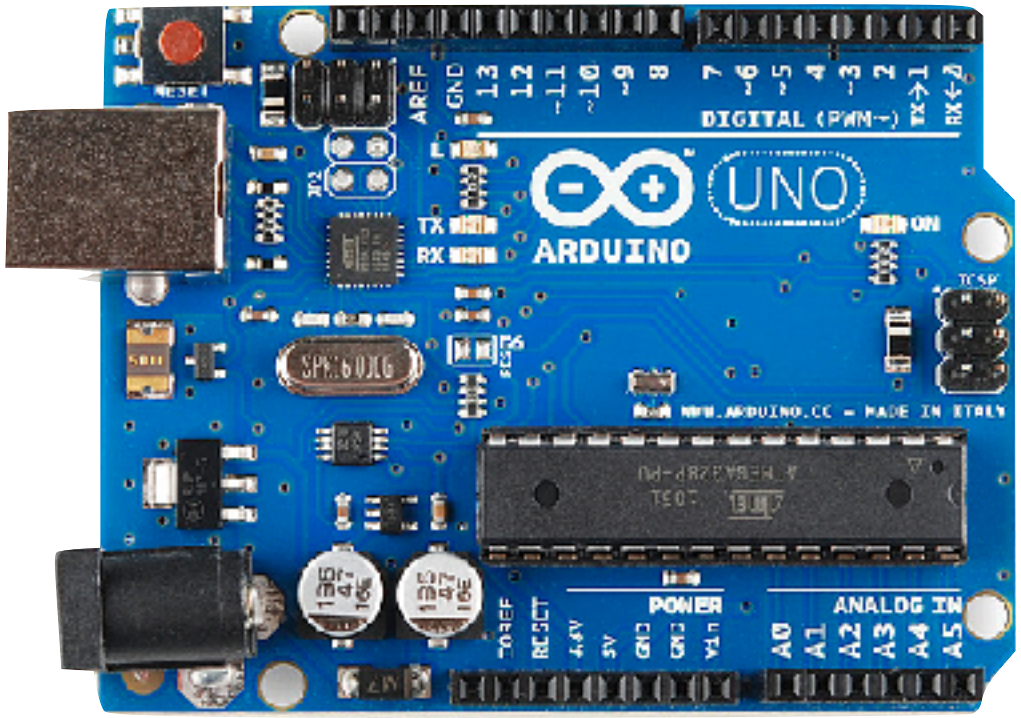


introduction to Digital Electronics

Install the Arduino IDE on your laptop if you haven't already!

Electronics can add interactivity!

Any sufficiently advanced technology is indistinguishable from magic
- Arthur Clark.

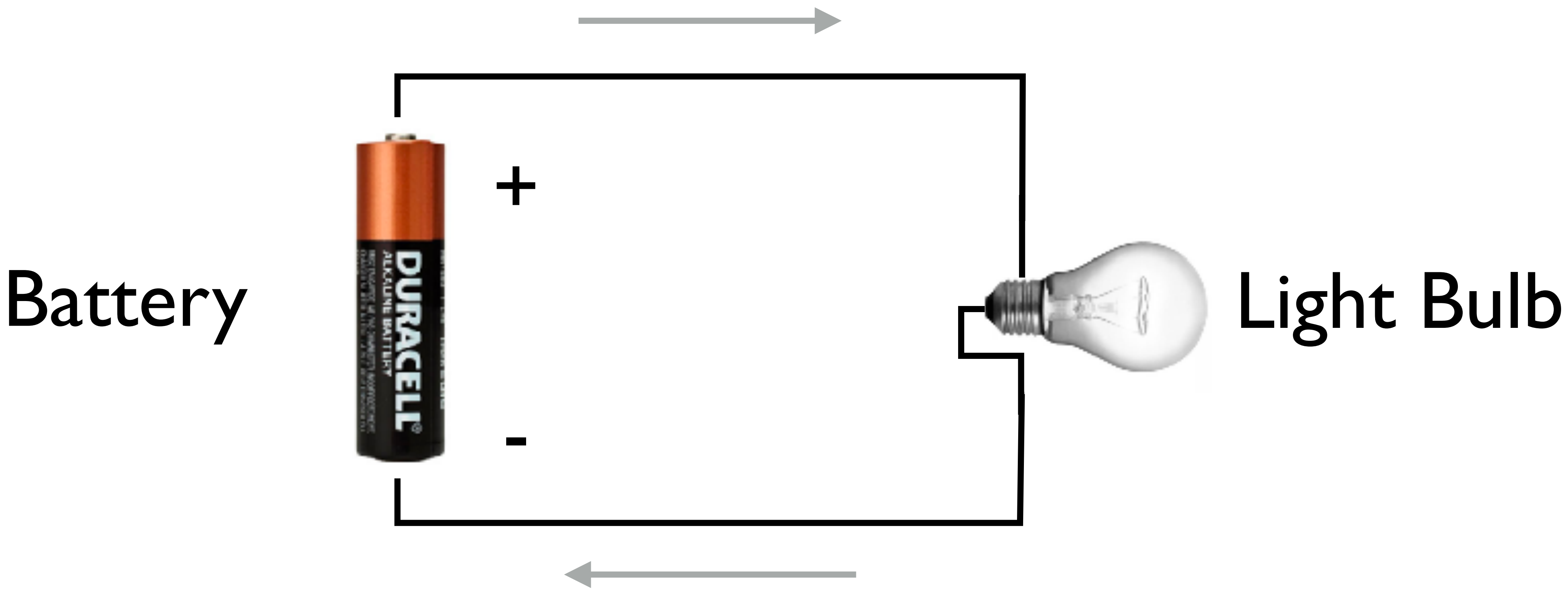
Electronics can add interactivity!

Any sufficiently ***well executed*** technology is indistinguishable from magic in the eyes of kids!

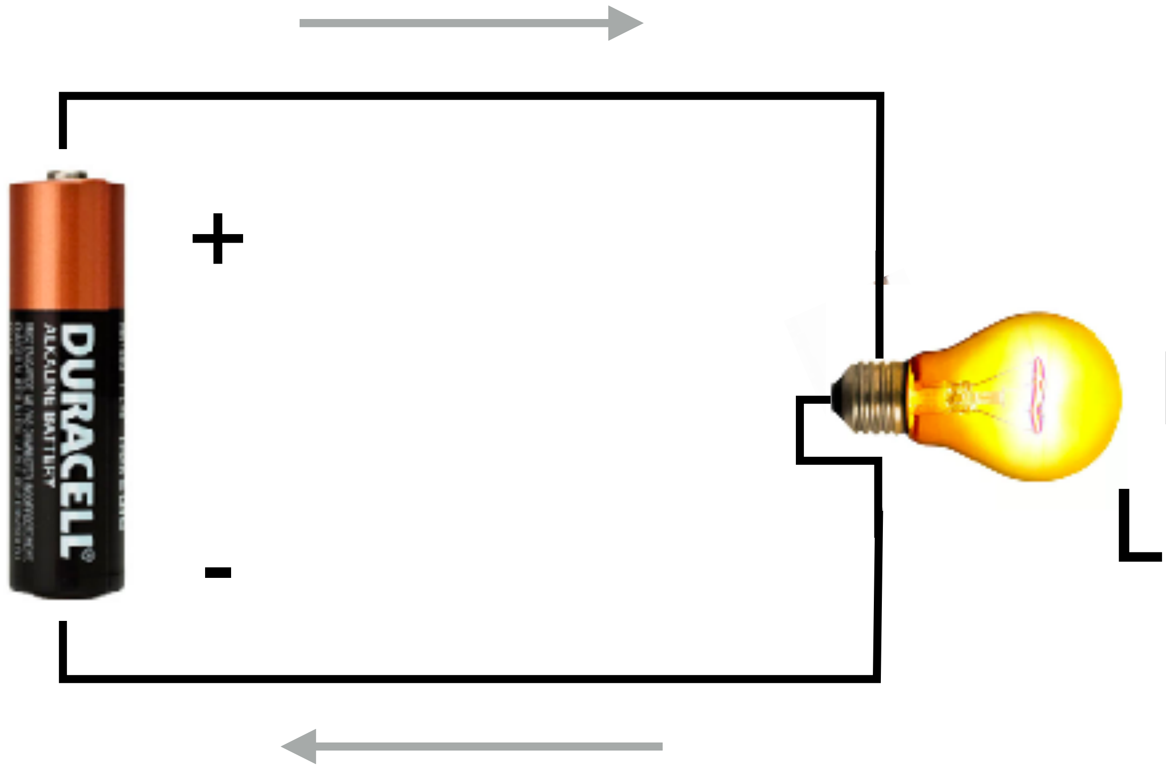
Electronics can be complex

Avoid unnecessary complexity!

Let's look at a circuit!



Let's look at a circuit!



Battery

Power Source

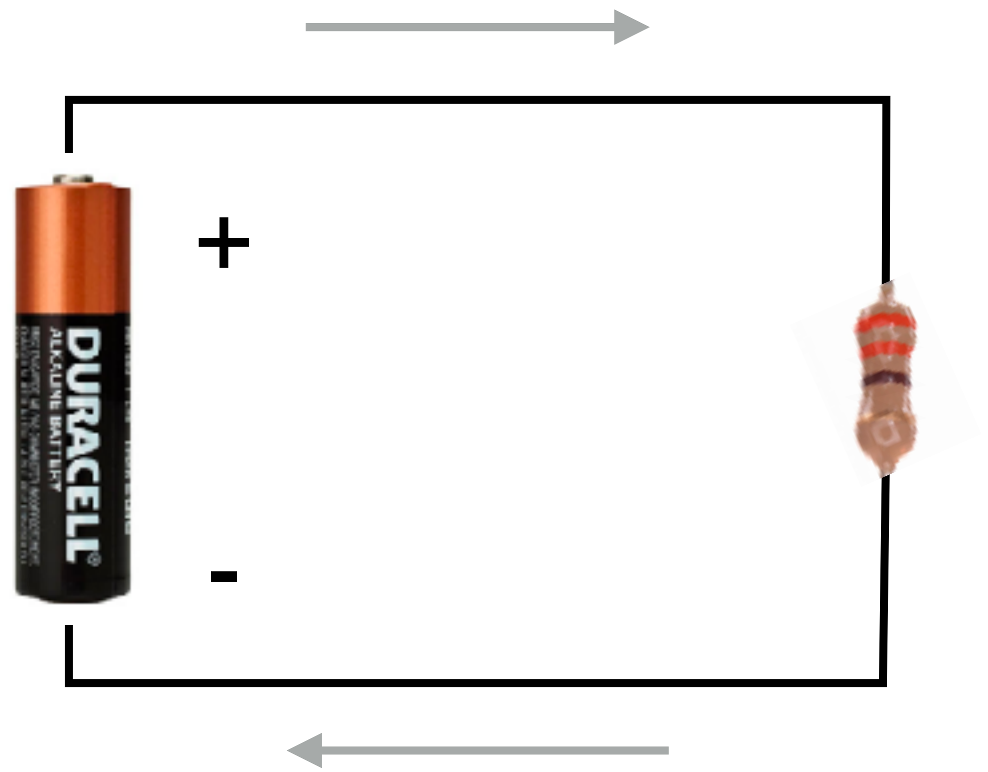
+

-

Light Bulb

Load

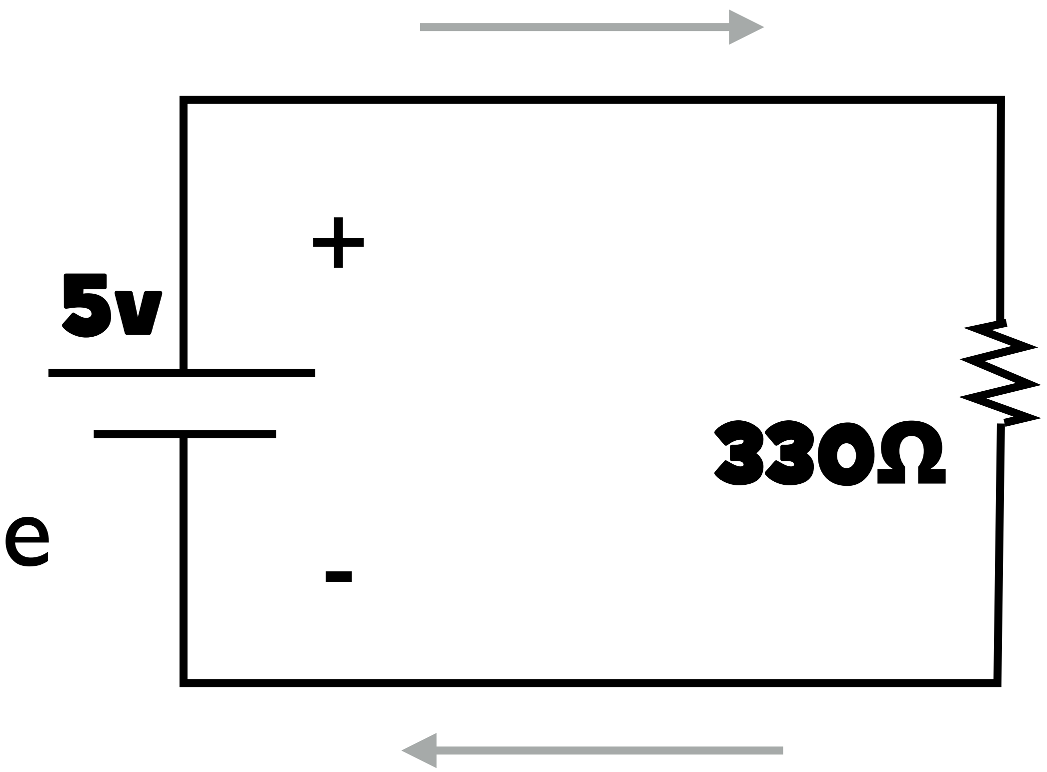
Let's look at a circuit!



Battery
Power Source

Resistor
Load

Let's look at a circuit!



Battery

Power Source

Resistor

Load

$$V = IR$$

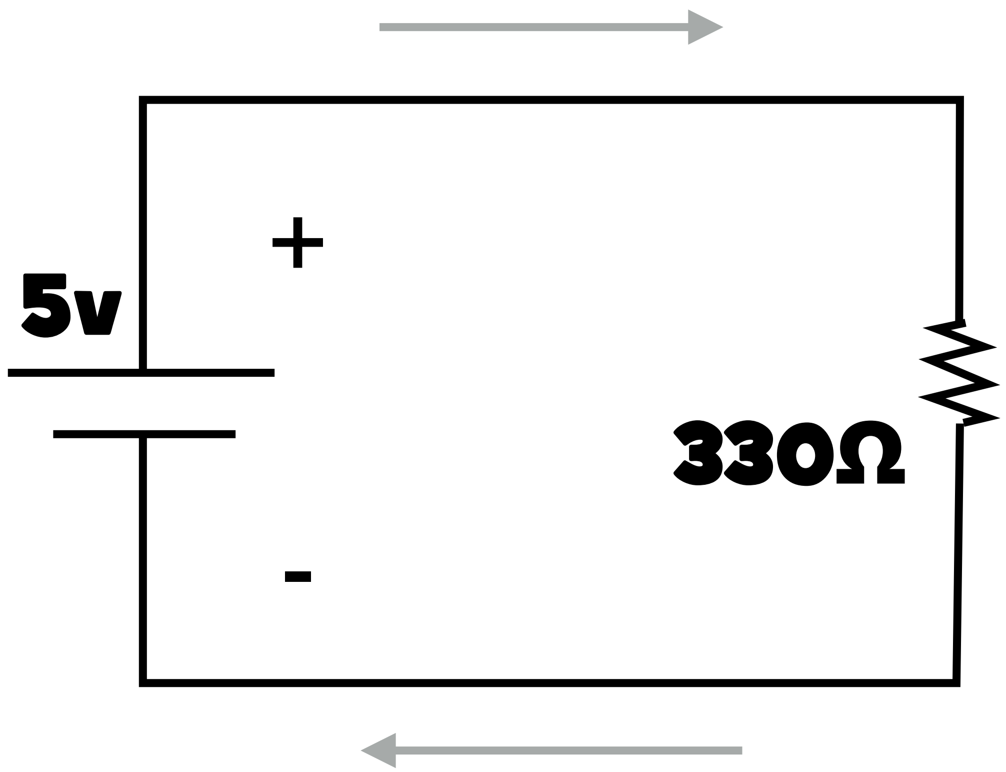
Voltage = Current × Resistance

Volts Amps Ohms

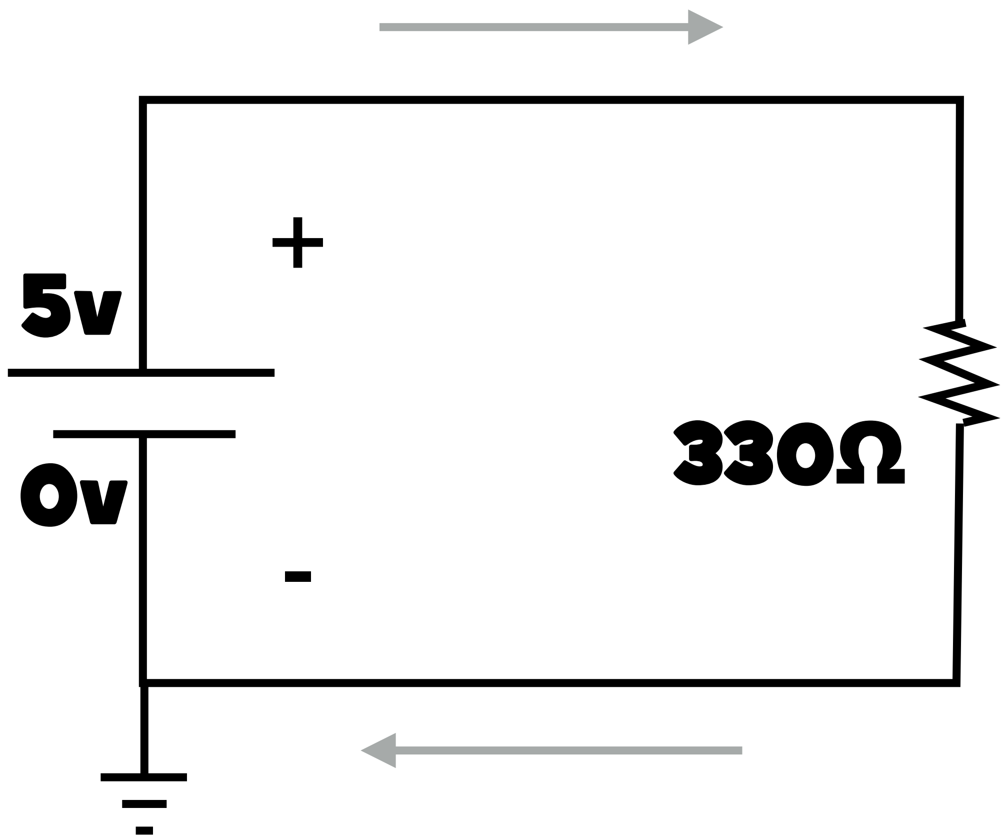
$$5 = I \cdot 330$$

Current ≈0.015A

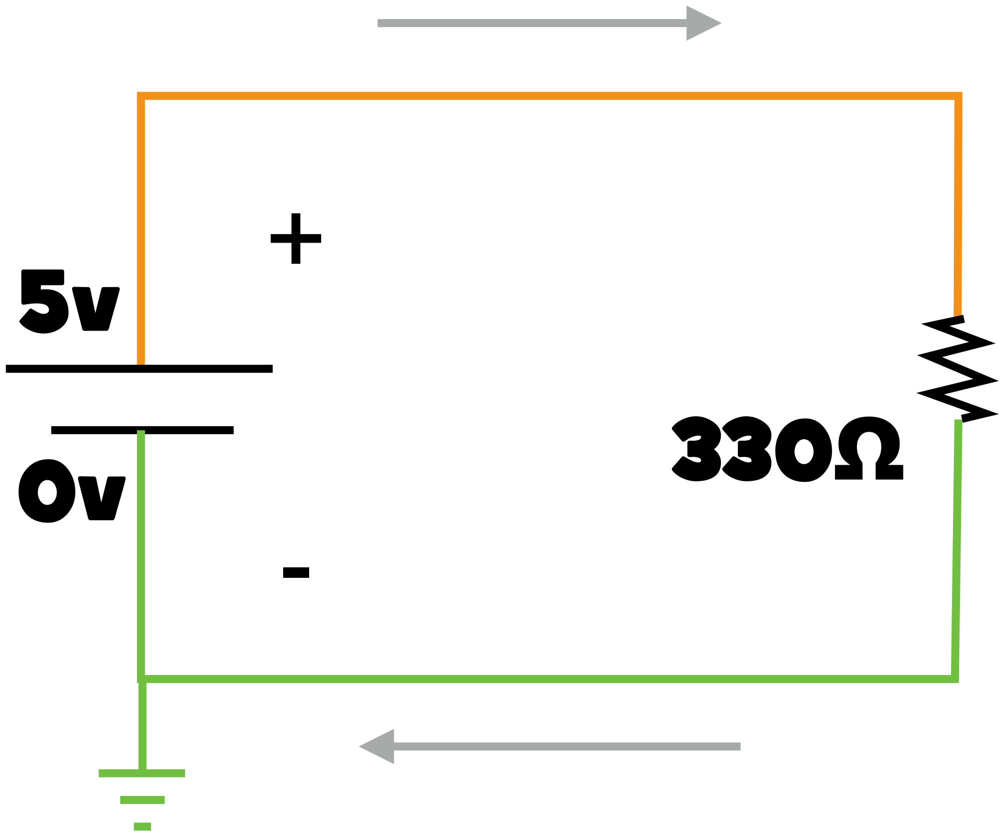
Let's look at a circuit!



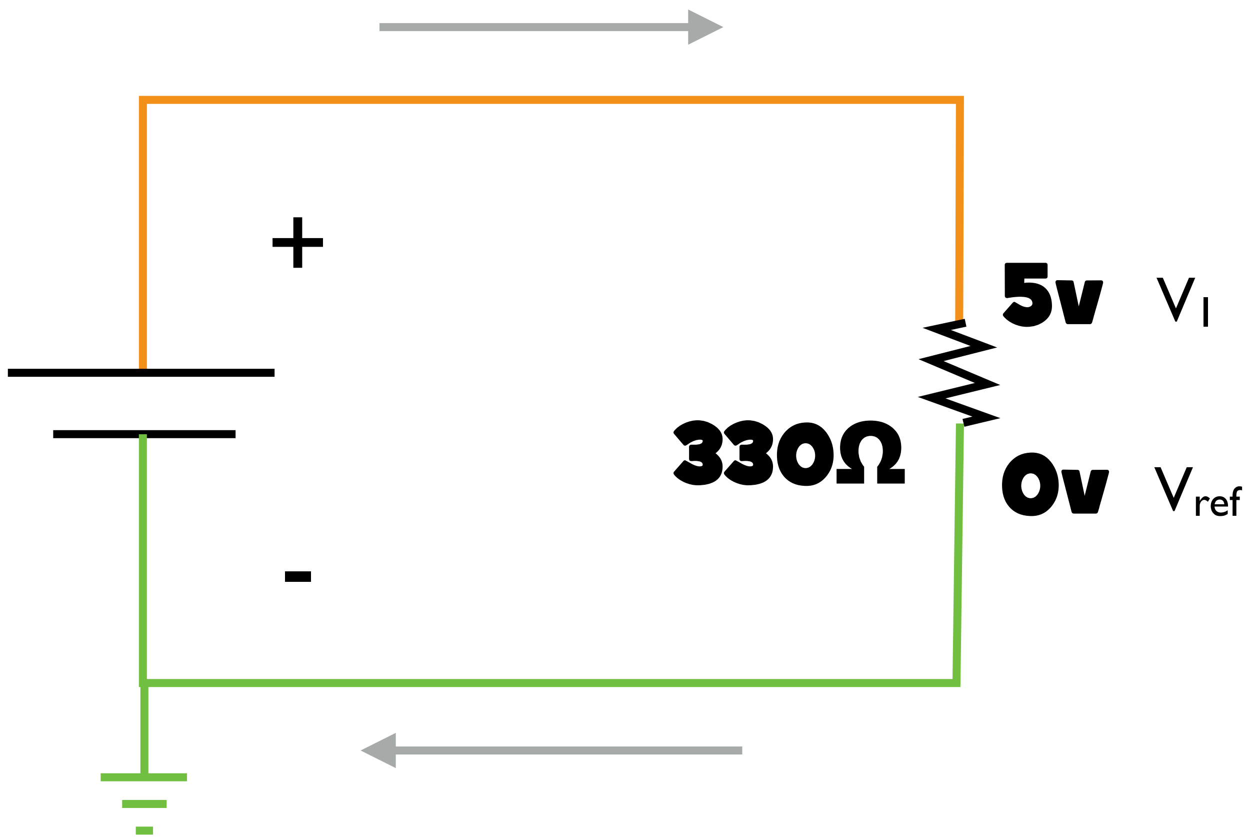
Let's look at a circuit!



Let's look at a circuit!

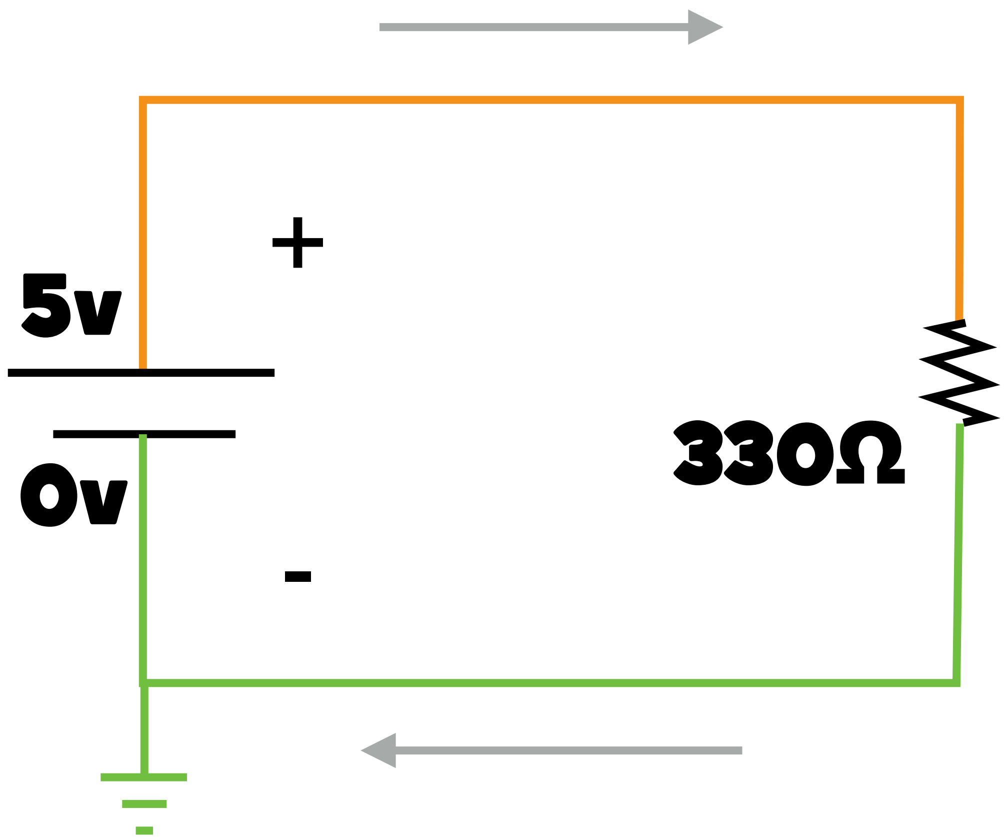


Let's look at a circuit!

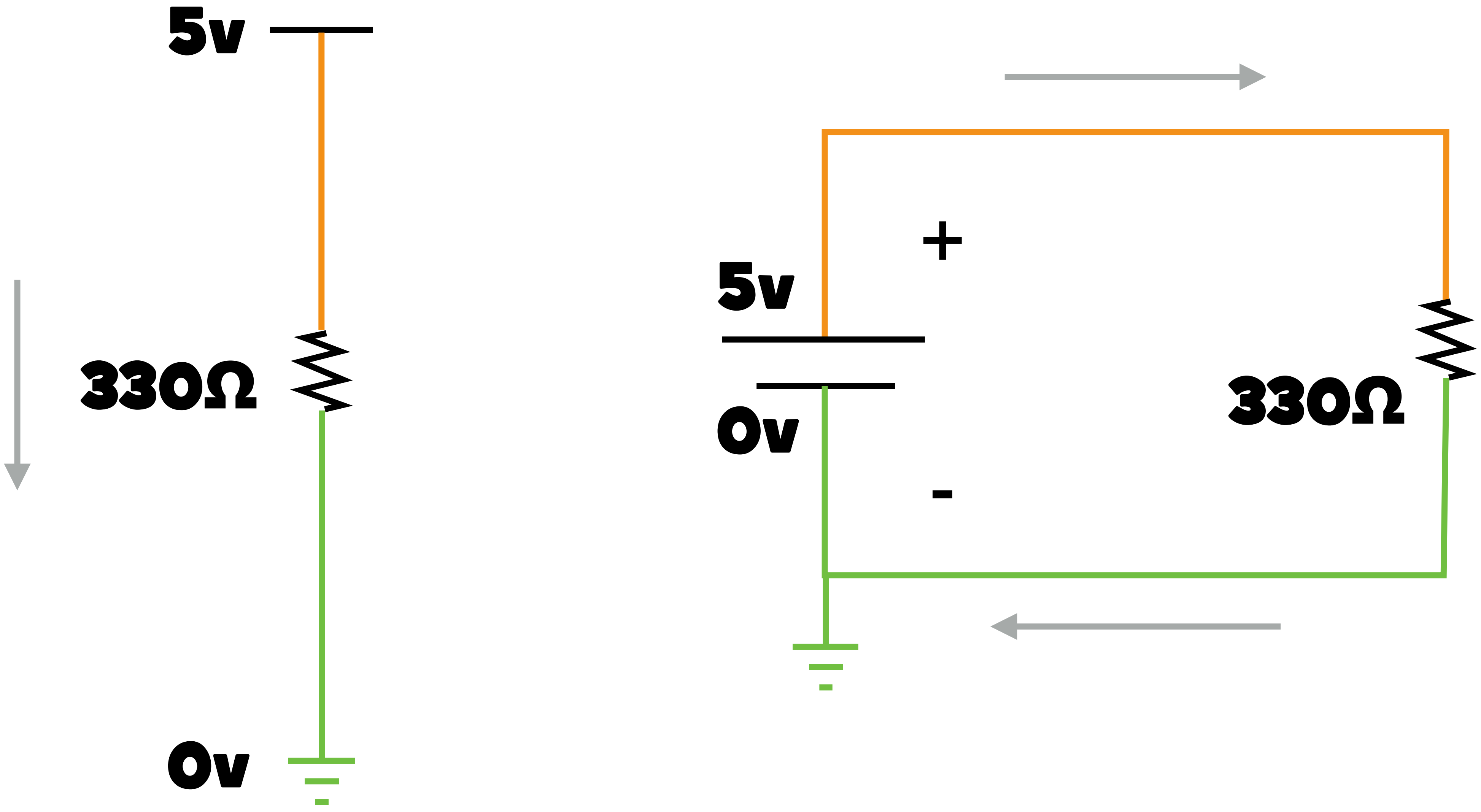


$$V_l - V_{ref} = IR$$

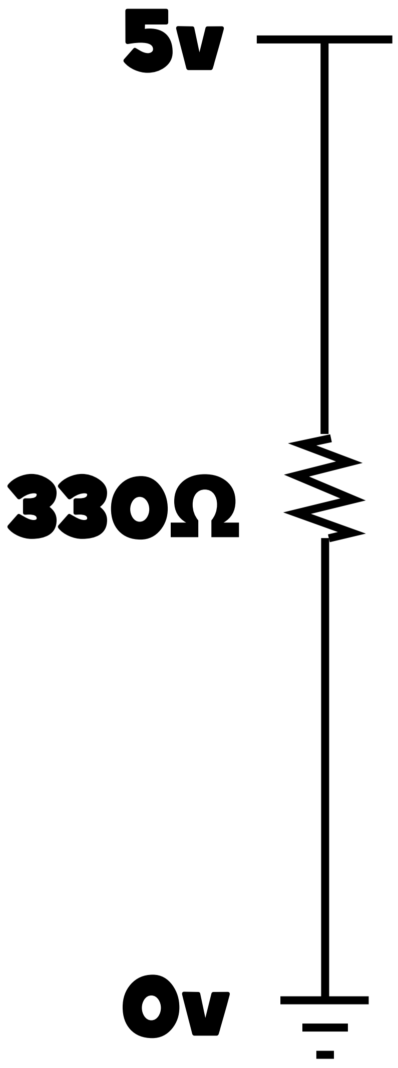
Let's look at a circuit!



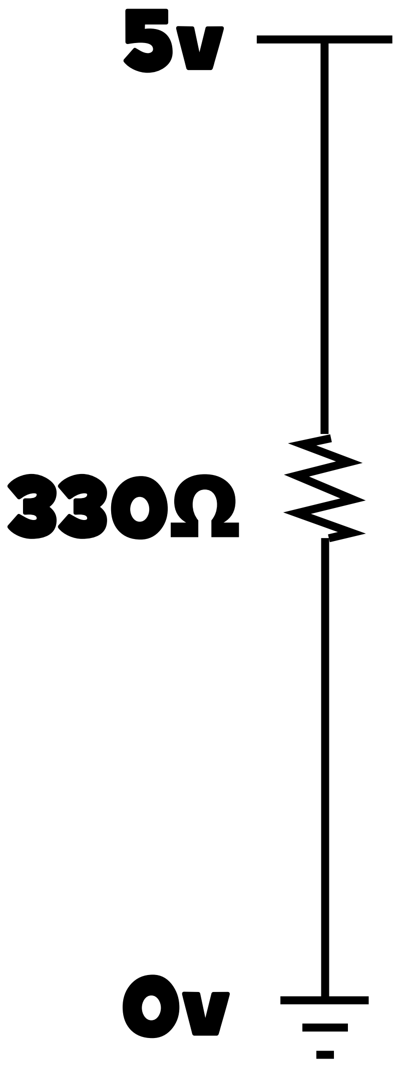
Let's look at a circuit!



Let's look at a circuit!



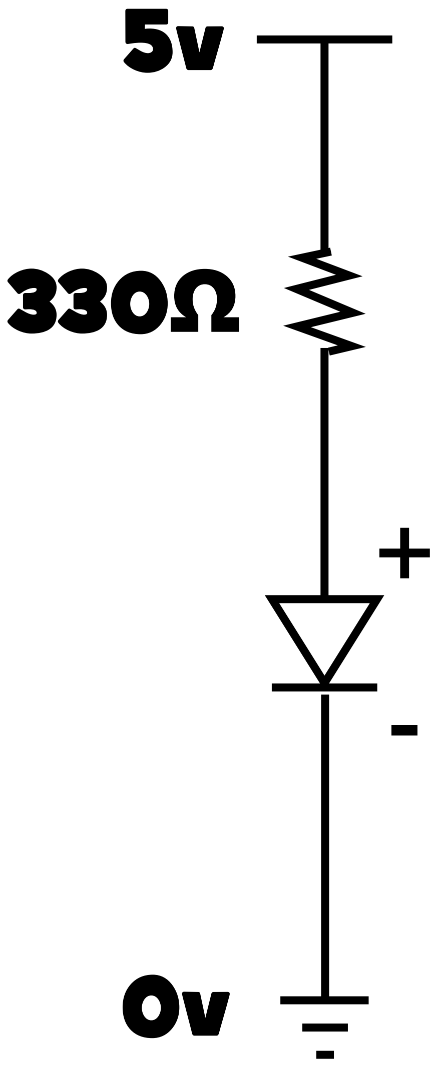
Time for some LIGHT!

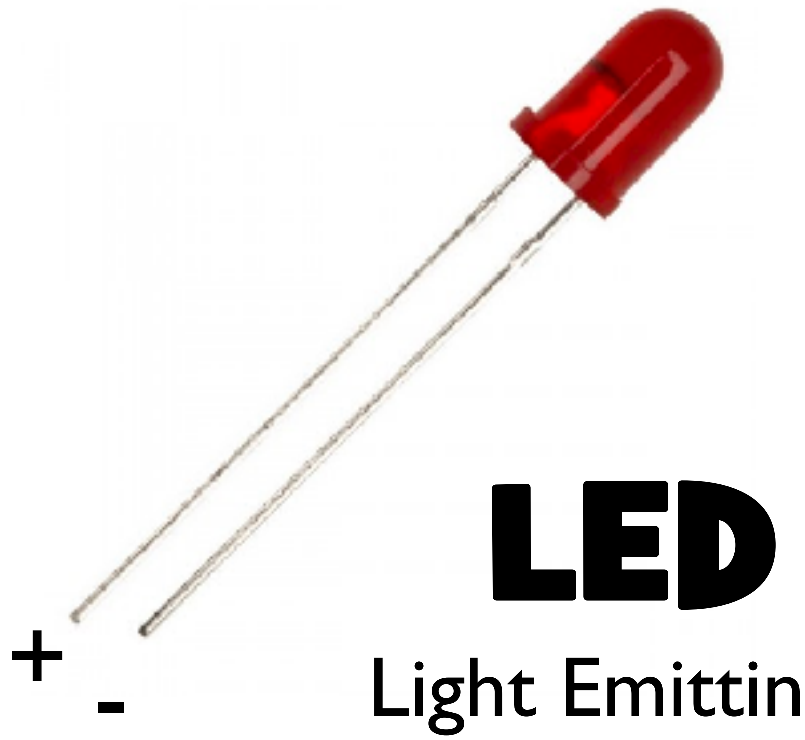


LED

Light Emitting Diode

Time for some LIGHT!





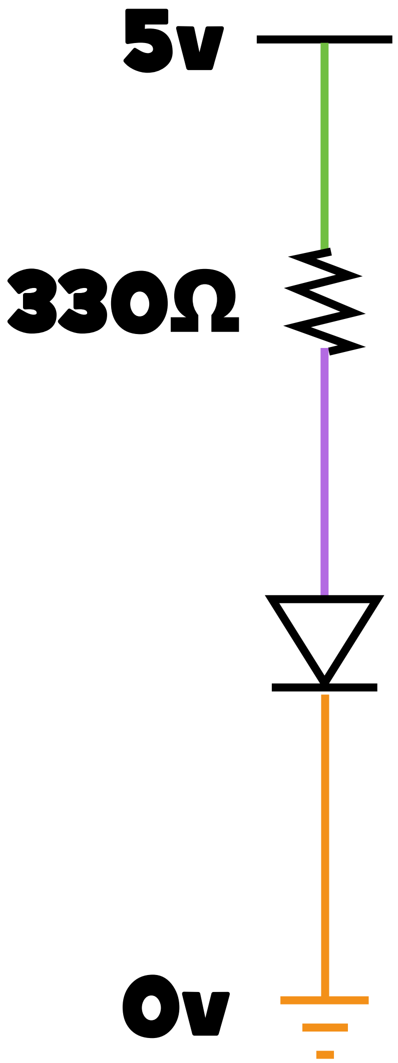
Longer is positive!

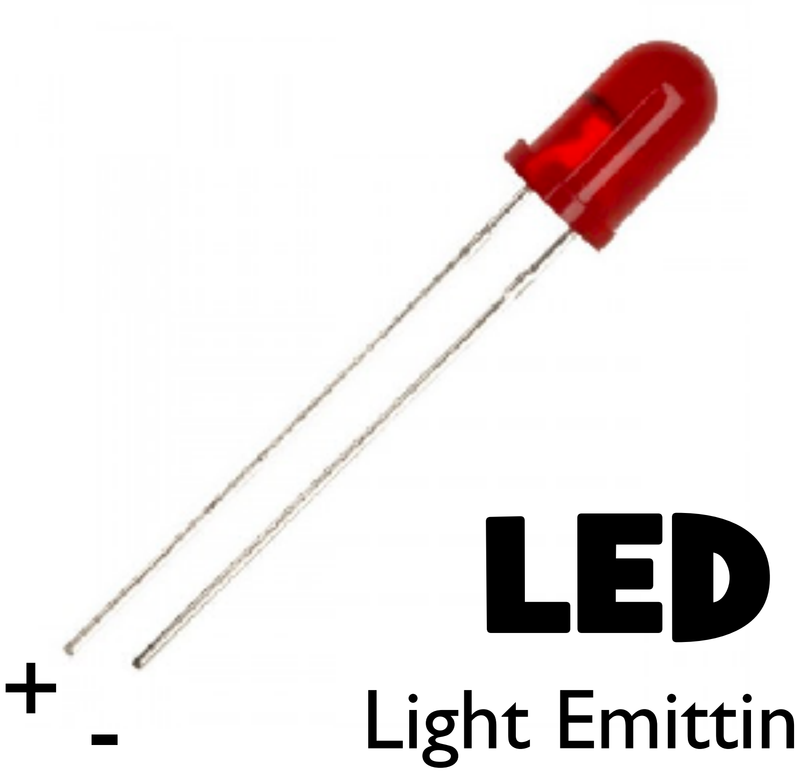
LED

Light Emitting Diode

Diodes conducts current primarily in one direction
Needs resistor!

Time for some LIGHT!



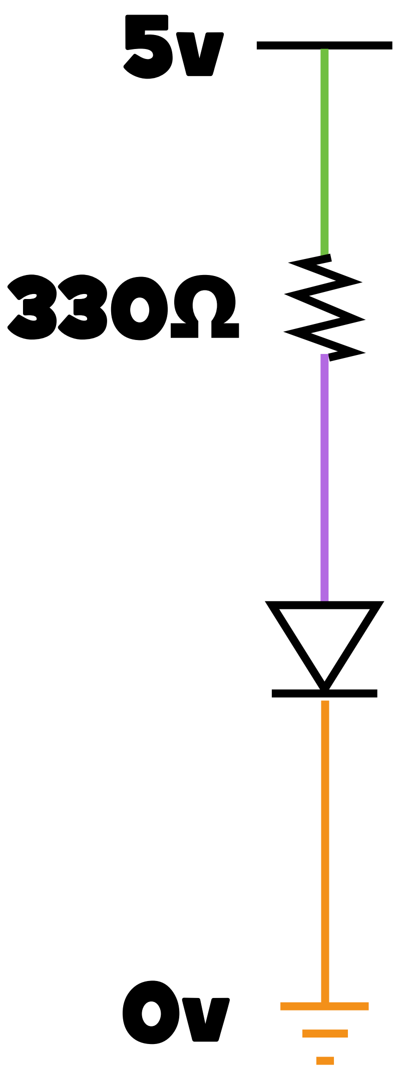


Longer is positive!

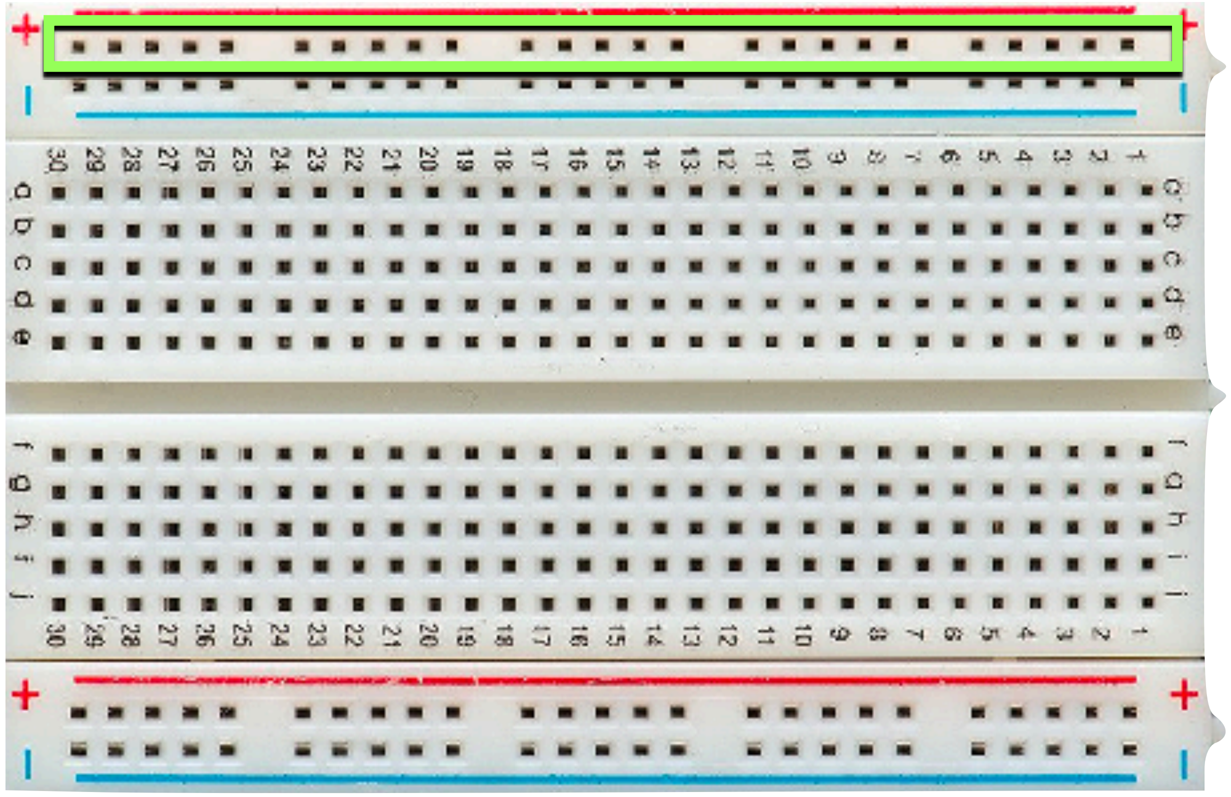
LED

Light Emitting Diode

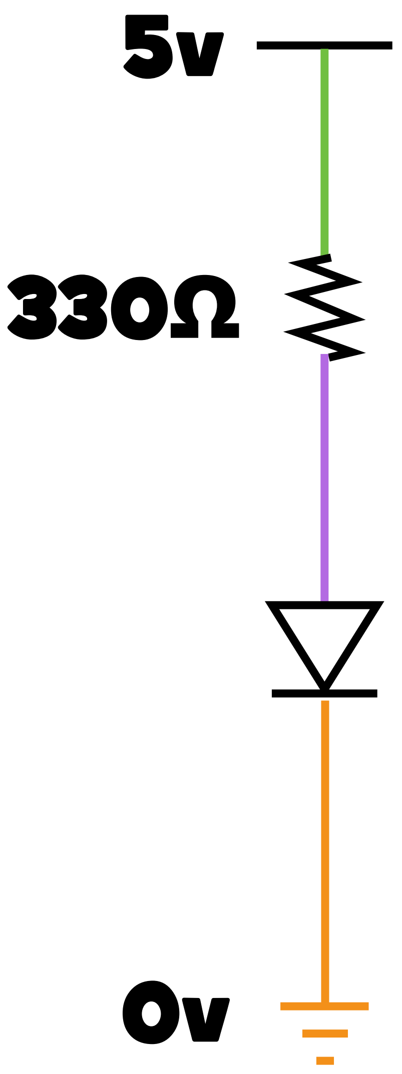
In Real Life!

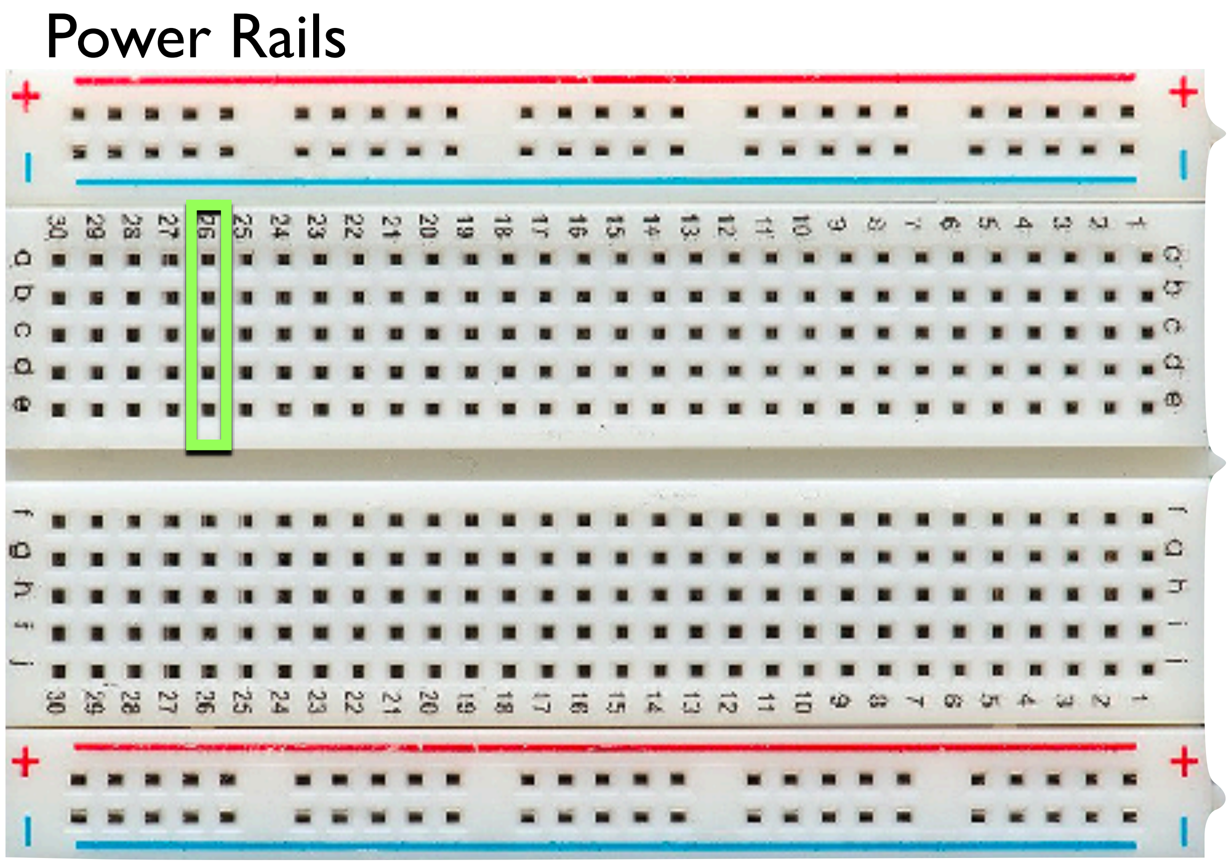


Power Rails

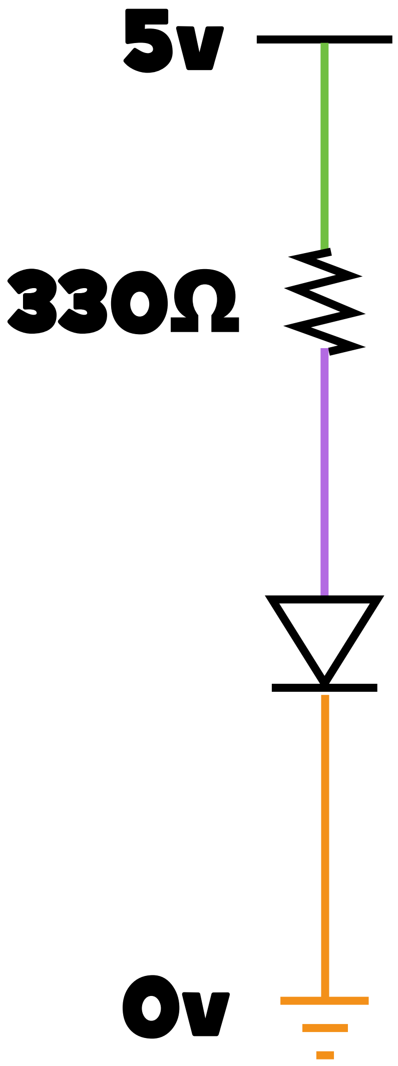


In Real Life!

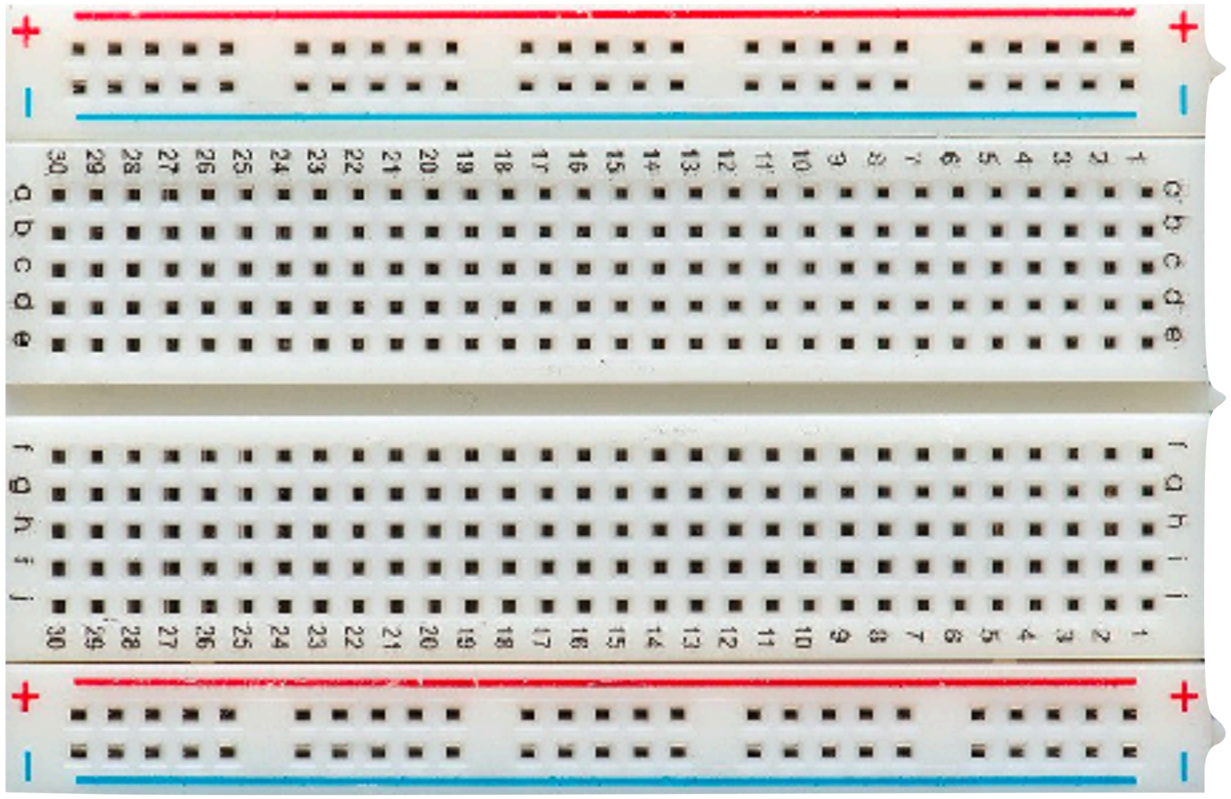




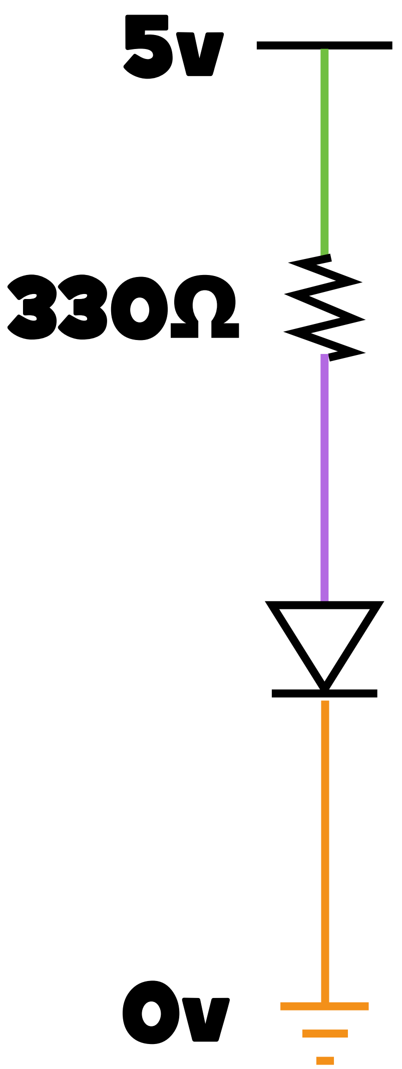
In Real Life!

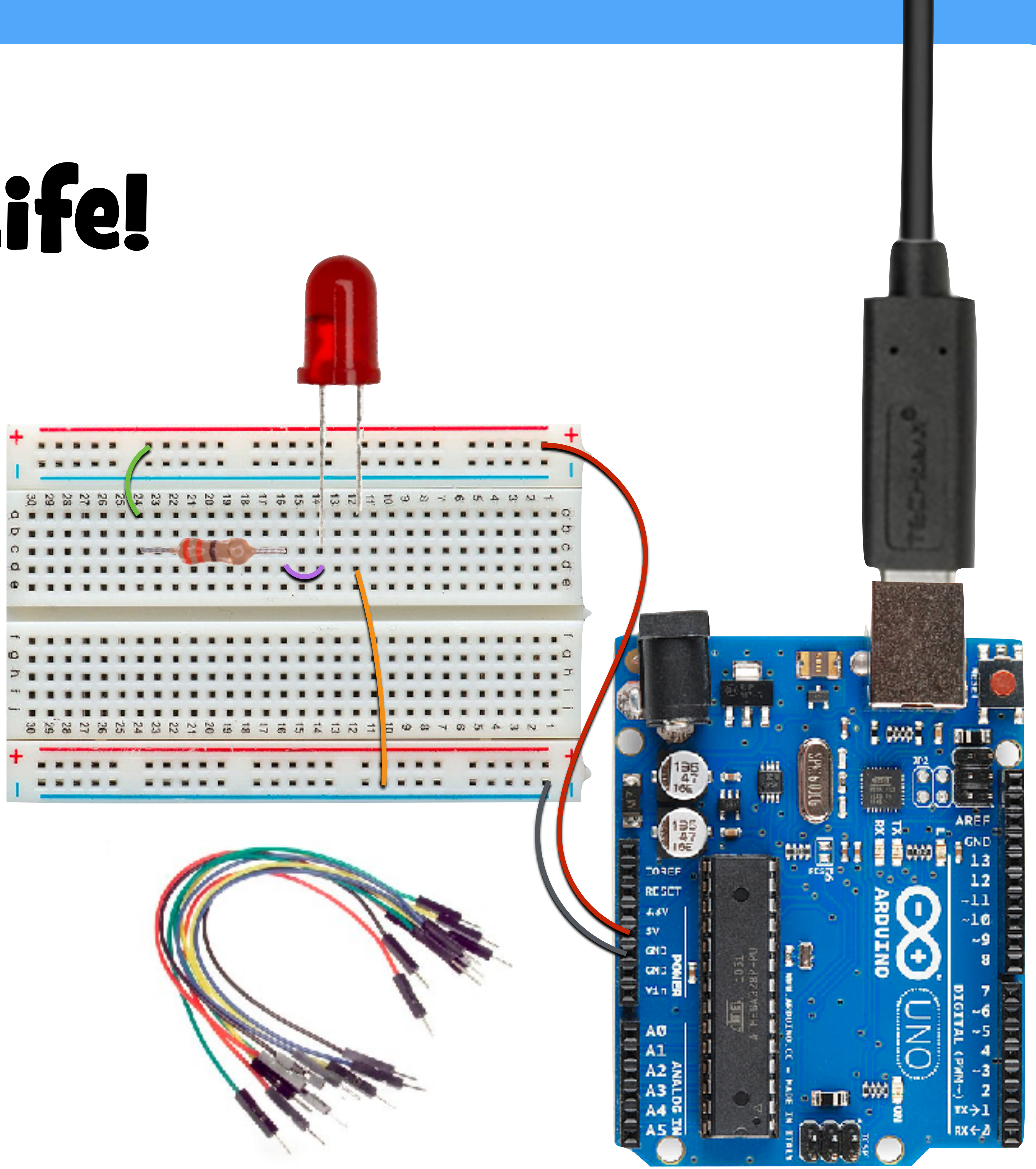


Power Rails

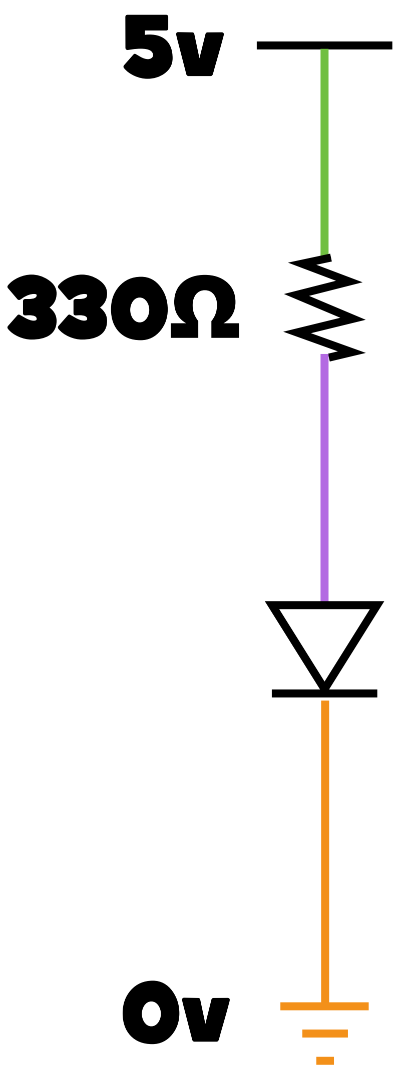


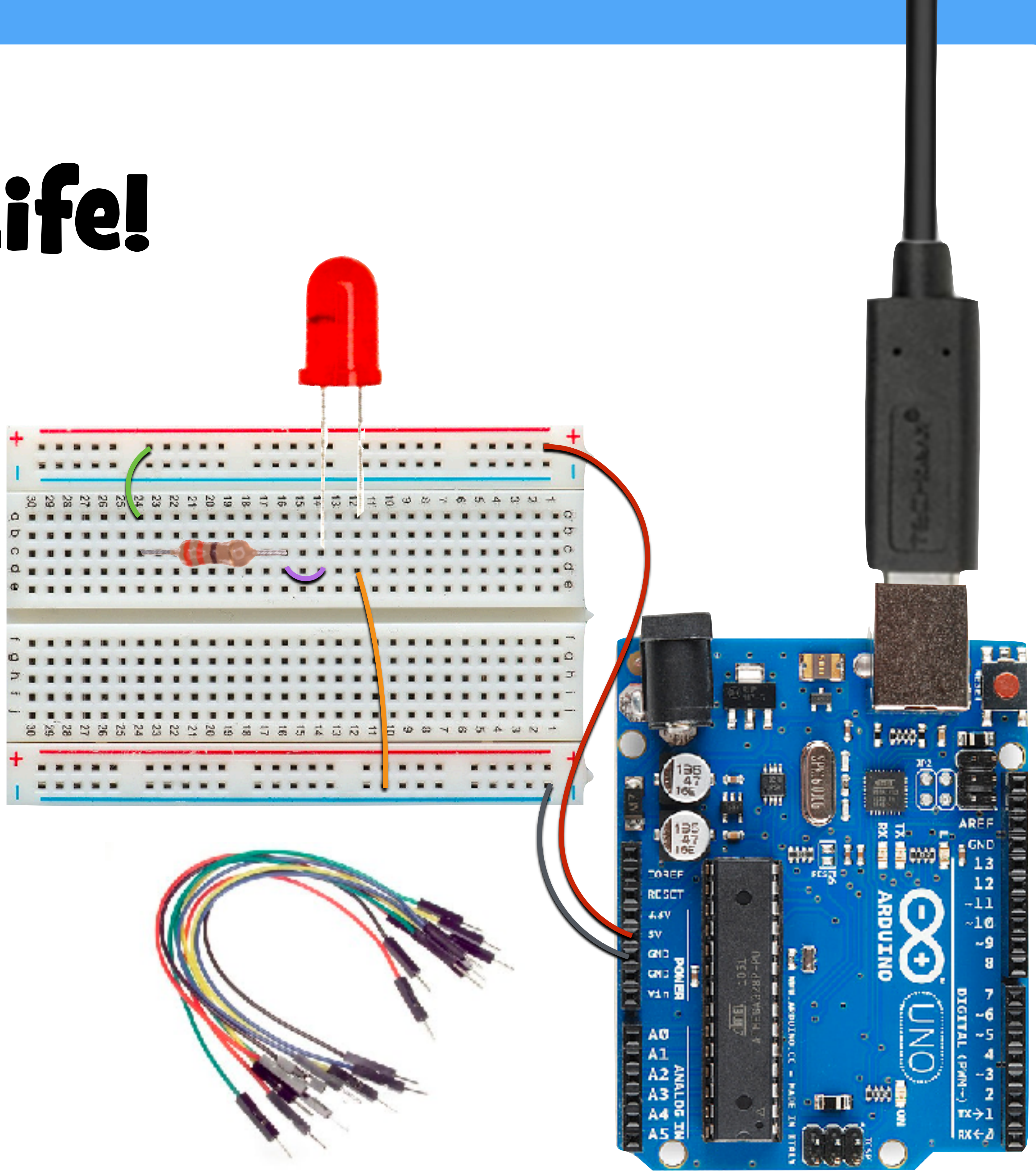
In Real Life!


5v
330Ω
0v

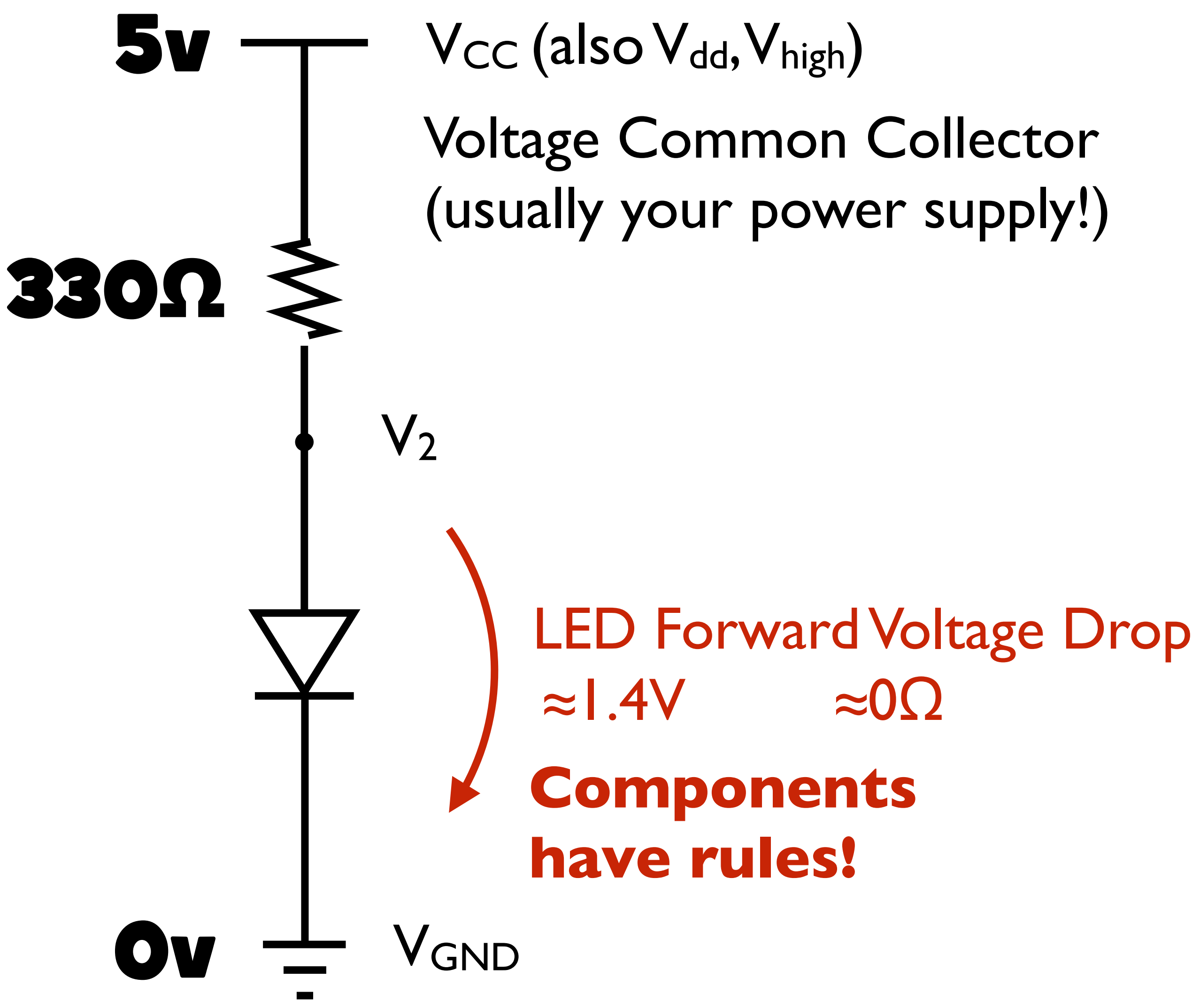

ARDUINO
UNO
IOREF
RESET
3.3V
5V
GND
GND
Vin
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
TX
RX

In Real Life!


5v
330Ω
0v


ARDUINO
UNO
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
IOREF
RESET
3.3V
5V
GND
GND
Vin
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN

A closer look...



V_{CC} (also V_{dd}, V_{high})

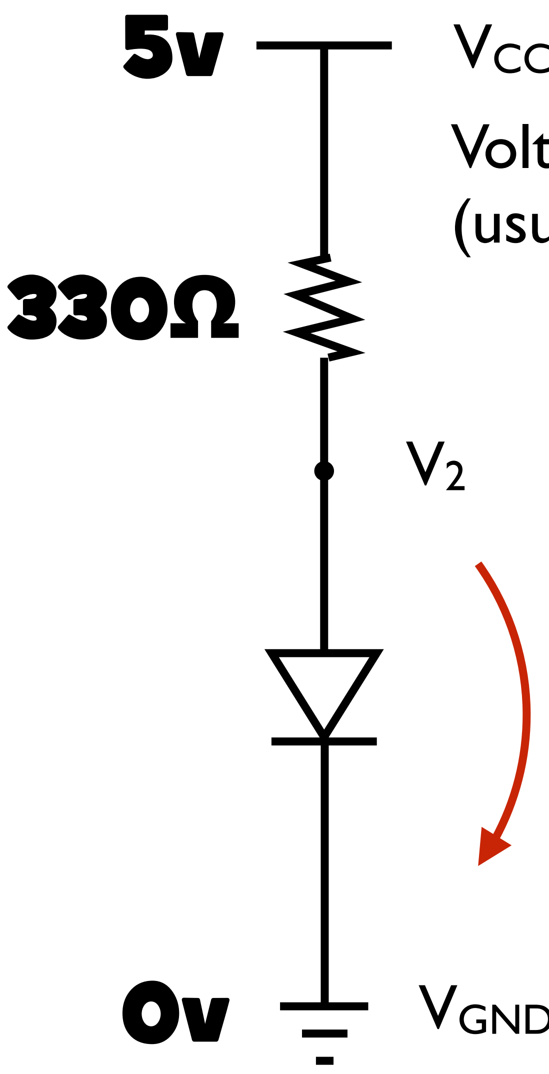
Voltage Common Collector
(usually your power supply!)

LED Forward Voltage Drop
≈1.4V ≈0Ω

Components have rules!

$$V_2 - V_{GND} = 1.4V$$

A closer look...



V_{CC} (also V_{dd}, V_{high})

Voltage Common Collector
(usually your power supply!)

LED Forward Voltage Drop
≈1.4V ≈0Ω

Components have rules!

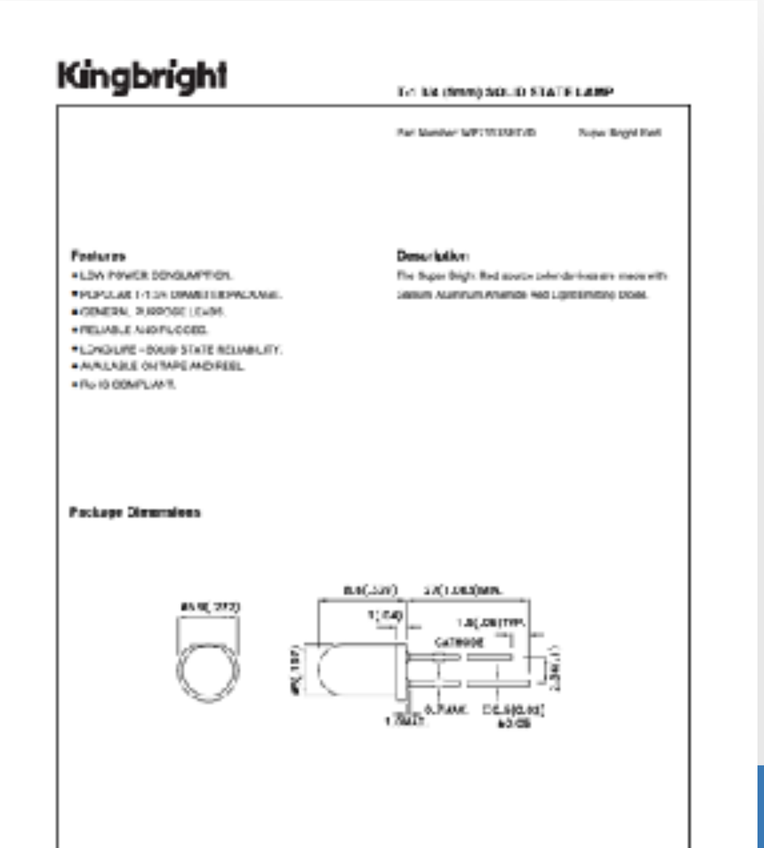
V_{GND}

$$V_2 = 1.4V$$

$$V_{CC} - V_2 = I \cdot R$$

$$5 - 1.4 = I \cdot 330$$

$$I \approx 10 \text{ mA}$$



Kingbright

T-1 3/4 (5mm) SOLID STATE LAMP

Part Number: WP7113SRD/D Super Bright Red

Features

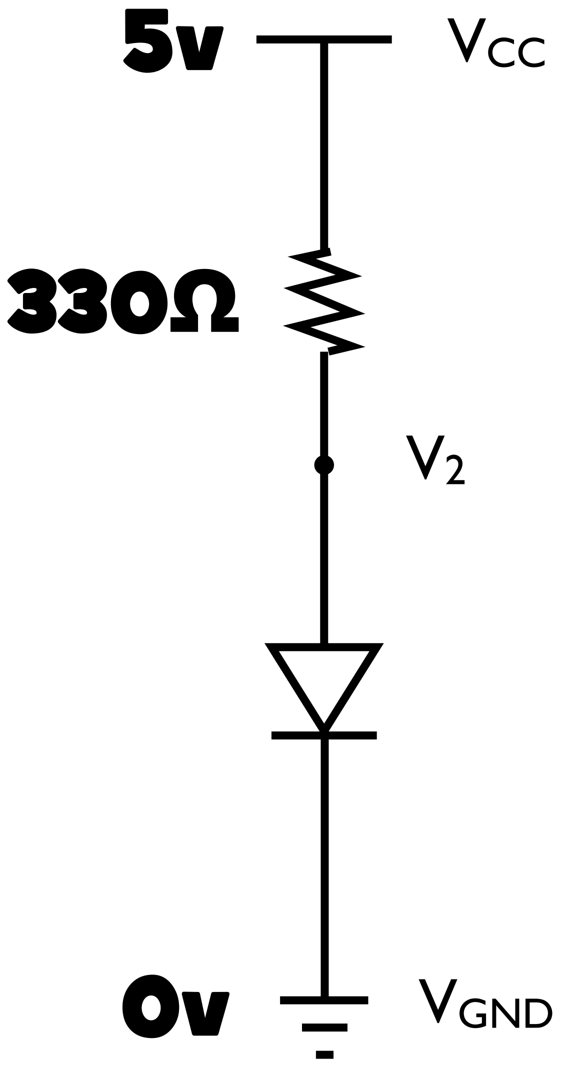
- LOW POWER CONSUMPTION.
- POPULAR T-1 3/4 DIAMETER PACKAGE.
- GENERAL PURPOSE LEADS.
- RELIABLE AND RUGGED.
- LONG LIFE - SOLID STATE RELIABILITY.
- AVAILABLE ON TAPE AND REEL.
- RoHS COMPLIANT.

Description

The Super Bright Red source color devices are made with Gallium Aluminum Arsenide Red Light Emitting Diode.

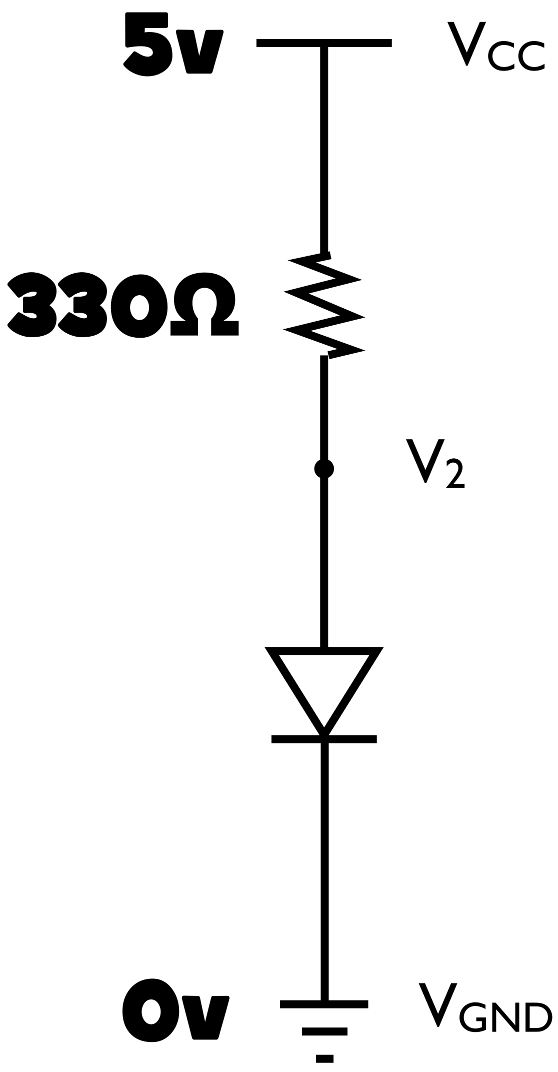
Package Dimensions

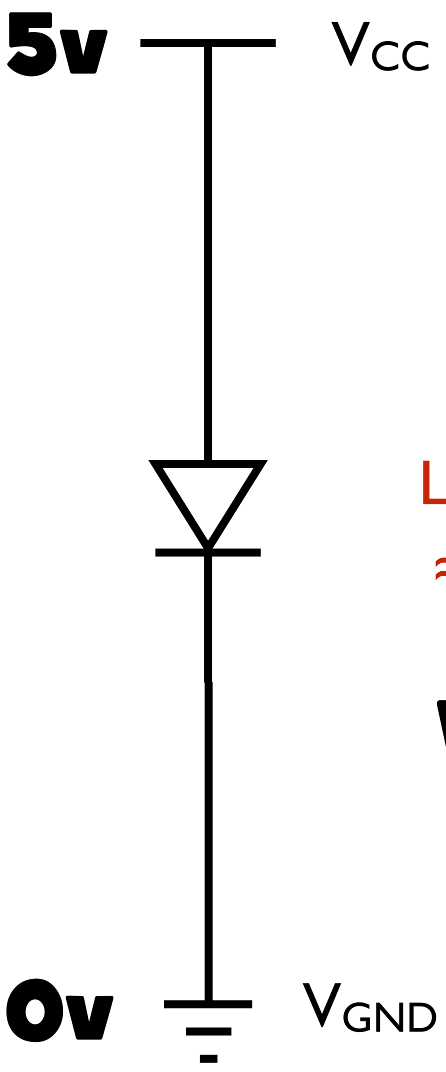
A closer look...



$I \approx 10$ mA

A closer look...





LED Max Current
≈20mA

$$V = IR$$

$I \approx 10$ mA

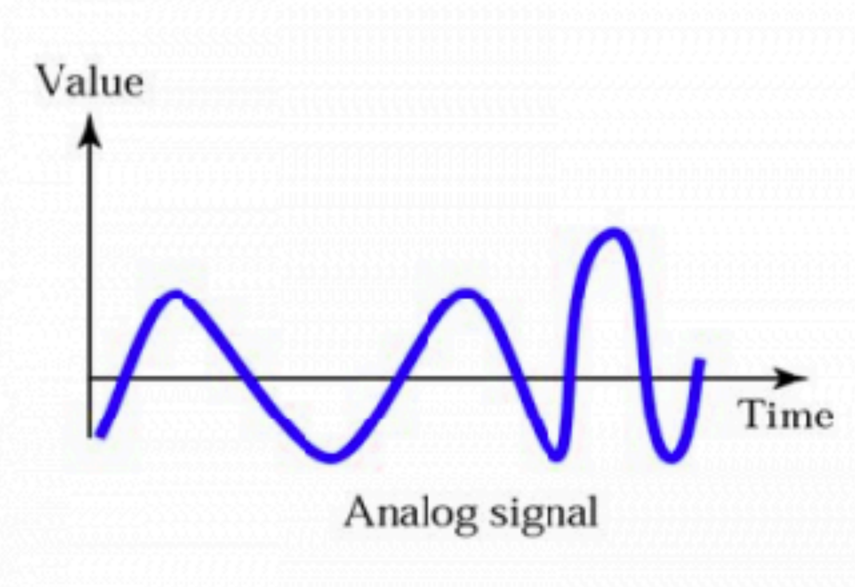
$I \approx$ BIG NUMBER

Digital vs Analog Circuits

Analog Circuits

Range of voltages

Usually requires math!



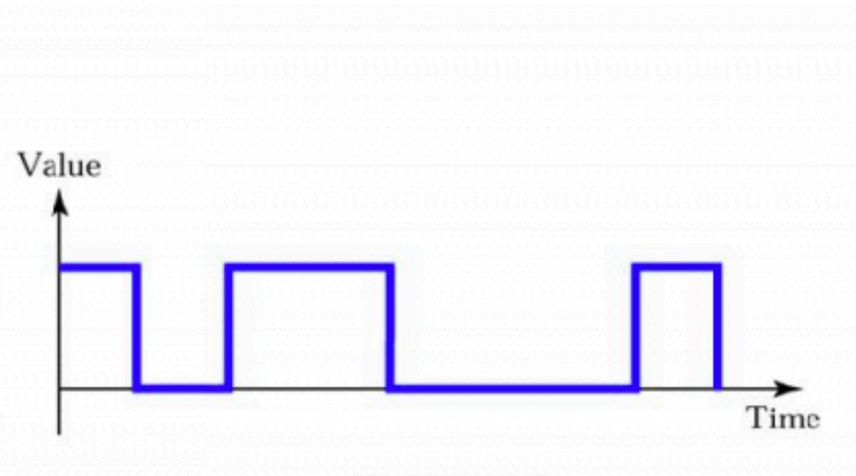
Analog signal

Digital Circuits

Usually 2 distinct voltages (high & low)

5v and 0v (roughly)

01100101011011010110000101101001
01101100011101010111001101100010
01101001011101000111001100100001



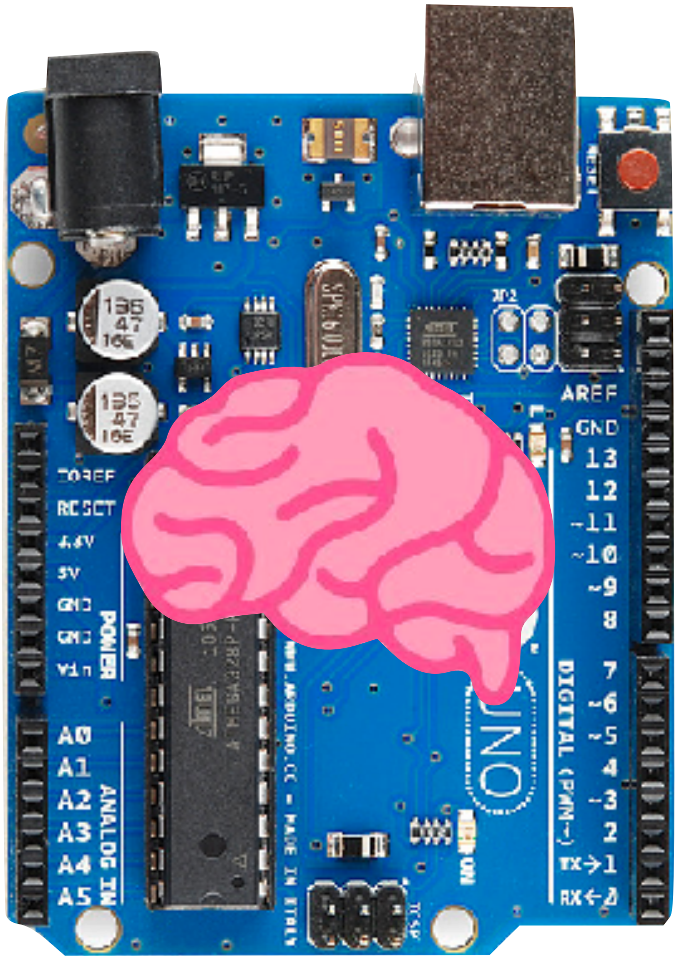
Digital signal

Using your Arduino!

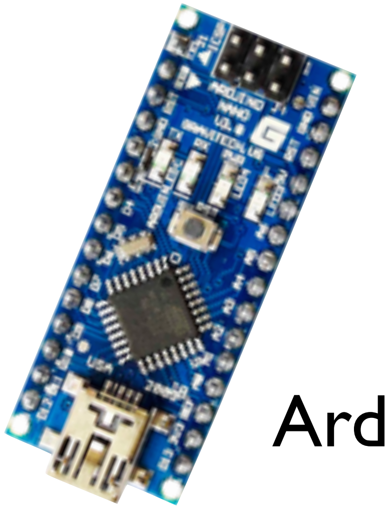
A Microcontroller

.. or a small computer!

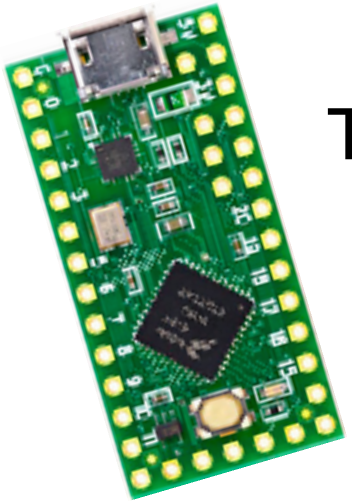
Has inputs and outputs you can control



Arduino Uno

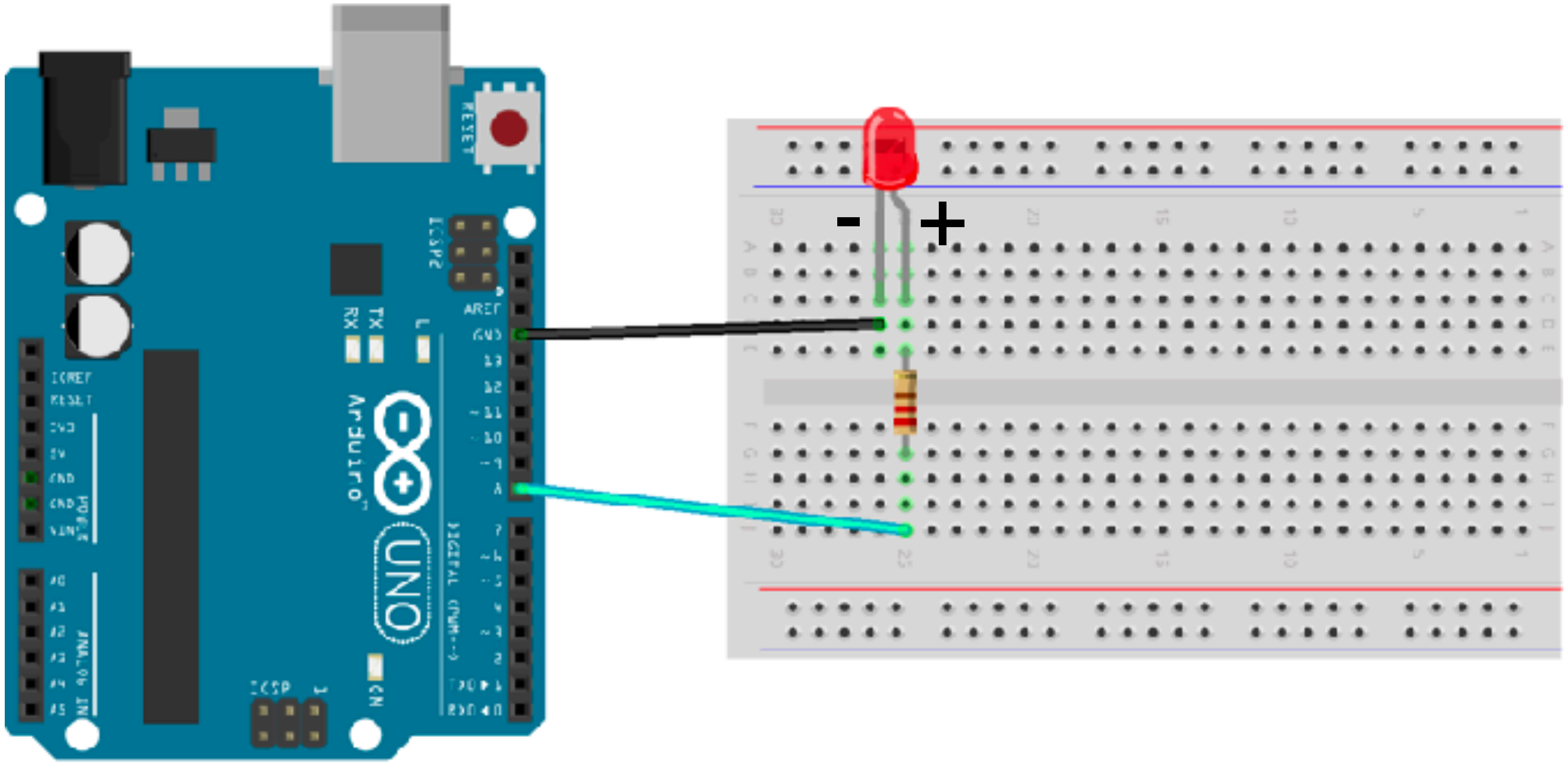


Arduino Nano

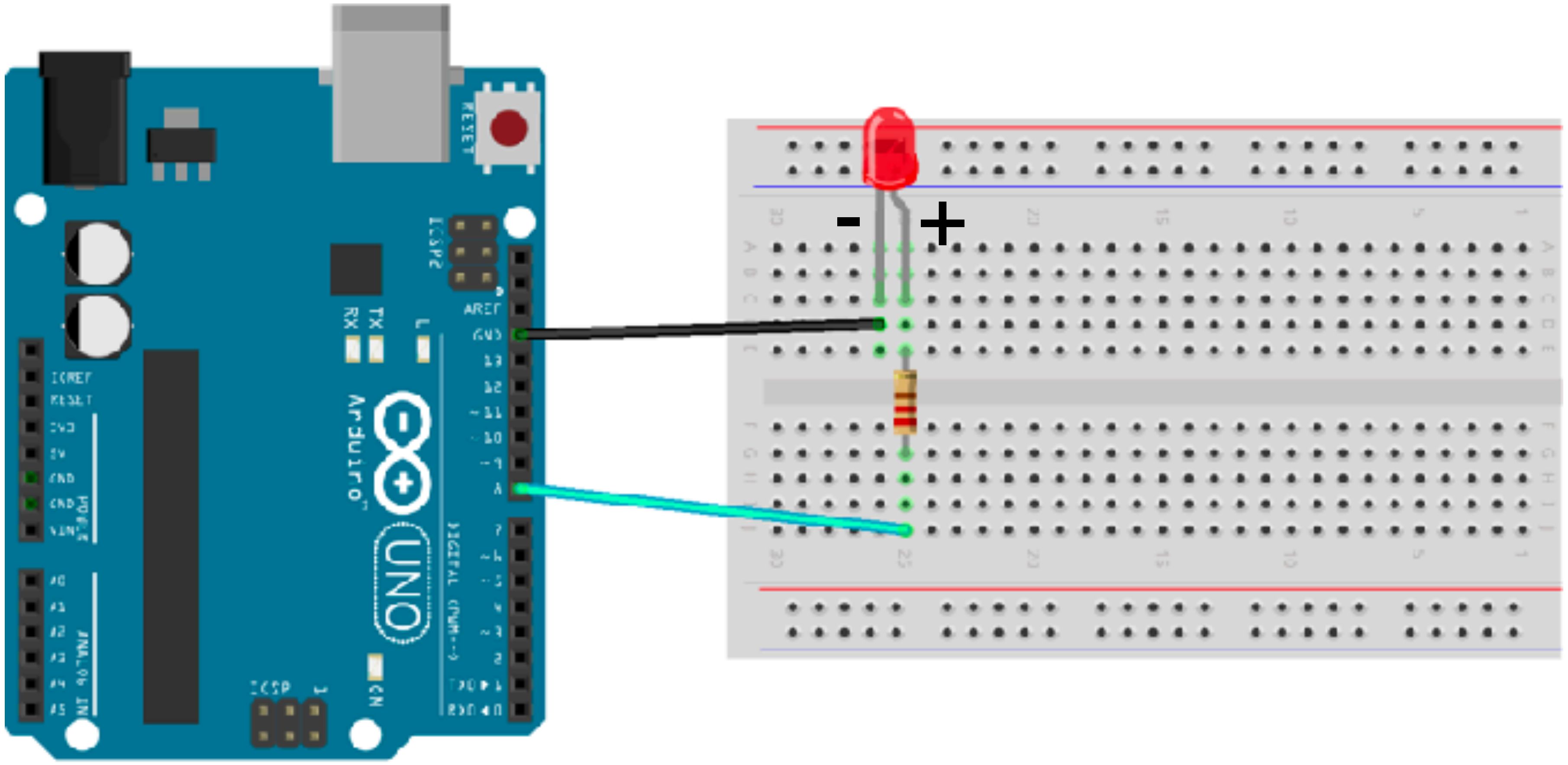


Teensy

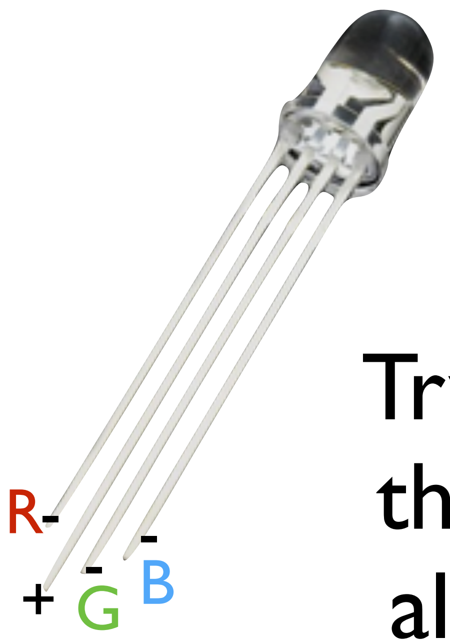
Using your Arduino!



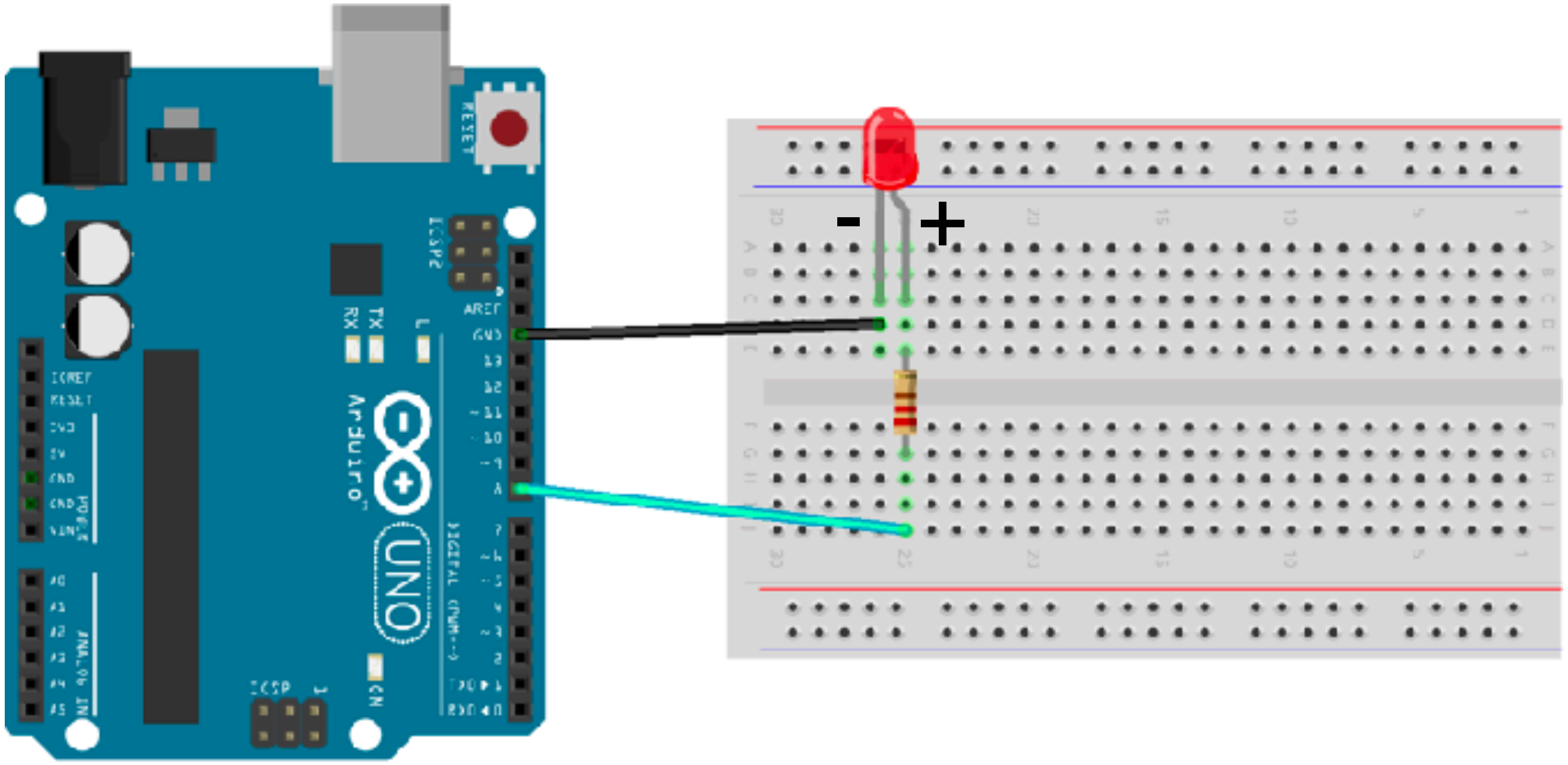
Blink!


-
+
RESET
ICSP2
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX▶1
RX◀0
DIGITAL (PWM~)
TX
RX
L
Arduino
UNO
ON
ICSP
IOREF
RESET
3V3
5V
GND
GND
VIN
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
30
25
20
15
10
5
1

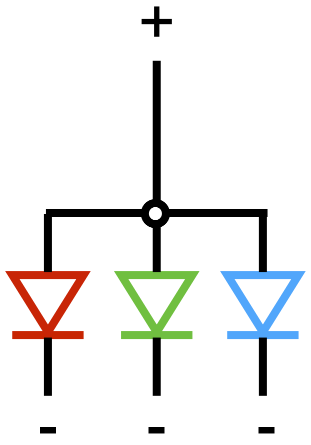
Blink!

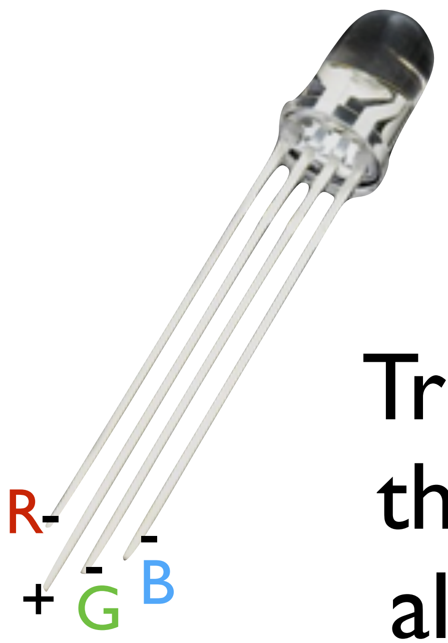


Try making the colors alternate!

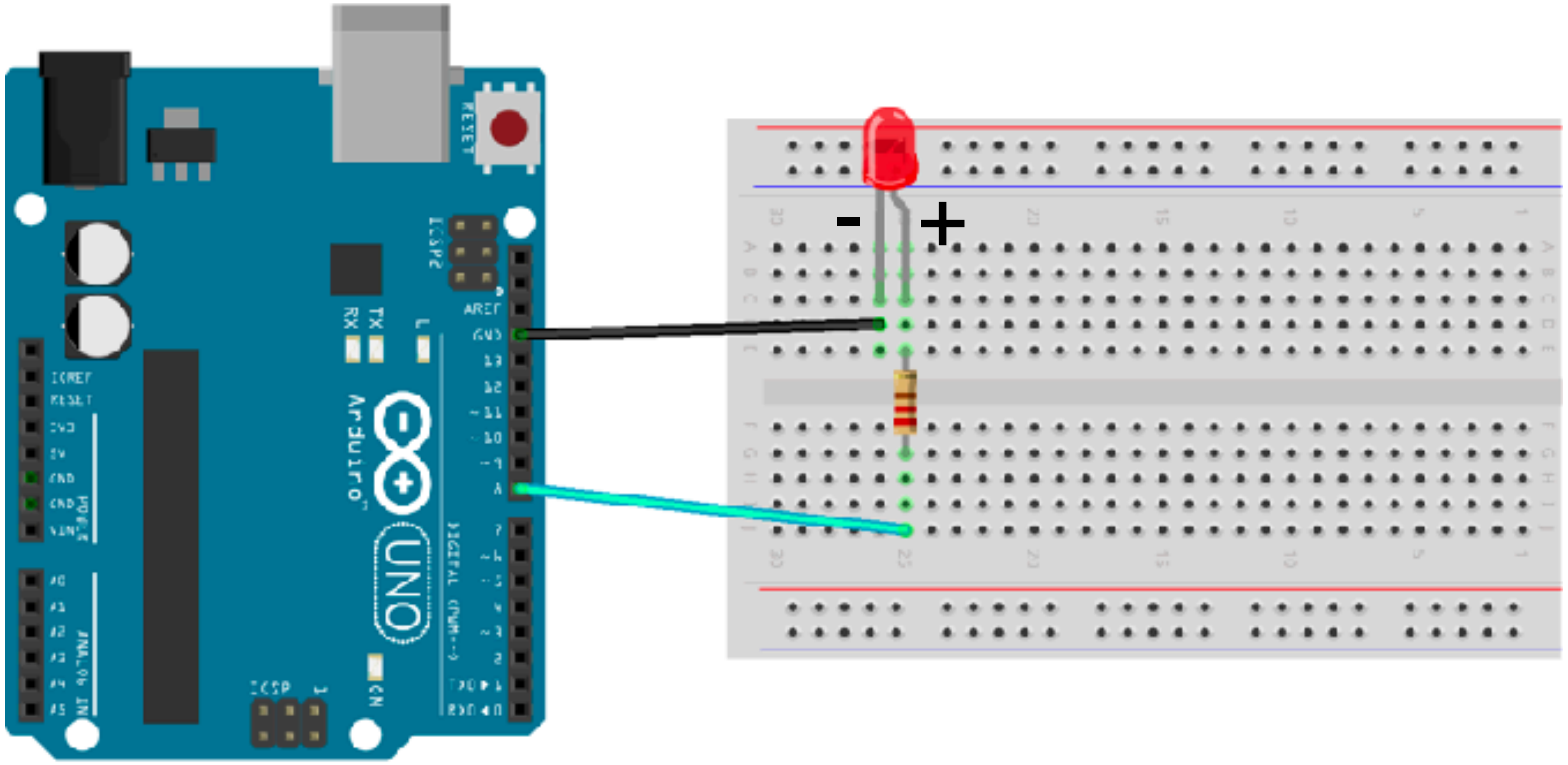


Blink!

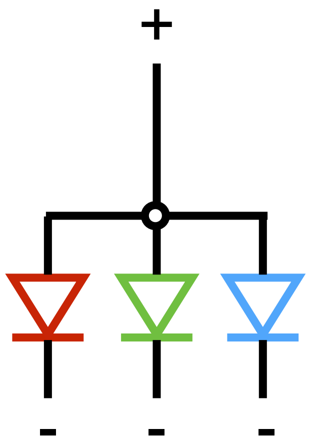


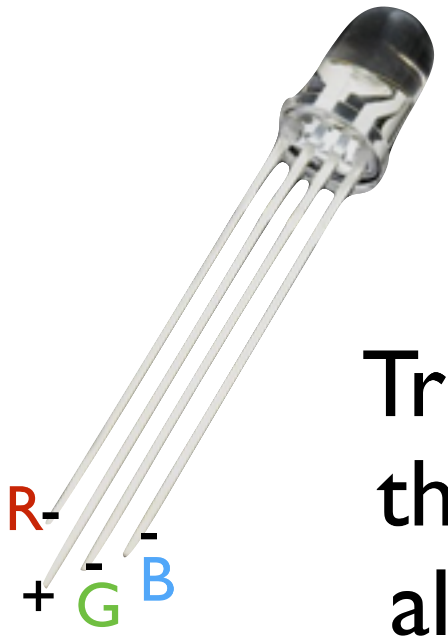


Try making the colors alternate!

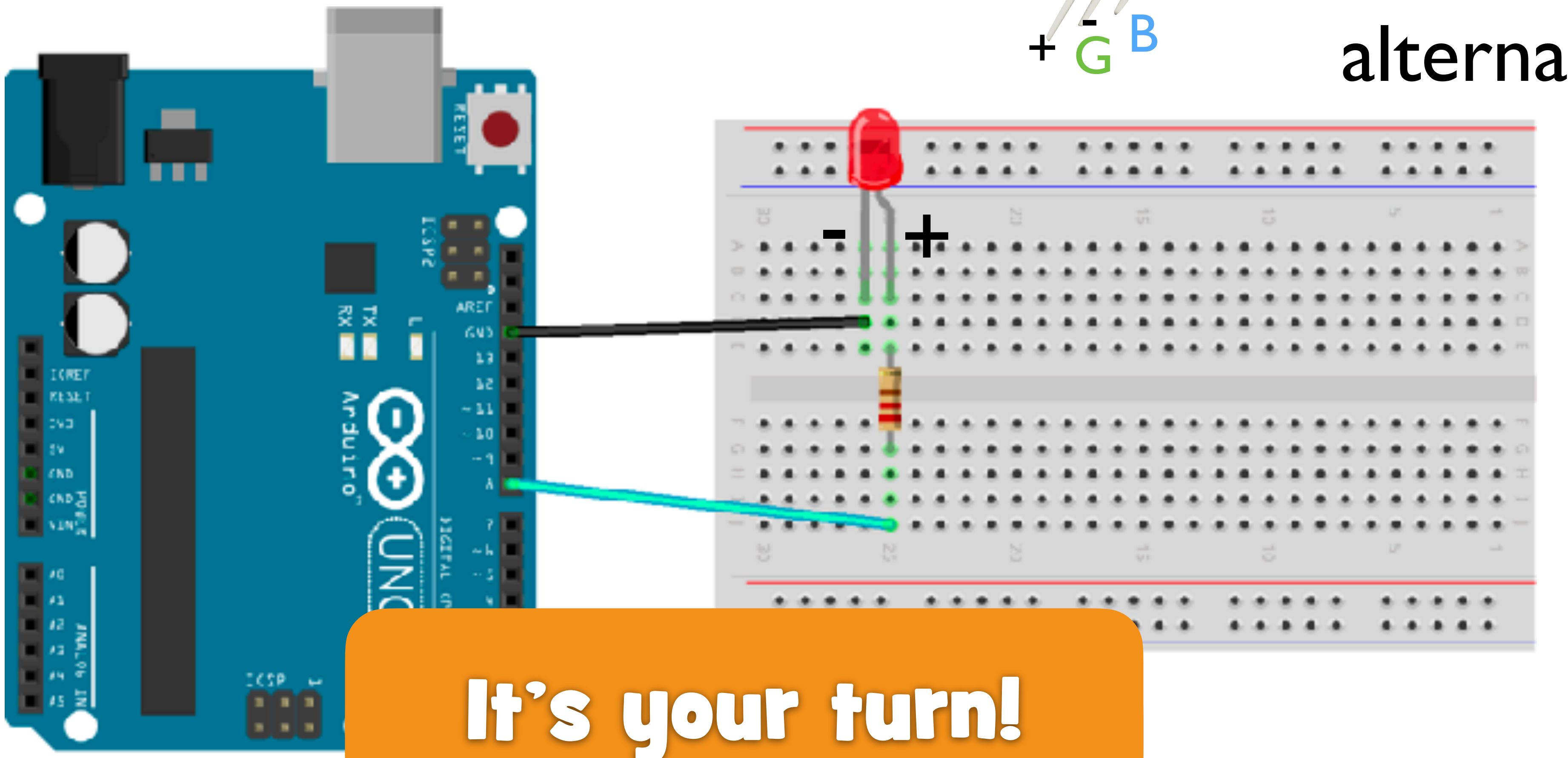


Blink!





Try making the colors alternate!

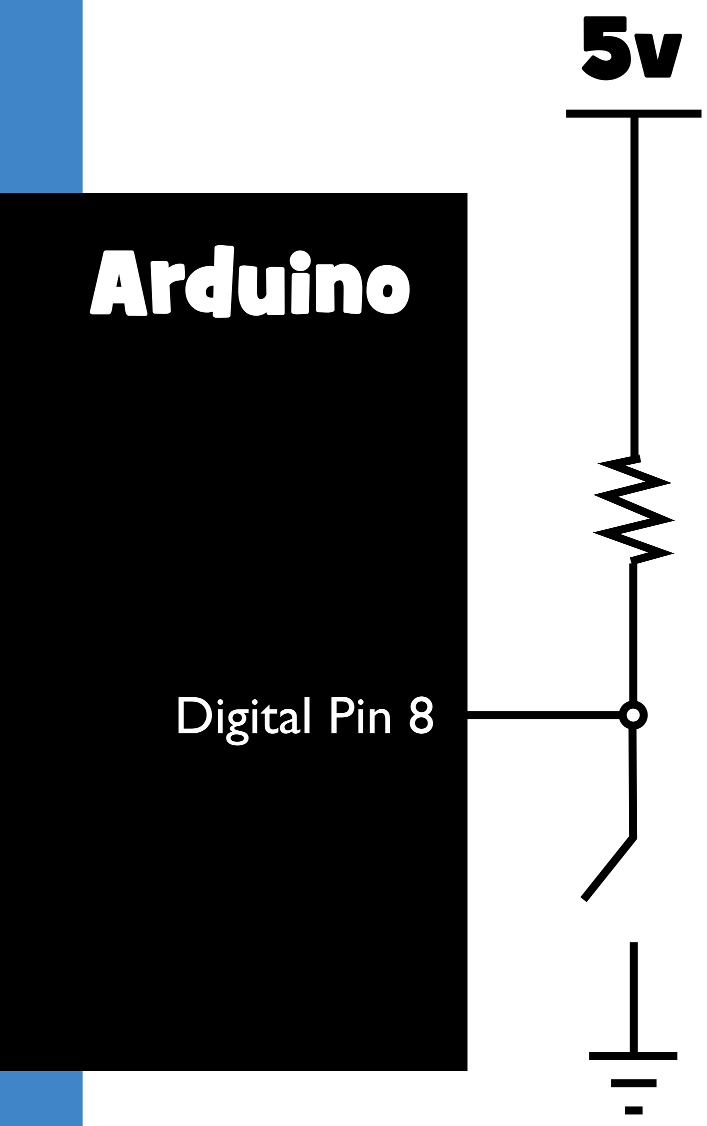


It's your turn!

`1_blink`

Turn the LED on for 1 seconds, and off for 2 seconds

Button Inputs



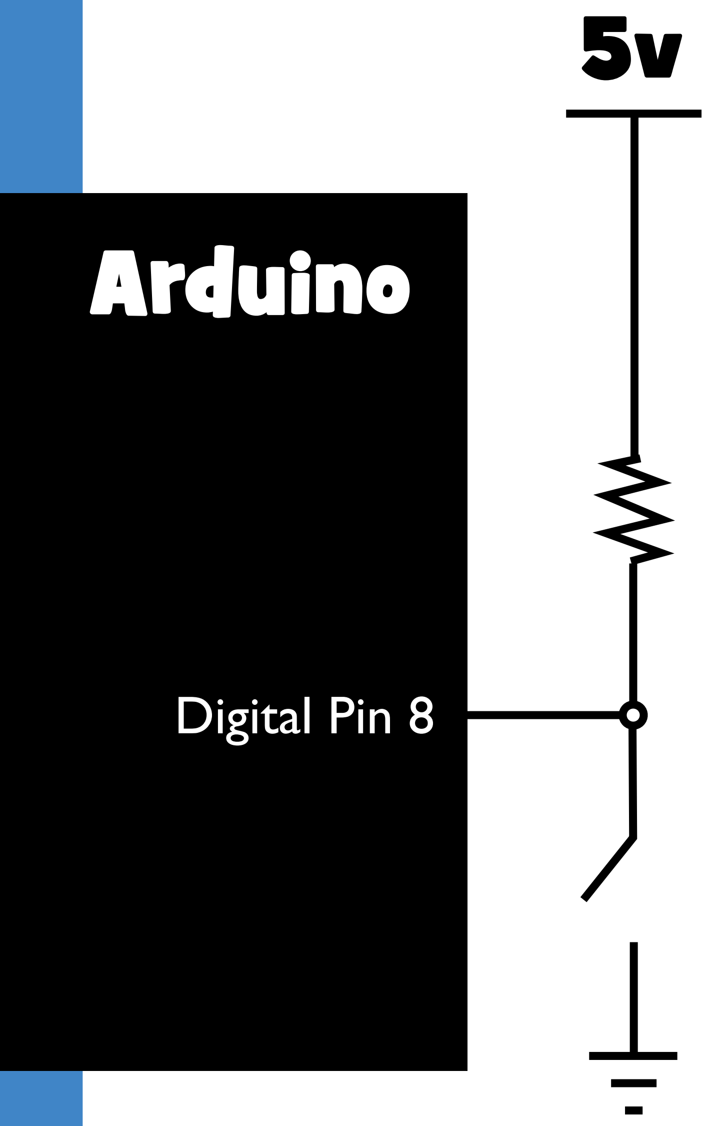
Big resistor!
> 1kΩ

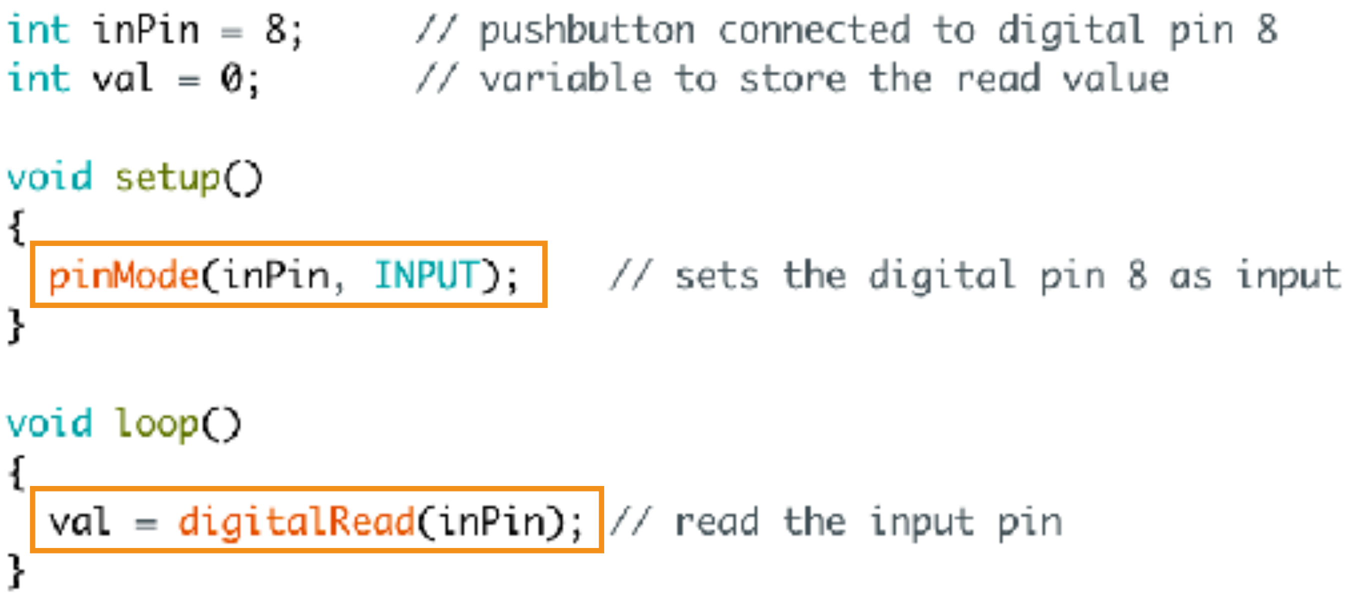
What does the Arduino sense when it’s not connected to GND?

Value is *floating!*

Pull up resistor!

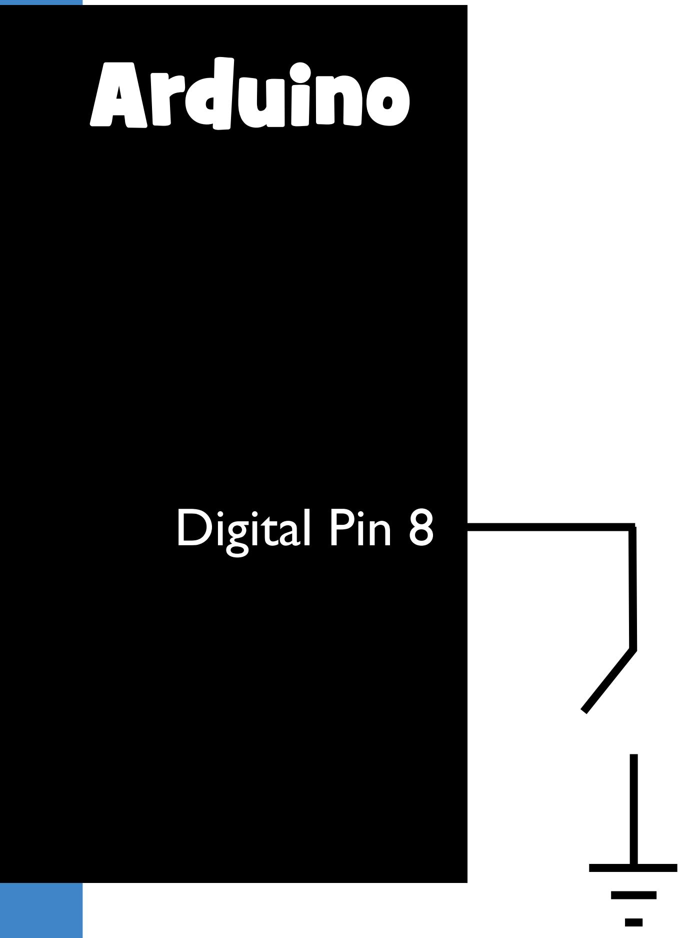
Button Inputs





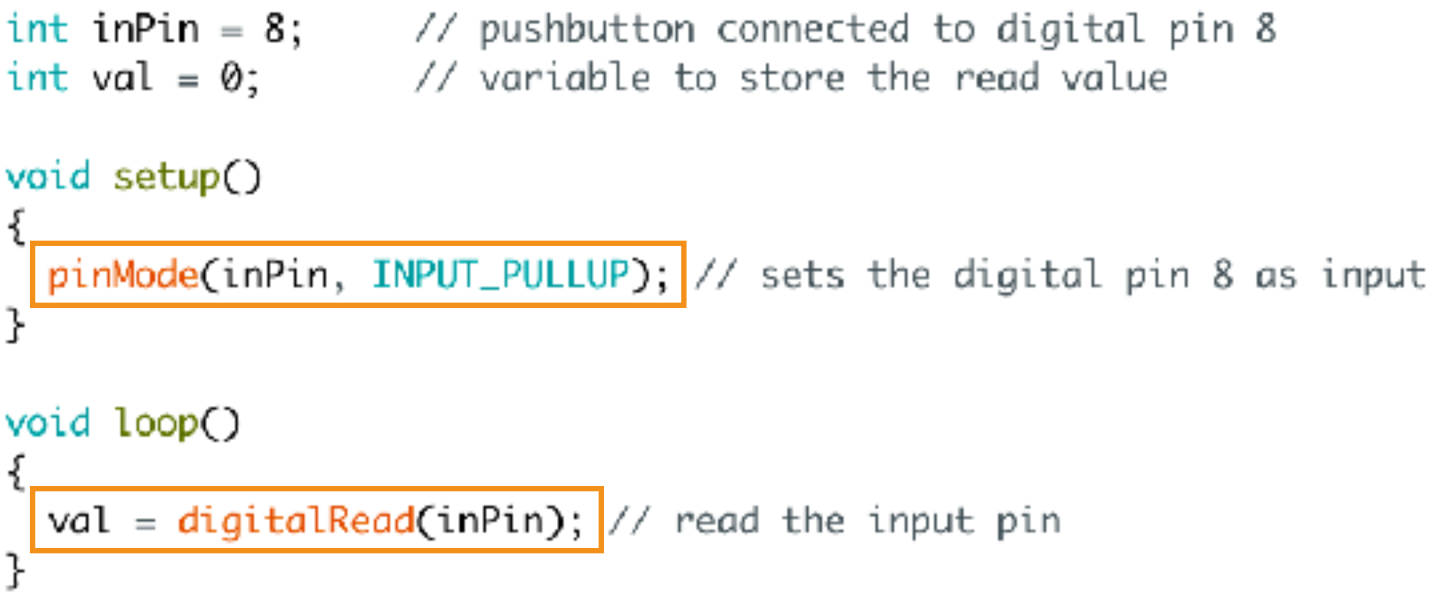
```
int inPin = 8;      // pushbutton connected to digital pin 8
int val = 0;        // variable to store the read value

void setup()
{
  pinMode(inPin, INPUT);    // sets the digital pin 8 as input
}

void loop()
{
  val = digitalRead(inPin); // read the input pin
}
```

Button Inputs



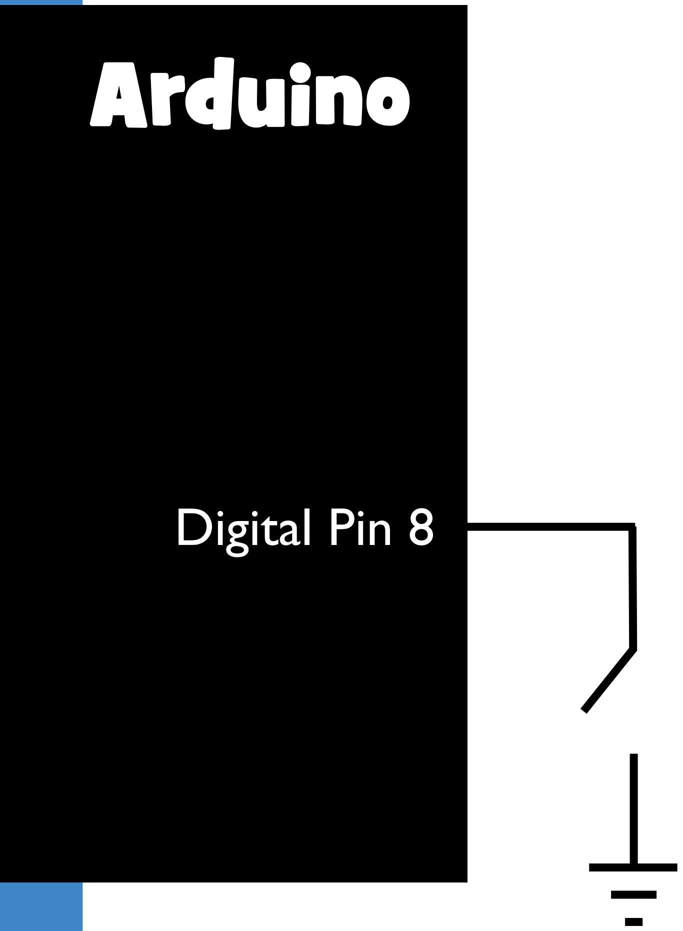
Arduino

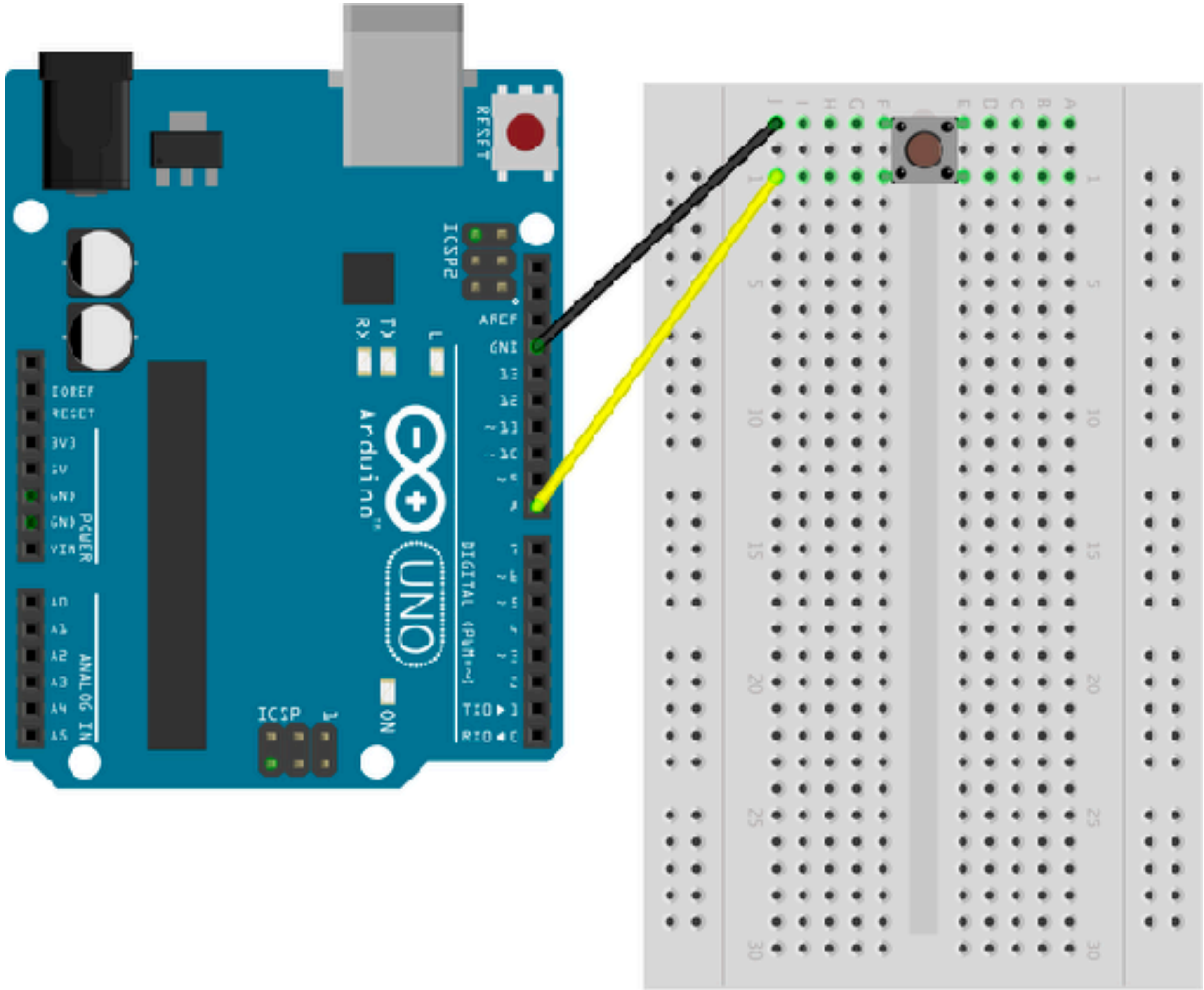
Digital Pin 8



```
int inPin = 8;      // pushbutton connected to digital pin 8
int val = 0;        // variable to store the read value

void setup()
{
  pinMode(inPin, INPUT_PULLUP); // sets the digital pin 8 as input
}

void loop()
{
  val = digitalRead(inPin); // read the input pin
}
```

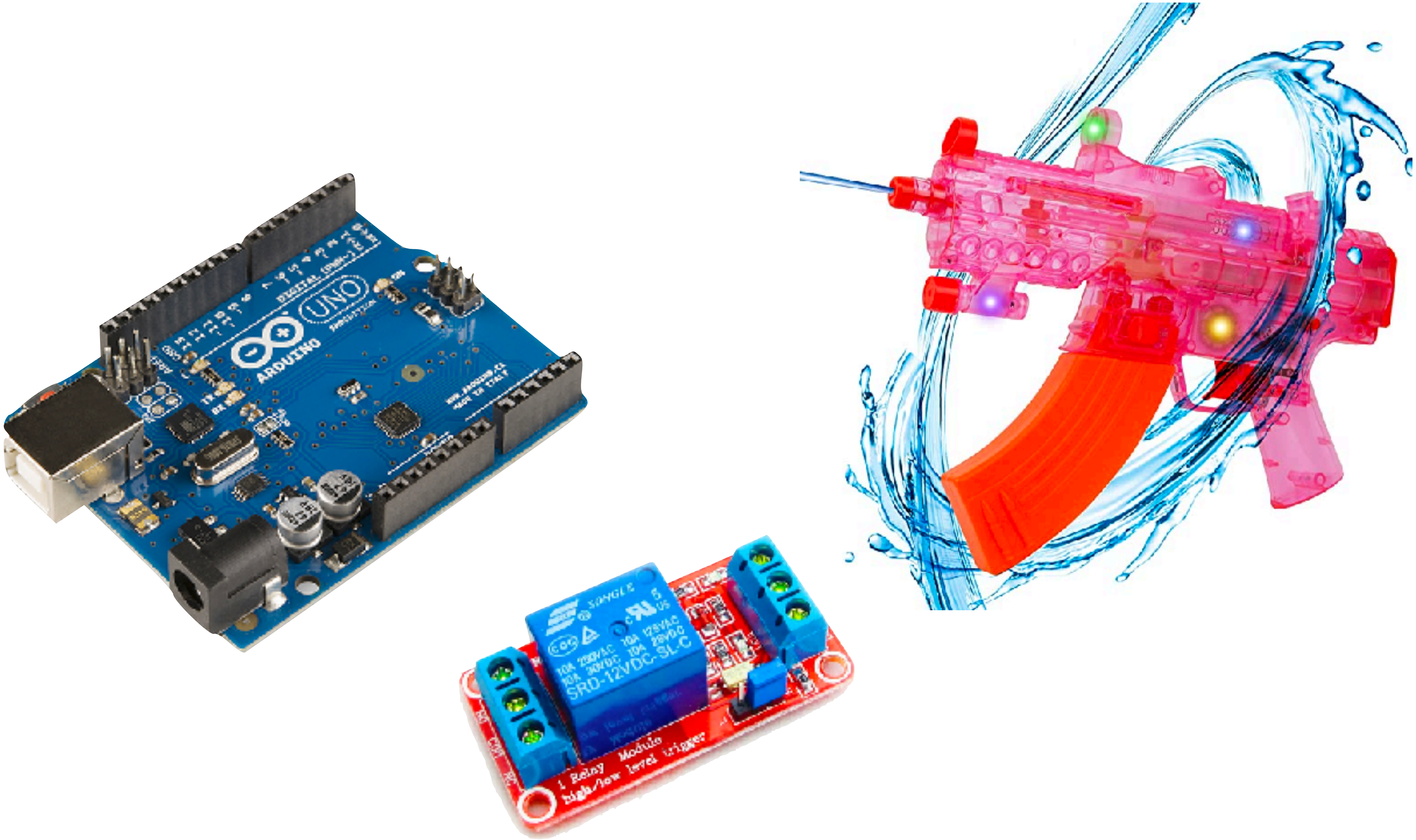
Button Inputs



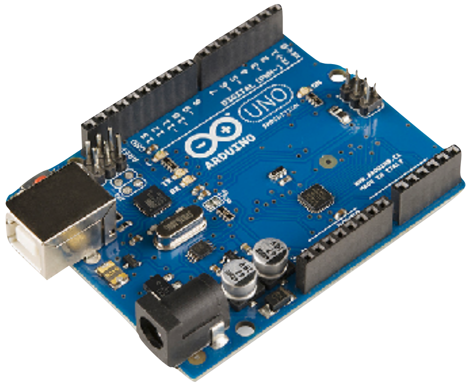


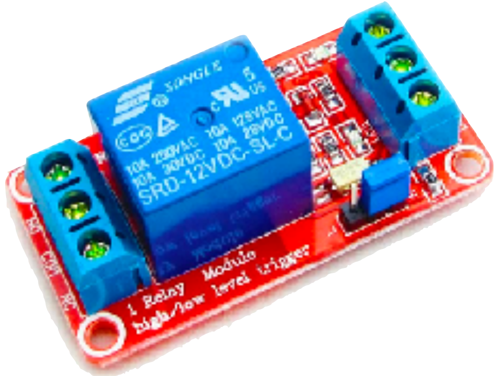
Button Press

Arduino Triggered

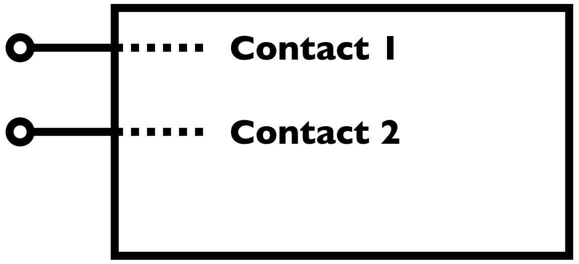


Arduino Triggered





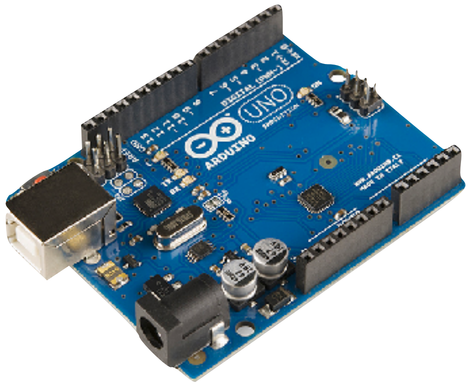
Watergun

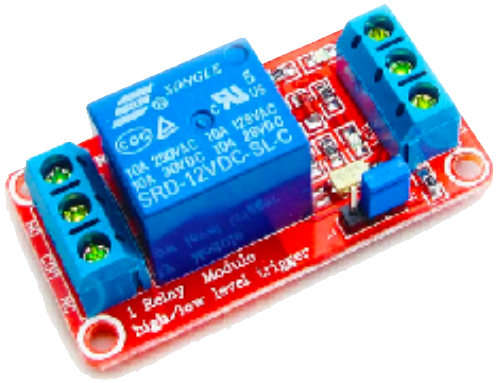


Contact 1

Contact 2

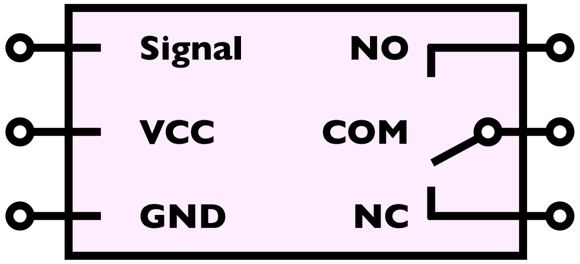
Arduino Triggered





Relay

Normally Open



Signal

VCC

GND

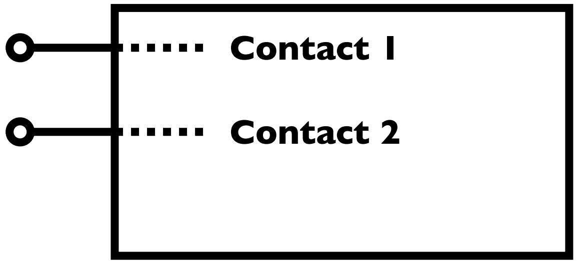
NO

COM

NC

Normally Closed

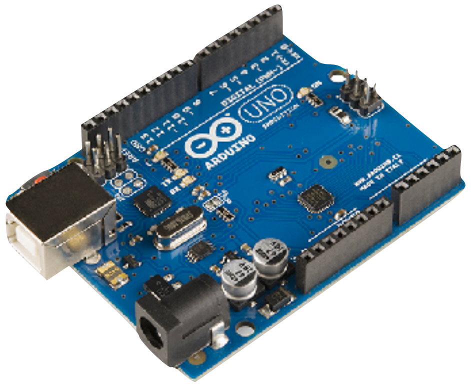
Watergun

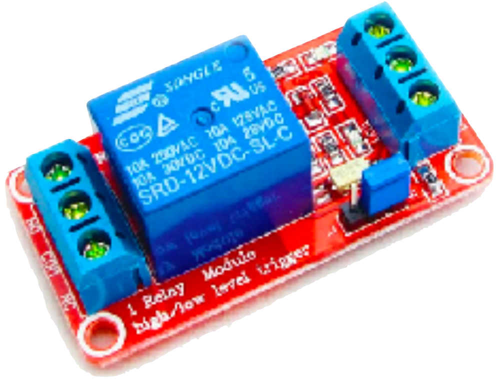


Contact 1

Contact 2

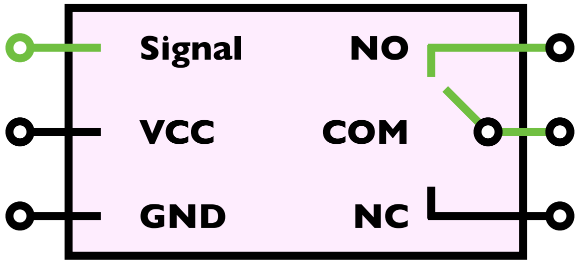
Arduino Triggered





Normally Open

Relay



Signal

VCC

GND

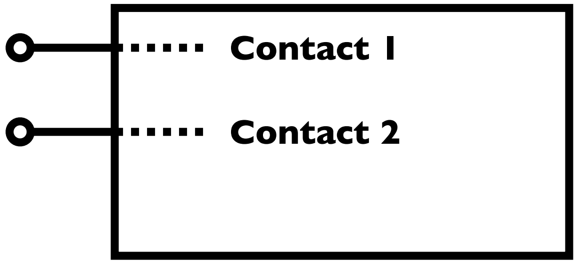
NO

COM

NC

Normally Closed

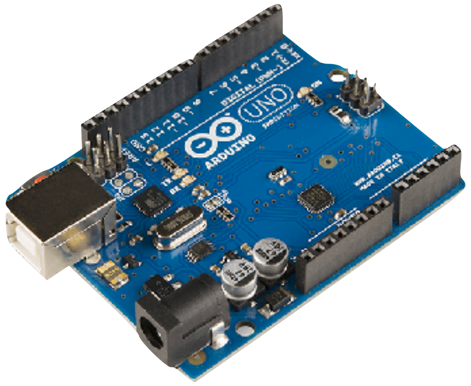
Watergun

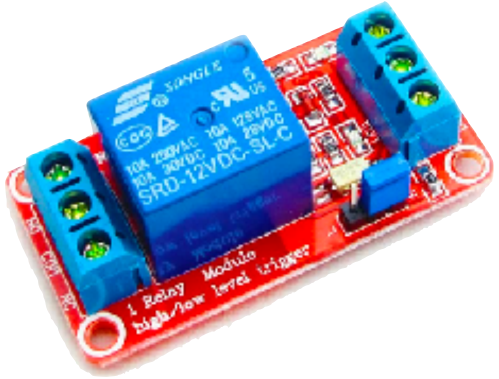


Contact 1

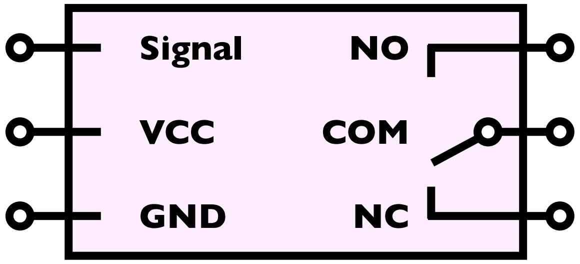
Contact 2

Arduino Triggered





Relay



Signal

VCC

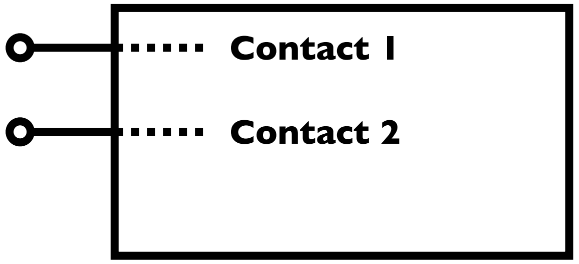
GND

NO

COM

NC

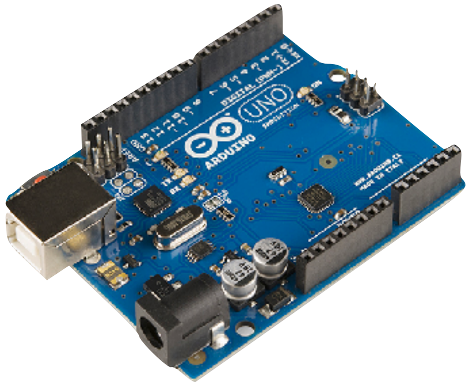
Watergun

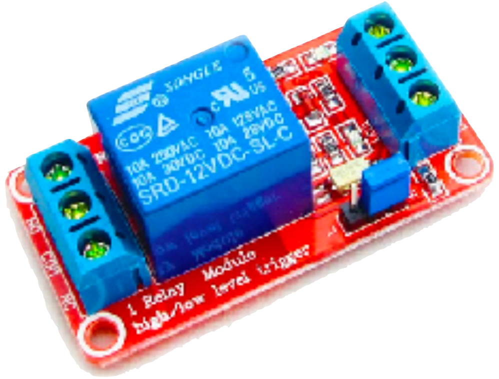


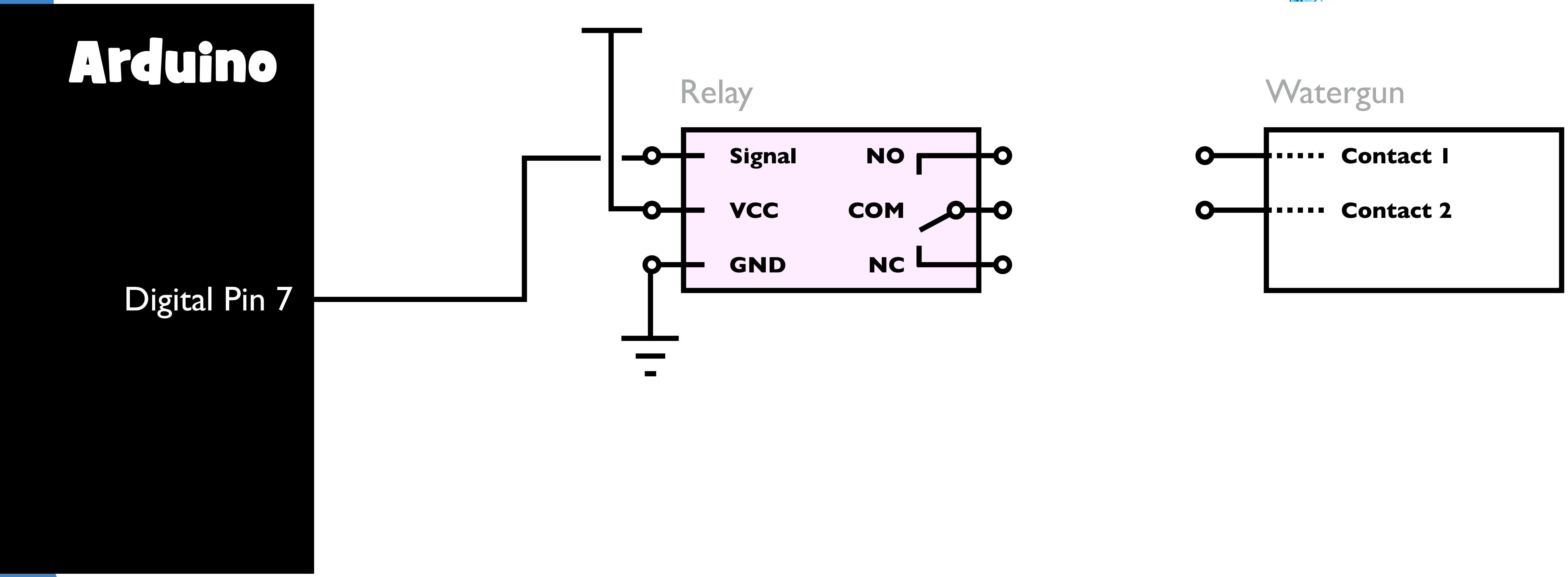
Contact 1

Contact 2

Arduino Triggered







Arduino

Digital Pin 7

Relay

Signal

VCC

GND

NO

COM

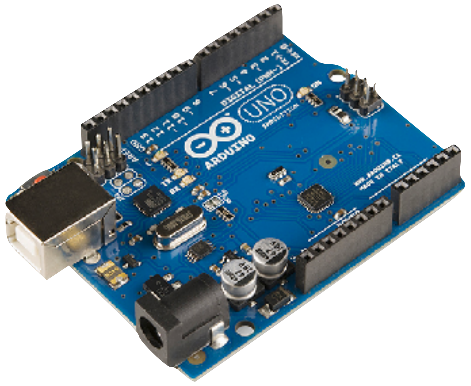
NC

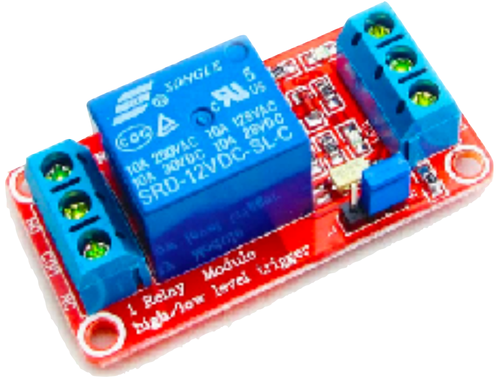
Watergun

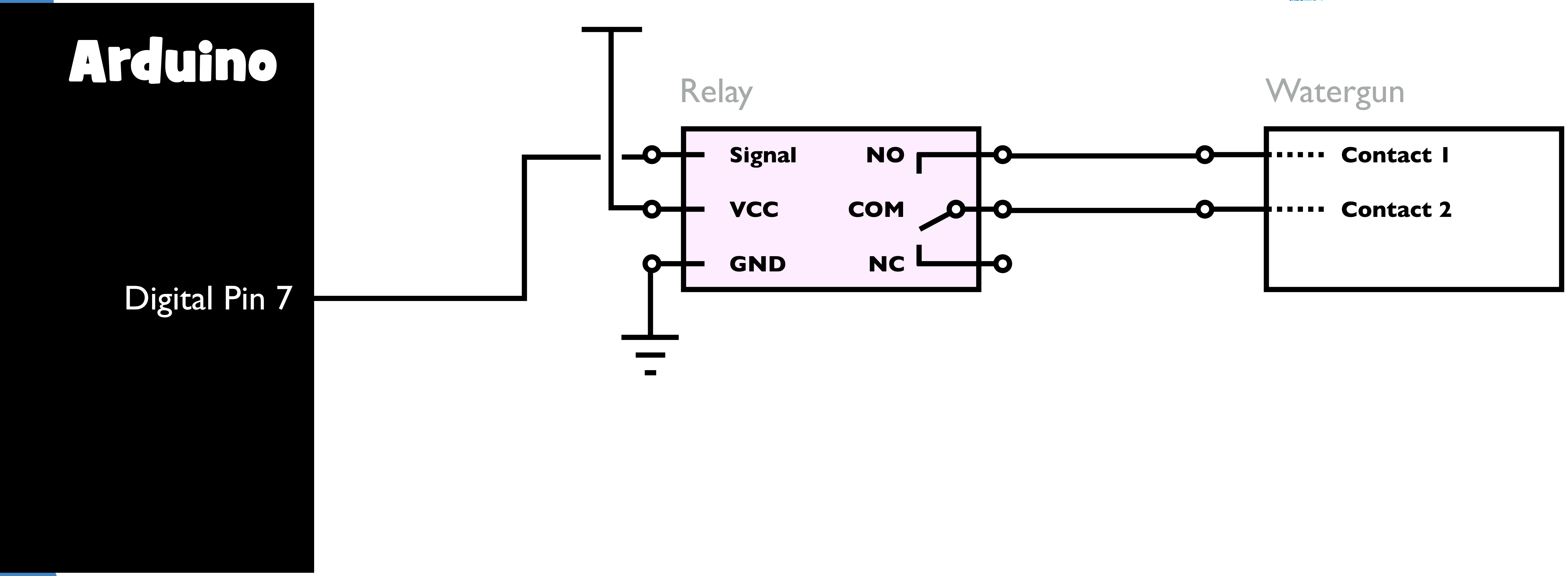
Contact 1

Contact 2

Arduino Triggered







Arduino

Digital Pin 7

Relay

Signal

VCC

GND

NO

COM

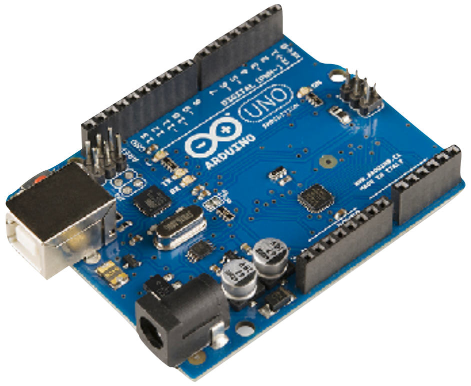
NC

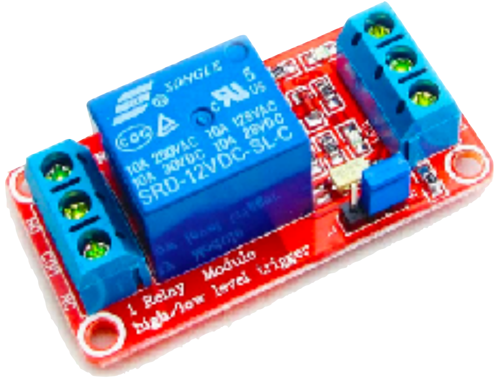
Watergun

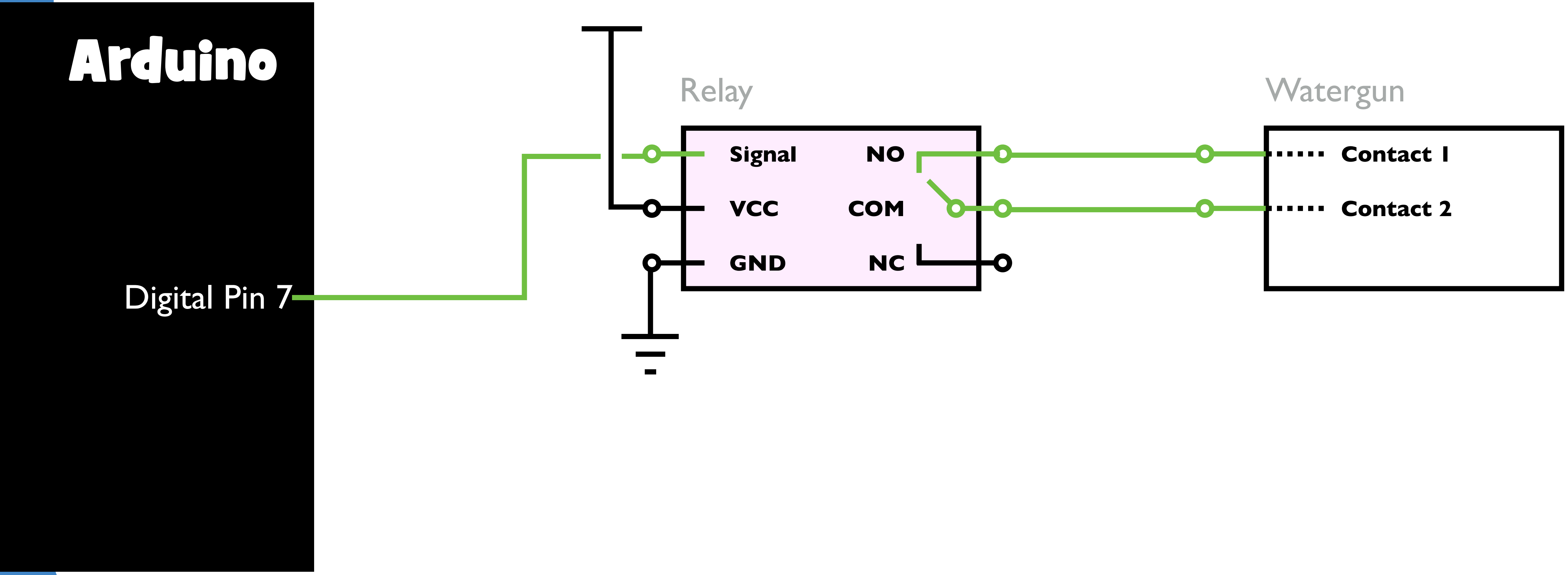
Contact 1

Contact 2

Arduino Triggered







Arduino

Digital Pin 7

Relay

Signal

VCC

GND

NO

COM

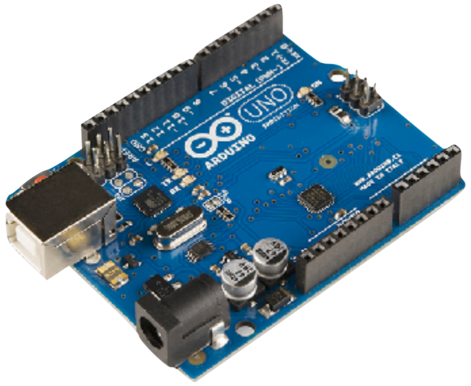
NC

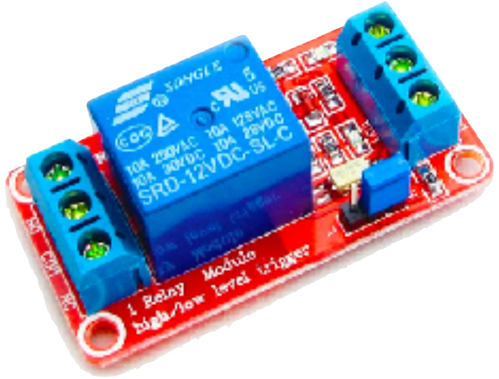
Watergun

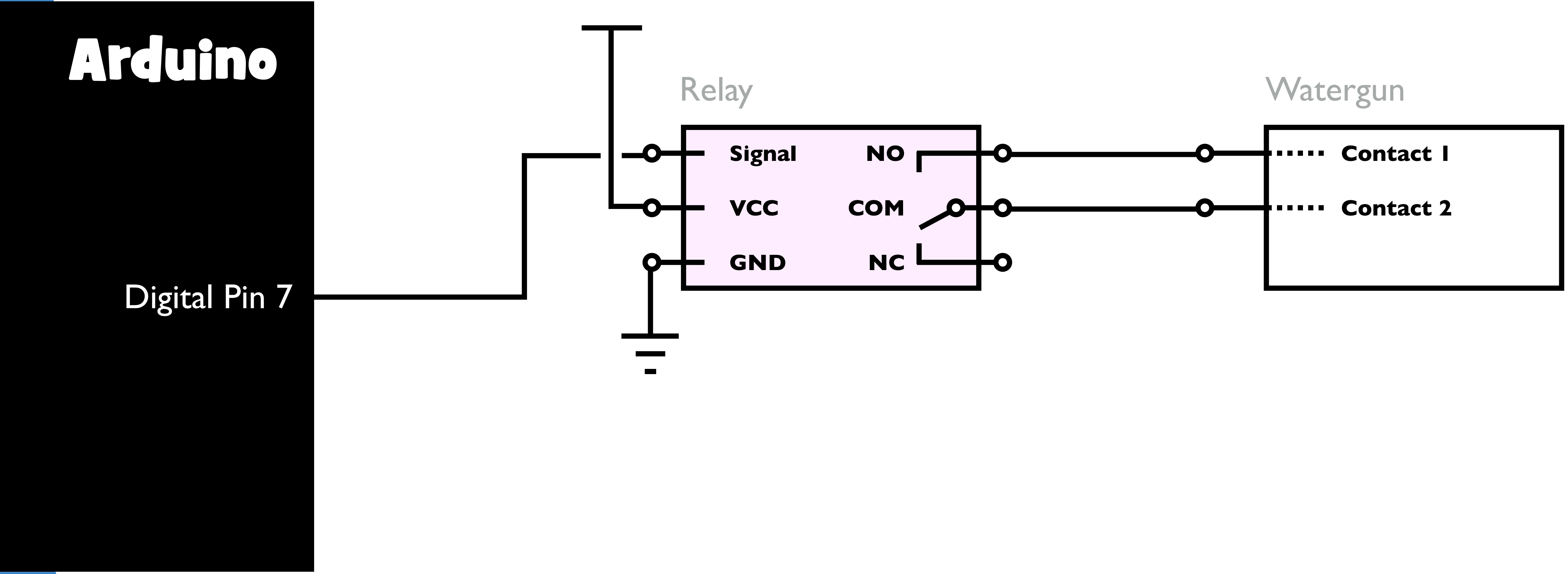
Contact 1

Contact 2

Arduino Triggered







Arduino

Digital Pin 7

Relay

Signal

VCC

GND

NO

COM

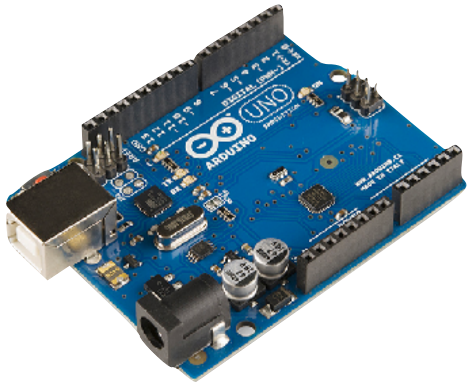
NC

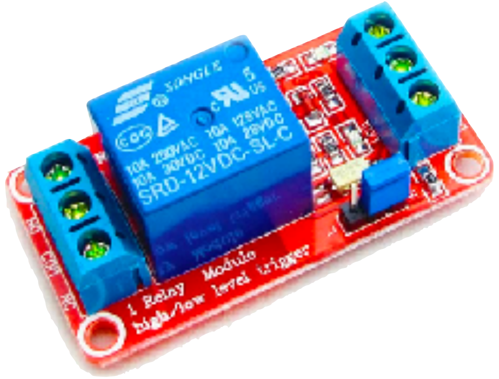
Watergun

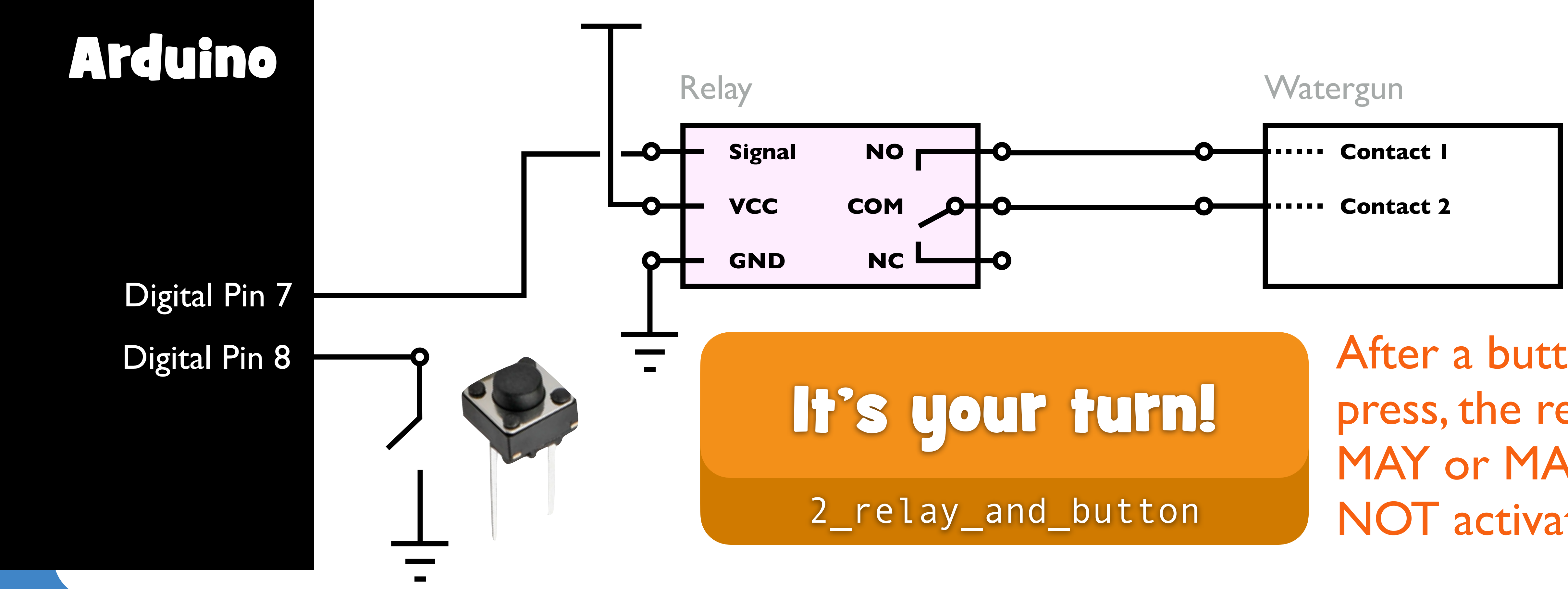
Contact 1

Contact 2

Arduino Triggered





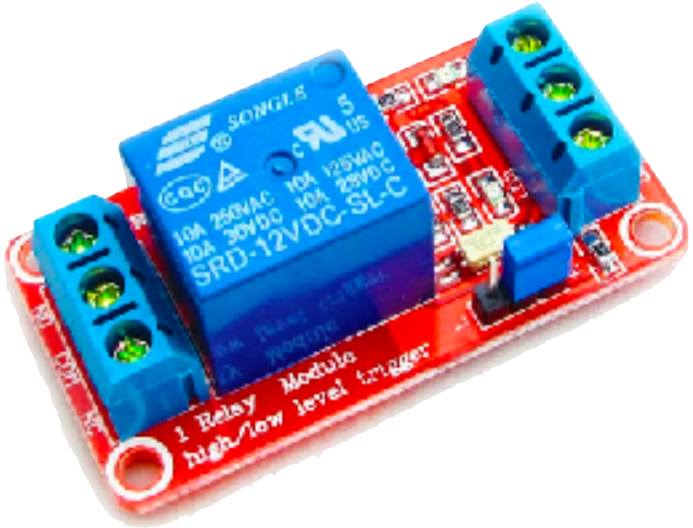


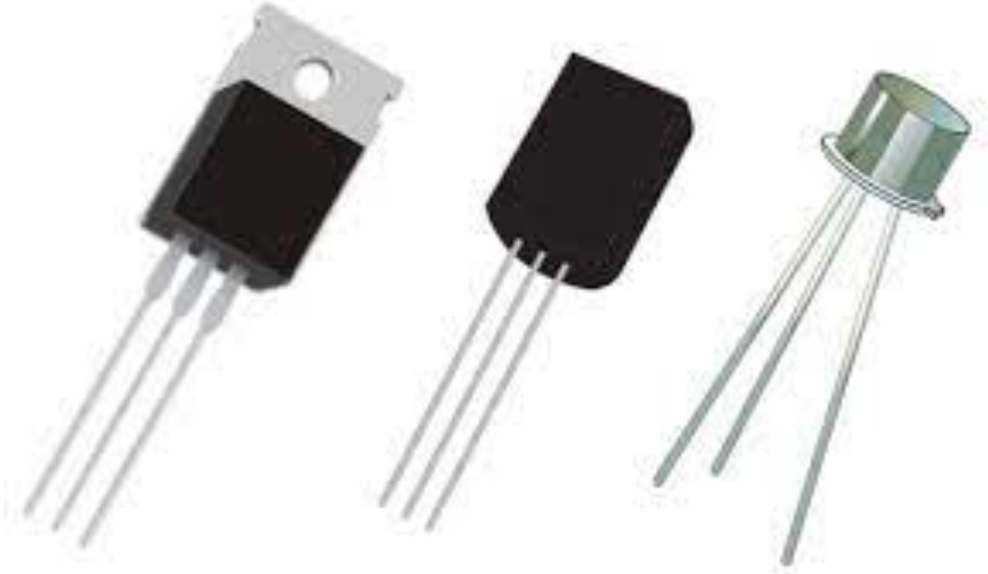
It's your turn!

`2_relay_and_button`

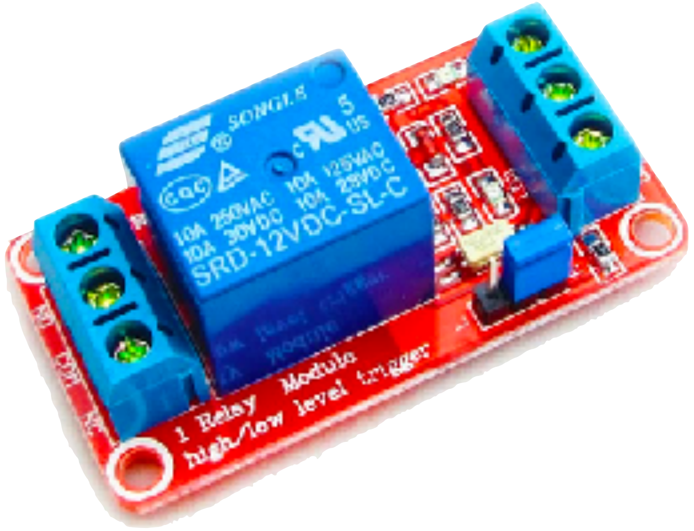
After a button press, the relay MAY or MAY NOT activate

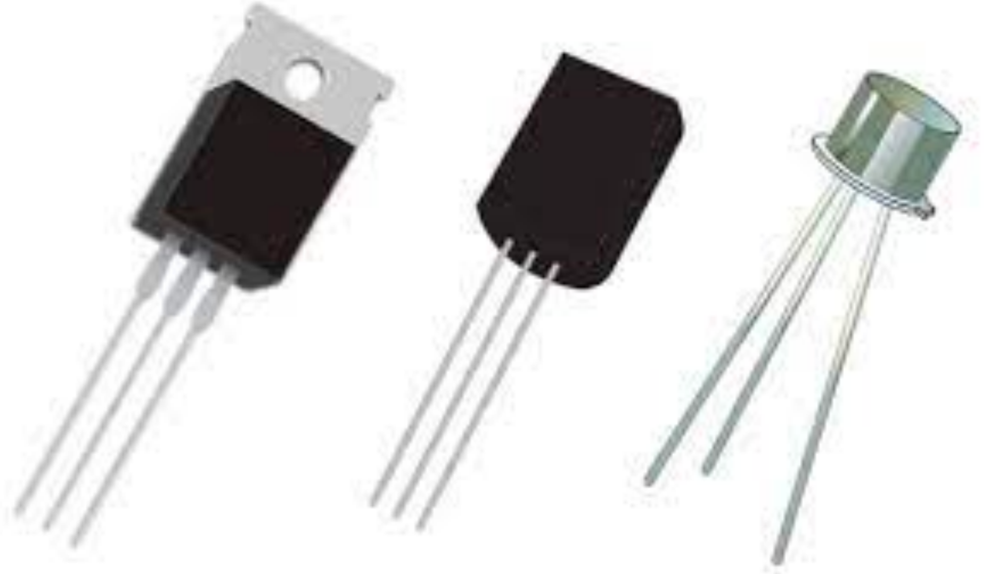
Relays & Transistors





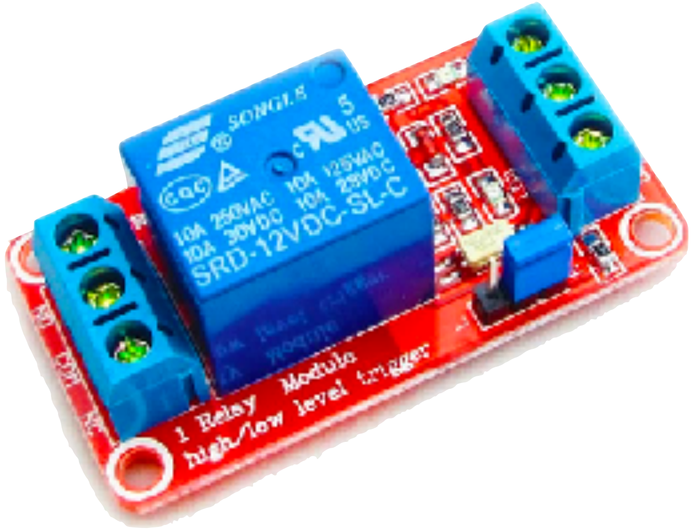
Relays & Transistors

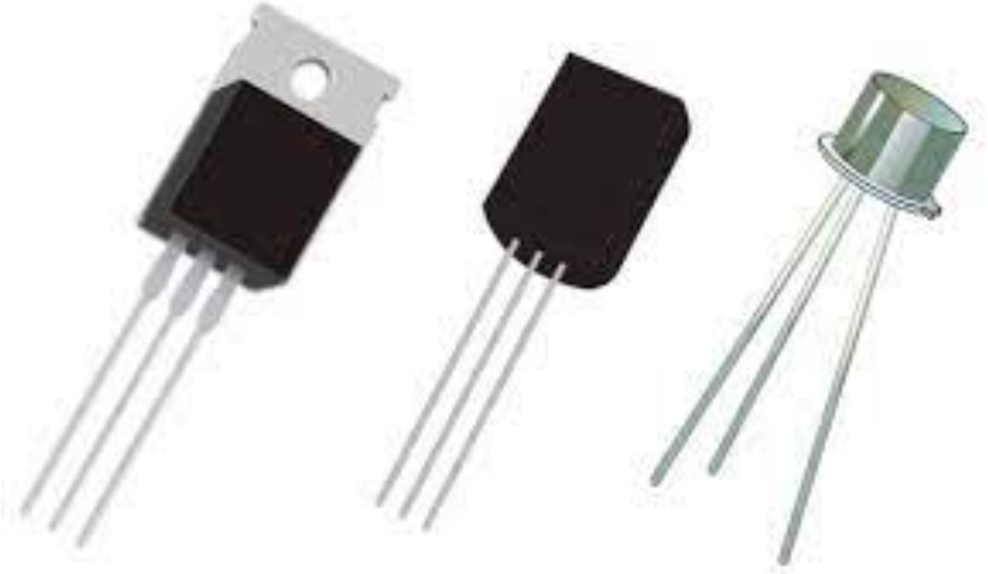




Smallest transistors are 2 nanometers (IBM - 2021)

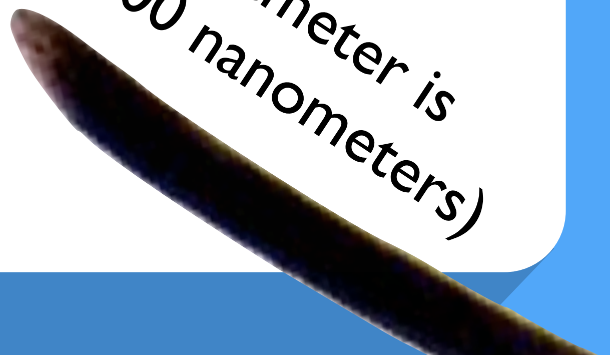
Relays & Transistors





Smallest transistors are
2 nanometers (IBM - 2021)

(Hair diameter is
65,000 nanometers)

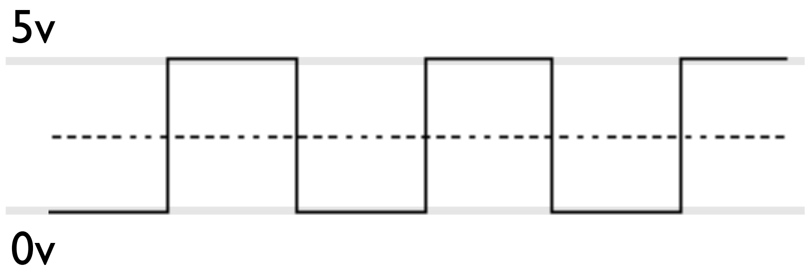


Pulse Width Modulation

PWM!

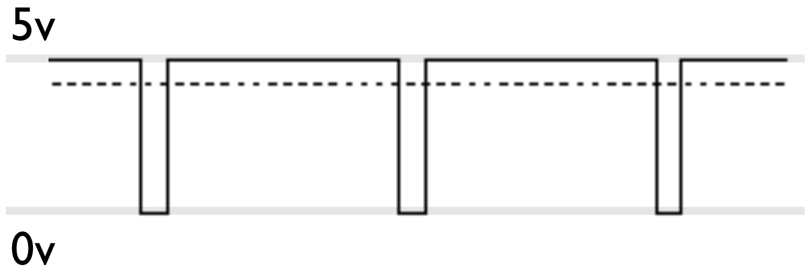
Digital is only 0v or 5v, so how do we get values in between?

PWM to make the LED seem ‘dimmer’



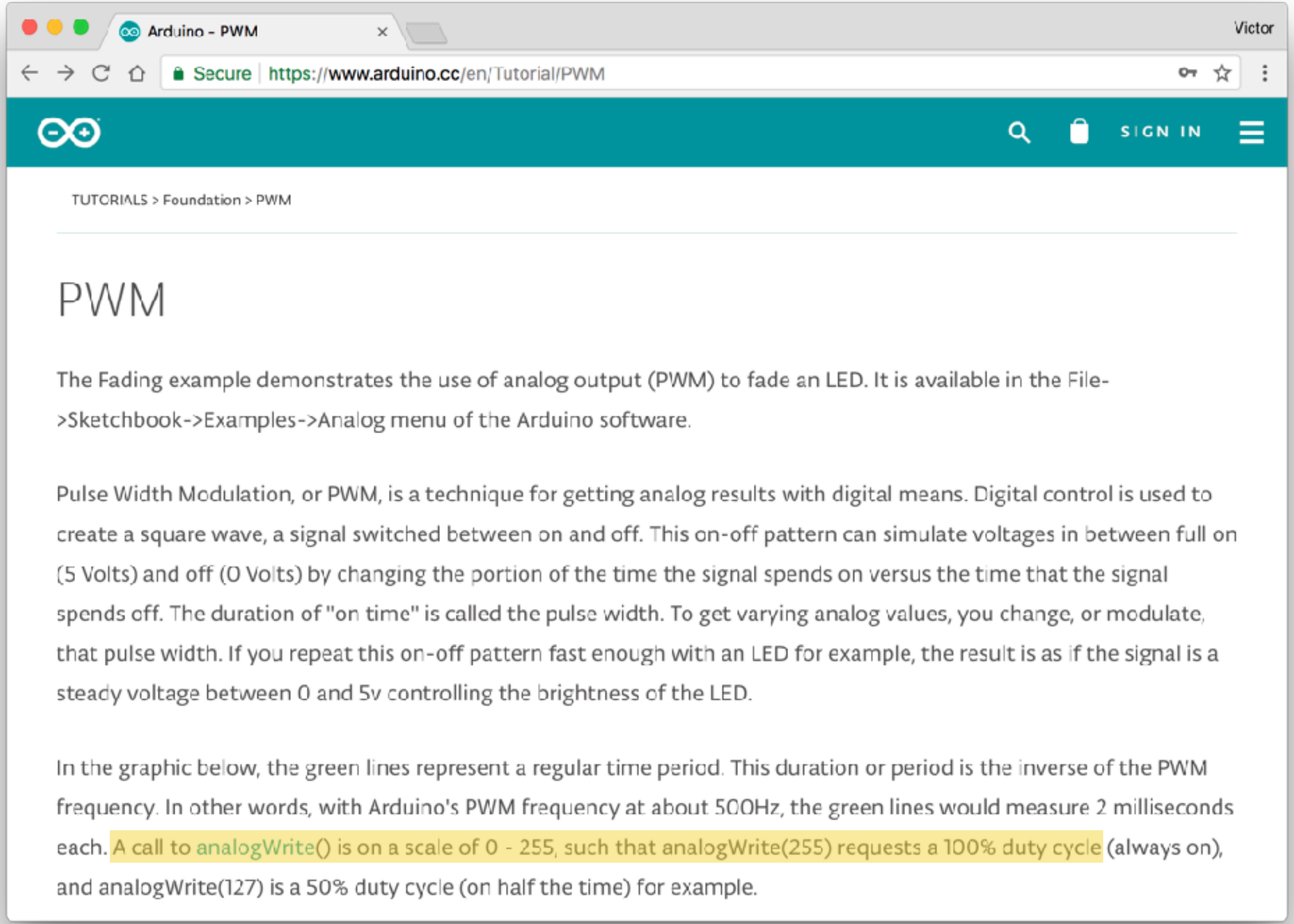
50% Duty Cycle

50% of the time on,
50% of the time off



90% Duty Cycle

90% of the time on,
10% of the time off



Arduino - PWM
Victor
Secure
https://www.arduino.cc/en/Tutorial/PWM
SIGN IN
TUTORIALS > Foundation > PWM

PWM

The Fading example demonstrates the use of analog output (PWM) to fade an LED. It is available in the File->Sketchbook->Examples->Analog menu of the Arduino software.

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.

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.

Servo and Moving Parts

Rotary actuator that allows for precise control of position

Arduino friendly!

Built-in Library

0 -180 Degrees

🟧 SIGNAL (Pin 6)

🟥 POWER (5V)

🟫 GND

Continuous Rotation Servo

Simple ‘motors’

Don’t allow you to specify the exact location, but can rotate CCW or CW at different speeds.

0 - 180 Degrees becomes:
CCW full speed,
stationary,
CW full speed

Continuous Rotation Servo

Simple ‘motors’

Don’t allow you to specify the exact location, but can rotate CCW or CW at different speeds.

0 - 180 Degrees becomes:
CCW full speed,
stationary,
CW full speed

🟧 SIGNAL (Pin 6)
🟥 POWER (5V)
🟫 GND

It’s your turn!

3_servo

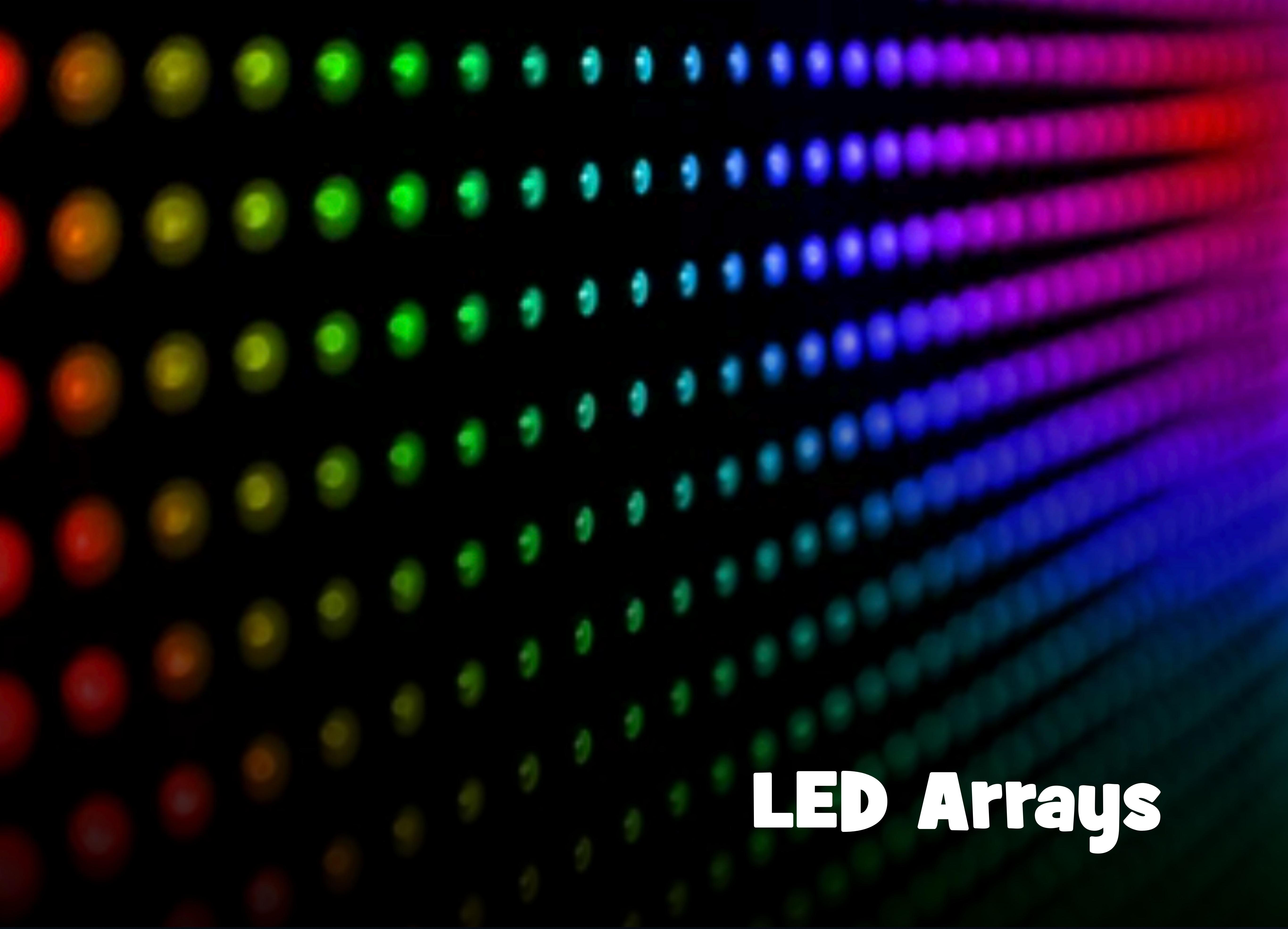
Servos

Two Types!

1. Standard Servos (plenty in lab)

2. Continuous Rotation Servos (in your kits)

LED Arrays



More Pins?

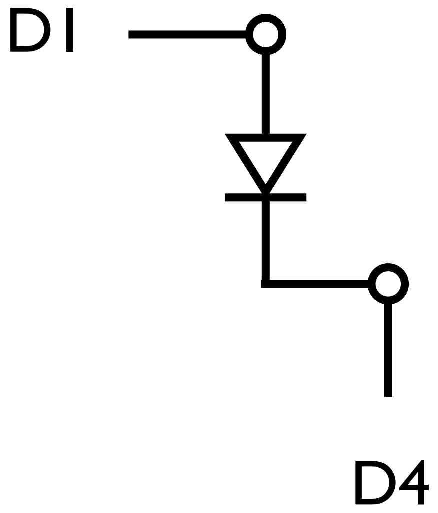
Arduinos only have a limited number of output.

There are different methods we can “get more outputs”

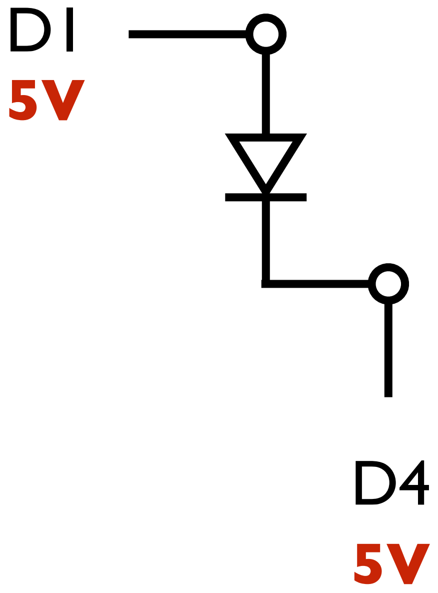
Method 1: **Multiplexing**

Method 2: Specific **Communication** Protocols

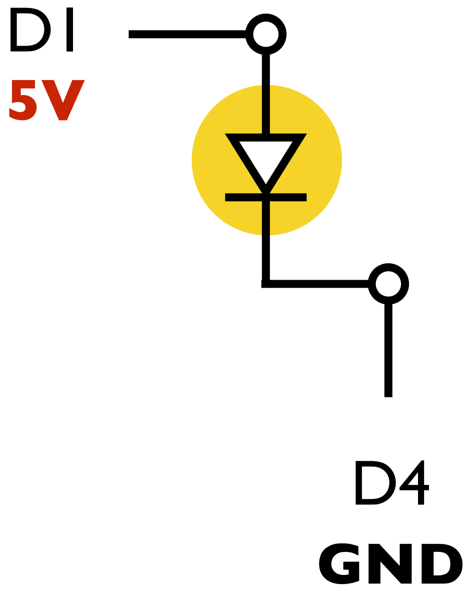
Multiplexing



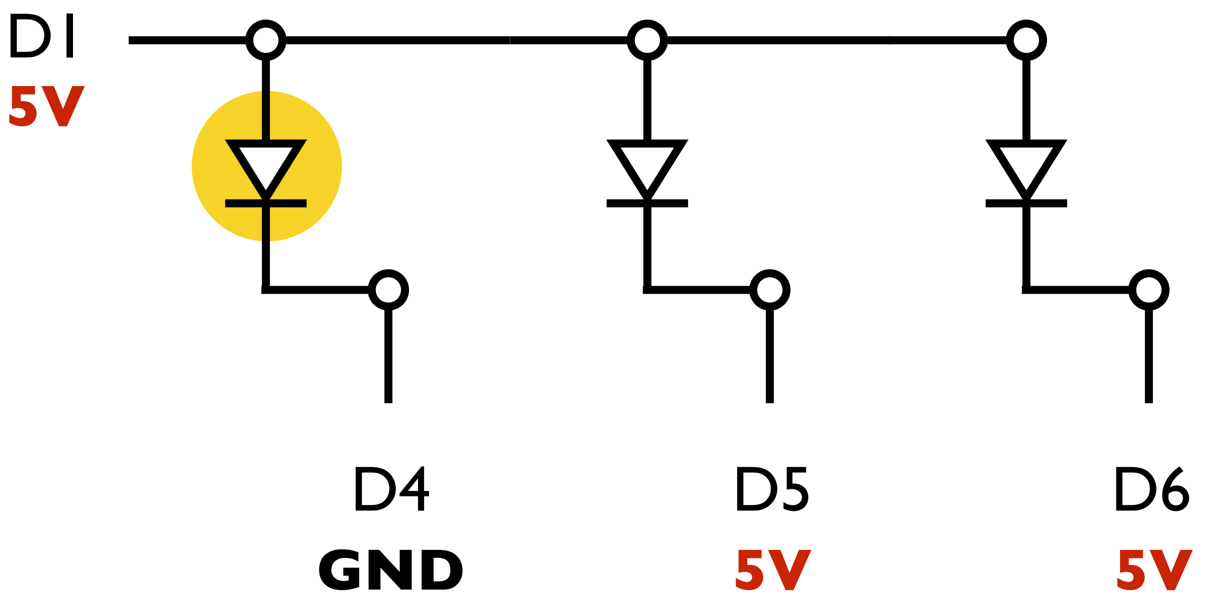
Multiplexing



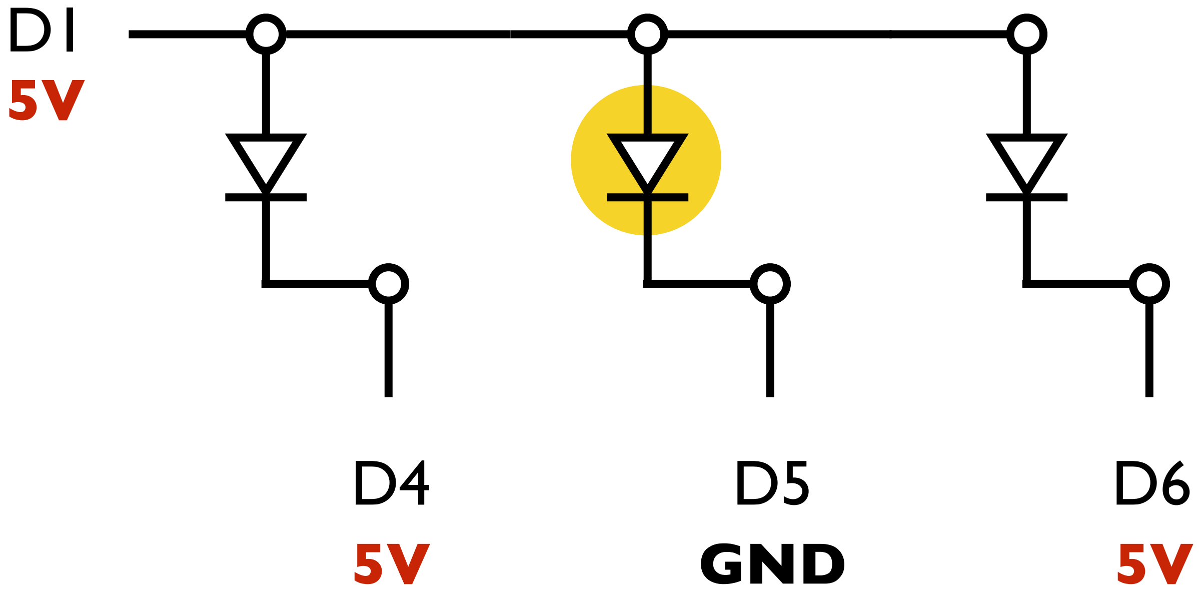
Multiplexing



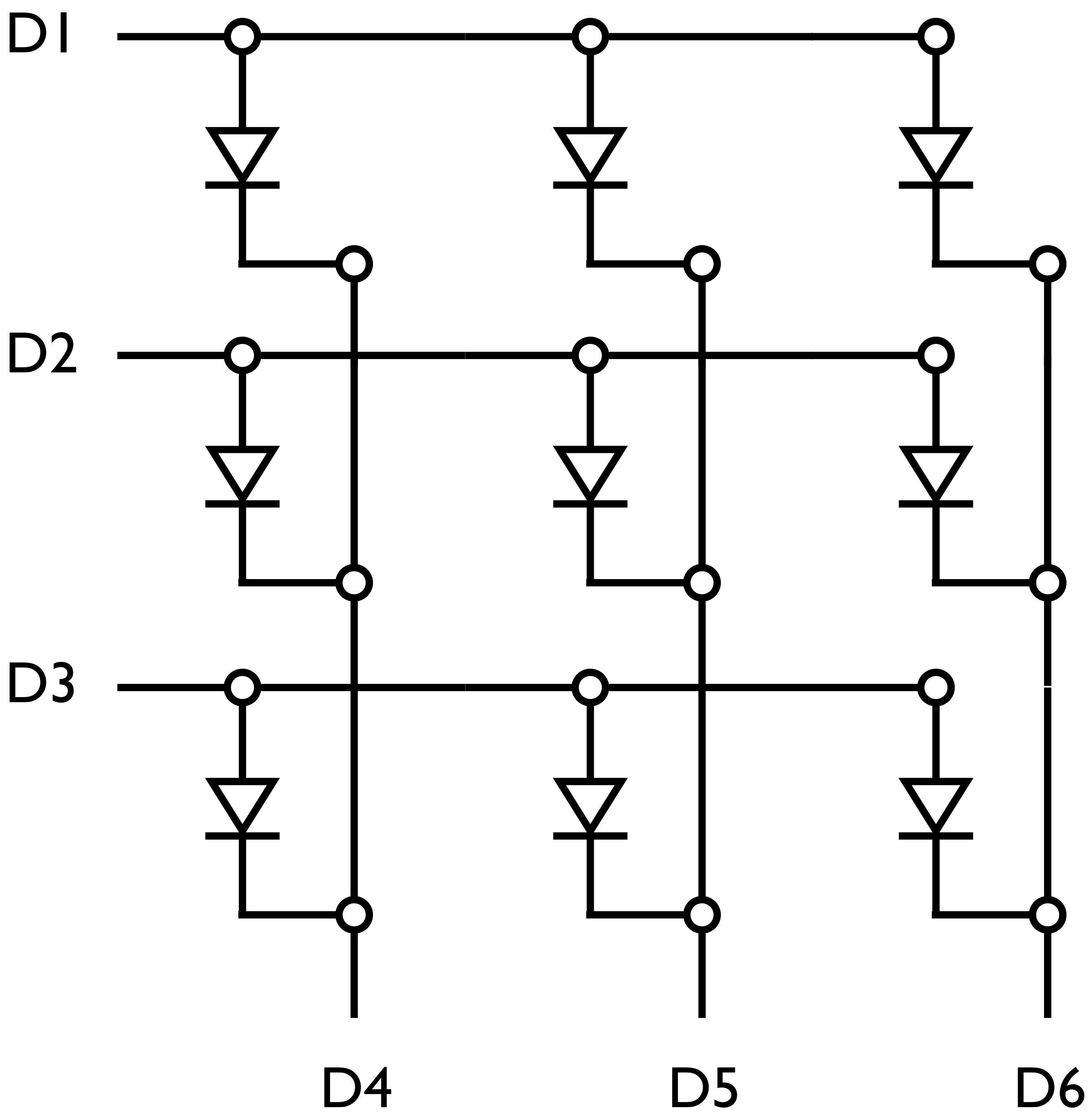
Multiplexing



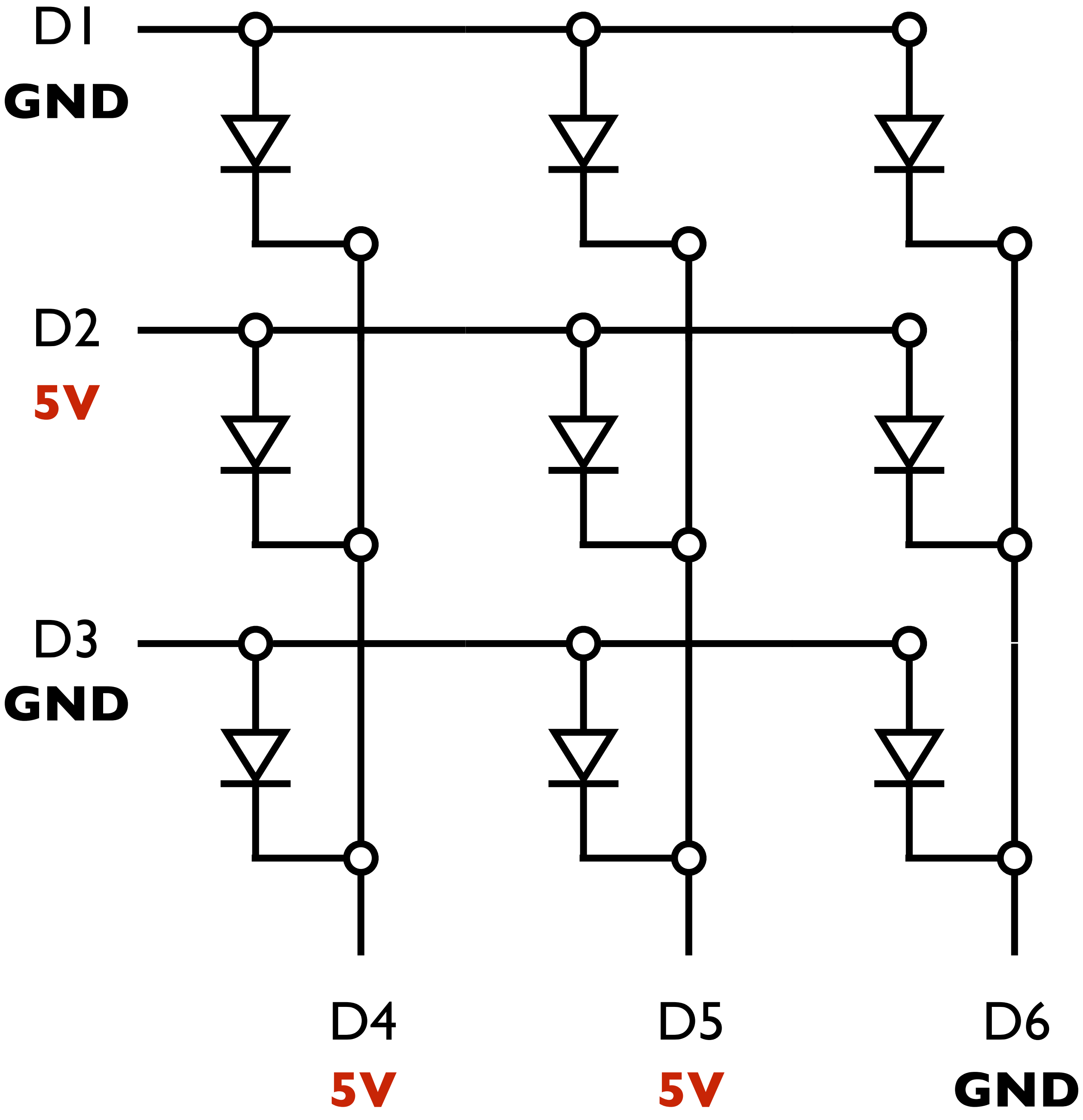
Multiplexing



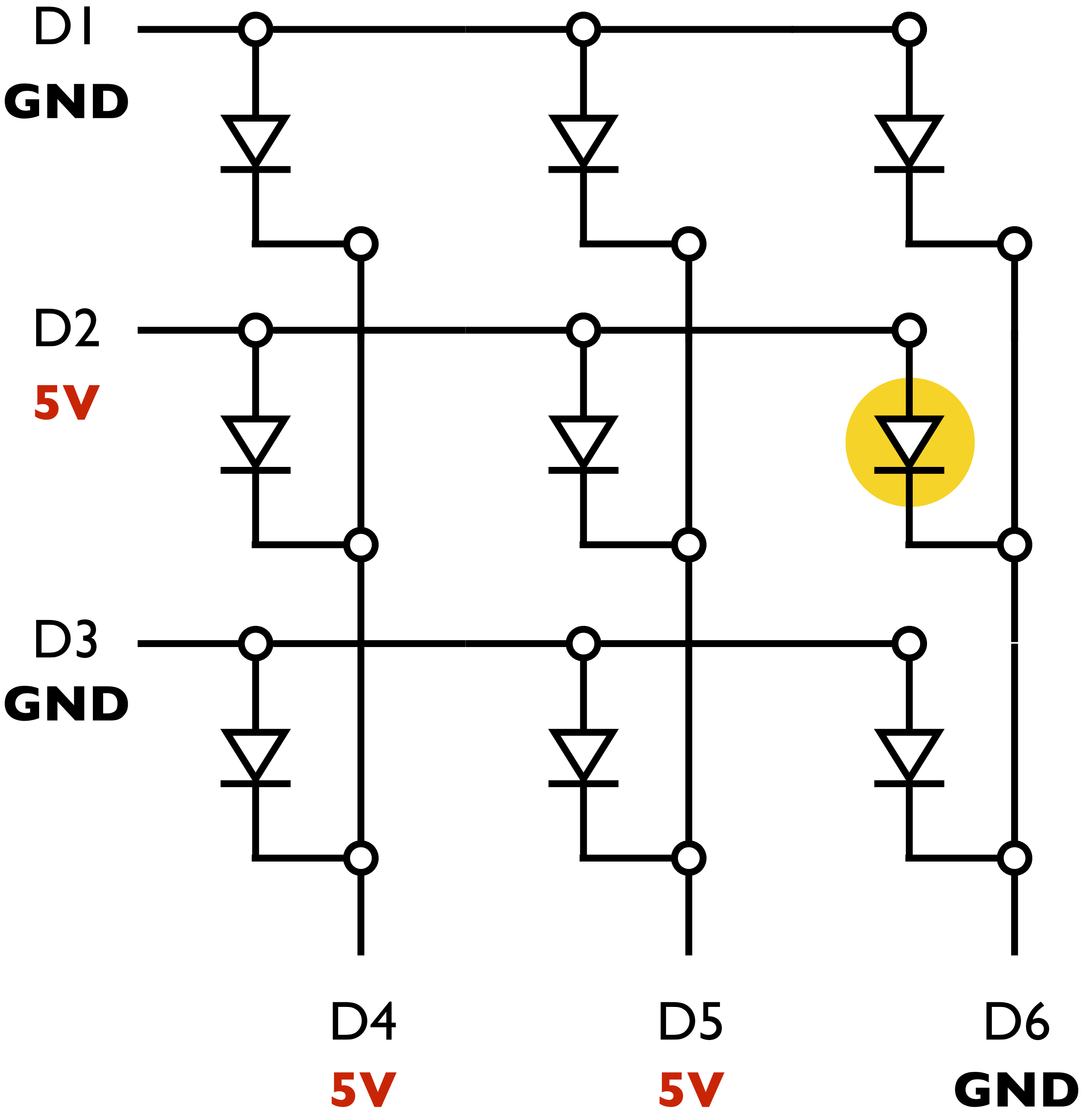
Multiplexing



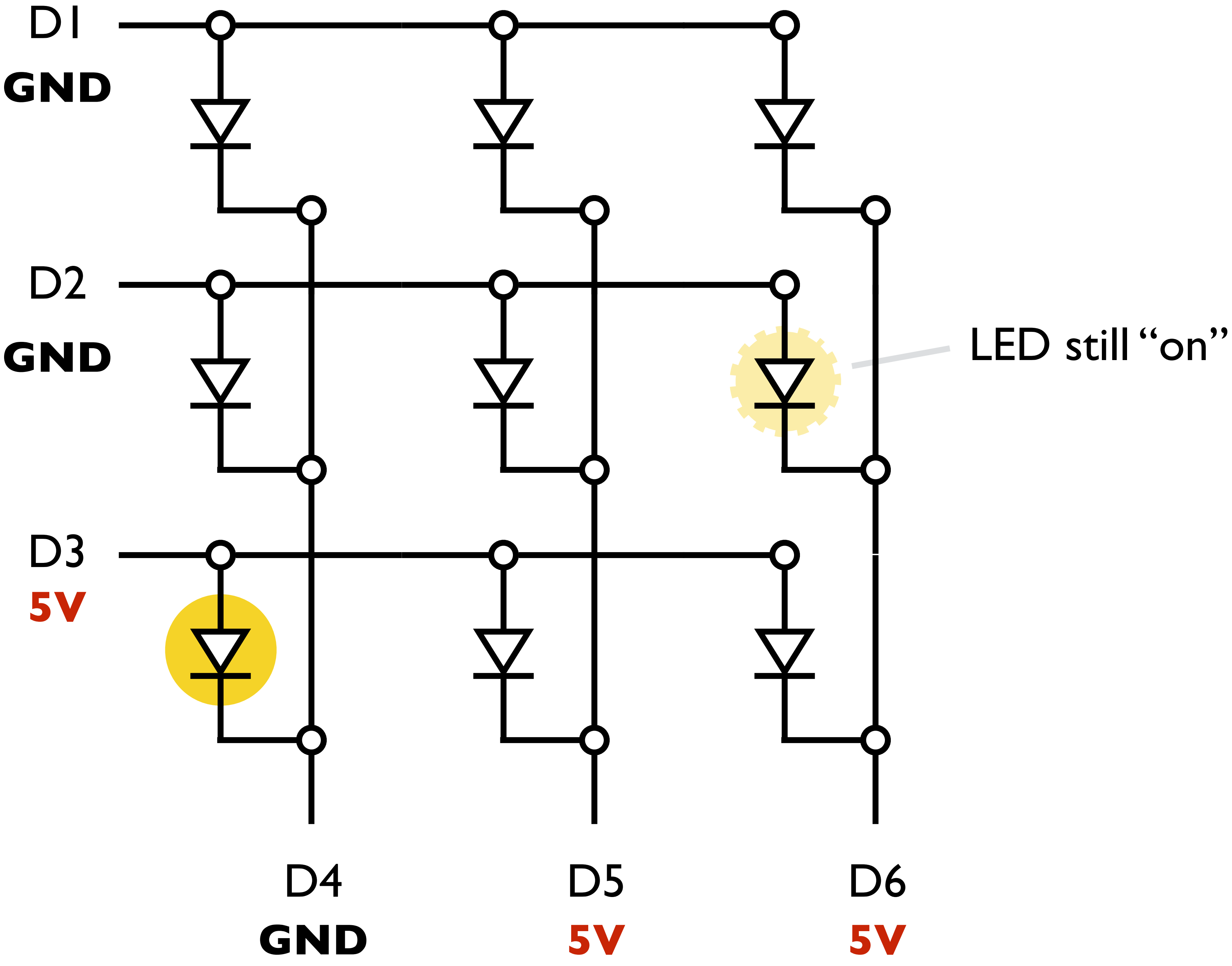
Multiplexing



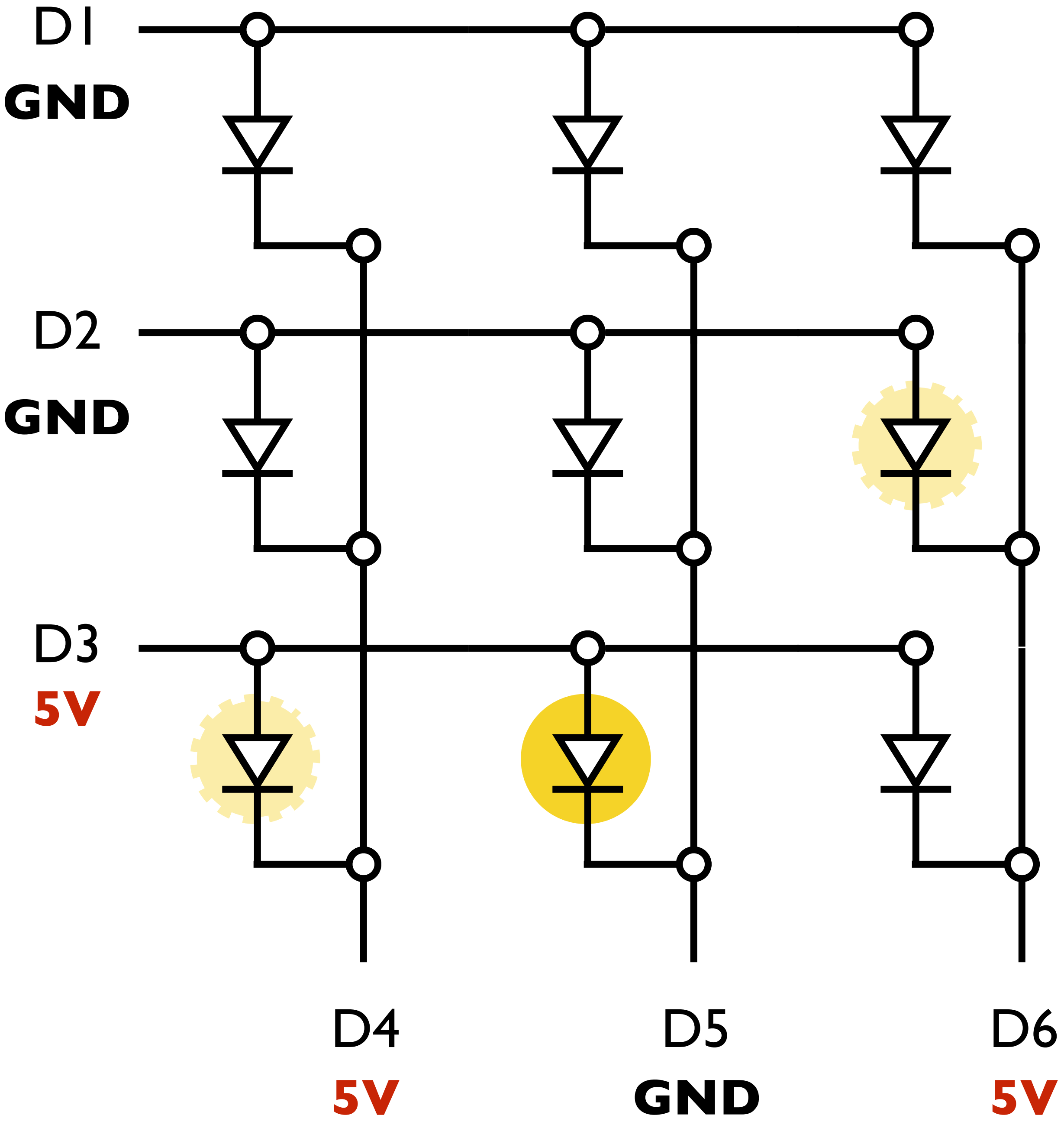
Multiplexing



Multiplexing



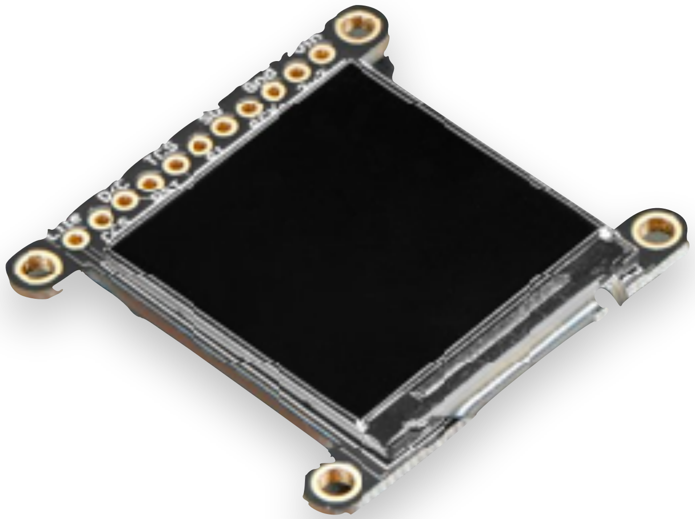
Multiplexing

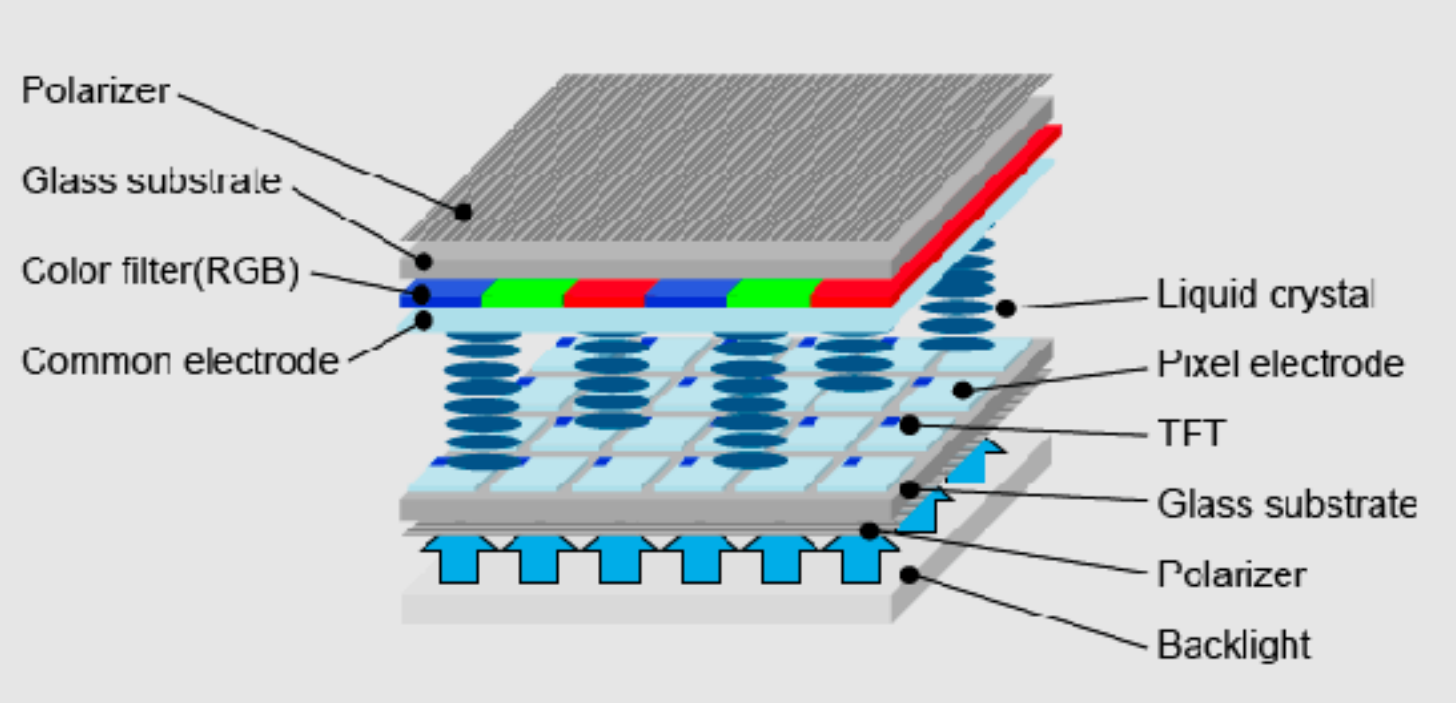


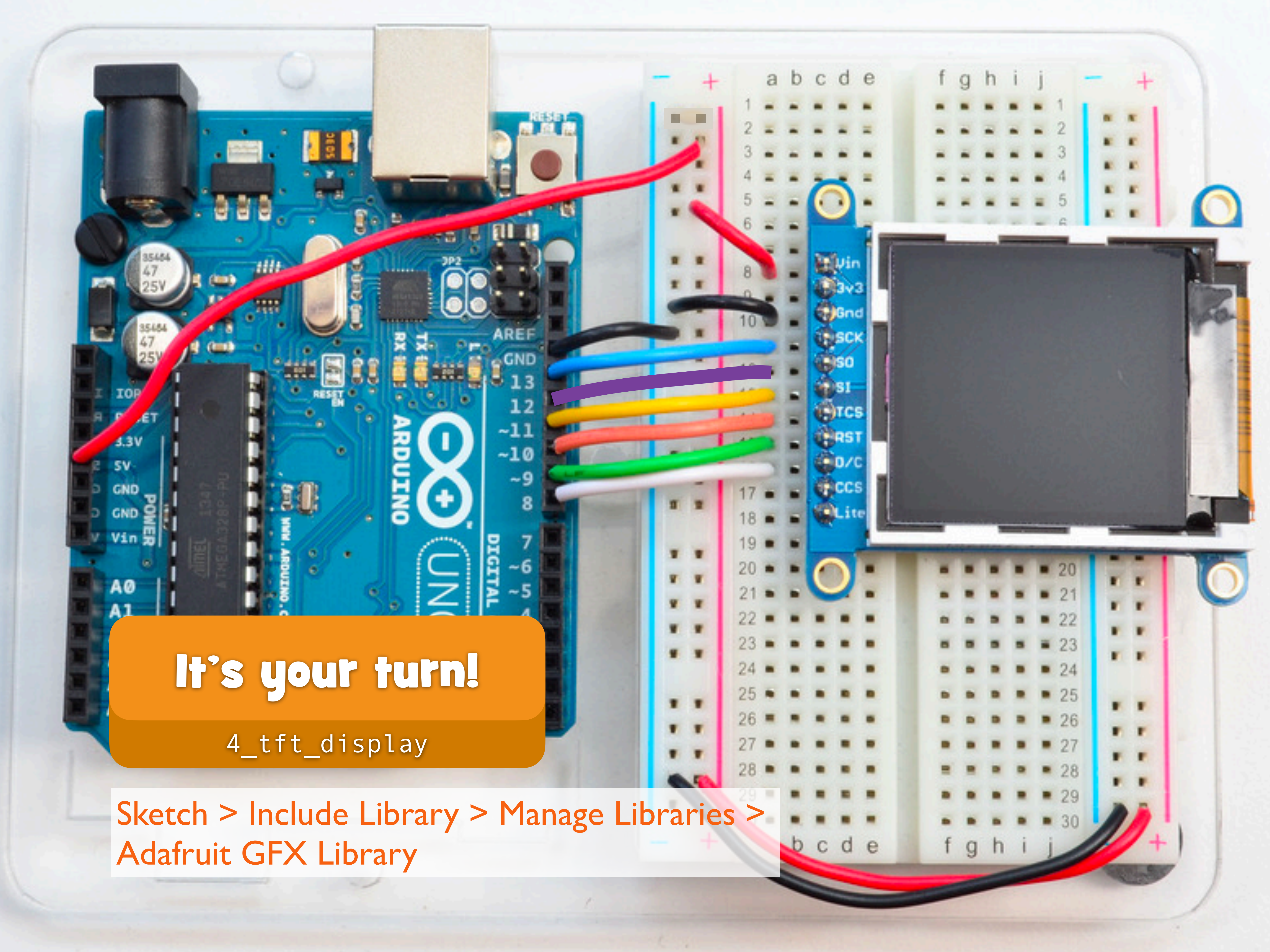
TFT Display

Thin **F**ilm **T**ransistor Displays

Breakout board/ Arduino library handles a lot of the logic!







It's your turn!

4_tft_display

Sketch > Include Library > Manage Libraries > Adafruit GFX Library

LED strips

A strip of LEDs

Individually Addressable LED strips!

Objectively pretty neat!

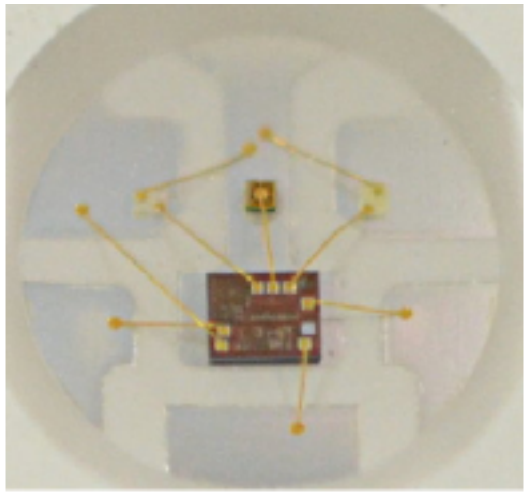
Integrated Circuit & I²C communication

ws2812b RGB LED

Run on 5v and can be controlled with an Arduino!

Watch out for current!
Each color ~ 15mA, total 50mA on 'white'.

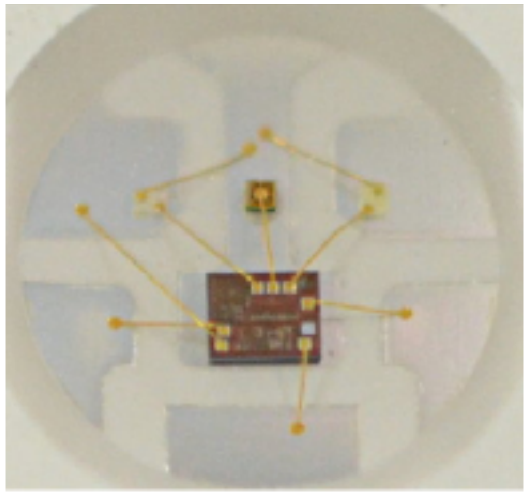
Arduino max current ~1A



Individually Addressable LED strips!

Can be individually powered (can consume a lot of power. Use rechargeable batteries!)

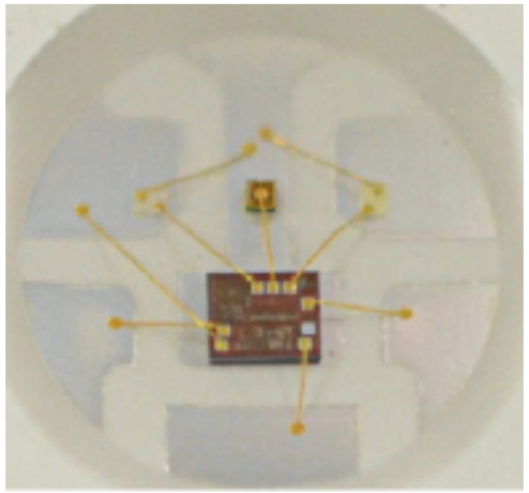
Connectors to help you split your LED strips into multiple lengths!



Individually Addressable LED strips!

Can be individually powered (can consume a lot of power. Use rechargeable batteries!)

Connectors to help you split your LED strips into multiple lengths!



It's your turn!

5_led_party

Sketch > Include Library > Manage Libraries > FastLED

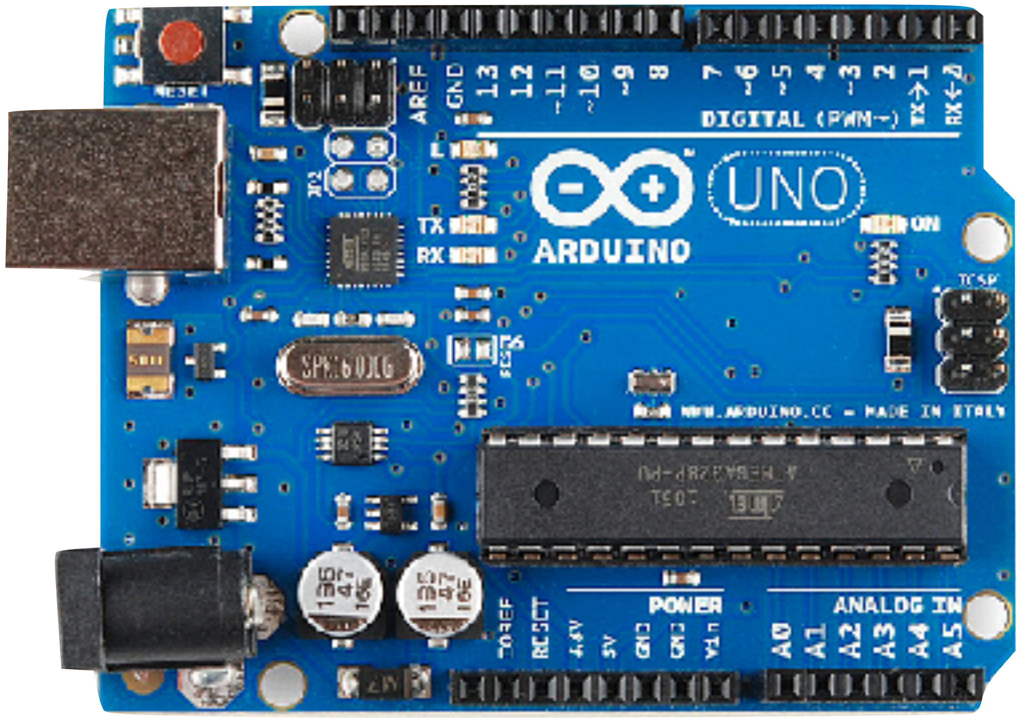
Wrap-up!

… wrapping up wires!

Wrap-up!

All the parts today actually came from Amazon. Other places you can order electronic parts from:
- Adafruit
- Jameco
- Digikey

Take anything you'd like to keep with you (you can use these on your toys/sketch models)



introduction to Digital Electronics