microcontroller workshop part 1

Computational Craft Week 5
Spring 2013

agenda

WHERE WE ARE GOING

What's on for today:

Quick shareout

Sorting/Mapping Exercise

What is a microcontroller?

Arduino

// the IDE

// the board

Digital vs. Analog

//INPUT = Switches + Variable resistors

//OUTPUT = PWM

Debugging

a note before we start

JUST TO BE CLEAR

We will skim the surface of the awesome powers of physical computing.

AND REMEMBER: You can do many, many things with very basic code.

Unfortunately, this is not an intro to p comp class. We don't have the time to actually go in depth in structure or syntax :(

So, for those of you who want to learn more, hit the Interwebs. The Arduino community is one of the best tech communities ever. If you run into a problem you absolutely cannot solve, see me. However, our shared goal in this course is your self-guided learning.

I expect you to try to find a solution (and there is always more than one) first.

sorting+mapping

INTERSECTIONS

Let’s do some free associating:

computer

sorting+mapping

INTERSECTIONS

Let’s do some free associating:

computer

materials

sorting+mapping

INTERSECTIONS

Let’s do some free associating:

computer

materials

interaction

microcontrollers

THEY'RE JUST LITTLE COMPUTERS

Microcontroller = mini-computer

IC (integrated circuit) with a processor, memory, and programmable input/output.

Every day objects embedded with them to controller behaviors.

Not usually reprogrammable because developed for one use.

the arduino

TA DAAAA!!

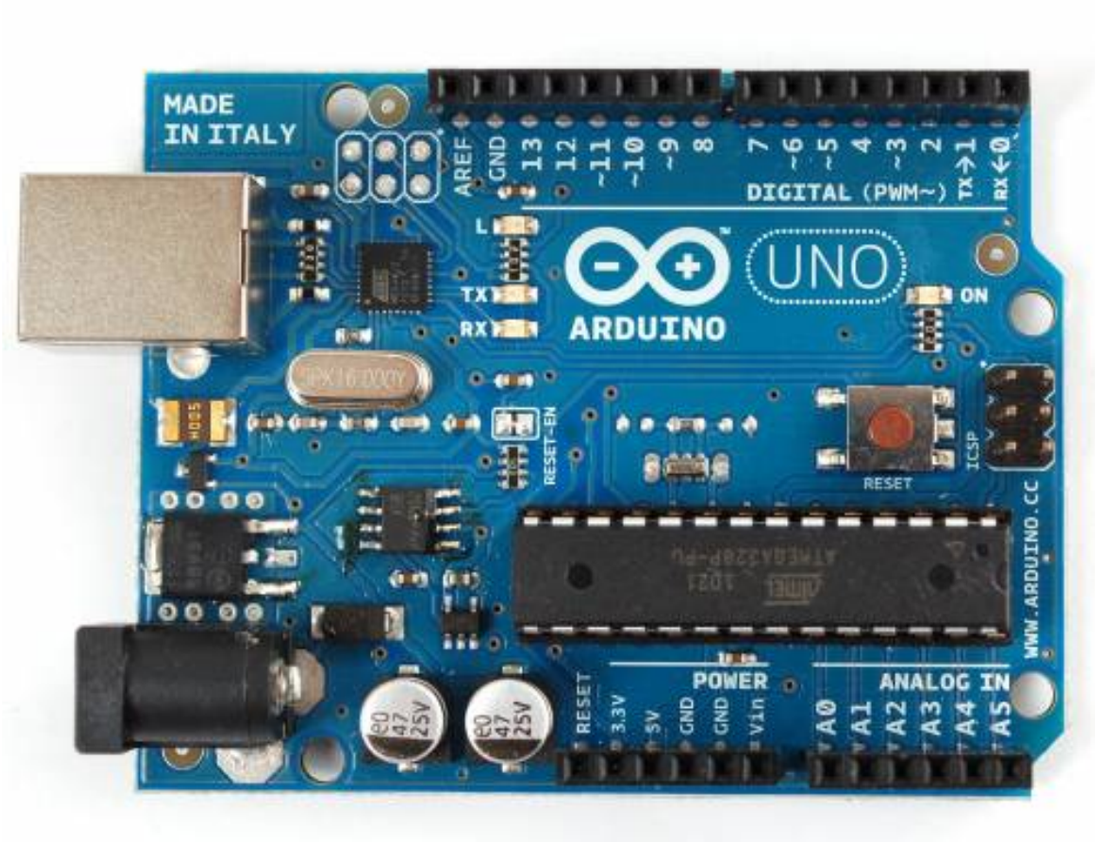
An open-source electronics prototyping platform.

Hardware + software

Makes our lives SO much easier!

arduino

TECH SPECS



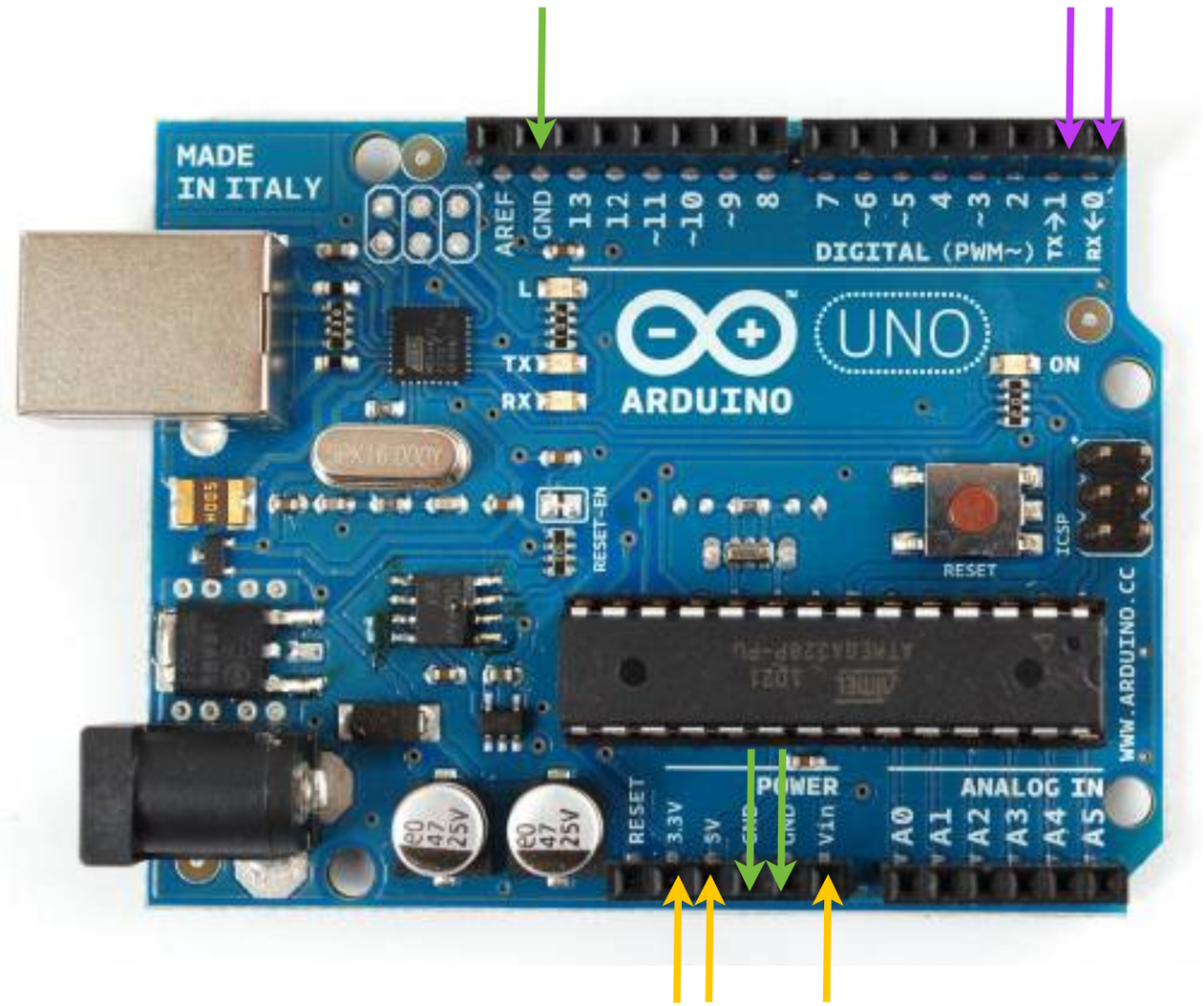
Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

arduino

PIN DESCRIPTION

A pin provides an input or output through which the controller can communicate with components.



TX/RX (serial - transmit/receive)

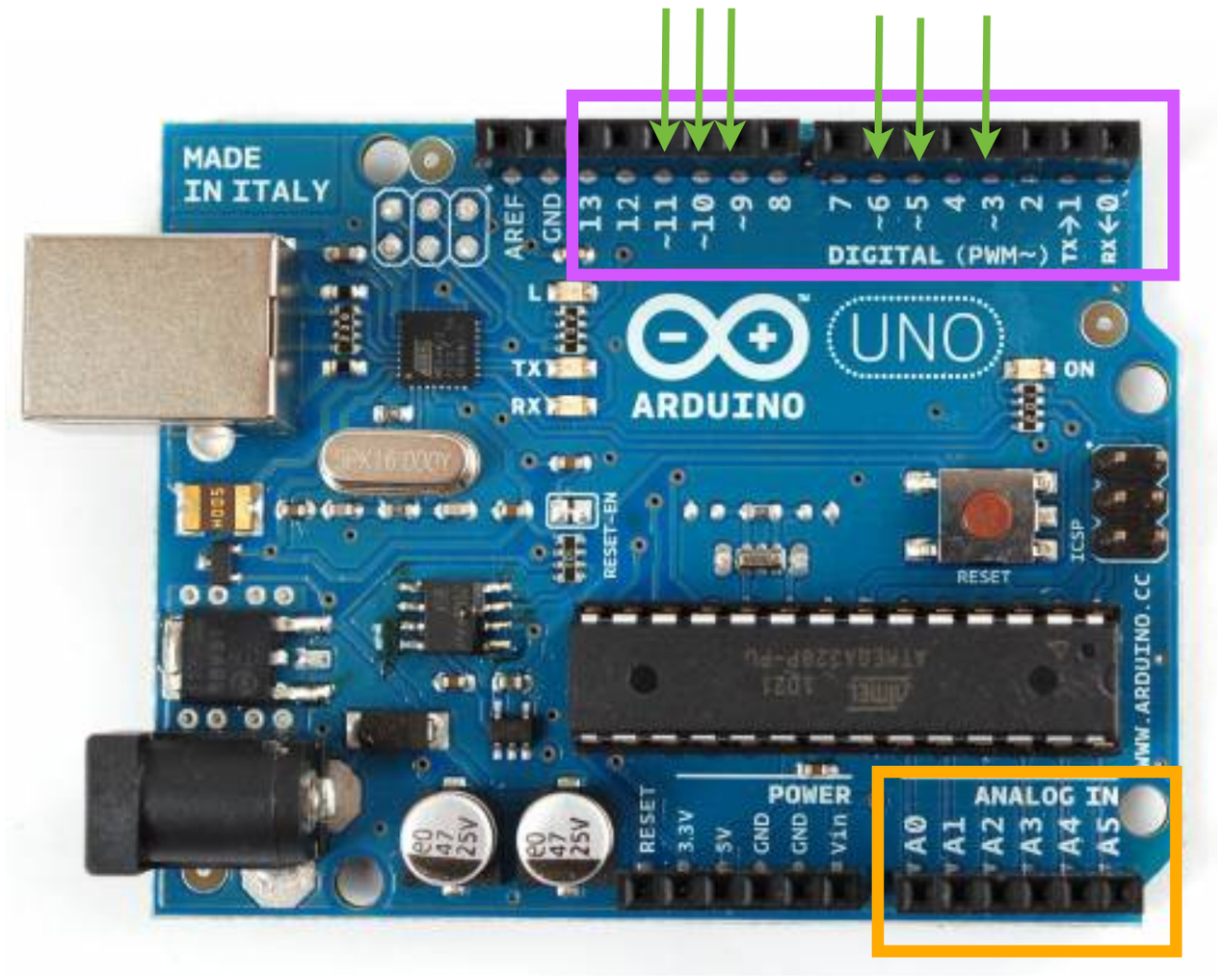
3 ground pins

3 power pins
// 5 volts
// 3 volts
// VIN - can plug 9 volts here

arduino

PIN DESCRIPTION

A pin provides an input or output through which the controller can communicate with components.



14 Digital pins

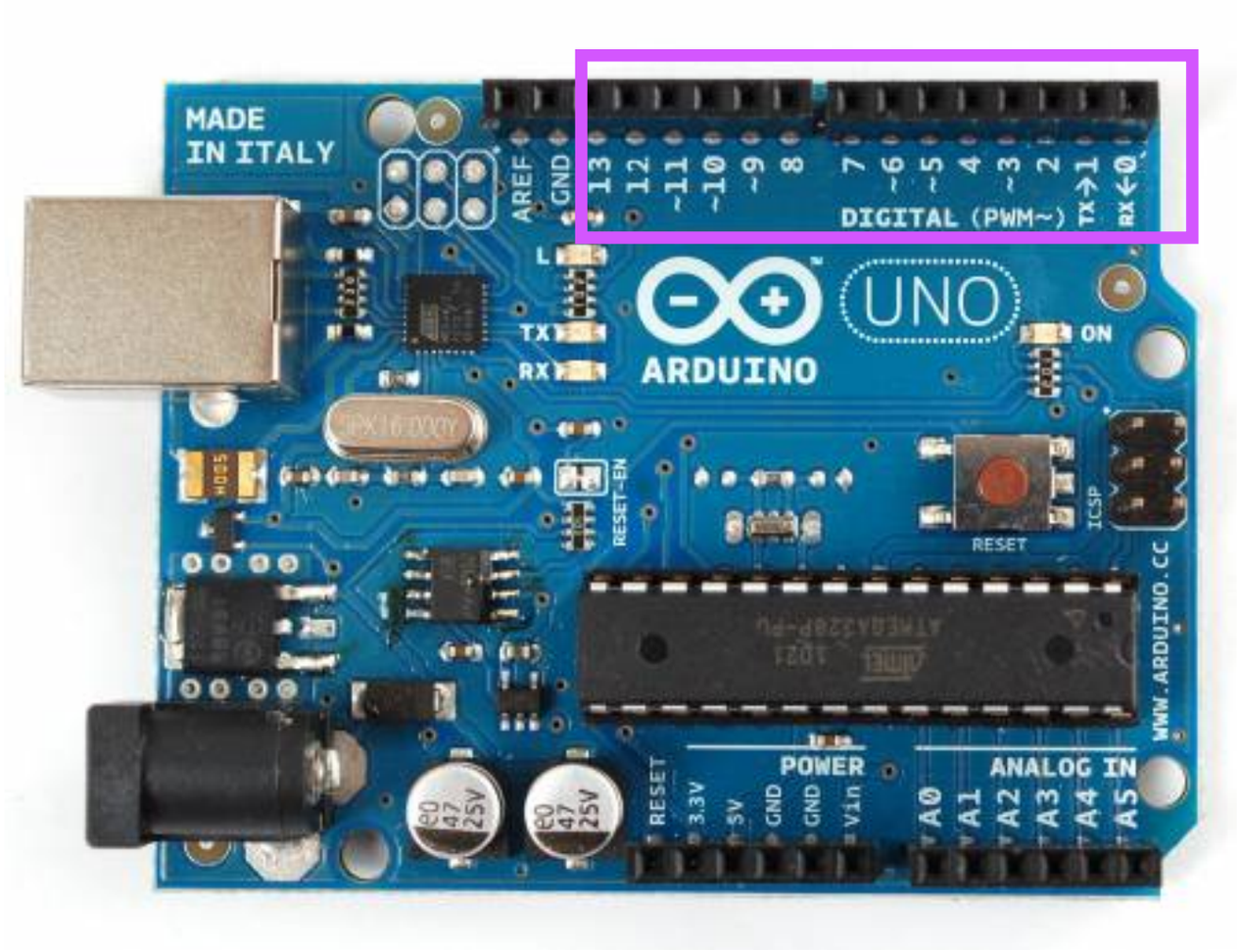
6 Pulse Width Modulation enabled pins

6 Analog input pins

digital

arduino

DIGITAL PINS



14 Digital pins

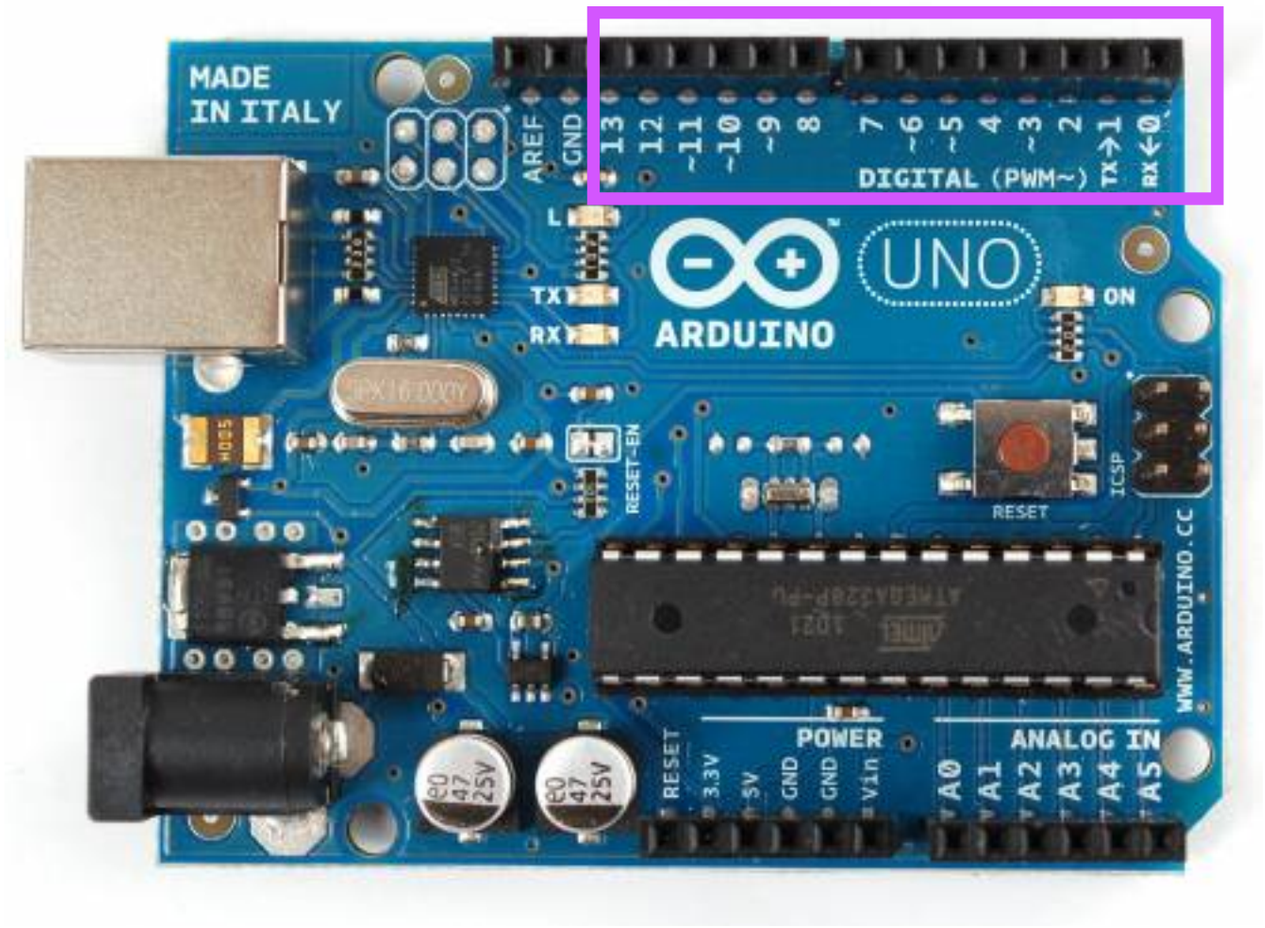
You can read or write 2 different values to them:

HIGH	LOW
5 volts	0 volts

You can think of HIGH as on and LOW as off

arduino

DIGITAL PINS



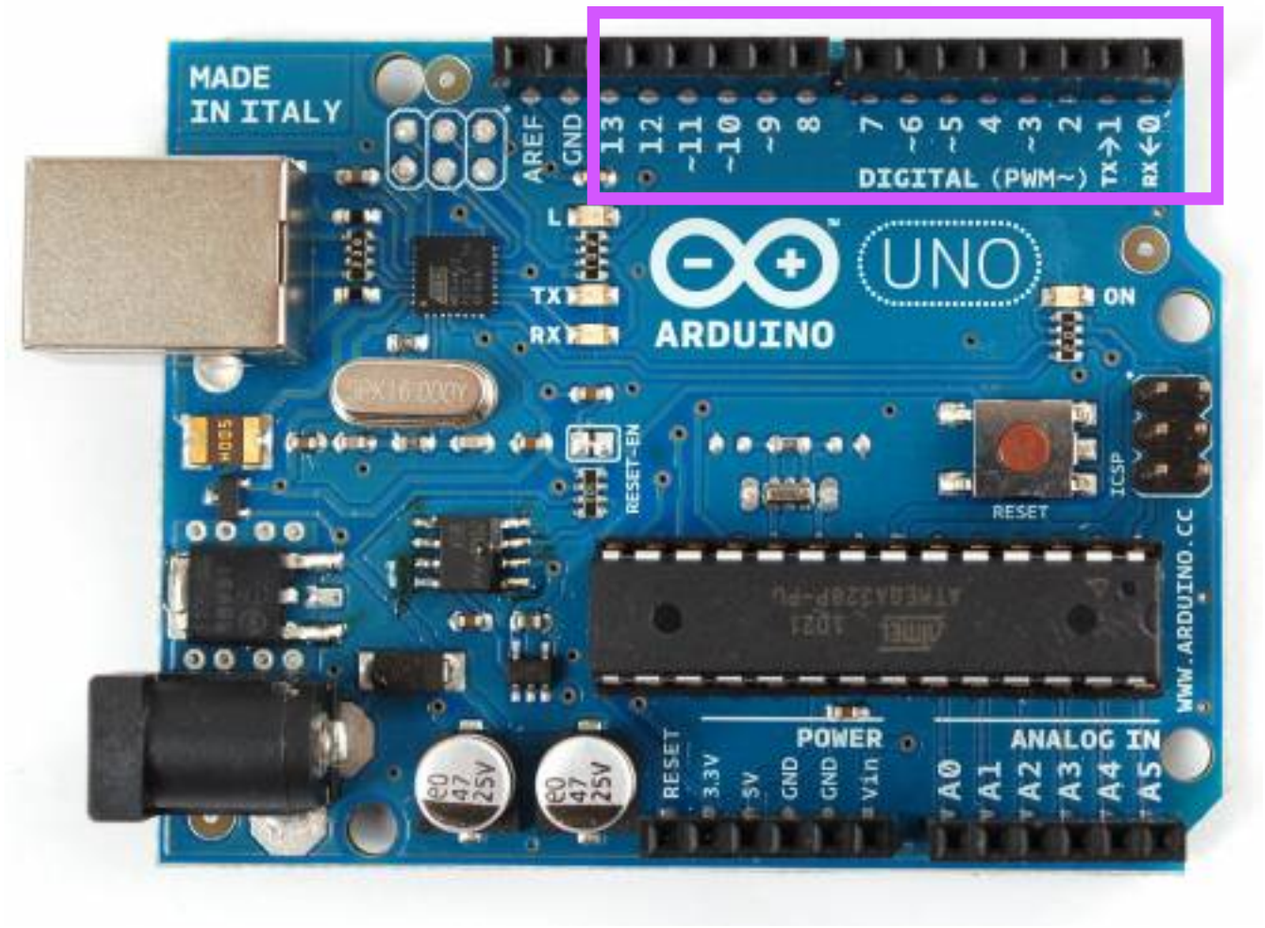
`pinMode(pin,mode)`

Sets the pin to be INPUT or OUTPUT

You don't have to do this every time but it is GOOD PRACTICE

arduino

DIGITAL PINS



`pinMode(pin,mode)`

Sets the pin to be INPUT or OUTPUT

You don't have to do this every time but it is GOOD PRACTICE

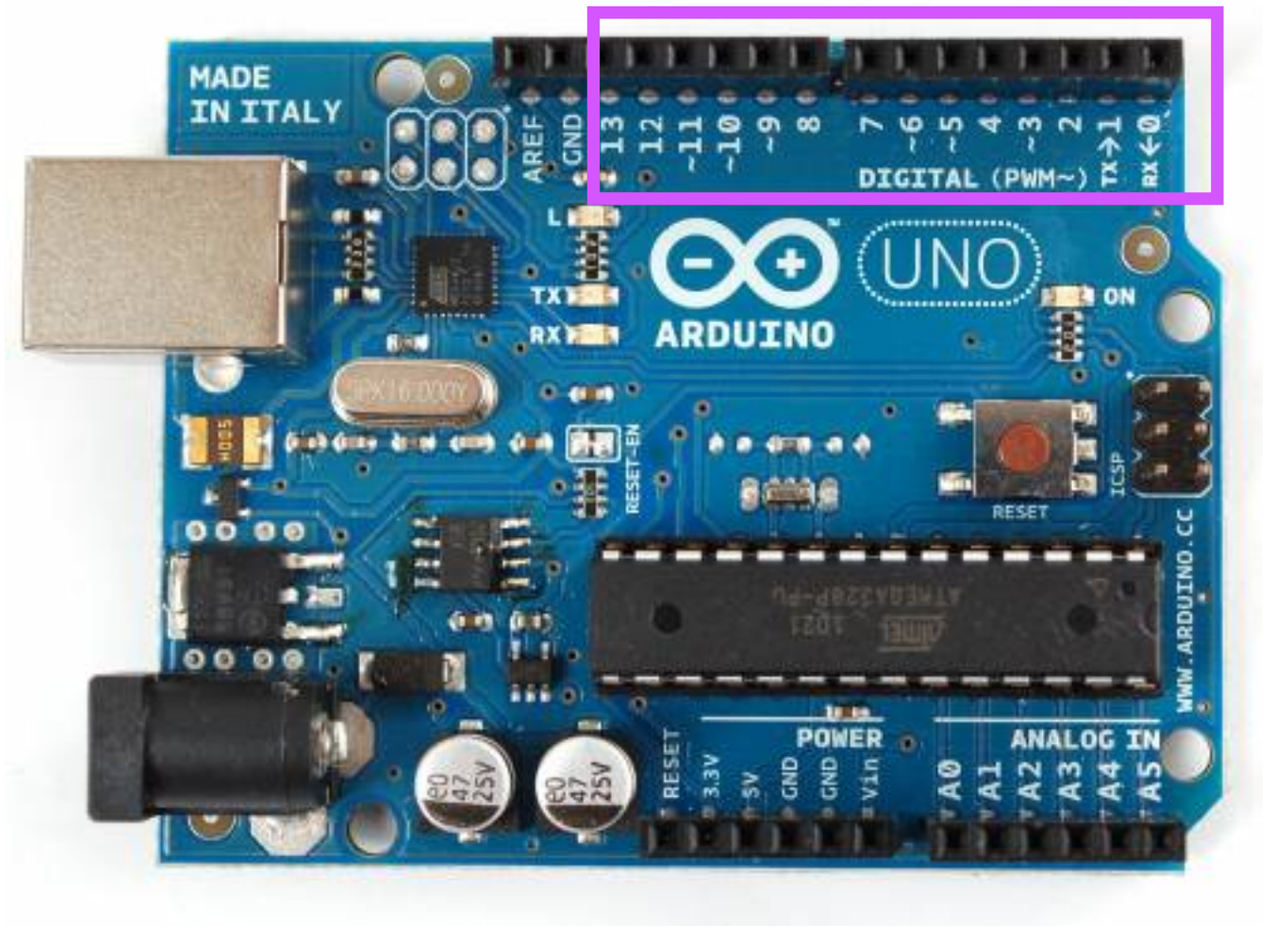
`digitalRead(pinNumber)`

Returns value from specified

For INPUT

arduino

DIGITAL PINS



`pinMode(pin,mode)`

Sets the pin to be INPUT or OUTPUT

You don't have to do this every time but it is GOOD PRACTICE

`digitalRead(pinNumber)`

Returns value from specified

For INPUT

`digitalWrite(pinNumber,value)`

Writes a value to the pin. Here we are talking HIGH (5V/on) or LOW (0V/off)

For OUTPUT

new components

BREADBOARDS, RESISTORS, AND JUMPERS, O MY!

It is best to test your sensors/buttons on a breadboard FIRST for debugging.

We need more components to help us do this:

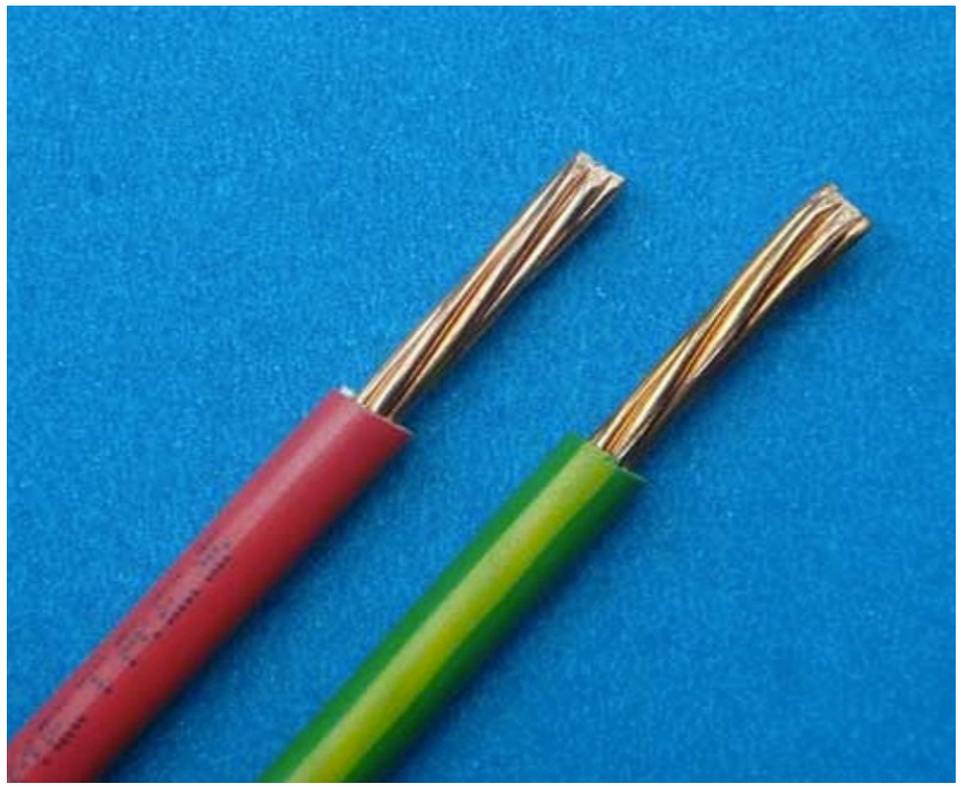
_ jumper wire
_ breadboard
_ resistors

jumper wires

INSULATED

Jumper wire is what we use to connect components in a circuit. We have been substituting this for copper tape and other materials.

20 gauge
30 gauge
//wire wrapping



breadboard

BREAK IT TO MAKE IT

A prototype board for putting together circuits.

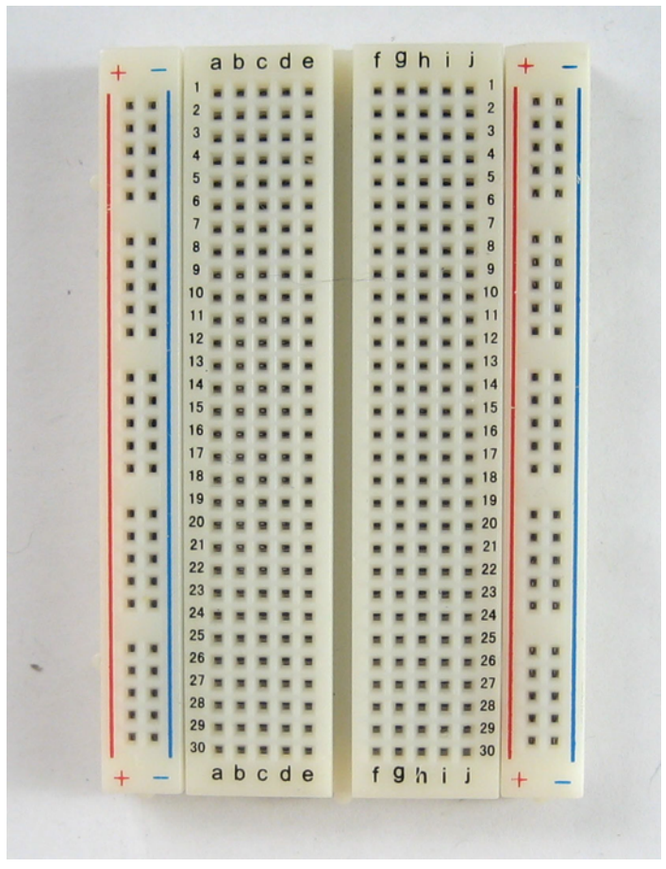
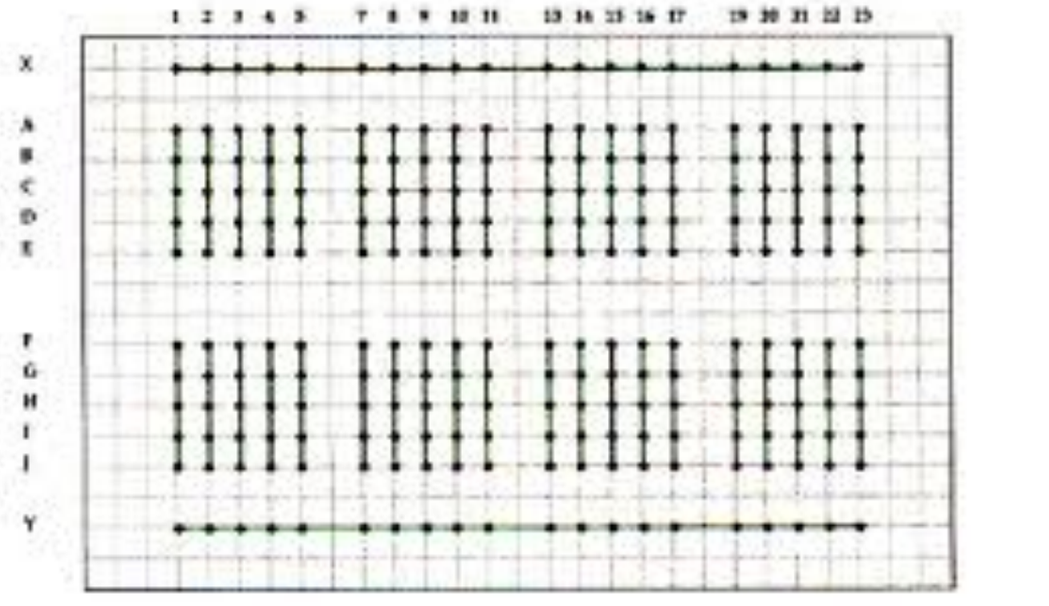


Figure 1. Pin Diagram of a Solderless Breadboard



Top view of a solderless breadboard. Colored lines indicate internal connections. Dots mark plug points.

resistors

YOUR ELECTRICAL BODYGUARDS

Remember this?

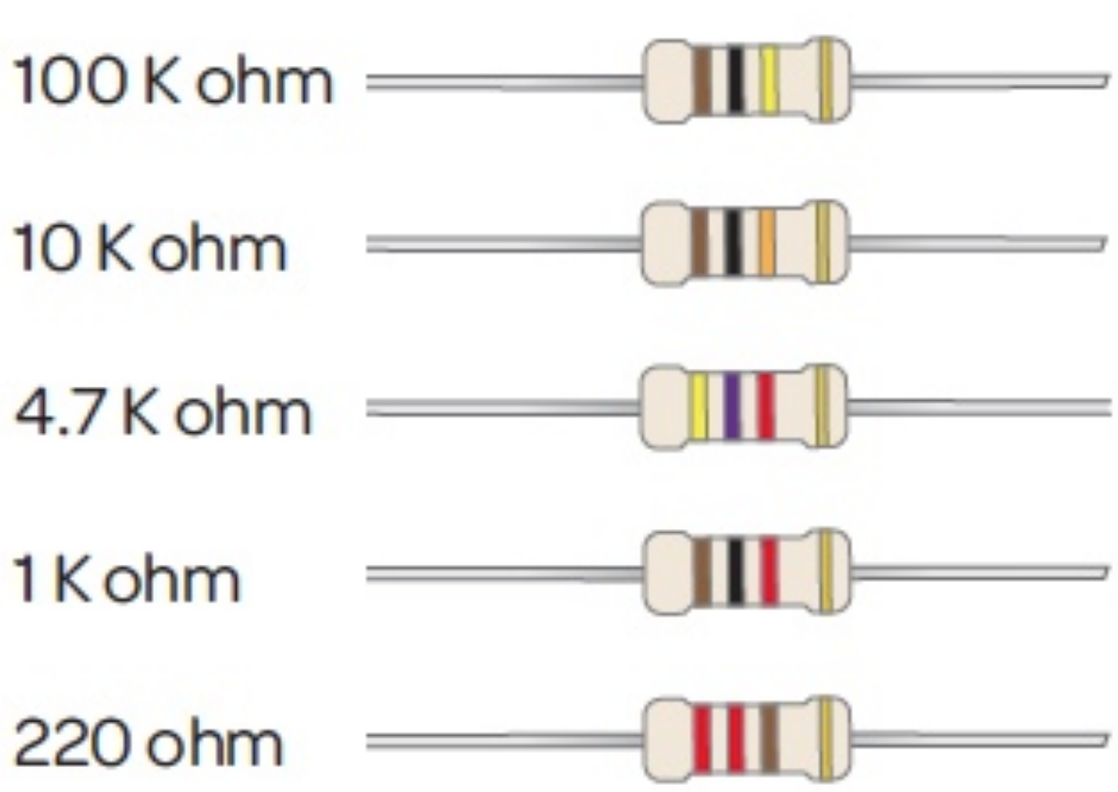
A resistor limits the amount of electricity that can flow through a certain point.

If we didn’t have these, we would fry our components.

resistors

COLOR-CODED

Resistors are basic building blocks. Here are some of the ones you will use most often:

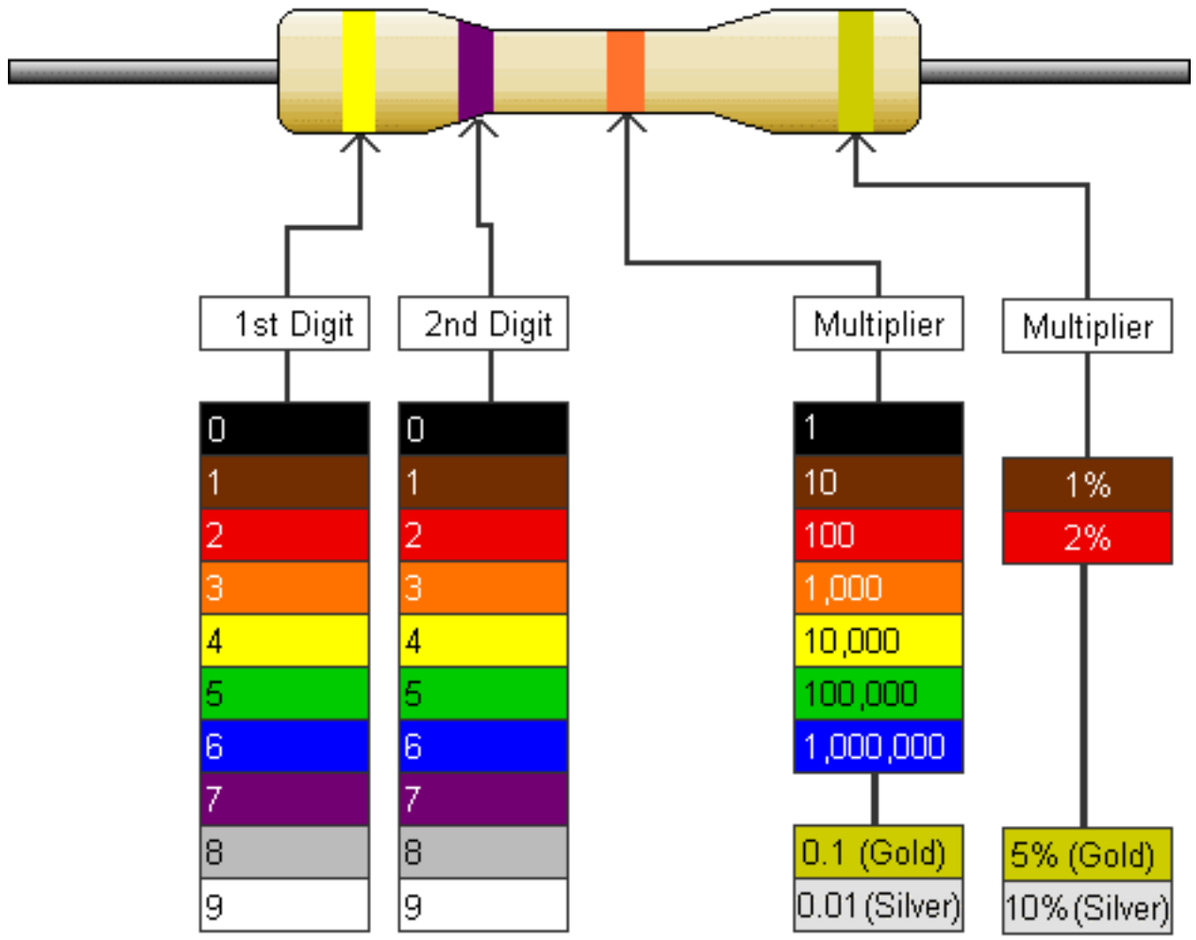


resistors

COLOR-CODED

Resistors have an interesting coding system.

Created by engineers, not designers.



resistors

COLOR-CODED

But how do you know which value to use?

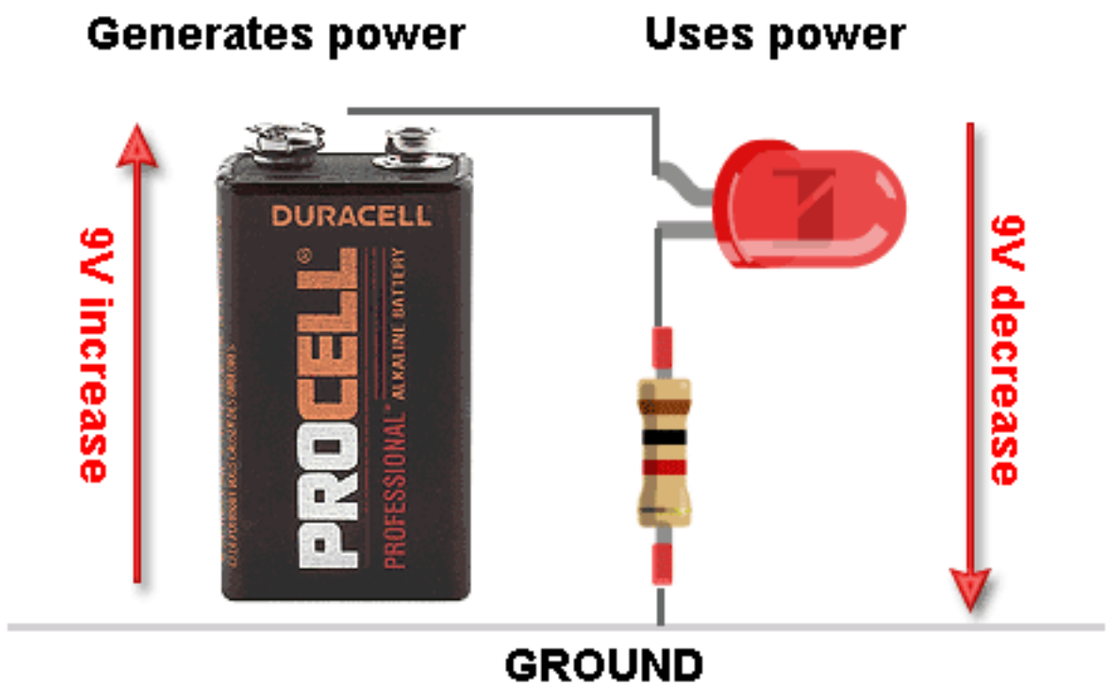
Generally you will only need to use the ones on the last slide. However, the more you understand resistance and it's relationship to current and voltage, the more you will be able to do.

If you want to know more about this, see Week 5: Ohm Sweet Ohm

kirchoff’s voltage law

LOOPS!

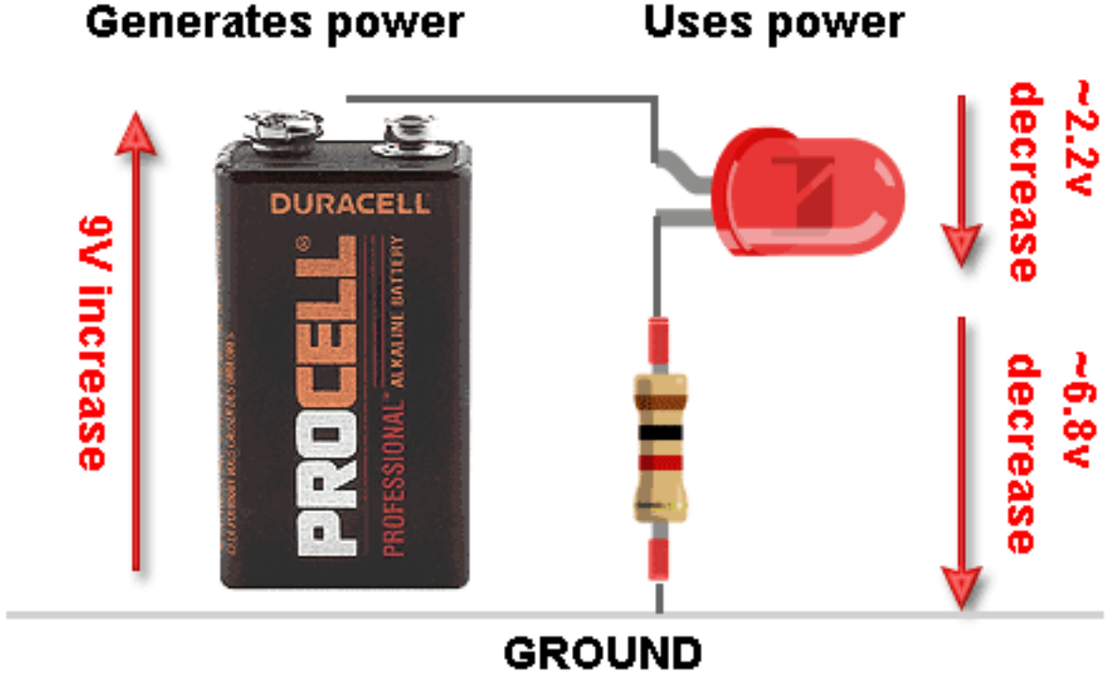
In any 'loop' of a circuit, the voltages must balance: the amount generated = the amount used



kirchoff’s voltage law

LOOPS!

Let’s break it down:



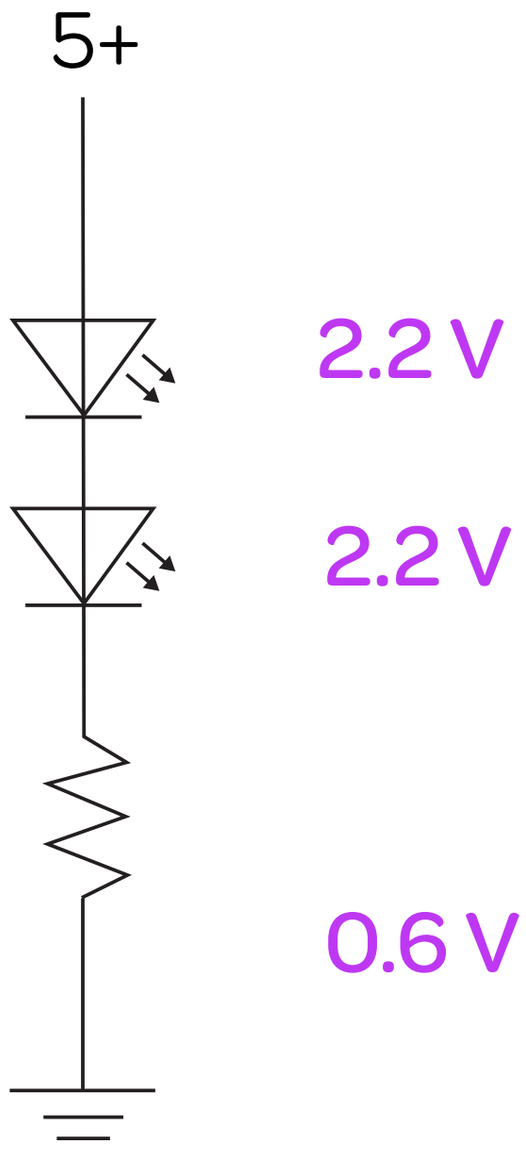
Forward Voltage (V_F)
= voltage lost to the LED (or other component)

kvl + ohm

AN EXAMPLE

We want to know the resistance to we need to have the LEDs running at full brightness.

$R = V/I$

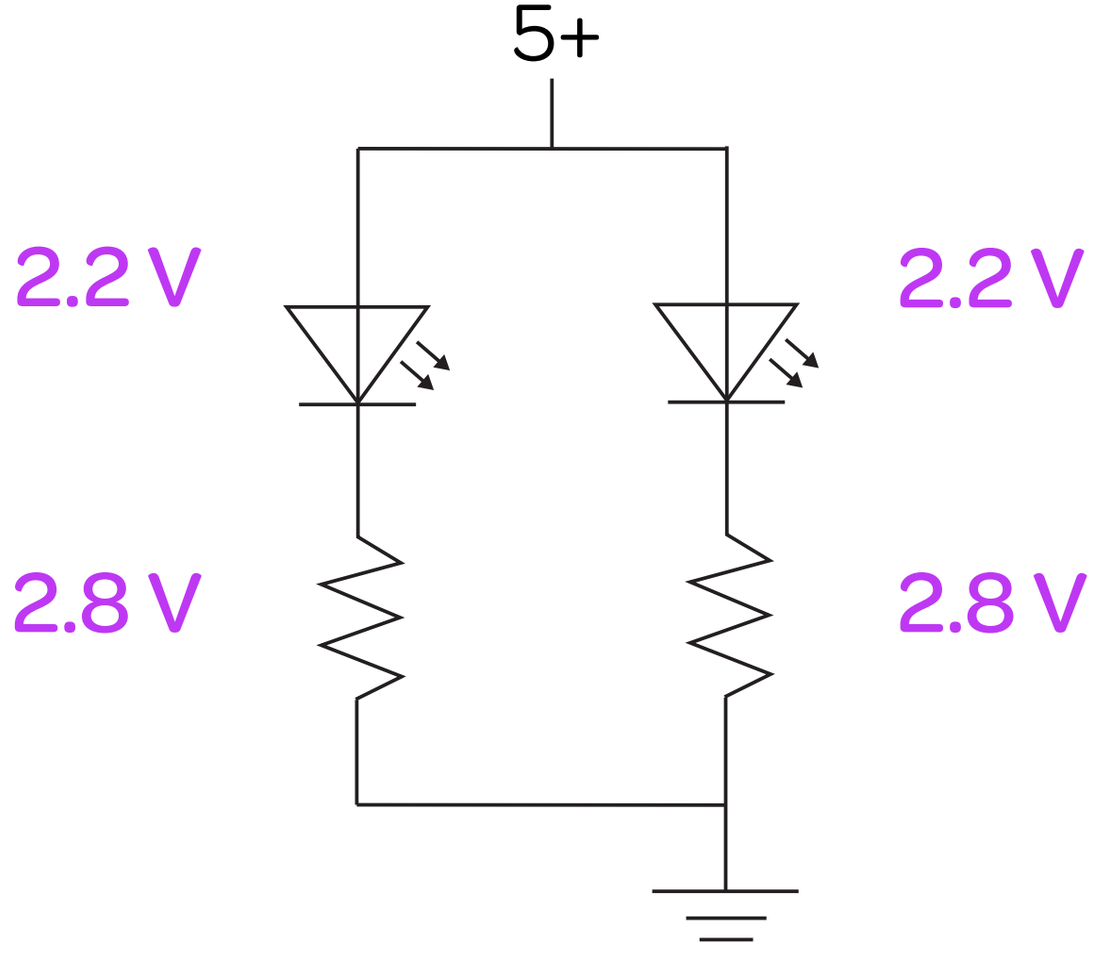


kvl + ohm

AN EXAMPLE

We want to know the current to know how bright the LED will be.

$I = V/R$

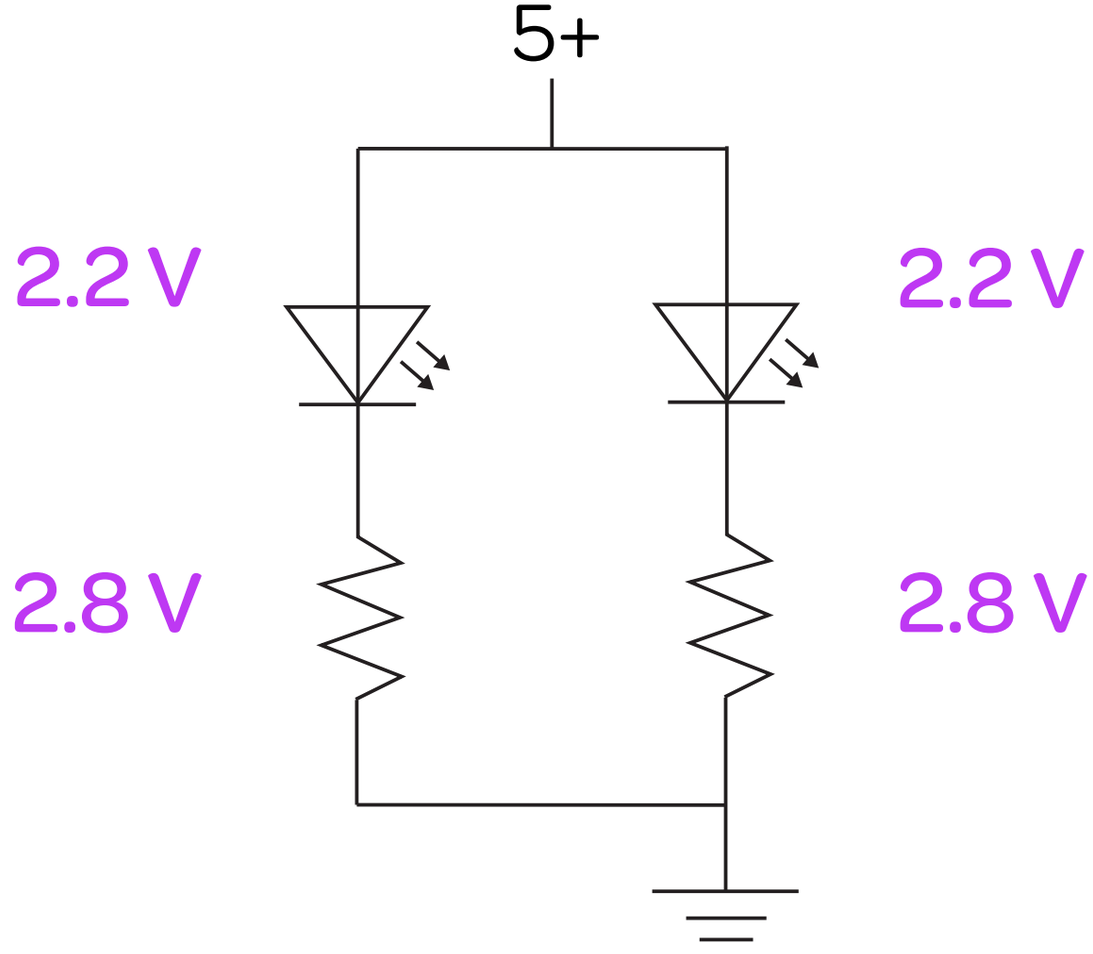


kvl + ohm

AN EXAMPLE

We want to know the current to know how bright the LED will be.

$I = 2.8/R$

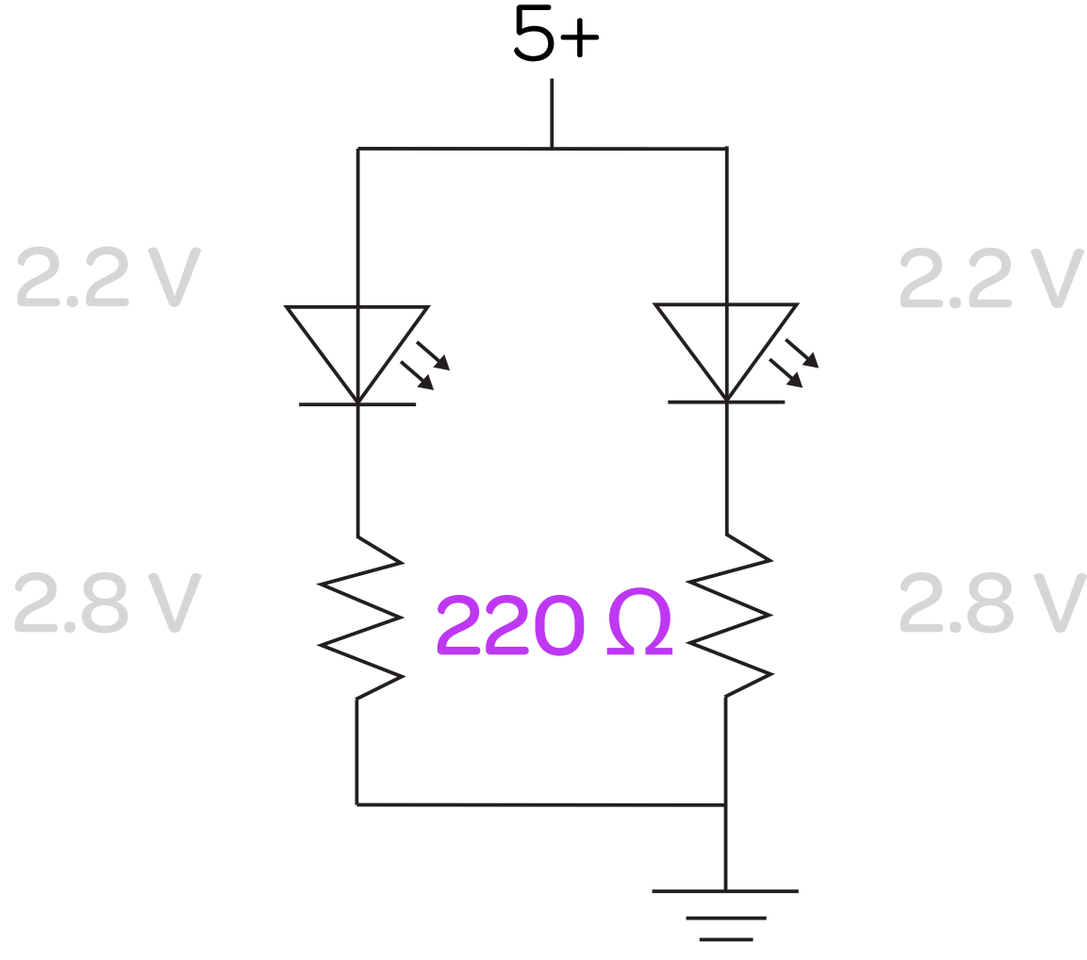


kvl + ohm

AN EXAMPLE

We want to know the current to know how bright the LED will be.

$$I = 2.8/220$$



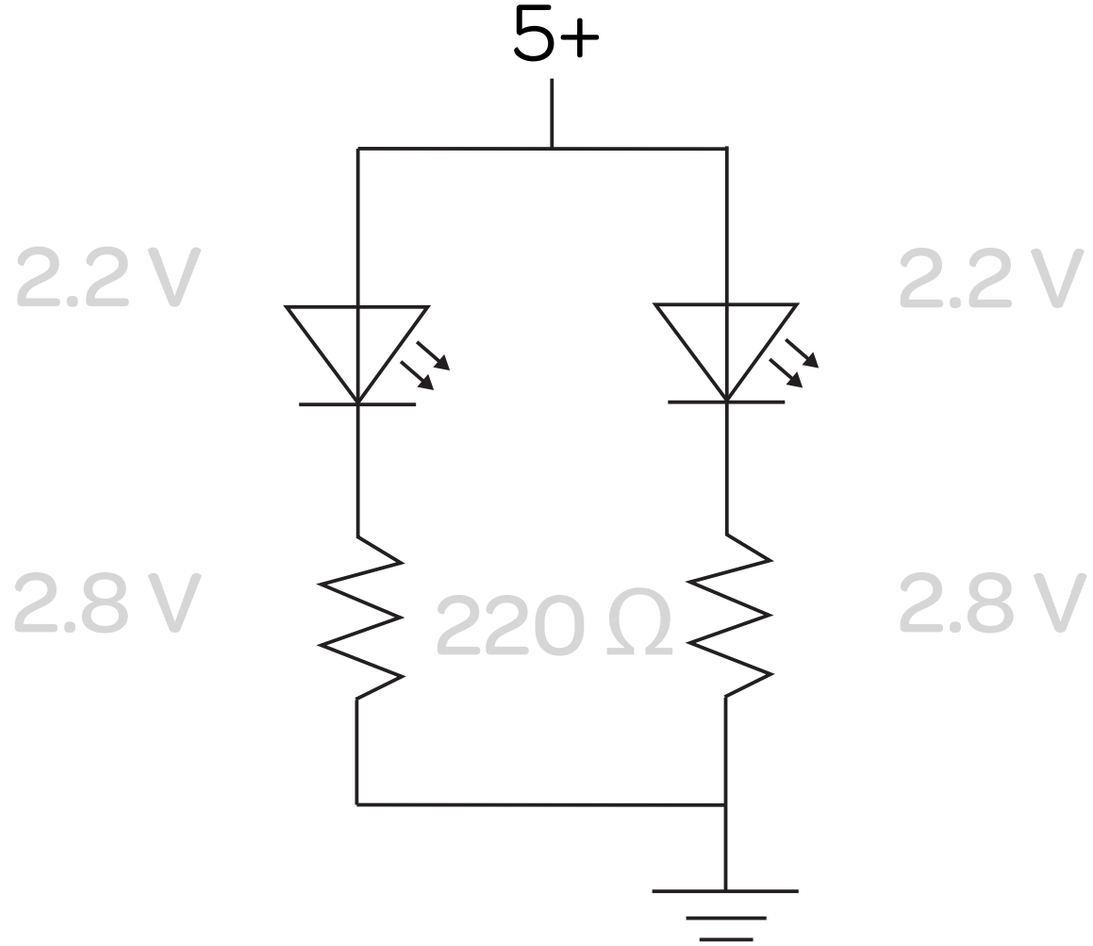
kvl + ohm

AN EXAMPLE

We want to know the current to know how bright the LED will be.

0.0127 =
2.8/220

12.7 mA

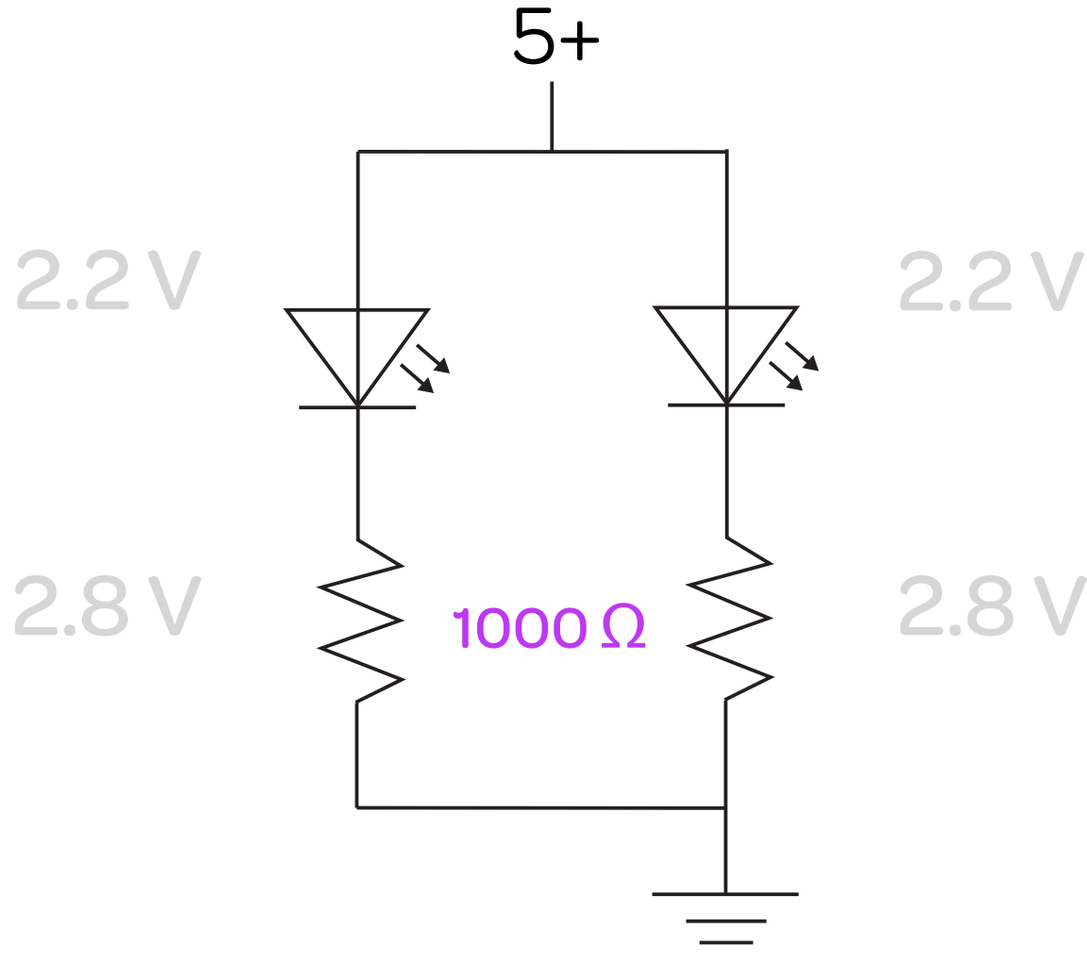


kvl + ohm

AN EXAMPLE

AGAIN! This time with 1K!

$I = 2.8/1000$



kvl + ohm

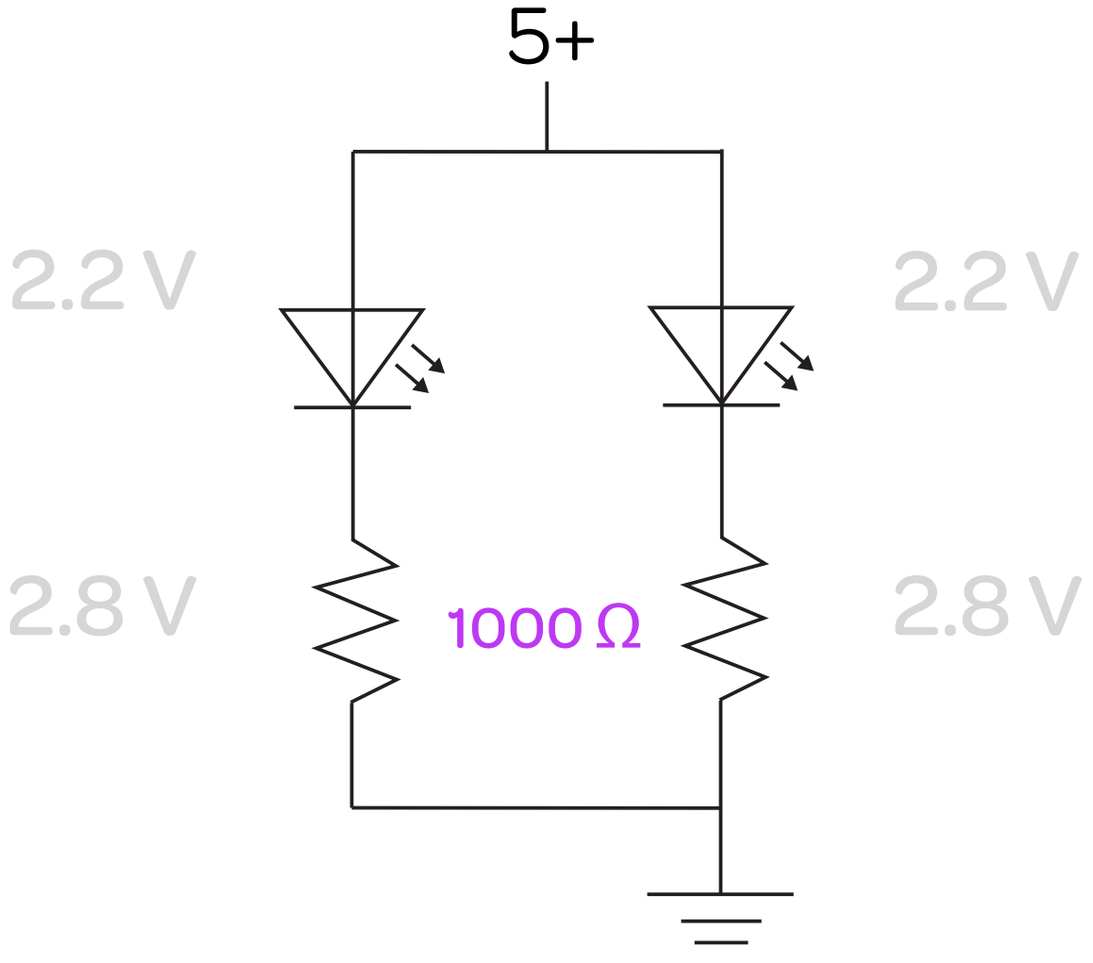
AN EXAMPLE

AGAIN! This time with 1K!

$I = 2.8/1000$

2.8 mA

So which is brighter?

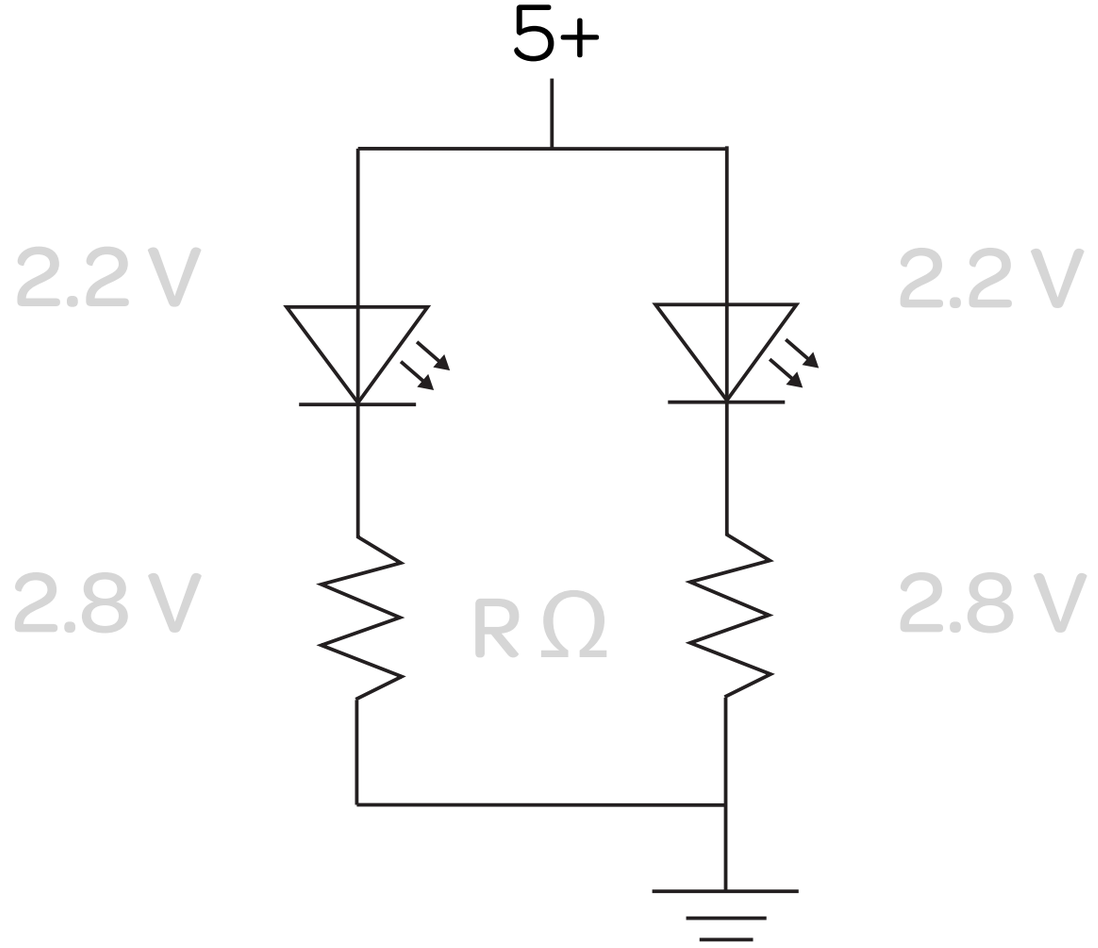


kvl + ohm

ANOTHER EXAMPLE

Let’s find the least valued resistor we can use without burning our LED.

$R = V/I$



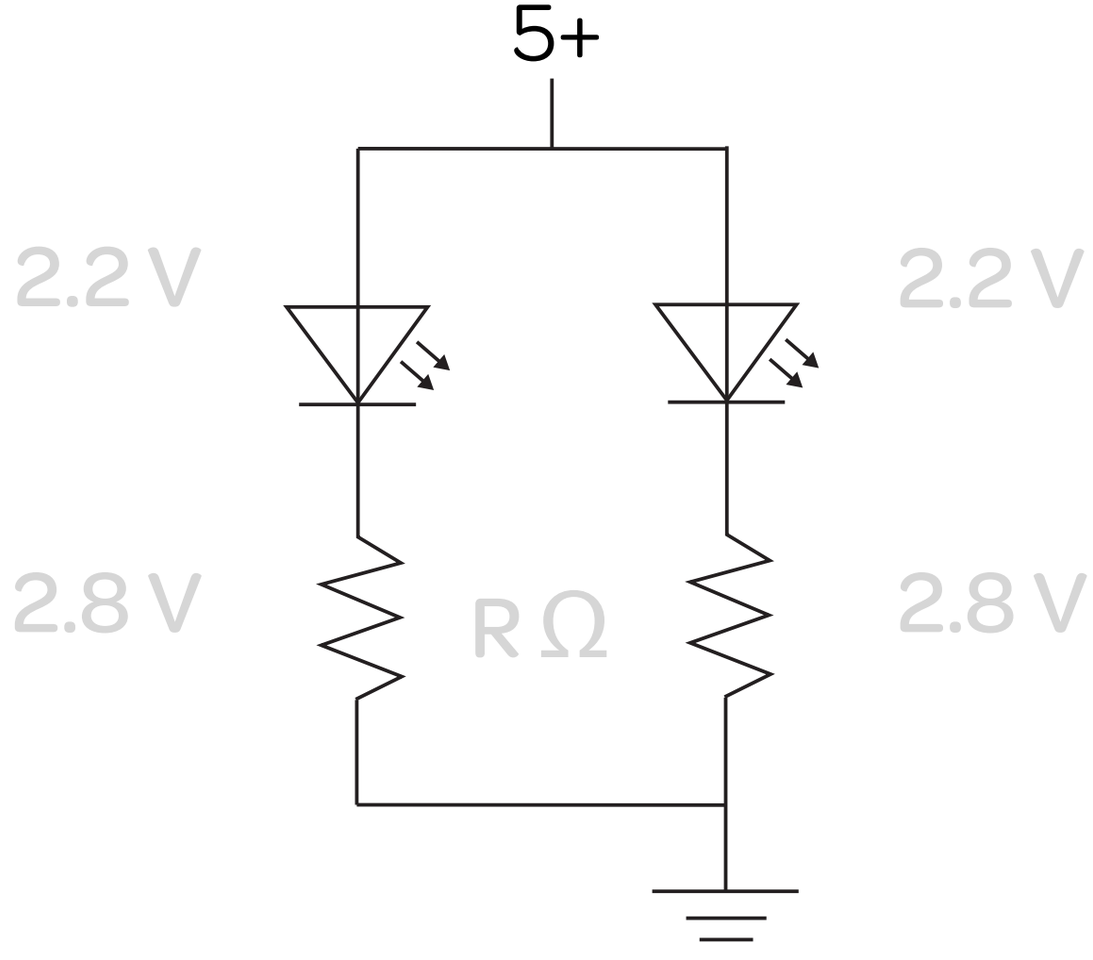
kvl + ohm

ANOTHER EXAMPLE

Let’s find the least valued resistor we can use without burning our LED.

$R = V/I$

$R = 2.8 / 0.02$



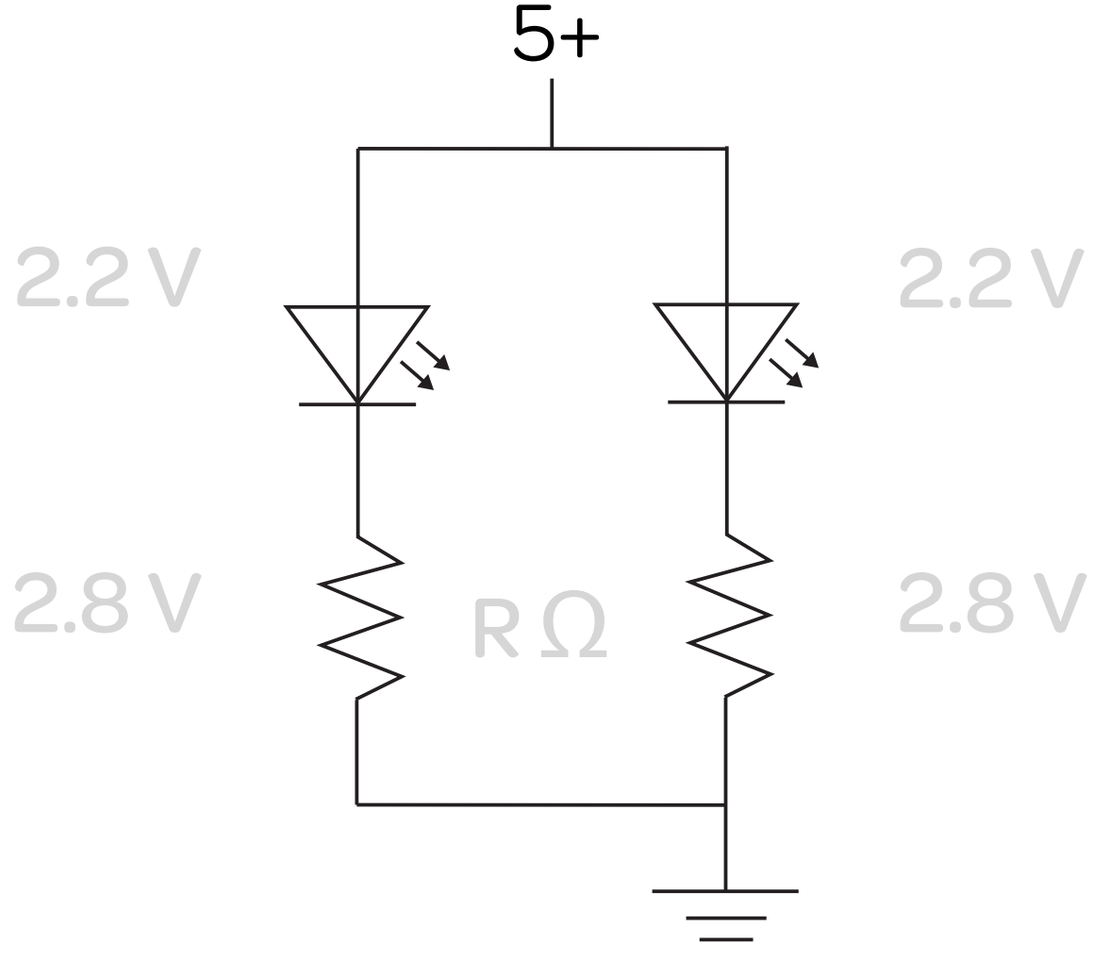
kvl + ohm

ANOTHER EXAMPLE

Let’s find the least valued resistor we can use without burning our LED.

$R = V/I$

$R = 140\ \Omega$



hello world

YOUR FIRST PROGRAM

Let’s make a light blink.

Put the positive end of the LED into pin 13 and the short end into ground.

You can only do this on pin 13!!
//There is a resistor enabled on pin 13

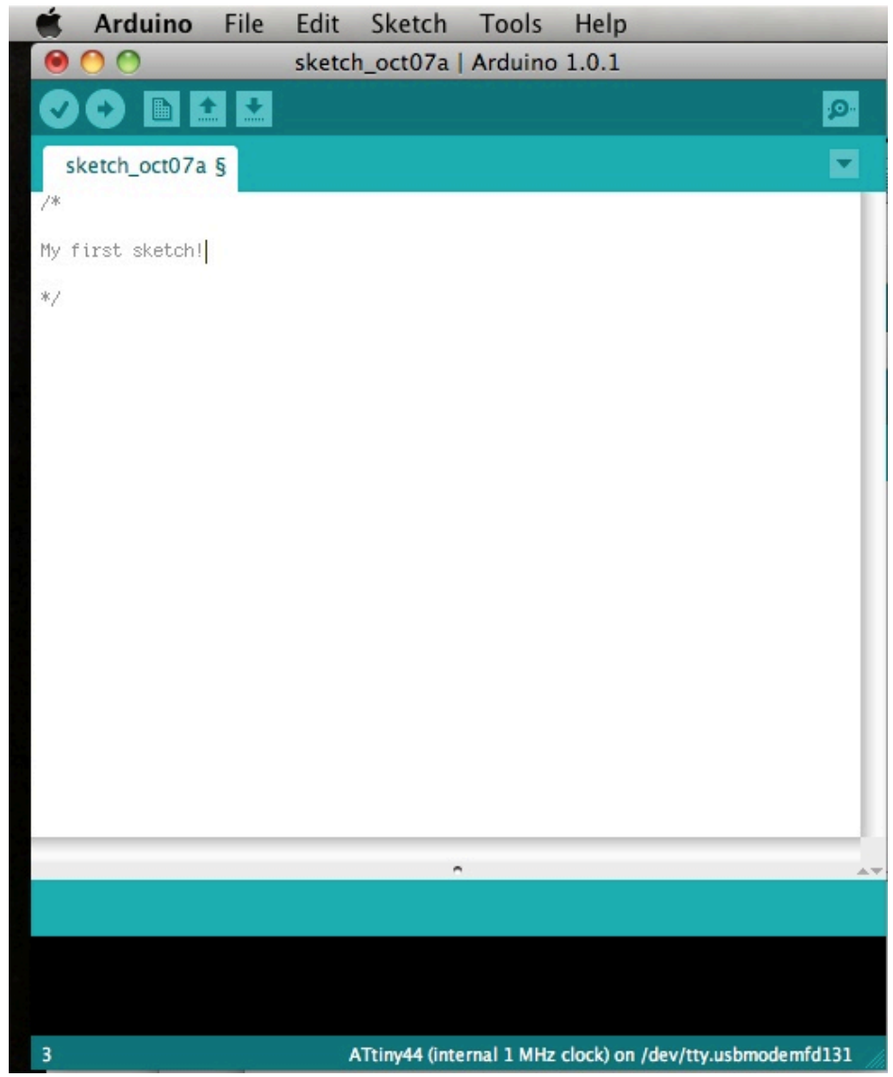
hello world

YOUR FIRST PROGRAM

Let's make a light blink.

STEP 0: Plug your Arduino into your computer.

Open the Arduino IDE.

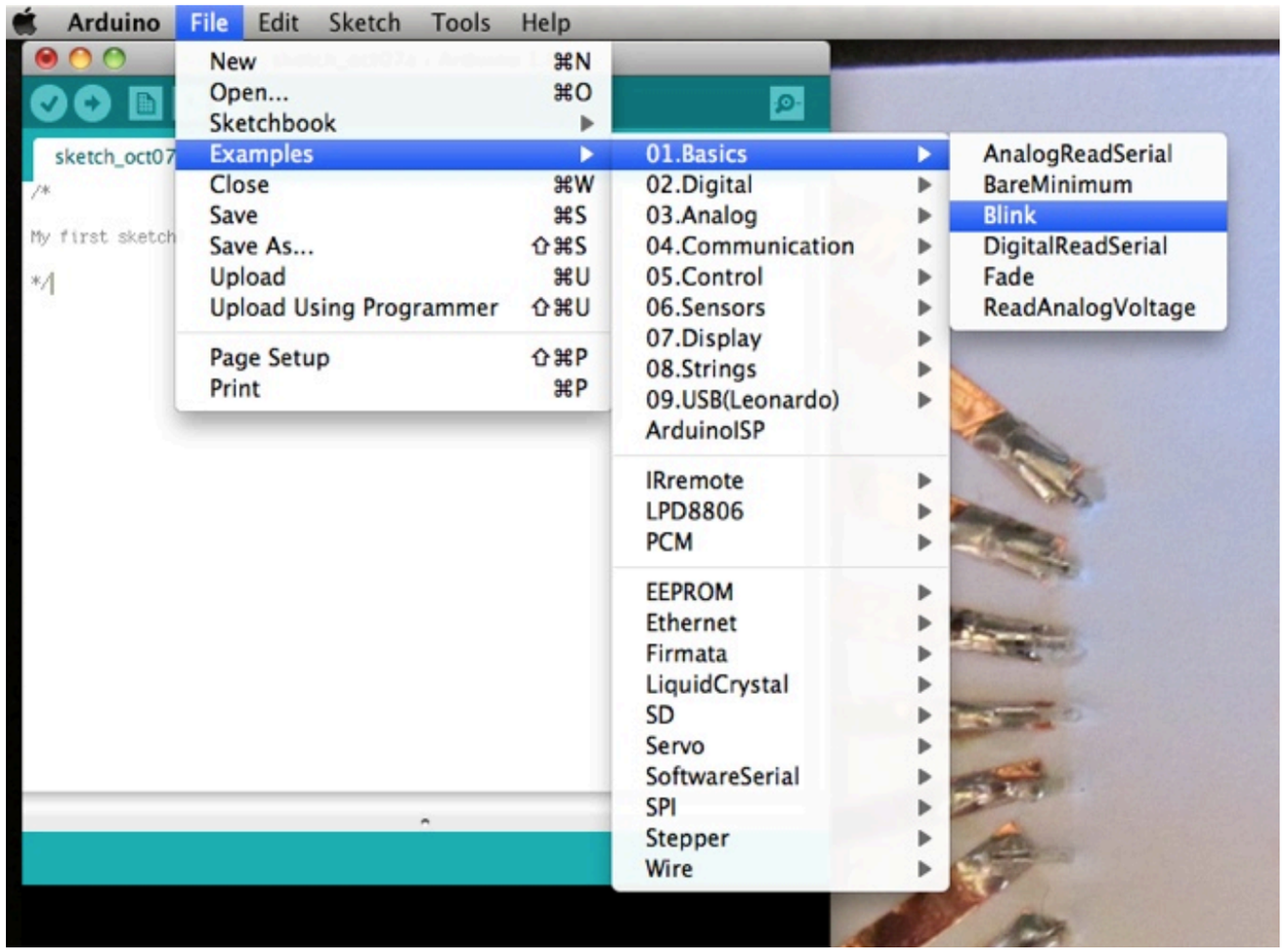


hello world

YOUR FIRST PROGRAM

Let's make a light blink.

STEP 1:
Open the sketch you want to upload.
File >> Examples >> Basics >> Blink



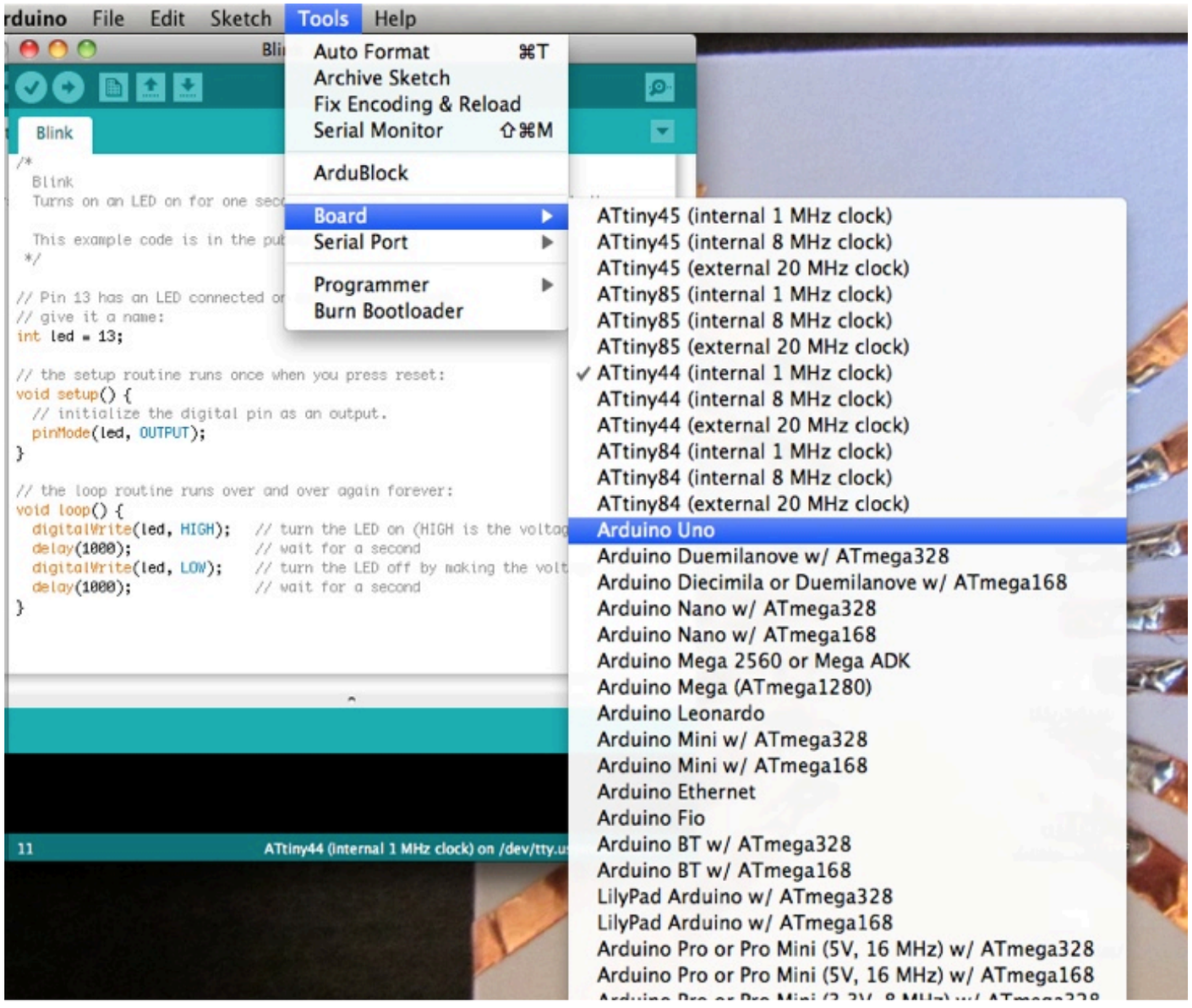
hello world

YOUR FIRST PROGRAM

Let’s make a light blink.

STEP 2:
Make sure Arduino knows the board you are using.

Tools >> Board
>> Arduino Uno



hello world

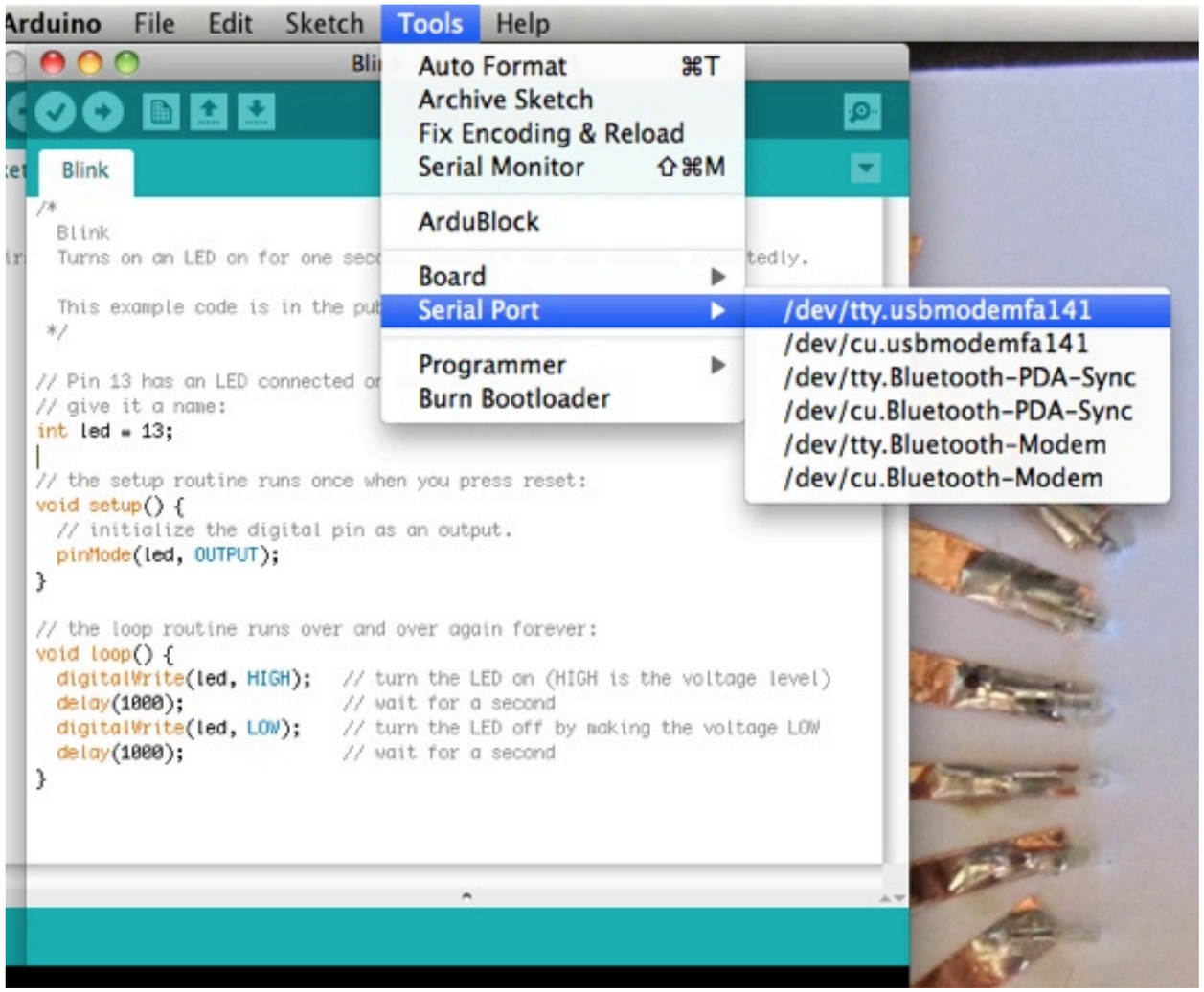
YOUR FIRST PROGRAM

Let’s make a light blink.

STEP 3:
Make sure Arduino knows the Serial port you are using.

Tools >> Serial Port >>
/dev/tty.usbmodemmfa141
(or something that looks like that!)

//Don’t worry about knowing what this means just yet.

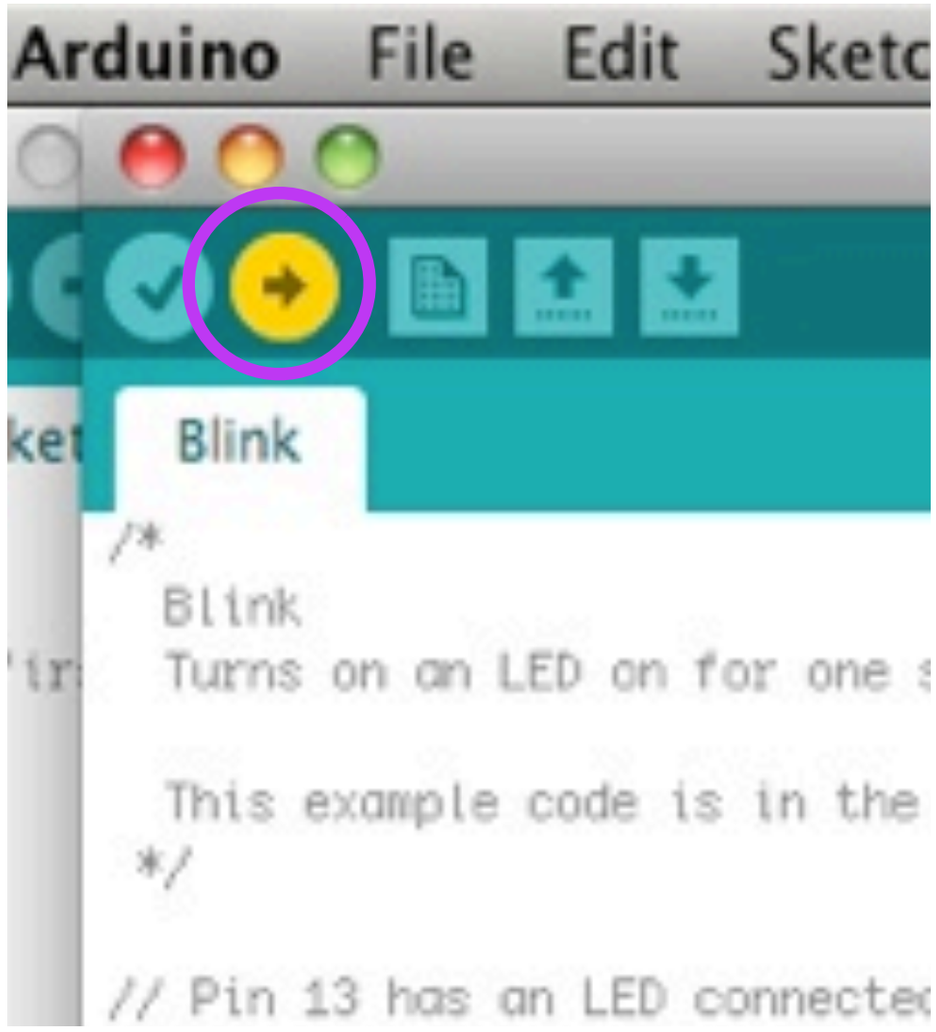


hello world

YOUR FIRST PROGRAM

Let's make a light blink.

STEP 4:
Upload the sketch onto the Arduino.

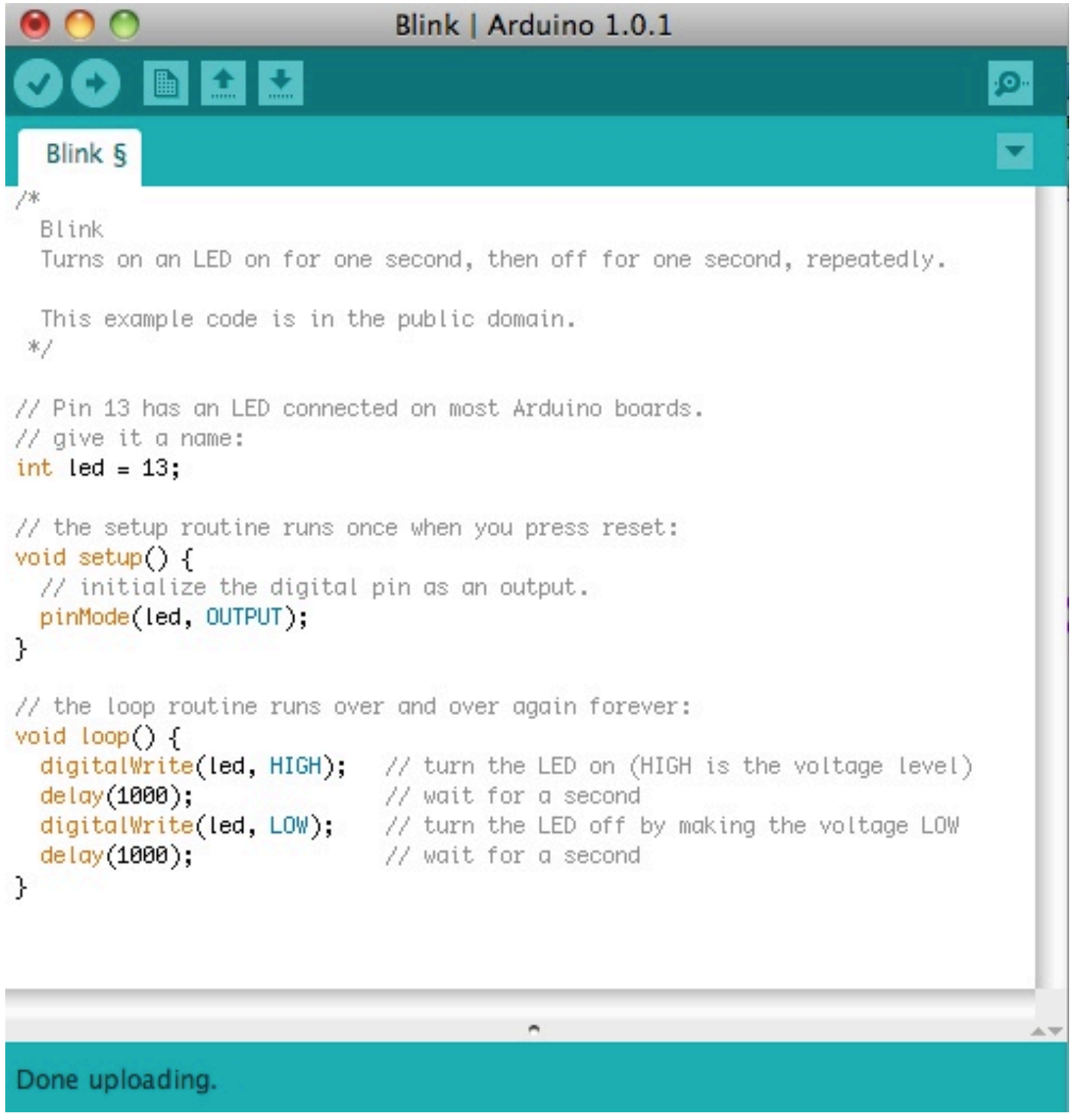


hello world

YOUR FIRST PROGRAM

Let’s make a light blink.

STEP 5:
What happens?



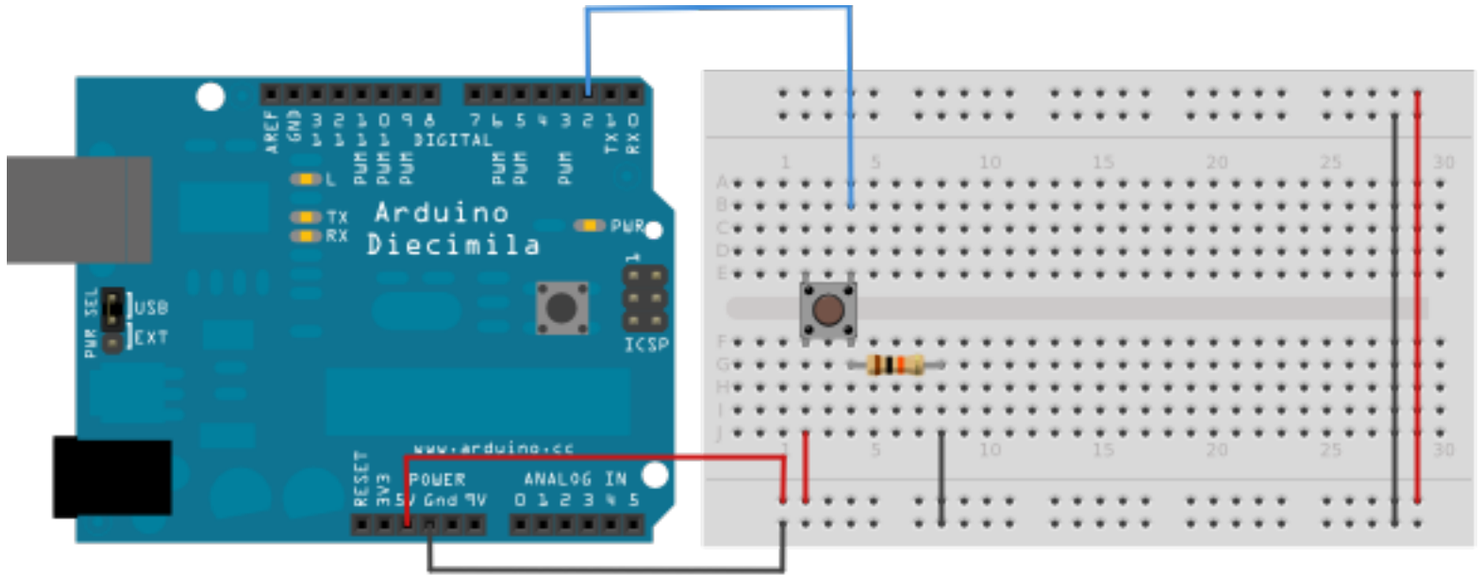
buttons!

MAKE IT

Let's build a button.

You need:

_ 10K resistor
_ handmade button
_ LED
_ jump wire



buttons!

MAKE IT

Let’s build a button.

Upload this sketch:
`File>Examples>Digital>Button`

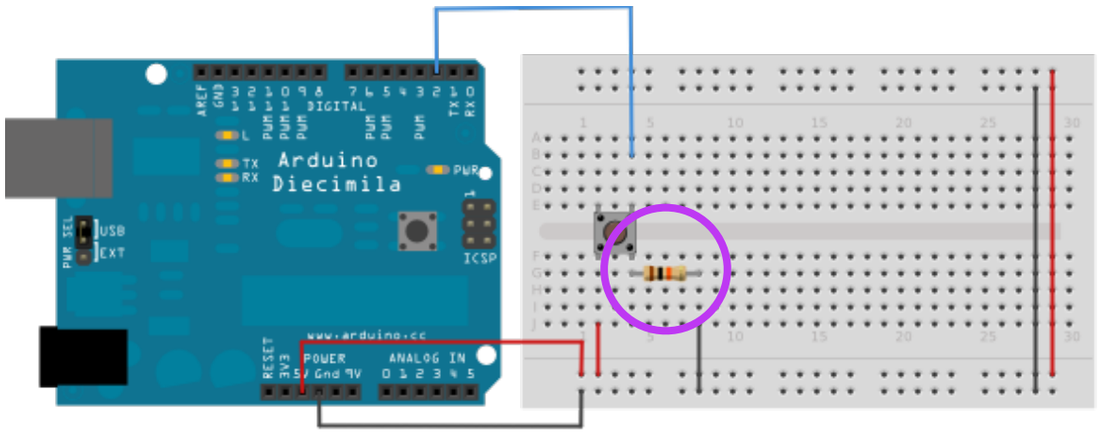
Once you’ve done that, try this one:
`File>Examples>Digital>Debounce`

Notice a difference?

buttons!

MAKE IT

Why do we need a resistor here?



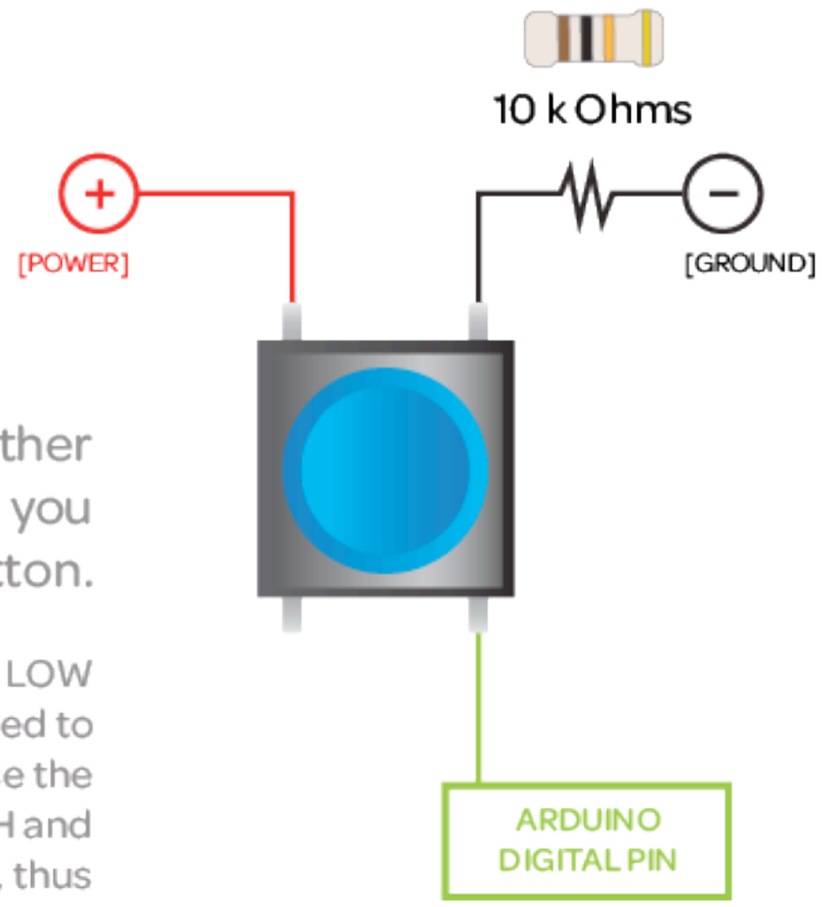
We need to ground electricity or else it will be flying all over the place.

We do this by adding a resistor to ground or power on the same path as the pin that controls the button.

buttons!

MAKE IT

Pull Up Resistor

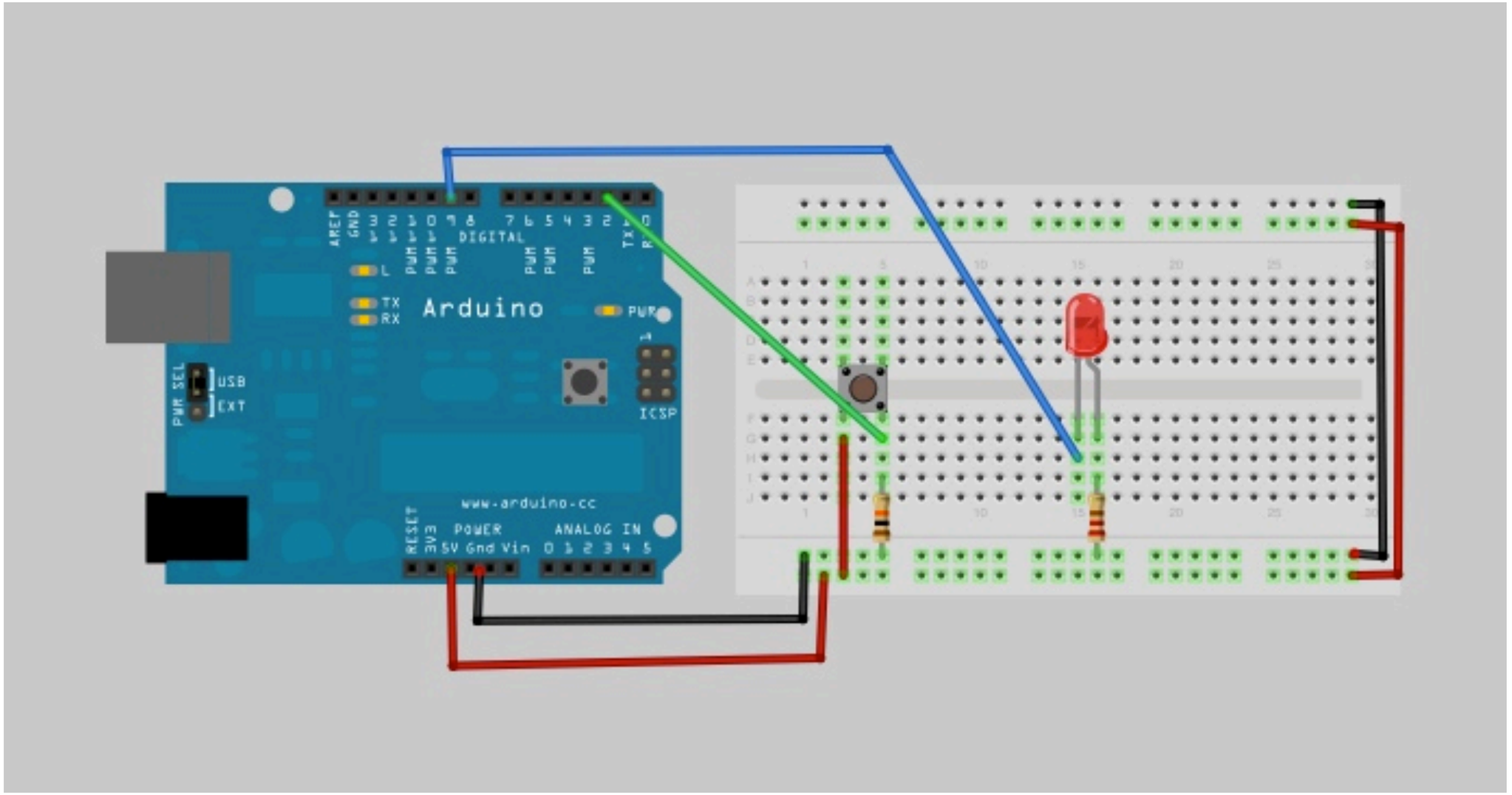


The LED (or other actuator) is ON until you press the button.

Arduino reads the pin as LOW when it is connected to ground. When we close the circuit, it reads HIGH and sends 5 volts to our LED, thus turning it on.

buttons!

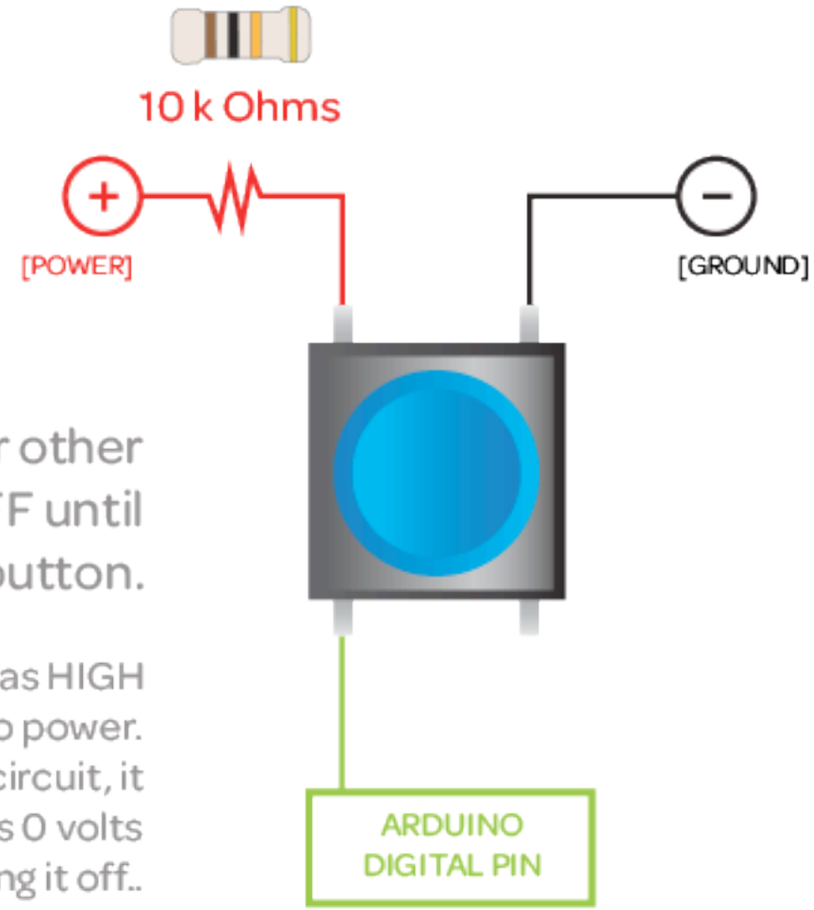
PULL UP RESISTOR



buttons!

MAKE IT

Pull Down Resistor

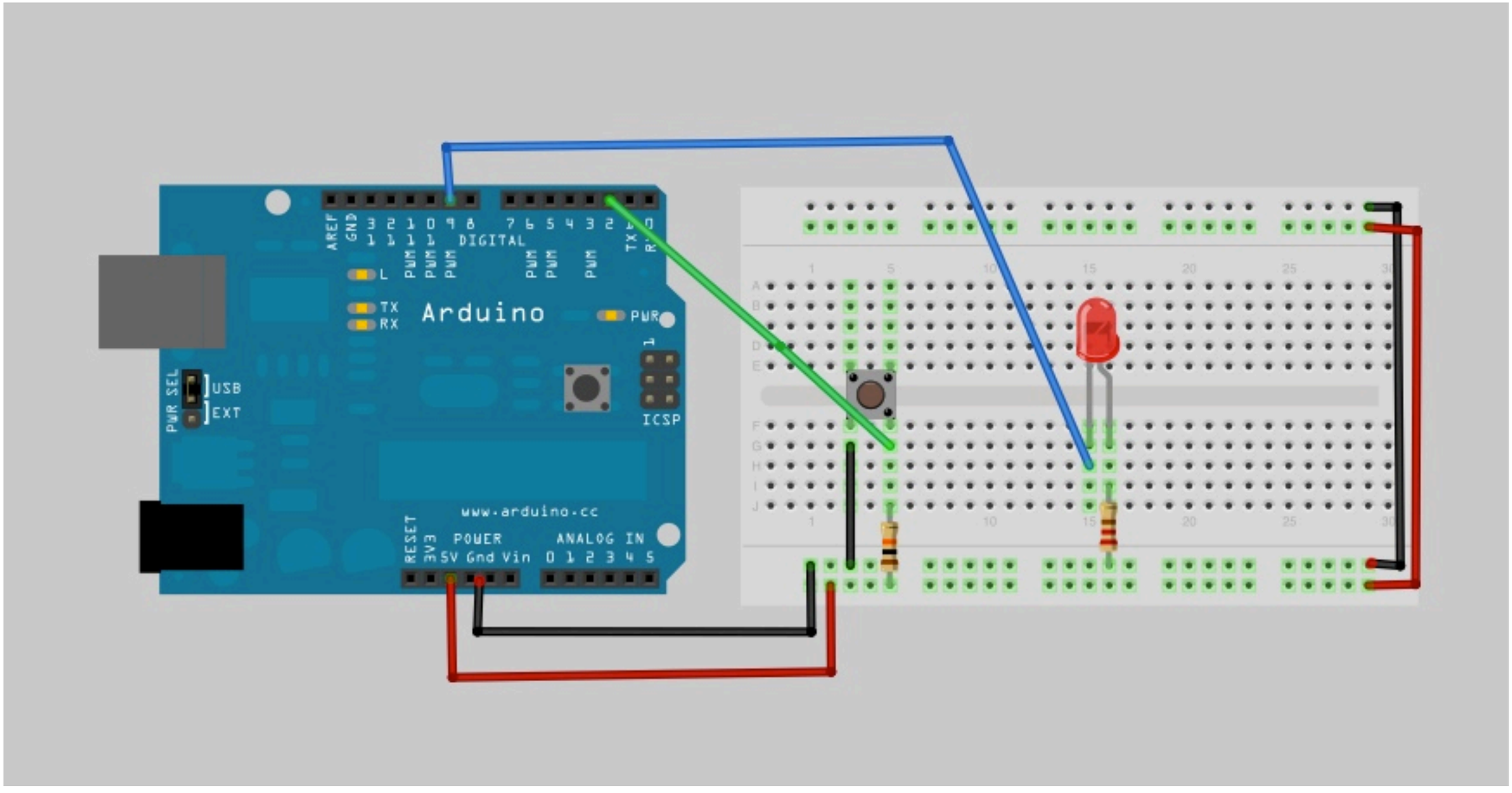


The LED (or other actuator) is OFF until you press the button.

Arduino reads the pin as HIGH when it is connected to power. When we close the circuit, it reads LOW and sends 0 volts to our LED, thus turning it off..

buttons!

PULL DOWN RESISTOR



analog output

arduino

ANALOG OUTPUT PINS

When we faded LEDs using analog means, we controlled the brightness of the LED by controlling the voltage.

//Remember variable resistance?

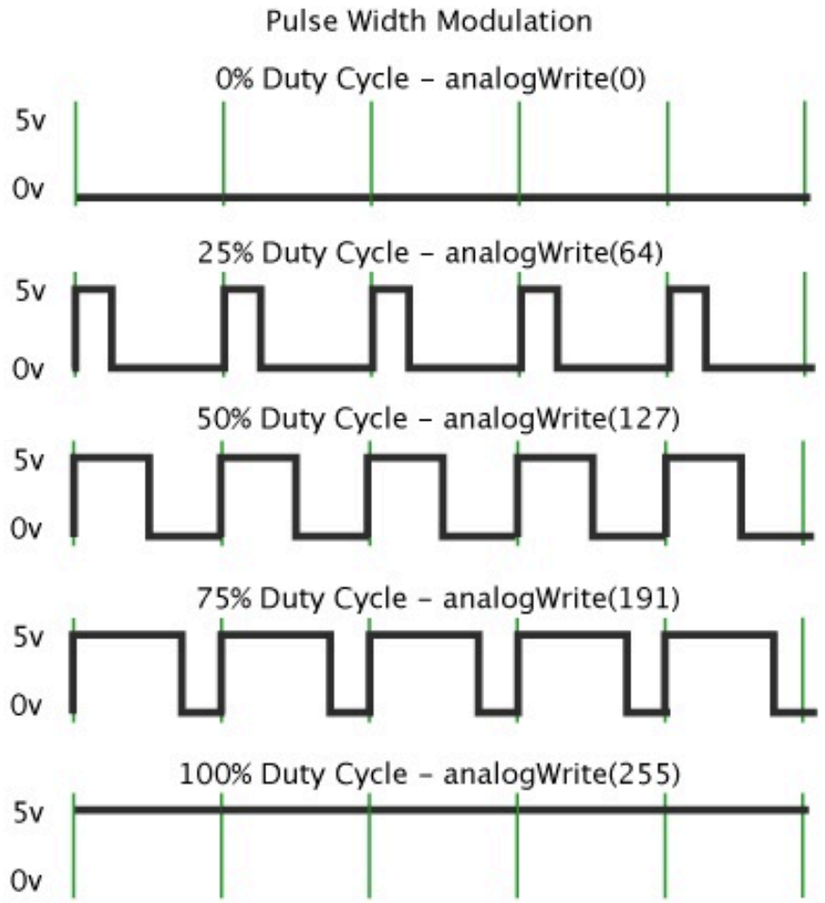
arduino

ANALOG OUTPUT PINS

When we fade an LED with Arduino, we don't actually change the voltage.

We simulate analog behavior by turning a signal on and off at different frequencies, because Arduino is not truly analog.

This is called Pulse Width Modulation.

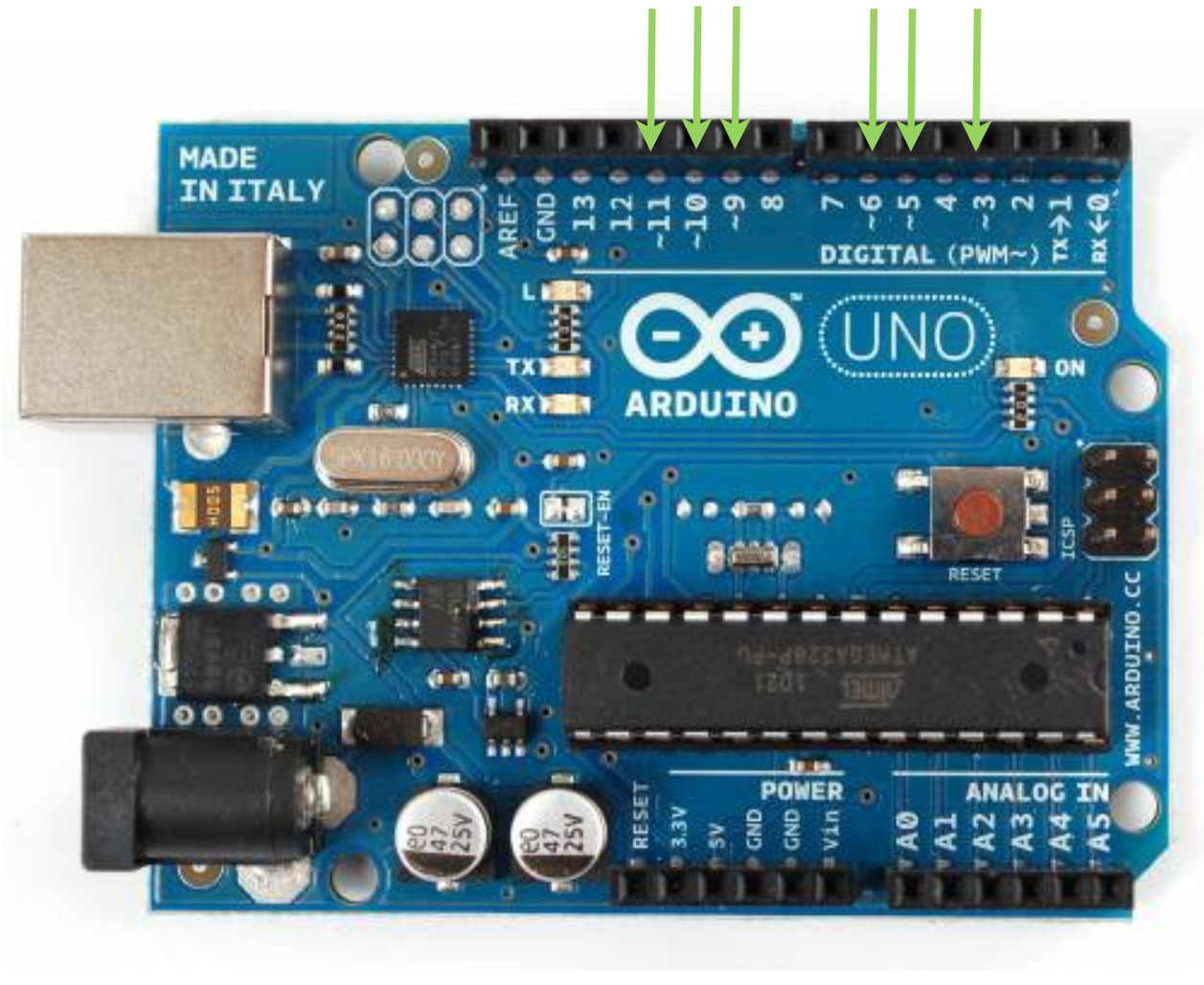


pulse width modulation

ANALOG OUTPUT PINS

Only a few pins can execute this function: 3, 5, 6, 9, 10, and 11.

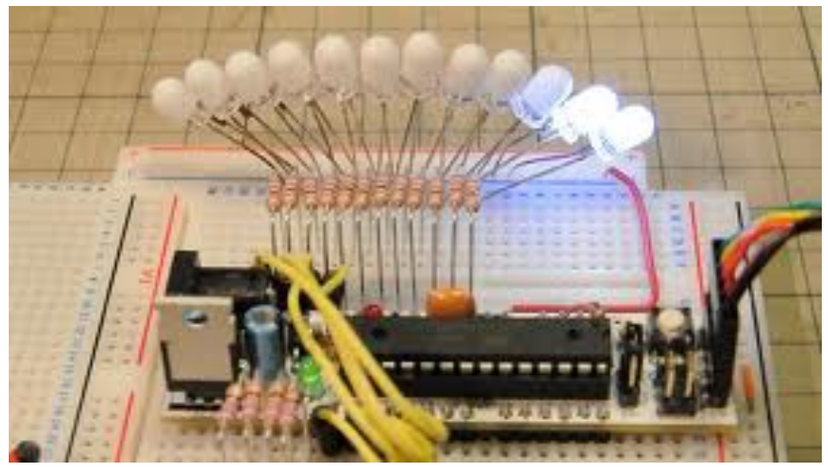
These are special because they can be digital I/O OR analog out.



pulse width modulation

ANALOG OUTPUT PINS

Let’s see what this looks like. Because fading is pretty.



Upload this sketch:

`File>Examples>Basics>Fade`

Once you’ve done that, try this one:

`File>Examples>Analog>Fade`

Notice a difference?

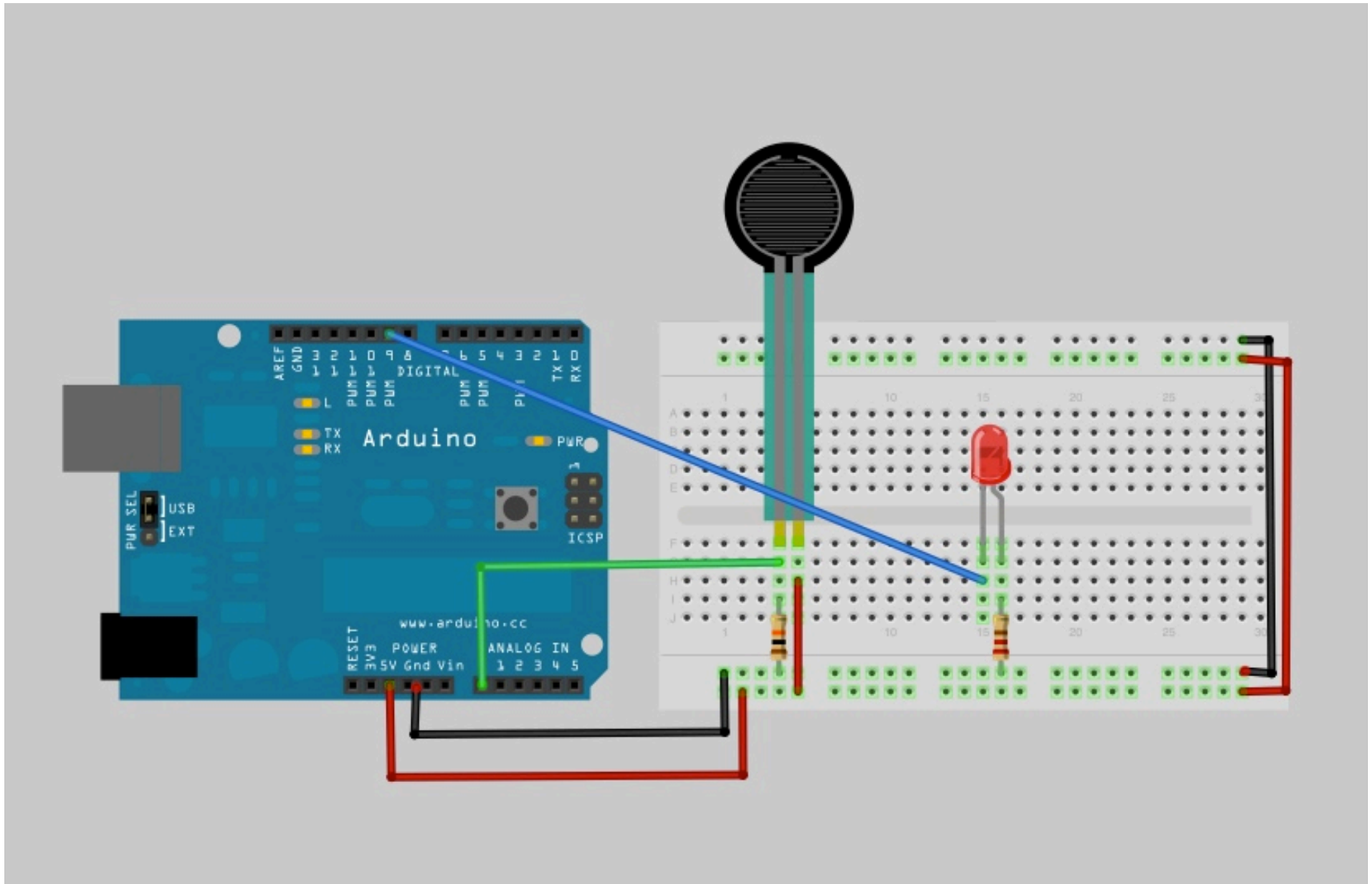
Try changing the values to fade the LED at different speeds.

pulse width modulation

ANALOG OUTPUT PINS

Now add an LED to pin 9 and upload

`SensorValSketch_map`



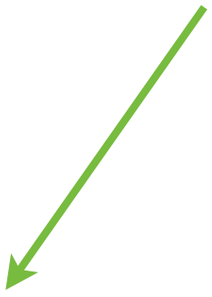
pulse width modulation

ANALOG OUTPUT PINS

Add an LED to pin 9 (you should still have your sensor attached)

Upload the
`SensorValSketch_map`

This is where your serial monitor becomes important!



`analogWrite(pin,value)`

The value can be between 0 - 255.
For OUTPUT

`map(value, fromLow, fromHigh, toLow, toHigh)`

We can control the range of the values by using this function.

analog input

arduino

ANALOG INPUT PINS

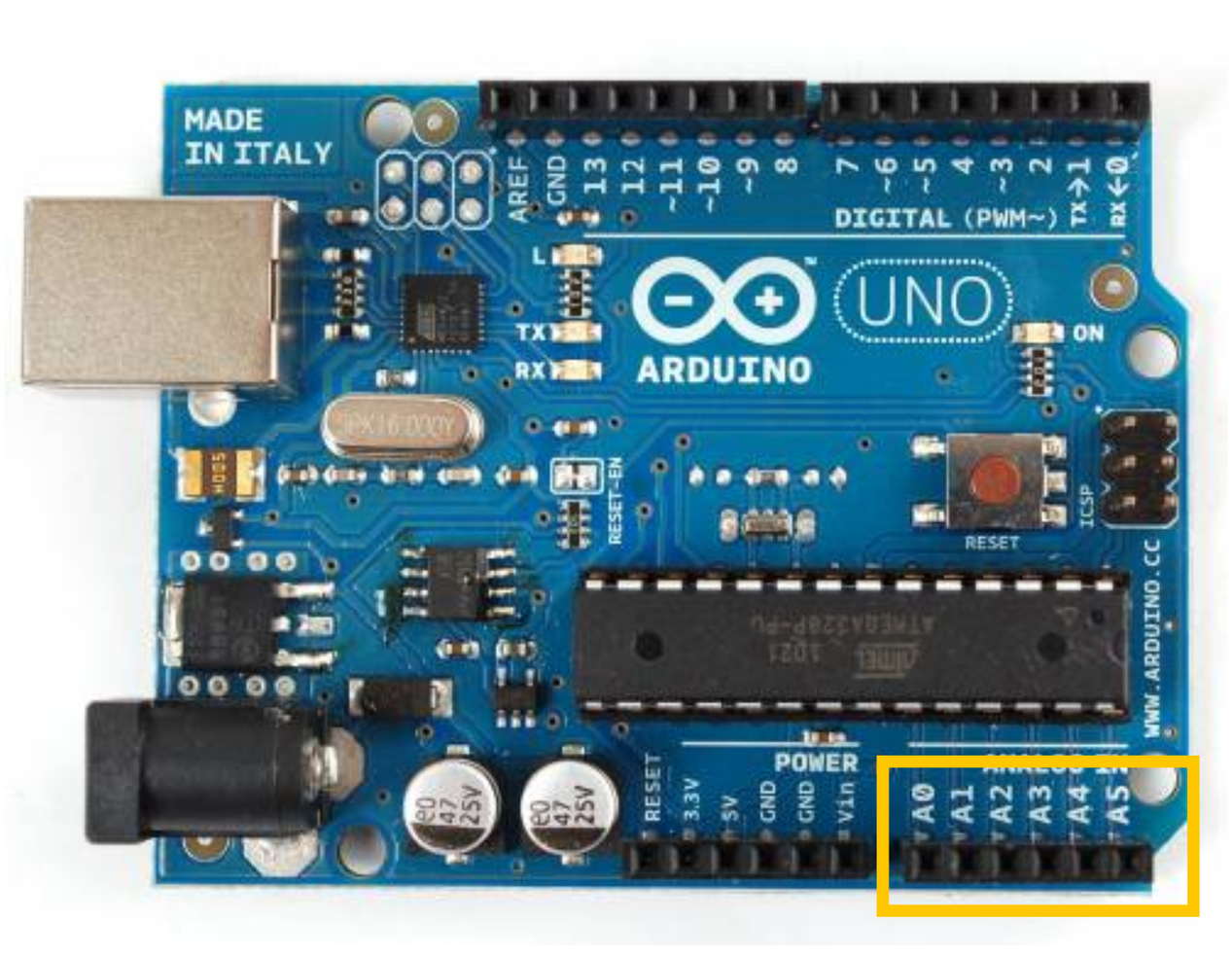
6 Analog Input pins

You can read or write a wide range of values

Read	0 - 1023
Written	0 - 255

arduino

ANALOG INPUT PINS



`analogRead(pinNumber)`

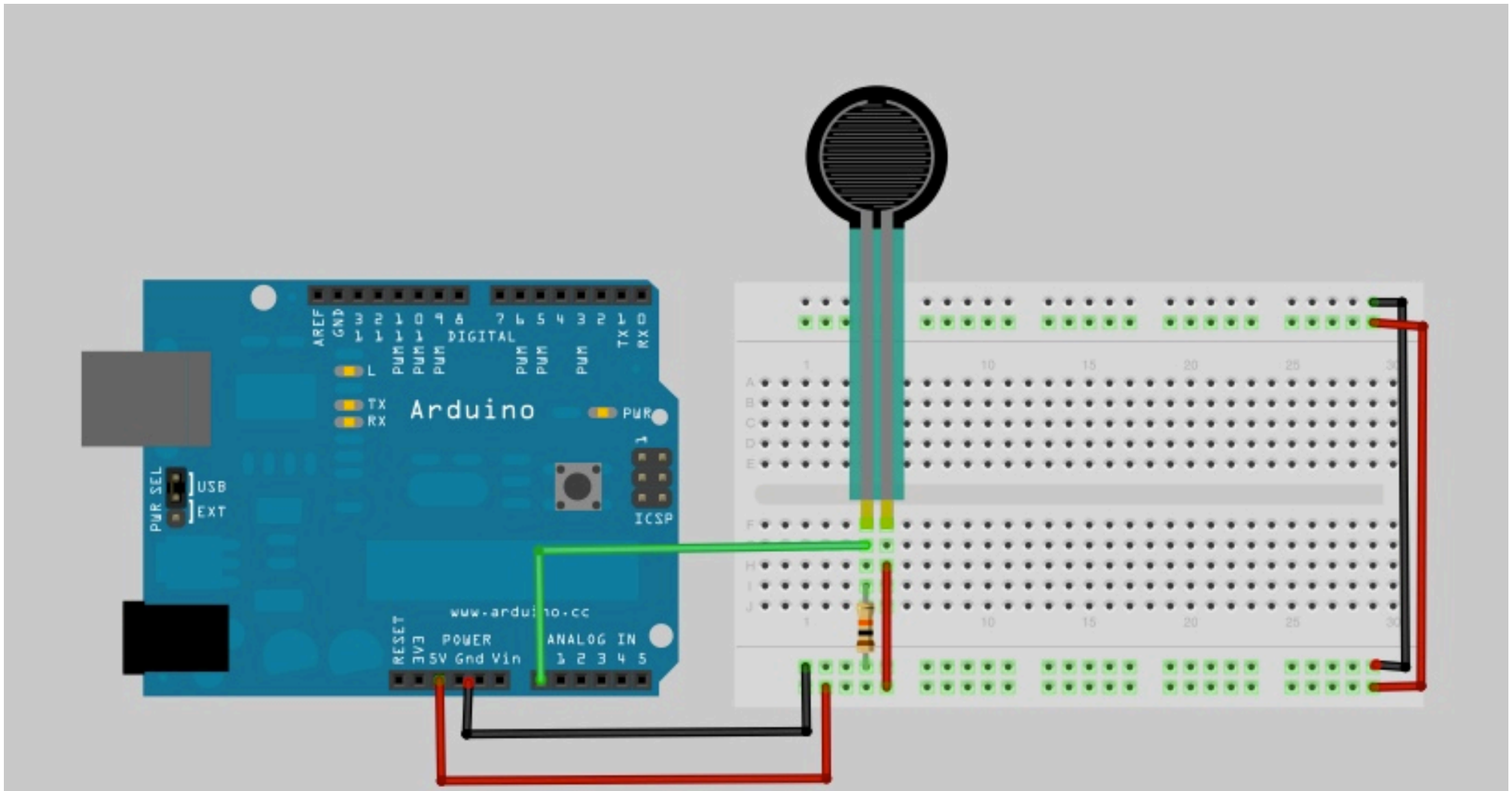
Reads value from specified analog in pin

For INPUT

arduino

MAKE IT

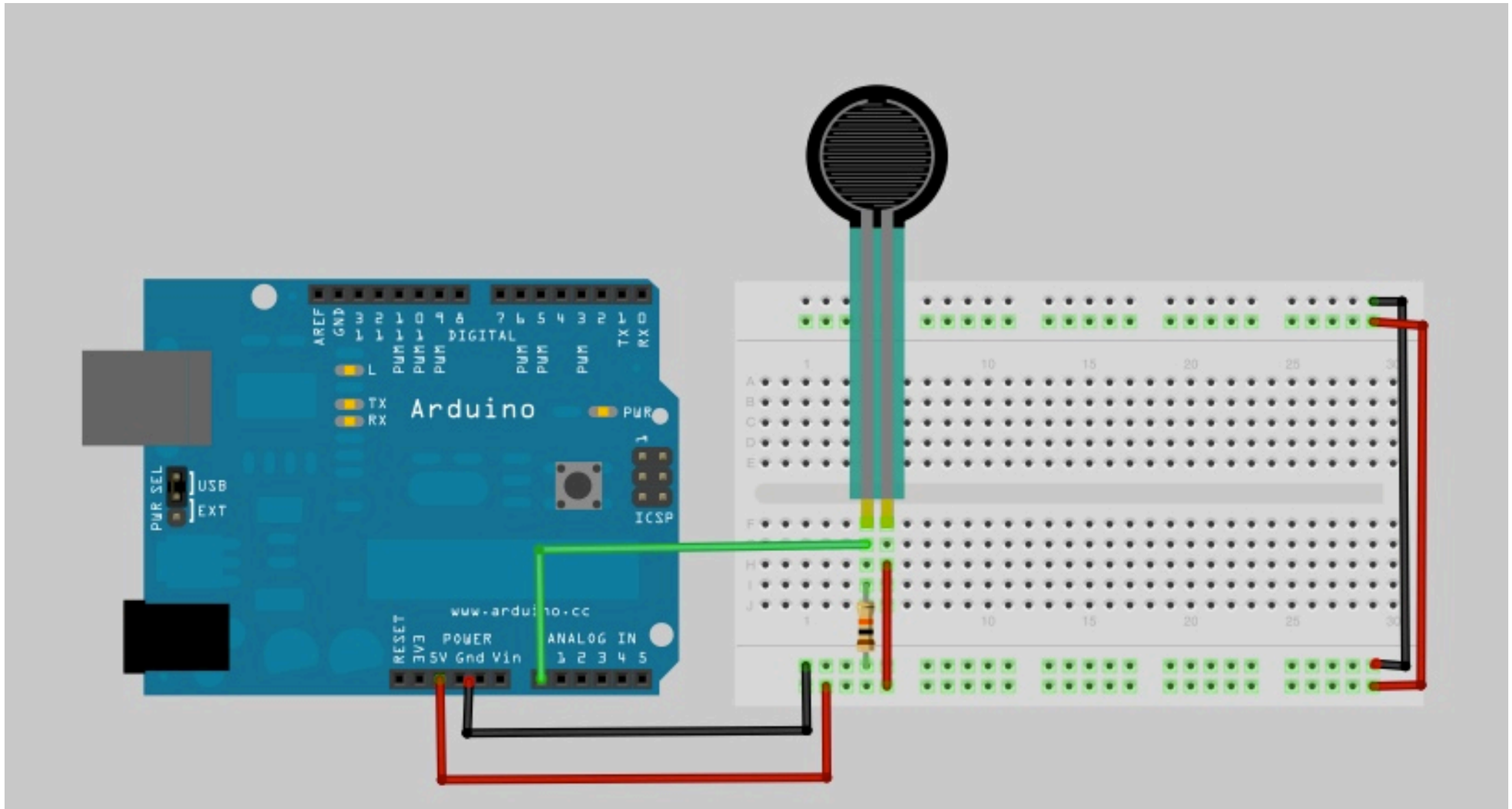
Build a circuit using one of your sensors.



arduino

MAKE IT

Upload the `SensorValSketch_no_map`



arduino

MAKE IT

Build a circuit using one of your sensors.

Click the Serial Monitor button.

serial communication

THIS AIN'T YO BREAKFAST

If you want to read the values coming in from `analogRead()`, use the serial monitor.

Arduino usually communicates at a baud rate of 9600. This is the rate Arduino and the computer agree to exchange information.

This is also your best way to debug your program.

resistors

COLOR-CODED

But how do you know which value to use?

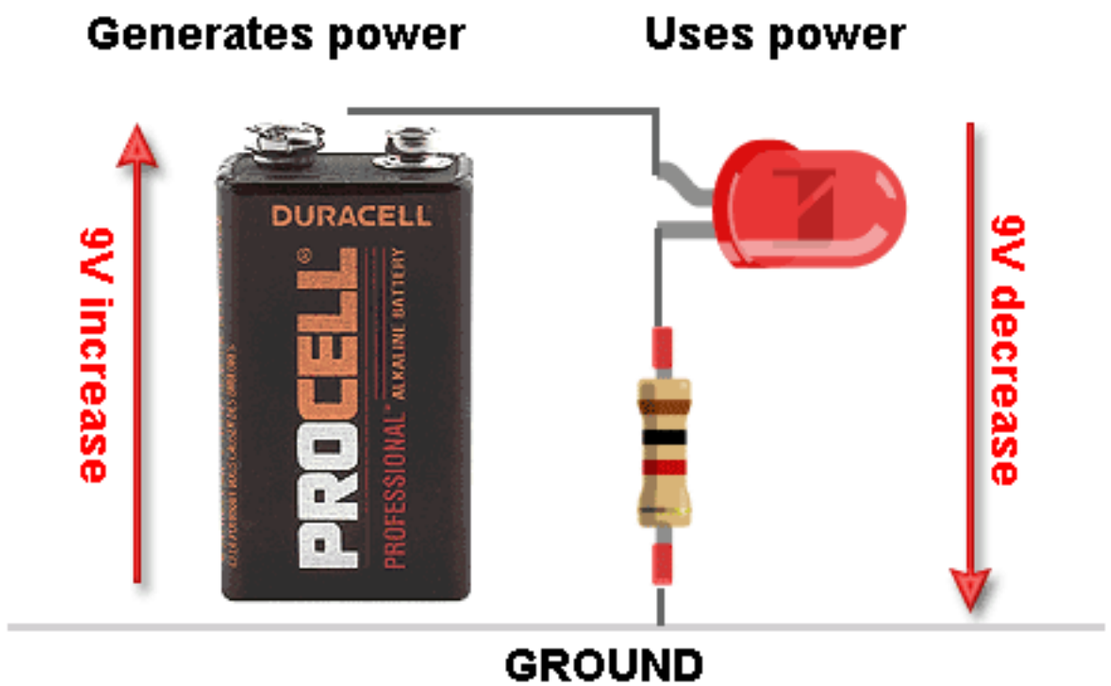
Generally you will only need to use the ones on the last slide. However, the more you understand resistance and it's relationship to current and voltage, the more you will be able to do.

If you want to know more about this, see Week 5: Ohm Sweet Ohm

kirchoff’s voltage law

LOOPS!

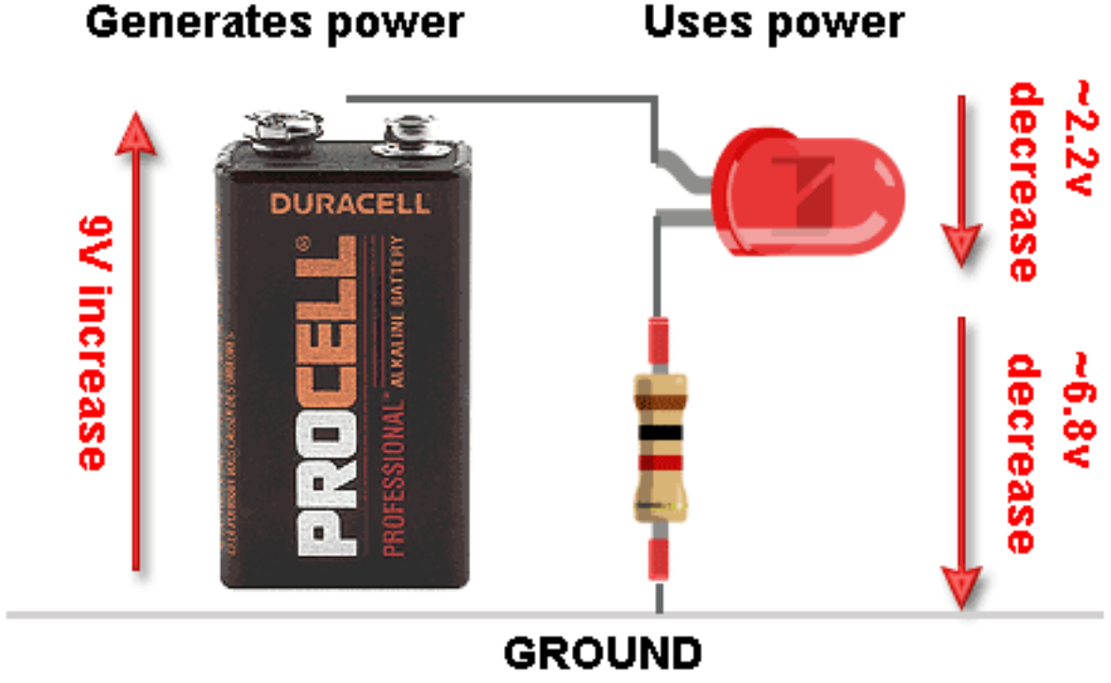
In any 'loop' of a circuit, the voltages must balance: the amount generated = the amount used



kirchoff’s voltage law

LOOPS!

Let’s break it down:



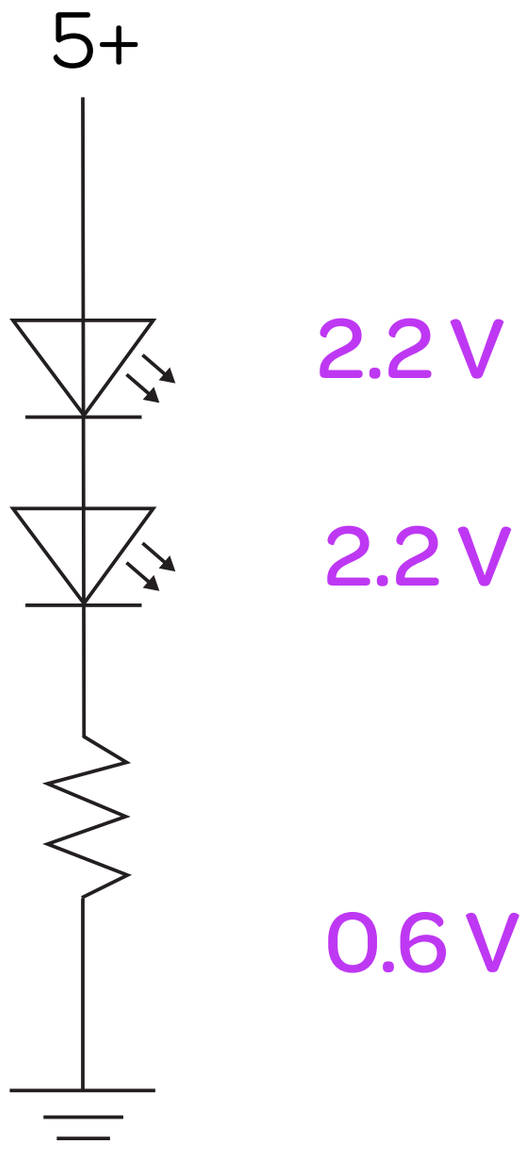
Forward Voltage (V_F)
= voltage lost to the LED (or other component)

kvl + ohm

AN EXAMPLE

We want to know the resistance to we need to have the LEDs running at full brightness.

$R = V/I$

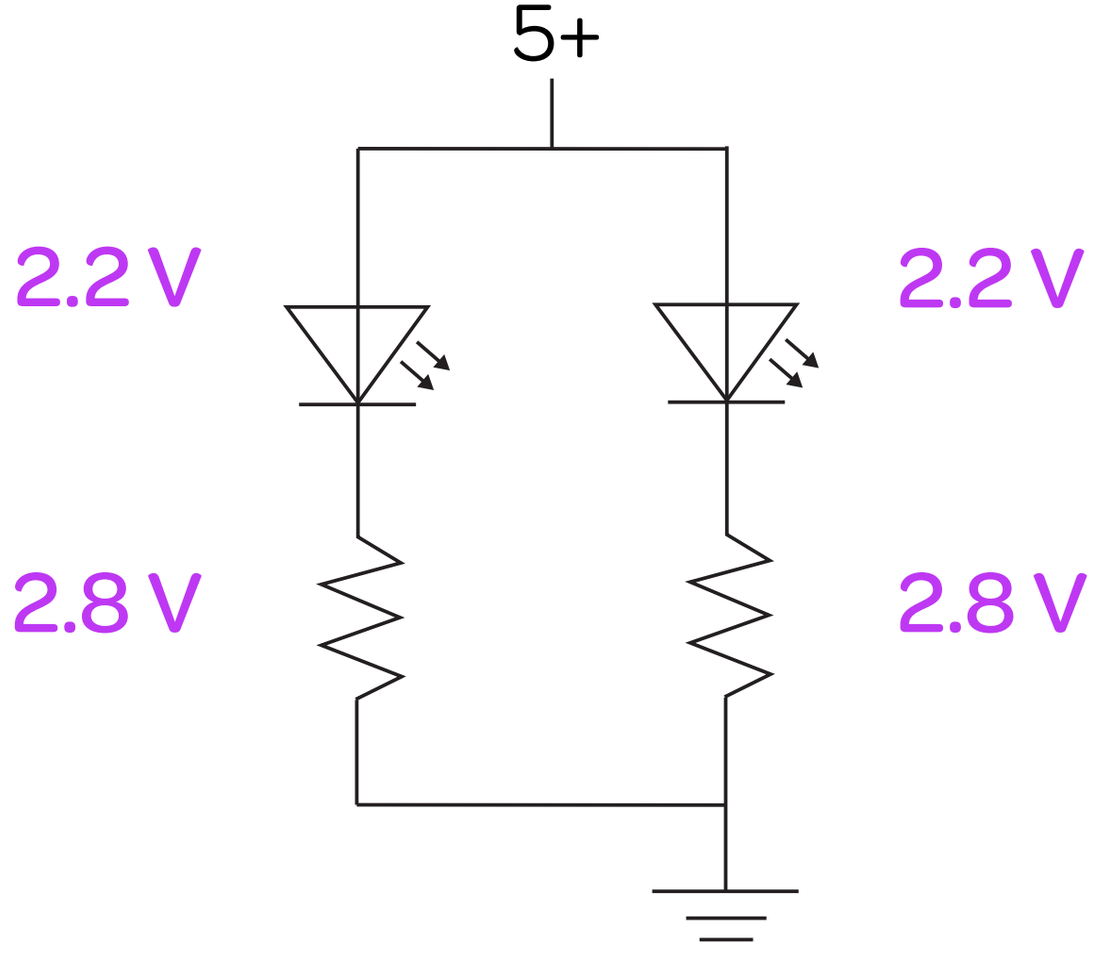


kvl + ohm

AN EXAMPLE

We want to know the current to know how bright the LED will be.

$I = V/R$

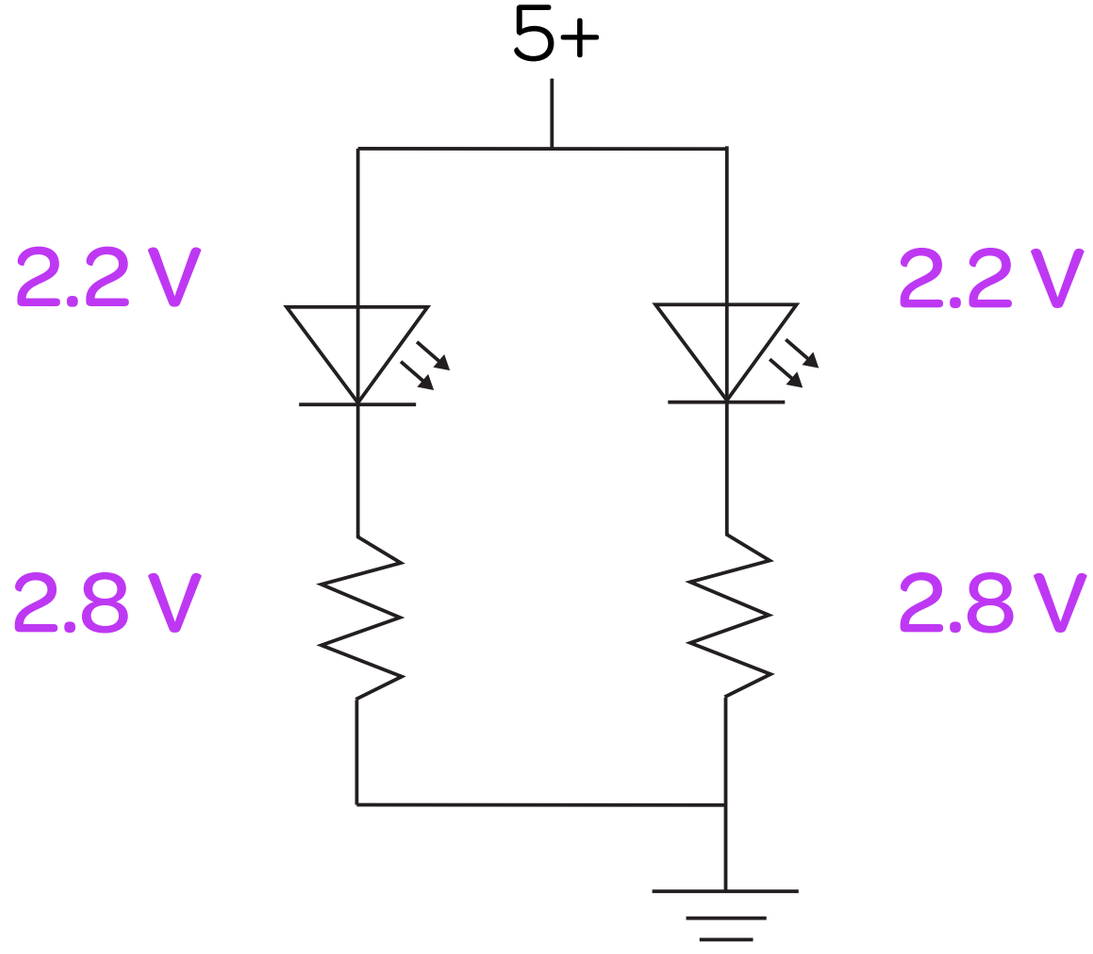


kvl + ohm

AN EXAMPLE

We want to know the current to know how bright the LED will be.

$I = 2.8/R$

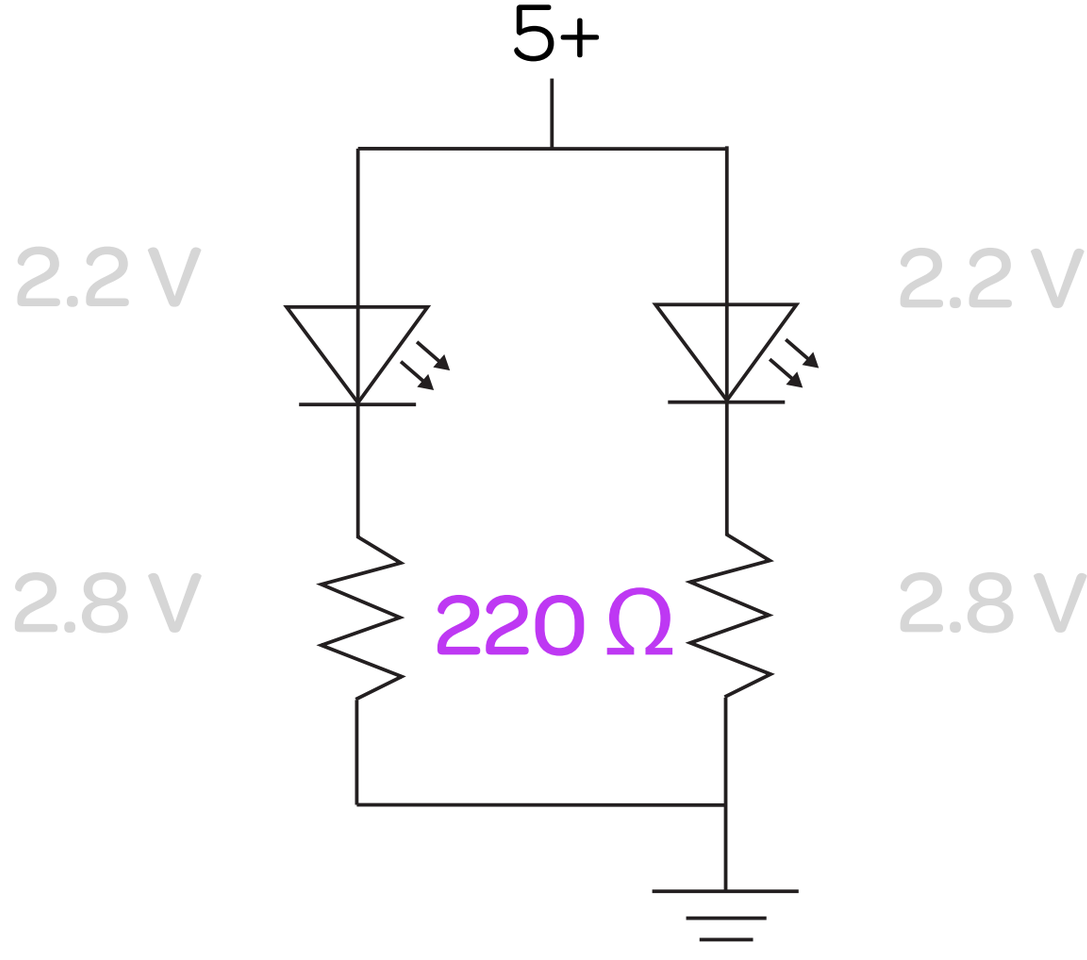


kvl + ohm

AN EXAMPLE

We want to know the current to know how bright the LED will be.

$$I = 2.8/220$$



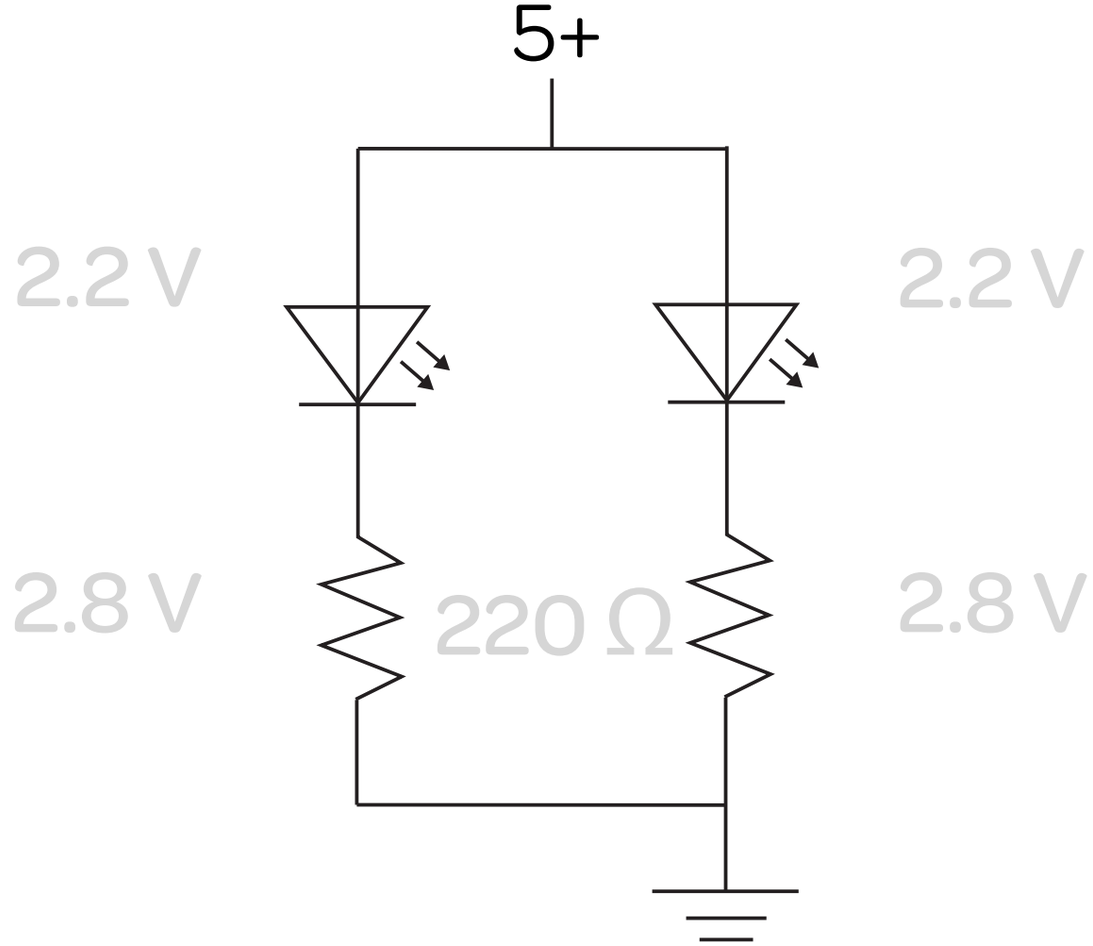
kvl + ohm

AN EXAMPLE

We want to know the current to know how bright the LED will be.

0.0127 =
2.8/220

12.7 mA

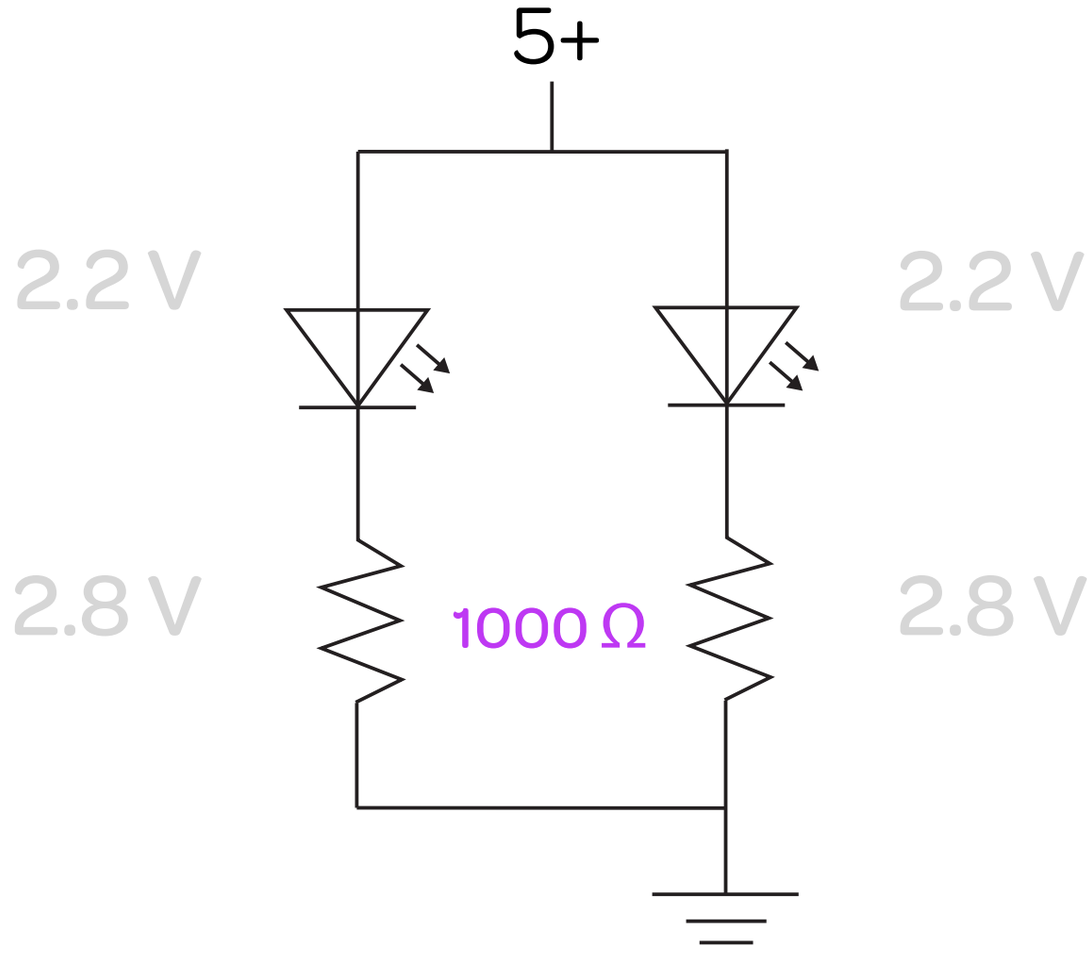


kvl + ohm

AN EXAMPLE

AGAIN! This time with 1K!

$I = 2.8/1000$



kvl + ohm

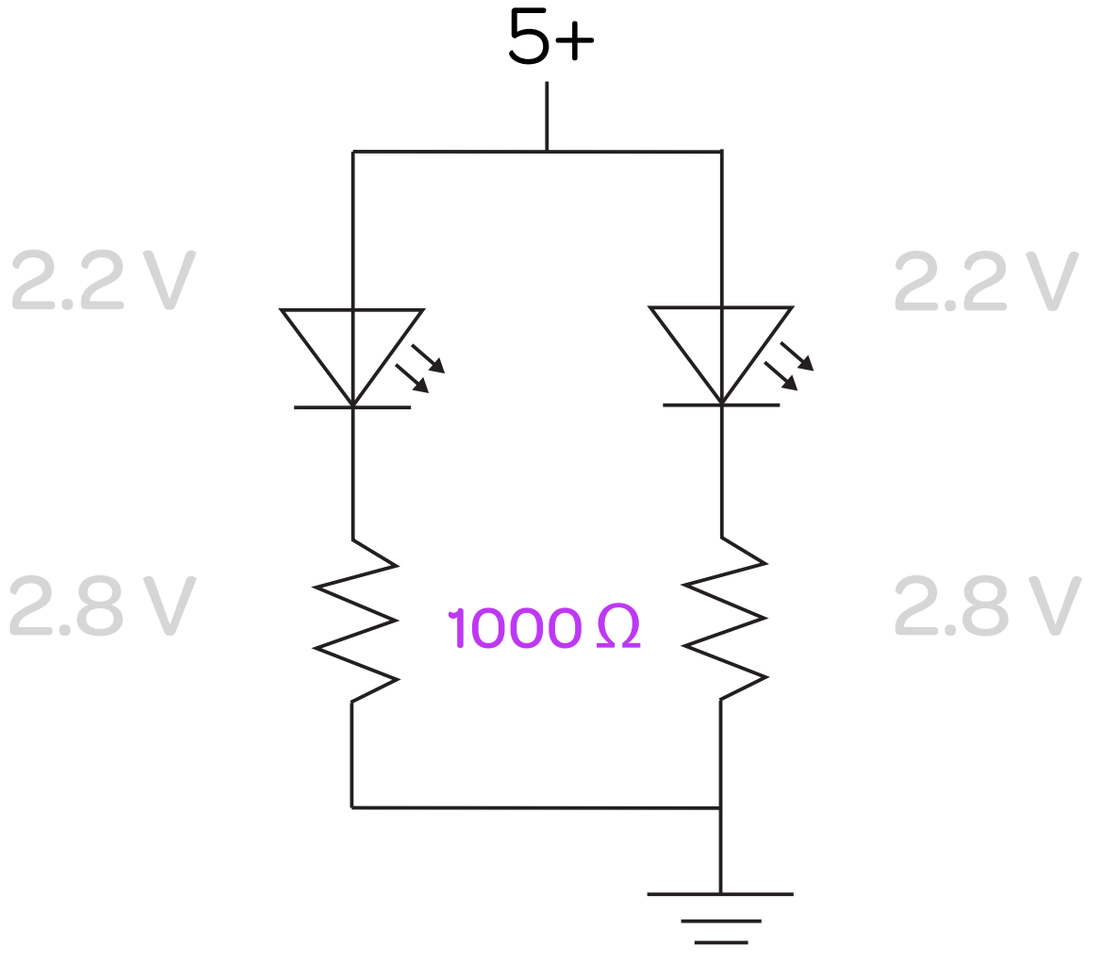
AN EXAMPLE

AGAIN! This time with 1K!

$I = 2.8/1000$

2.8 mA

So which is brighter?

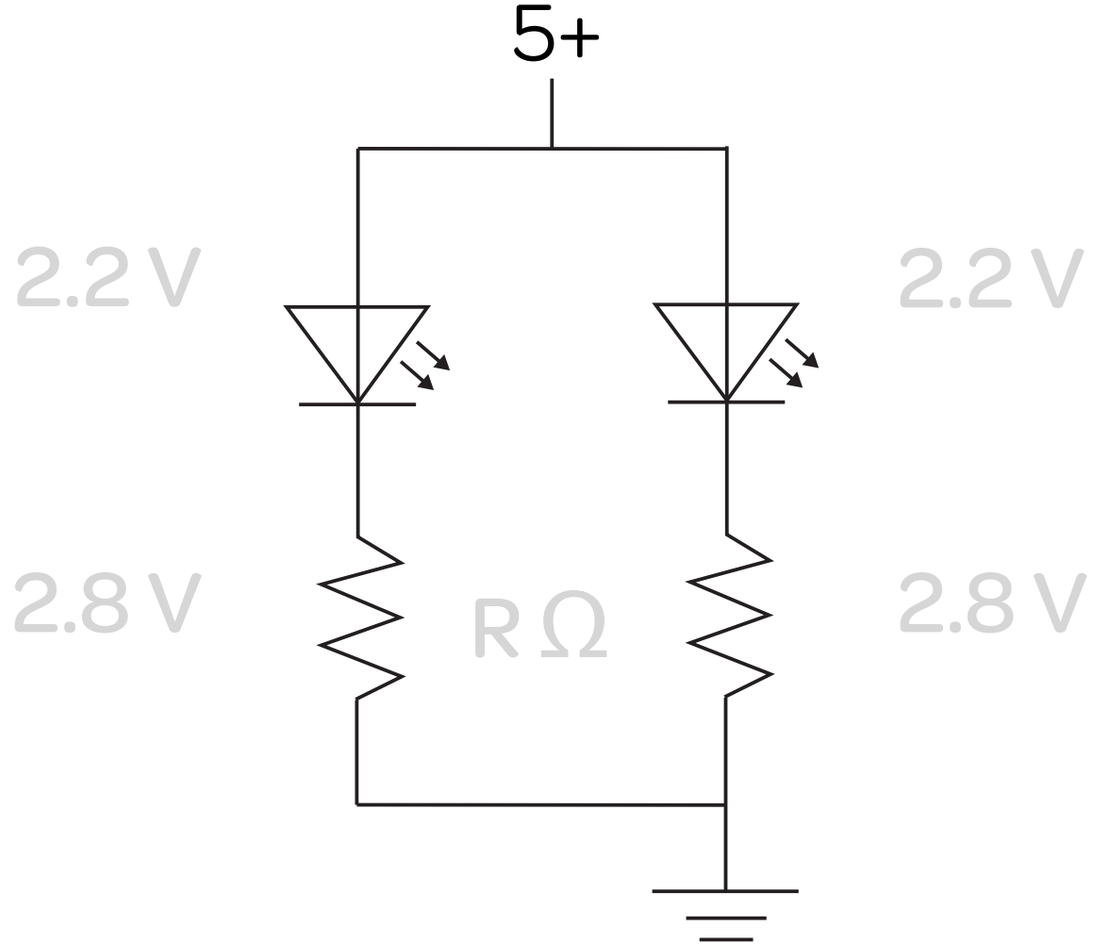


kvl + ohm

ANOTHER EXAMPLE

Let’s find the least valued resistor we can use without burning our LED.

$R = V/I$



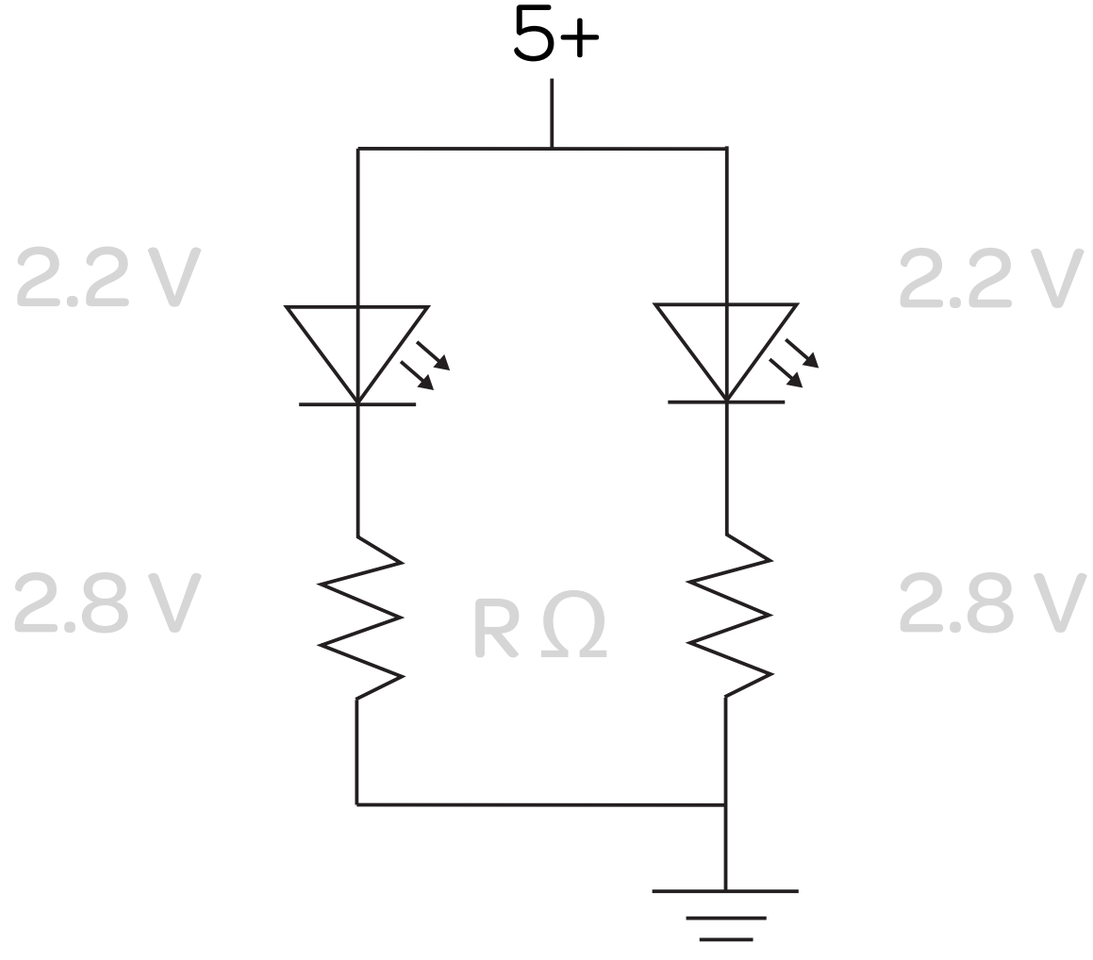
kvl + ohm

ANOTHER EXAMPLE

Let’s find the least valued resistor we can use without burning our LED.

$R = V/I$

$R = 2.8 / 0.02$



kvl + ohm

ANOTHER EXAMPLE

Let’s find the least valued resistor we can use without burning our LED.

$R = V/I$

$R = 140\ \Omega$