

*SAE2019*

*Semana Acadêmica das Engenharias  
ULBRA Canoas 2019*

# Simulação de circuitos com Tinkercad

Felipe Kühne  
[fkuhne@gmail.com](mailto:fkuhne@gmail.com)

Maio/2019

[github.com/fkuhne/sae2019](https://github.com/fkuhne/sae2019)



[https://youtu.be/WynYhZx\\_qds](https://youtu.be/WynYhZx_qds)

(1) Dicas da Tchellita - Tinkercad x

https://www.youtube.com/watch?v=WynYhZx\_qds&t=630s

YouTube Search

**DICAS DA TCHELLITA**

**COMO "BRINCAR" COM ARDUINO ANTES DE COMPRAR A PLACA E OS COMPONENTES ELETRÔNICOS ?**

0:01 / 14:11

Dicas da Tchellita - Tinkercad | Marcela Santos

22 views

3 0 SHARE SAVE

Marcela Santos  
Published on Mar 13, 2019

SUBSCRIBED 116

Tinkercad é um ambiente de simulação pra você que gostaria de aprender conceitos de

Up next


AUTOPLAY

SolidWorks Tutorial #181: Cola Bottle  
SolidWorks Tutorial 4.2M views



27:34


<https://www.tinkercad.com>

Gallery of Things | Tinkercad x +

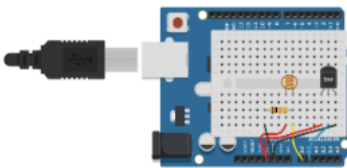
← → ↻ [https://www.tinkercad.com/things?type=circuits&view\\_mode=default](https://www.tinkercad.com/things?type=circuits&view_mode=default) ☆ S | 




**TINKERCAD** AUTODESK® TINKERCAD™

Galeria Blog Aprenda Ensinar  

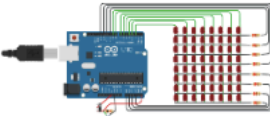
**Galeria** **Desenhos** **Circuits** ★ Favoritos da equipe ▾  Pequeno




★



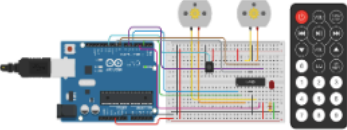
 **Arduino Datalogger - simple**  
(há 17 dias) jmascorella@loreto.nsw.ed...  6  0




★



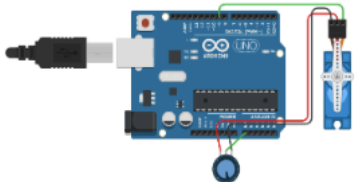
 **multiplexing**  
(há 2 meses) simmen  111  10




★



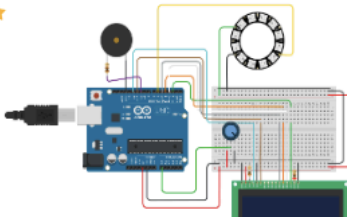
 **Arduino RC Robot**  
(há 3 meses) Helektrika  151  11




★



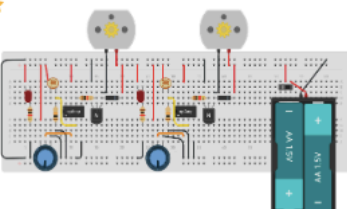
 **Movimiento servo**  
(há 3 meses) #aprendeConDynamo  71  6




★



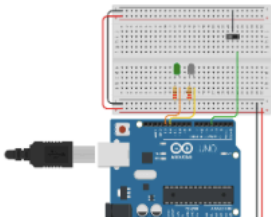
 **Arduino circuit diagram with potentiometer**  
(há 1 dia) jmascorella@loreto.nsw.ed...  6  0




★



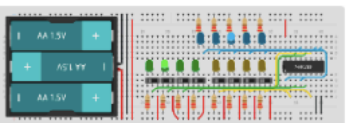
 **Arduino circuit diagram with two servos**  
(há 1 dia) jmascorella@loreto.nsw.ed...  6  0




★



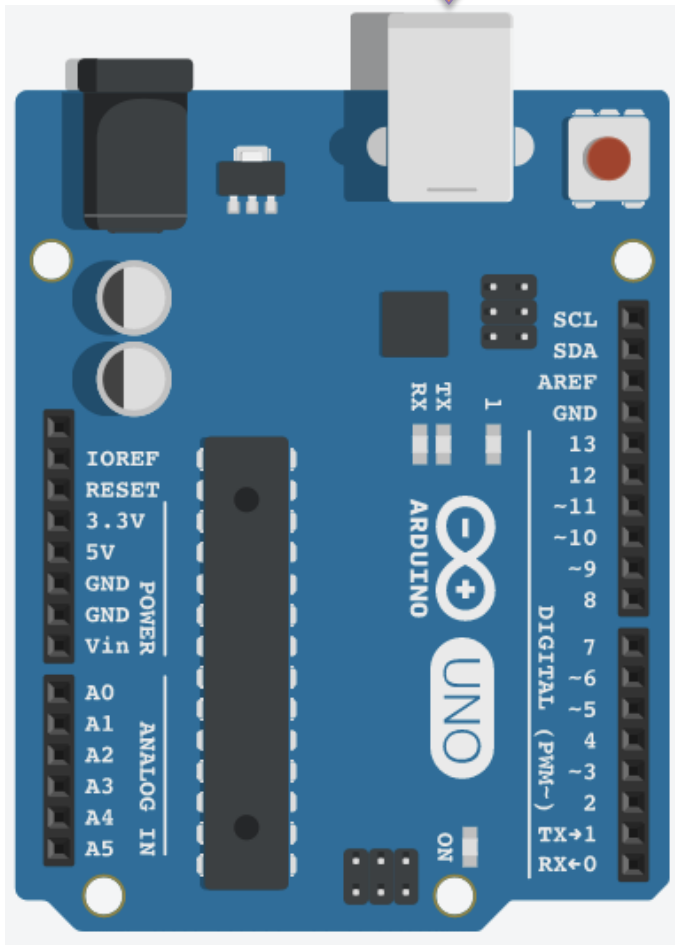
 **Arduino circuit diagram with breadboard**  
(há 1 dia) jmascorella@loreto.nsw.ed...  6  0

★



 **Arduino circuit diagram with three AA batteries**  
(há 1 dia) jmascorella@loreto.nsw.ed...  6  0

USB PC



# Nosso cenário



```
Blink | Arduino 1.6.8

/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * Most Arduinos have an on-board LED you can control. On the Uno and Leonardo, it is attached to digital pin 13. If you're unsure what pin the on-board LED is connected to on your Arduino model, check the documentation at http://www.arduino.cc
 *
 * This example code is in the public domain.
 *
 * modified 8 May 2014
 * by Scott Fitzgerald
 */

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(13, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

chipKIT DP32 on /dev/cu.usbmodem1411





fkuhne

Desenhos

Circuits

Lições

Projetos

+ Criar projeto

Tweets

Seguir



Tinkercad @tinkercad

Enter the Design for Space Travel challenge to create an inventive spacecraft using @tinkercad !

## Circuits

Create new Circuit



**LCD e sensor ultrassônico**  
há 2 minutos  
Público



**Sensor ultrassônico**  
há 21 minutos  
Público



**Entrada e saída analógicas**  
há 44 minutos  
Público



**Potenciômetro e servo**  
há uma hora  
Público



**Entrada analógica**  
há 2 horas



**Blink Analógico**  
há 2 horas



**Botão e led**  
há 3 horas

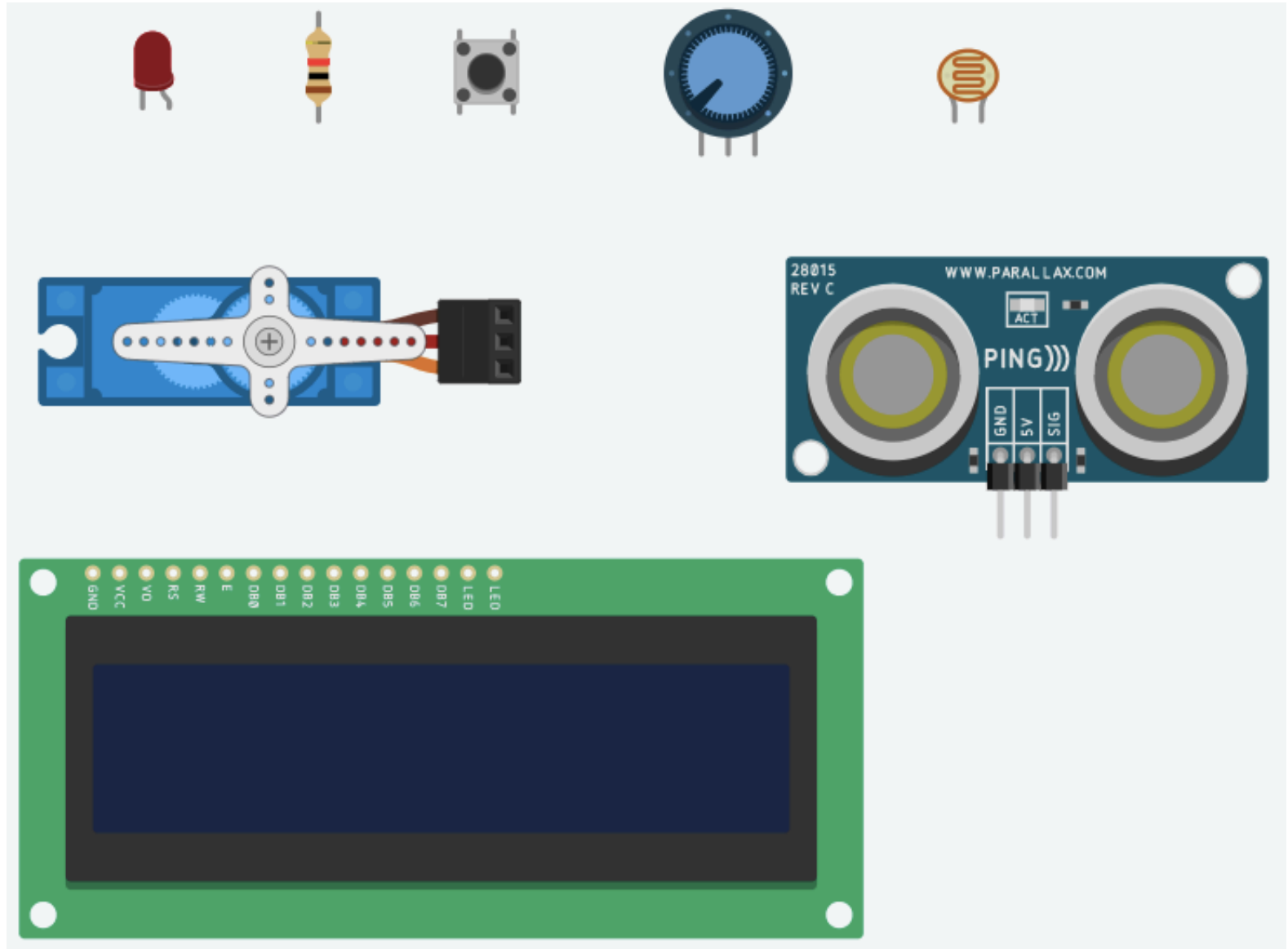


**Blink**  
há 3 horas

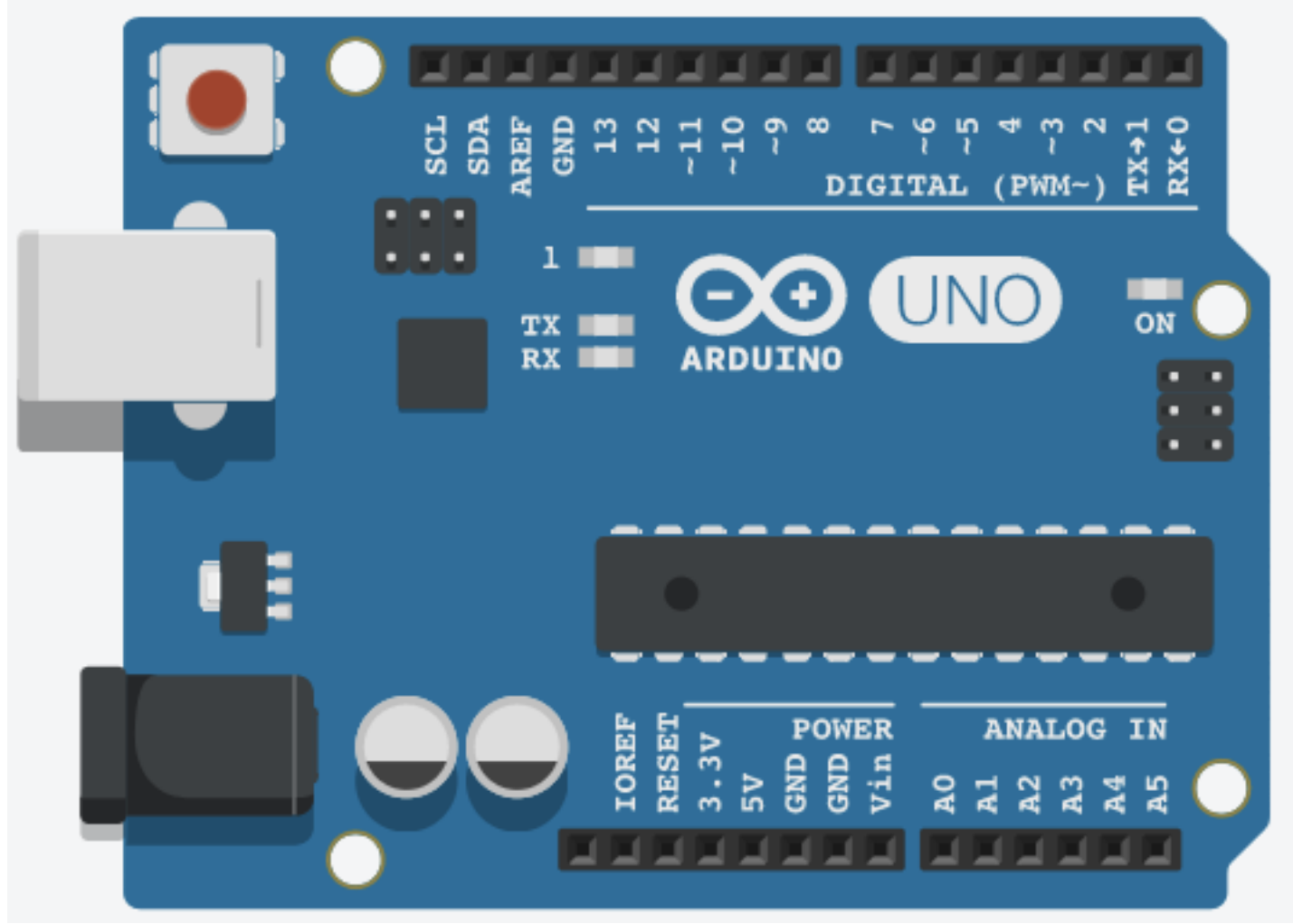
# Funções básicas

- pinMode()
- digitalWrite()
- digitalRead()
- delay()
- analogWrite()
- analogRead()
- Classe Serial
- Outras libs...

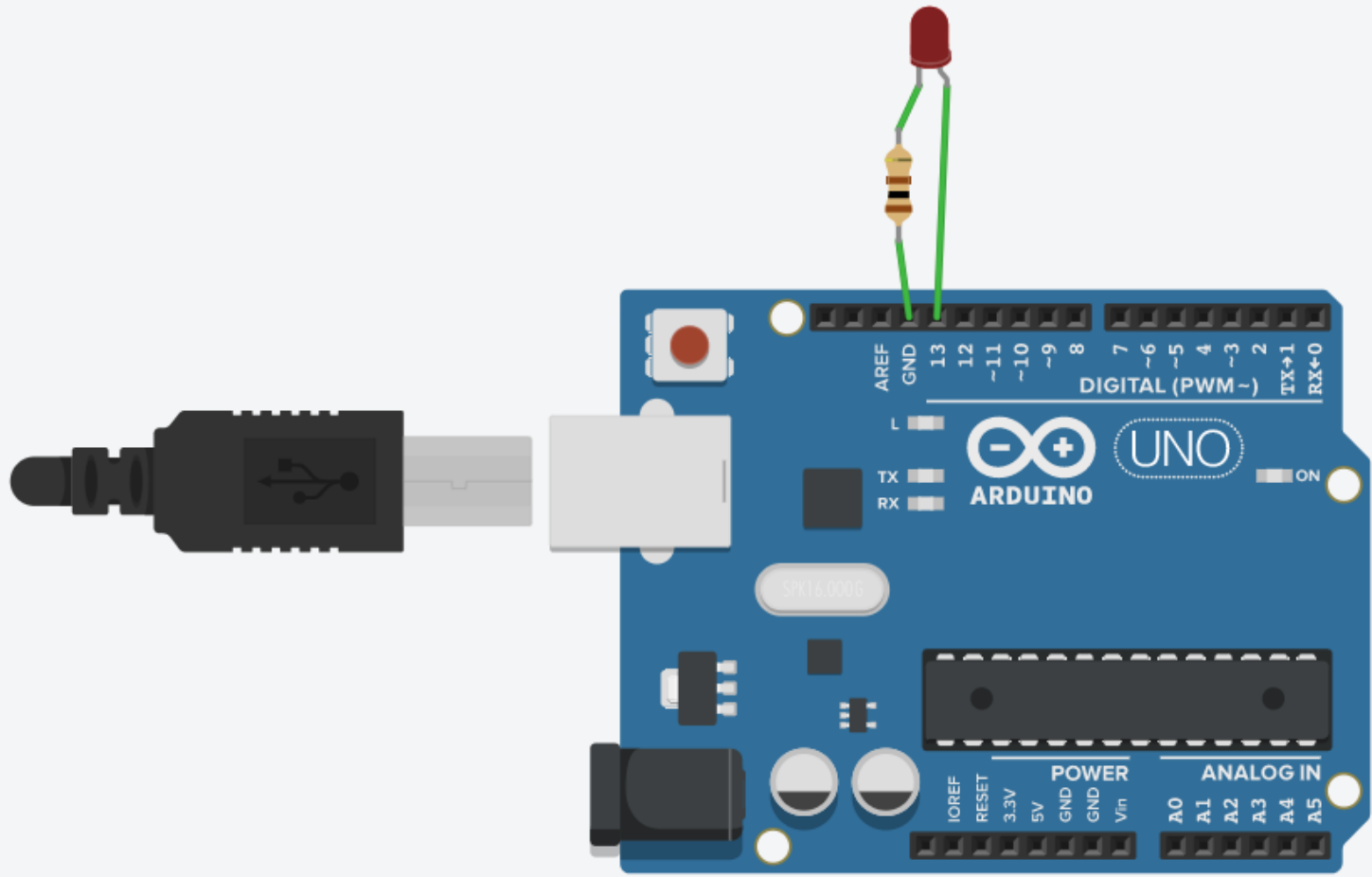
# Componentes básicos







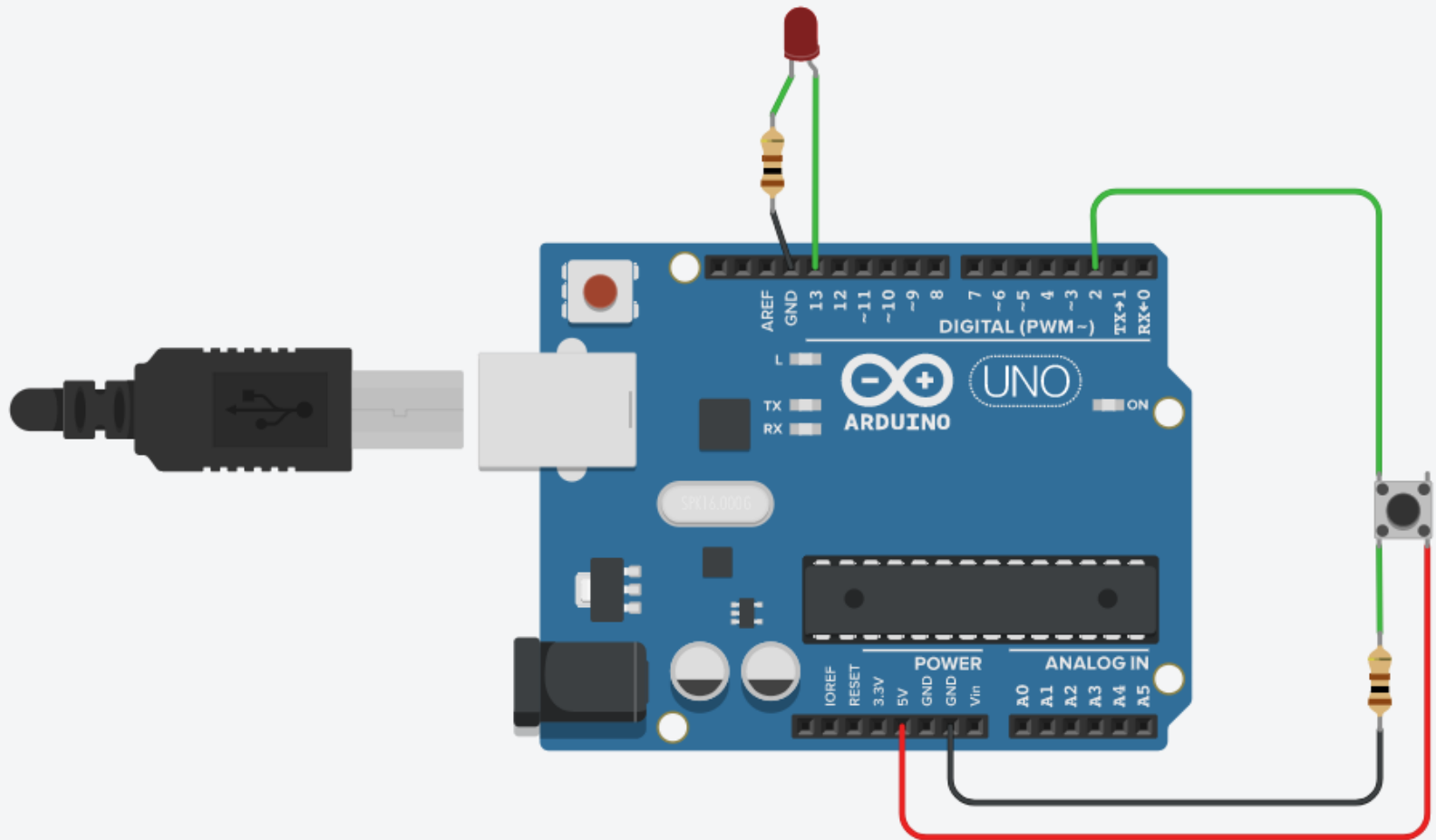
# Blink



# Blink

```
1  void setup()  
2  {  
3      pinMode(13, OUTPUT);  
4  }  
5  
6  void loop()  
7  {  
8      digitalWrite(13, HIGH);  
9      delay(1000);  
10     digitalWrite(13, LOW);  
11     delay(1000);  
12 }
```

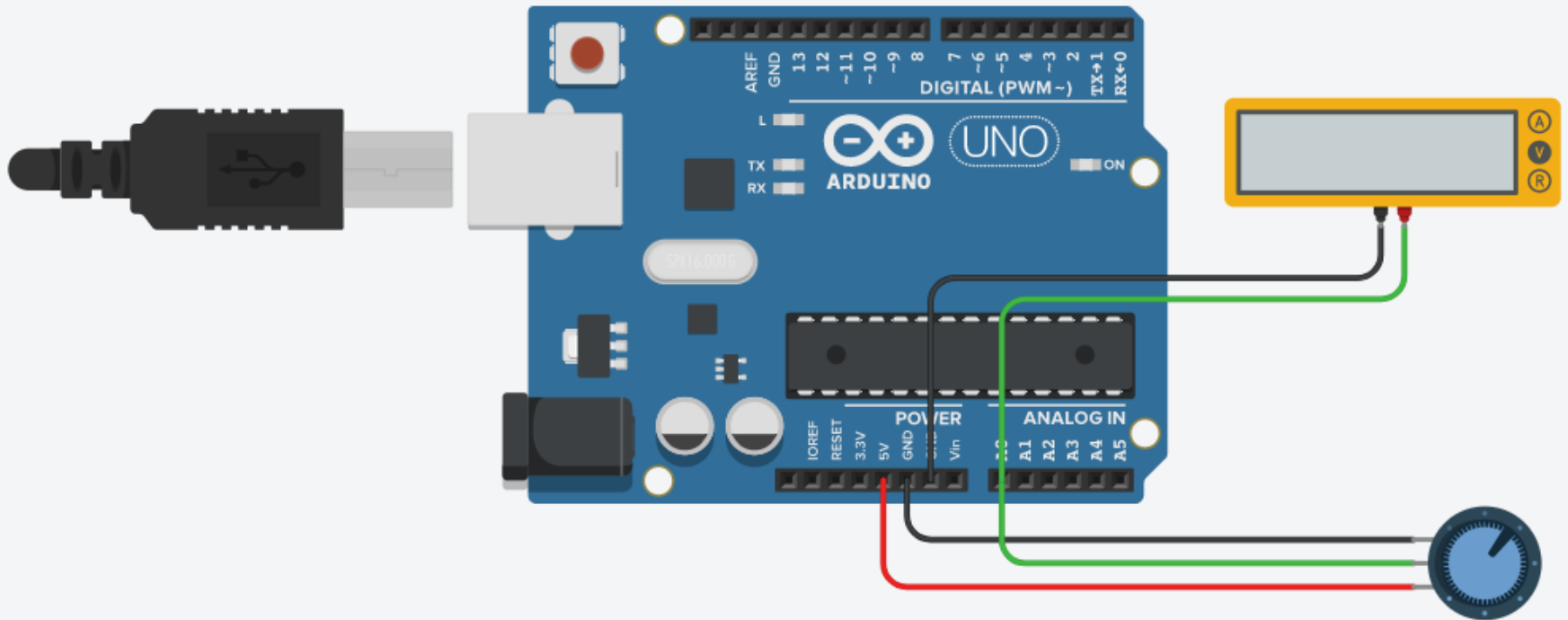
# Botão e LED



# Botão e LED

```
1 void setup()  
2 {  
3     pinMode(2, INPUT);  
4     pinMode(13, OUTPUT);  
5 }  
6  
7 void loop()  
8 {  
9     bool state = digitalRead(2);  
10    digitalWrite(13, state);  
11 }
```

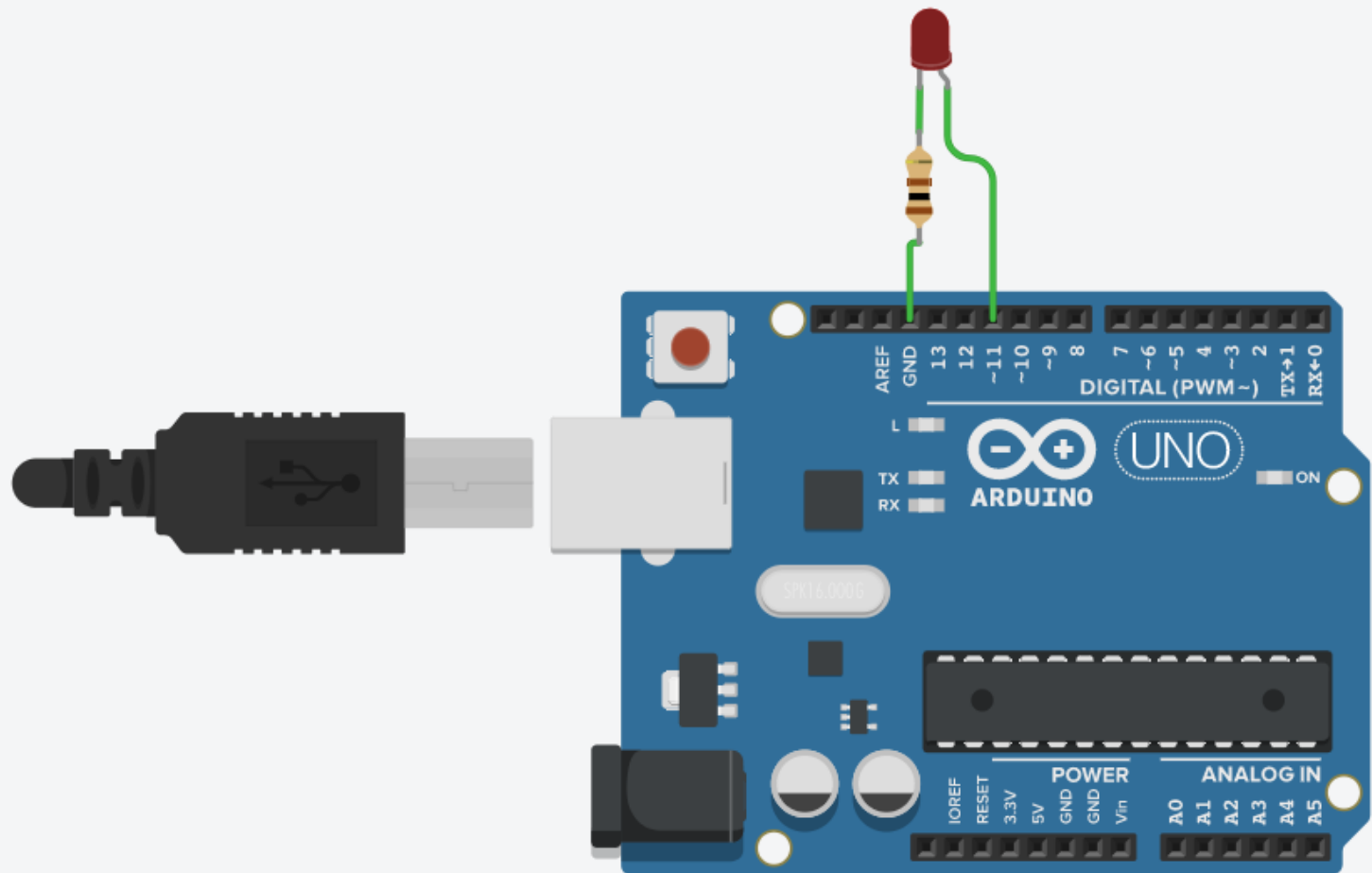
# Entrada analógica



# Entrada analógica

```
1  void setup()  
2  {  
3      Serial.begin(9600);  
4      pinMode(A0, INPUT);  
5  }  
6  
7  void loop()  
8  {  
9      int analogInput = analogRead(A0);  
10     Serial.println(analogInput);  
11 }
```

# Saída analógica

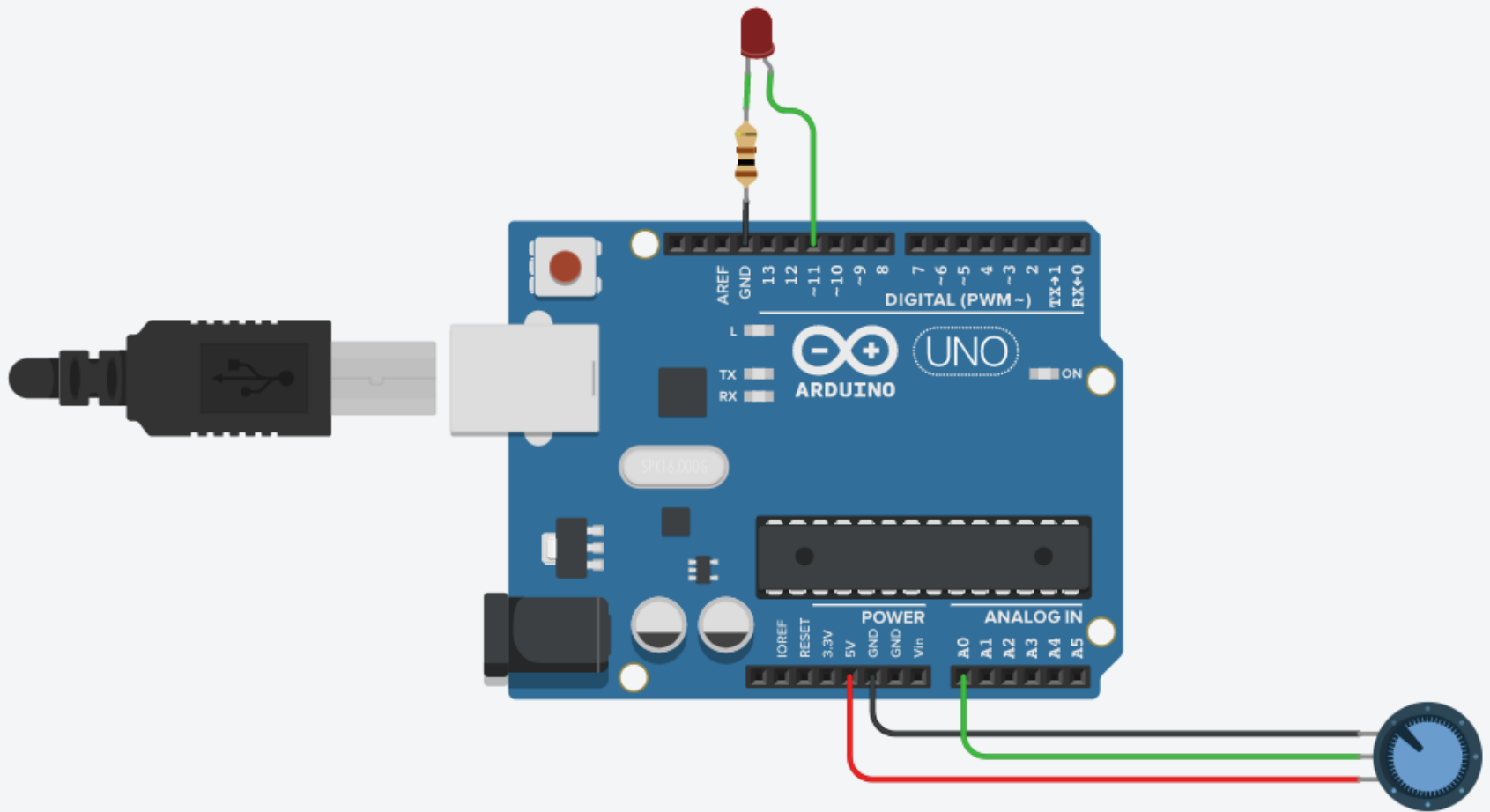




# Saída analógica

```
1  int fade = 0;
2
3  void setup()
4  {
5      pinMode(11, OUTPUT);
6  }
7
8  void loop()
9  {
10     for(fade = 0; fade <= 255; fade++)
11     {
12         analogWrite(11, fade);
13         delay(10);
14     }
15
16     for(fade = 255; fade > 0; fade--)
17     {
18         analogWrite(11, fade);
19         delay(10);
20     }
21
22     delay(500);
23 }
```

# Entrada e saída analógicas

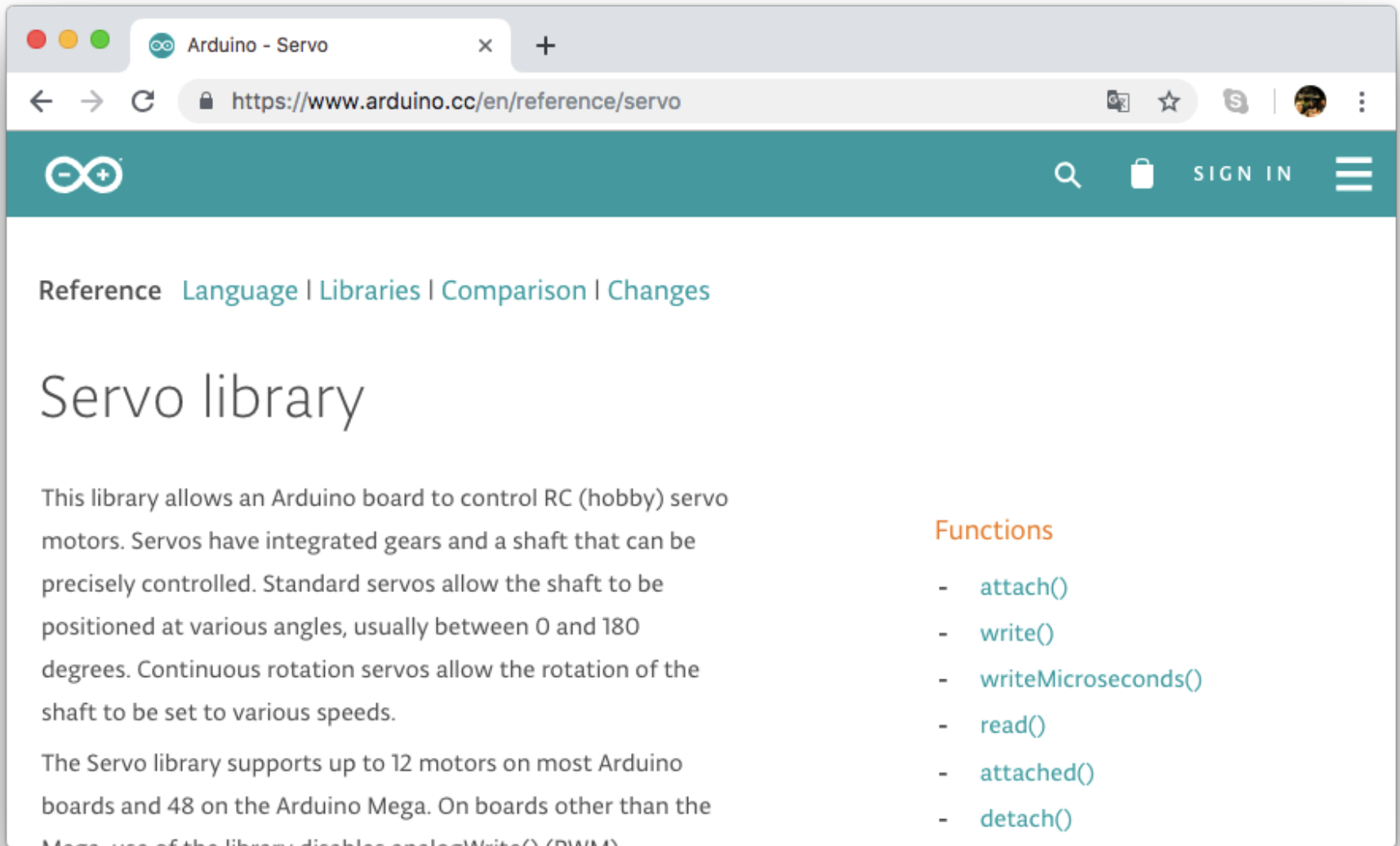


# Entrada e saída analógicas

```
1 void setup()
2 {
3   pinMode(11, OUTPUT);
4 }
5
6 void loop()
7 {
8   int analogInput = analogRead(A0);
9
10  int analogOutput = map(analogInput, 0, 1023, 0, 255);
11
12  analogWrite(11, analogOutput);
13 }
```



<https://www.arduino.cc/en/reference/servo>



The image is a screenshot of a web browser displaying the Arduino Servo library reference page. The browser's address bar shows the URL <https://www.arduino.cc/en/reference/servo>. The page has a teal header with the Arduino logo on the left and search, shopping cart, and sign-in icons on the right. Below the header, there are navigation links: Reference, Language, Libraries, Comparison, and Changes. The main heading is "Servo library". The text describes the library's purpose: controlling RC (hobby) servo motors with integrated gears and shafts, allowing for precise positioning (0 to 180 degrees) and continuous rotation with variable speeds. It also mentions support for up to 12 motors on most boards and 48 on the Arduino Mega. On the right side, under the heading "Functions", a list of functions is provided: `attach()`, `write()`, `writeMicroseconds()`, `read()`, `attached()`, and `detach()`.

Reference [Language](#) | [Libraries](#) | [Comparison](#) | [Changes](#)

## Servo library

This library allows an Arduino board to control RC (hobby) servo motors. Servos have integrated gears and a shaft that can be precisely controlled. Standard servos allow the shaft to be positioned at various angles, usually between 0 and 180 degrees. Continuous rotation servos allow the rotation of the shaft to be set to various speeds.

The Servo library supports up to 12 motors on most Arduino boards and 48 on the Arduino Mega. On boards other than the Mega, use of the library disables `analogWrite()` (PWM).

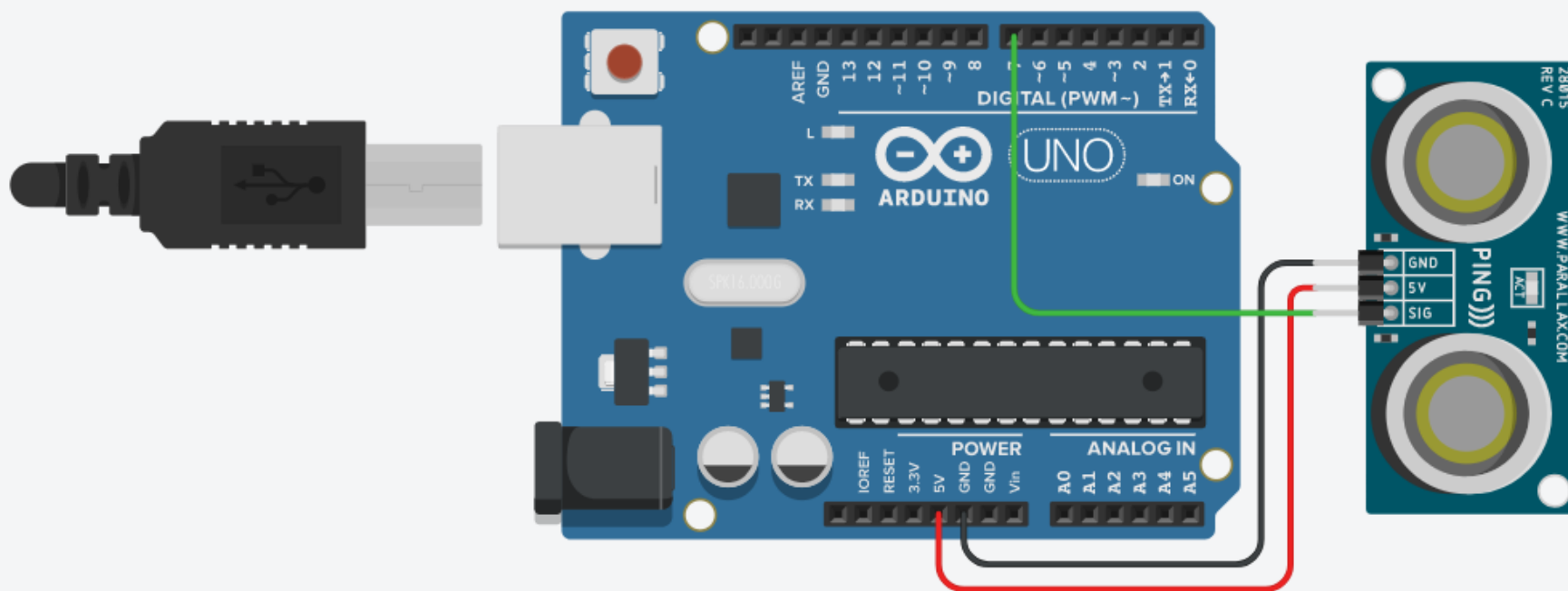
### Functions

- `attach()`
- `write()`
- `writeMicroseconds()`
- `read()`
- `attached()`
- `detach()`

# Potenciômetro e Servo

```
1  #include <Servo.h>
2
3  Servo servo;
4
5  void setup()
6  {
7      pinMode(A1, INPUT);
8      servo.attach(6);
9  }
10
11 void loop()
12 {
13     int analogInput = analogRead(A1);
14
15     int servoPosition = map(analogInput, 0, 1023, 180, 0);
16
17     servo.write(servoPosition);
18
19     delay(10);
20 }
```

# Sensor ultrassônico



# Sensor ultrassônico

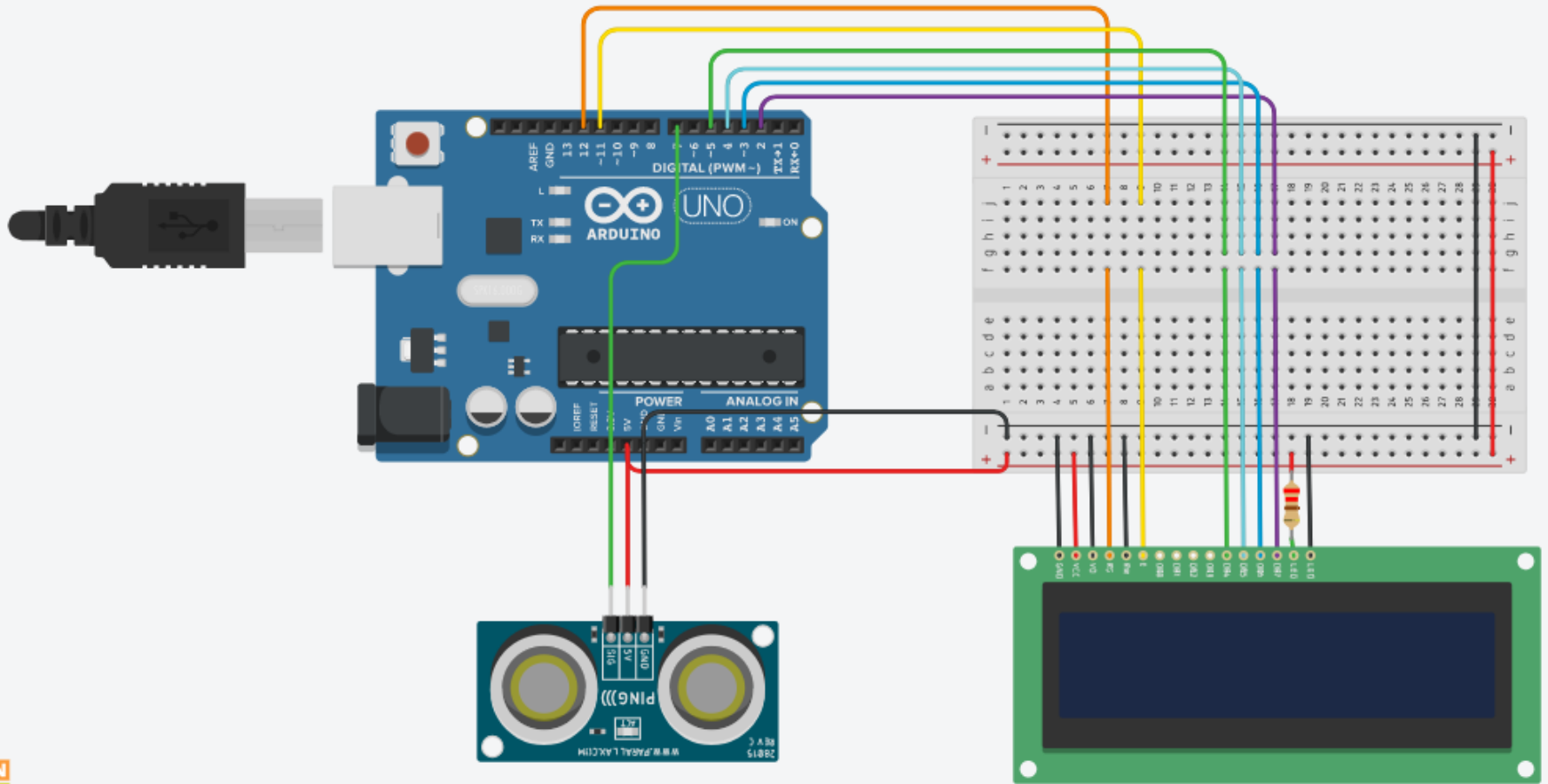
```
1  long readUltrasonicDistance()  
2  {  
3      pinMode(7, OUTPUT);  
4  
5      digitalWrite(7, LOW);  
6      delayMicroseconds(2);  
7      digitalWrite(7, HIGH);  
8      delayMicroseconds(10);  
9      digitalWrite(7, LOW);  
10  
11     pinMode(7, INPUT);  
12  
13     return pulseIn(7, HIGH);  
14 }
```



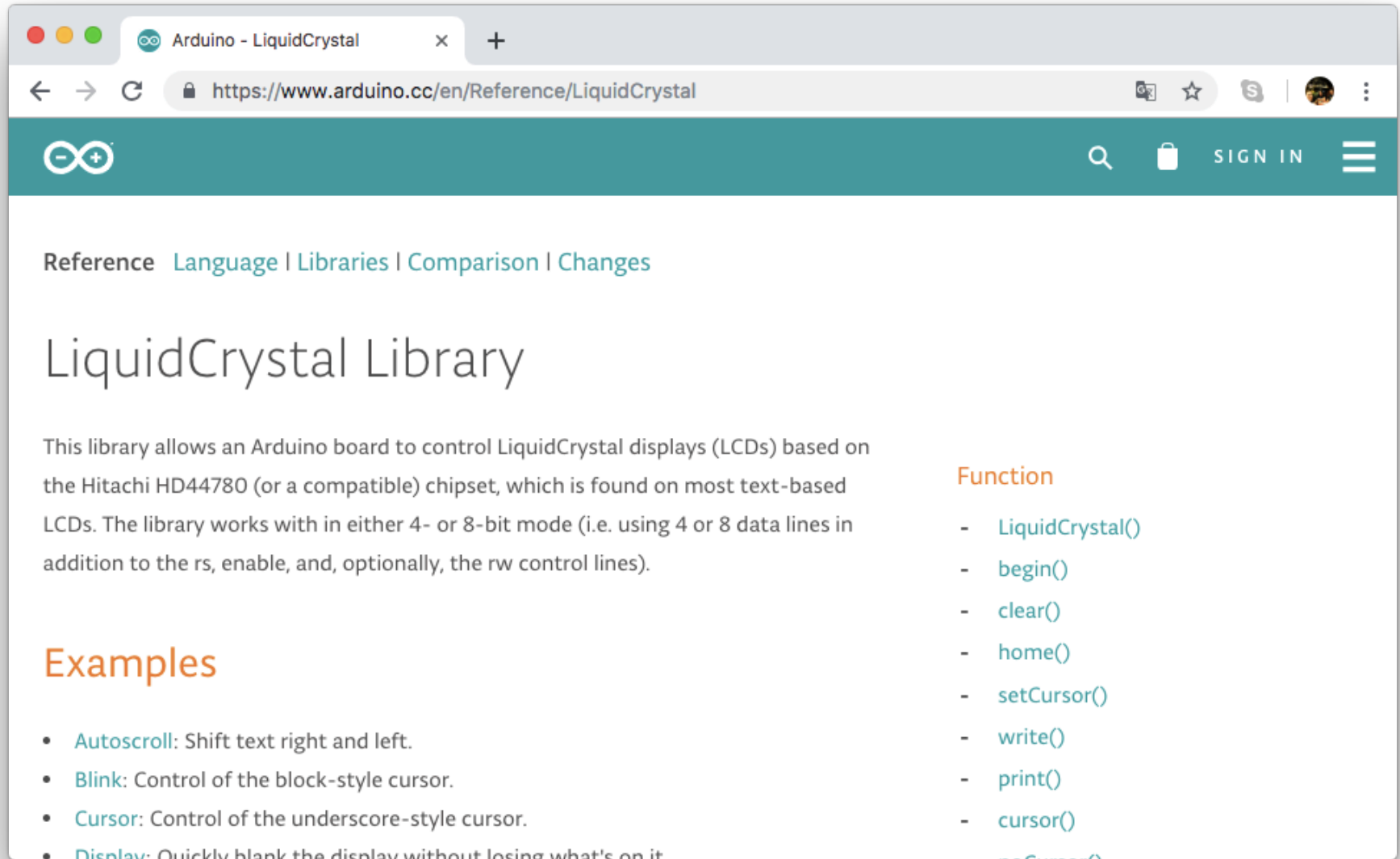
# Sensor ultrassônico

```
--
15 void setup()
16 {
17     pinMode(7, INPUT);
18     Serial.begin(9600);
19 }
20
21 /* Speed of sound is 343 m/s, or 0,0343 cm/us. Divides
22  *   by two because the pulse goes forward and backward. */
23 const double soundSpeed = 343 * 100.0 / 1000000.0;
24
25 void loop()
26 {
27     double cm = (soundSpeed / 2) * readUltrasonicDistance();
28
29     Serial.print(cm);
30     Serial.println("cm");
31
32     delay(100);
33 }
```

# Sensor ultrassônico e LCD



<https://www.arduino.cc/en/Reference/LiquidCrystal>

A screenshot of a web browser displaying the Arduino LiquidCrystal reference page. The browser's address bar shows the URL 'https://www.arduino.cc/en/Reference/LiquidCrystal'. The page has a teal header with the Arduino logo on the left and search, shopping cart, and 'SIGN IN' links on the right. Below the header, there are navigation links: 'Reference', 'Language', 'Libraries', 'Comparison', and 'Changes'. The main title 'LiquidCrystal Library' is prominently displayed. A descriptive paragraph explains that the library allows an Arduino board to control LiquidCrystal displays (LCDs) based on the Hitachi HD44780 chipset. To the right, under the heading 'Function', a list of functions is provided. On the left, under the heading 'Examples', a list of example programs is shown.

Reference [Language](#) | [Libraries](#) | [Comparison](#) | [Changes](#)

# LiquidCrystal Library

This library allows an Arduino board to control LiquidCrystal displays (LCDs) based on the Hitachi HD44780 (or a compatible) chipset, which is found on most text-based LCDs. The library works with in either 4- or 8-bit mode (i.e. using 4 or 8 data lines in addition to the rs, enable, and, optionally, the rw control lines).

## Examples

- [Autoscroll](#): Shift text right and left.
- [Blink](#): Control of the block-style cursor.
- [Cursor](#): Control of the underscore-style cursor.
- [Display](#): Quickly blank the display without losing what's on it.

## Function

- [LiquidCrystal\(\)](#)
- [begin\(\)](#)
- [clear\(\)](#)
- [home\(\)](#)
- [setCursor\(\)](#)
- [write\(\)](#)
- [print\(\)](#)
- [cursor\(\)](#)

# Sensor ultrassônico e LCD

```
20 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
21
22 void setup()
23 {
24     lcd.begin(16, 2);
25     lcd.print("Hello world!");
26 }
27
28 void loop()
29 {
30     lcd.setCursor(0, 1);
31     lcd.print("                ");
32     lcd.setCursor(0, 1);
33     String cm = String((soundSpeed / 2) * readUltrasonicDistance());
34     String msg = String(cm + " cm");
35     lcd.print(msg);
36
37     delay(500);
38 }
```

# Referências

- <https://www.arduino.cc>
- <https://www.arduino.cc/reference/en>
- <https://www.tinkercad.com>
- [https://youtu.be/WynYhZx\\_qds](https://youtu.be/WynYhZx_qds)
- <https://www.arduino.cc/en/Tutorial/Knob>
- <http://www.arduino.cc/en/Tutorial/Ping>
- <http://www.arduino.cc/en/Tutorial/LiquidCrystal>

# Obrigado!

Felipe Kühne

[fkuhne@gmail.com](mailto:fkuhne@gmail.com)



[github.com/fkühne/sae2019](https://github.com/fkühne/sae2019)