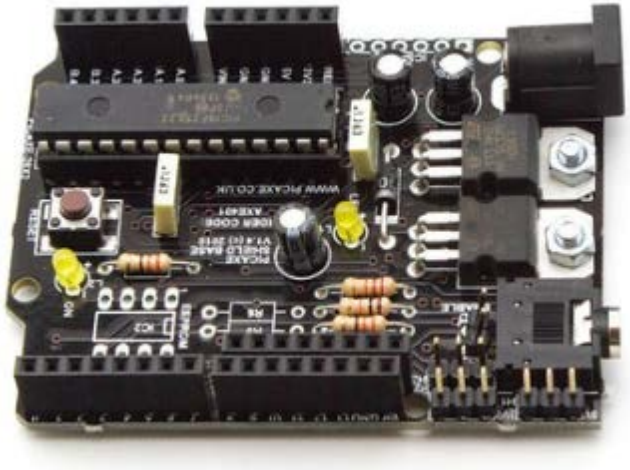
DUE - $50

Arduino Alternatives

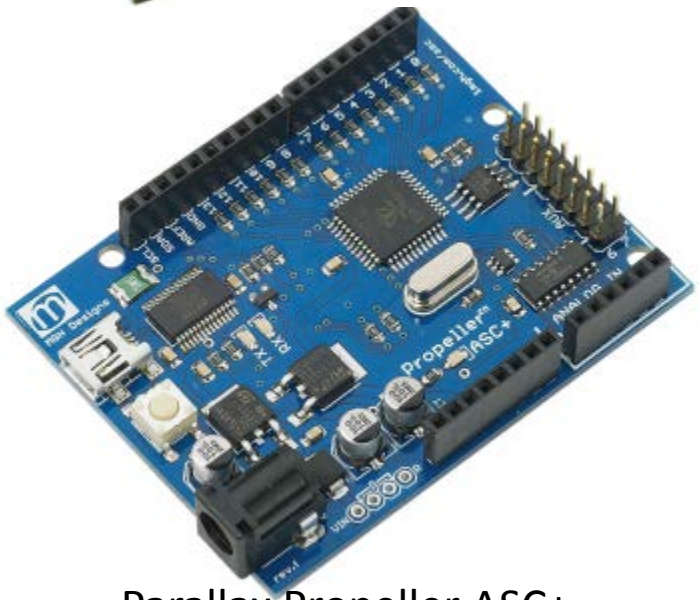
Ti Launch Pad MSP430

NetDuino Plus 2





Picaxe 28X2 Shield Base



Parallax Propeller ASC+

Build your own Arduino

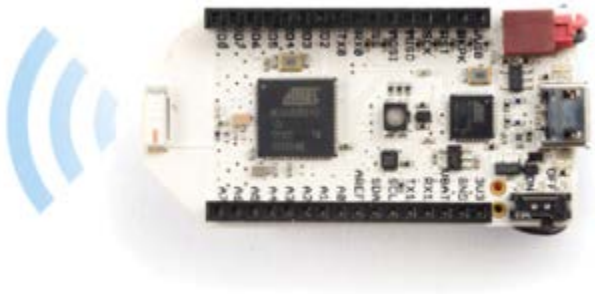
Adafruit MENTA -$35

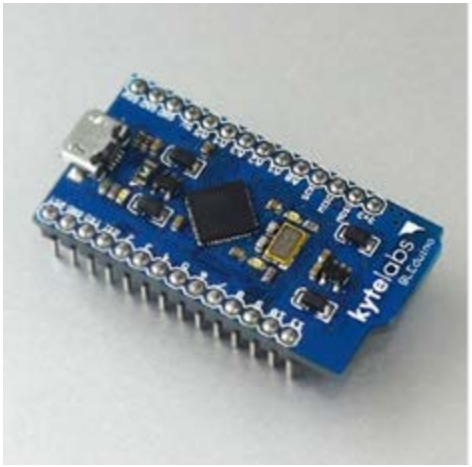
Makershed MintDuino -$25

Going Wireless

Arduino YUN

Pinoccio





BLEduino

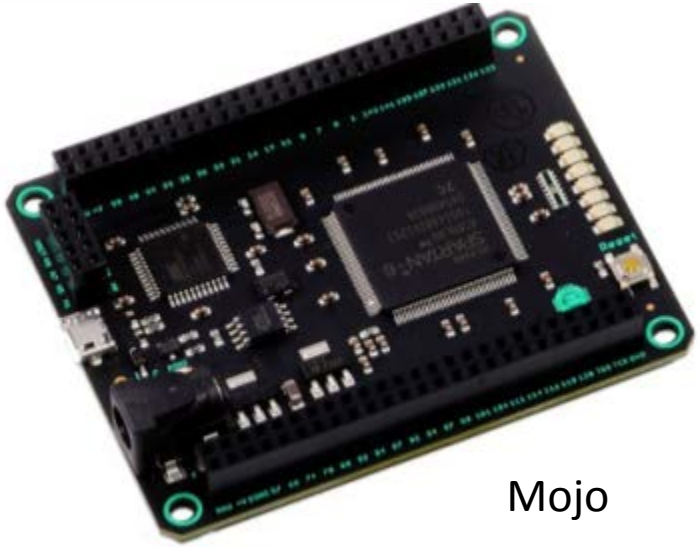
Geogram One

Single Board Computers & FPGA

BeagleBone Black

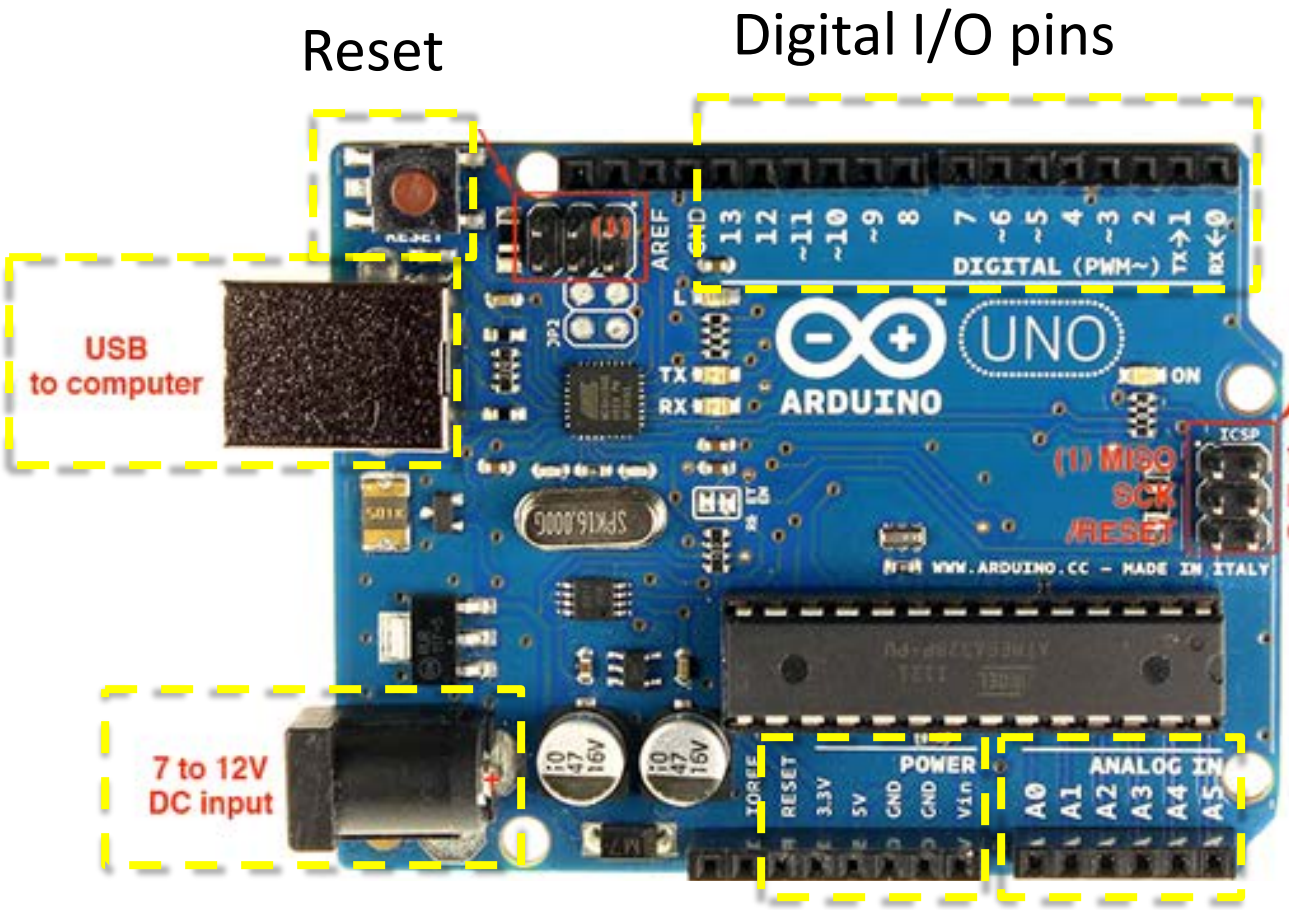
Raspberry Pi

Papilio One

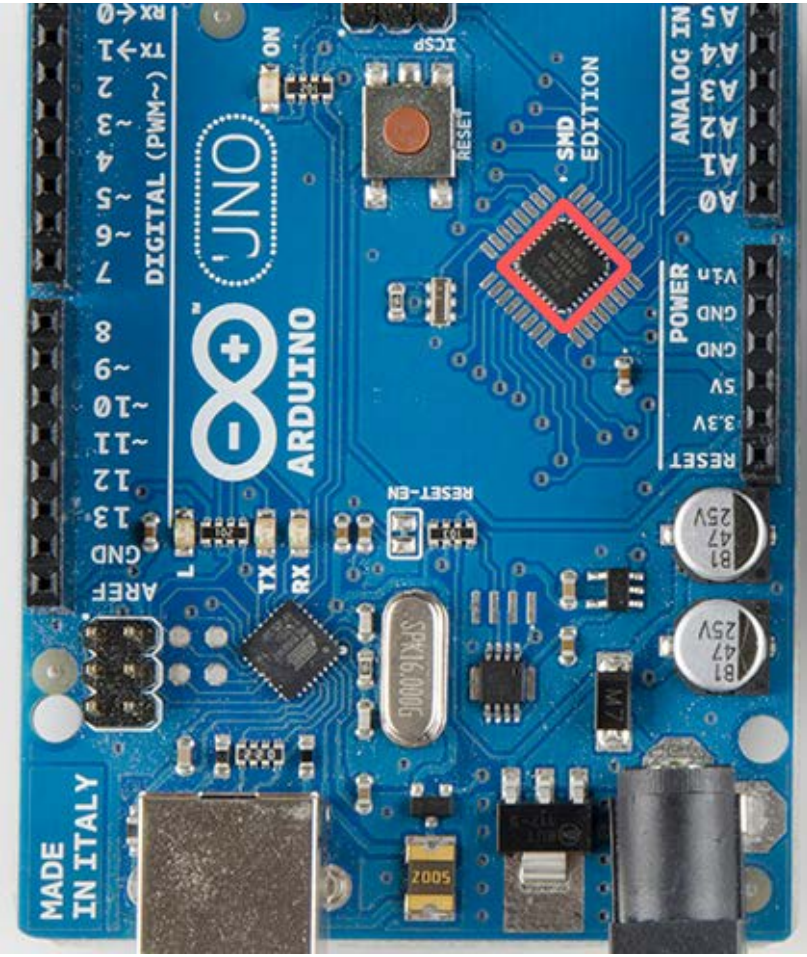


Mojo

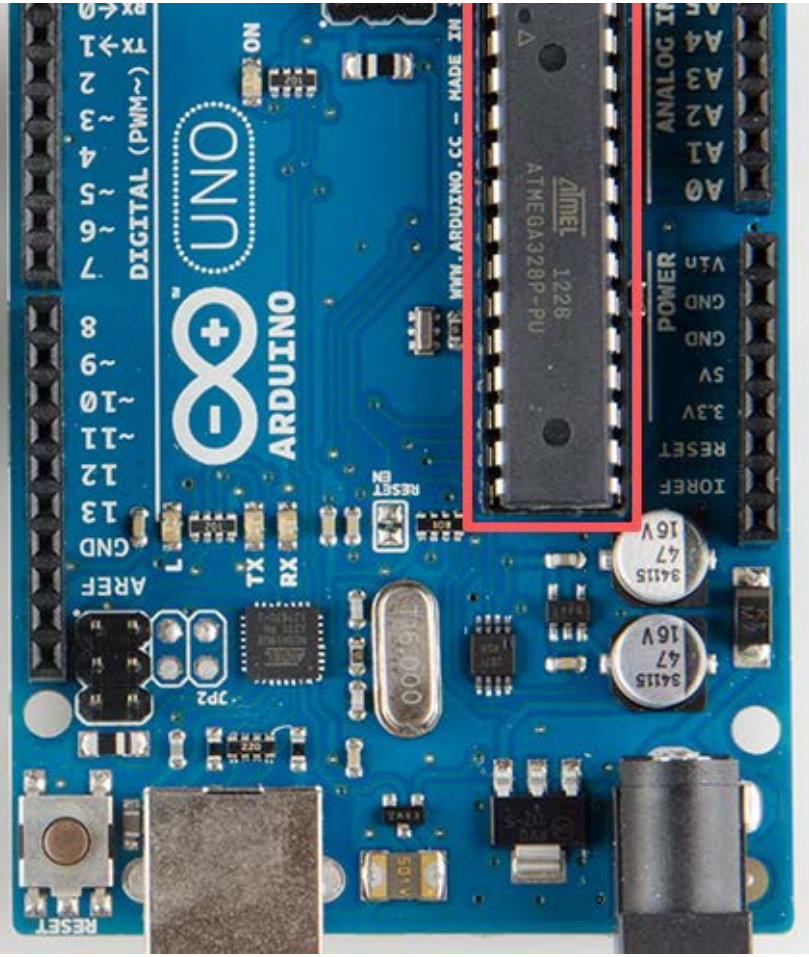
Arduino UNO Pin Layout



Types of chips: SMD vs. DIP



Surface mount device



Dual in-line package

Arduino IDE

Download the Arduino Software

ARDUINO 1.6.0

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board.
Refer to the Getting Started page for Installation instructions.

Windows Installer
Windows ZIP file for non admin install

Mac OS X for Java 6 (recommended)
Mac OS X for Java 7 (experimental)

Linux 32 bits
Linux 64 bits

Release Notes Source Code

ARDUINO 1.0.x / 1.5.x

PREVIOUS RELEASES

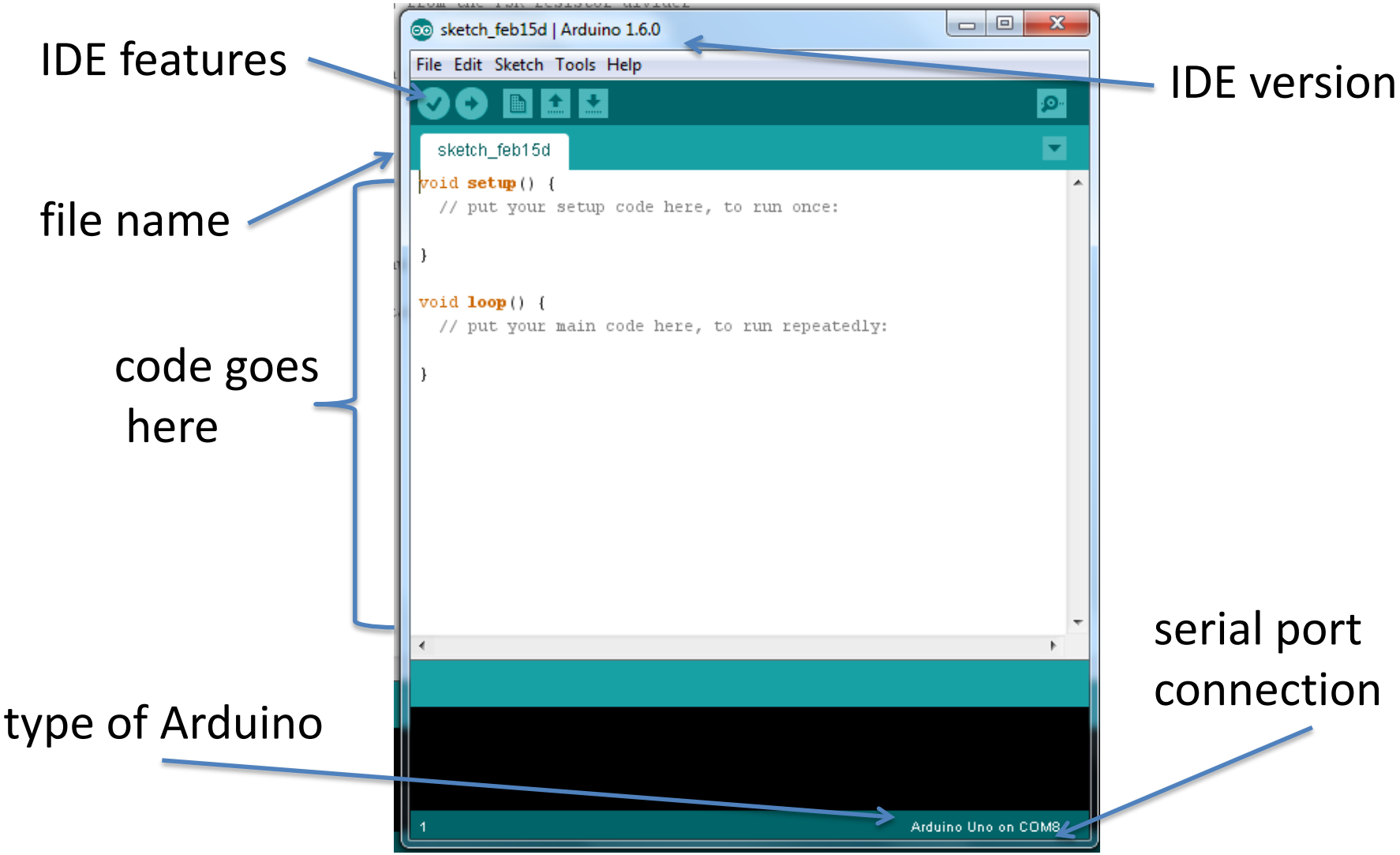
Download the Arduino 1.0.6 and all the previous versions of the Arduino Software. Available for Windows, Linux, and Mac OS X.

ARDUINO IDE

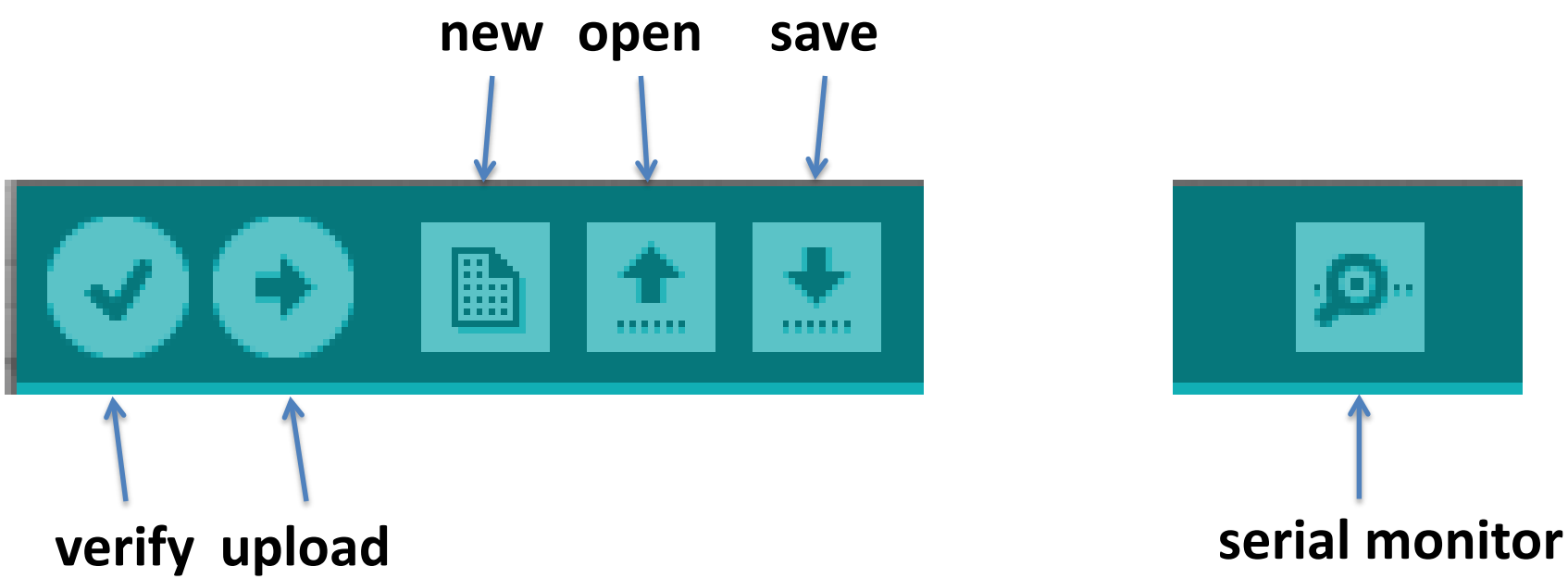
INTEL GALILEO AND EDISON

Download the Arduino IDE that supports the Intel Galileo and the Intel Edison boards. Available for Windows, Linux, and Mac OS X.

Programming Environment

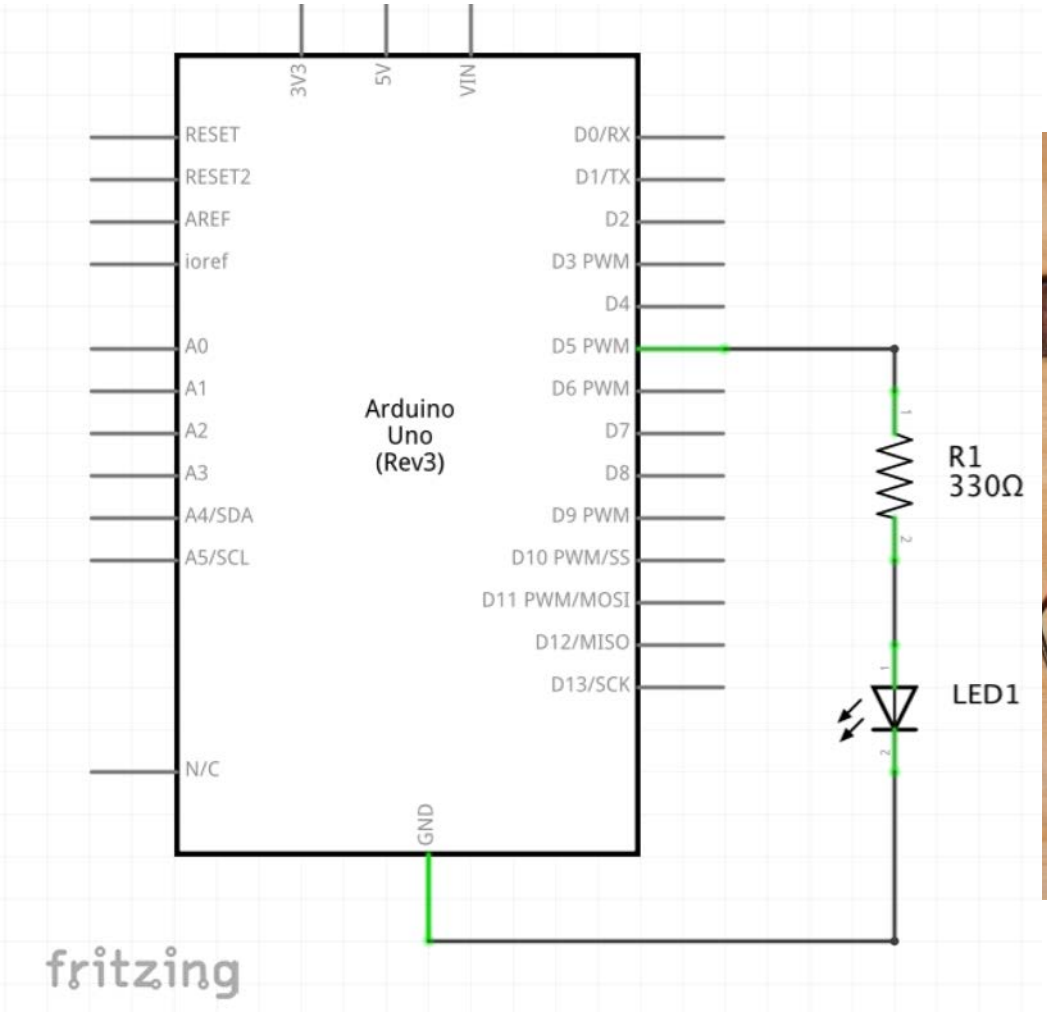


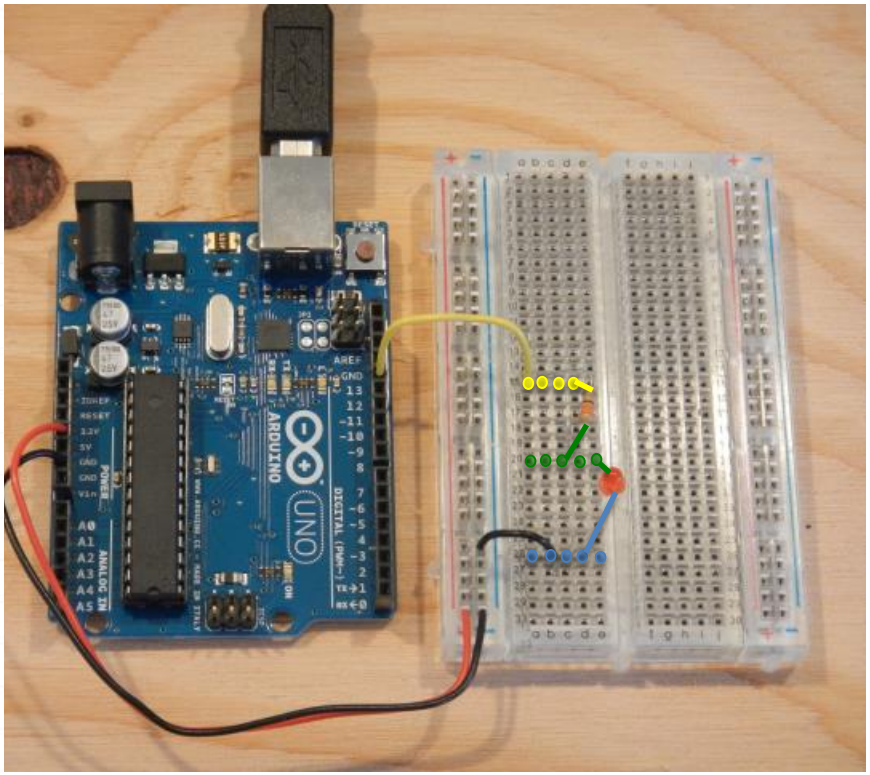
Arduino IDE Interface



Questions?

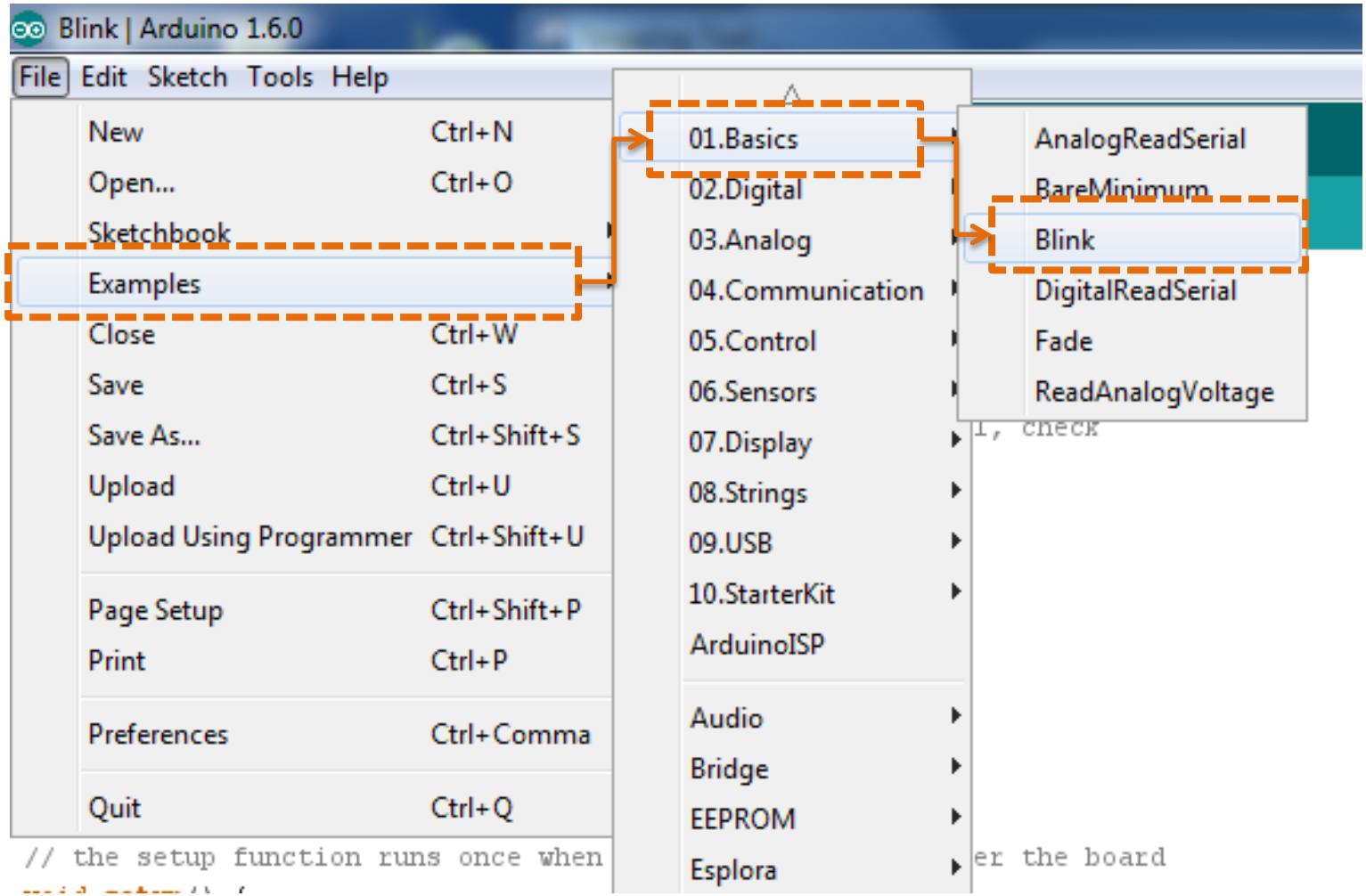
Project 4 – Blink





Place yellow wire in Pin 5, <u>NOT</u> Pin 13
(this image is incorrect)

File> Example> 01.Basics> Blink



Sketch layout

- **Add libraries, write comments, and declare variables**
  - Very first thing, add reference to library files
  - Set constants, define variables
  - Important, if you define a variable within a loop, it cannot be used outside of that loop
- **void setup()**
  - Declare pinmodes, set the pins for inputs and outputs
  - Setup serial monitor for communication
- **void loop()**
  - Runs your code
  - Starts at the top, ends at the bottom, then repeats
  - This is where you read sensors, compute values and control the outputs
  - The more stuff in the void loop, the longer it takes to cycle through the loop and get back to the start
- **Define functions**
  - Define functions to be used in the program (can also be at beginning).

Blink Sketch

Write comments

void setup()

void loop ()

Blink | Arduino 1.6.0

File Edit Sketch Tools Help

Upload

Blink §

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the Uno and
  Leonardo, it is attached to digital pin 13. If you're unsure what
  pin the on-board LED is connected to on your Arduino model, check
  the documentation at http://arduino.cc

  This example code is in the public domain.

 */

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

14

Edit the Blink sketch

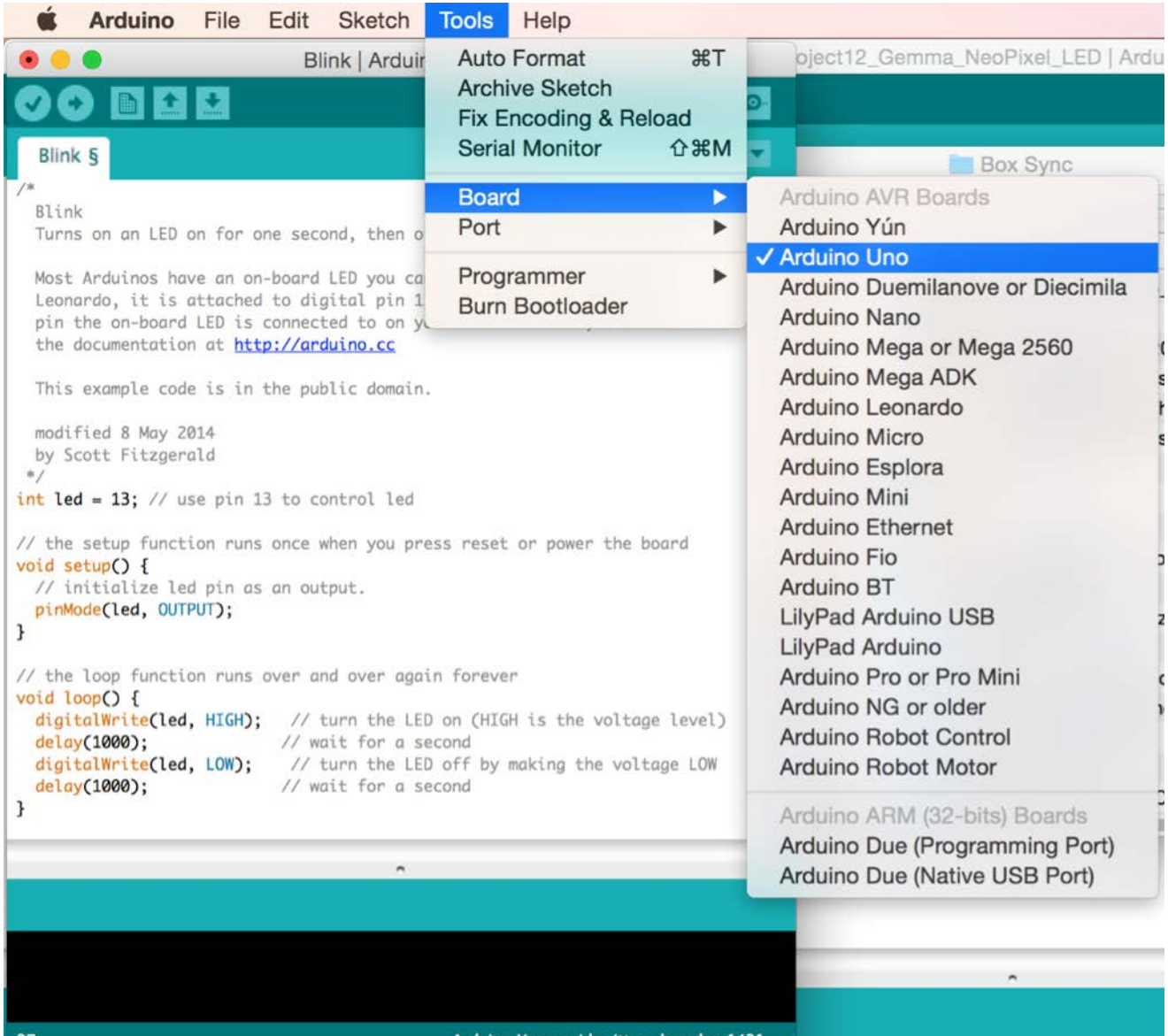
- The standard sketch uses Pin 13.
- At the factory, each Uno is tested with this sketch to make sure it works properly.
- An onboard LED will blink once you connect the USB cable.
- To test our circuit, we need to change every instance of 13 to 5 so the Arduino knows we plugged the LED into a Pin 5.

Edit the Blink Sketch

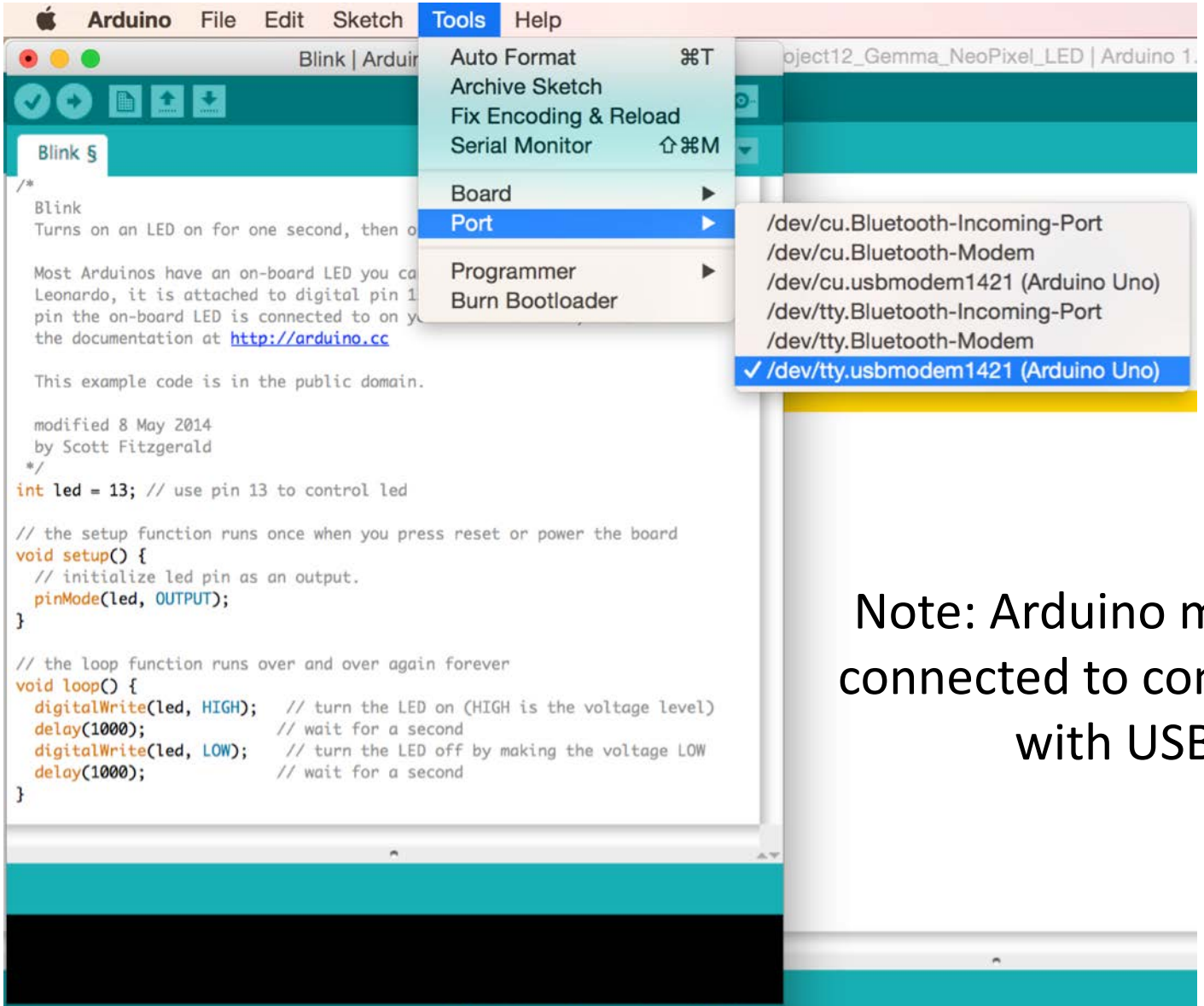
Blink §

```
  Most Arduinos have an on-board LED you can control. On the Uno and
  Leonardo, it is attached to digital pin 13. If you're unsure what
  pin the on-board LED is connected to on your Arduino model, check
  the documentation at http://arduino.cc

  This example code is in the public domain.

  modified 8 May 2014
  by Scott Fitzgerald
 */


// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 5 as an output.
  pinMode(5, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(5, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(5, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

Tools > Board > Arduino UNO

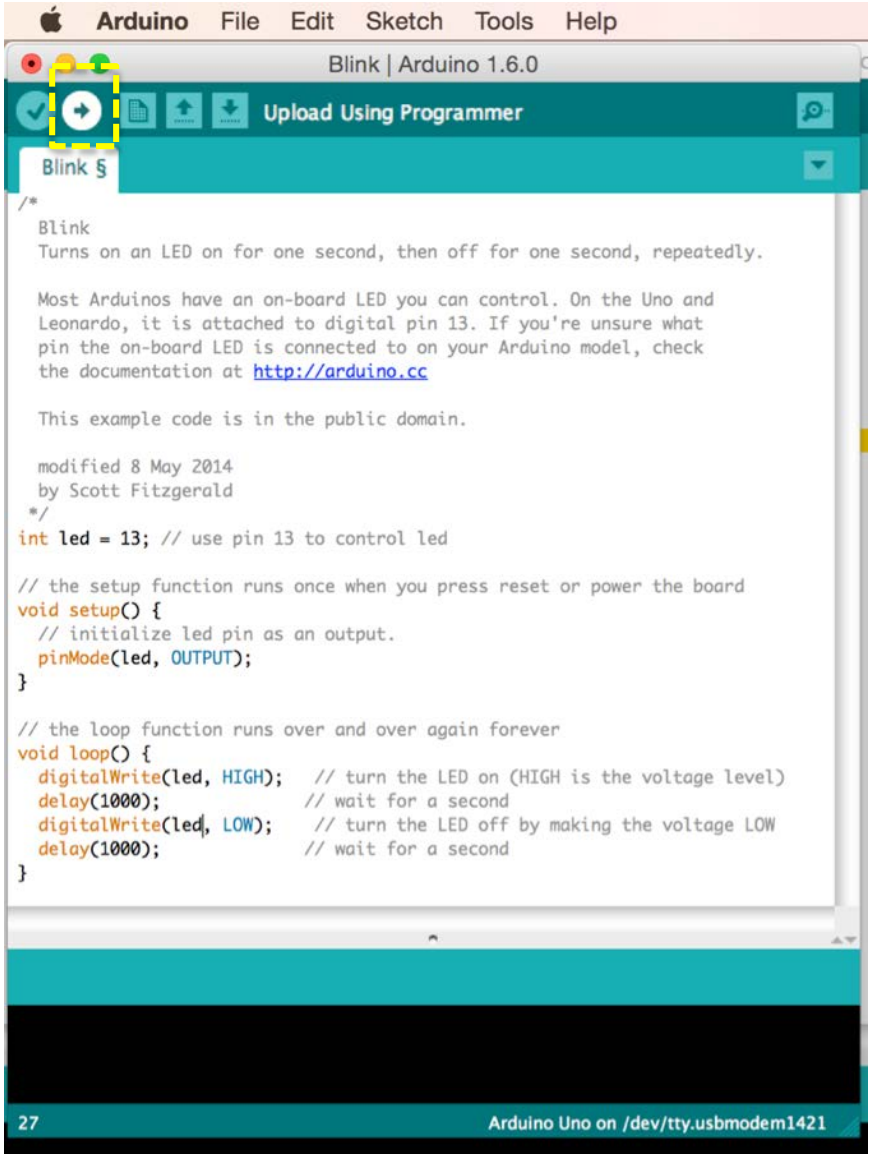


Tools > Port > usbmodem###



Note: Arduino must be connected to computer with USB cable.

Upload


Arduino File Edit Sketch Tools Help
Blink | Arduino 1.6.0
Upload Using Programmer
Blink §
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.
Most Arduinos have an on-board LED you can control. On the Uno and
Leonardo, it is attached to digital pin 13. If you're unsure what
pin the on-board LED is connected to on your Arduino model, check
the documentation at http://arduino.cc
This example code is in the public domain.
modified 8 May 2014
by Scott Fitzgerald
*/
int led = 13; // use pin 13 to control led
// the setup function runs once when you press reset or power the board
void setup() {
// initialize led pin as an output.
pinMode(led, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
delay(1000); // wait for a second
digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
delay(1000); // wait for a second
}
27
Arduino Uno on /dev/tty.usbmodem1421

The LED should Blink!

- If not, check the wiring to Pin 5
- Then check the code
- If you get an error message, make sure you change the port and board settings.

Now, make the code more flexible

Blink §

```
int ledPin = 9

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 9 as an output.
  pinMode(ledPin, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(ledPin, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                  // wait for a second
  digitalWrite(ledPin, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                  // wait for a second
}
```

But, how can we control the brightness of the LED?

Digital vs. Analog Outputs

- Digital – either ON or OFF
  - Binary, we use either 0 or 1, HIGH or LOW
    - 0 = false (off)
    - 1 = true (on)

- Analog – range of values
  - Hardware >>> analog to digital convertor (A/D)
  - Reads digital signal, discretizes it into range of values
  - UNO has 10 bit A/D, so values range 0-1023 (note: 2^{10} =1023)
  - Arduino can both read and send analog signals*

PWM with analogWrite()

Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on and off. This on-off pattern can simulate voltages in between full on (5 Volts) and off (0 Volts) by changing the portion of the time the signal spends on versus the time that the signal spends off. The duration of "on time" is called the pulse width. To get varying analog values, you change, or modulate, that pulse width. If you repeat this on-off pattern fast enough with an LED for example, the result is as if the signal is a steady voltage between 0 and 5v controlling the brightness of the LED.

Syntax

analogWrite(pin, value)

Parameters

pin: the pin to write to.

value: the duty cycle: between 0 (always off) and 255 (always on).

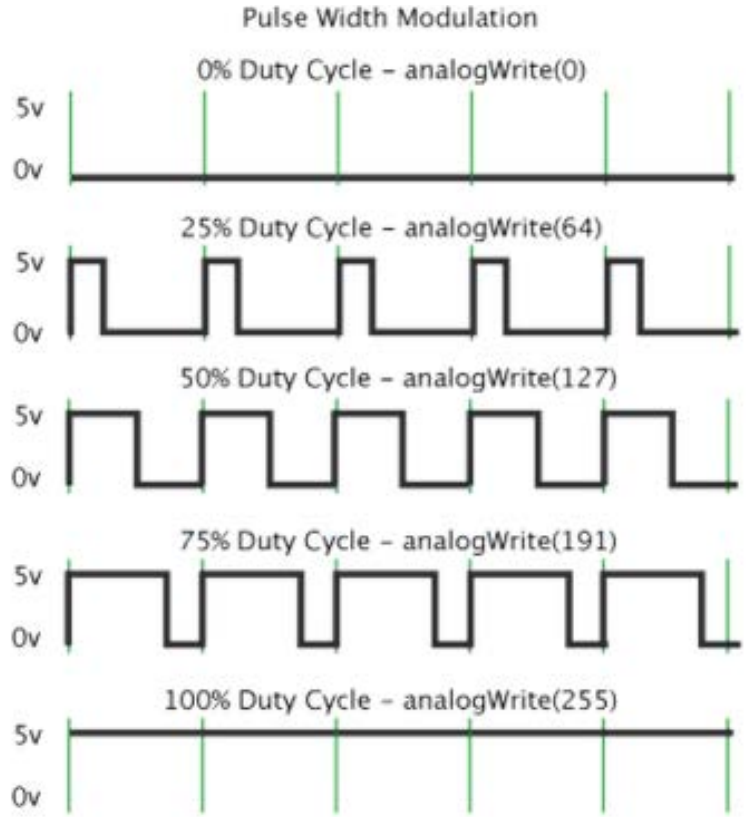
Returns

nothing

analogWrite() from scale 0-255

Arduino PWM

In the graphic below, the green lines represent a regular time period. This duration or period is the inverse of the PWM frequency. In other words, with Arduino's PWM frequency at about 500Hz, the green lines would measure 2 milliseconds each. A call to analogWrite() is on a scale of 0 - 255, such that analogWrite(255) requests a 100% duty cycle (always on), and analogWrite(127) is a 50% duty cycle (on half the time) for example.



File> Example> Basic> Fade

Fade | Arduino 1.6.0

File Edit Sketch Tools Help

Save

Fade

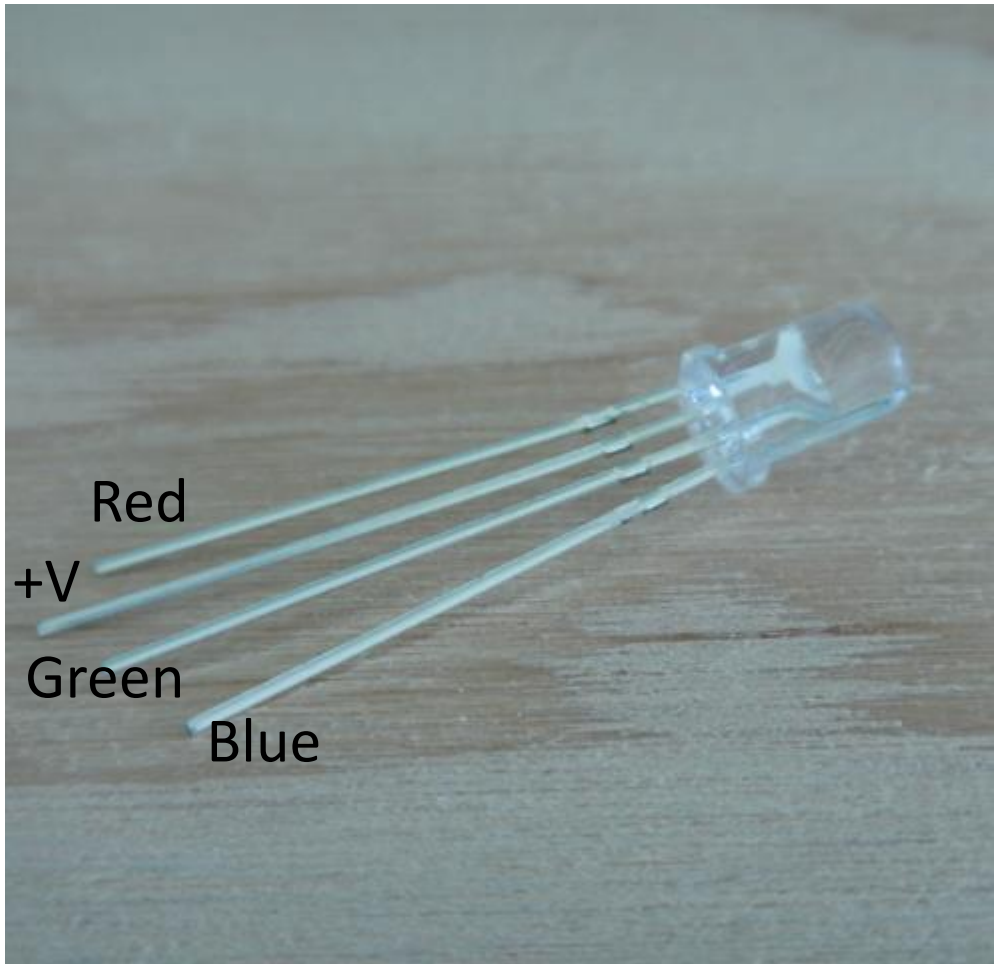
```
/*
 Fade

 This example shows how to fade an LED on pin 9
 using the analogWrite() function.

 This example code is in the public domain.
 */

int led = 9;           // the pin that the LED is attached to
int brightness = 0;    // how bright the LED is
int fadeAmount = 5;    // how many points to fade the LED by

// the setup routine runs once when you press reset:
void setup() {
  // declare pin 9 to be an output:
  pinMode(led, OUTPUT);
}
```

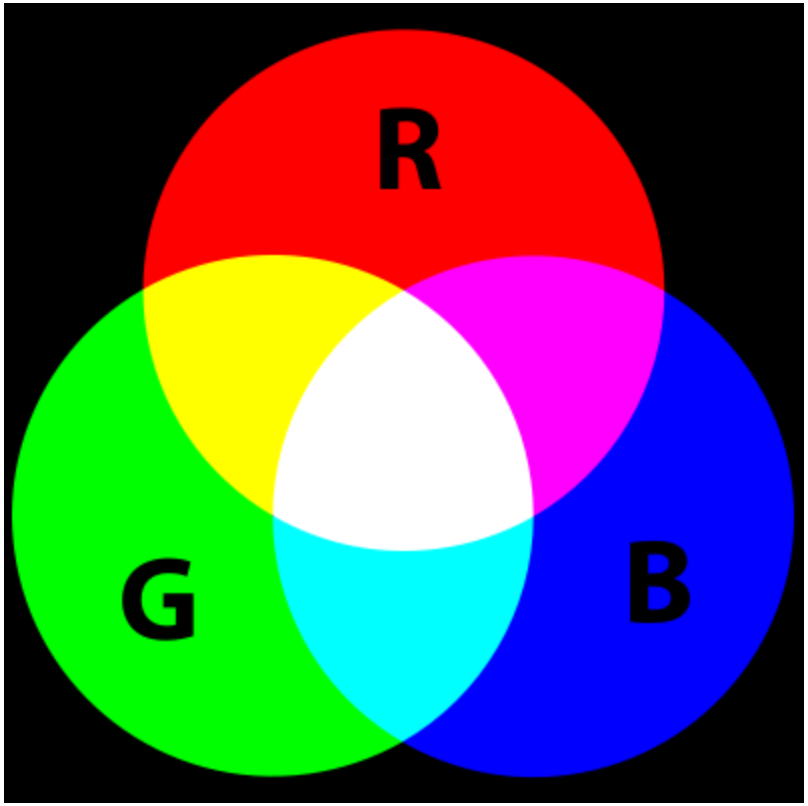
File> Example> Basic> Fade

```
// the loop routine runs over and over again forever:
void loop() {
  // set the brightness of pin 9:
  analogWrite(led, brightness);

  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade:
  if (brightness == 0 || brightness == 255) {
    fadeAmount = -fadeAmount ;
  }
  // wait for 30 milliseconds to see the dimming effect
  delay(30);
}
```

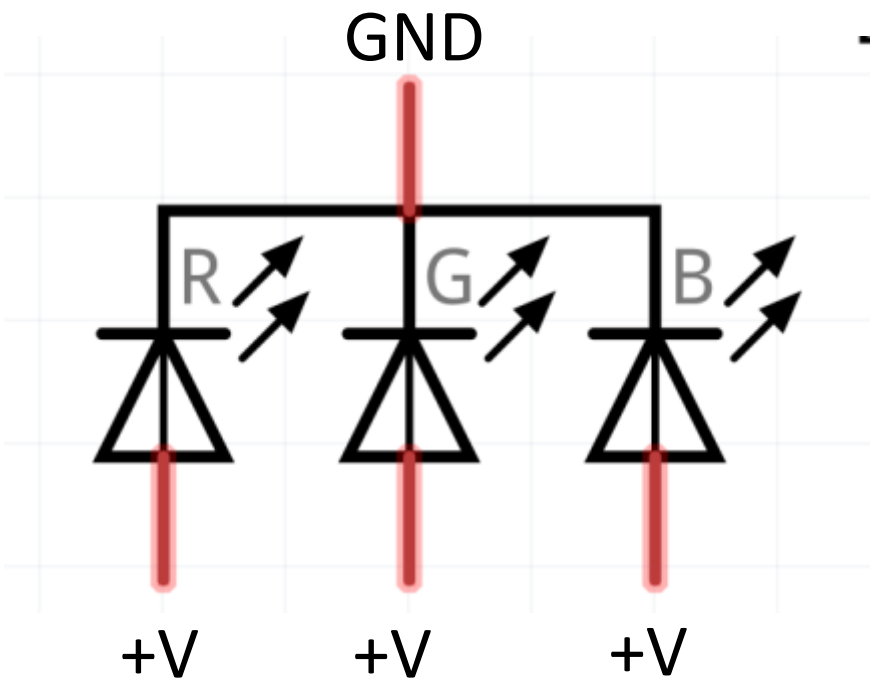
Questions?

Outputs - RGB LED

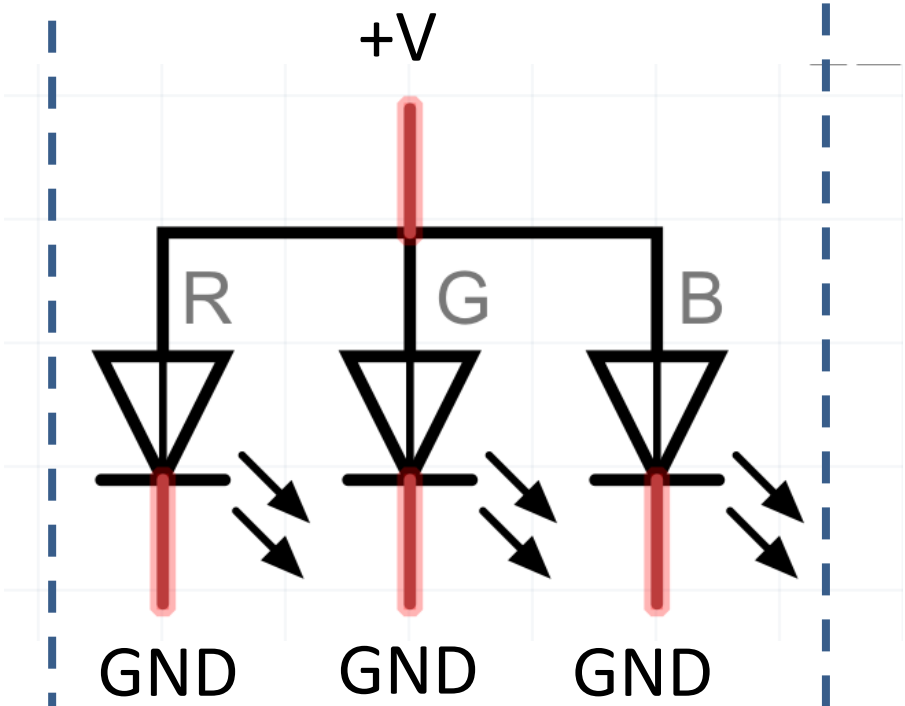




Types of RGB LED



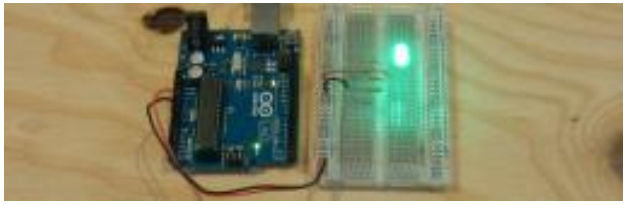
Common Cathode



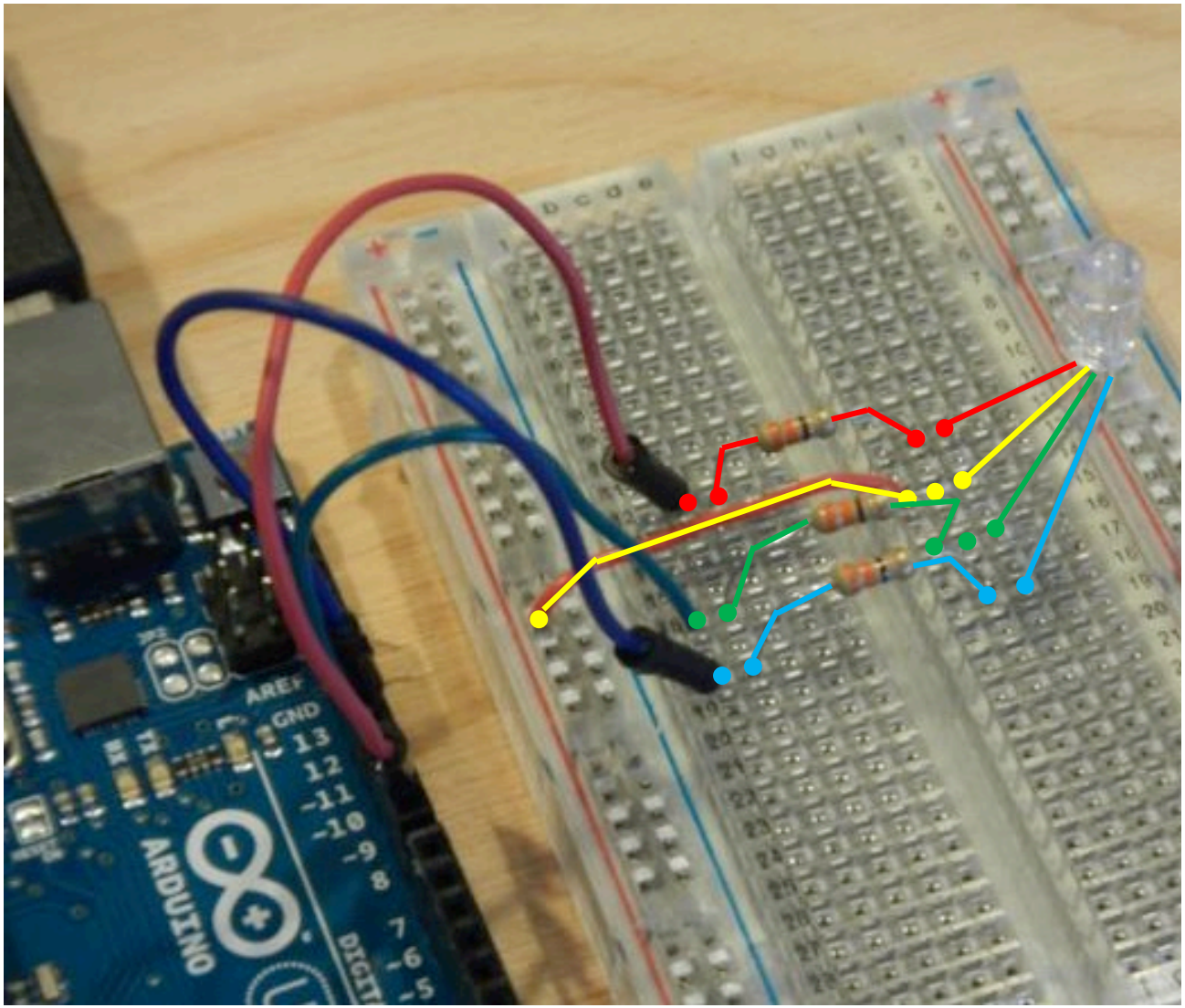
Common Anode

Standard RGB LED Colors

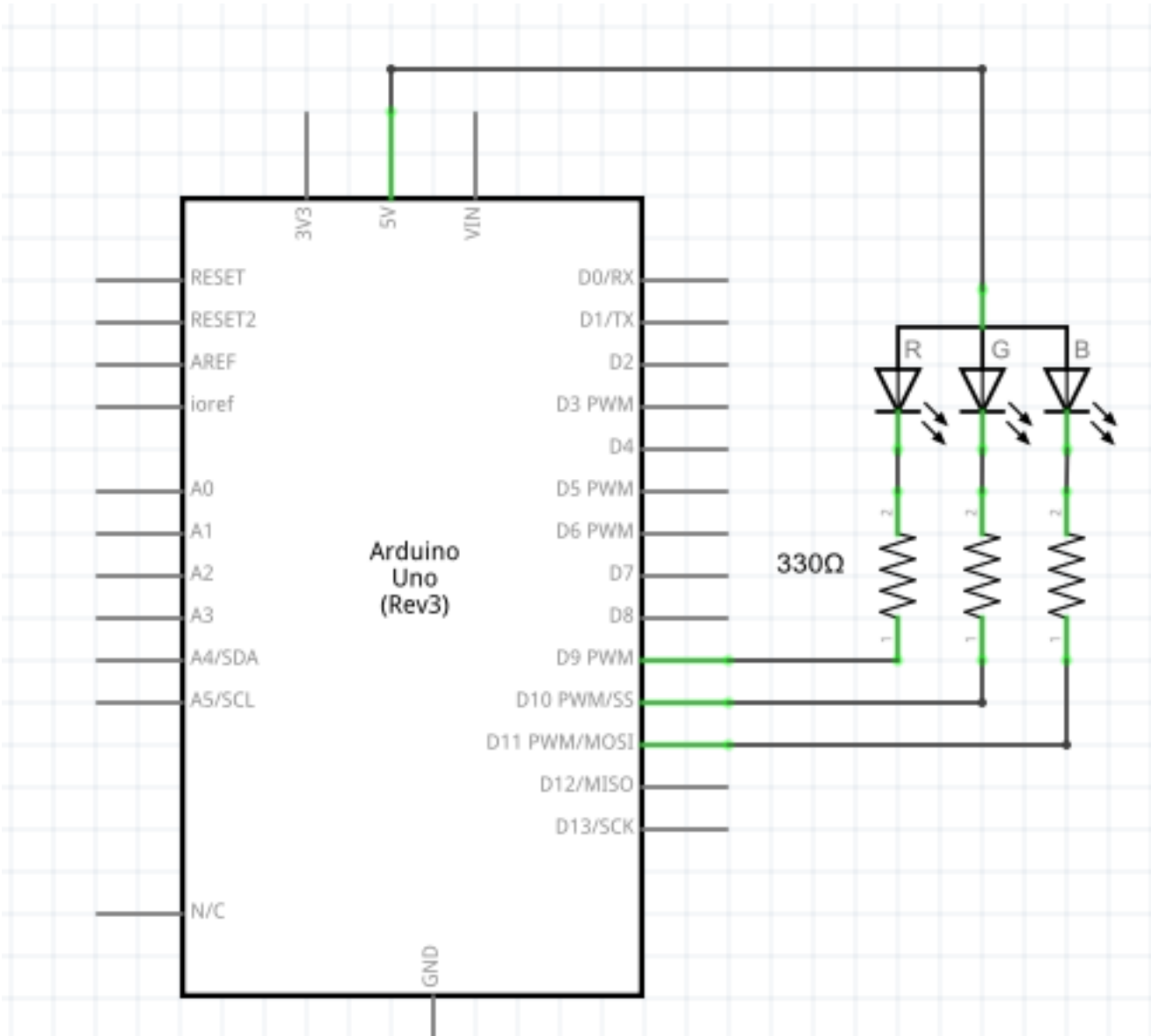
- Red
- Green
- Blue
- Red + Green
- Red + blue
- Green + Blue
- Red + Green + Blue



Project 5 – Color Changing RGB LED



Project 5 – Color Changing RGB LED



Sketch Layout

Define variables

void setup()

void loop()

Function definition

Project_05_RGB_LED | Arduino 1.0.5

Project_05_RGB_LED §

```
// sketch to change the color of a Common Anode RGB LED
int redPin = 9;
int greenPin = 10;
int bluePin = 11;

void setup(){
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
}

void loop(){
  setColor(0, 255, 255);  // red
  delay(1000);
  setColor(255, 0, 255);  // green
  delay(1000);
  setColor(255, 255, 0);  // blue
  delay(1000);
  setColor(0, 0, 255);  // yellow
  delay(1000);
  setColor(0, 255, 0);  // purple
  delay(1000);
  setColor(255, 0, 0);  // aqua
  delay(1000);
  setColor(255, 255, 255); //turn off
  delay(1000);
}

void setColor(int red, int green, int blue){
  analogWrite(redPin, red);
  analogWrite(greenPin, green);
  analogWrite(bluePin, blue);
}
```

Project_05_RGB_LED

Project_05_RGB_LED | Arduino 1.0.5

Project_05_RGB_LED §

```
// sketch to change the color of a Common Anode RGB LED
int redPin = 9;
int greenPin = 10;
int bluePin = 11;

void setup(){
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
}
```

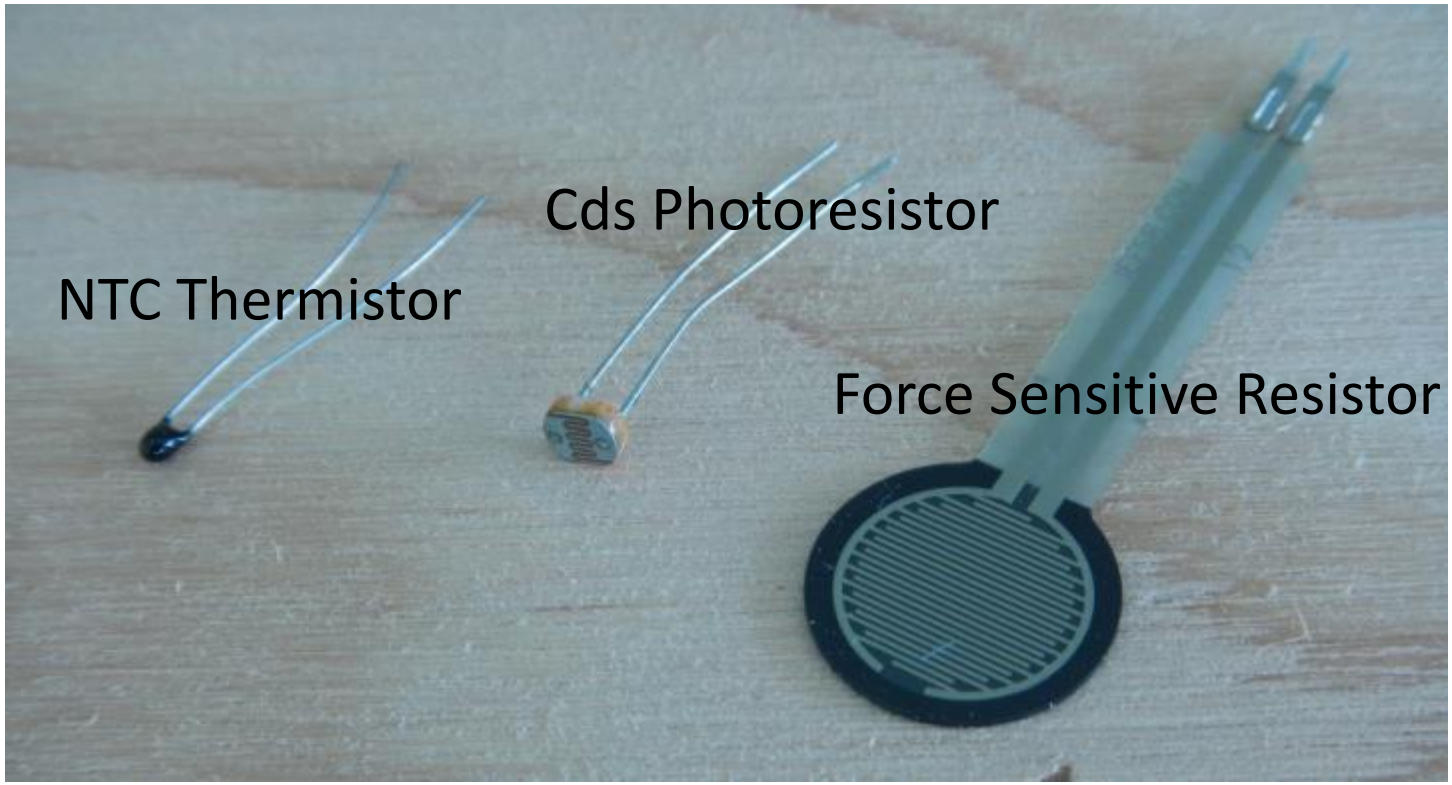
Project_05_RGB_LED

```
void loop(){
  setColor(0, 255, 255);  // red
  delay(1000);
  setColor(255, 0, 255);  // green
  delay(1000);
  setColor(255, 255, 0);  // blue
  delay(1000);
  setColor(0, 0, 255);  // yellow
  delay(1000);
  setColor(0, 255, 0);  // purple
  delay(1000);
  setColor(255, 0, 0);  // aqua
  delay(1000);
  setColor(255, 255, 255); //turn off
  delay(1000);
}

void setColor(int red, int green, int blue){
  analogWrite(redPin, red);
  analogWrite(greenPin, green);
  analogWrite(bluePin, blue);
}
```

- Plug in the Arduino, upload the code and the RGB LED should start to change colors.
- If not, check the wiring and double check the value of the resistors.
- They should be 330 ohm (orange, orange, brown, gold)

Questions?

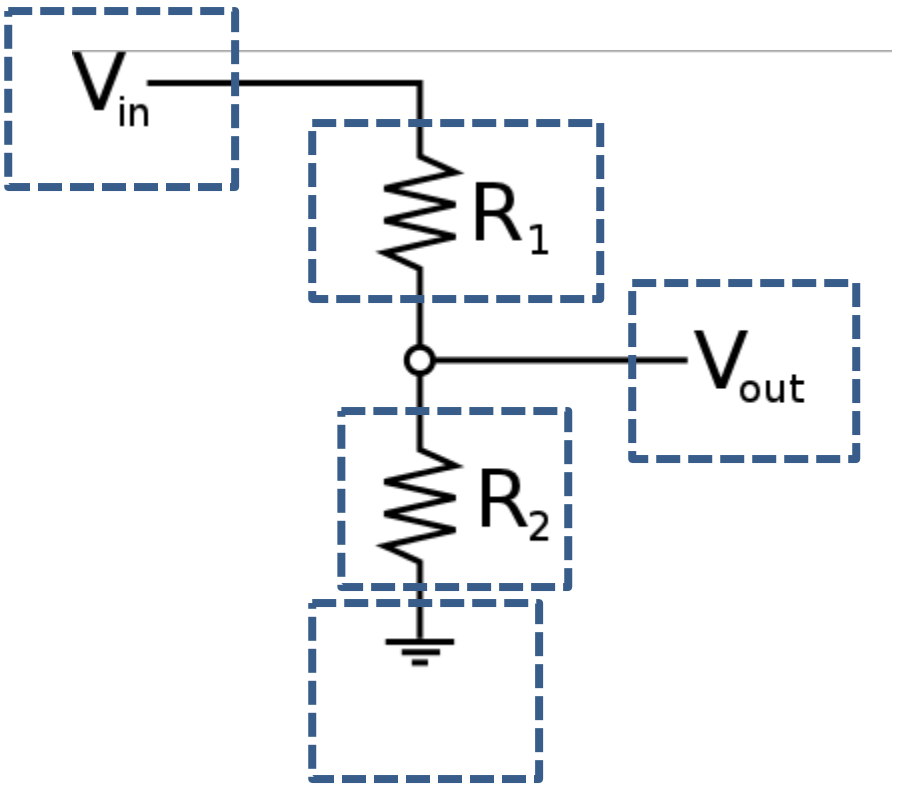
Resistive Sensors



- **NTCT Thermistor** - NTC-type thermistors decrease in resistance as temperature rises.
- **Photoresistor** – When exposed to more light, the resistance goes down.
- **Force Sensitive Resistor** – Resistance decrease when force is applied.

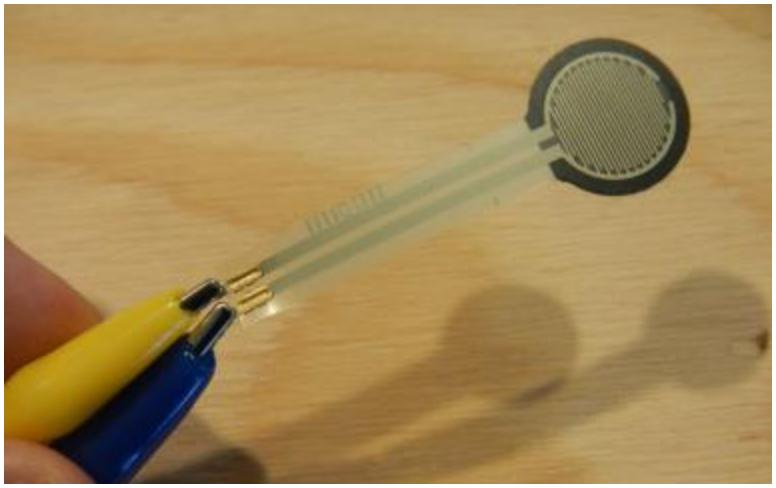
Voltage Divider

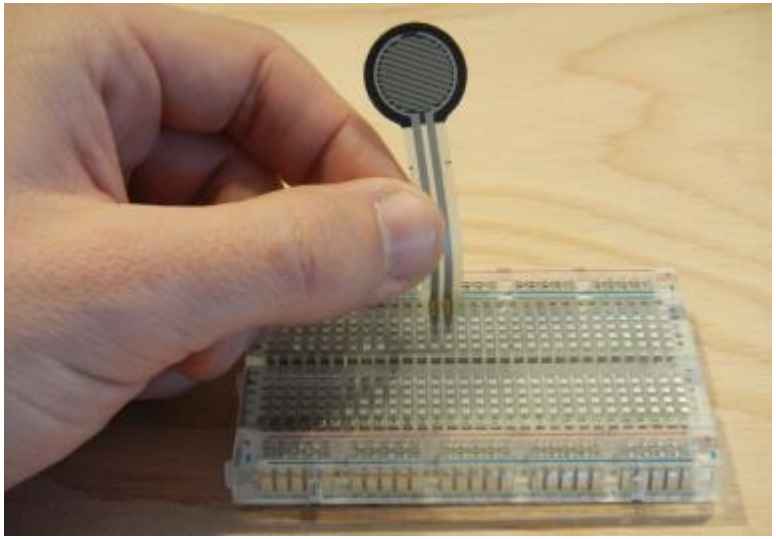
- Arduino does not have a resistance meter,
- Must convert change in resistance into chance in voltage.
- Read with A/D.



$$V_{\text{out}} = \frac{R_2}{R_1 + R_2} \cdot V_{\text{in}}$$

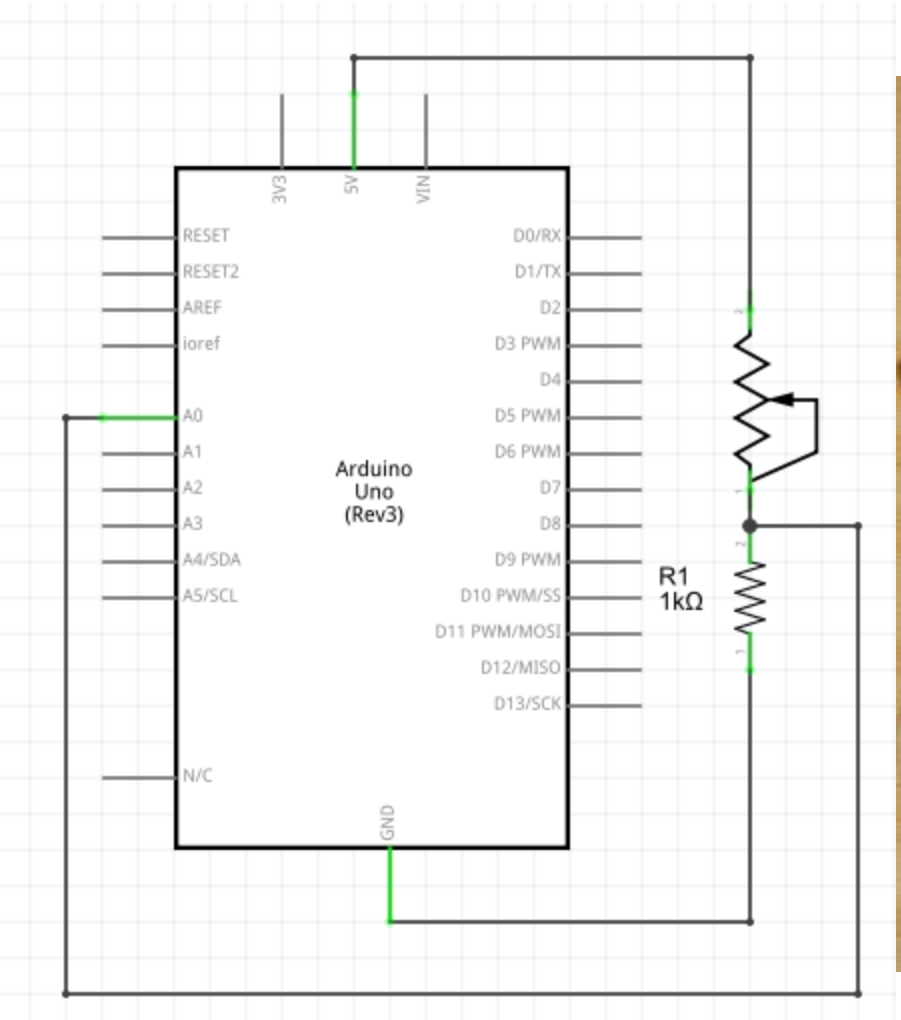
Force Sensitive Resistors

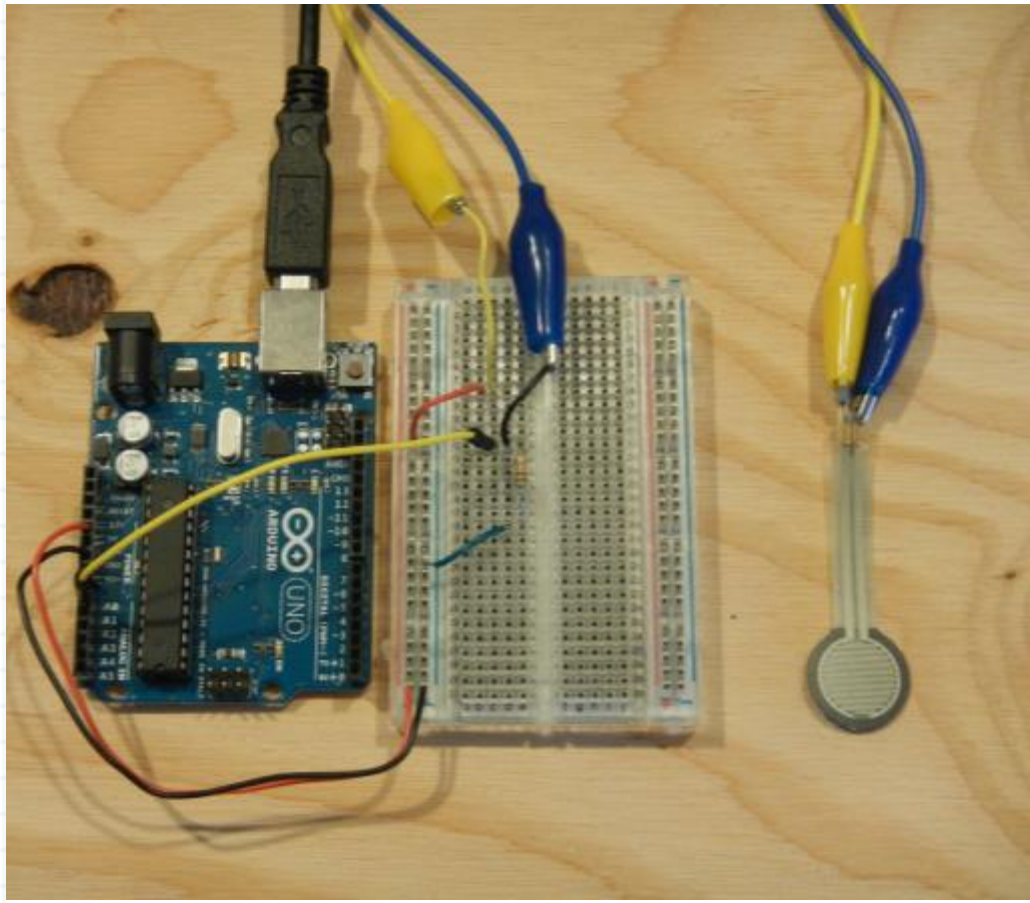




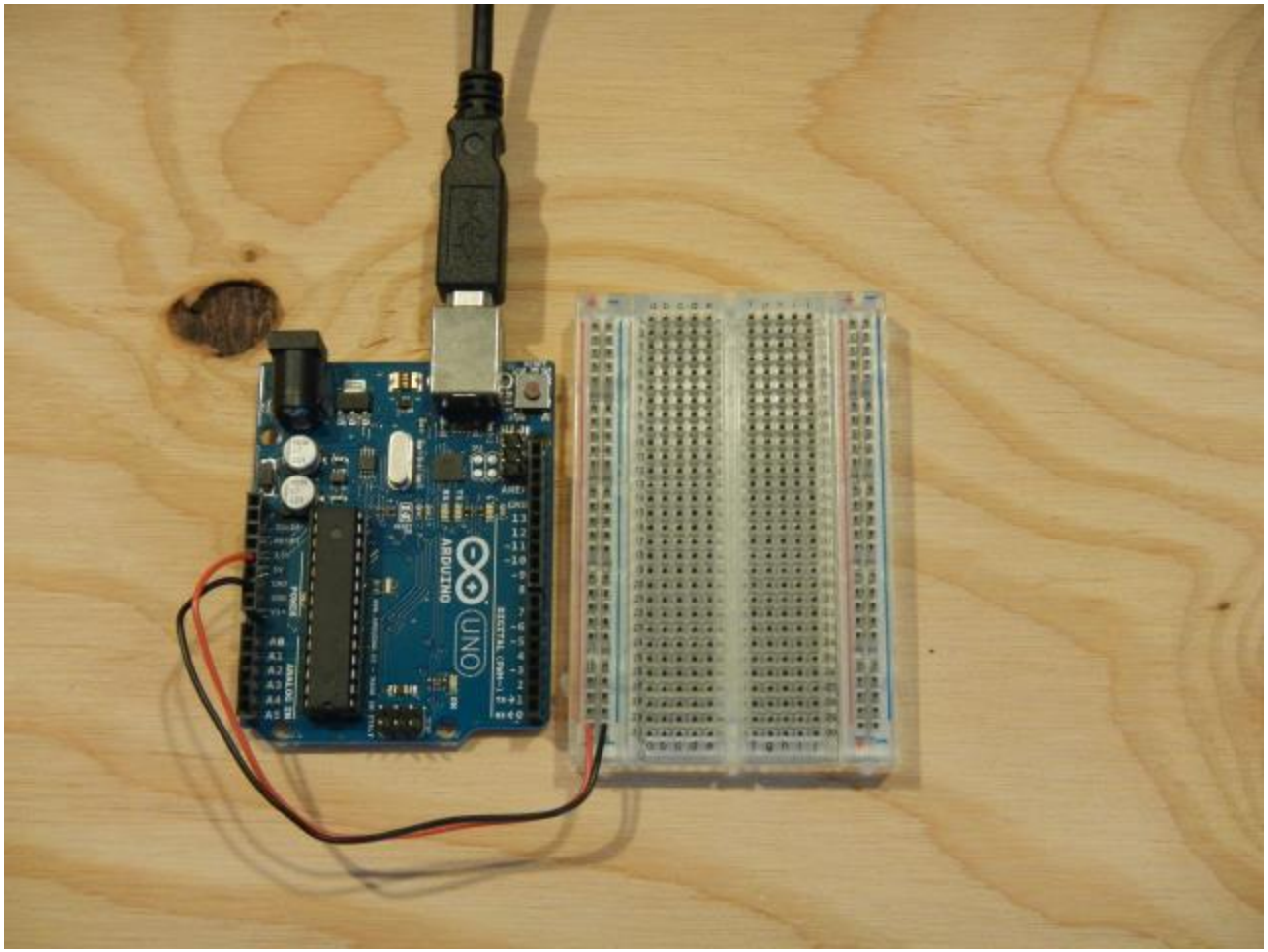
Force (lb)	Force (N)	FSR Resistance	(FSR + R) ohm	Current thru FSR+R	Voltage across R
None	**None**	Infinite	Infinite!	0 mA	0V
0.04 lb	**0.2 N**	30 Kohm	40 Kohm	0.13 mA	1.3 V
0.22 lb	**1 N**	6 Kohm	16 Kohm	0.31 mA	3.1 V
2.2 lb	**10 N**	1 Kohm	11 Kohm	0.45 mA	4.5 V
22 lb	**100 N**	250 ohm	10.25 Kohm	0.49 mA	4.9 V

Build the Circuit

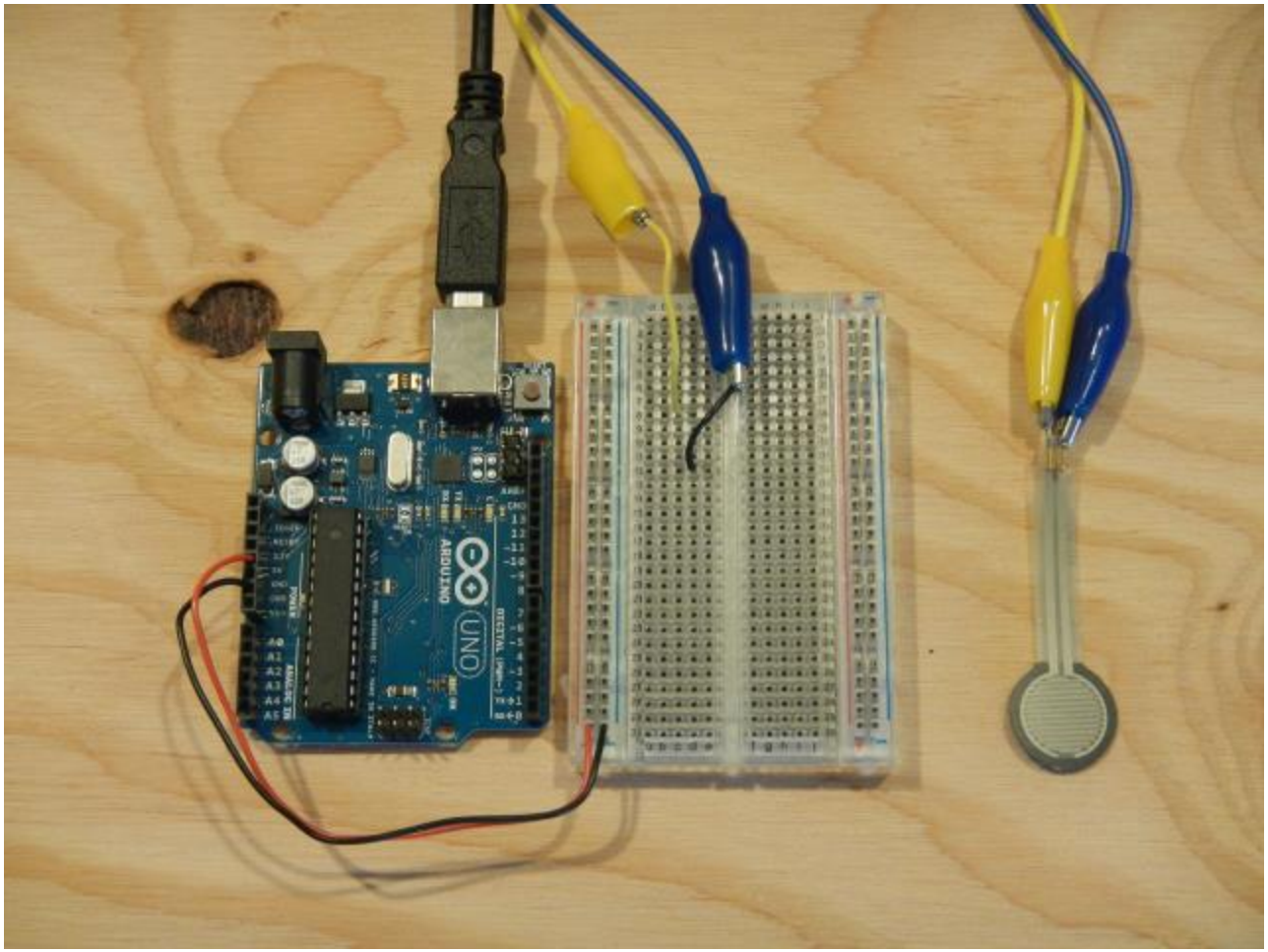




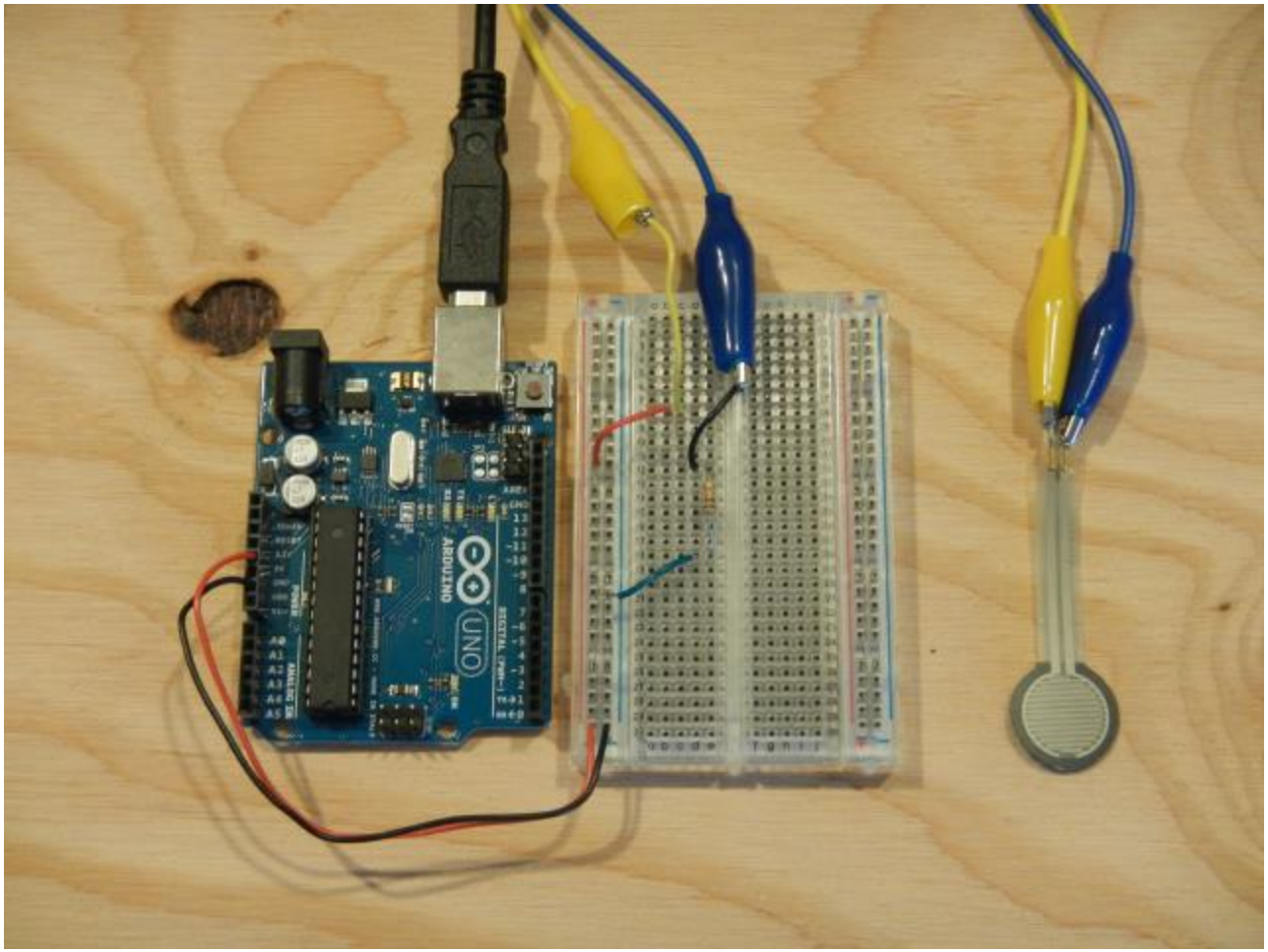
Step 1



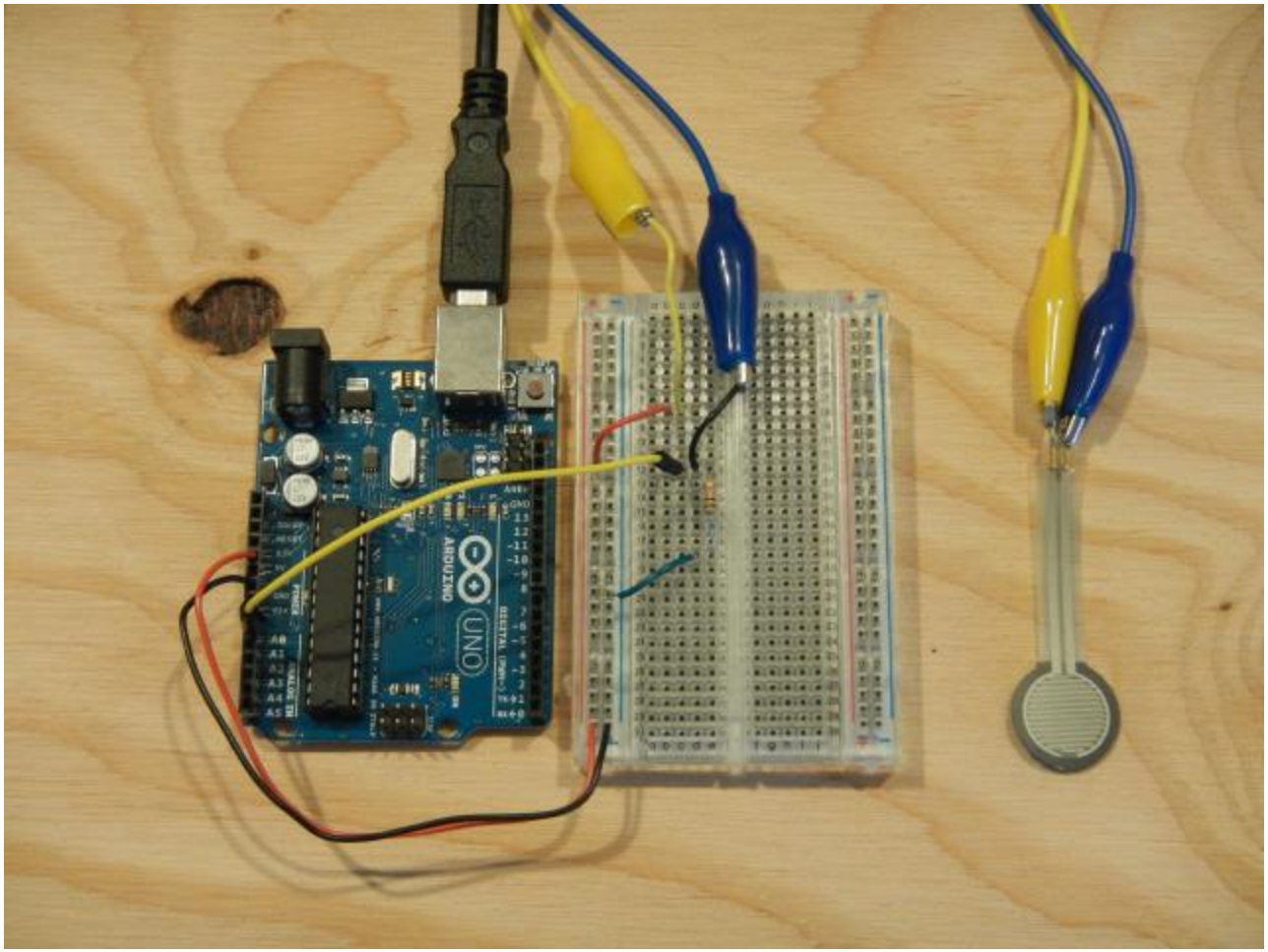
Step 2



Step 3



Step 4



Project 6 – Force Sensitive Resistor

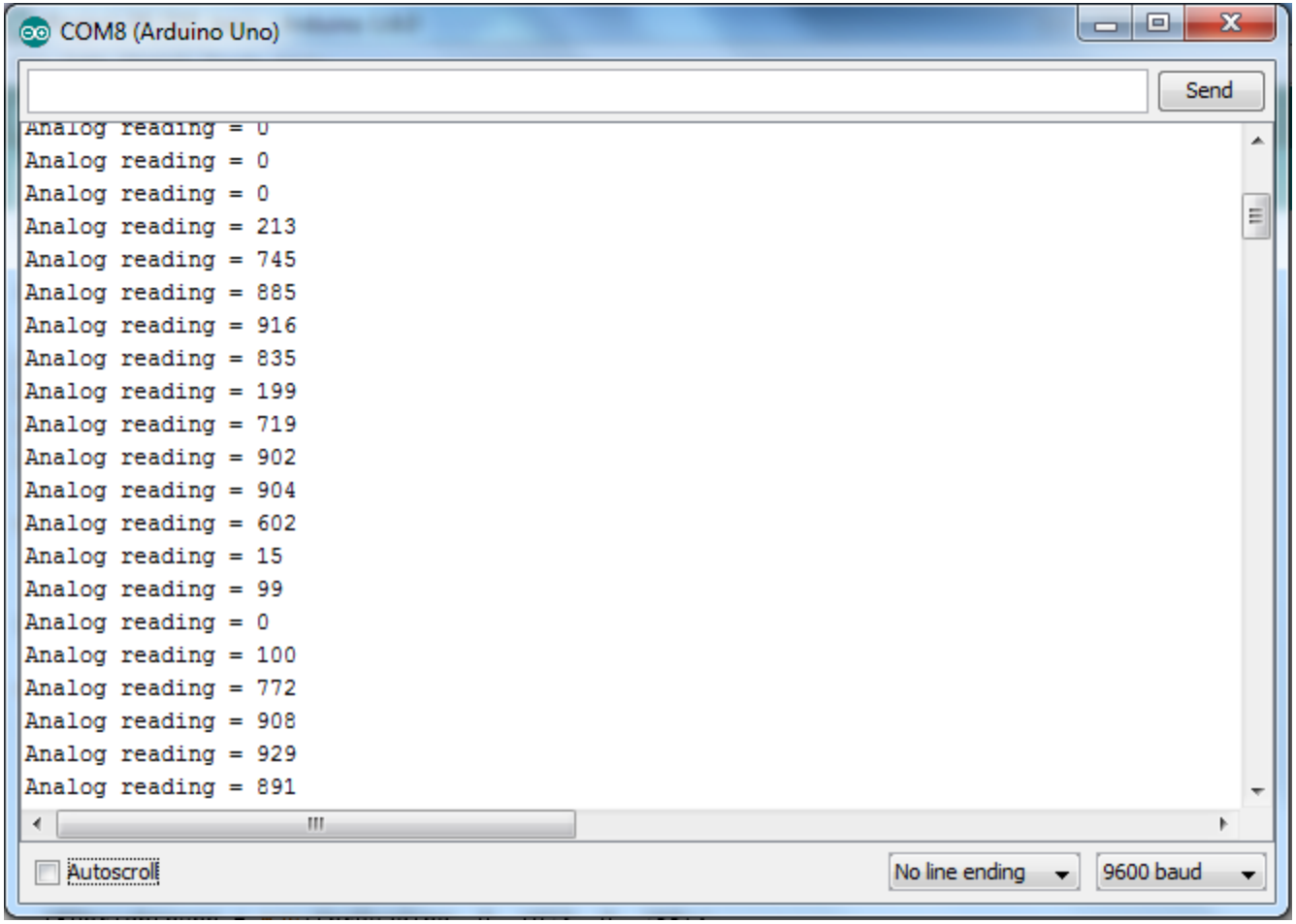
Project_06_FSR_basic

```
/* FSR testing sketch.

Connect one end of FSR to 5V, the other end to Analog 0.
Then connect one end of a 10K resistor from Analog 0 to ground
Connect LED from pin 11 through a resistor to ground

For more information see www.ladyada.net/learn/sensors/fsr.html */

int fsrAnalogPin = 0; // FSR is connected to analog 0
int LEDpin = 11;      // connect Red LED to pin 11 (PWM pin)
int fsrReading;      // the analog reading from the FSR resistor divider
int LEDbrightness;

void setup(void) {
  Serial.begin(9600);   // We'll send debugging information via the Serial monitor
  pinMode(LEDpin, OUTPUT);
}
```

Project 6 – Force Sensitive Resistor

```
void loop(void) {
  fsrReading = analogRead(fsrAnalogPin);
  Serial.print("Analog reading = ");
  Serial.println(fsrReading);

  // we'll need to change the range from the analog reading (0-1023) down to the range
  // used by analogWrite (0-255) with map!
  LEDbrightness = map(fsrReading, 0, 1023, 0, 255);
  // LED gets brighter the harder you press
  analogWrite(LEDpin, LEDbrightness);

  delay(100);
}
```

Sensor Readings



Project 06 - Squeeze Gauge

FRS_squeeze_gauge | Arduino 1.6.0

File Edit Sketch Tools Help

Verify

FRS_squeeze_gauge

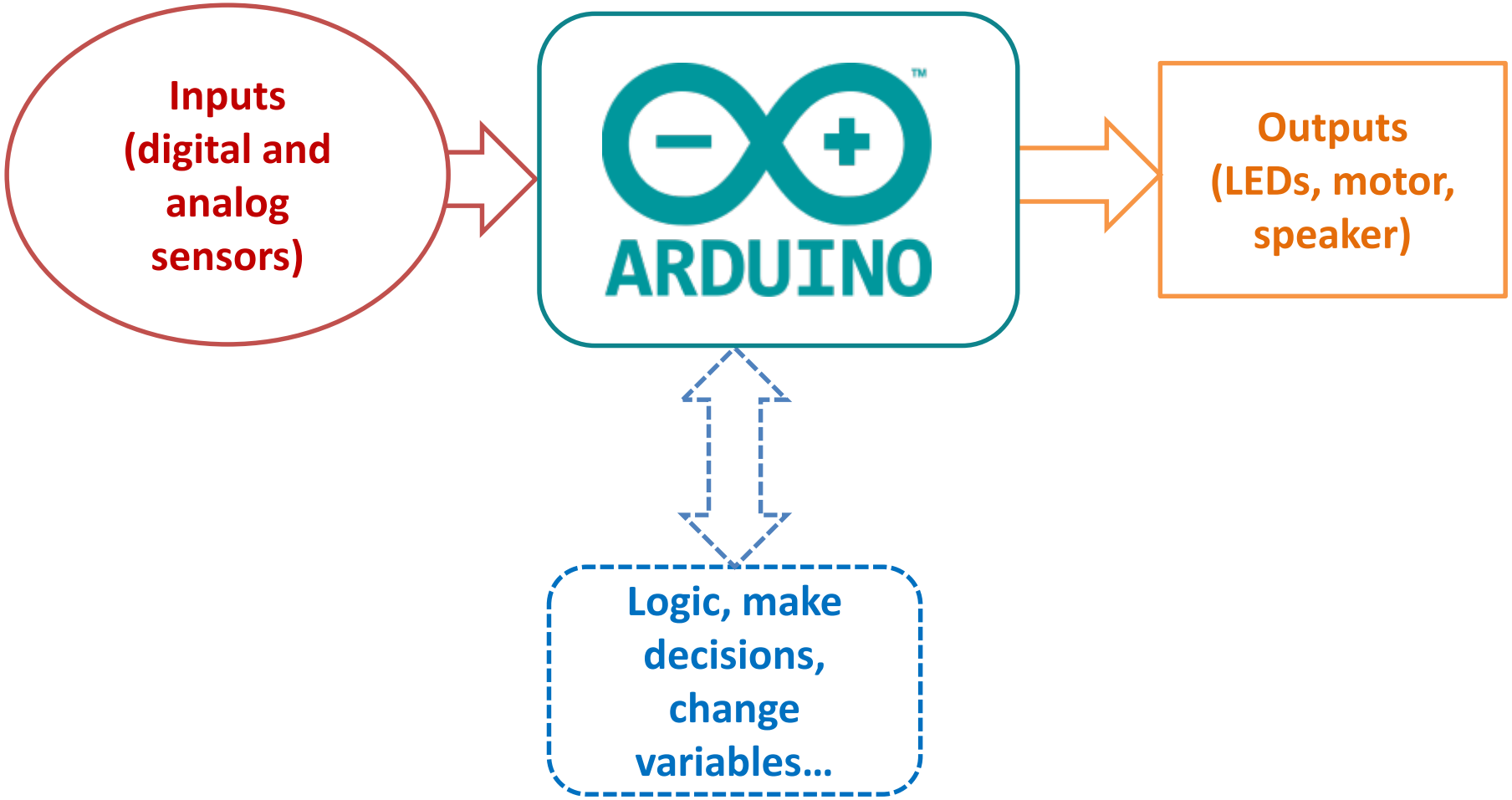
```
/* FSR simple testing sketch.

Connect one end of FSR to power, the other end to Analog 0.
Then connect one end of a 10K resistor from Analog 0 to ground

For more information see www.ladyada.net/learn/sensors/fsr.html */

int fsrPin = 0;     // the FSR and 10K pulldown are connected to a0
int fsrReading;     // the analog reading from the FSR resistor divider

void setup(void) {
  // We'll send debugging information via the Serial monitor
  Serial.begin(9600);
}
```

Project 06 – Squeeze Gauge

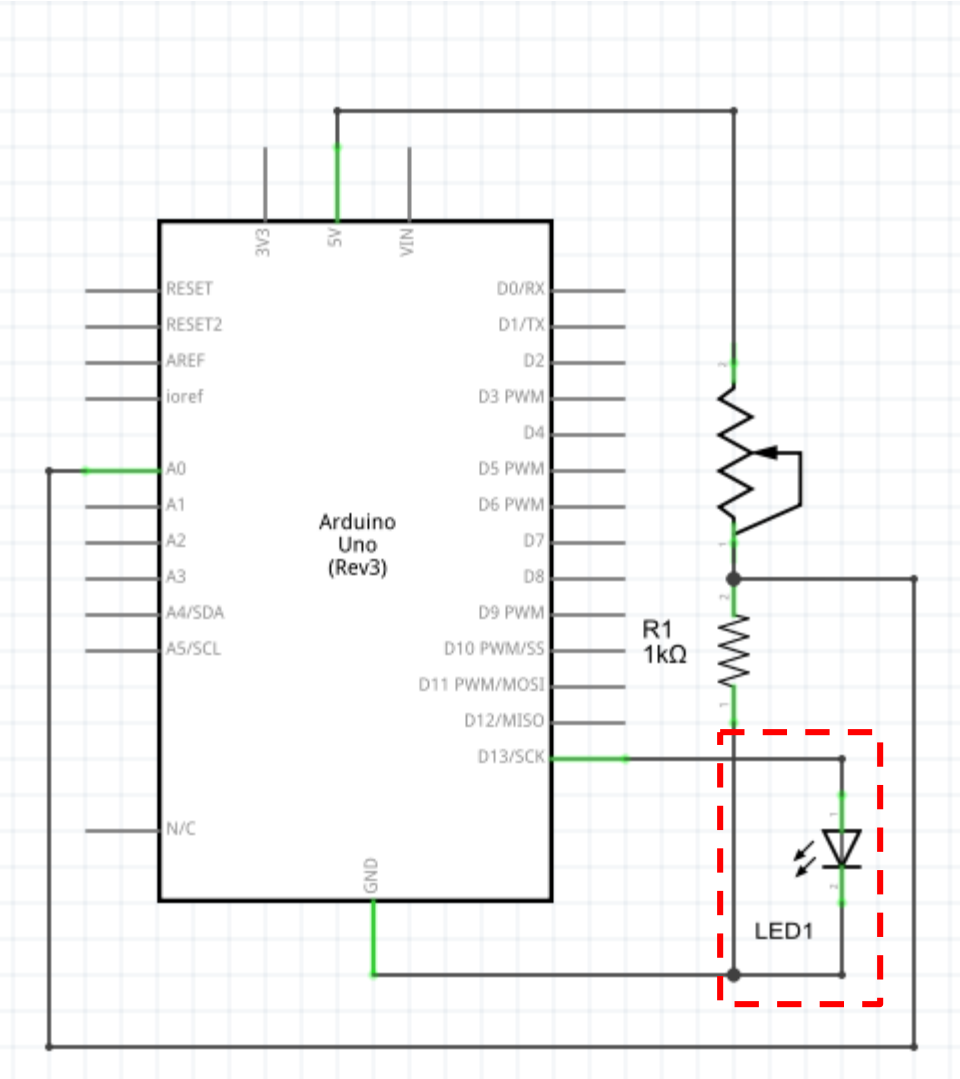
```
void loop(void) {
  fsrReading = analogRead(fsrPin);

  Serial.print("Analog reading = ");
  Serial.print(fsrReading);     // the raw analog reading

  // We'll have a few threshholds, qualitatively determined
  if (fsrReading < 10) {
    Serial.println(" - No pressure");
  } else if (fsrReading < 200) {
    Serial.println(" - Light touch");
  } else if (fsrReading < 500) {
    Serial.println(" - Light squeeze");
  } else if (fsrReading < 800) {
    Serial.println(" - Medium squeeze");
  } else {
    Serial.println(" - Big squeeze");
  }
  delay(1000);
}
```

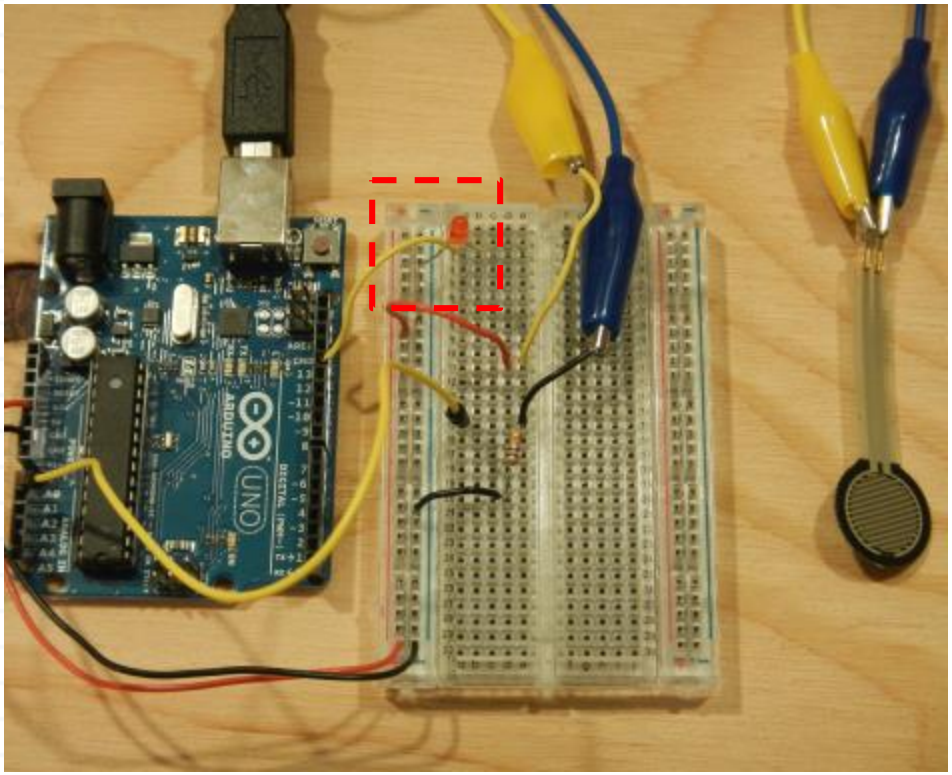
Questions?

Inputs and Outputs

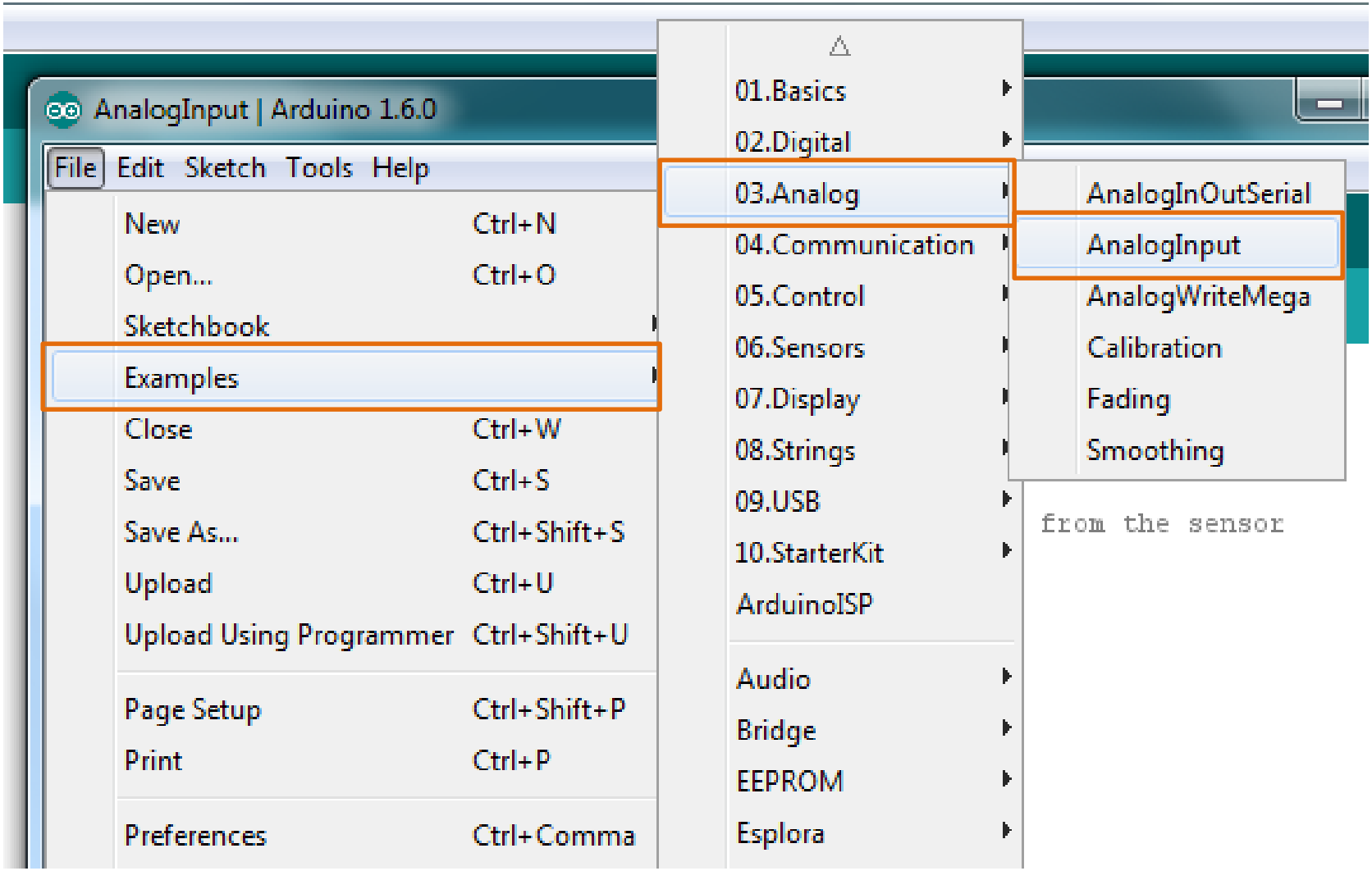


Build the Circuit





Project 7 – AnalogInput



Project 7 – AnalogInput Code

AnalogInput | Arduino 1.6.0

File Edit Sketch Tools Help

New

AnalogInput

```
turning on and off a light emitting diode(LED)  connected to digital pin 13.
The amount of time the LED will be on and off depends on
the value obtained by analogRead().

The circuit:
* Potentiometer attached to analog input 0
* center pin of the potentiometer to the analog pin
* one side pin (either one) to ground
* the other side pin to +5V
* LED anode (long leg) attached to digital output 13
* LED cathode (short leg) attached to ground

* Note: because most Arduinos have a built-in LED attached
to pin 13 on the board, the LED is optional.


Created by David Cuartielles
modified 30 Aug 2011
By Tom Igoe

This example code is in the public domain.

http://arduino.cc/en/Tutorial/AnalogInput

*/
```

Project 7 – AnalogInput Code

```
int sensorPin = A0;    // select the input pin for the potentiometer
int ledPin = 13;      // select the pin for the LED
int sensorValue = 0;  // variable to store the value coming from the sensor
```

```
void setup() {
  // declare the ledPin as an OUTPUT:
  pinMode(ledPin, OUTPUT);
}
```

```
void loop() {
  // read the value from the sensor:
  sensorValue = analogRead(sensorPin);
  // turn the ledPin on
  digitalWrite(ledPin, HIGH);
  // stop the program for <sensorValue> milliseconds:
  delay(sensorValue);
  // turn the ledPin off:
  digitalWrite(ledPin, LOW);
  // stop the program for for <sensorValue> milliseconds:
  delay(sensorValue);
}
```

Done uploading.

Questions?