

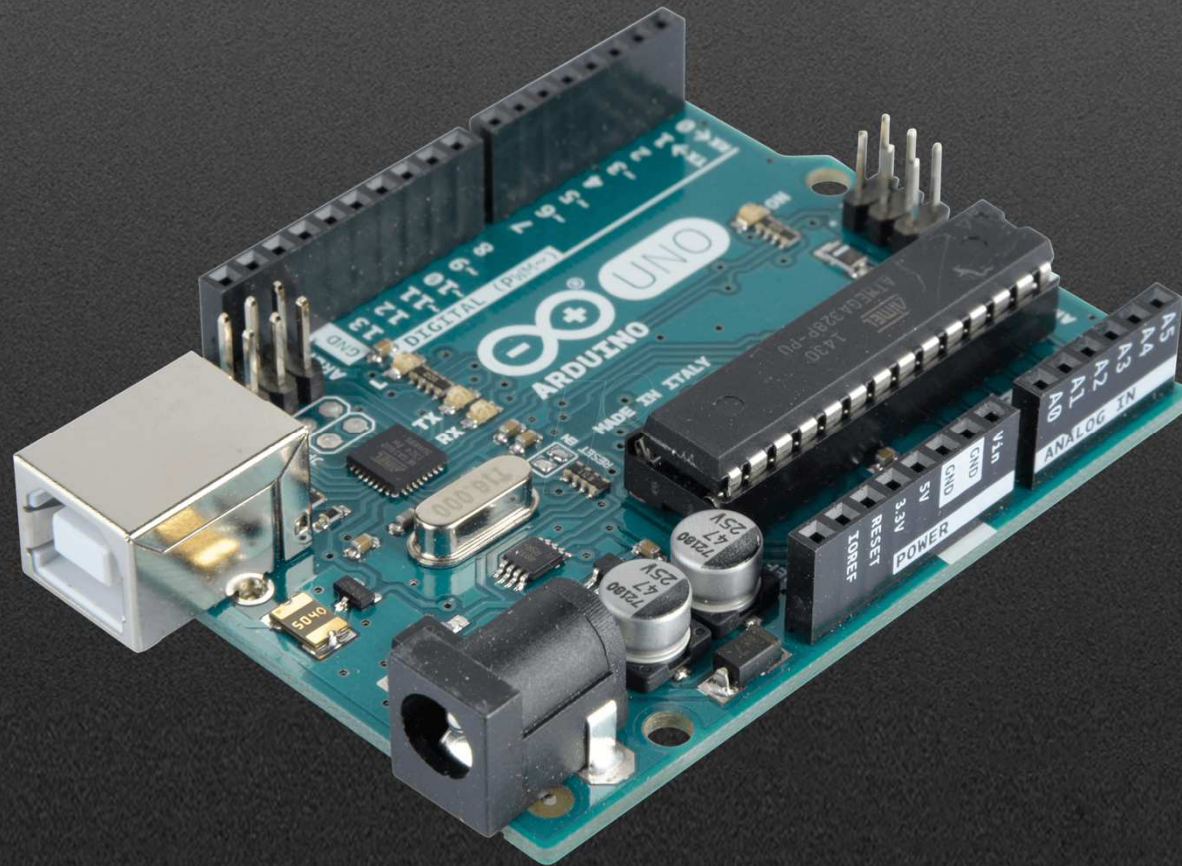
Sustainable Mobility



Autonomous cars on budget
Duarte Abreu

ARDUINO UNO

Arduino UNO



Arduino IDE

- www.arduino.cc



Download the Arduino IDE



ARDUINO 1.8.1

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

Windows app 

Mac OS X 10.7 Lion or newer

Linux 32 bits

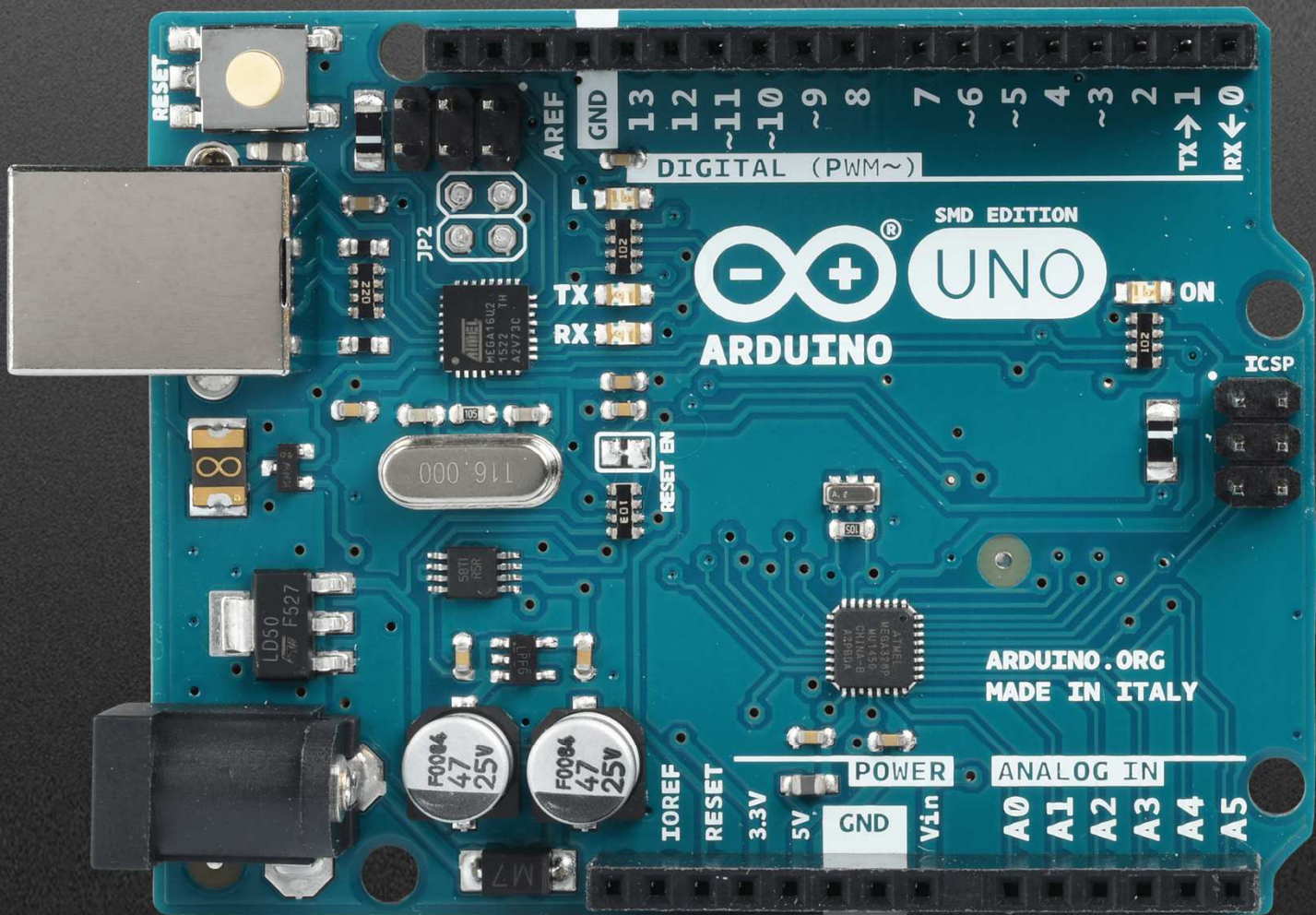
Linux 64 bits

Linux ARM

[Release Notes](#)

[Source Code](#)

[Checksums \(sha512\)](#)



ARDUINO UNO SMD EDITION

ARDUINO.ORG
MADE IN ITALY

DIGITAL (PWM~)
13 GND
12
11 ~
10 ~
9
8
7
6 ~
5 ~
4
3
2
1 TX →
0 RX ←

POWER ANALOG IN
IOREF RESET 3.3V 5V GND Vin
A0 A1 A2 A3 A4 A5

ATMEL
MEGA16U2
1522
AVV73C

ATmega328P
B-4NHQ
89A10M
94C83R5N
15N11

116.000

F0084
47
25V

F0084
47
25V

LD50
F527

RESET

RESET EN

ICSP

TX
RX

ON

AREF
GND

JP2

IOREF

RESET

3.3V

5V

GND

Vin

A0

A1

A2

A3

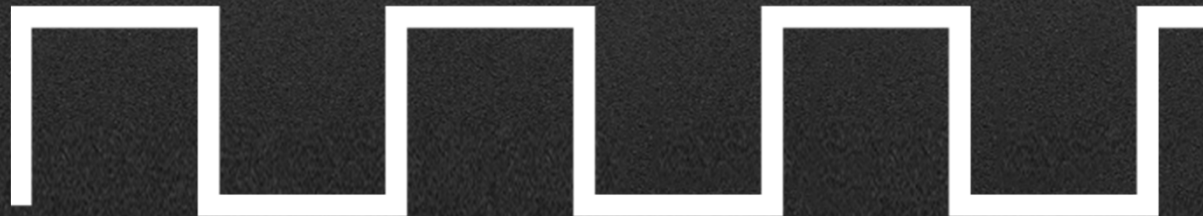
A4

A5

Digital VS Analog

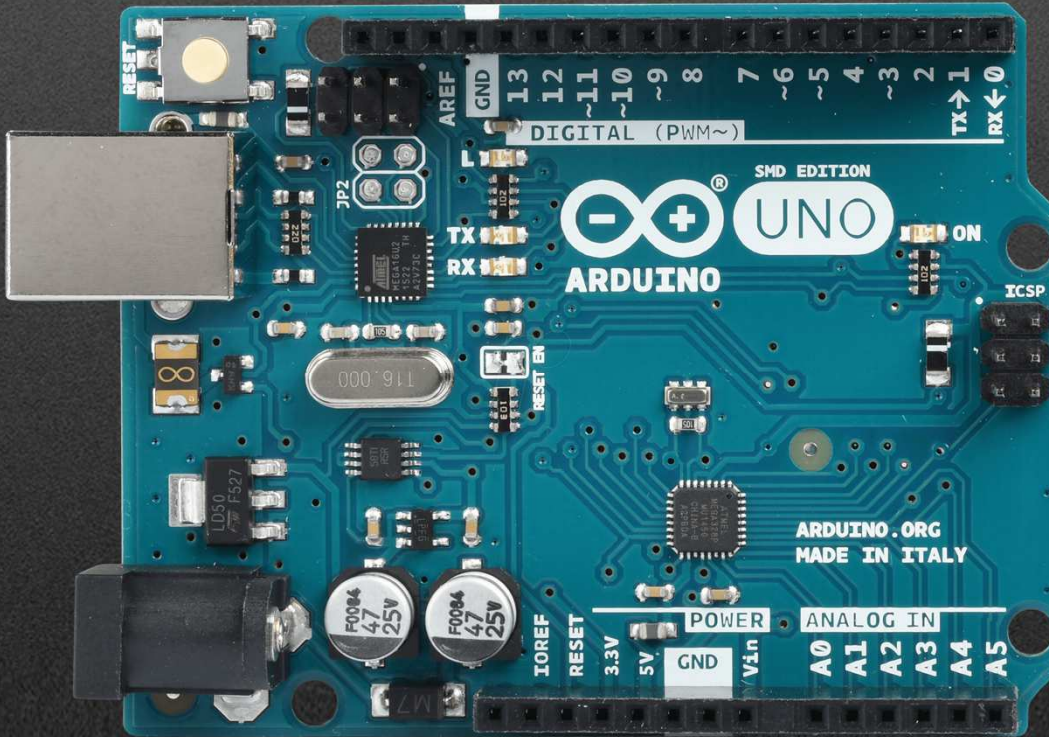


Analog Signal



Digital Signal

Arduino UNO



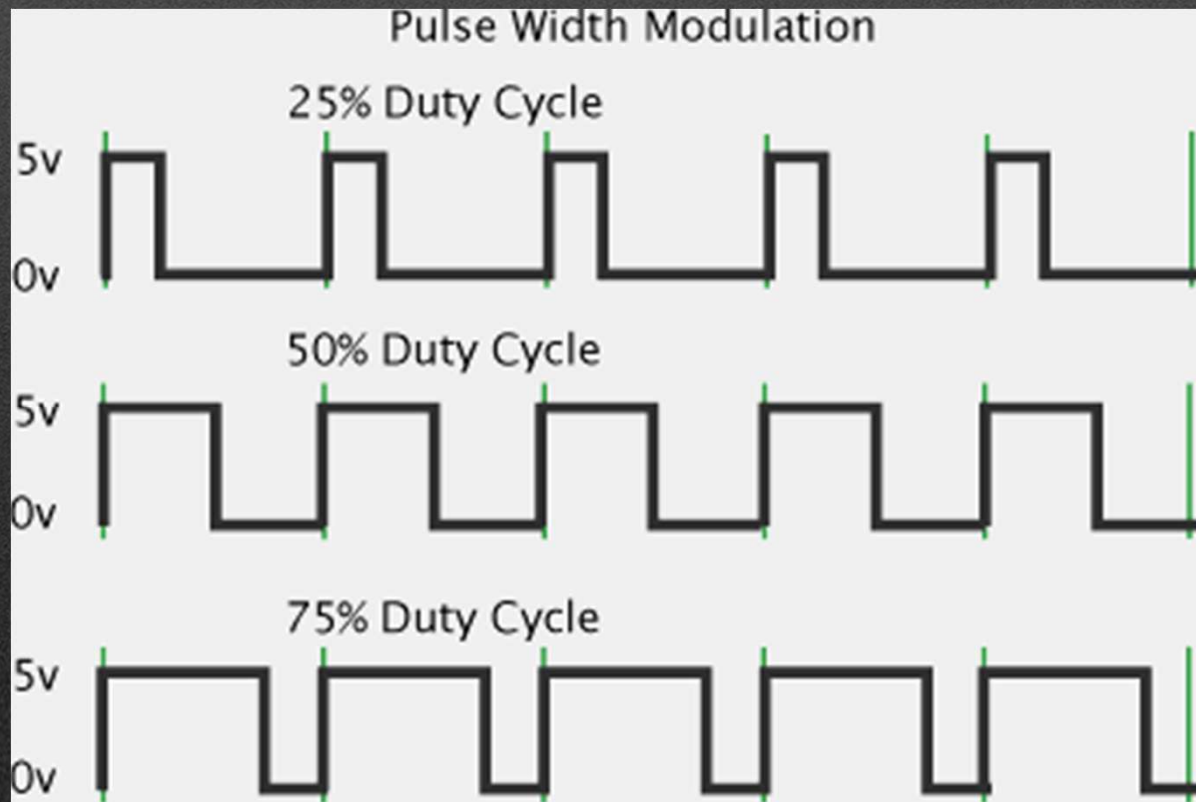
DIGITAL



ANALOG



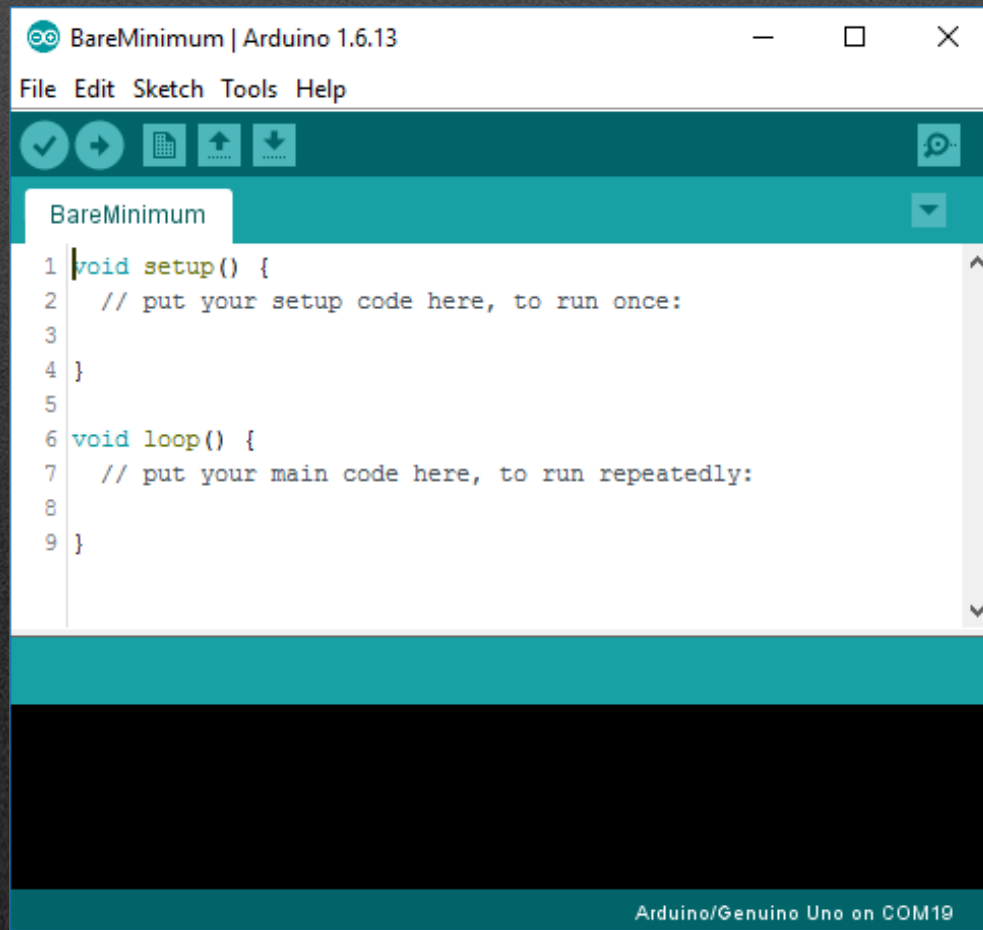
PWM (Pulse Width Modulation)



It works using moving averages to represent “analog” values.

By default it has 8 bits of resolution (256 different values). It can however be altered with the command `analogWriteResolution()` to 12 bits (65536 values).

Main window



EDIT

Undo	Ctrl+Z
Redo	Ctrl+Y
<hr/>	
Cut	Ctrl+X
Copy	Ctrl+C
Copy for Forum	Ctrl+Shift+C
Copy as HTML	Ctrl+Alt+C
Paste	Ctrl+V
Select All	Ctrl+A
Go to line...	Ctrl+L
<hr/>	
Comment/Uncomment	Ctrl+Slash
Increase Indent	Tab
Decrease Indent	Shift+Tab
<hr/>	
Find...	Ctrl+F
Find Next	Ctrl+G
Find Previous	Ctrl+Shift+G

SKETCH

Verify/Compile	Ctrl+R
Upload	Ctrl+U
Upload Using Programmer	Ctrl+Shift+U
Export compiled Binary	Ctrl+Alt+S
<hr/>	
Show Sketch Folder	Ctrl+K
Include Library	>
Add File...	

FILE

New	Ctrl+N
Open...	Ctrl+O
Open Recent	>
Sketchbook	>
Examples	>
Close	Ctrl+W
Save	Ctrl+S
Save As...	Ctrl+Shift+S
<hr/>	
Page Setup	Ctrl+Shift+P
Print	Ctrl+P
<hr/>	
Preferences	Ctrl+Comma
<hr/>	
Quit	Ctrl+Q

TOOLS

Auto Format	Ctrl+T
Archive Sketch	
Fix Encoding & Reload	
Serial Monitor	Ctrl+Shift+M
Serial Plotter	Ctrl+Shift+L
<hr/>	
WiFi101 Firmware Updater	
<hr/>	
Board: "Arduino/Genuino Uno"	>
Port	>
Get Board Info	
<hr/>	
Programmer: "AVRISP mkII"	>
Burn Bootloader	

Bare Minimum

```
sketch_dec01a | Arduino 1.6.13
File Edit Sketch Tools Help
sketch_dec01a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Setup code.

Here you should indicate what you will connect to the microcontroller, declare variables and initiate SPI communication.

Loop code.

Here you should put the loop you want to iterate. Like the name implies, all the code in here will be executed fully several times, until it either runs out of power or until the reset button is pressed.

Blink

```
Blink | Arduino 1.6.13
File Edit Sketch Tools Help

Blink $

void setup() {
  int LED = 13;
  pinMode(LED, OUTPUT);
}

void loop() {
  digitalWrite(LED, HIGH);
  delay(1000);
  digitalWrite(LED, LOW);
  delay(1000);
}
```

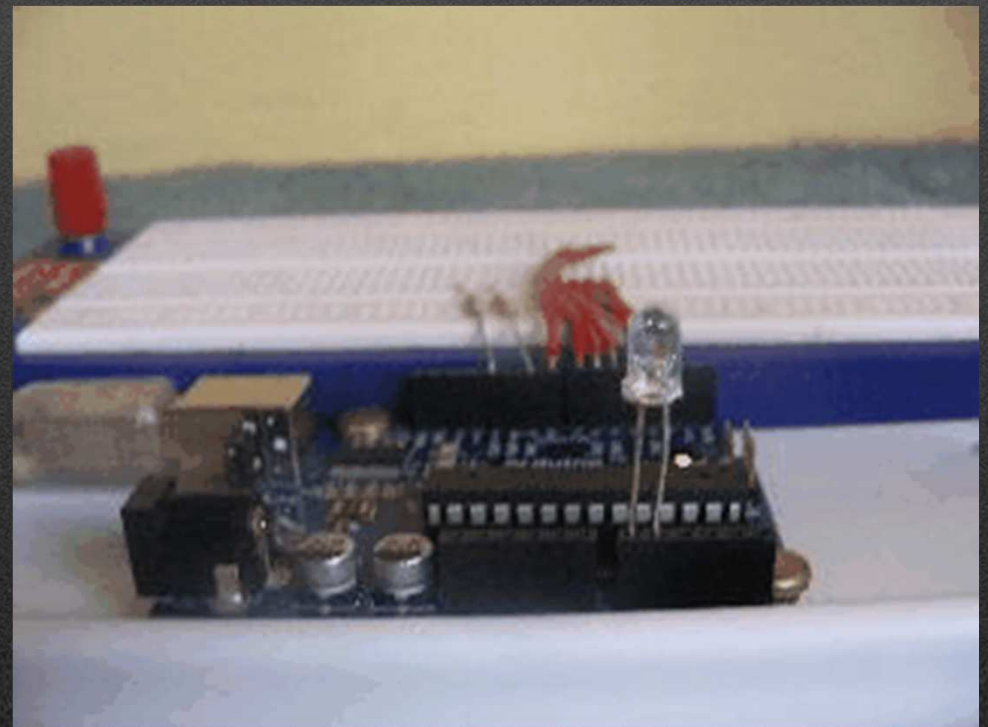
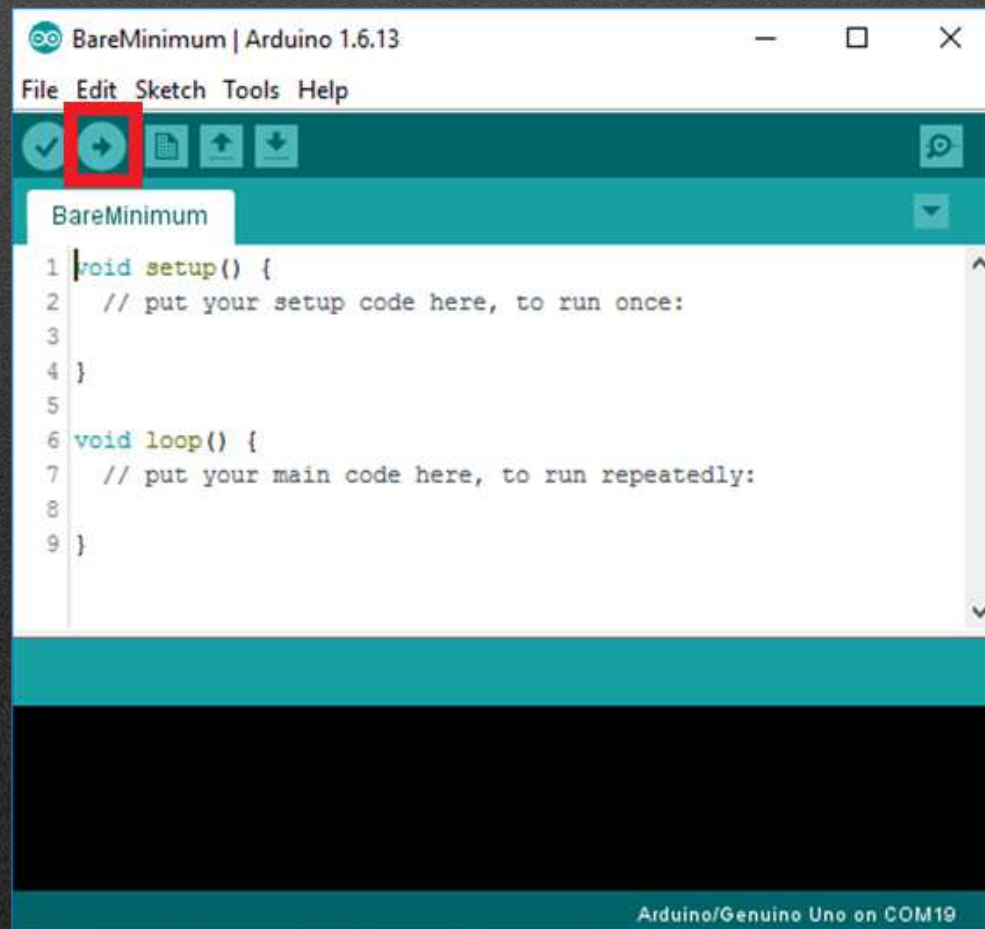
We'll use the integrated led, on pin #13 in the arduino board. We'll call it LED for convenience and declare it an OUTPUT.

This means the LED will turn on for 0 LED liga durante um 1000 milliseconds and turn off for 1000 milliseconds.

Important Commands

- `pinMode(#pin,OUTPUT);` - declares the selected pin as an output.
- `pinMode(#pin,INPUT);` - same, but as input.
- `Delay(milissegundos);` - defines milliseconds between the previous command and the following one.
- `// THIS IS A COMMENT` – This part is not read by the compiler, informs the coder of useful information usually.
- The **COLORS** are important!!! – In Arduino's IDE, colors indicate if we're dealing with functions, libraries, variables, etc. They also help the coder to identify quicker if they are declaring their code correctly.

Uploading Code

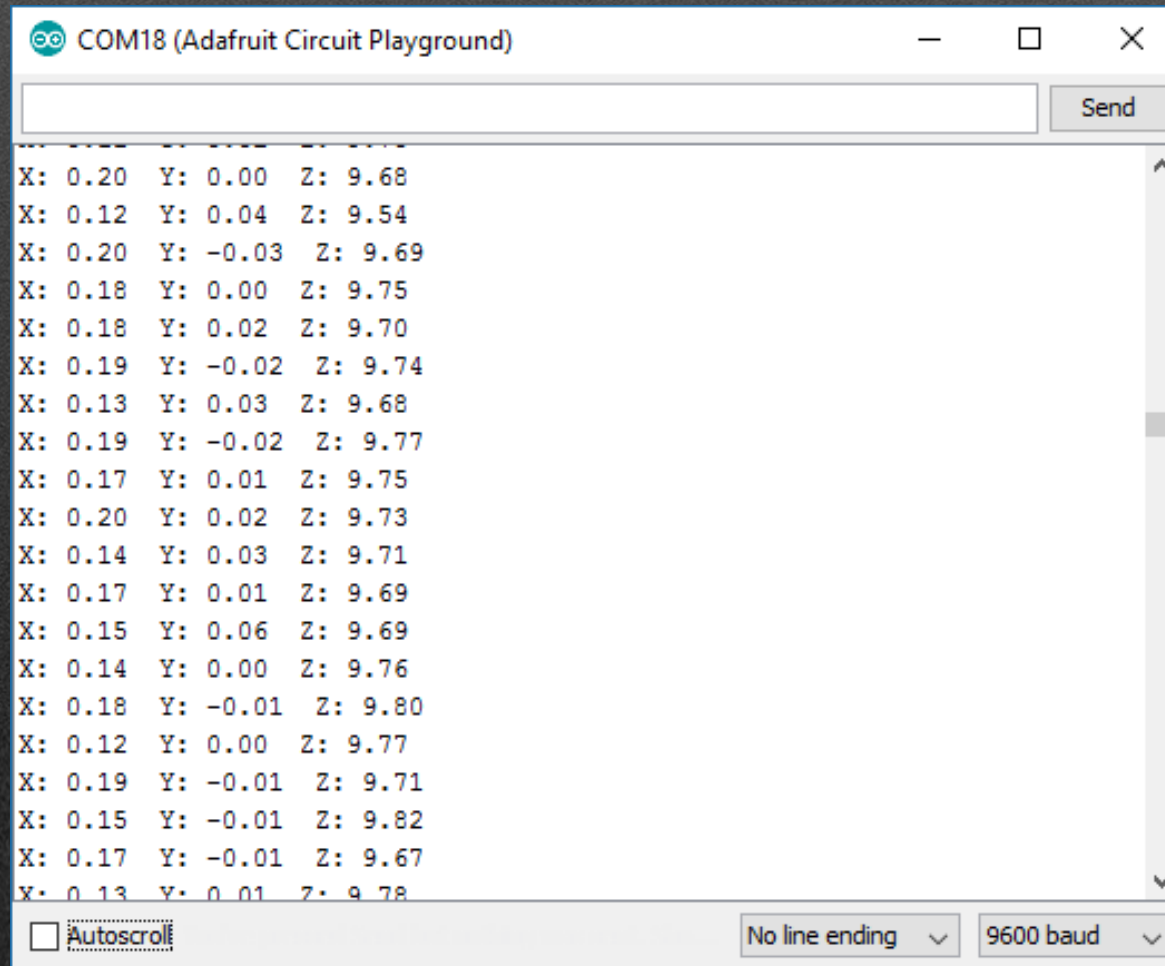


Serial Communication

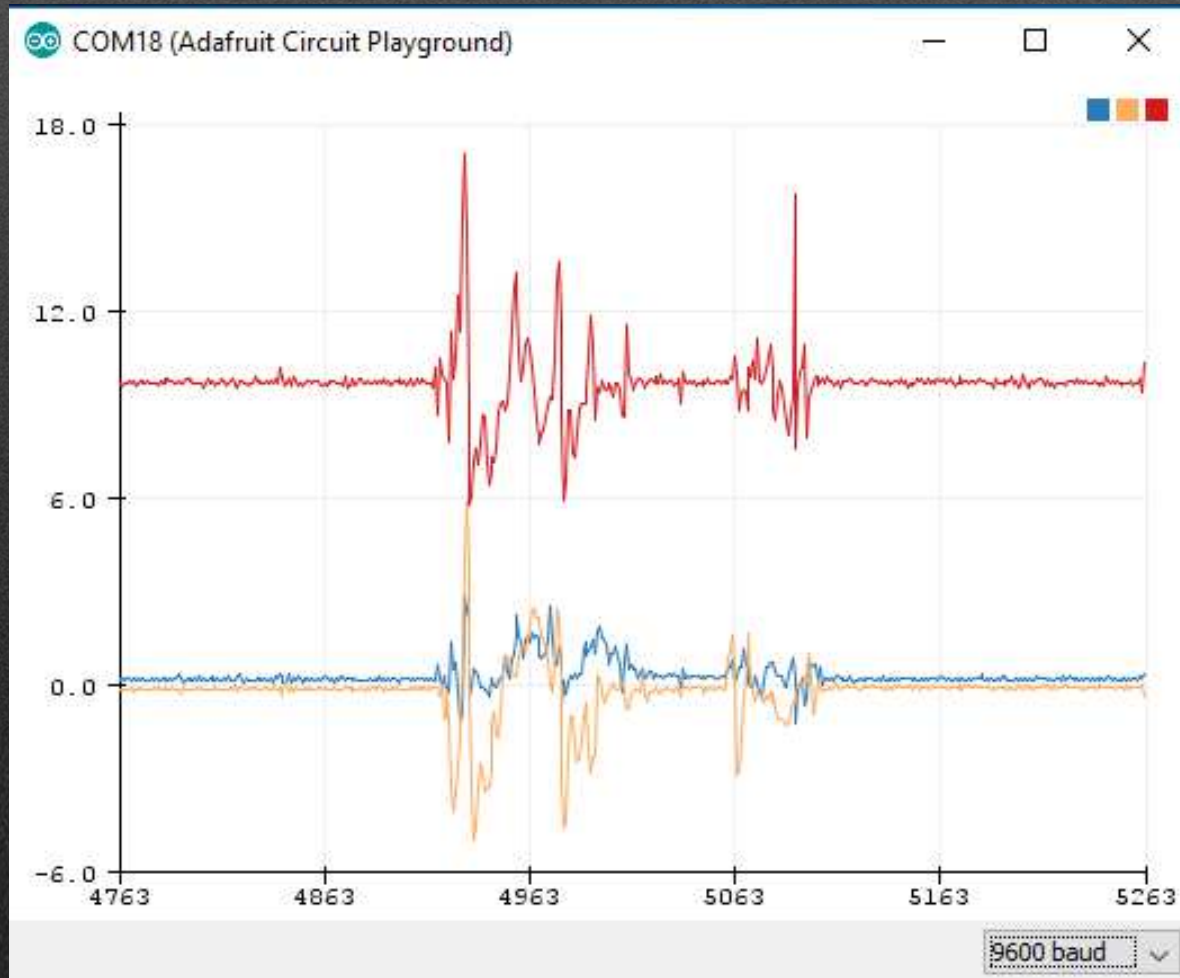
- It's the communication between the computer and the arduino board.
- Also serves to communicate between several arduinos.

- `Serial.begin(baudRate);`
- `Serial.print("text",var);`
- `Serial.println("text",var);`
- `Serial.write();`
- `Serial.read();`

Serial Console



Serial Plotter



Fade

```
Fade | Arduino 1.6.13
File Edit Sketch Tools Help

Fade §

int led = 9;
int brightness = 0;
int fadeAmount = 5;

void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  analogWrite(led, brightness);
  brightness = brightness + fadeAmount;
  if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
  }
  delay(30);
}
```

Declares pin #9 as the LED pin.
Initiates the variables needed.

Configures pin #9 as OUTPUT. To it we are gonna write “analog” values utilizing PWM.

In each iteration, we increase the value of the “brightness” variable, from 0 to 255. When we reach the max value of 255, it starts decreasing in the same rate as before. In this case, we will use the “`analogWrite(led,brightness);`” command so we can see the values changing in our screen.

Fade



Libraries

- Libraries are sets of function not yet included in arduino.
- Must be called in the beginning of the code.
- .h type files.
- example:

```
include<DHT.h>
```

Libraries instalation

Personal Open source Business Explore Pricing Blog Support This repository Search Sign in Sign up

adafruit / **DHT-sensor-library** Watch 134 Star 726 Fork 637

Code Issues 9 Pull requests 7 Projects 0 Wiki Pulse Graphs

Arduino library for DHT11DHT22, etc Temp & Humidity Sensors <http://www.ladvada.net/learn/sensors/...>

54 commits 10 contributors

Branch: master New pull request

Find file Clone or download

microbuilder Merged unified and raw libraries 8 months ago

.github Add GitHub issue template 8 months ago

examples Merged unified and raw libraries 3 months ago

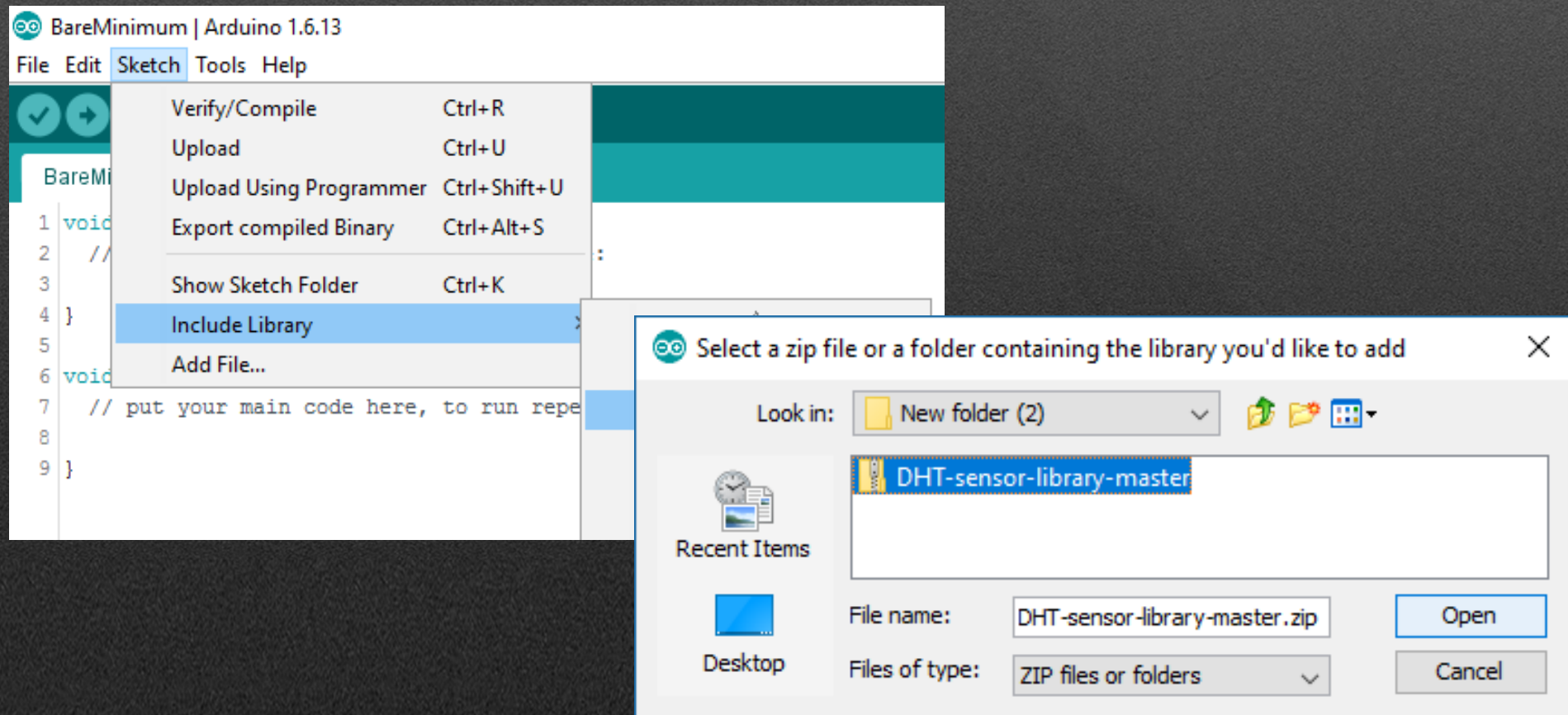
DHT.cpp Fix #44 by conditionally excludng unused port and bitmask state on n... a year ago

DHT.h Fix #44 by conditionally excludng unused port and bitmask state on n... a year ago

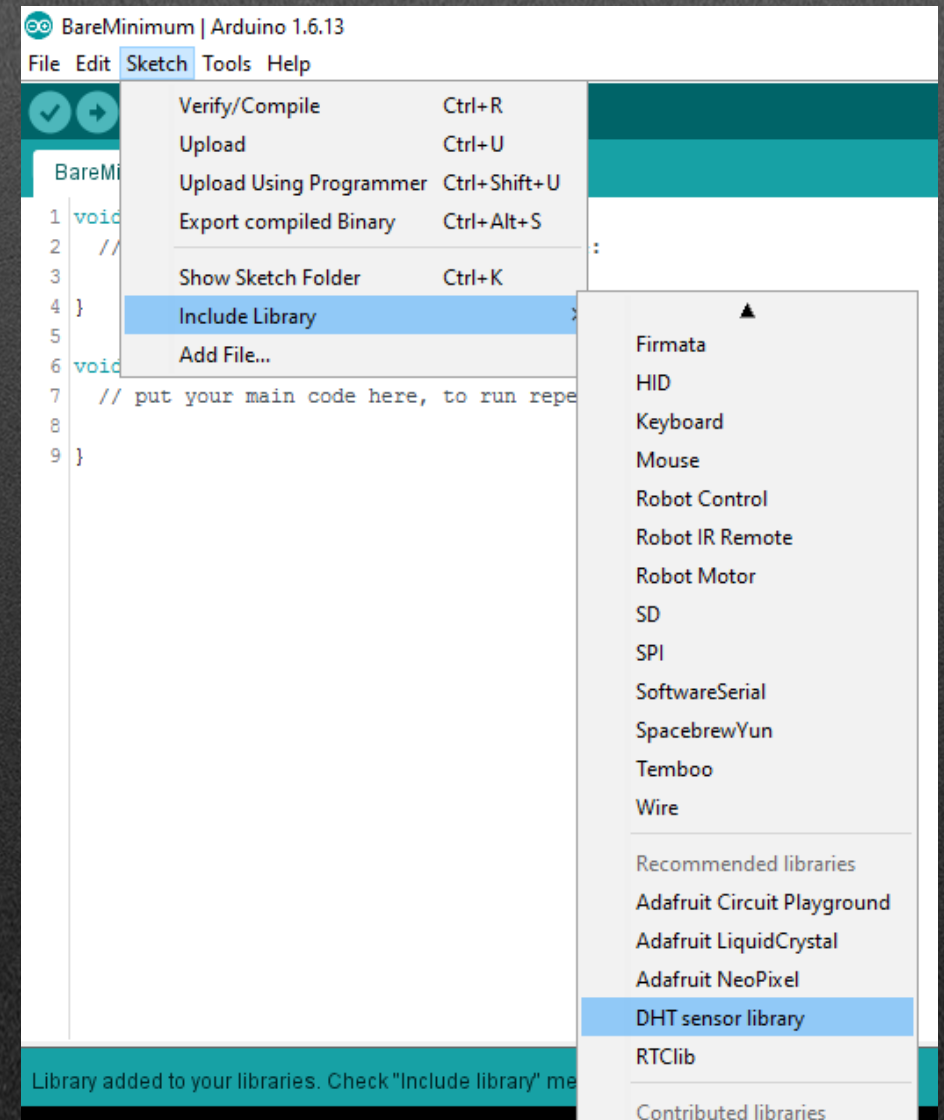
DHT_U.cpp Merged unified and raw libraries 3 months ago

Show All Downloads

Libraries installation



Libraries instalation

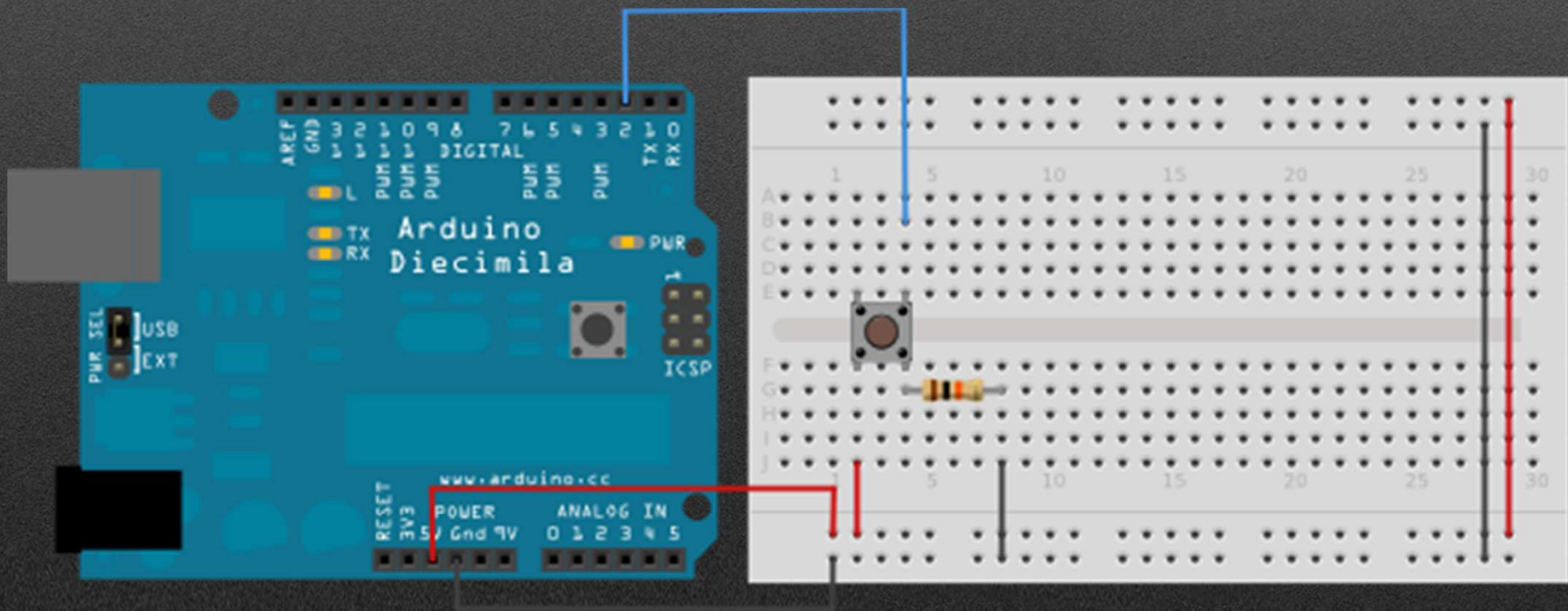


Calling libraries in code

```
1 #include <Adafruit_CircuitPlayground.h>
2 #include <IRremote.h>
3 #include <IRremoteInt.h>
4 #include <IRremoteTools.h>
5 #include <Keyboard.h>
6 #include <Wire.h>
7
8 void setup() {
9     // put your setup code here, to run once:
10 |
11 }
12
13 void loop() {
14     // put your main code here, to run repeatedly:
15
16 }
```

Arduino and sensors

Simple Digital Sensors



Challenge: Representing the position of the button in a graphic

Tips:

```
pinMode(#pin, MODE);
```

Identify pin and declares what it is(input?output?);

```
Input=DigitalRead(#pin);
```

Reads the value of the pin we associate as the button;

```
If(input==HIGH){
```

```
Serial.println("Button pressed"); }
```

```
Else{
```

```
Serial.println("Button not pressed"); }
```

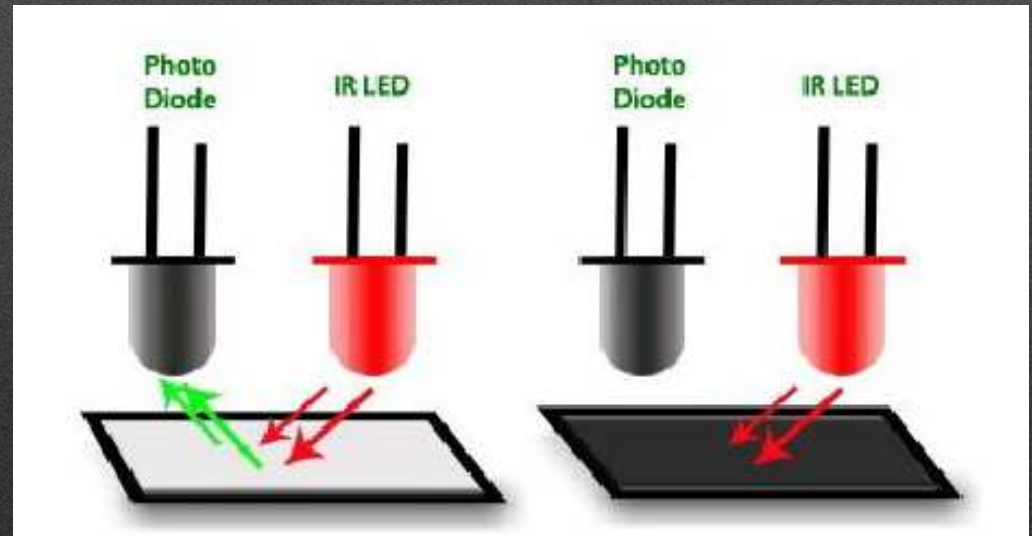


```
int pushButton = 2;
void setup() {
  Serial.begin(9600);
  pinMode(pushButton, INPUT);
}

void loop() {
  int input = digitalRead(pushButton);
  if (input == HIGH) {
    Serial.println("Botão premido");
  }
  else {
    Serial.println("Botão não premido");
  }

  delay(1);
}
```

Examples

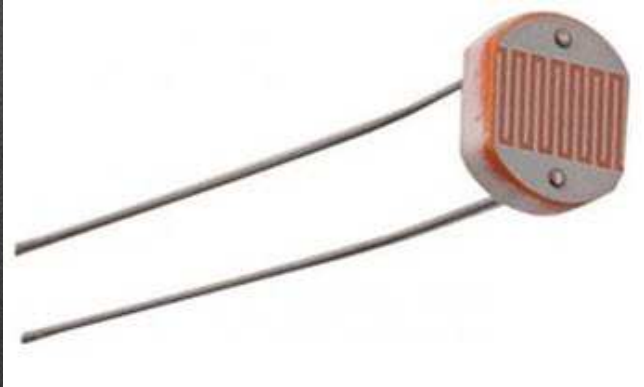
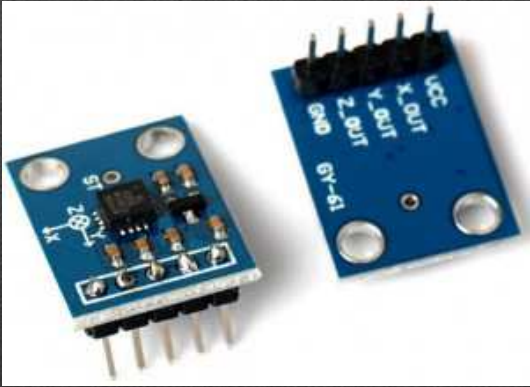


0

1

https://www.waveshare.com/wiki/Infrared_Proximity_Sensor

Analog Sensors

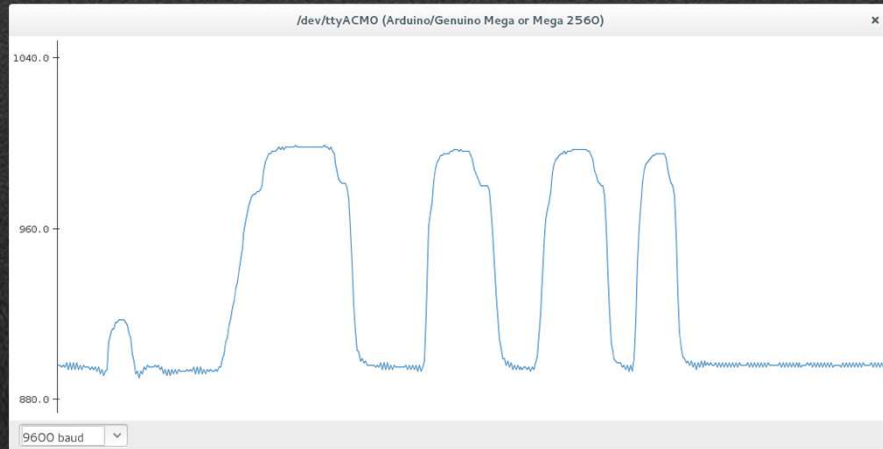


Analog Read

- `analogRead(#pin)`



0-1023 (10-bit)

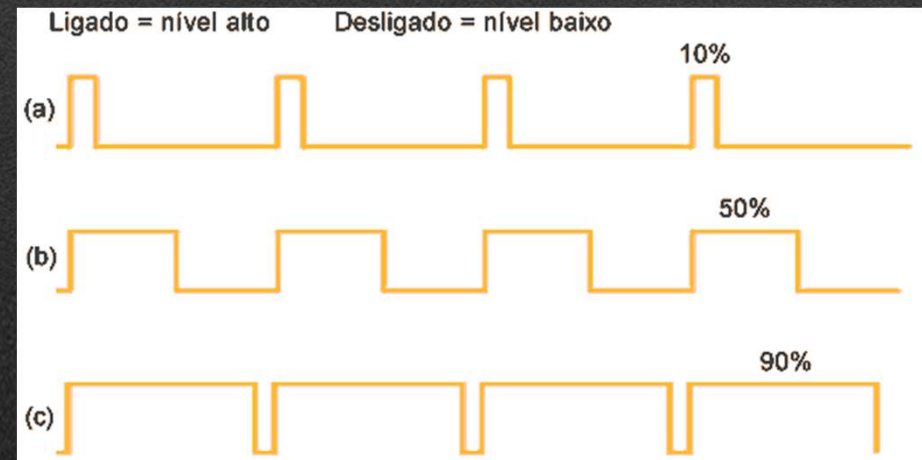


Digital Read

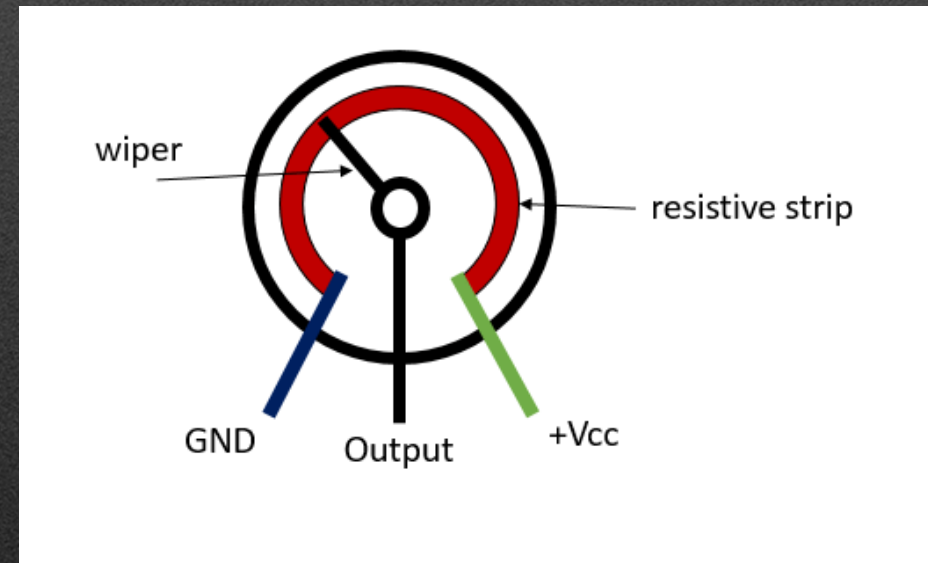
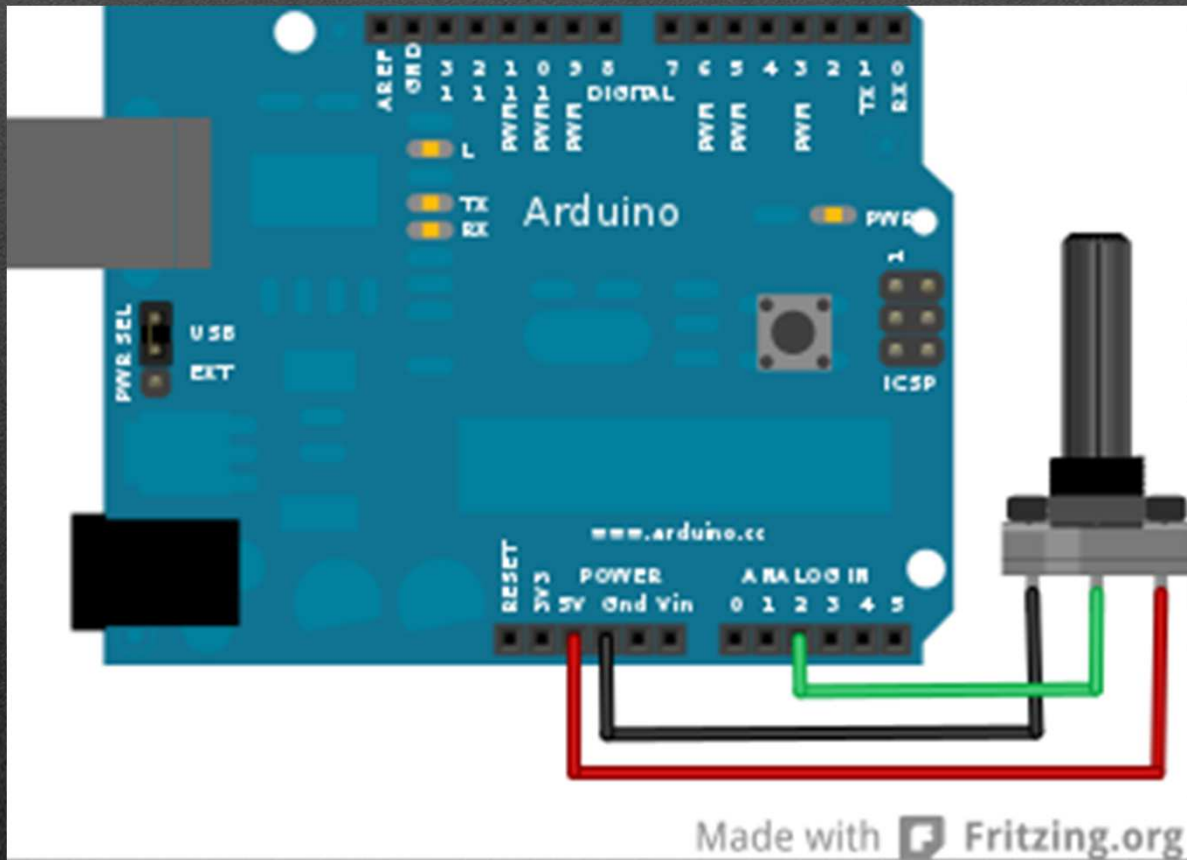
- `pinMode(#pin, MODE)`
- `digitalRead(#pin)`



0-255 (8-bit)



Potentiometer



Challenge: Representing the position of the potentiometer in a graphic

Tips:

```
AnalogRead(#pin);
```

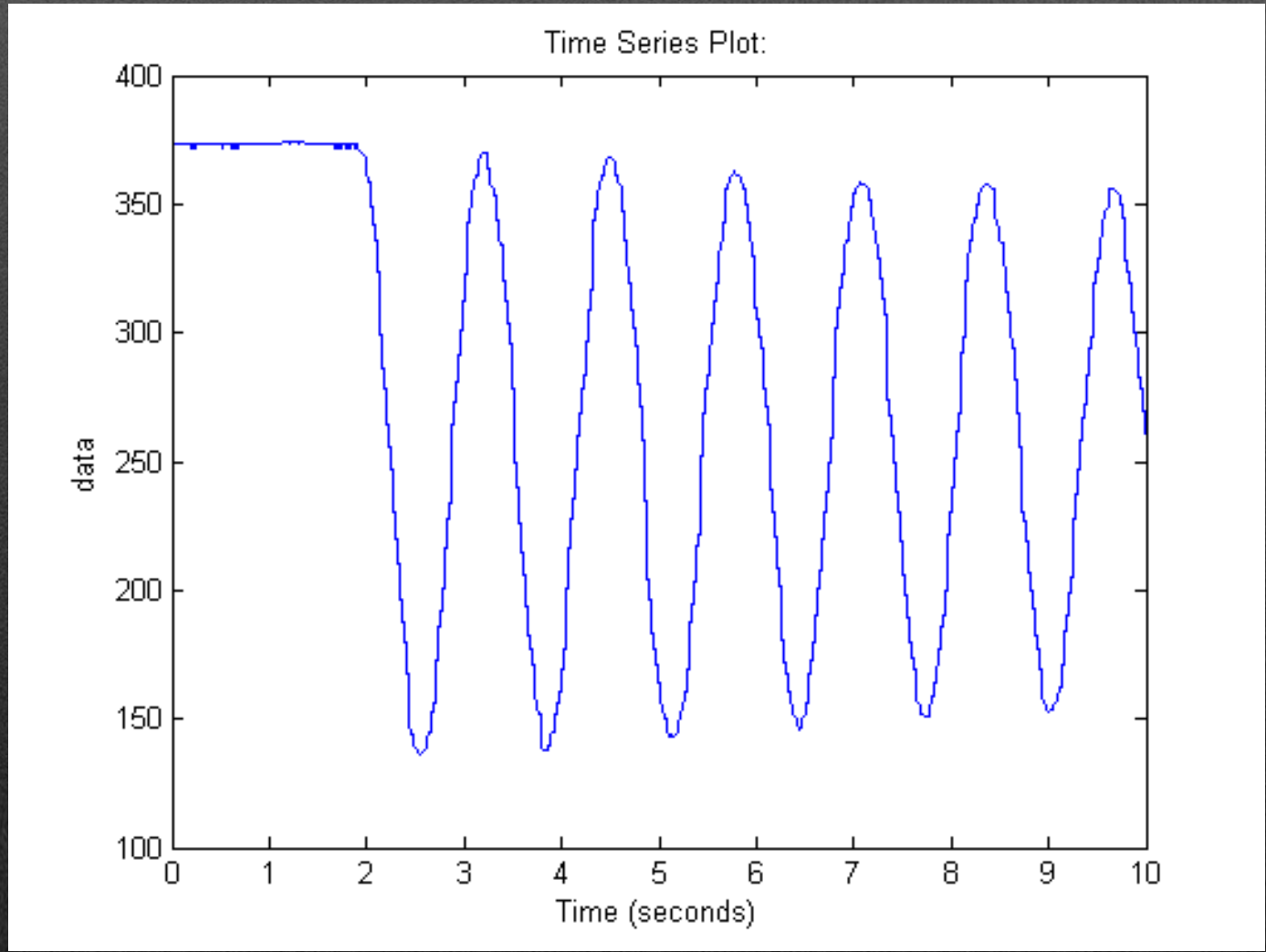
Read the tension on the potentiometer;

```
Output=Map(input, min_input, max_input, min_output, max_output);
```

Make “rule of 3” between two sets of values;

```
Serial.print("Pos="); Serial.println(valor);
```

Show result on computer.



```
const int analogInPin = A0;
const int analogOutPin = 9;

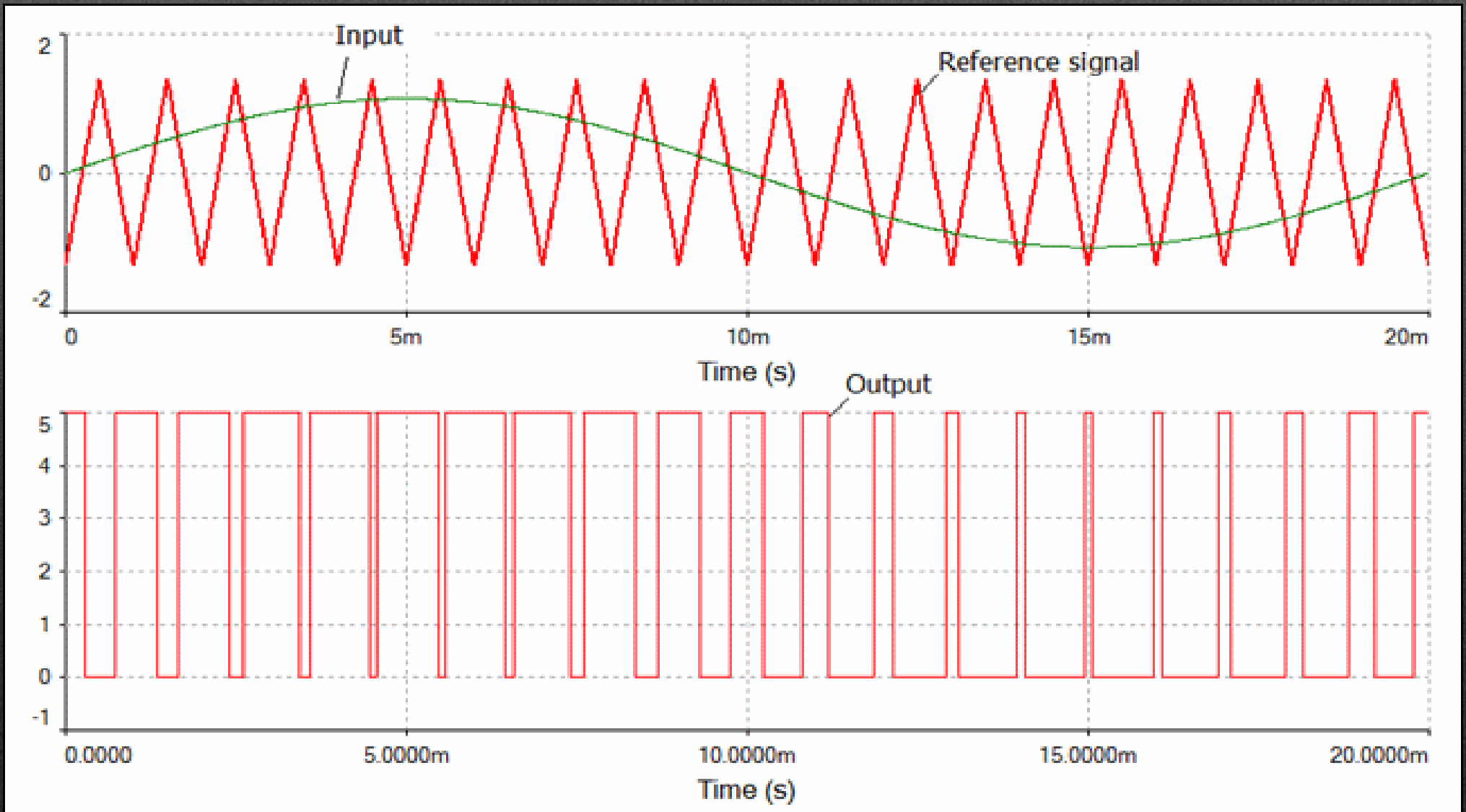
int sensorValue = 0;
int outputValue = 0;

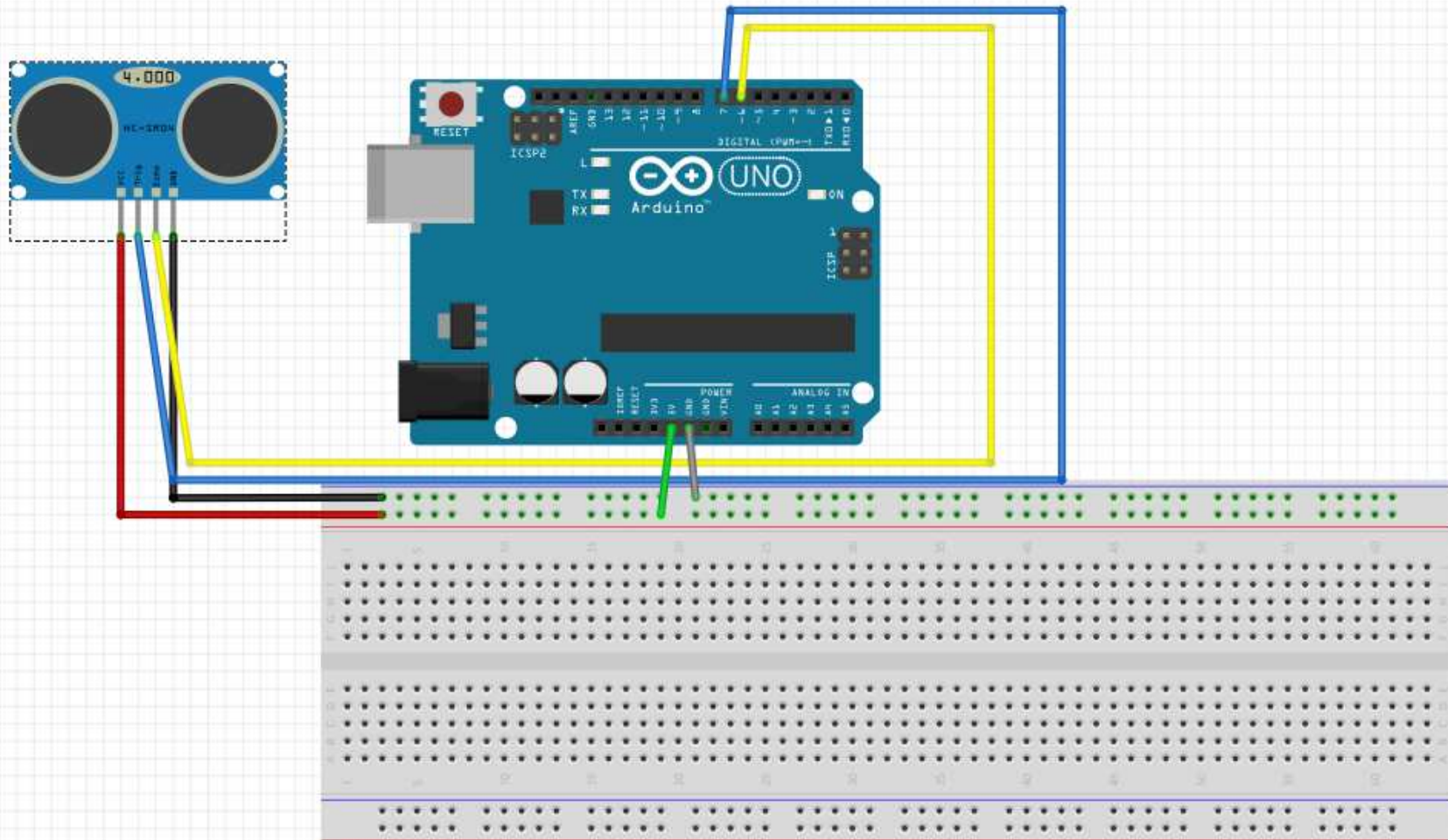
void setup() {
  Serial.begin(9600);
}

void loop() {
  sensorValue = analogRead(analogInPin);
  outputValue = map(sensorValue, 0, 1023, 0, 255);
  analogWrite(analogOutPin, outputValue);

  Serial.print("sensor = ");
  Serial.print(sensorValue);
  Serial.print("\t output = ");
  Serial.println(outputValue);

  delay(2);
}
```




```
#include <NewPing.h>

#define TRIGGER_PIN 7 // Arduino pin tied to trigger pin on the ultrasonic sensor.
#define ECHO_PIN 6 // Arduino pin tied to echo pin on the ultrasonic sensor.
#define MAX_DISTANCE 400 // Maximum distance we want to ping for (in centimeters). Maximum sensor distance is rated at 400-500cm.

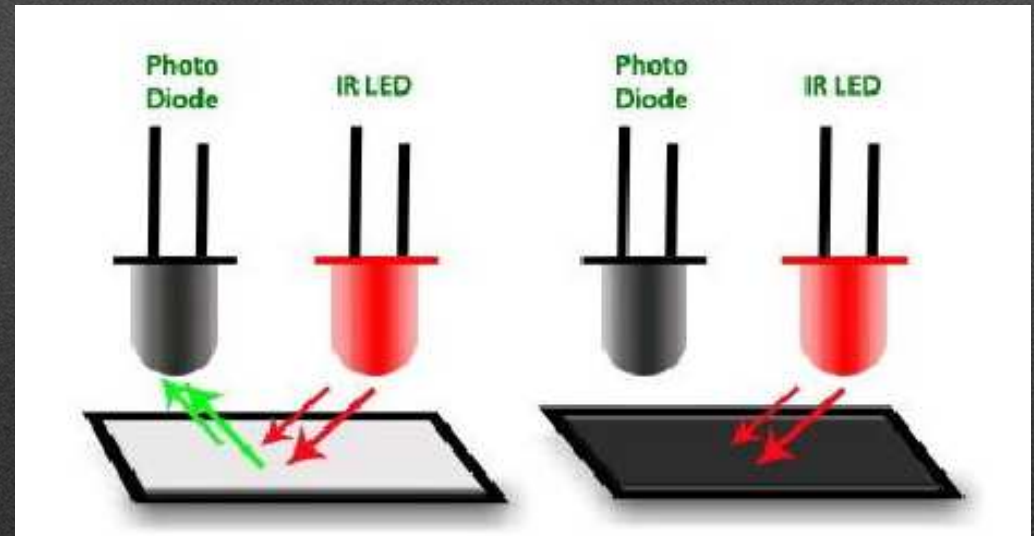
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing setup of pins and maximum distance.

void setup() {
  Serial.begin(115200); // Open serial monitor at 115200 baud to see ping results.
}

void loop() {
  delay(50); // Wait 50ms between pings (about 20 pings/sec). 29ms should be the shortest delay between pings.
  Serial.print("Ping: ");
  Serial.print(sonar.ping_cm()); // Send ping, get distance in cm and print result (0 = outside set distance range)
  Serial.println("cm");
}
```

So, what exactly is gonna be used in this project?

Examples



0

1

https://www.waveshare.com/wiki/Infrared_Proximity_Sensor

HC SR04-Sonar

