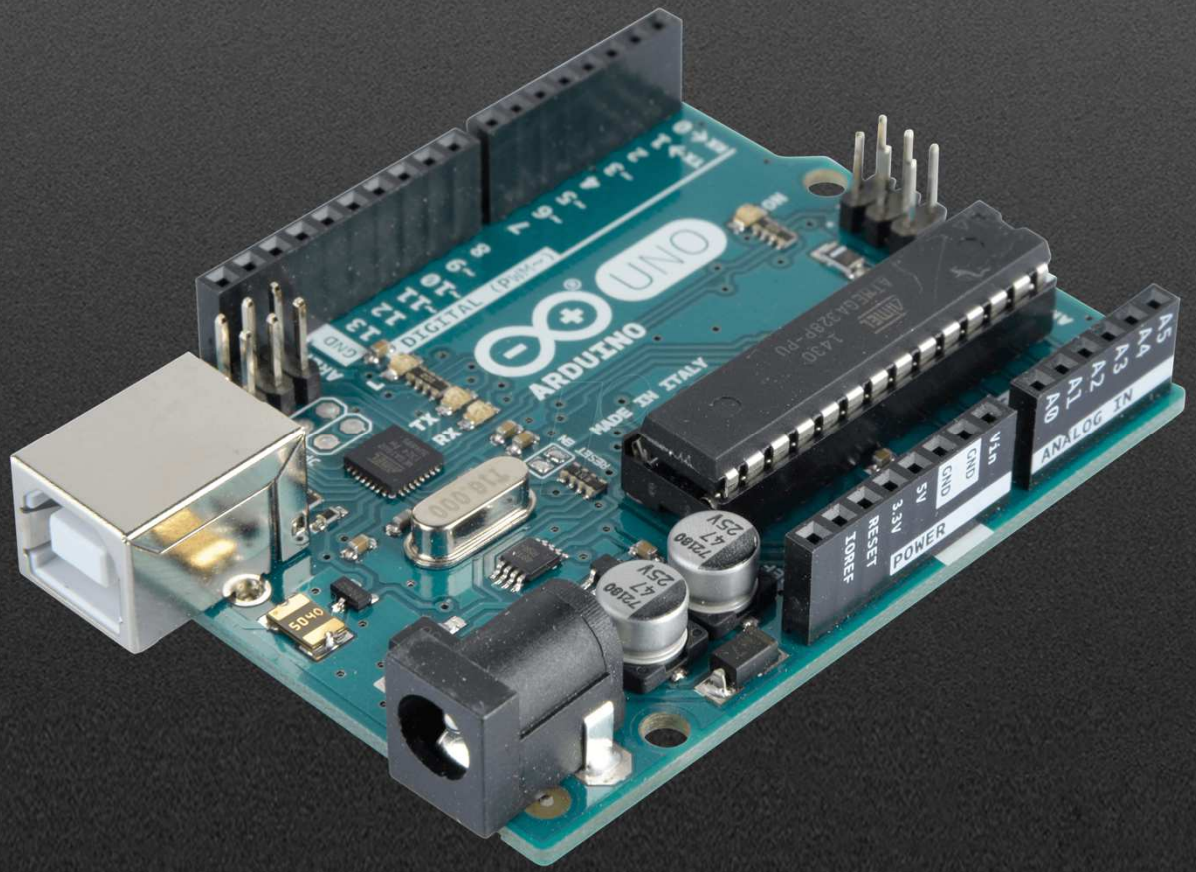
MOBILIDADE SUSTENTÁVEL

Luís Martins
Duarte Abreu

ARDUINO UNO

Arduino UNO


UNO
ARDUINO
MADE IN ITALY
DIGITAL (PWM~)
POWER
ANALOG IN
RESET
3.3V
5V
GND
Vin
A0
A1
A2
A3
A4
A5

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.

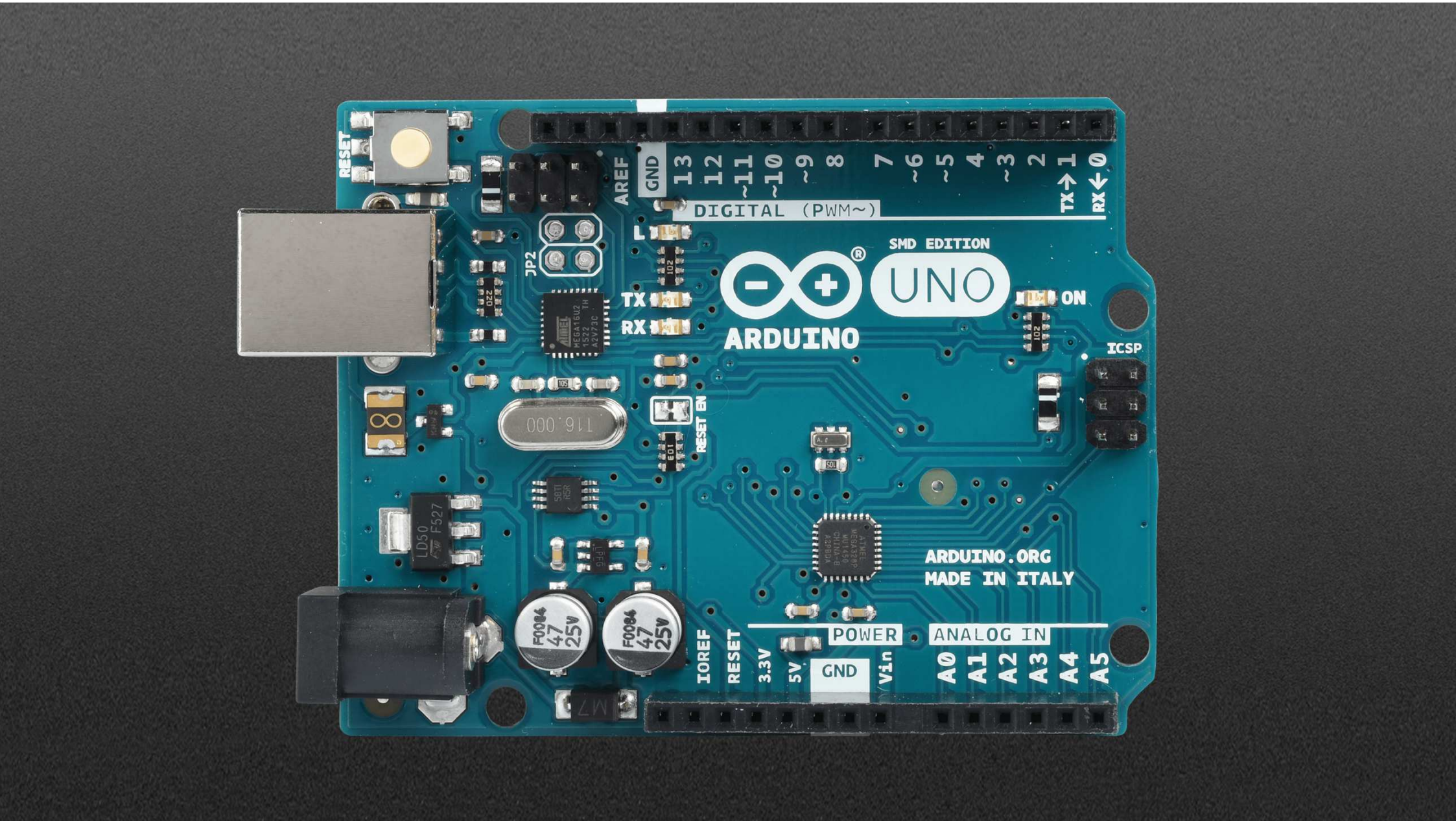
Windows Installer
Windows ZIP file for non admin install

Windows app Get

Mac OS X 10.7 Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM

Release Notes
Source Code
Checksums (sha512)

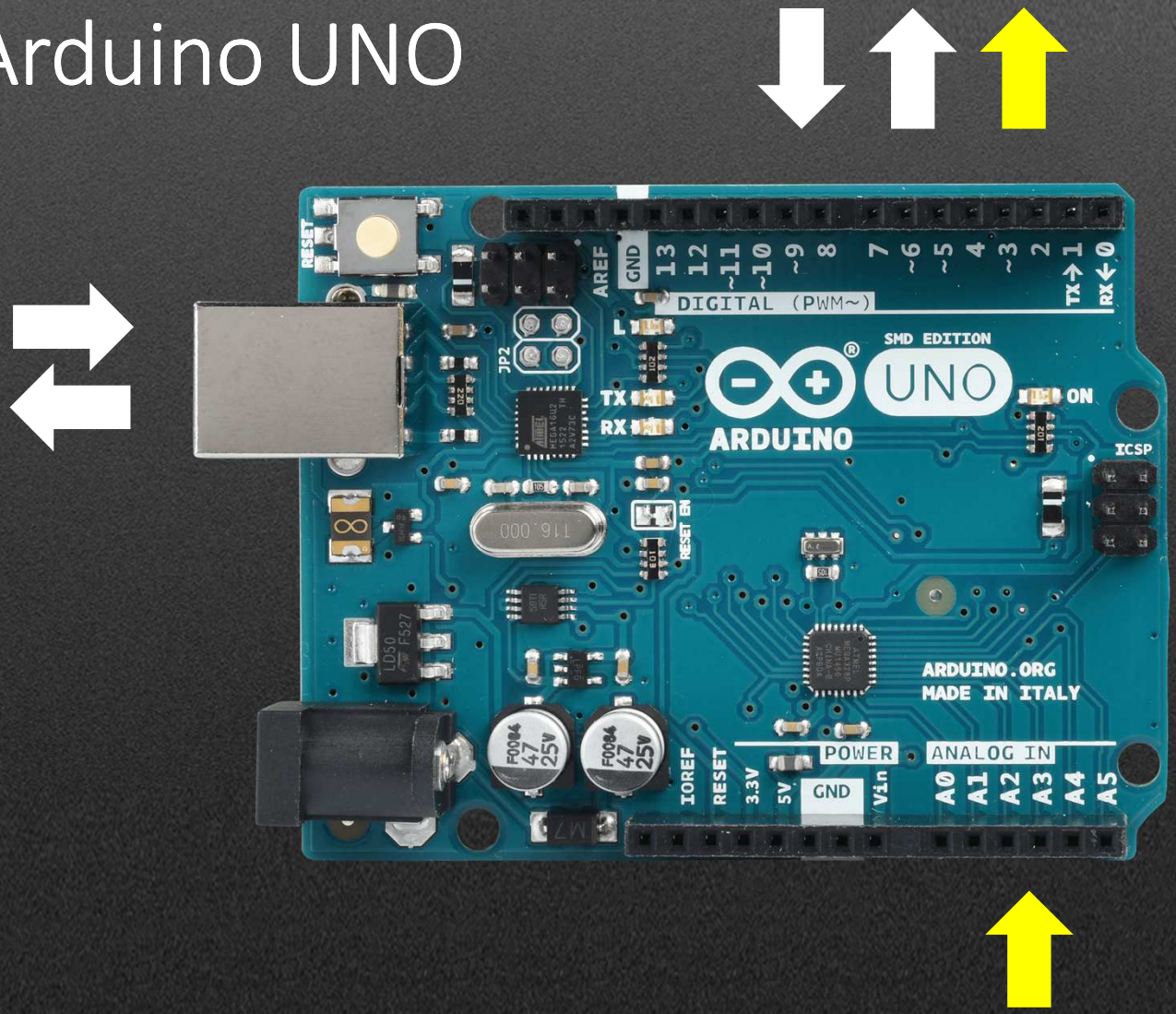

RESET
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
SMD EDITION
UNO
ARDUINO
ON
ICSP
JP2
TX
RX
RESET EN
ARDUINO.ORG
MADE IN ITALY
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5

Digital VS Analógico

Analog Signal

Digital Signal

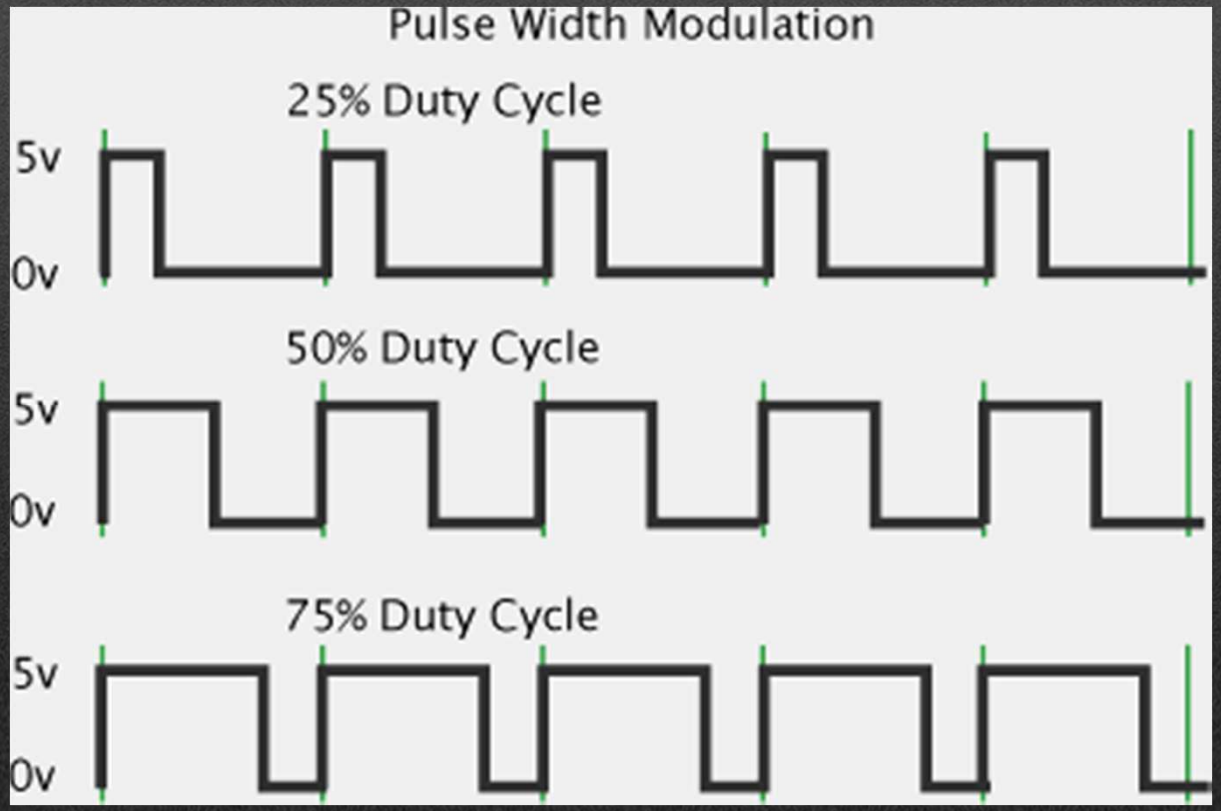
Arduino UNO



DIGITAL

ANALOG

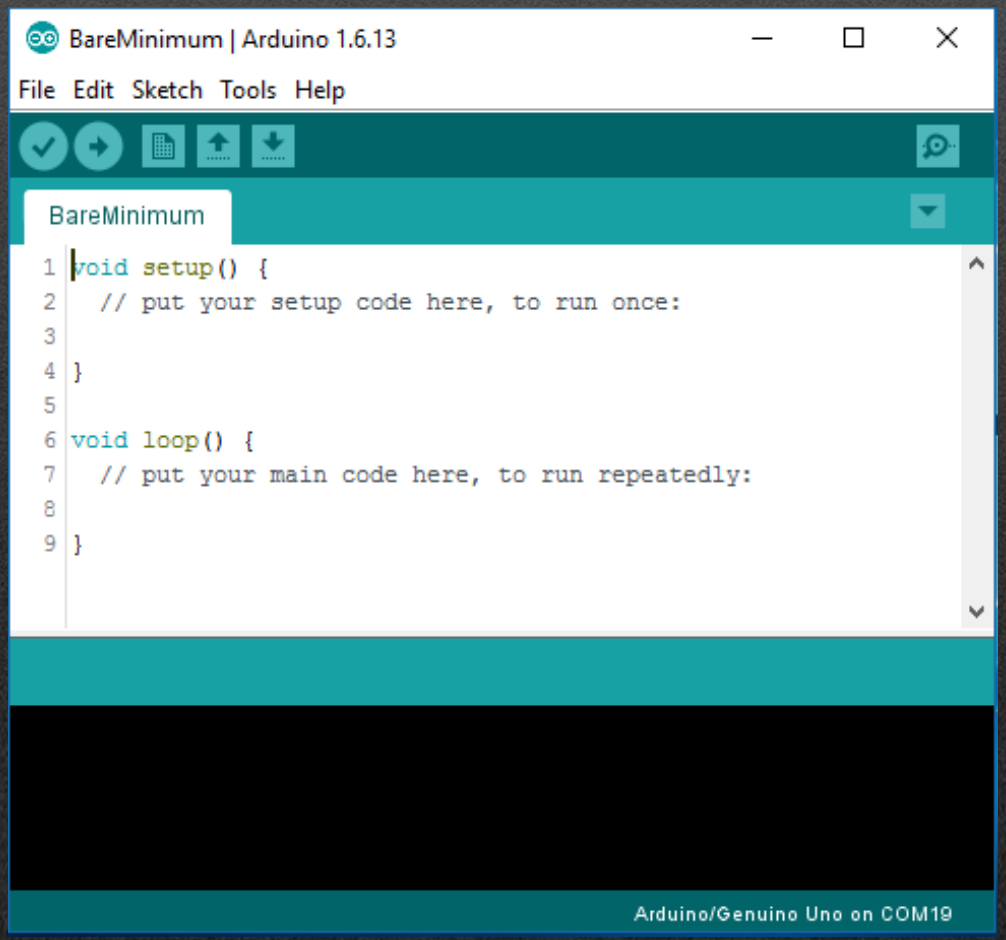
PWM (Pulse Width Modulation)



Funciona utilizando médias móveis para representar valores “analógicos”.

Por defeito tem 8 bits de resolução. (256 valores). Mas pode ser alterado com o comando analogWriteResolution() até 12 bits (65536 valores).

Janela principal



EDIT

Undo	Ctrl+Z
Redo	Ctrl+Y
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
Comment/Uncomment	Ctrl+Slash
Increase Indent	Tab
Decrease Indent	Shift+Tab
Find...	Ctrl+F
Find Next	Ctrl+G
Find Previous	Ctrl+Shift+G

FILE

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

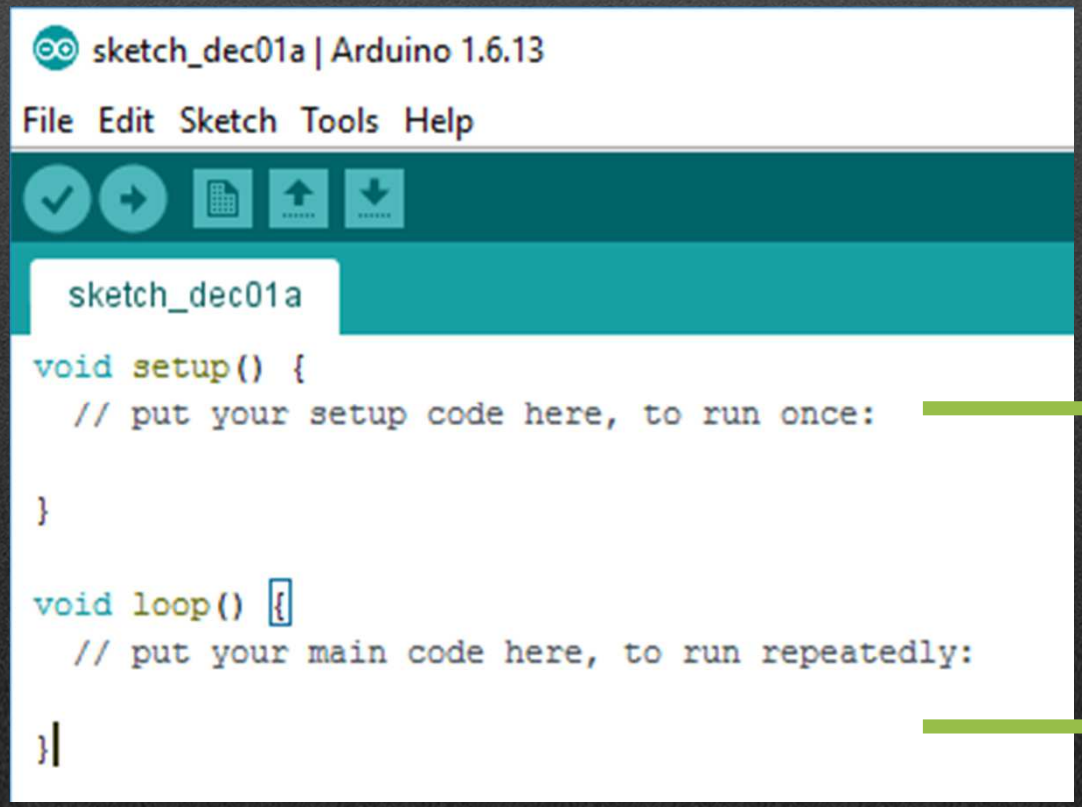
SKETCH

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

TOOLS

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

Bare Minimum



Código de preparação.
Aqui devo indicar o que vou ligar ao microcontrolador, seclarar as variáveis e iniciar as comunicações SPI

Código de funcionamento.
Aqui devo colocar a rotina que quero executar. Todo o código que estiver dentro do loop será executado de forma sequencial e repetida até se desligar o microcontrolador ou até ser pressionado o botão reset.

Blink

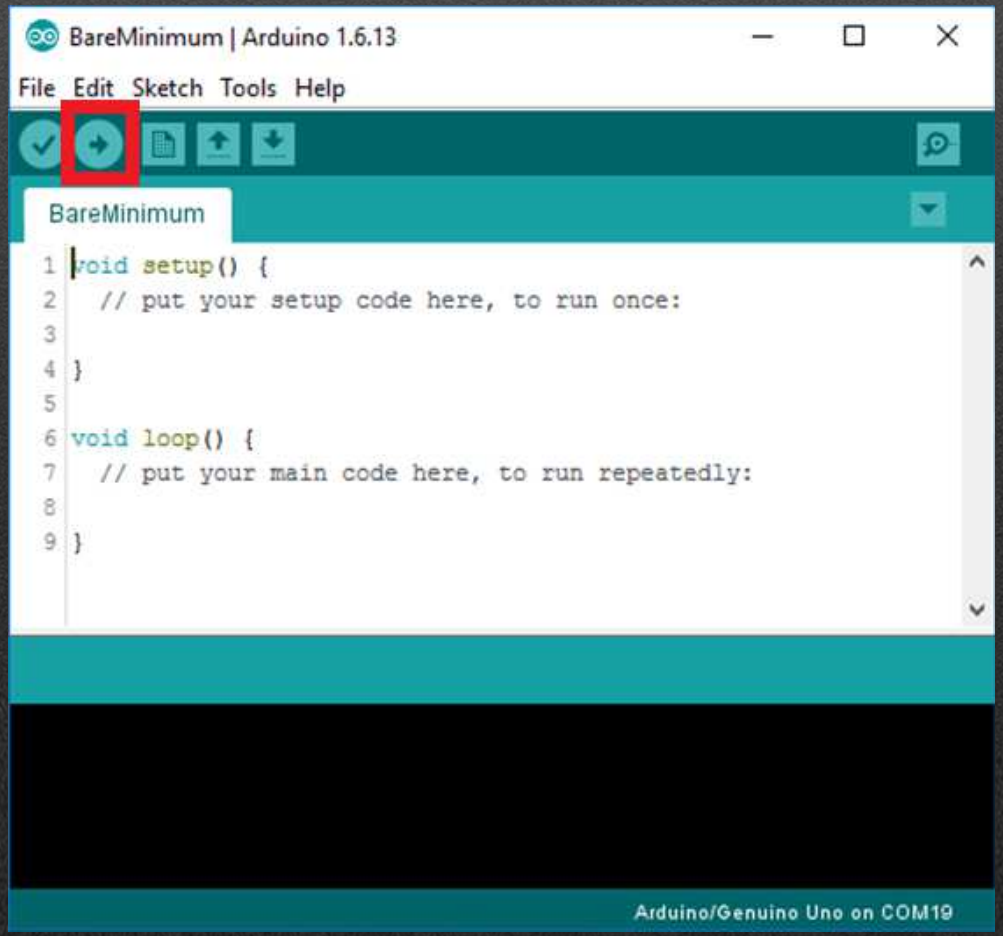
Vamos usar o LED integrado, no pin #13 do arduino. Vamos chamar-lhe LED e declará-lo como um OUTPUT.

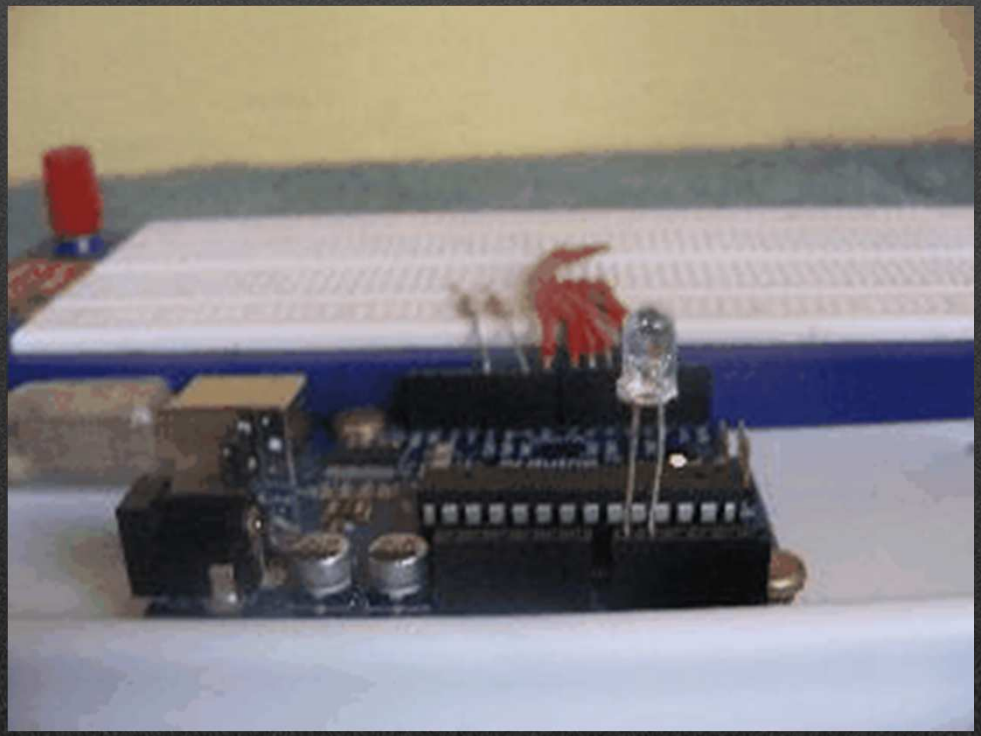
O LED liga durante um segundo e desliga durante um segundo. Este ciclo repete-se indeterminadamente.

COMANDOS IMPORTANTES

- pinMode(#pin,OUTPUT); - define o pin selecionado para ser um output digital.
- pinMode(#pin,INPUT); - define o pin selecionado para ser um input digital.
- Delay(milissegundos); - Conta x milissegundos antes de executar o comando seguinte
- // ISTO É UM COMENTÁRIO – Esta parte do código não é lida pelo compilador e serve para o programador tirar notas do código.
- As cores são importantes!!! – No IDE do Arduino, as cores indicam-nos se estamos a lidar com funções, variáveis, bibliotecas etc.. Servem para tornar o código legível e para dizer ao utilizador se está a declarar bem as variáveis e funções.

Carregar o Código



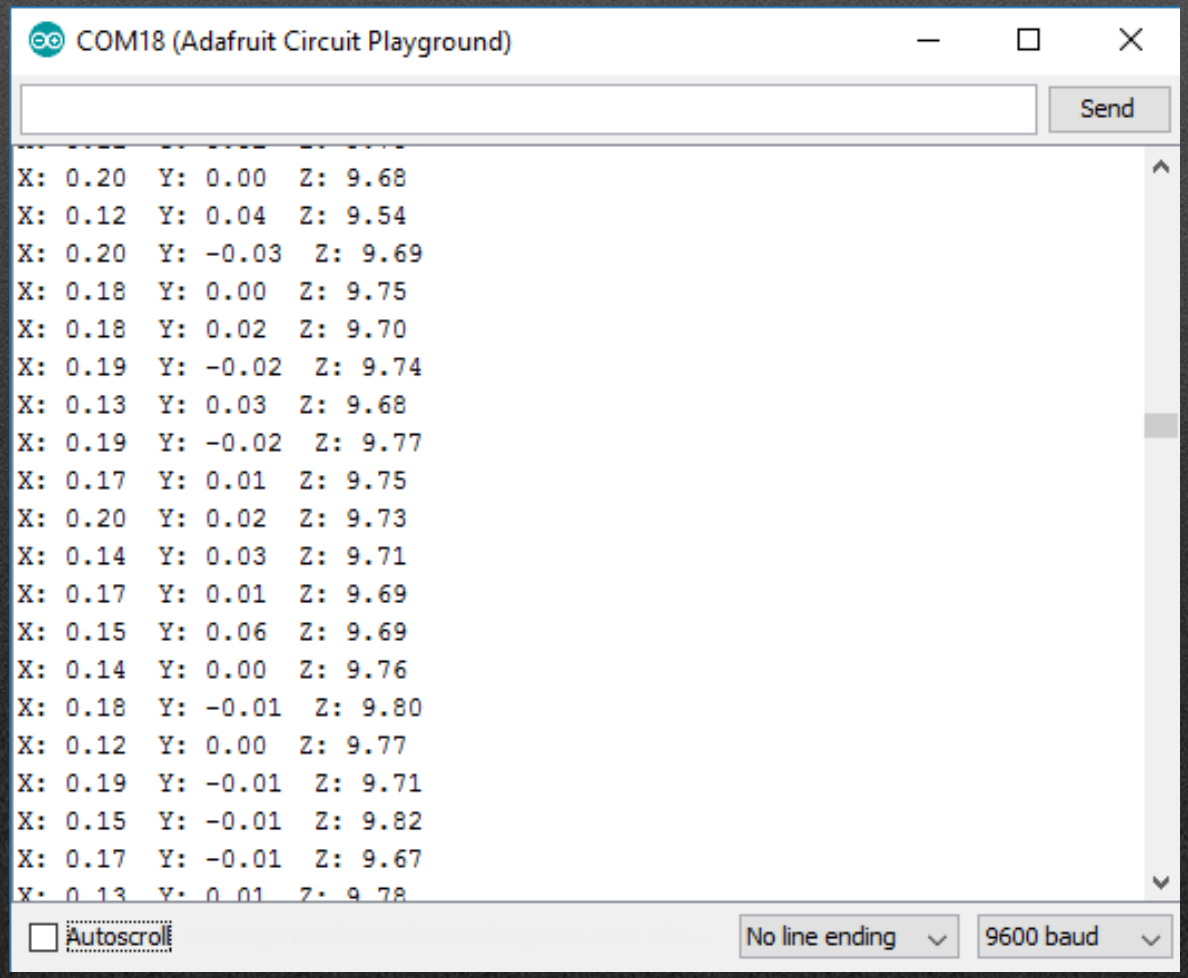


Comunicação Serial

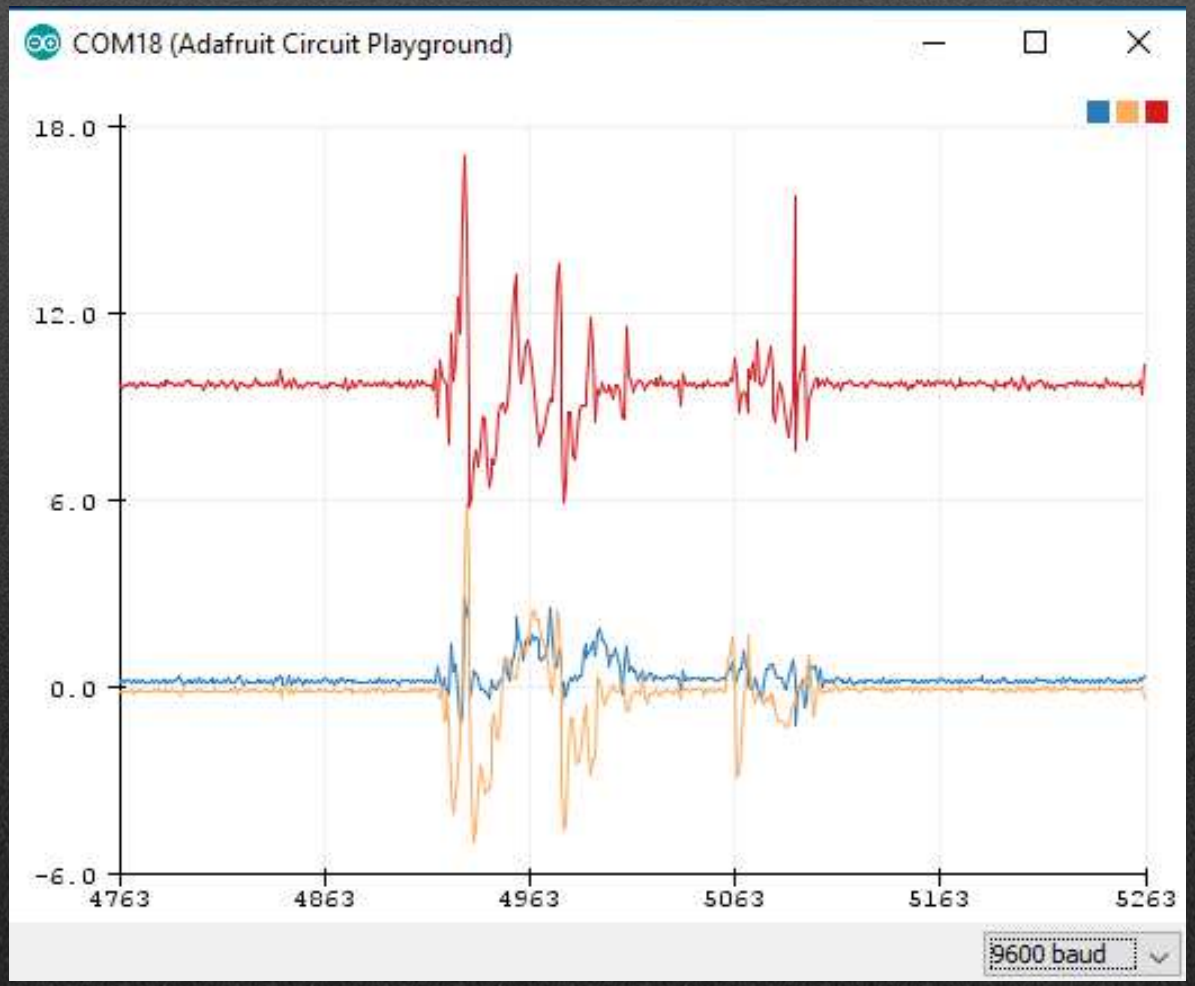
- Serve de comunicação base entre o arduino e o computador.
- Serve para comunicar entre arduinos.

- Serial.begin(baudRate);
- Serial.print(“texto”,var);
- Serial.println(“texto”,var);
- Serial.write();
- Serial.read();

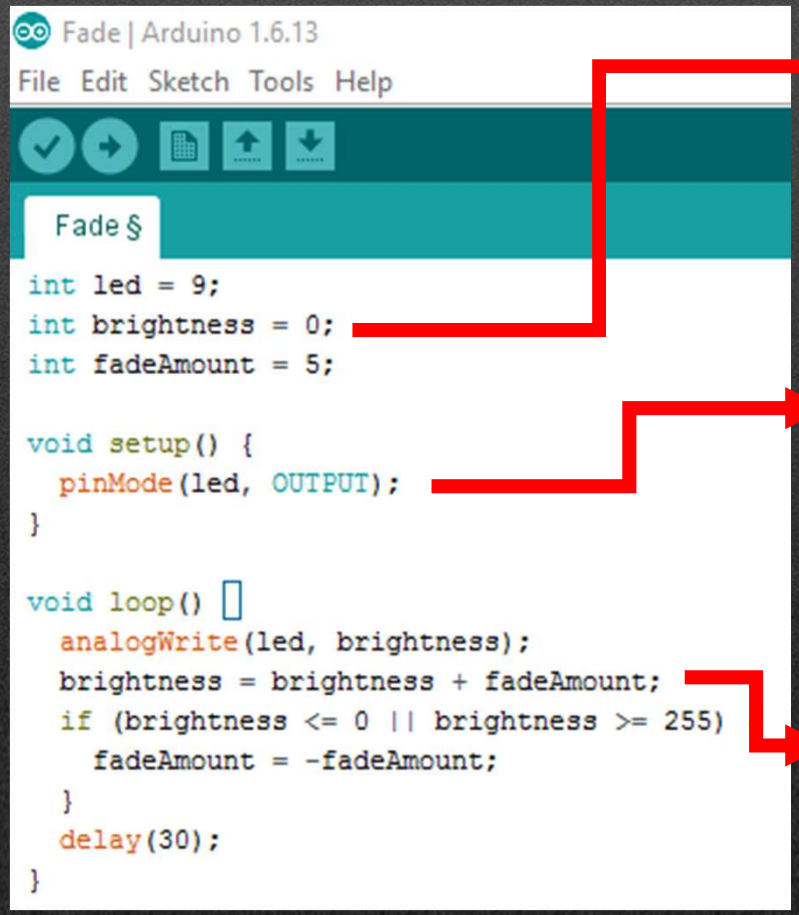
Serial Console



Serial Plotter



Fade

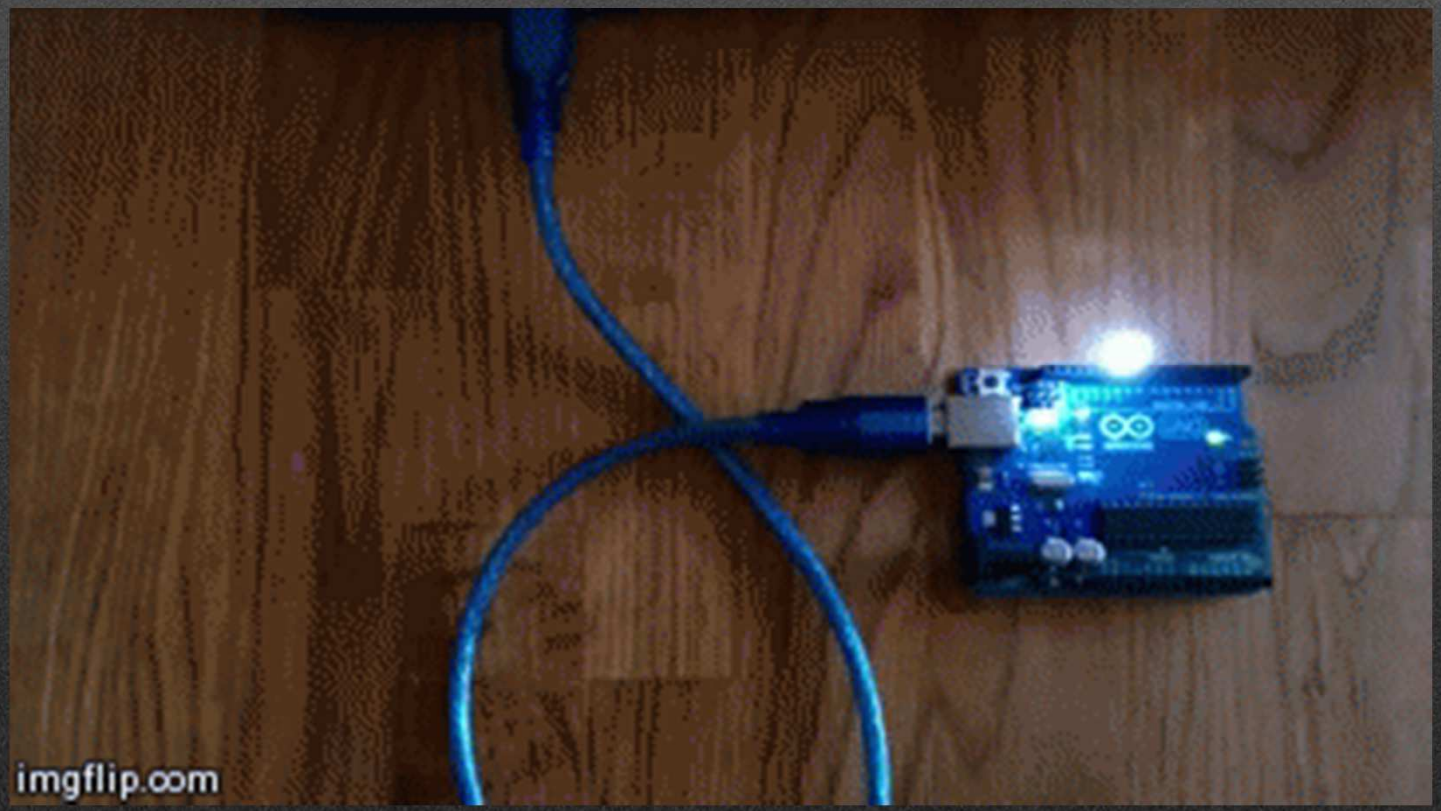


Identifica o pin#9 como sendo o pin referente ao LED.
Inicia as variáveis brilho e fadeAmount com os valores respetivos.

Configura o Pin #9 como OUTPUT. Onde vamos escrever os valores analógicos de intensidade da luz utilizando PWM.

A cada ciclo, incrementamos os valor da variável “brightness” desde 0 até 255. Quando esse valor for igual a 255, incrementa se de forma decrescente. Cada incremento/decréscimo desse valor é seguido pelo comando “analogWrite(led,brightness);” que escreve usando PWM o valor da variável brightness no pin associado ao led.

Fade

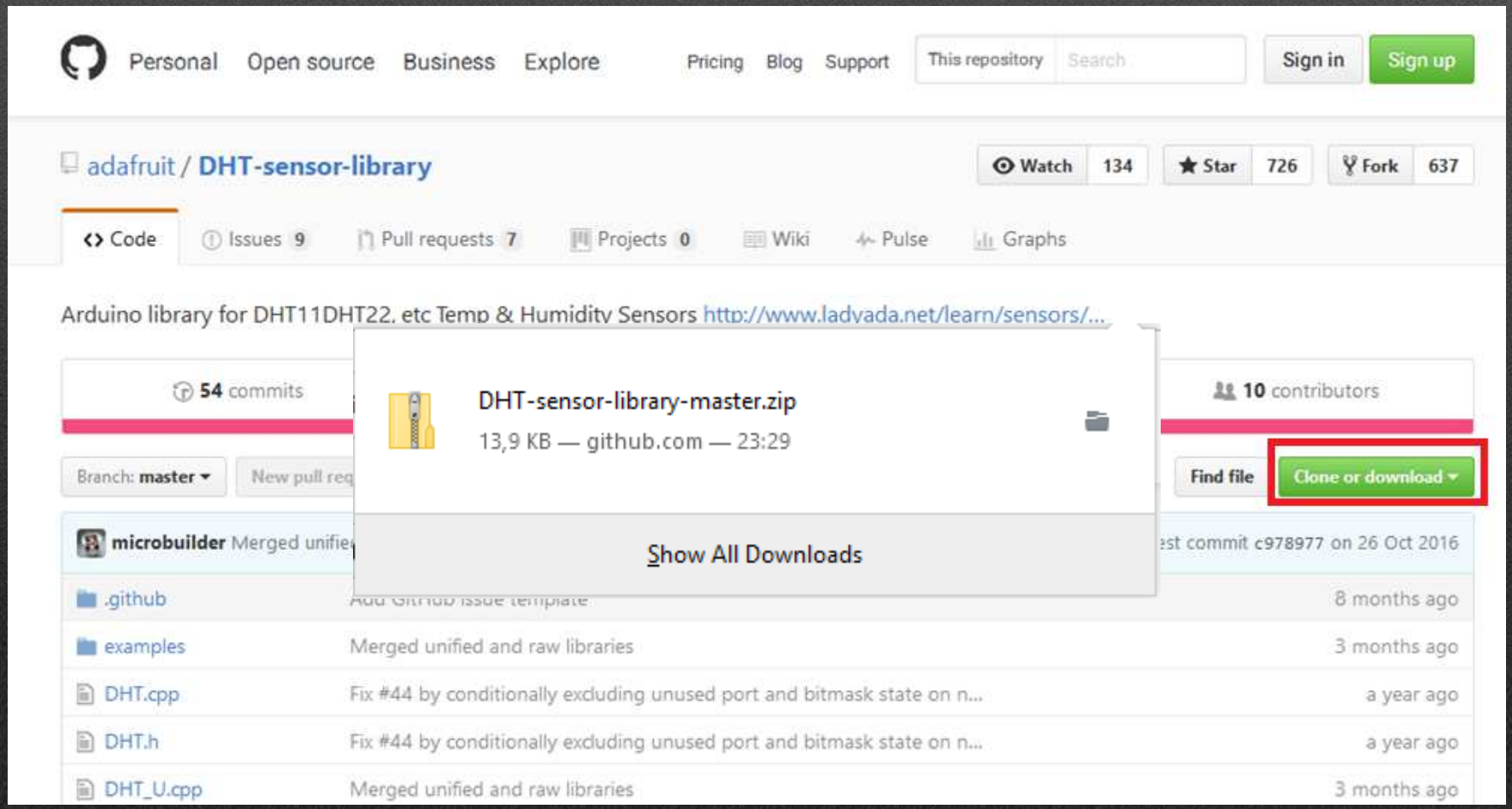


Libraries

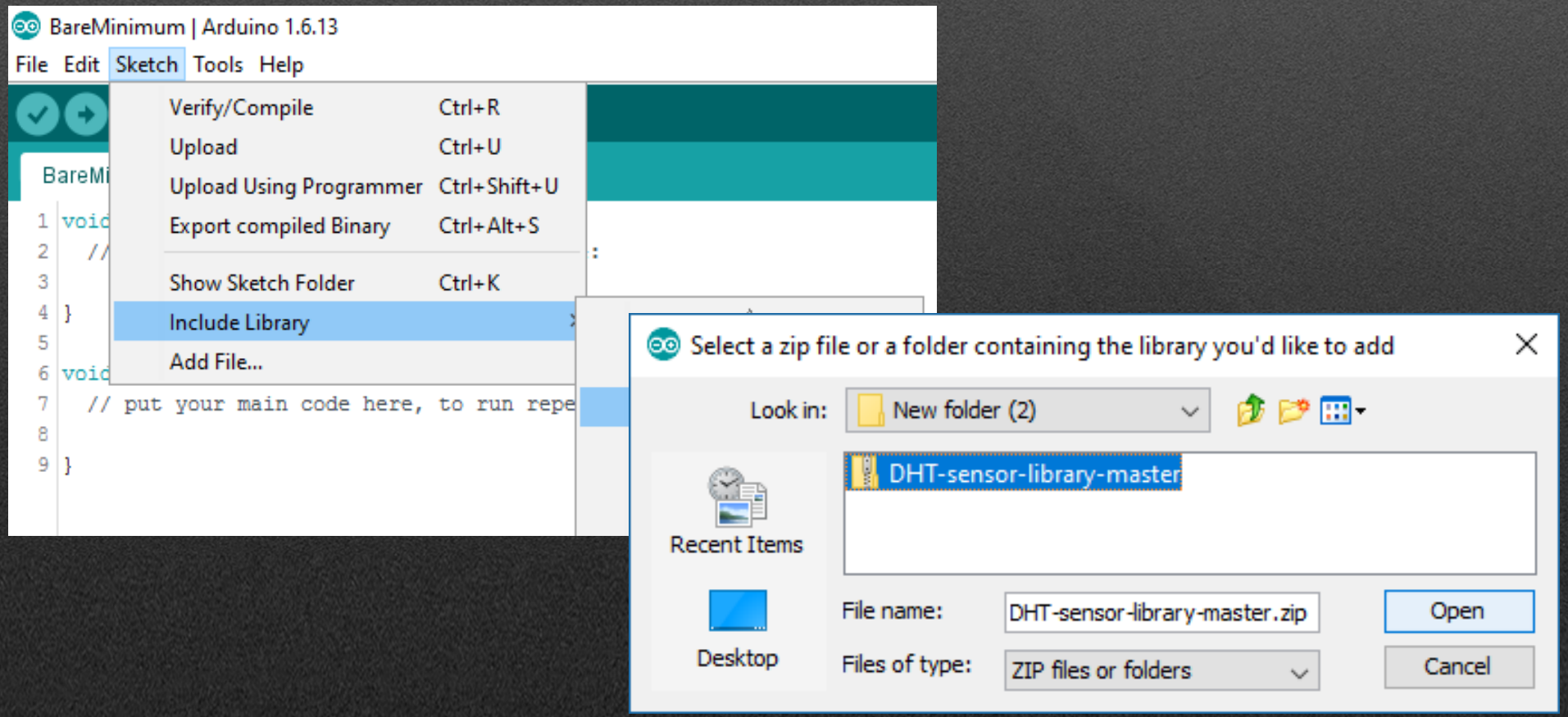
- Libraries são conjuntos de funções que não são nativas da linguagem do Arduino.
- Devem ser chamadas no inicio do código.
- São ficheiros do tipo .h
- Exemplo:

include<DHT.h>

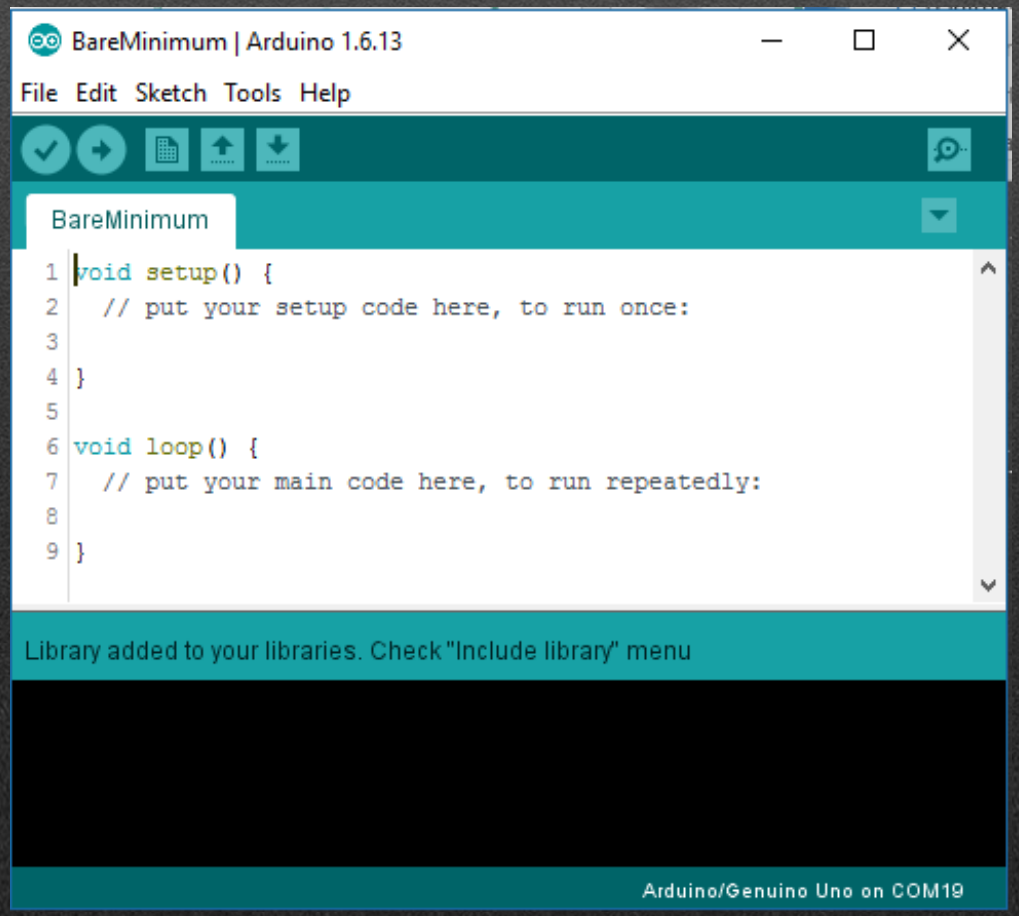
Instalação de libraries

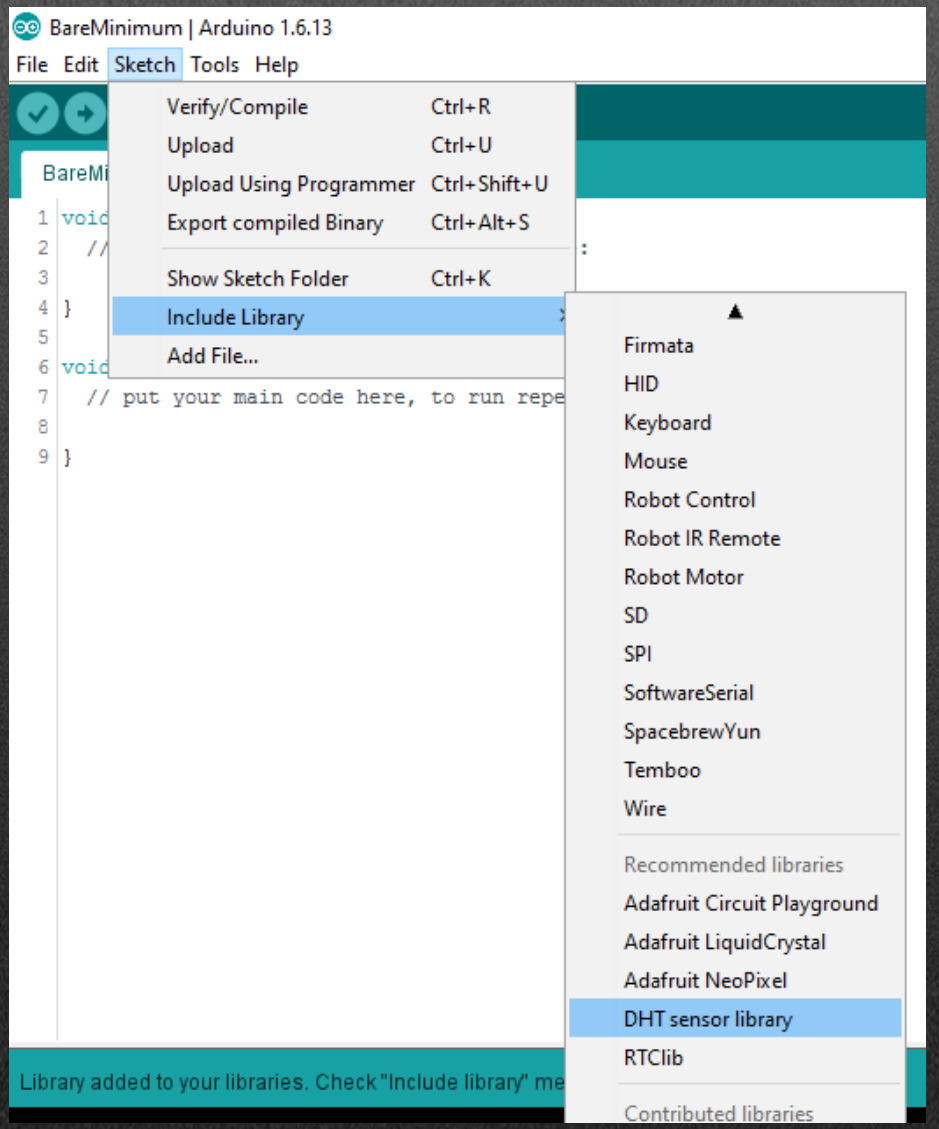


Instalação de libraries



Instalação de libraries



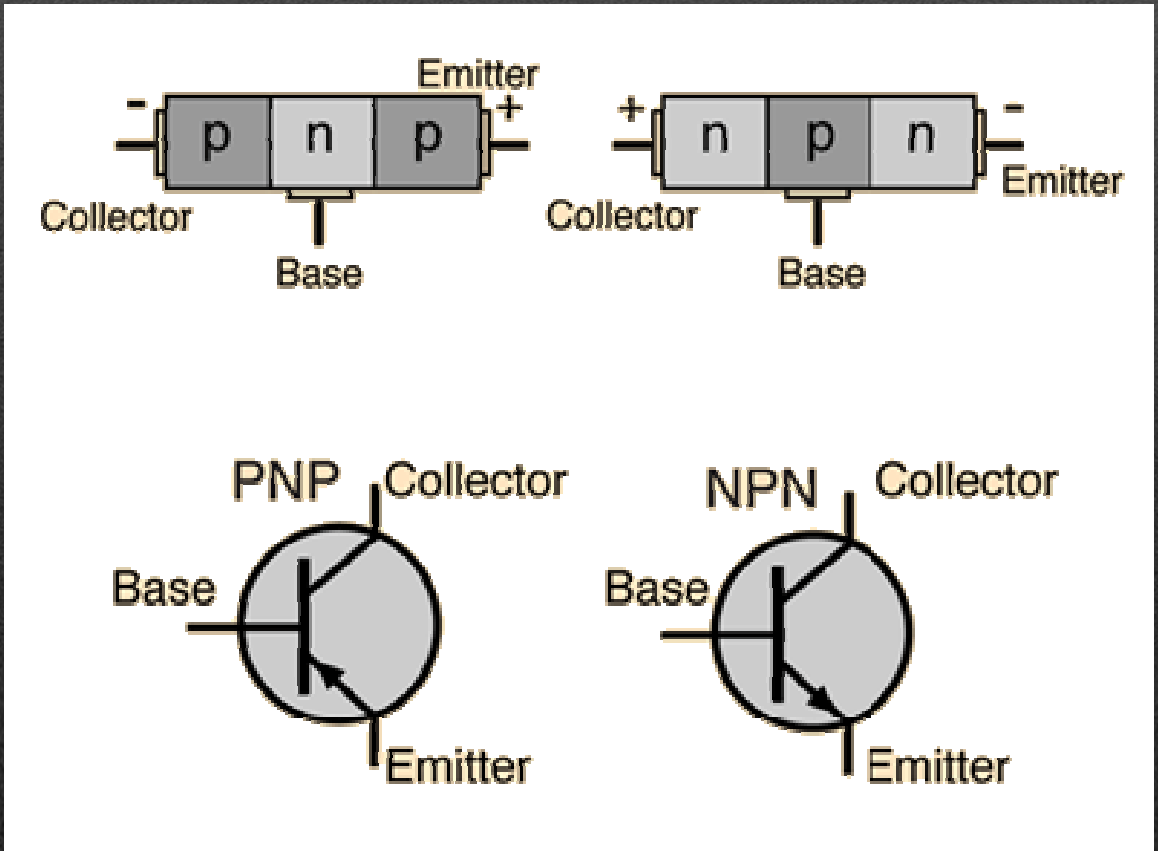


Chamar libraries no código

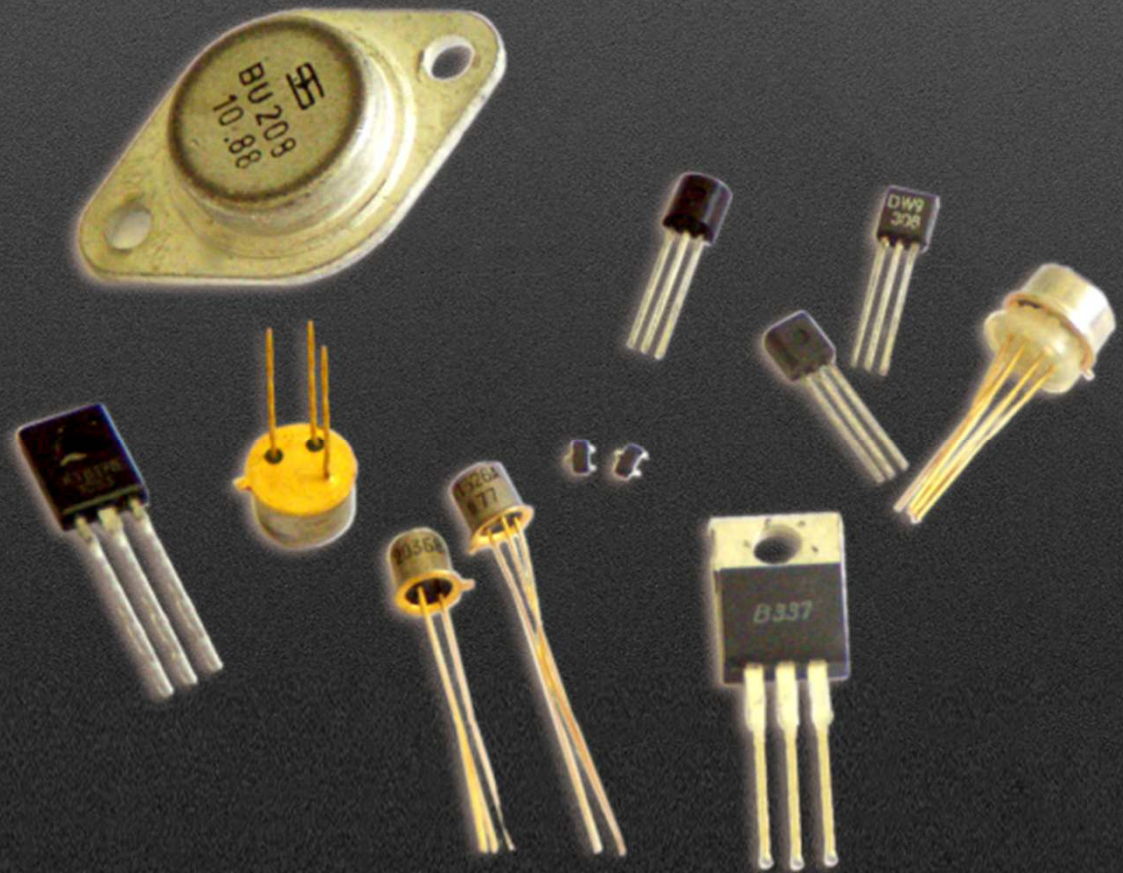
```
1  #include <Adafruit_CircuitPlayground.h>
2  #include <IRremote.h>
3  #include <IRremoteInt.h>
4  #include <IRremoteTools.h>
5  #include <Keyboard.h>
6  #include <Wire.h>
7
8  void setup() {
9    // put your setup code here, to run once:
10
11 }
12
13 void loop() {
14   // put your main code here,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

CONTROLO DE MOTORES

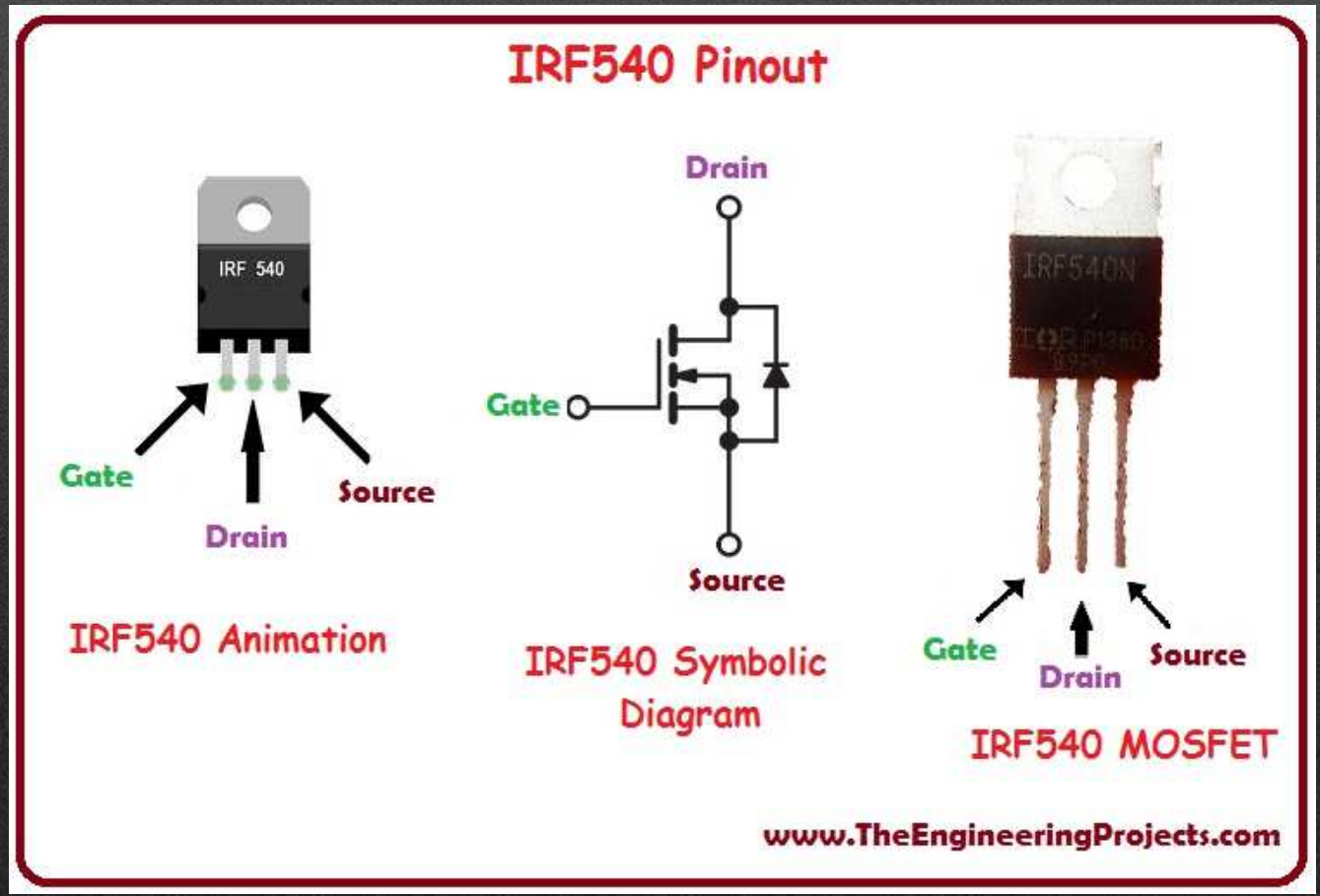
Transistor


Emitter
-
+
p
n
p
Collector
Base
+
-
n
p
n
Collector
Emitter
Base
PNP
Collector
Base
Emitter
NPN
Collector
Base
Emitter

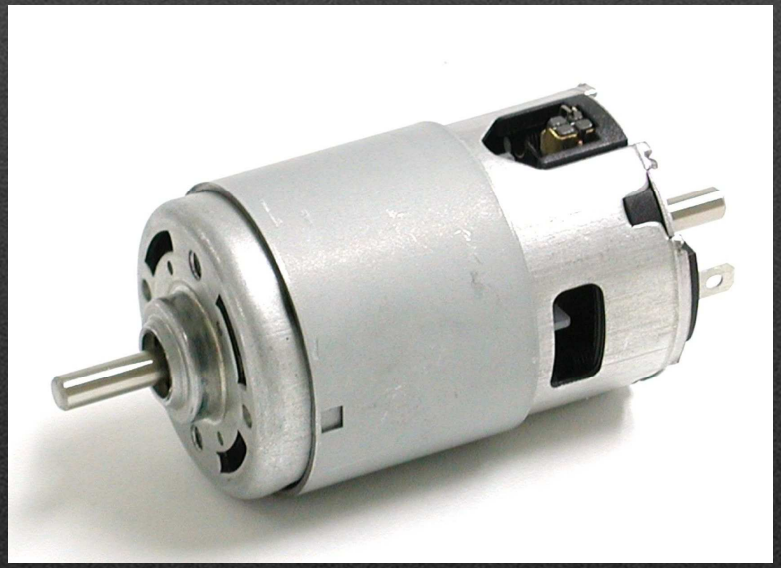
Tipos de transistor

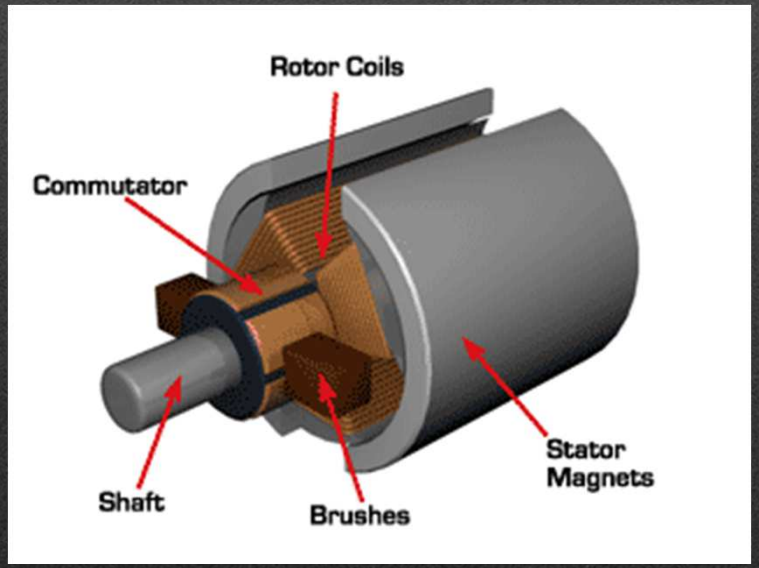


MOSFET IRF 520

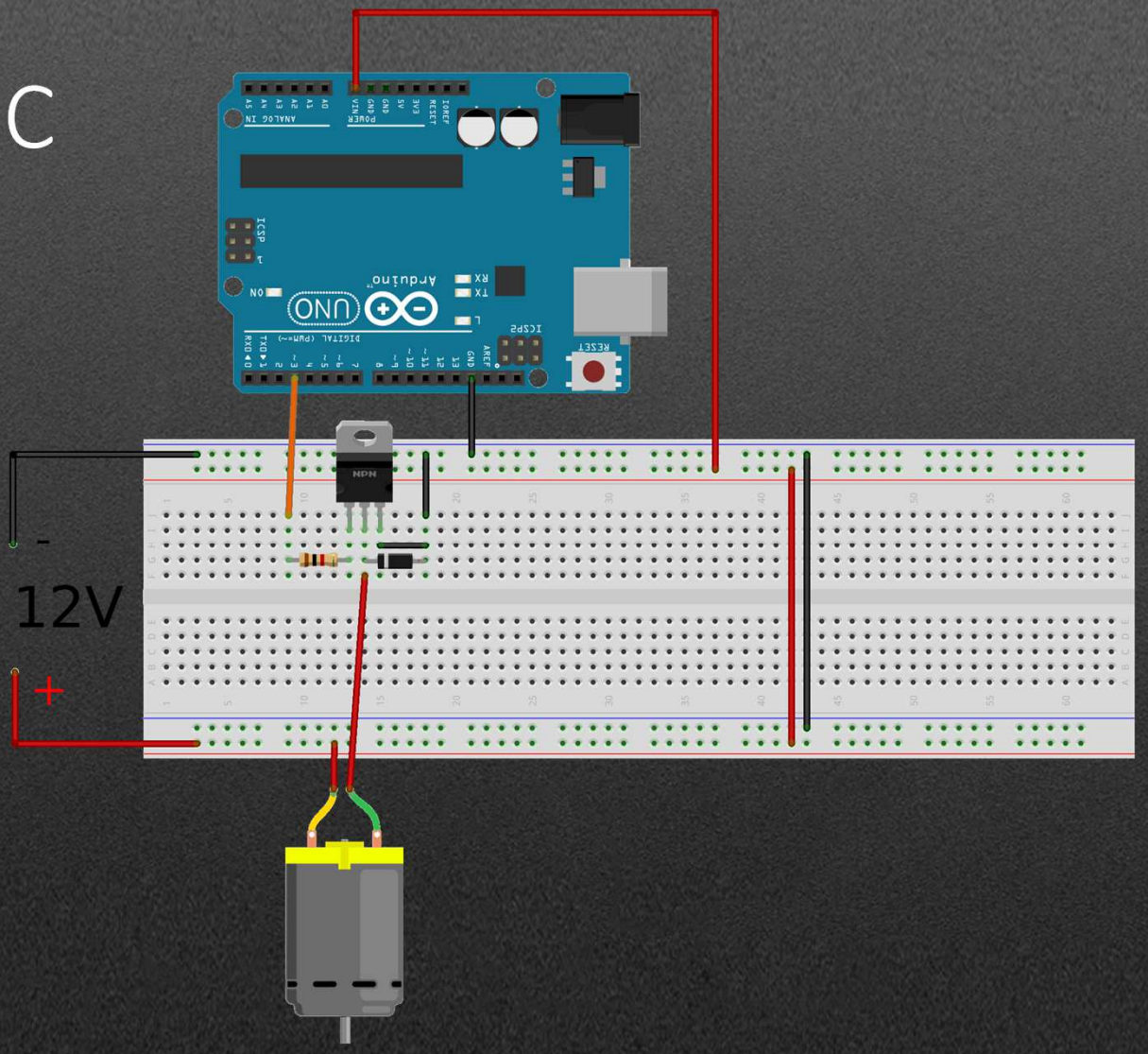


Motores, Servos e Steppers





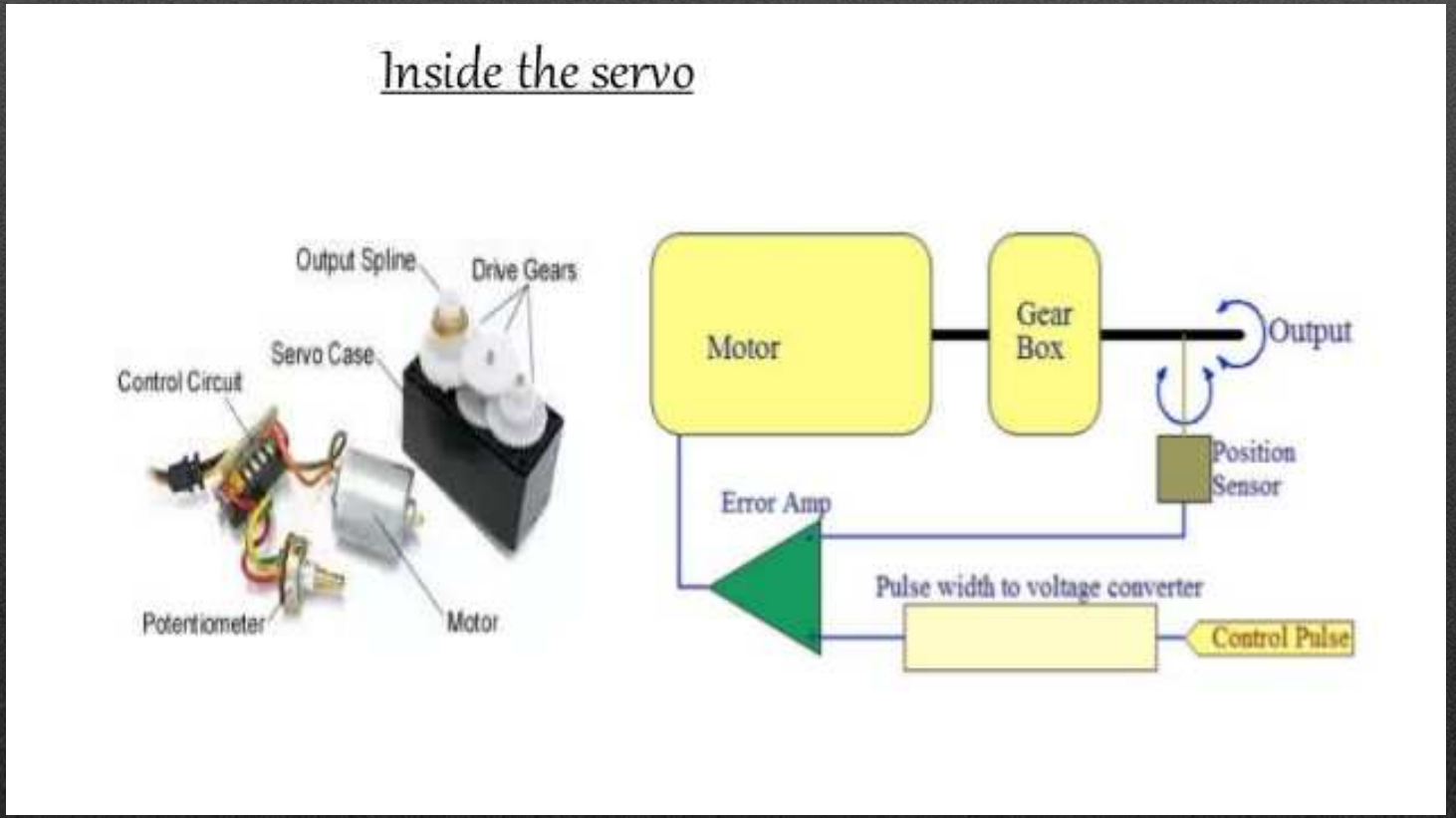
Circuito Motor DC



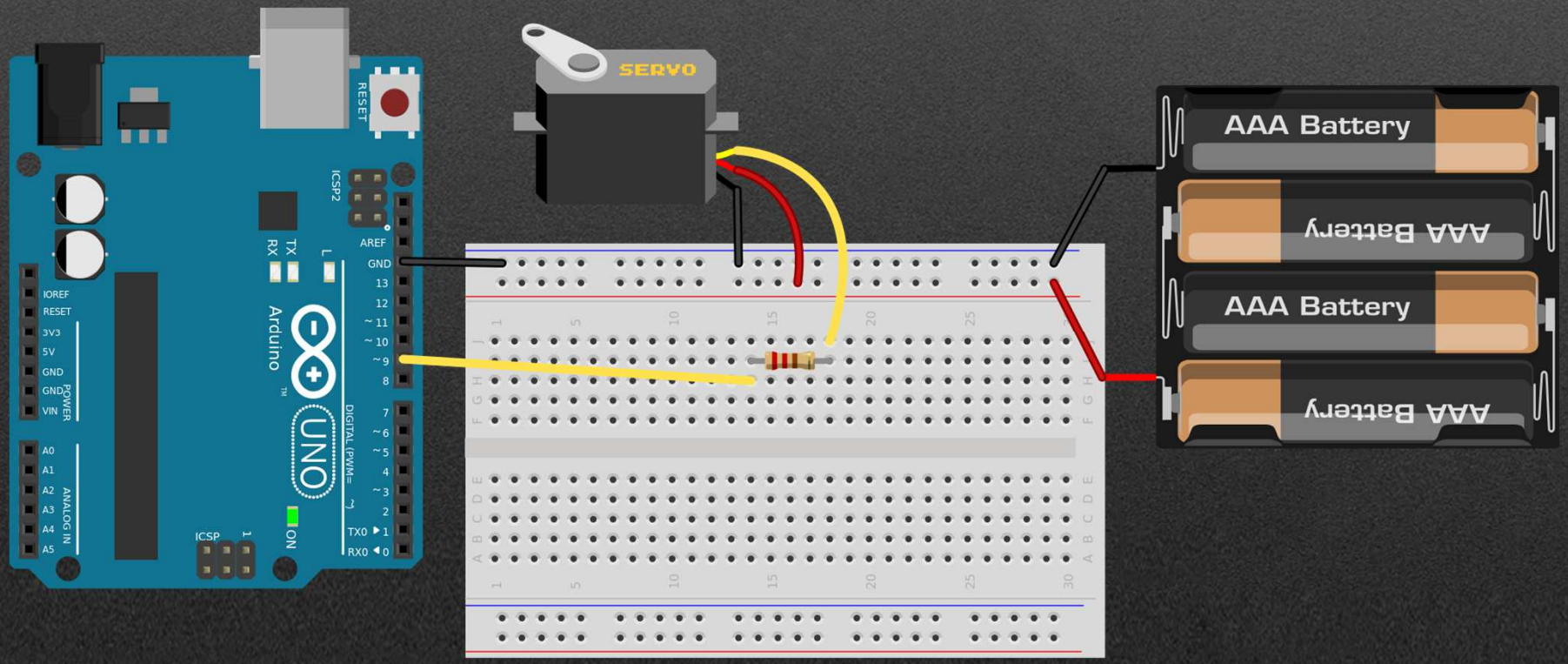
```ino
const int analogInPin = A0;
const int analogOutPin = 9;

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);
}
```

Servo


DELUXE
HITEC
HS-422

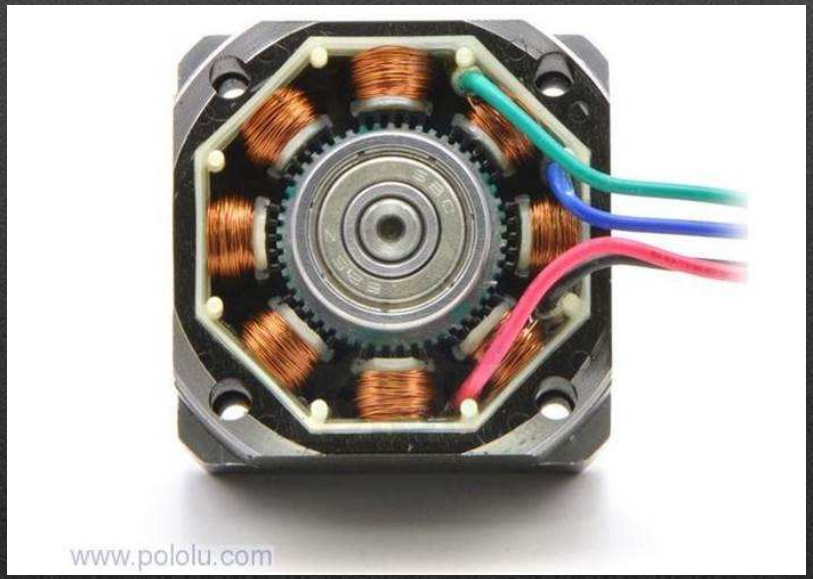

Inside the servo
Output Spline
Drive Gears
Servo Case
Control Circuit
Potentiometer
Motor
Motor
Gear Box
Output
Position Sensor
Error Amp
Pulse width to voltage converter
Control Pulse

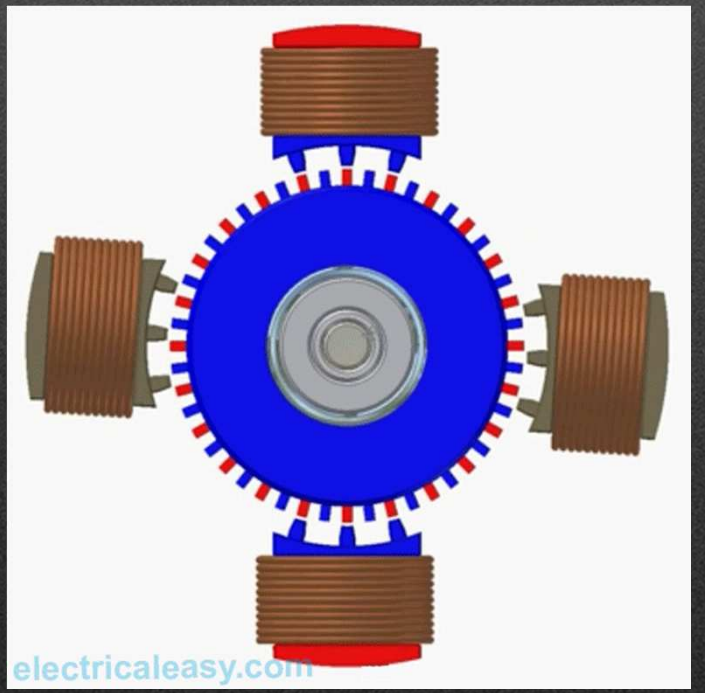
Circuito Servo



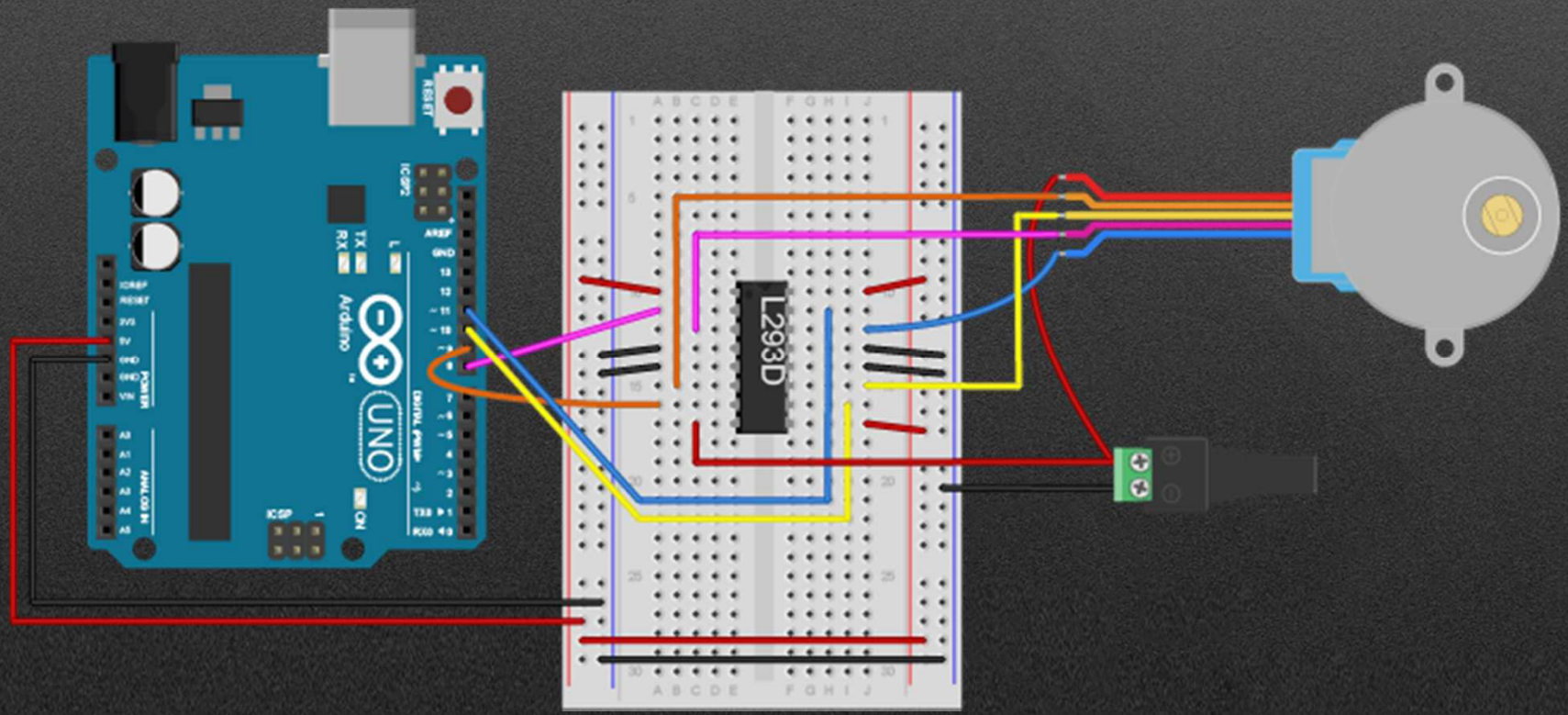
```
#include <Servo.h>

Servo myservo;

int pos = 0;

void setup() {
  myservo.attach(9);
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) {

    myservo.write(pos);
    delay(15);
  }
  for (pos = 180; pos >= 0; pos -= 1) {
    myservo.write(pos);
    delay(15);
  }
}
```

Circuito Stepper

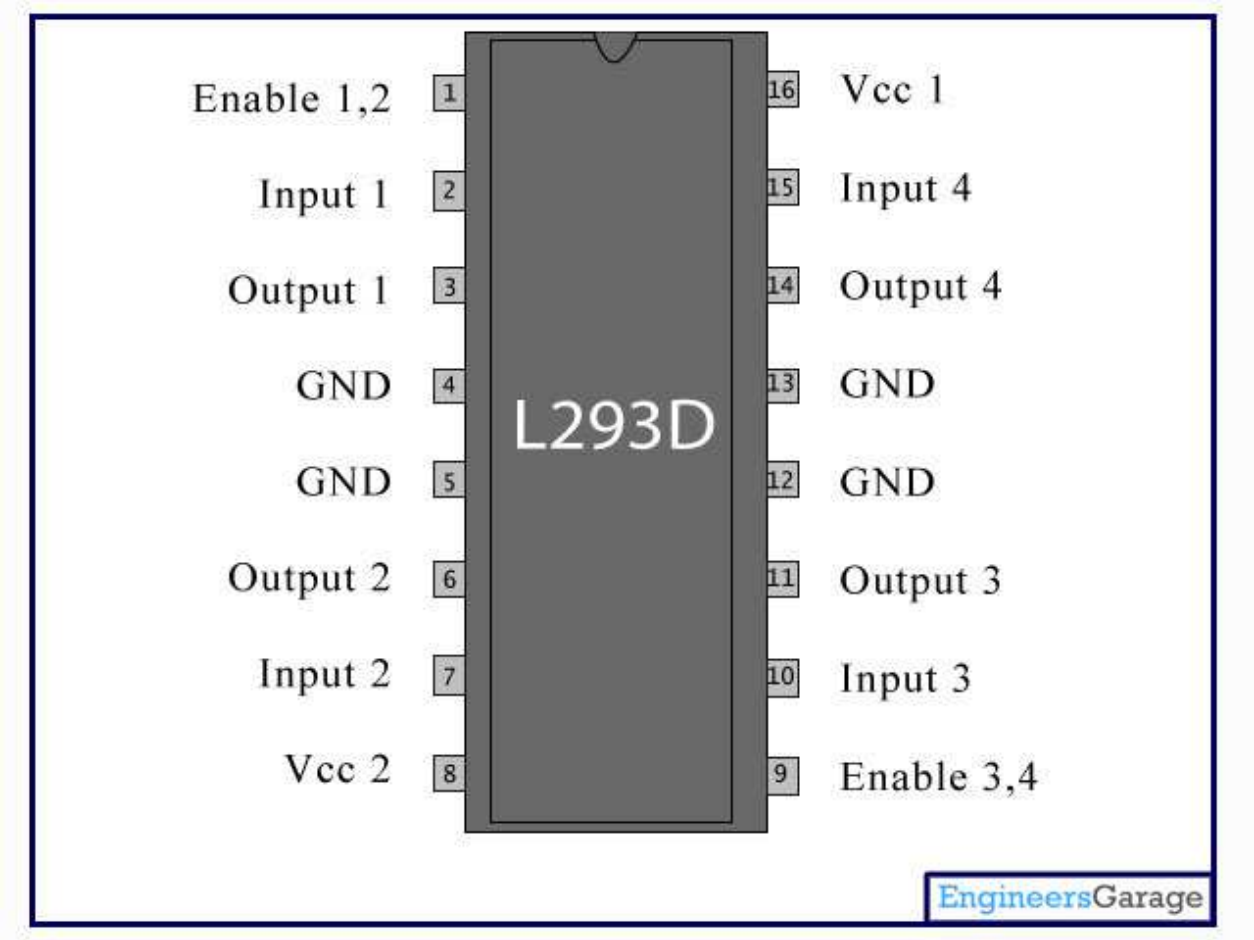




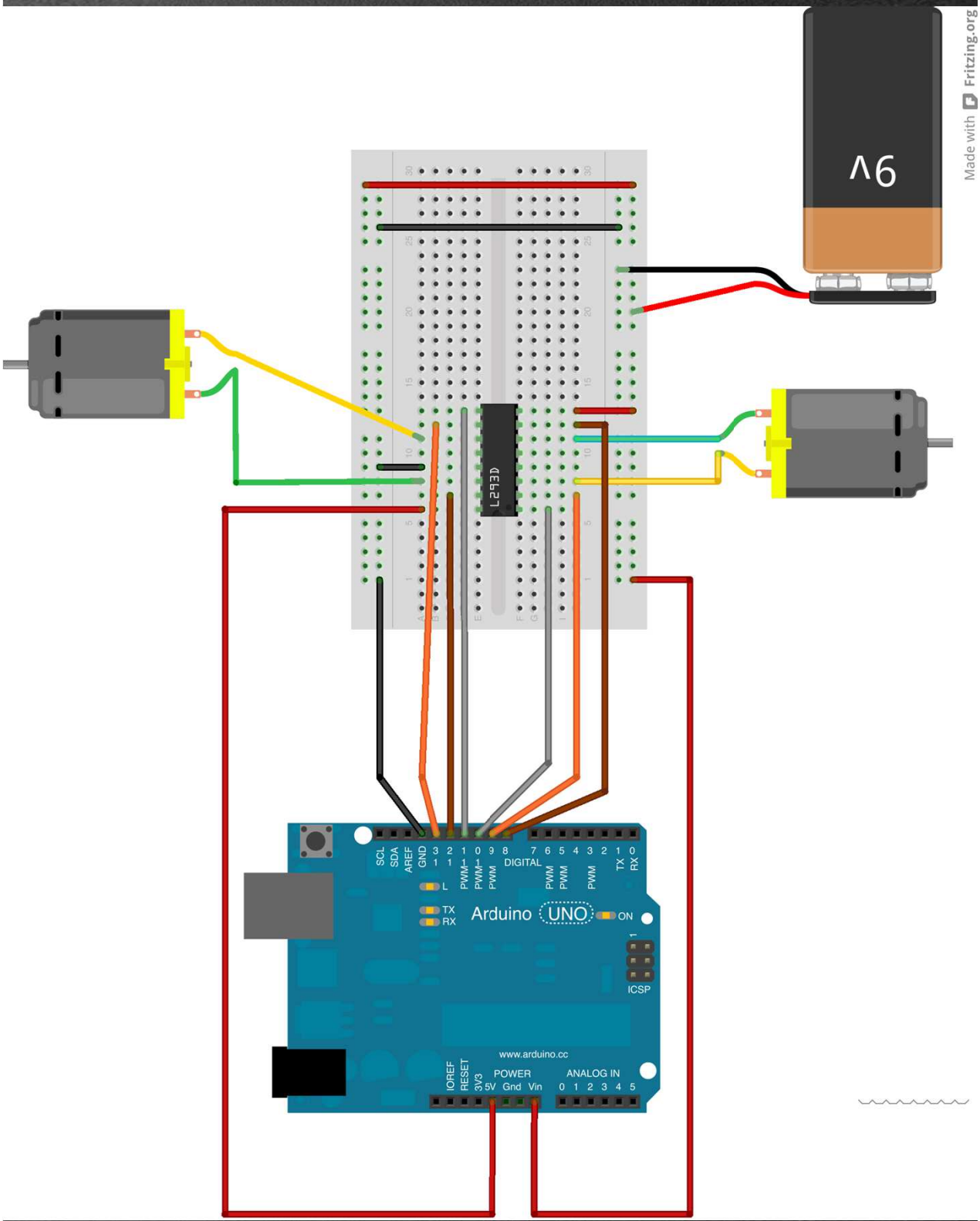
Circuito Stepper

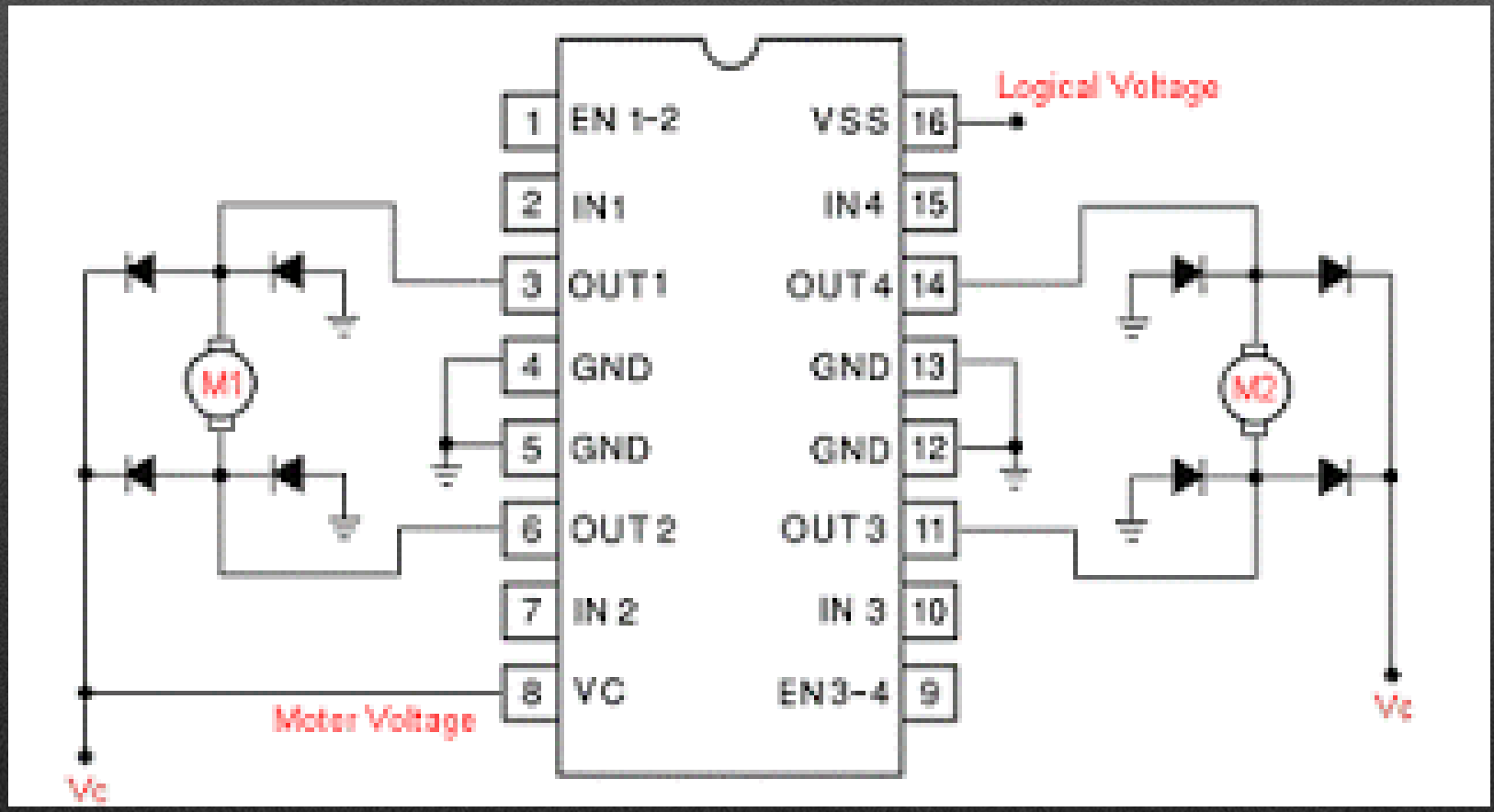


Controlador Stepper



```
#include <Stepper.h>

const int stepsPerRevolution = 200;

Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);

void setup() {
  myStepper.setSpeed(60);
  Serial.begin(9600);
}

void loop() {

  Serial.println("clockwise");
  myStepper.step(stepsPerRevolution);
  delay(500);

  Serial.println("counterclockwise");
  myStepper.step(-stepsPerRevolution);
  delay(500);
}
```


9V
L293D
SCL
SDA
AREF
GND
DIGITAL
PWM
TX
RX
L
ON
Arduino
UNO
ICSP
www.arduino.cc
POWER
ANALOG IN
IOREF
RESET
3V3
5V
Gnd
Vin
Made with Fritzing.org


Logical Voltage
1 EN 1-2
2 IN1
3 OUT1
4 GND
5 GND
6 OUT 2
7 IN 2
8 VC
VSS 16
IN4 15
OUT 4 14
GND 13
GND 12
OUT 3 11
IN 3 10
EN3-4 9
M1
M2
Motor Voltage
Vc
Vc

```
#define E1 10  // Enable Pin for motor 1
#define E2 11  // Enable Pin for motor 2

#define I1 8  // Control pin 1 for motor 1
#define I2 9  // Control pin 2 for motor 1
#define I3 12  // Control pin 1 for motor 2
#define I4 13  // Control pin 2 for motor 2

void setup() {

    pinMode(E1, OUTPUT);
    pinMode(E2, OUTPUT);

    pinMode(I1, OUTPUT);
    pinMode(I2, OUTPUT);
    pinMode(I3, OUTPUT);
    pinMode(I4, OUTPUT);
}
```

```
void loop() {

    analogWrite(E1, 153); // Run in half speed
    analogWrite(E2, 255); // Run in full speed

    digitalWrite(I1, HIGH);
    digitalWrite(I2, LOW);
    digitalWrite(I3, HIGH);
    digitalWrite(I4, LOW);

    delay(10000);

    // change direction

    digitalWrite(E1, LOW);
    digitalWrite(E2, LOW);

    delay(200);

    analogWrite(E1, 255);  // Run in full speed
    analogWrite(E2, 153);  // Run in half speed

    digitalWrite(I1, LOW);
    digitalWrite(I2, HIGH);
    digitalWrite(I3, LOW);
    digitalWrite(I4, HIGH);

    delay(10000);
}
```

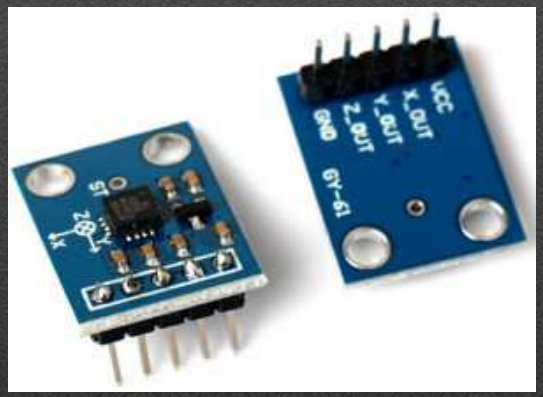
- https://www.youtube.com/watch?v=BBwEF6WBUQs

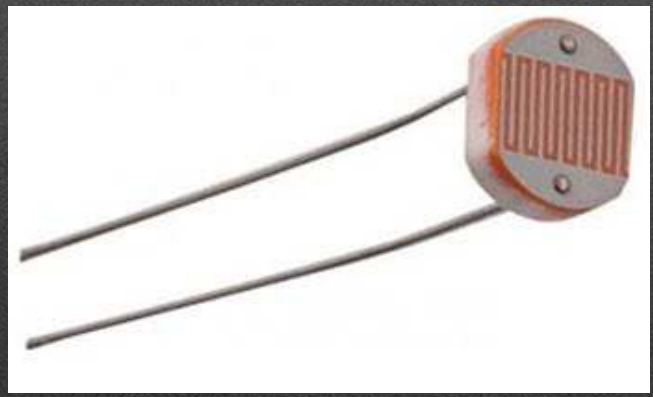
MOBILIDADE SUSTENTÁVEL

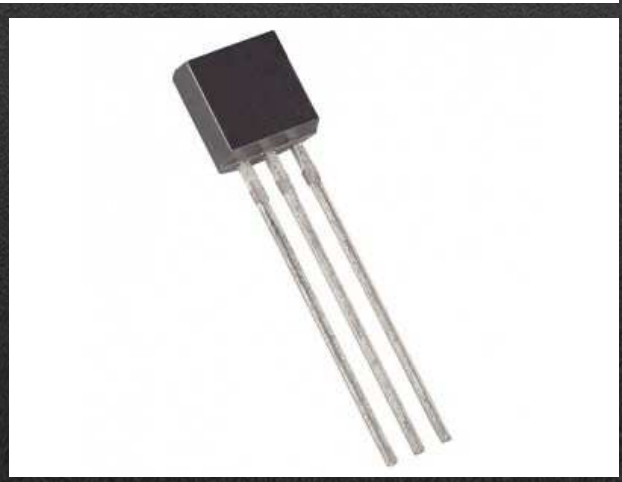
Luís Martins
Duarte Abreu

Arduino e Sensores

Sensores Analógicos



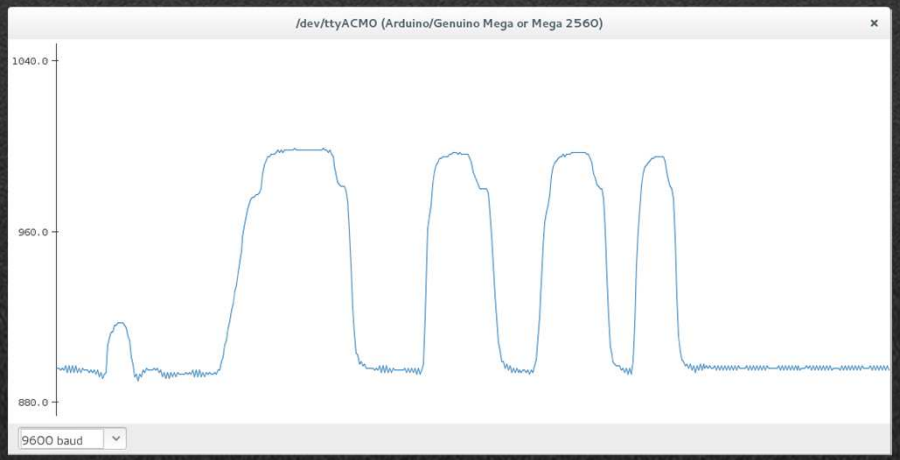




Analog Read

- analogRead(#pin)

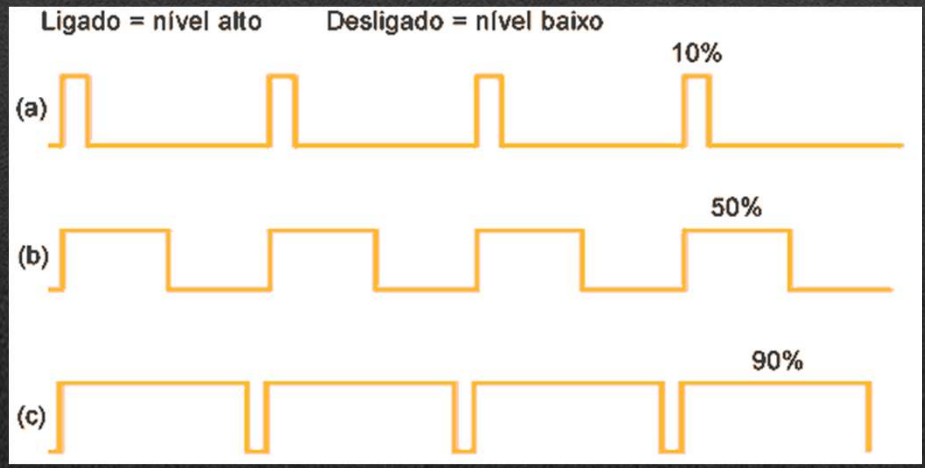
0-1023 (10-bit)



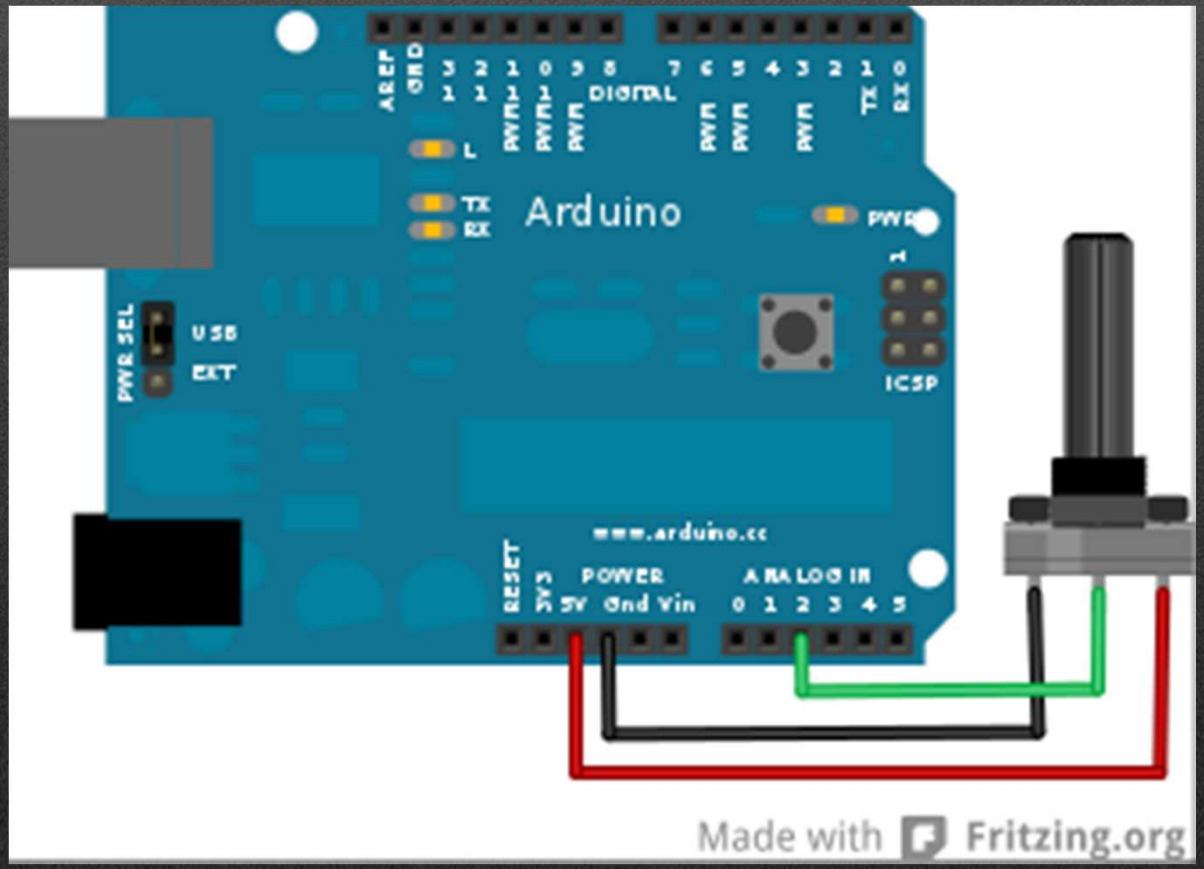
Digital Read

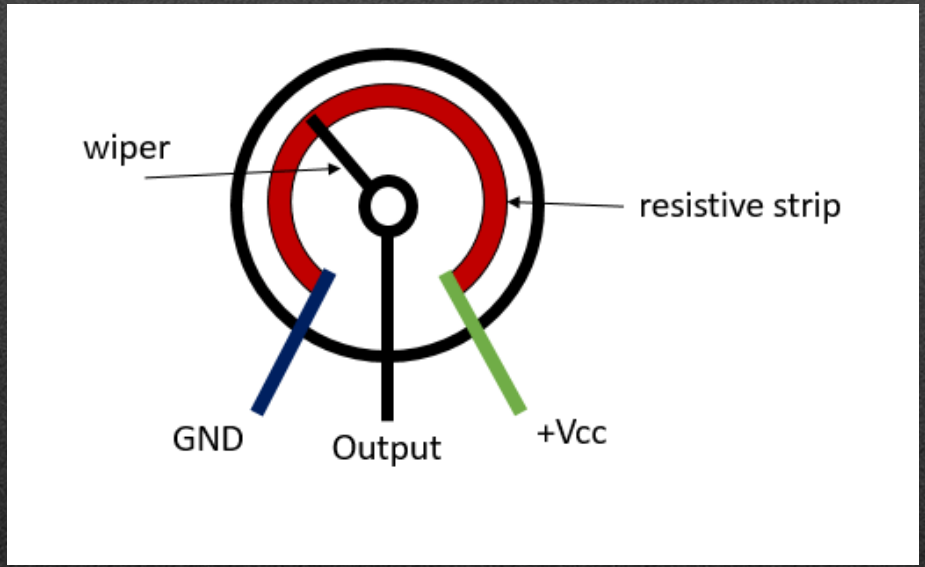
- pinMode(#pin, MODE)
- digitalRead(#pin)

0-255 (8-bit)



Potenciómetro





Desafio: Representar num gráfico a posição do potenciómetro entre 0 e 350

DICAS:

AnalogRead(#pin);

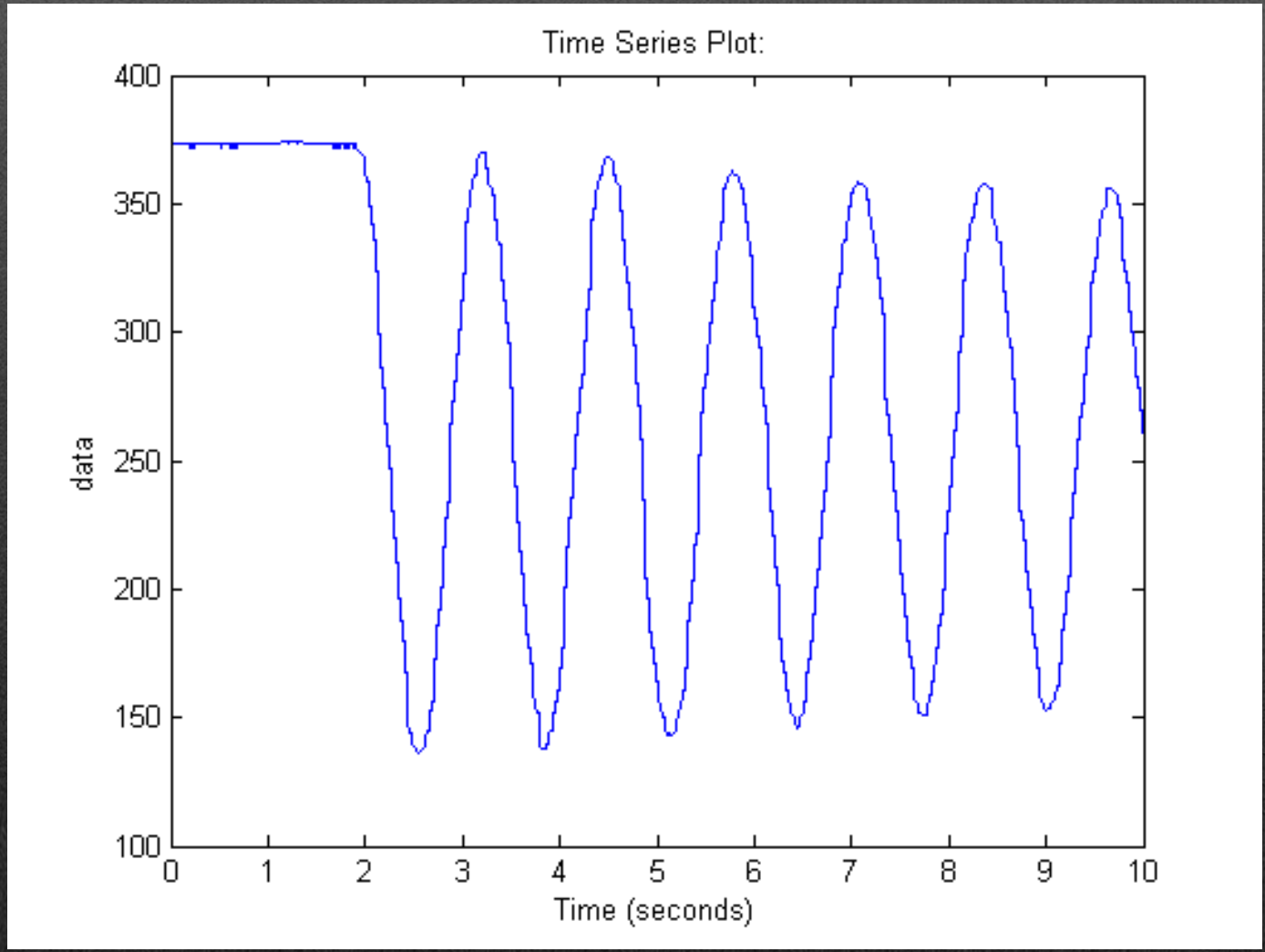
Ler a tensão no pin onde temos o potenciómetro

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

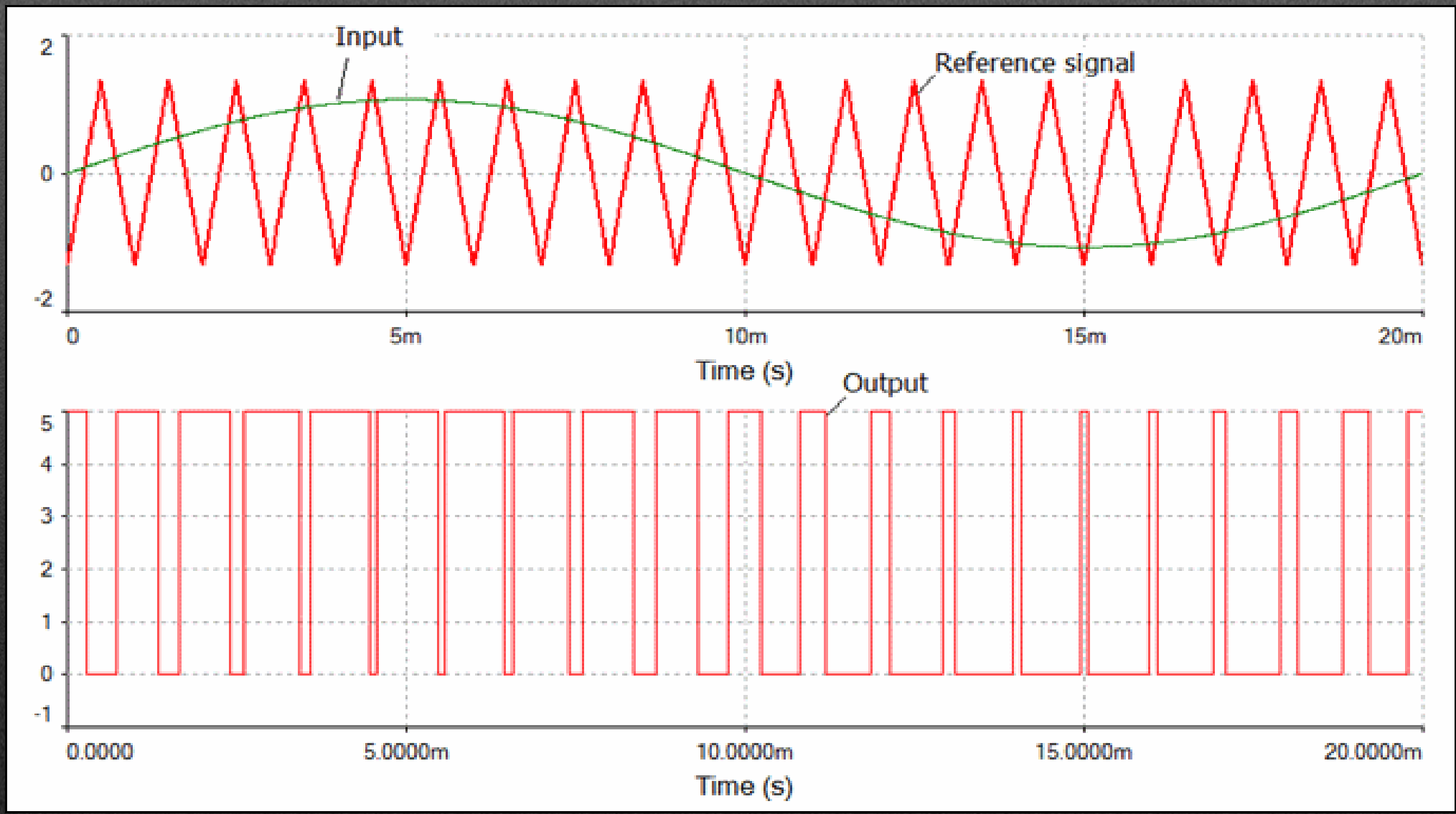
Esta linha faz a “regra de 3 simples” entre dois conjuntos de valores.

Serial.print(“Pos=”); Serial.println(valor);

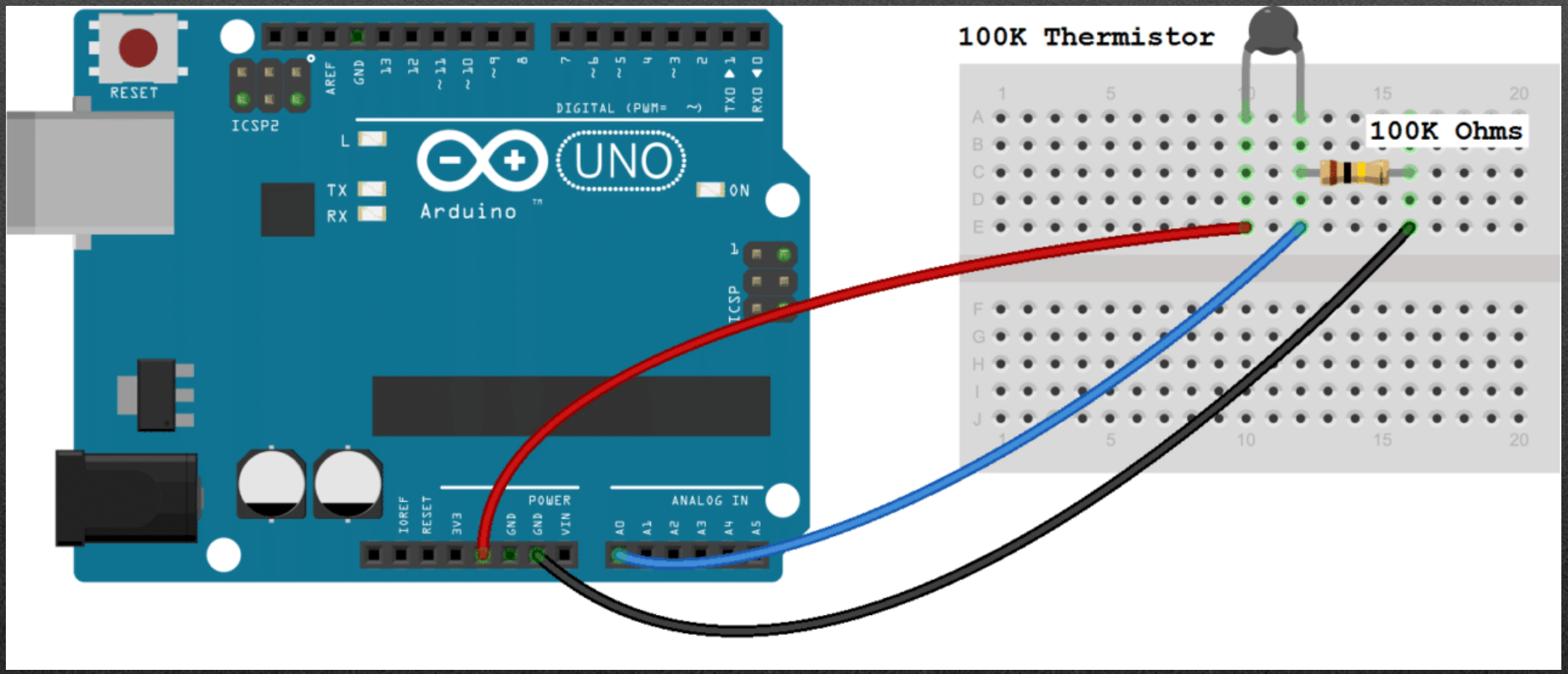
dar o resultado no computador


Time Series Plot:
400
350
300
250
200
150
100
data
0
1
2
3
4
5
6
7
8
9
10
Time (seconds)

```
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);
}
```

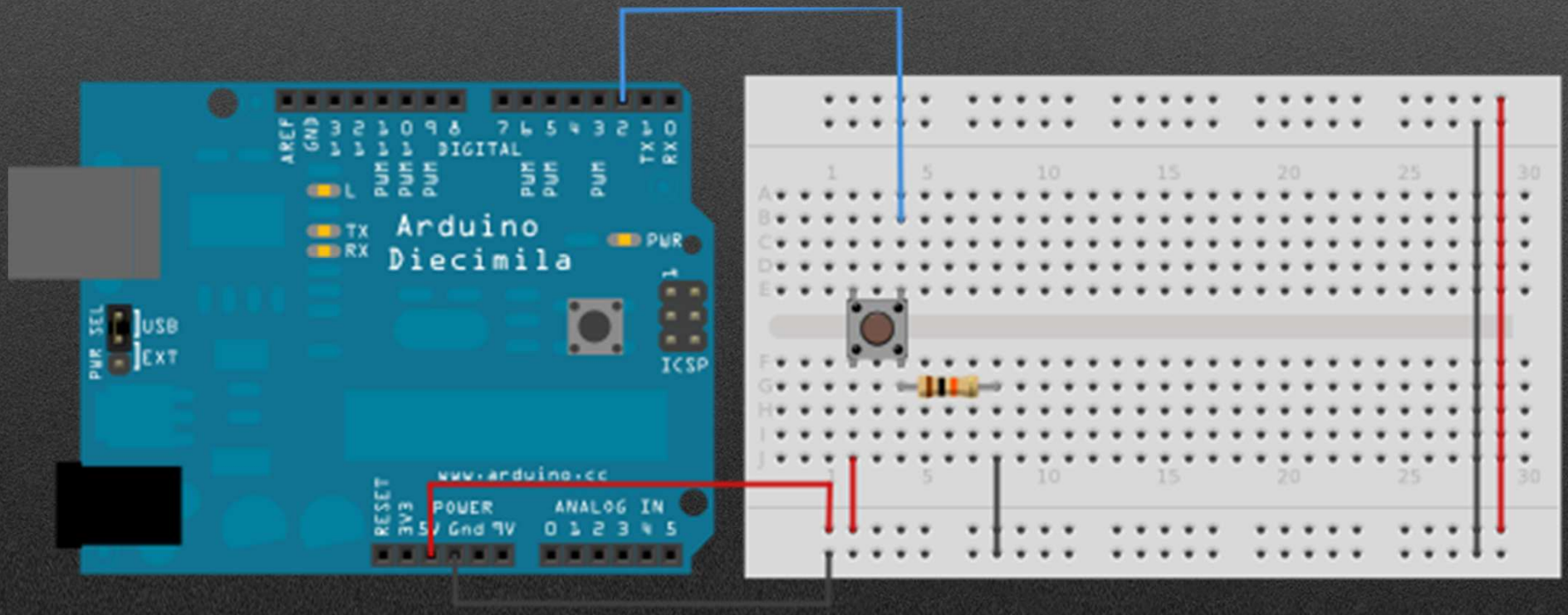

Input
Reference signal
2
0
-2
0
5m
10m
15m
20m
Time (s)
Output
5
4
3
2
1
0
-1
0.0000
5.0000m
10.0000m
15.0000m
20.0000m
Time (s)

Outros exemplos



R1
10kΩ
UNO
Arduino
ANALOG IN
POWER
DIGITAL (PWM~)
ICSP
ICSP2
RESET
ON
L
TX
RX

Sensores Digitais Simples



Desafio: Representar na consola o estado do botão

DICAS:

pinMode(#pin, MODE);

Identificar o pin

Input=DigitalRead(#pin);

Der o valor digital (0 ou 1)

If(input==HIGH){

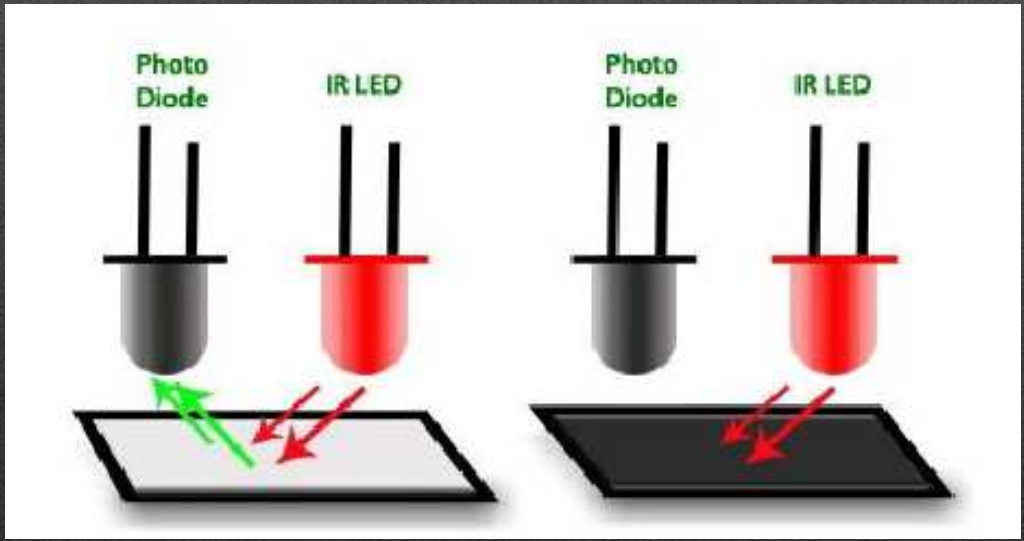
Serial.println(“Botão premido”); }

Else{

Serial.println(“Botão não premido”); }

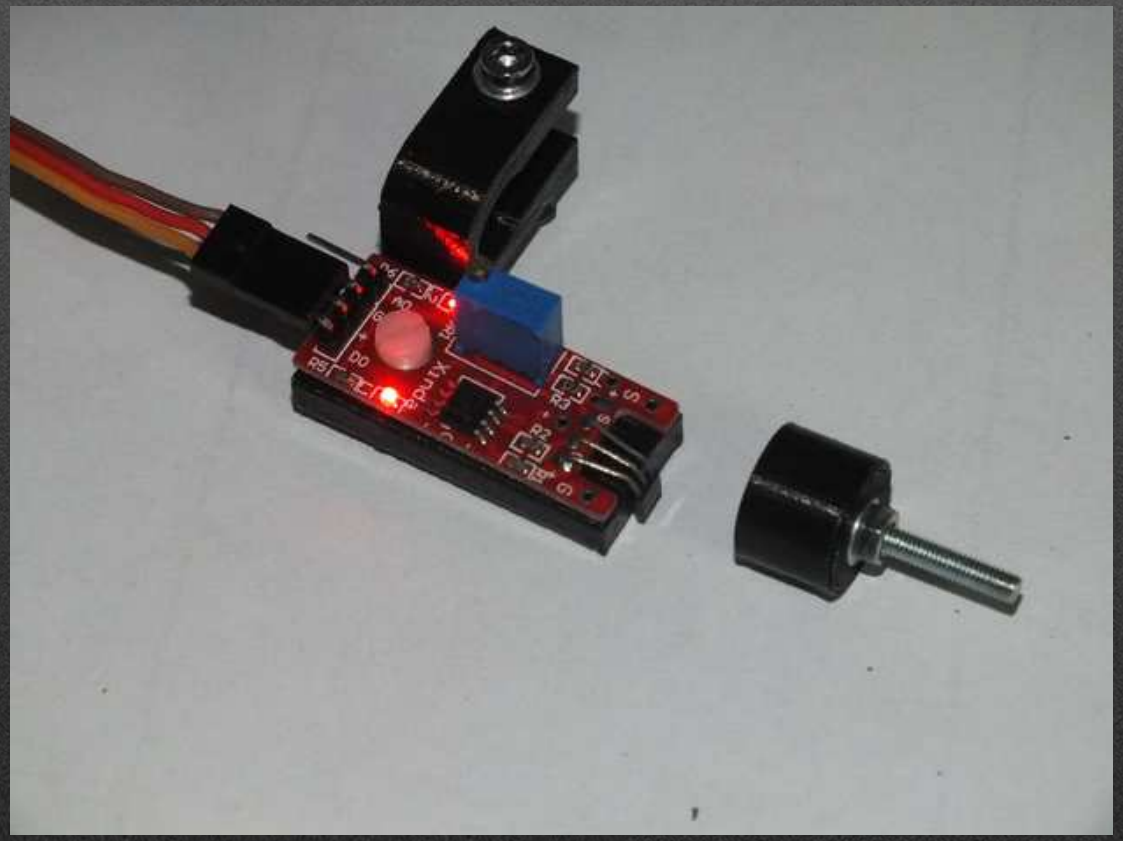
```
int pushButton = 2;
void setup() {
  Serial.begin(9600);
  pinMode(pushButton, INPUT);
}

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

  delay(1);
}
```

Exemplos

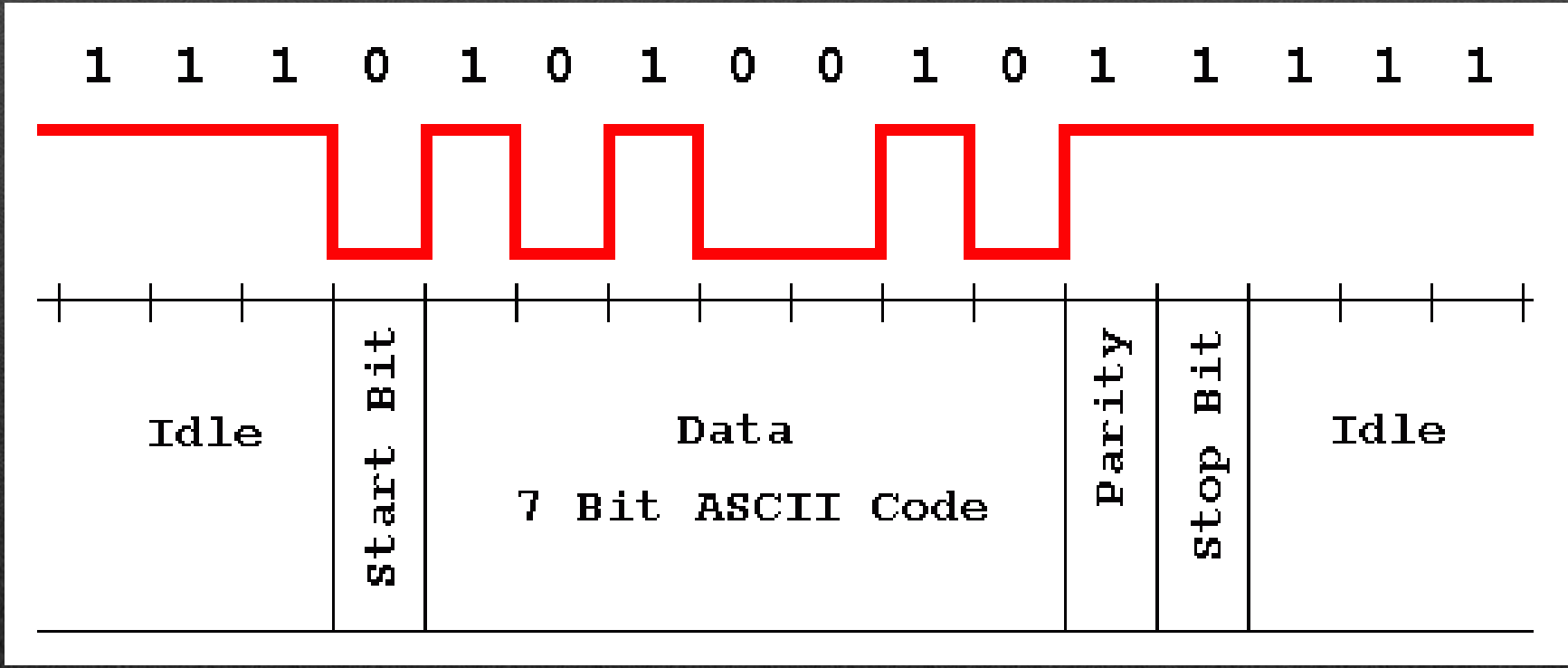


0 1

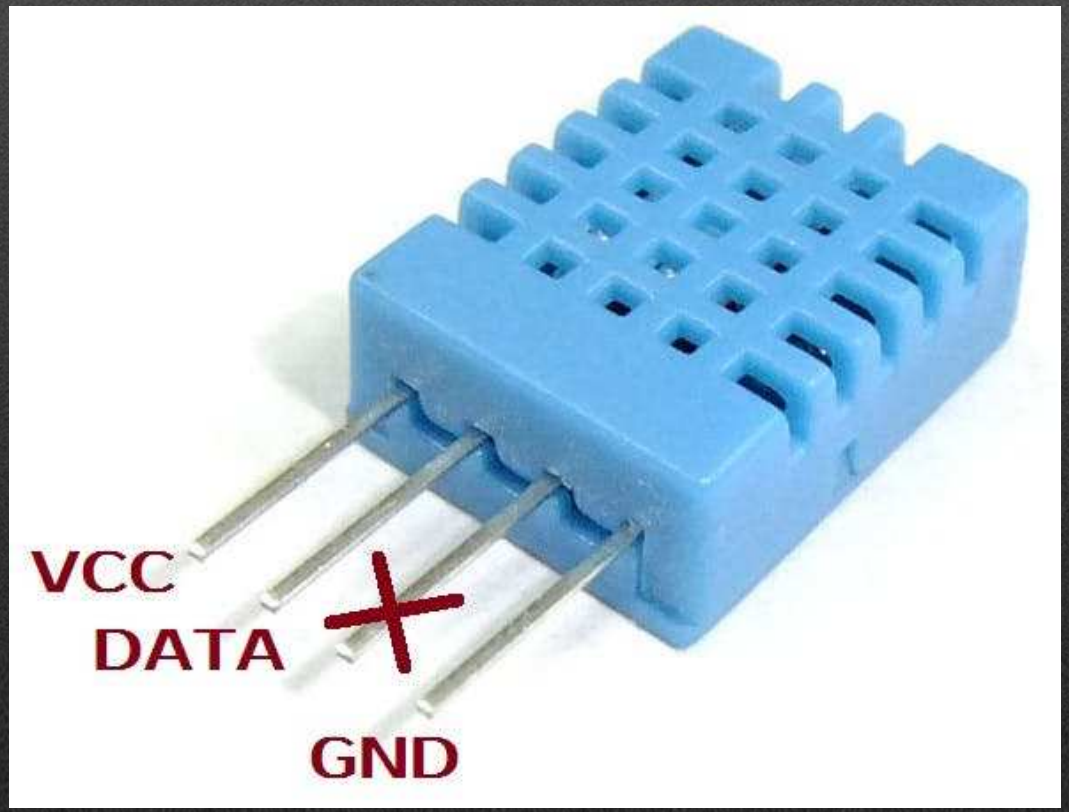

49E
6868G

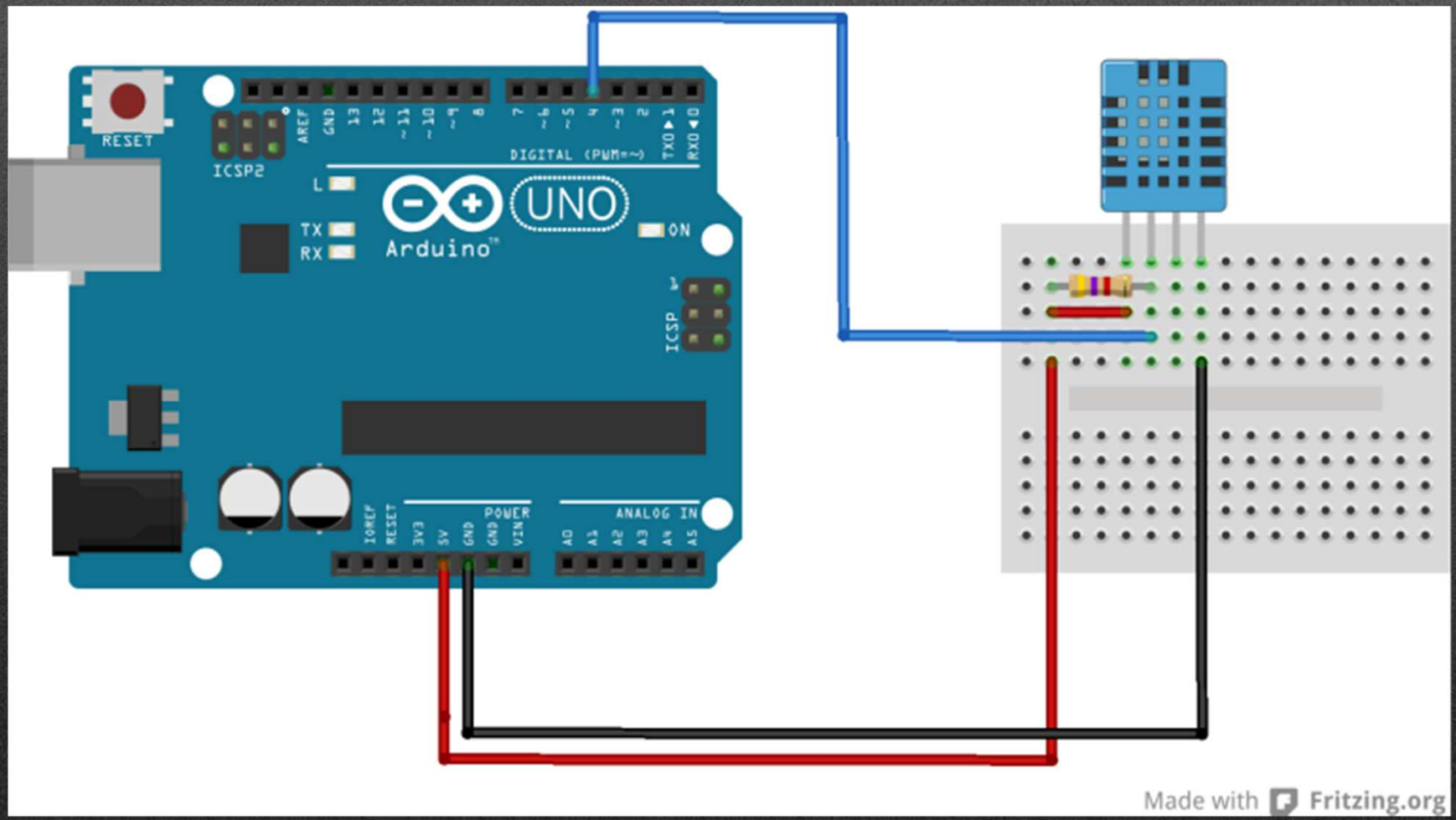


Sensores digitais complexos



DHT11



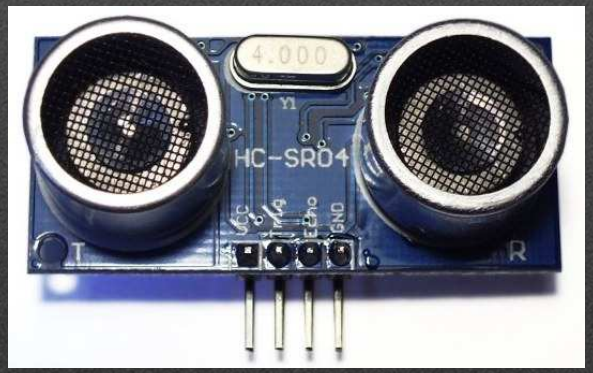

RESET
ICSP2
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM=~)
L
TX
RX
Arduino
UNO
ON
ICSP
POWER
ANALOG IN
IOREF
RESET
3V3
5V
GND
GND
VIN
A0
A1
A2
A3
A4
A5
Made with Fritzing.org

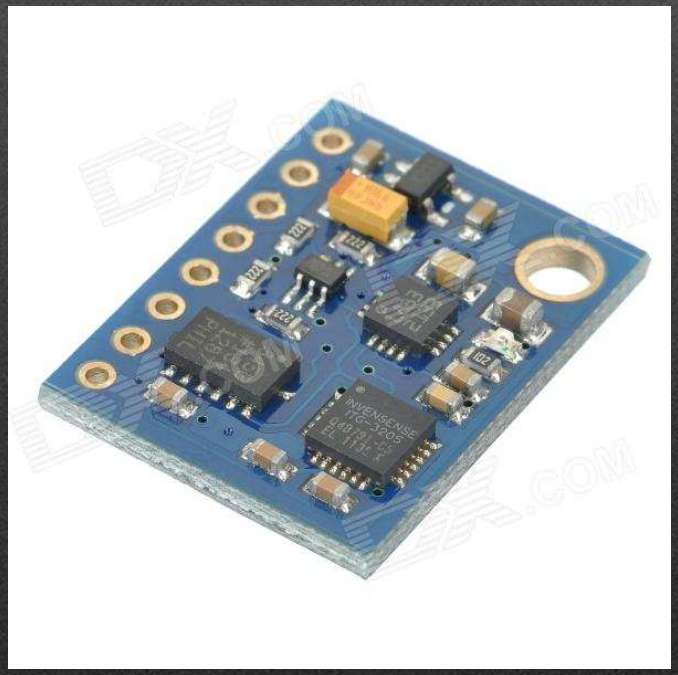
```
#include "DHT.h"
#define DHTPIN 2

#define DHTTYPE DHT11   // DHT 11

DHT dht(DHTPIN, DHTTYPE);

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

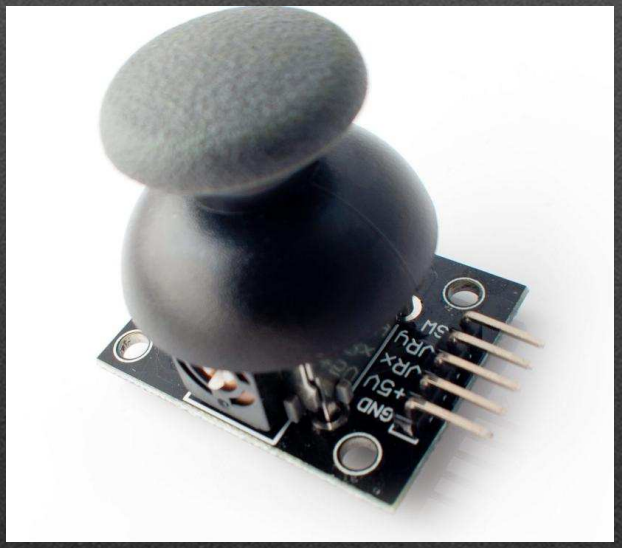
```
void loop() {
  delay(2000);
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float f = dht.readTemperature(true);
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Erro ao ler o sensor");
    return;
  }
```

```
  Serial.print("Humidade: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperatura: ");
  Serial.print(t);
  Serial.print(" *C ");
  Serial.print(f);
  Serial.print(" *F\t");
}
```

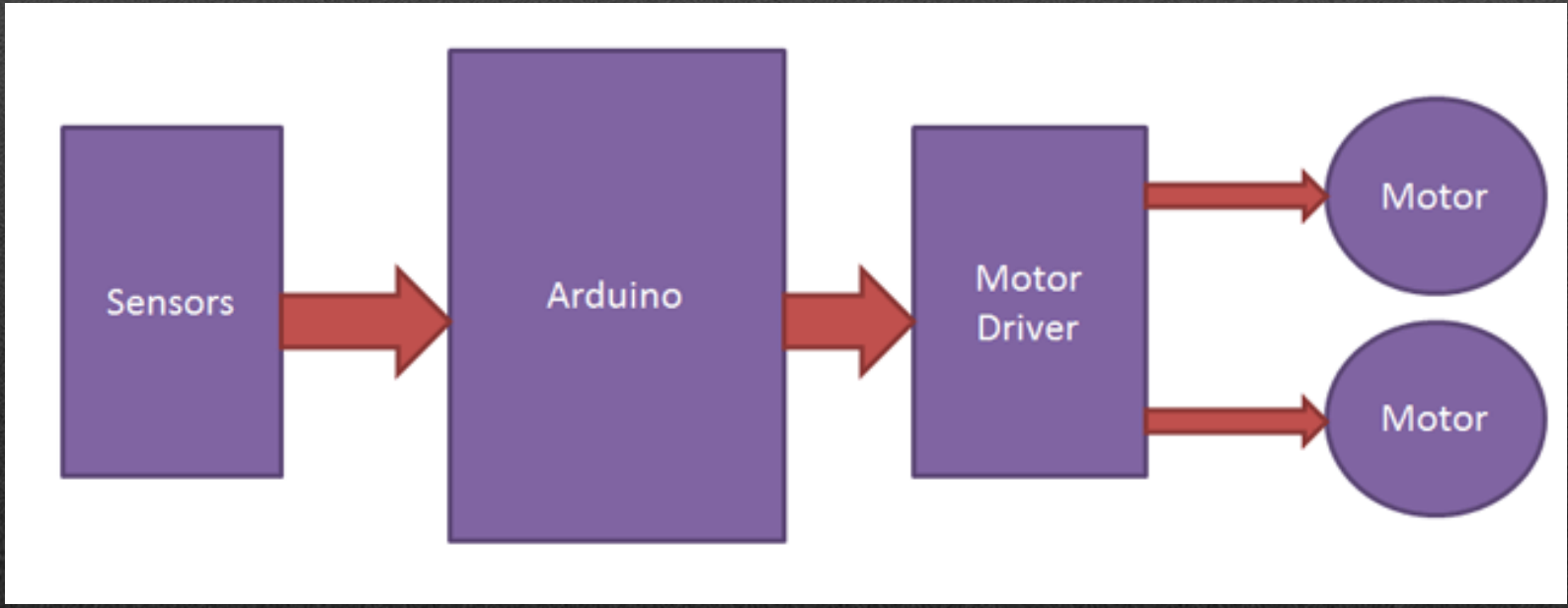
Exemplos



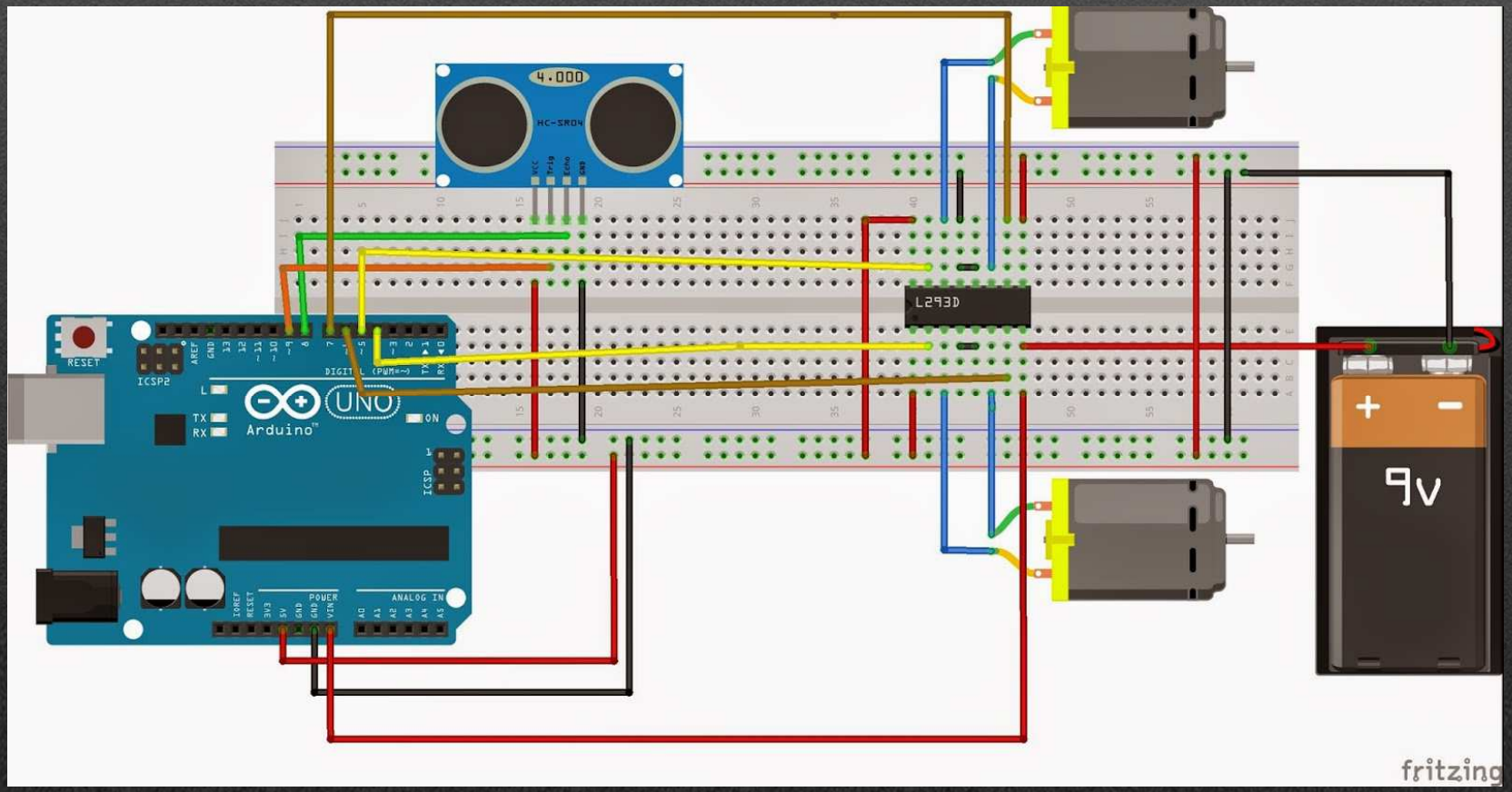




Carro autónomo



Exemplo carro autónomo:

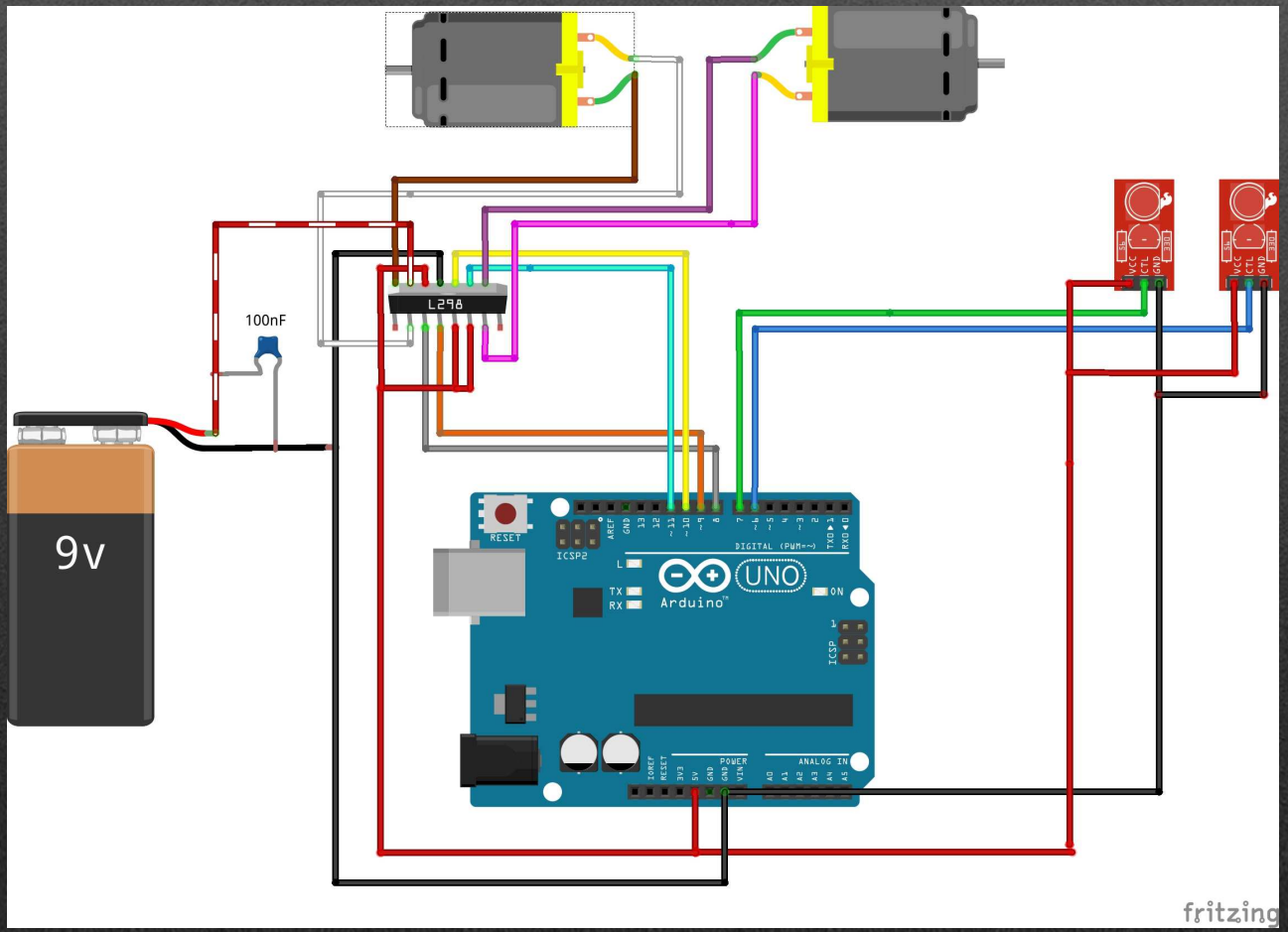


```
#include <Servo.h> //include Servo library

const int RForward = 0;
const int RBackward = 180;
const int LForward = RBackward;
const int LBackward = RForward;
const int RNeutral = 90;
const int LNeutral = 90; //constants for motor speed
const int pingPin = 7;
const int irPin = 0;  //Sharp infrared sensor pin
const int dangerThresh = 10; //threshold for obstacles (in cm)
int leftDistance, rightDistance; //distances on either side
Servo panMotor;
Servo leftMotor;
Servo rightMotor; //declare motors
long duration; //time it takes to recieve PING))) signal
```

```
void compareDistance()
{
  if (leftDistance>rightDistance) //if left is less obstructed
  {
    leftMotor.write(LBackward);
    rightMotor.write(RForward); //turn left
    delay(500);
  }
  else if (rightDistance>leftDistance) //if right is less obstructed
  {
    leftMotor.write(LForward);
    rightMotor.write(RBackward); //turn right
    delay(500);
  }
   else //if they are equally obstructed
  {
    leftMotor.write(LForward);
    rightMotor.write(RBackward); //turn 180 degrees
    delay(1000);
  }
}
```

```
long ping()
{
  // Send out PING))) signal pulse
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(pingPin, LOW);

  //Get duration it takes to receive echo
  pinMode(pingPin, INPUT);
  duration = pulseIn(pingPin, HIGH);

  //Convert duration into distance
  return duration / 29 / 2;
}
```

```
void loop()
{
  int distanceFwd = ping();
  if (distanceFwd>dangerThresh) //if path is clear
  {
    leftMotor.write(LForward);
    rightMotor.write(RForward); //move forward
  }
  else //if path is blocked
  {
    leftMotor.write(LNeutral);
    rightMotor.write(RNeutral);
    panMotor.write(0);
    delay(500);
    rightDistance = ping(); //scan to the right
    delay(500);
    panMotor.write(180);
    delay(700);
    leftDistance = ping(); //scan to the left
    delay(500);
    panMotor.write(90); //return to center
    delay(100);
    compareDistance();
  }
}
```


L298
100nF
9v
RESET
ICSP2
UNO
Arduino
TX
RX
ON
ICSP
DIGITAL (PWM~)
POWER
ANALOG IN
fritzing

```
#define LS 2       // left sensor
#define RS 3       // right sensor

#define LM1 4        // left motor
#define LM2 5        // left motor
#define RM1 6        // right motor
#define RM2 7        // right motor

void setup()
{
  pinMode(LS, INPUT);
  pinMode(RS, INPUT);
  pinMode(LM1, OUTPUT);
  pinMode(LM2, OUTPUT);
  pinMode(RM1, OUTPUT);
  pinMode(RM2, OUTPUT);
}
```

```
  if(!(digitalRead(LS)) && !(digitalRead(RS)))     // stop
  {
    digitalWrite(LM1, LOW);
    digitalWrite(LM2, LOW);
    digitalWrite(RM1, LOW);
    digitalWrite(RM2, LOW);
  }
}
```

```
void loop()
{
  if(digitalRead(LS) && digitalRead(RS))     // Move Forward
  {
    digitalWrite(LM1, HIGH);
    digitalWrite(LM2, LOW);
    digitalWrite(RM1, HIGH);
    digitalWrite(RM2, LOW);
  }

  if(!(digitalRead(LS)) && digitalRead(RS))     // Turn right
  {
    digitalWrite(LM1, LOW);
    digitalWrite(LM2, LOW);
    digitalWrite(RM1, HIGH);
    digitalWrite(RM2, LOW);
  }

  if(digitalRead(LS) && !(digitalRead(RS)))     // turn left
  {
    digitalWrite(LM1, HIGH);
    digitalWrite(LM2, LOW);
    digitalWrite(RM1, LOW);
    digitalWrite(RM2, LOW);
  }
```

Obrigado!