INDUSTRIAL UNIVERSITY OF HOCHIMINH CITY

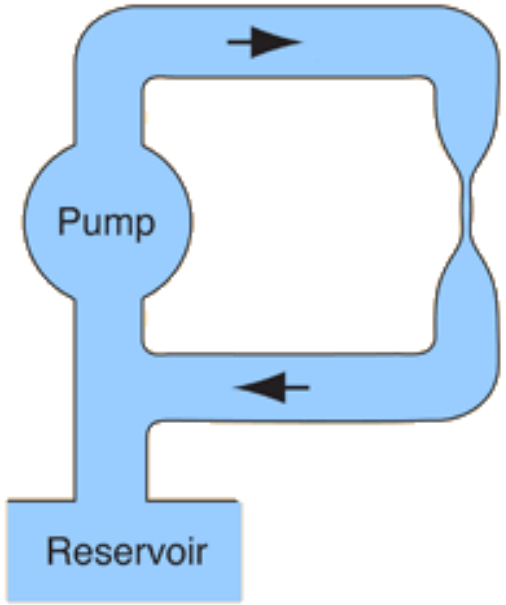
Modern IoT Technology

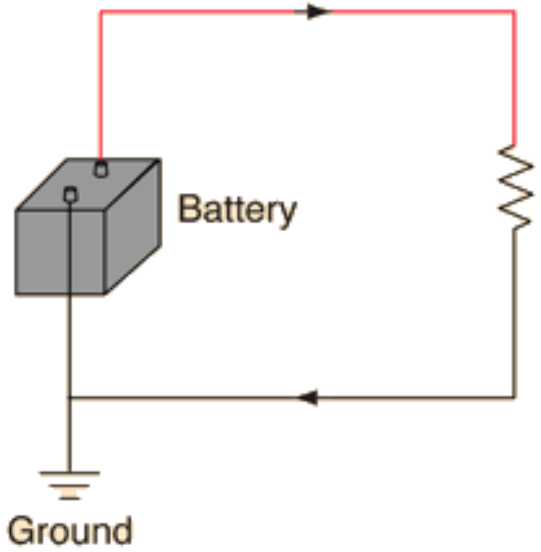
An Introduction to Arduino

Basics of Electricity

- Take part in class
- Tests
  - *Regular test*
  - *Midterm*
  - *Final Exam*

Electrical current



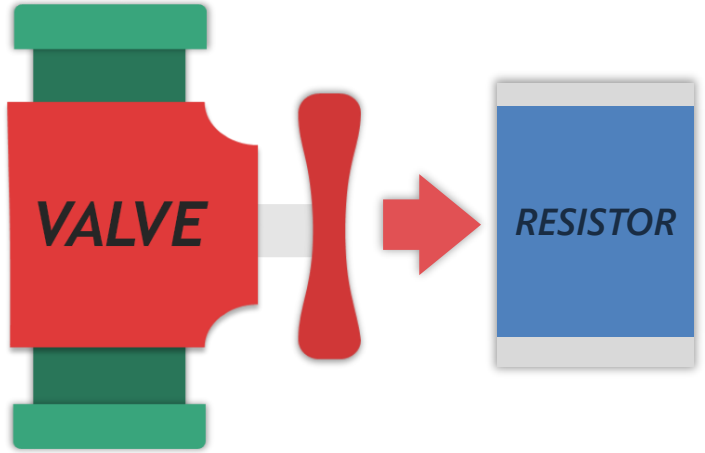


Power Source

- Have a positive and negative side (Battery Example)
- Difference in Charge or Potential Energy
- Measured in “Volts” (V)

Resistor

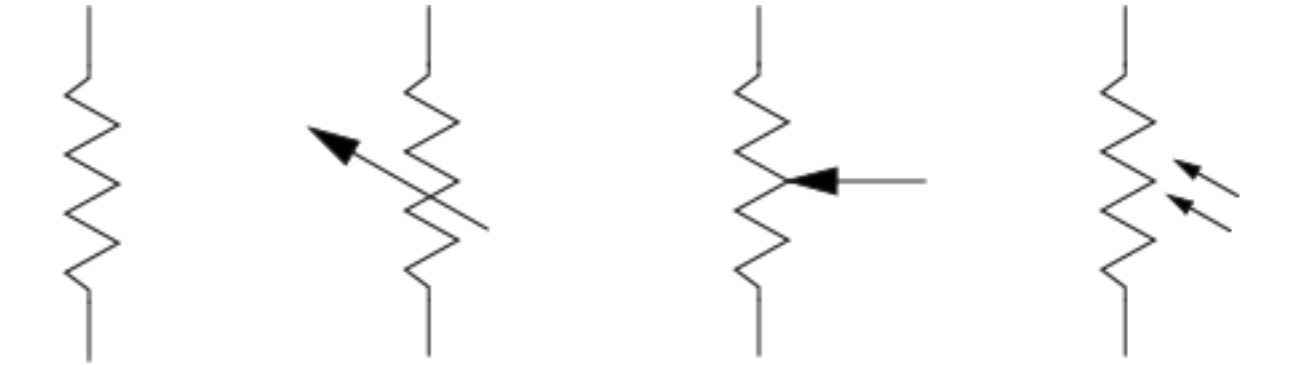
- Just as a valve limits the rate of flow of water, a resistor limits the rate of flow of electrons.



- Resistance describes how much a resistor "pushes back" against the flow of charge.

- The unit of resistance is the **ohm (Ω)**.

Resistor



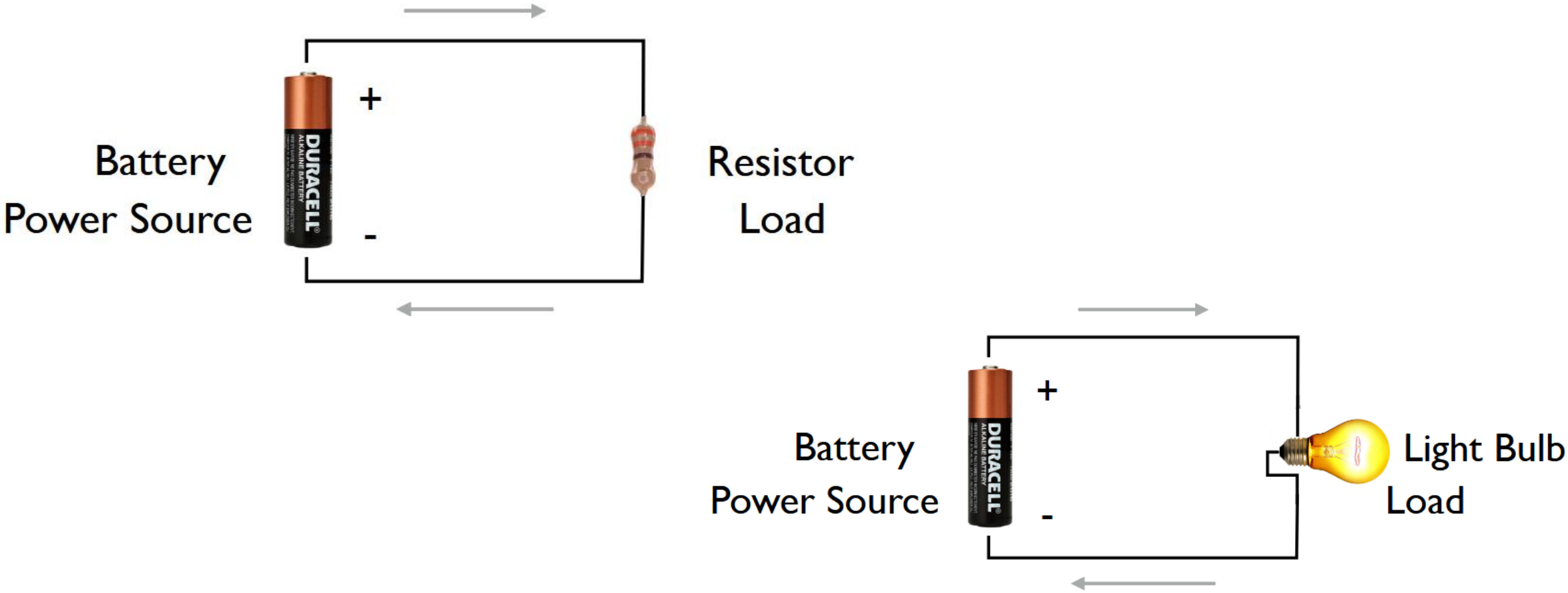
fixed value resistor

variable resistor

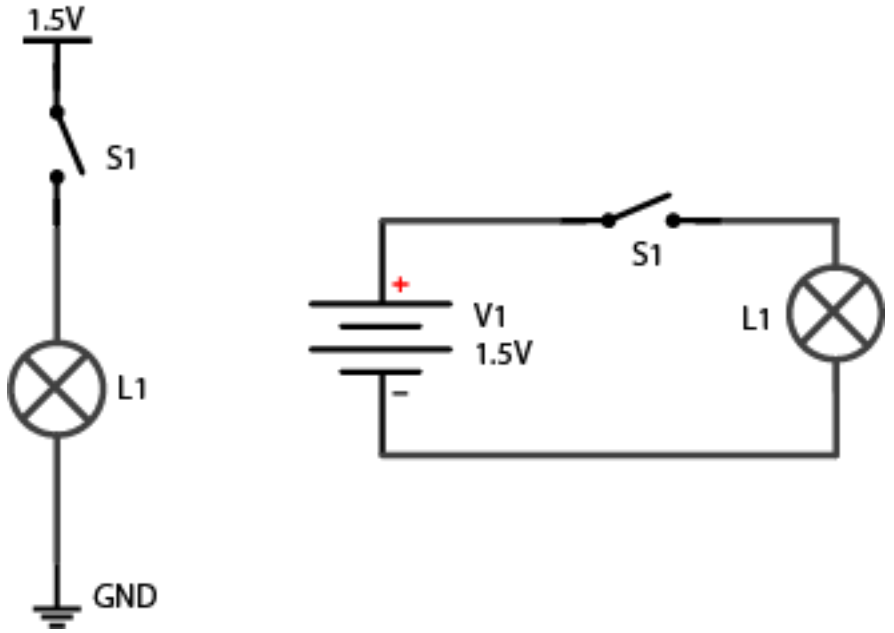
potentiometer

photoresistor or light-dependent resistor

A circuit

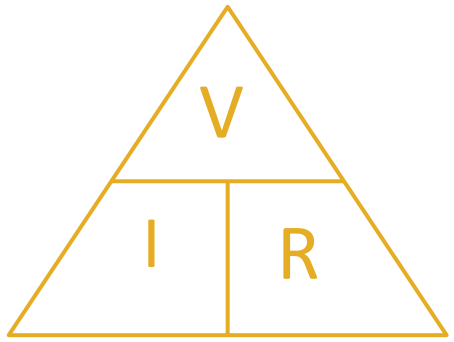

Battery
Power Source
+
-
Resistor
Load
Battery
Power Source
+
-
Light Bulb
Load

Two schematics of a circuit



Ohm's Law

- There is a simple relationship we can use to relate voltage, resistance, and current. It is called Ohm's Law.



Voltage

$V = I * R$

Current

$I = \frac{V}{R}$

Power

$P = I * V$

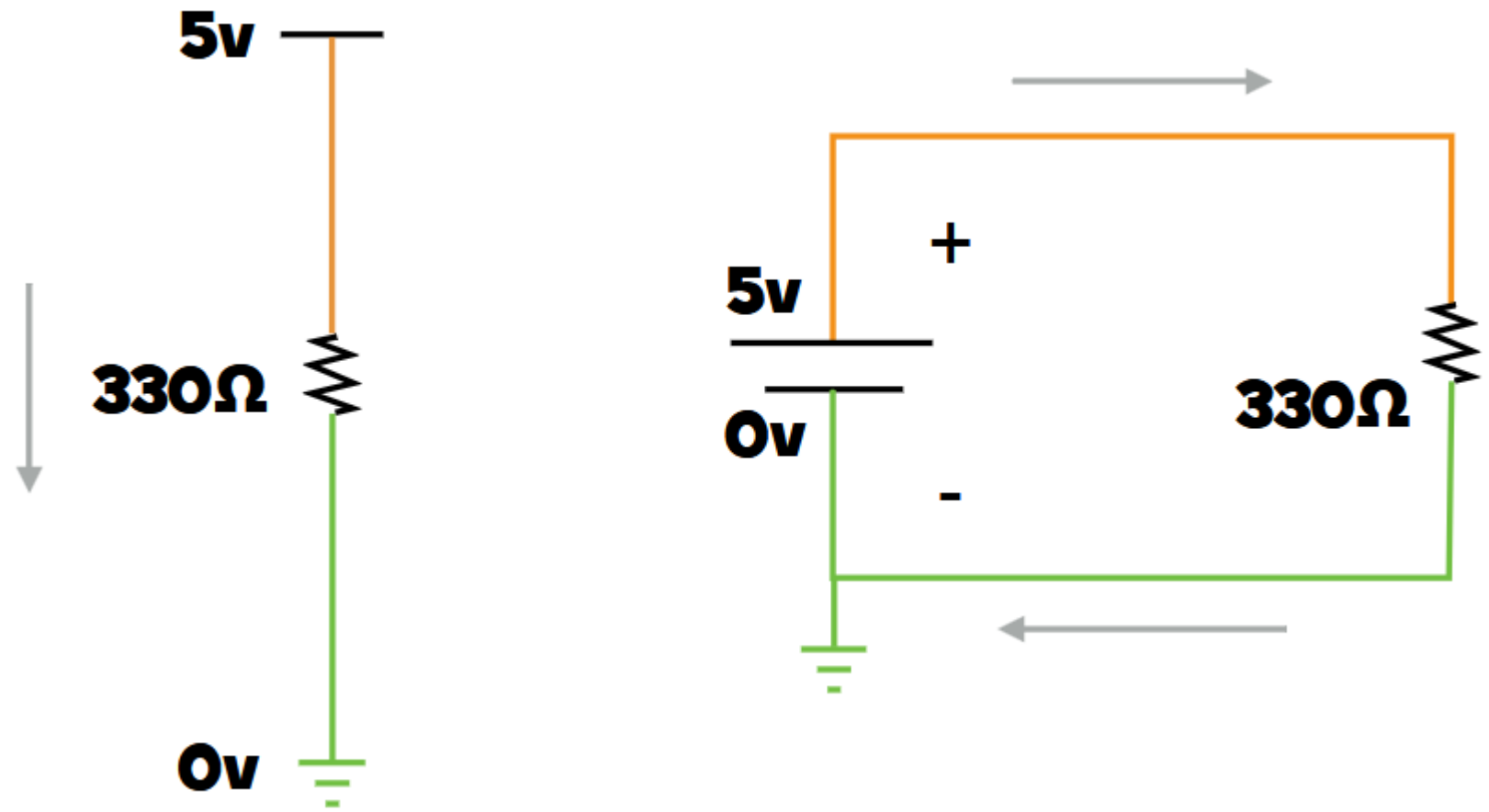
Resistance

$R = \frac{V}{I}$

Ohm's Law Exercise

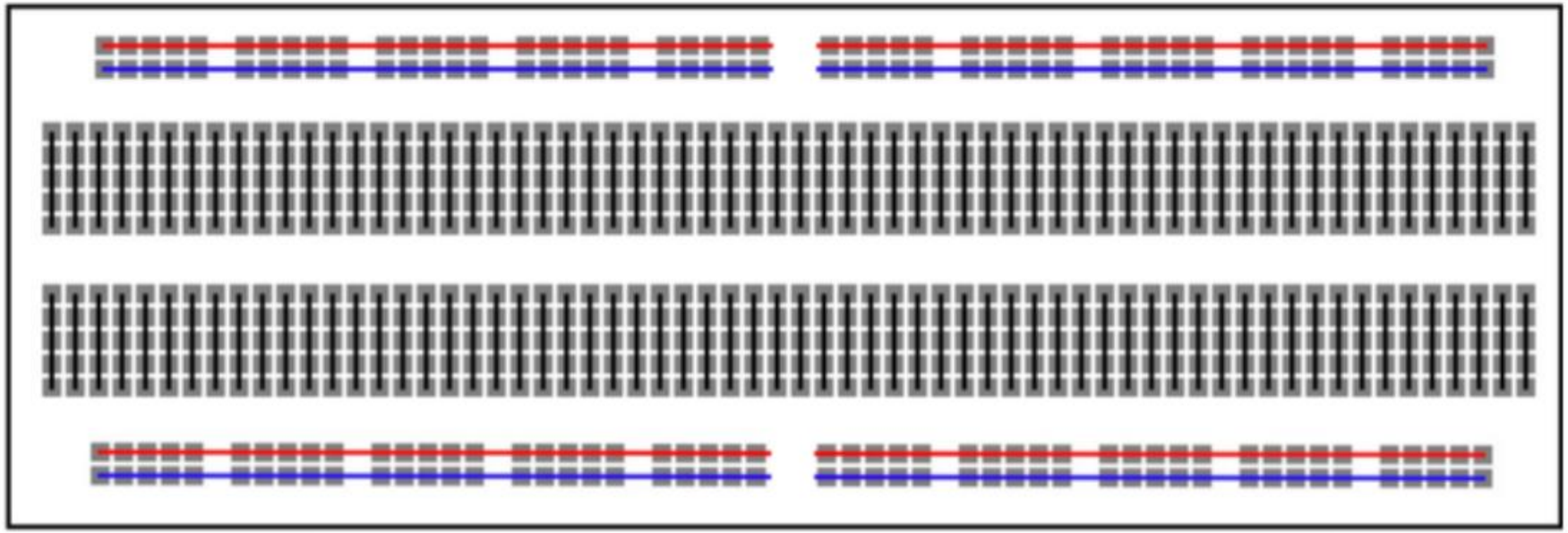
- For example, if we connect a 30Ω resistor across the terminals of a 1.5V battery, what will be the current?
- we replaced the 30Ω resistor with a 15Ω resistor, what will be the current?

Let’s look at a circuit



Kit for Arduino

Breadboard



Data acquisition

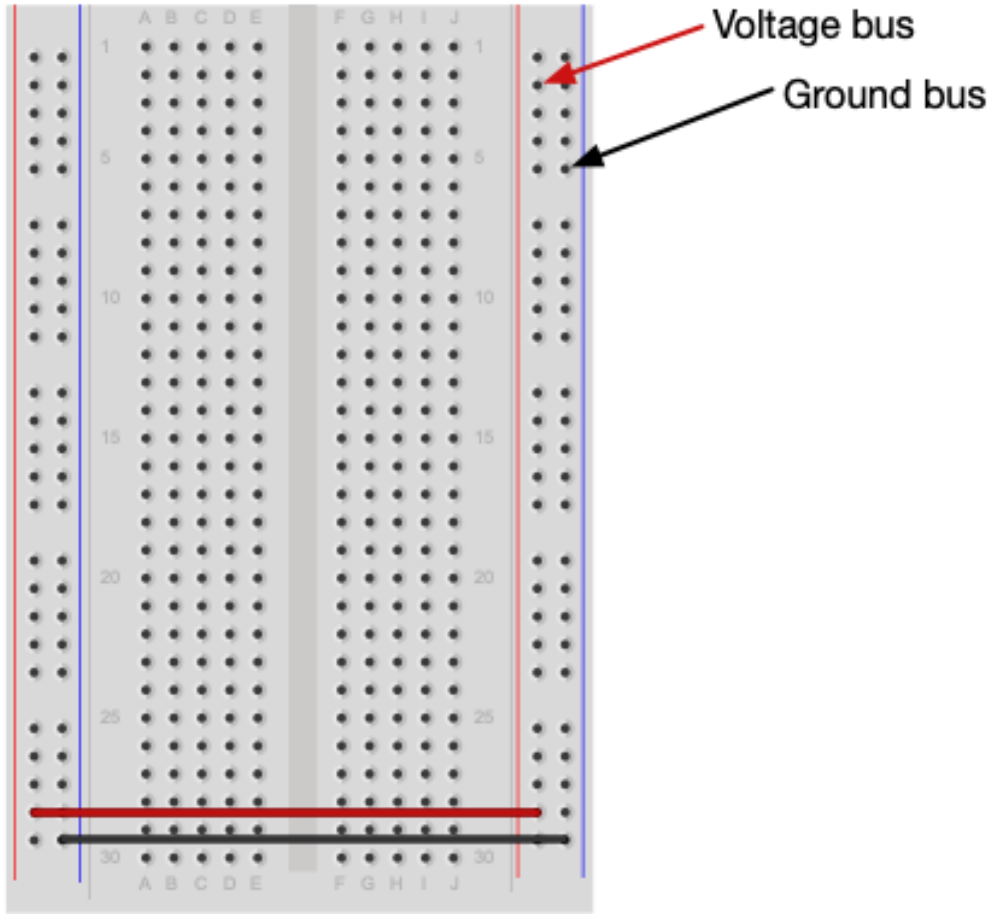
- In the form of sensors
- Function to collect data in the environment to provide real-time results and/or feedback
- Detect and measure physical quantities such as humidity, pressure, speed, light, and temperature.

Data processing and storage

- Battery-powered devices: The simpler the design, the lower the power consumption and costs.
- Externally powered devices: reduces service latency but also conserves wireless backhaul bandwidth

Powering the Breadboard

- The left and right sides of the board are connected via two wires which connect the voltage bus on the left with the one on the right, and the ground bus on the left with the one on the right.



Arduino

- Arduino is an open-source electronic platform for easy use of hardware and software
- It can sense the environment by receiving

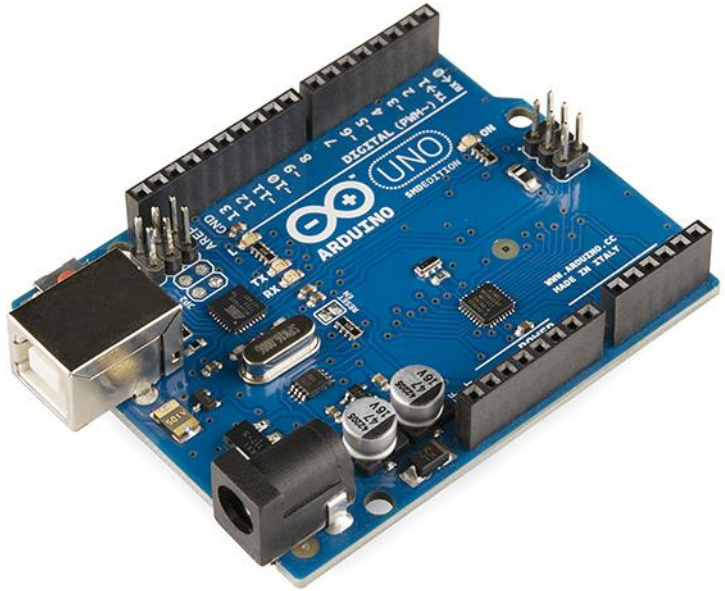
Input from variety of sensors and make

decision and then give the output

- It can be used to develop stand-alone

interactive objects or can be connected

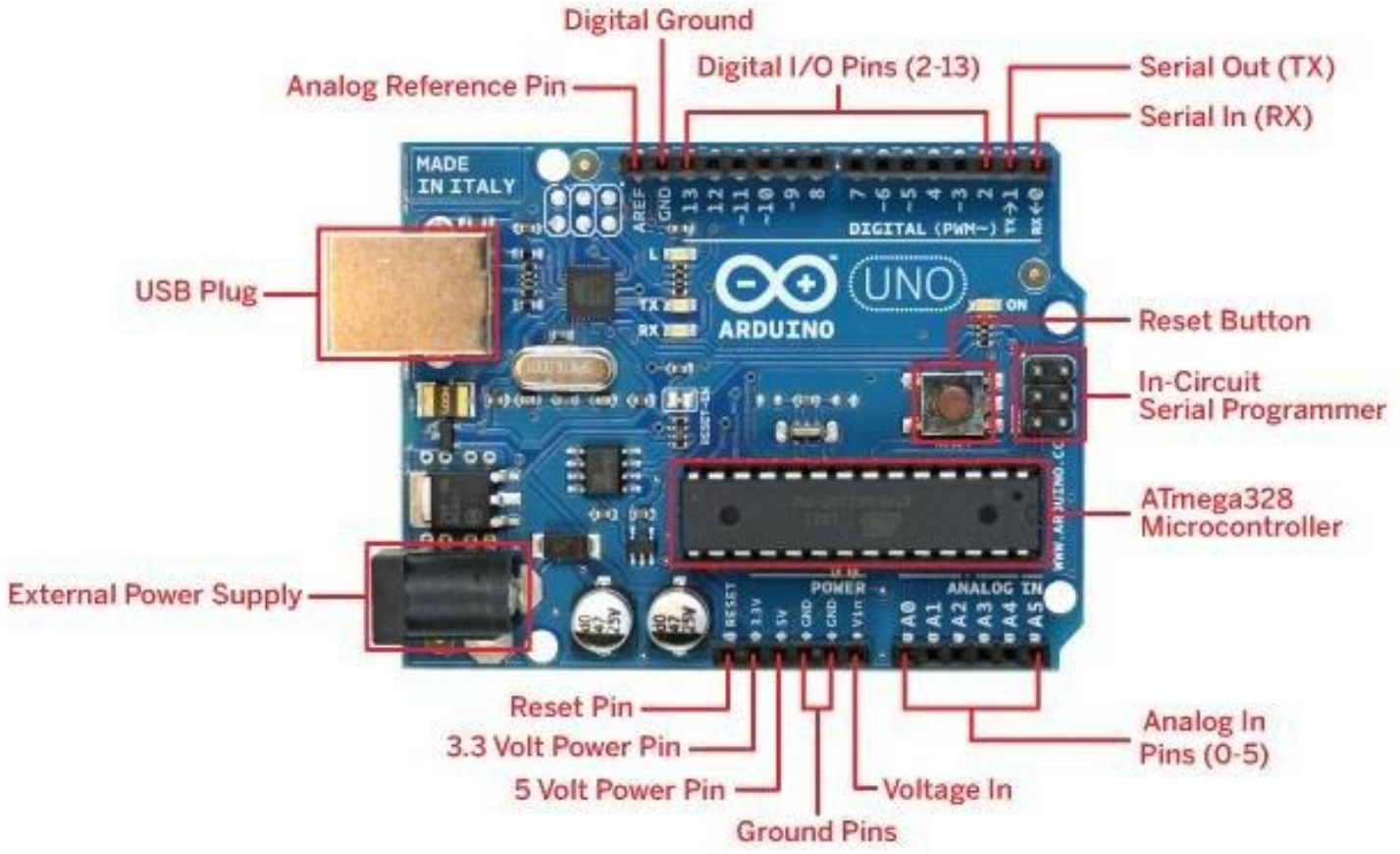
to software on your computer



Arduino

- Working: The microcontroller (computer) is programmed (code) to receive information (input) from the sensors and the output is given through computer (IDE) or other peripherals
- The word Arduino can mean 3 things: Hardware, An interface, A community

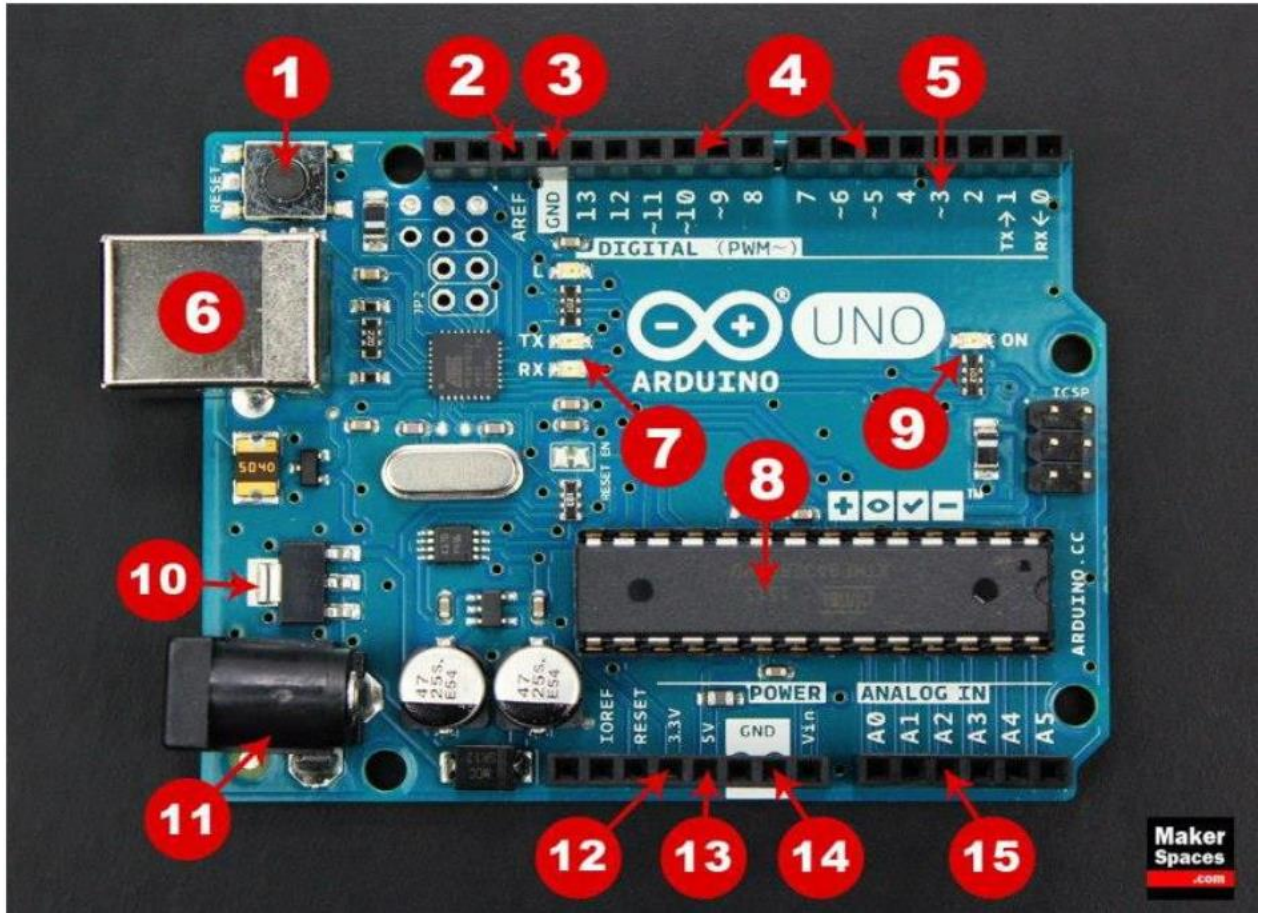
ARDUINO UNO


Digital Ground
Analog Reference Pin
Digital I/O Pins (2-13)
Serial Out (TX)
Serial In (RX)
USB Plug
Reset Button
In-Circuit
Serial Programmer
ATmega328
Microcontroller
External Power Supply
Reset Pin
3.3 Volt Power Pin
5 Volt Power Pin
Ground Pins
Voltage In
Analog In
Pins (0-5)
MADE
IN ITALY
UNO
ARDUINO
DIGITAL (PWM~)
POWER
ANALOG IN

A high-level design flow for a smart lightbulb

- Have a smart lightbulb that is able to detect a change to its state: On/Off
- communicates its ON status to the gateway
- Wi-Fi router
- Transmits the status to the AWS cloud
- Amazon Simple Notification Service (SNS)

Components of Arduino



Components of Arduino

1. Reset Button – This will restart any code that is loaded to the Arduino board
2. AREF – Stands for “Analog Reference” and is used to set an external reference voltage
3. Ground Pin – There are a few ground pins on the Arduino and they all work the same
4. Digital Input/Output – Pins 0-13 can be used for digital input or output
5. PWM – The pins marked with the (~) symbol can simulate analog output

Components of Arduino

6. USB Connection – Used for powering up your Arduino and uploading sketches
7. TX/RX – Transmit and receive data indication LEDs
8. ATmega Microcontroller – This is the brains and is where the programs are stored
9. Power LED Indicator – This LED lights up anytime the board is plugged in a power source
10. Voltage Regulator – This controls the amount of voltage going into the Arduino board

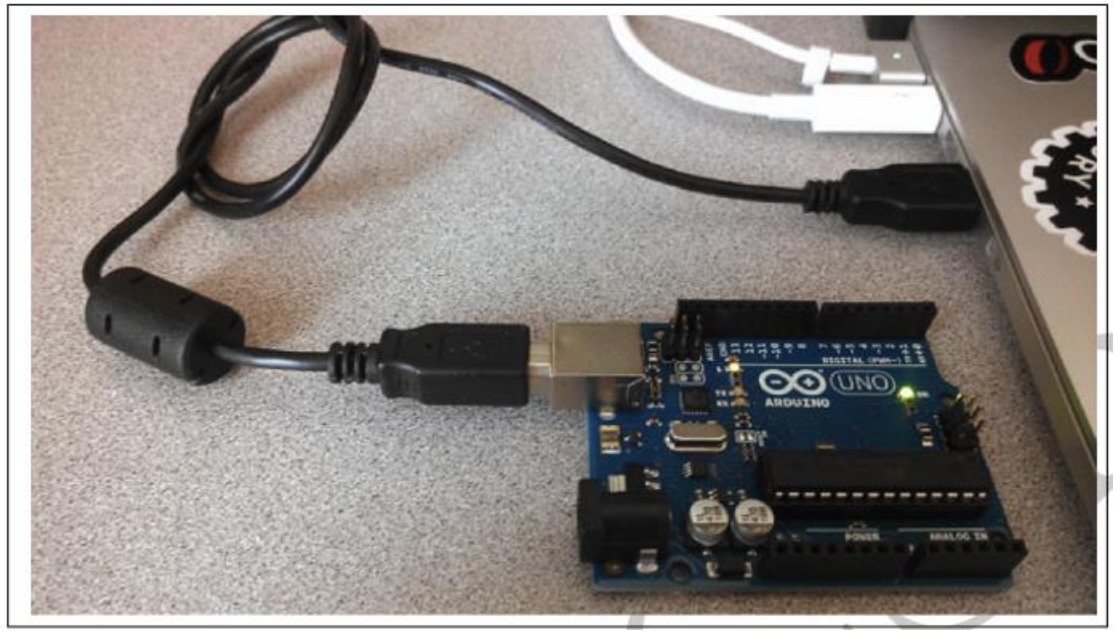
Components of Arduino

11. DC Power Barrel Jack – This is used for powering your Arduino with a power supply
12. 3.3V Pin – This pin supplies 3.3 volts of power to your projects
13. 5V Pin – This pin supplies 5 volts of power to your projects
14. Ground Pins – There are a few ground pins on the Arduino and they all work the same
15. Analog Pins - The Arduino Uno has 6 analog pins, which utilize ADC (Analog to Digital converter). These pins serve as analog inputs but can also function as digital inputs or digital outputs.

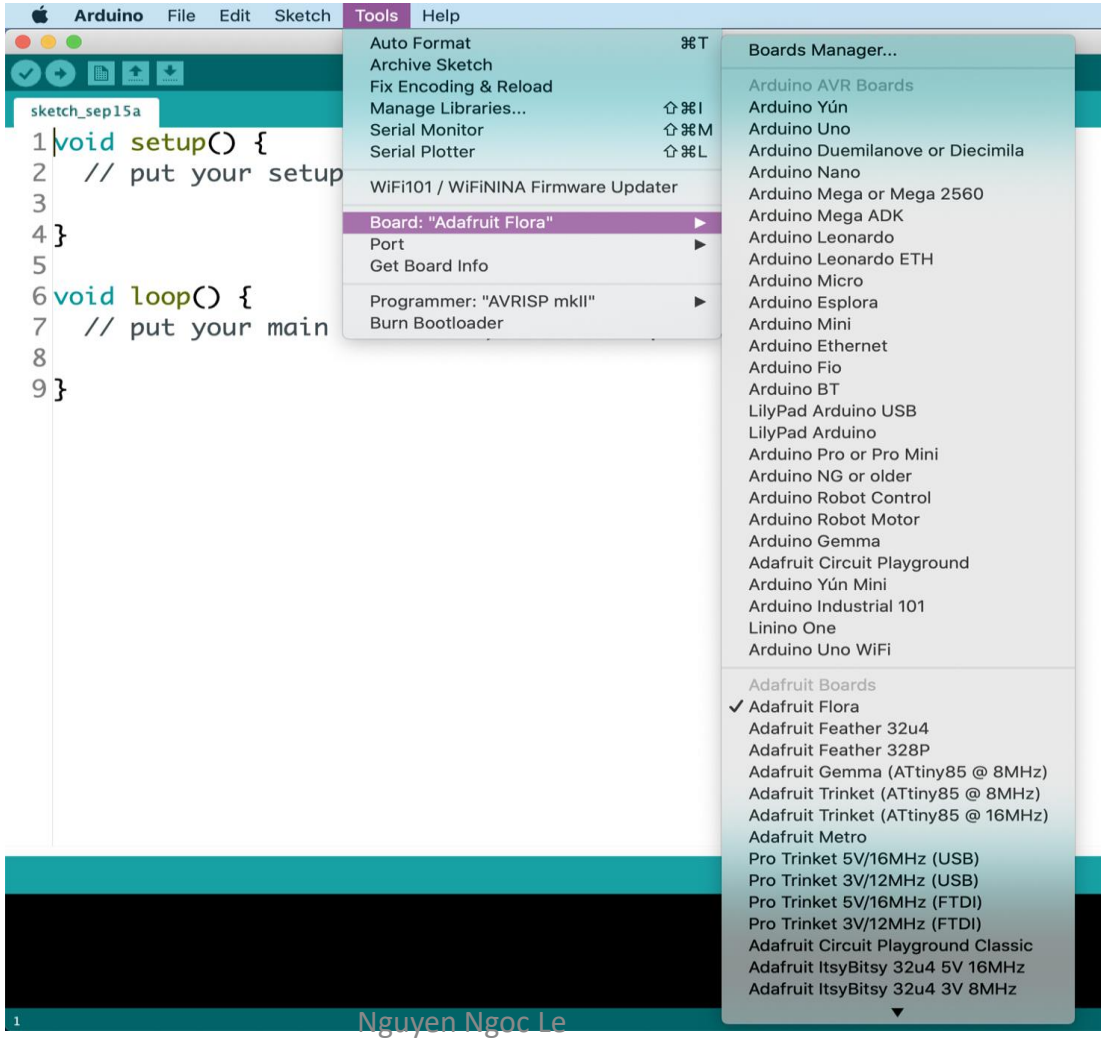
Arduino IDE

- http://arduino.cc/en

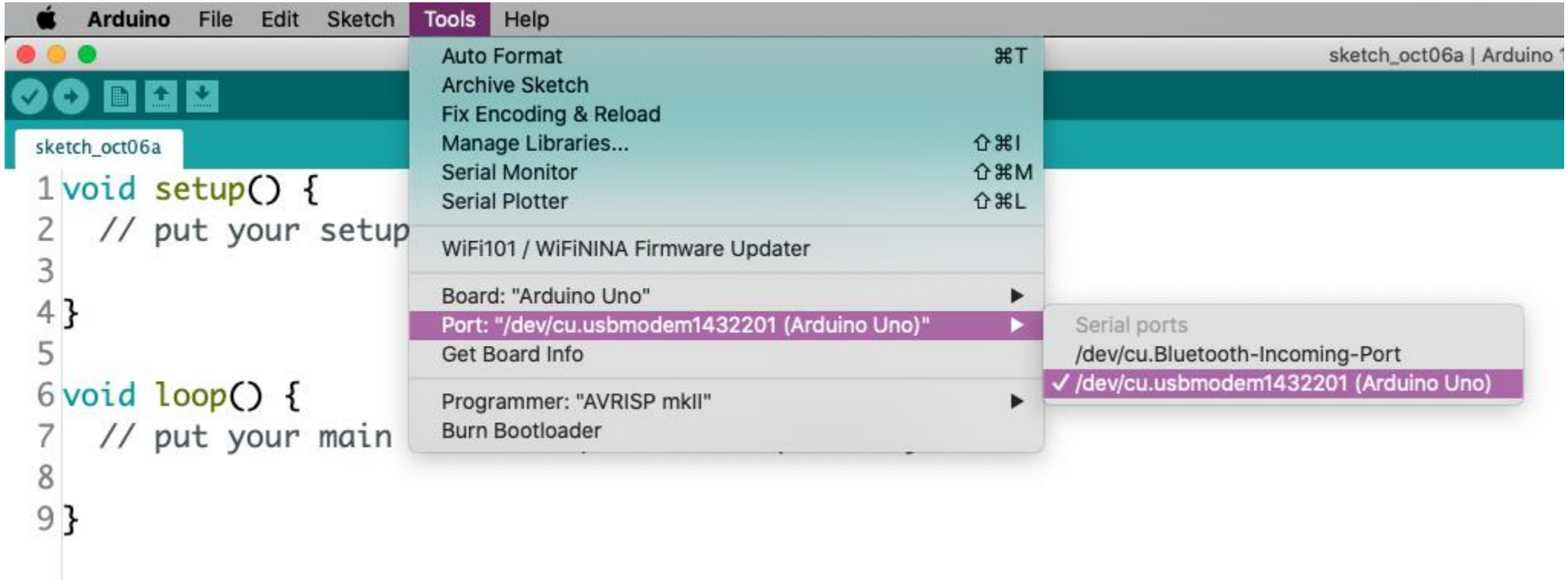
Connect the Arduino to the host computer



Check the board



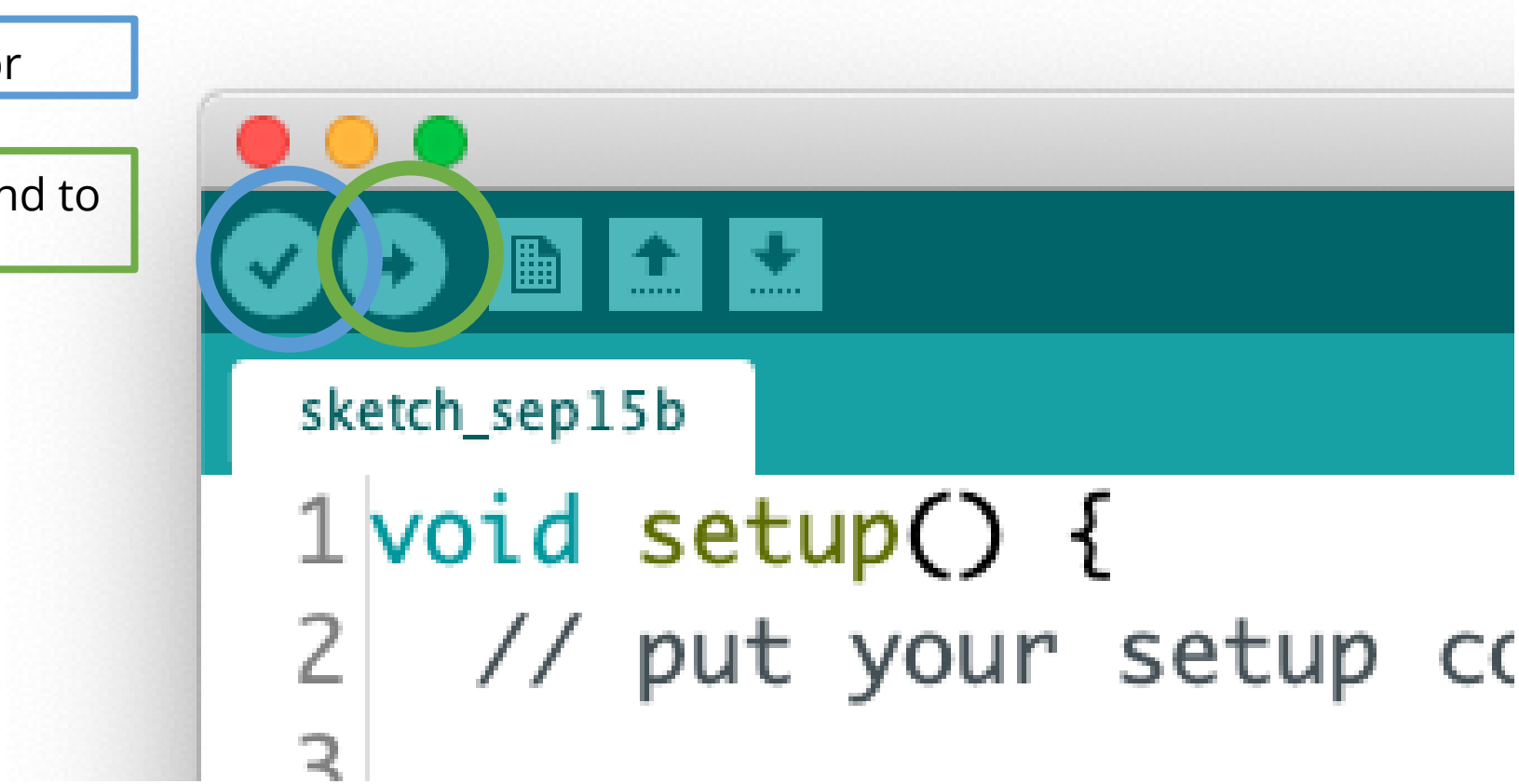
Check port


Arduino File Edit Sketch Tools Help
sketch_oct06a | Arduino
sketch_oct06a
1 void setup() {
2 // put your setup
3
4 }
5
6 void loop() {
7 // put your main
8
9 }
Auto Format ⌘T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... ⇧⌘I
Serial Monitor ⇧⌘M
Serial Plotter ⇧⌘L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Uno"
Port: "/dev/cu.usbmodem1432201 (Arduino Uno)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
/dev/cu.Bluetooth-Incoming-Port
✓ /dev/cu.usbmodem1432201 (Arduino Uno)

Upload Code

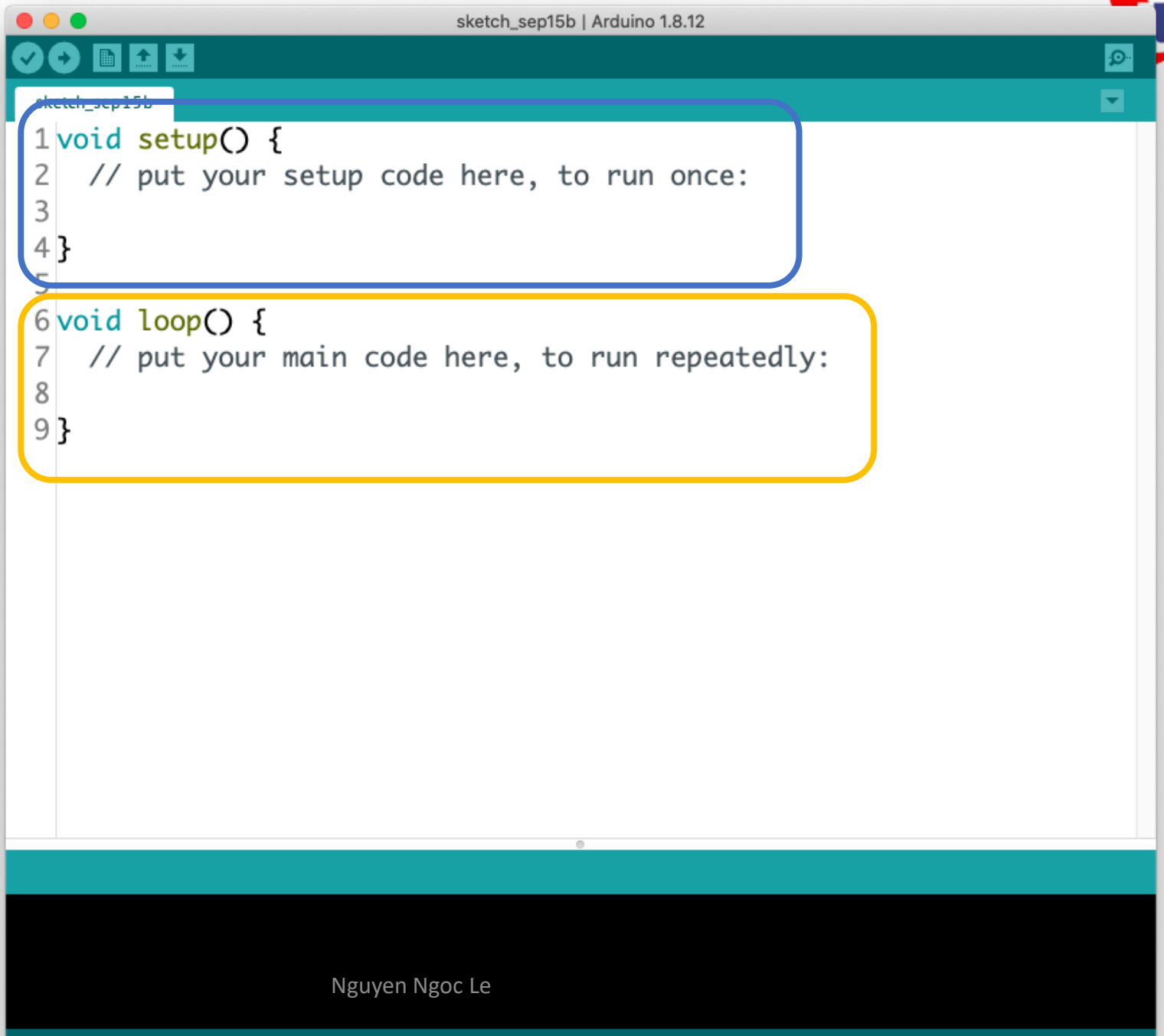
Check your code for errors

Check code and send to the Arduino

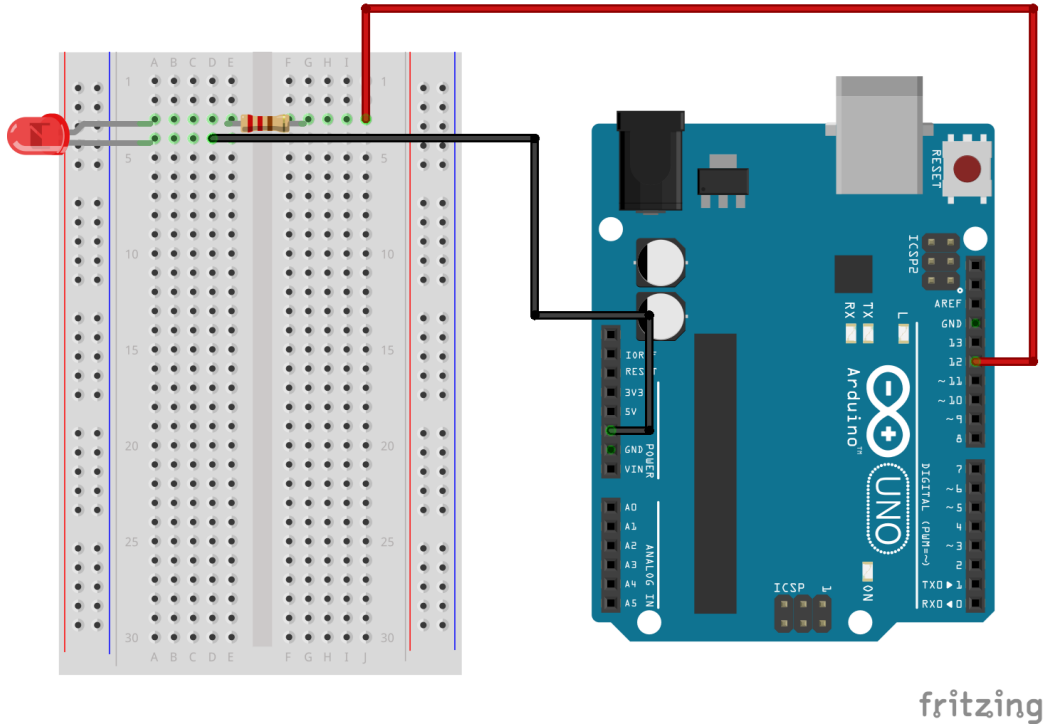


This code is run on initial start up of the Arduino. It's like the pre-flight checklist. It will run once.

This code will be run over and over again, until the Arduino loses power, or you send it new code.



Circuit Assembly

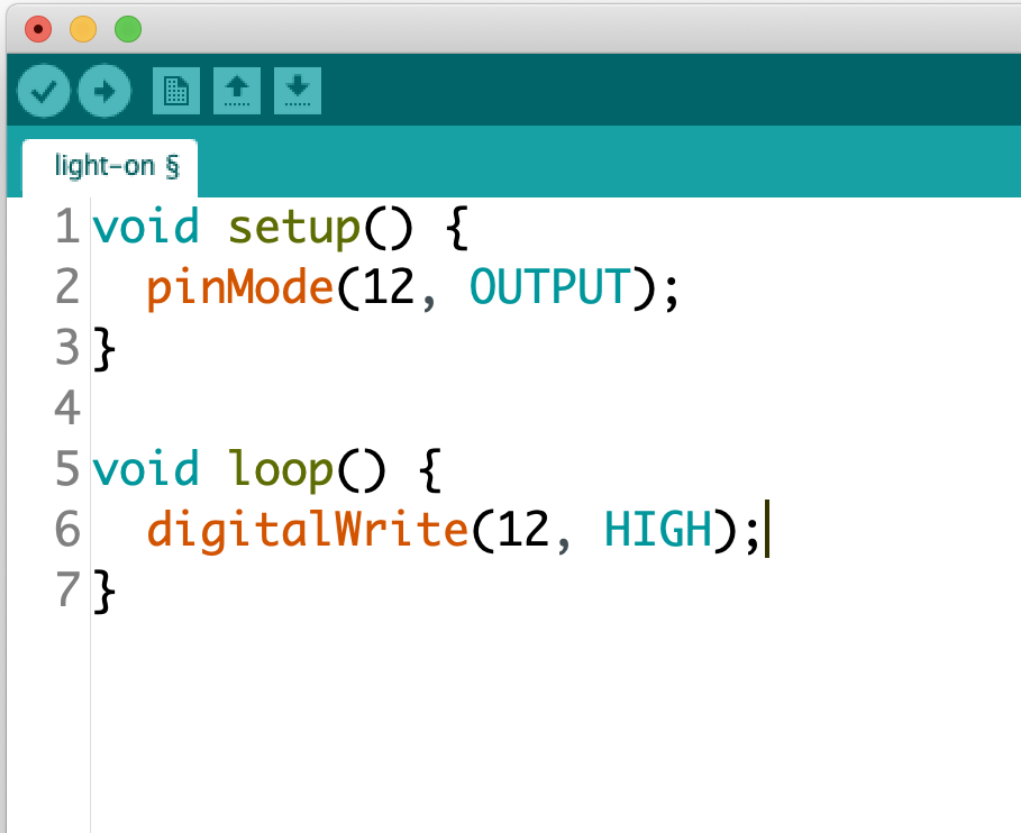


Light on!

```
// <pin #> = a number on the Arduino
// <type> = OUTPUT or INPUT
// format = pinMode(<pin #>, <type>);
pinMode(12, OUTPUT);
```

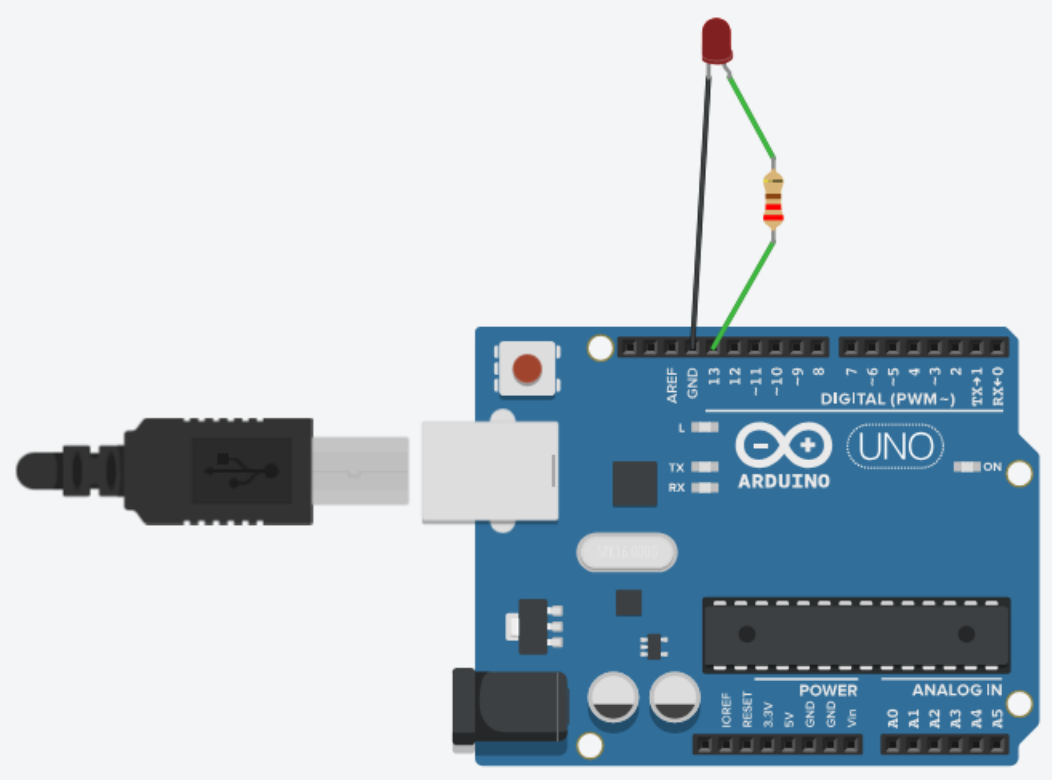
```
// <pin #> = a pin/hole on the Arduino board
// <type> = HIGH or LOW
// digitalWrite(<pin #>, <type>);
digitalWrite(12, HIGH);
```

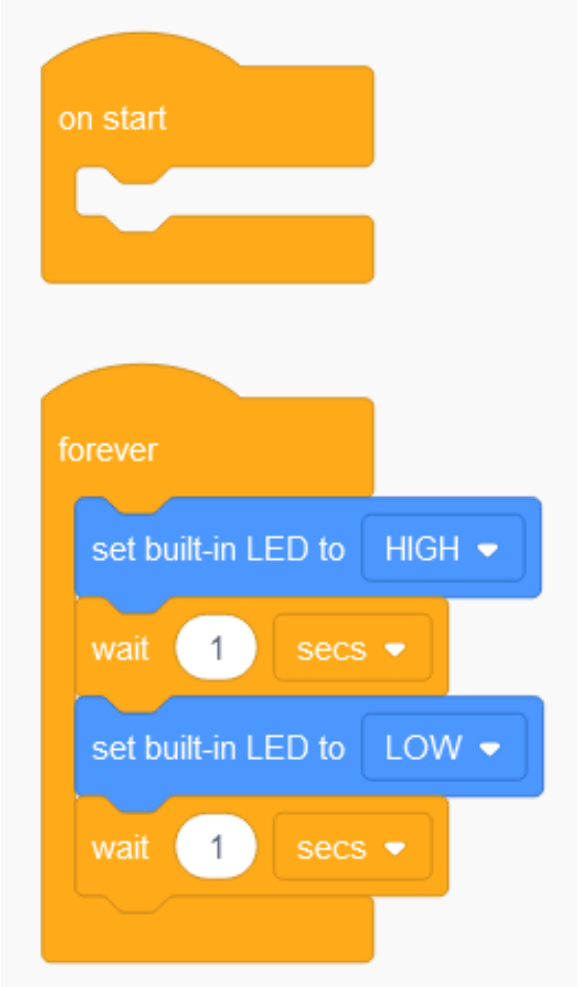
Light on!

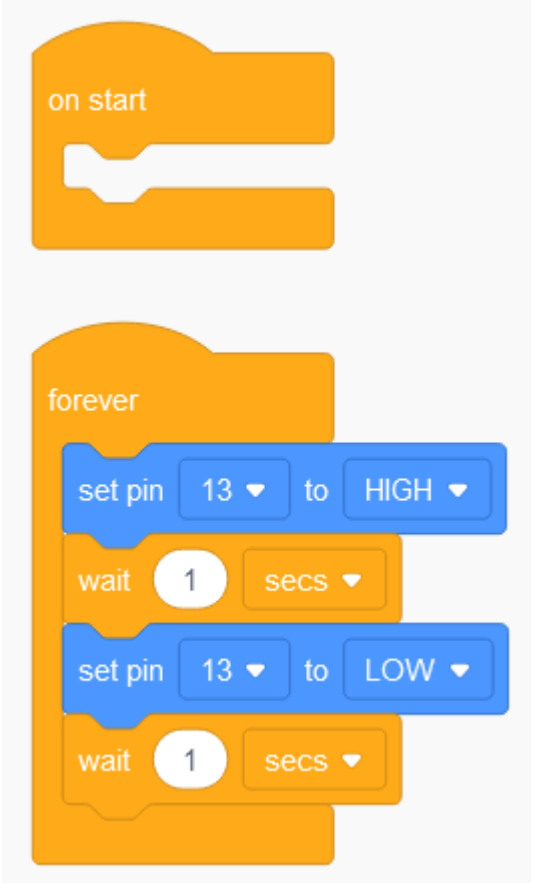



```
// Pause the script for # of milliseconds
// 1000 milliseconds = 1 second
// delay(#);
delay(1000);
```

Practice

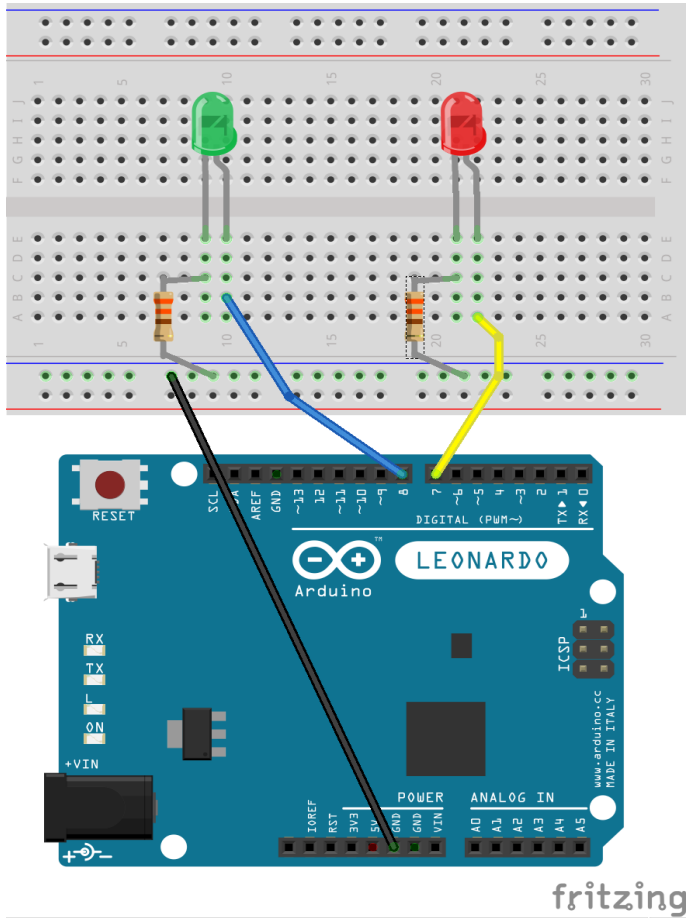

DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN


on start
forever
set built-in LED to HIGH
wait 1 secs
set built-in LED to LOW
wait 1 secs

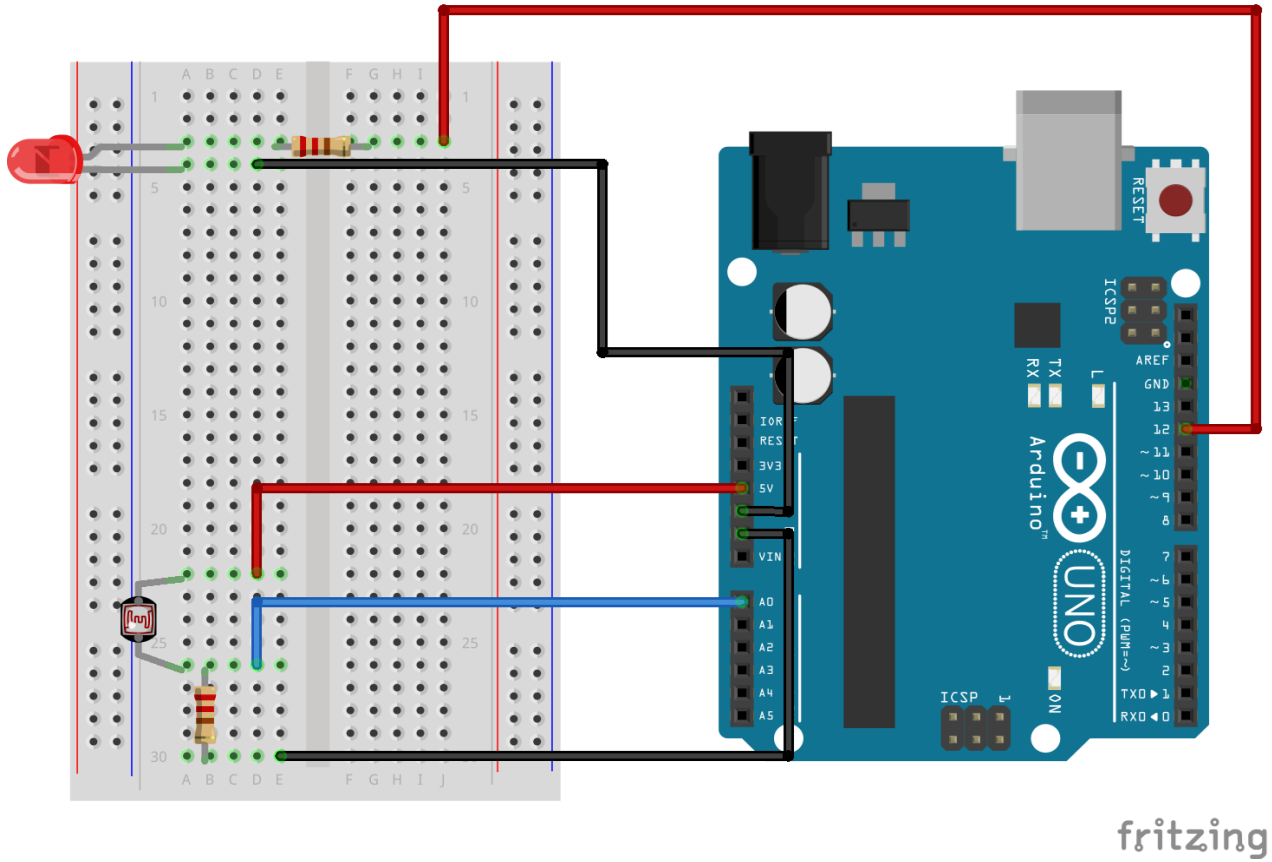

on start
forever
set pin 13 to HIGH
wait 1 secs
set pin 13 to LOW
wait 1 secs

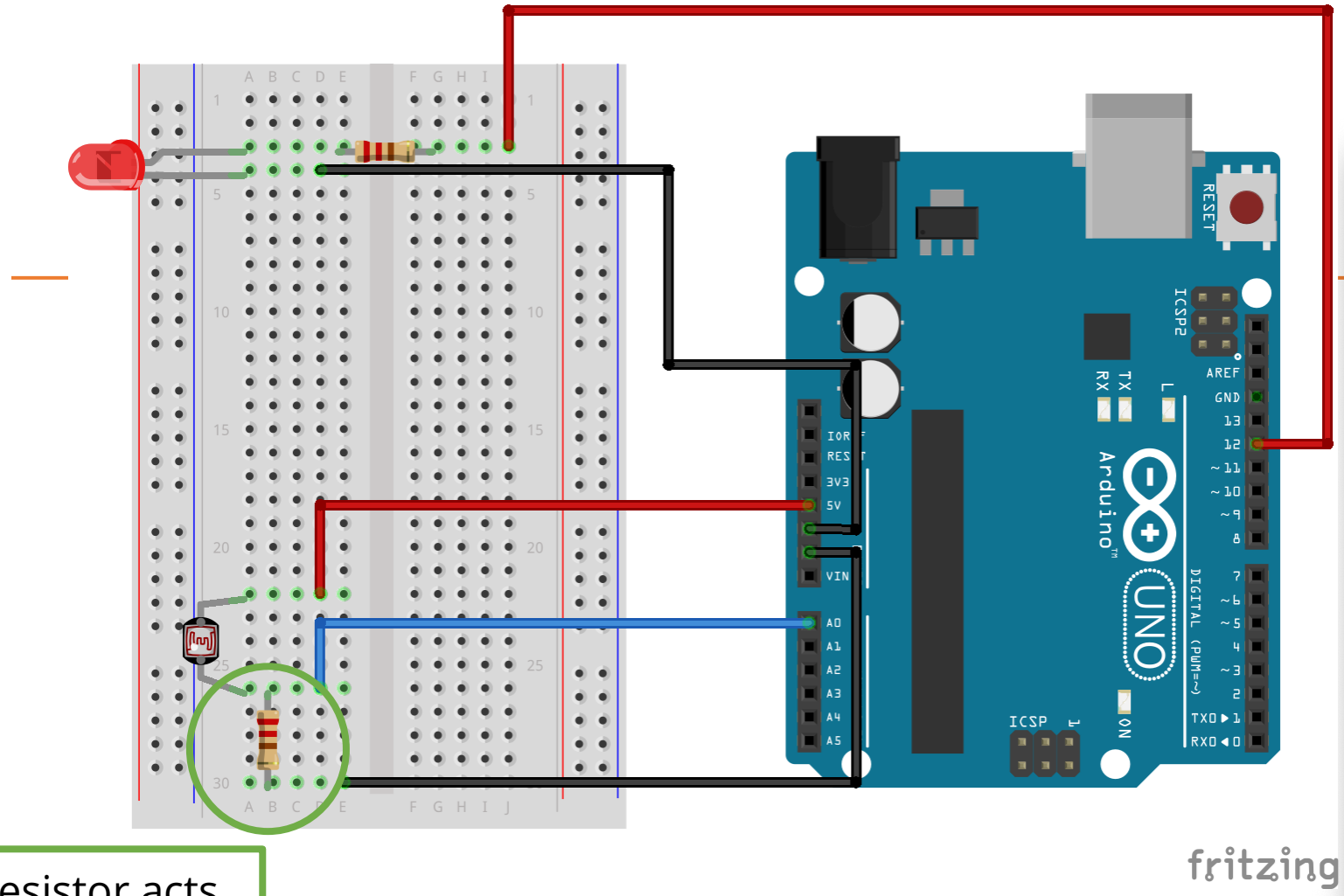
Practice

- Write code for circuit



Practice


RESET
ICSP2
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TXD 1
RXD 0
DIGITAL (PWM=~)
TX
RX
L
ON
ICSP
Arduino
UNO
IOREF
RESET
3V3
5V
VIN
A0
A1
A2
A3
A4
A5
fritzing




```
nightlight | Arduino 1.8.12

nightlight §

1  // Set up three variable to use later.
2
3  int threshold = 100;
4  int resistance;
5  int initialPhotoValue;
6
7  void setup() {
8    pinMode(12, OUTPUT);
9    initialPhotoValue = analogRead(0);
10 }
11
12 void loop() {
13   // Read the data from analog pin 0, and
14   // store in in the variable named 'resistance'.
15   resistance = analogRead(0);
16
17   // logic test, is the new value lower than the
18   // initial value minus the threshold?
19   if (resistance < initialPhotoValue - threshold) {
20     digitalWrite(12, HIGH);
21   } else {
22     digitalWrite(12, LOW);
23   }
24
25 }

Done uploading.

Sketch uses 1976 bytes (6%) of program storage space. Maximum is 32256
Global variables use 192 bytes (9%) of dynamic memory, leaving 1856 by

18                                   Arduino Uno on /dev/cu.usbmodem1432201
```

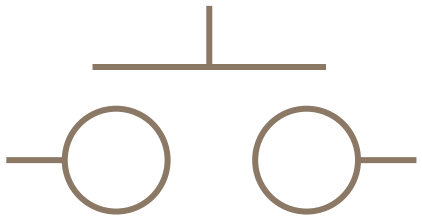
This resistor acts as a pull down resistor.

Exercise: Push Button Switch

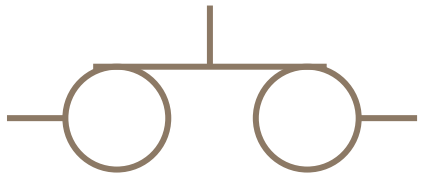
Stops the flow of current when they’re open.

Current flows freely when closed.

Normally Open
(NO)



Open Switch
(At Rest)



Closed Switch
(Depressed)

Exercise: Push Button Switch