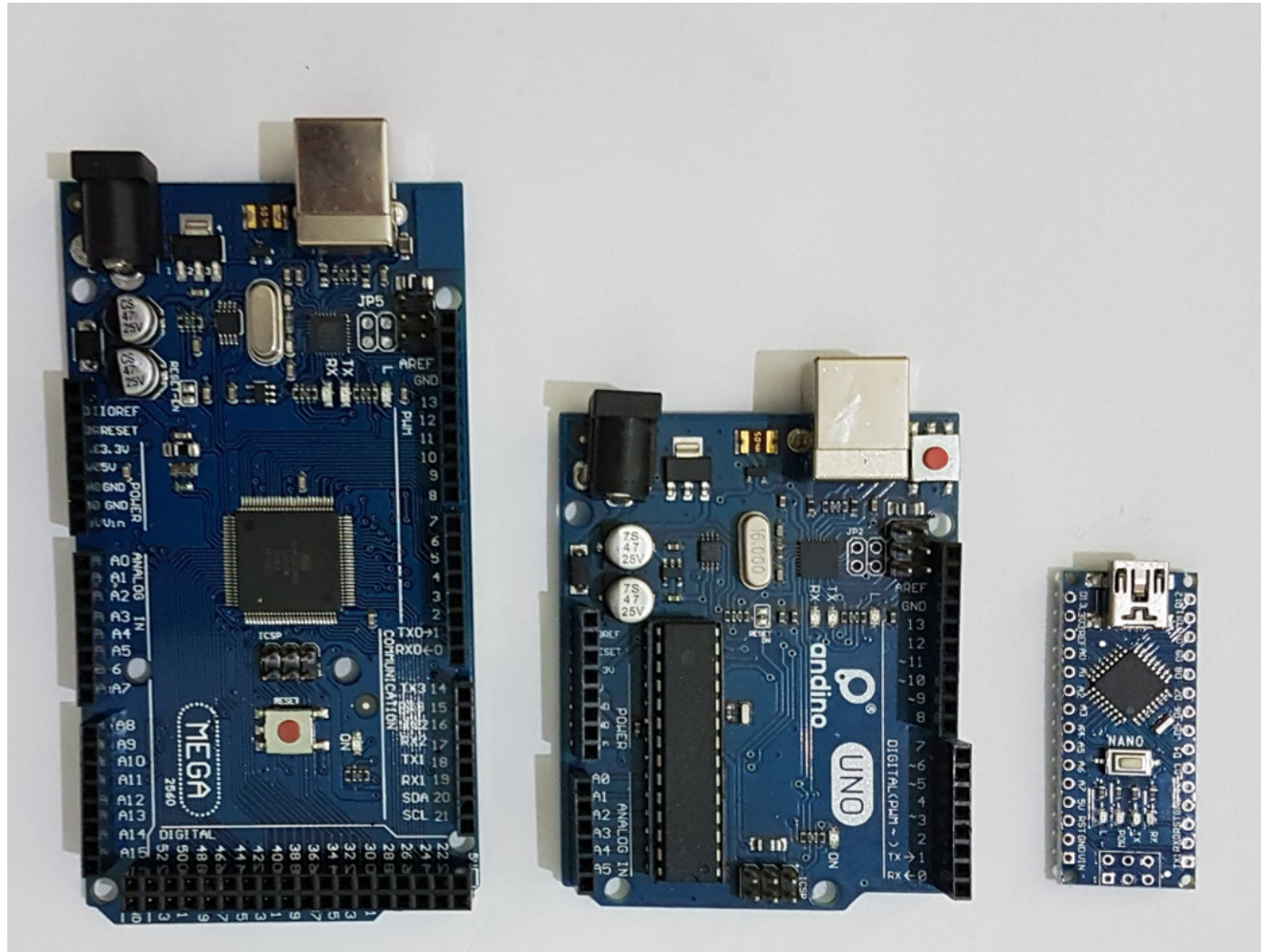




<https://www.arduino.cc>

Primeros pasos en robótica con Arduino hardware libre y software libre

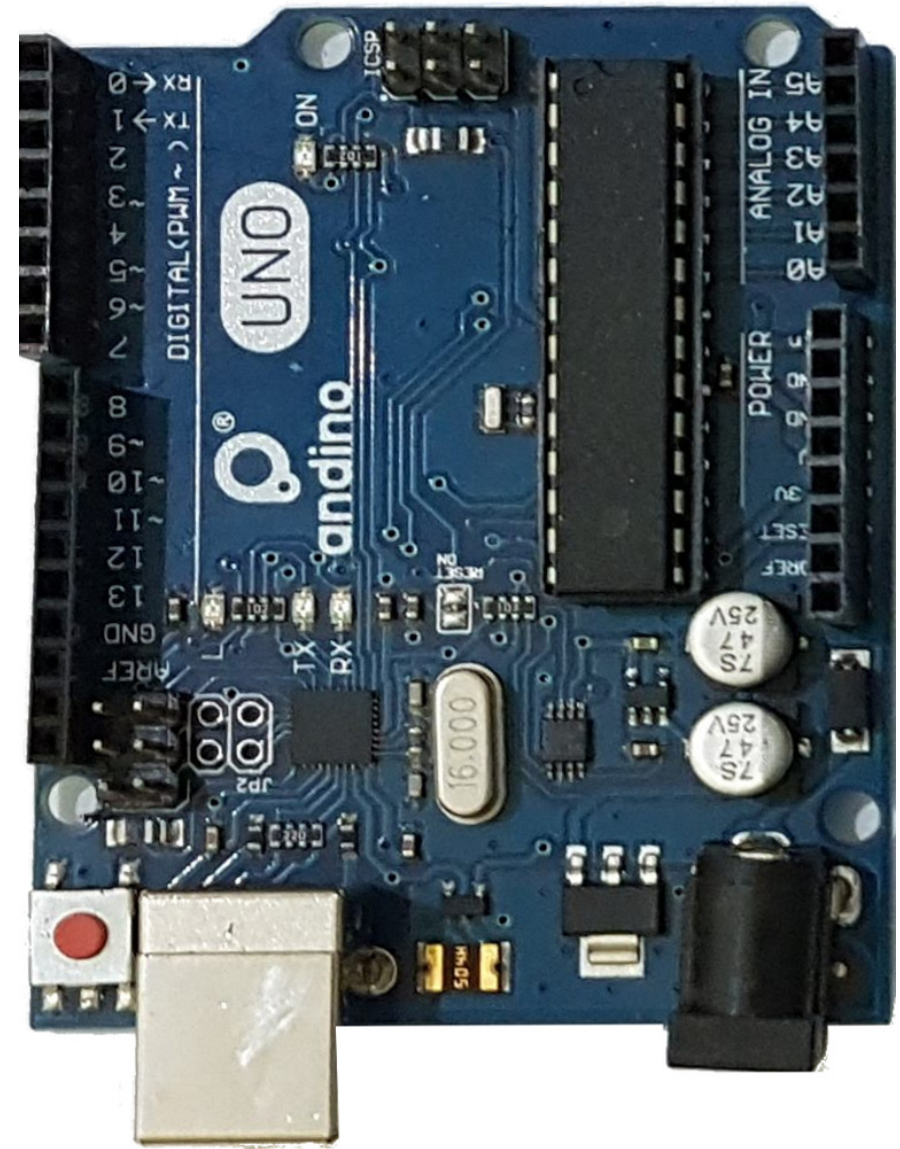
Diferentes modelos: Arduino nano, Arduino UNO, Arduino MEGA



Arduino UNO



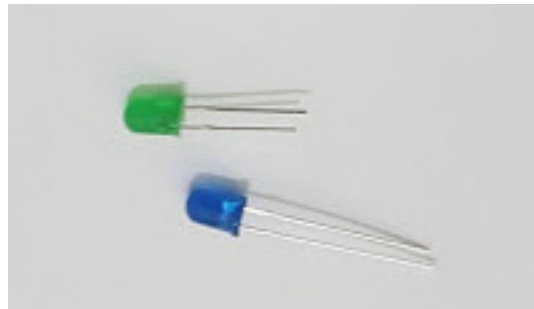
Pines 2 al 13: Digitales (valor 1 o 0)
Pines A0 al A5: Analógicos
GND: Ground (tierra)
5v: 5 volts



Arduino UNO



LED:
Diodo emisor de
luz



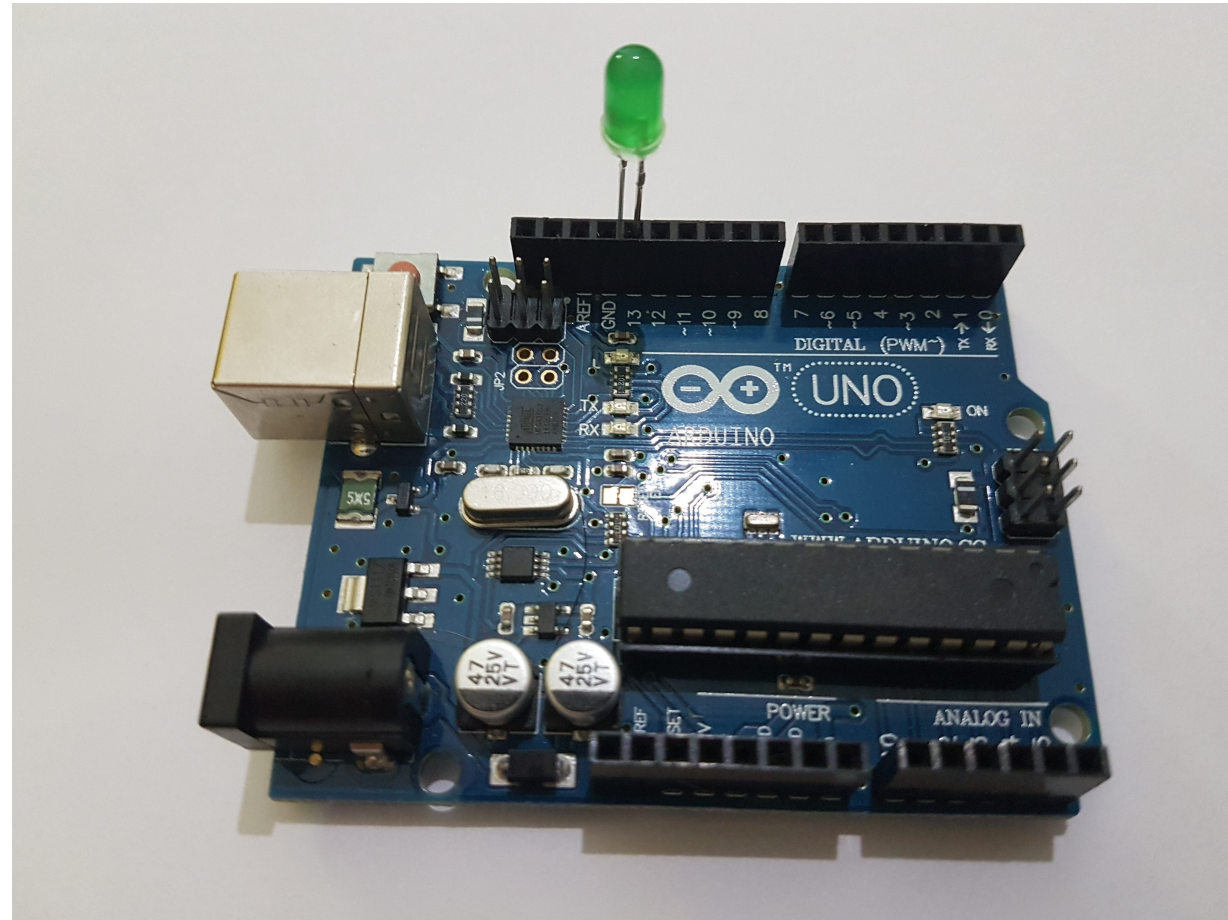
Pata larga: ánodo. Positivo.
Pata corta: cátodo. Negativo.

Arduino UNO



Pin 13: Pin digital que tiene incorporada una resistencia

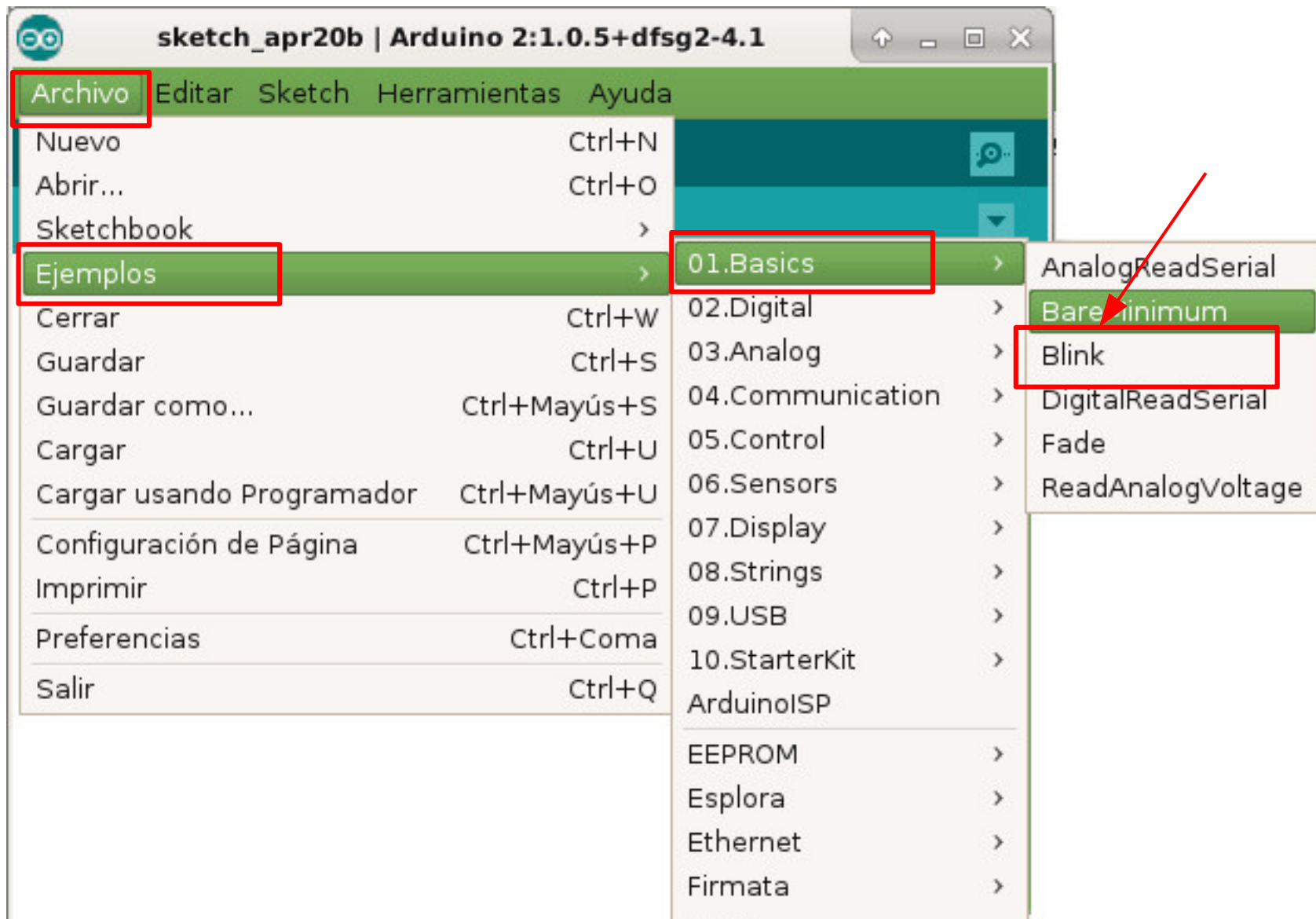
Vamos a conectar un led con su pata larga (positiva) en el pin 13, y su pata corta (negativa) en tierra (GND)



Arduino IDE

IDE significa "Integrated Development Environment" ("Entorno de Desarrollo Integrado").

Vamos a abrir un programa de ejemplo: blink



<https://www.arduino.cc/en/Main/Software>

Arduino IDE

```
/* Blink: enciende y apaga un led */  
  
// Le ponemos un nombre al pin 13:  
int led = 13;  
  
// la parte del programa que está en "void setup" se ejecuta una sola vez  
void setup() {  
  // le decimos que el pin 13 será de salida:  
  pinMode(led, OUTPUT);  
}  
  
// esta parte se repite una y otra vez:  
void loop() {  
  digitalWrite(led, HIGH); // encender LED  
  delay(1000); // esperar un segundo  
  digitalWrite(led, LOW); // apagar LED  
  delay(1000); // esperar un segundo  
}
```

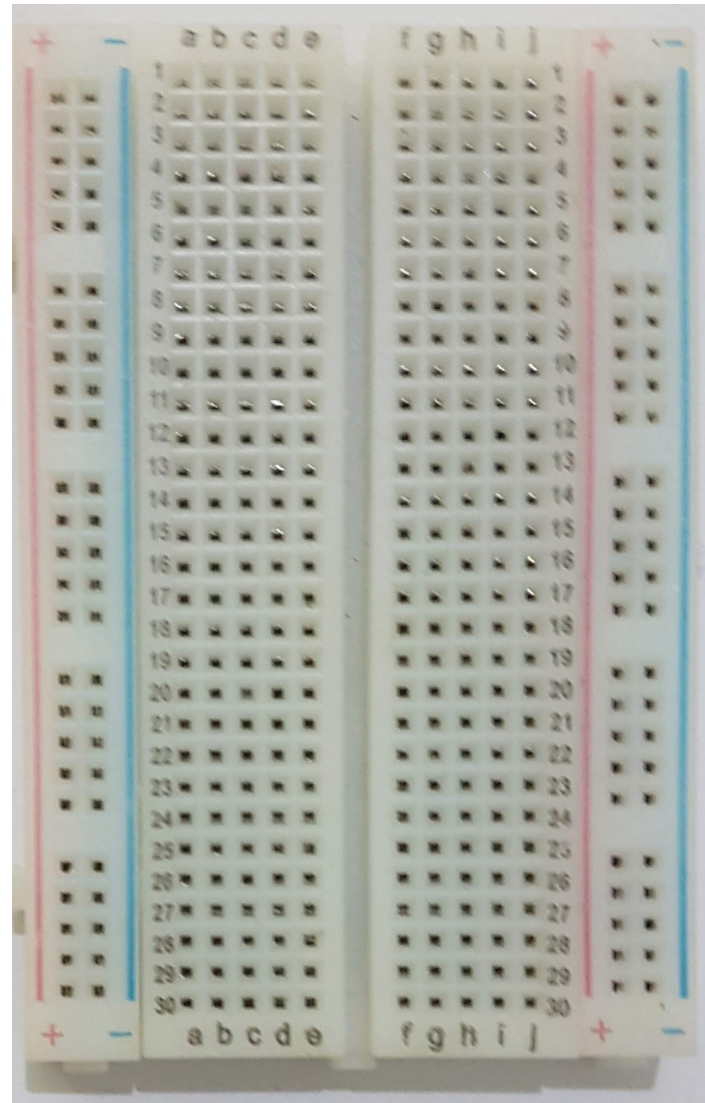
Los comentarios
sirven para
entender el
código.

Vamos a
modificar la
cantidad de
segundos de
encendido y
apagado.

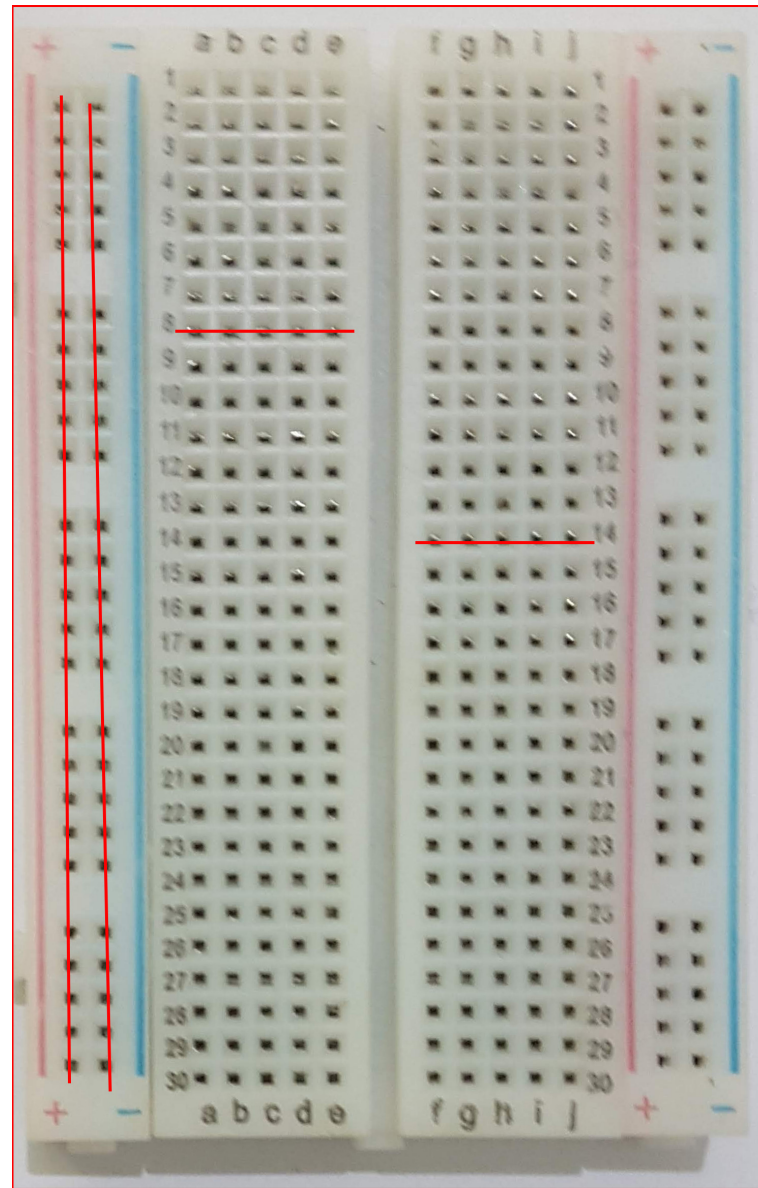
<https://www.arduino.cc/en/Main/Software>

Protoboard

Placa de pruebas o placa sin soldaduras. Se utiliza para conectar los elementos sin tener que soldar.



Protoboard

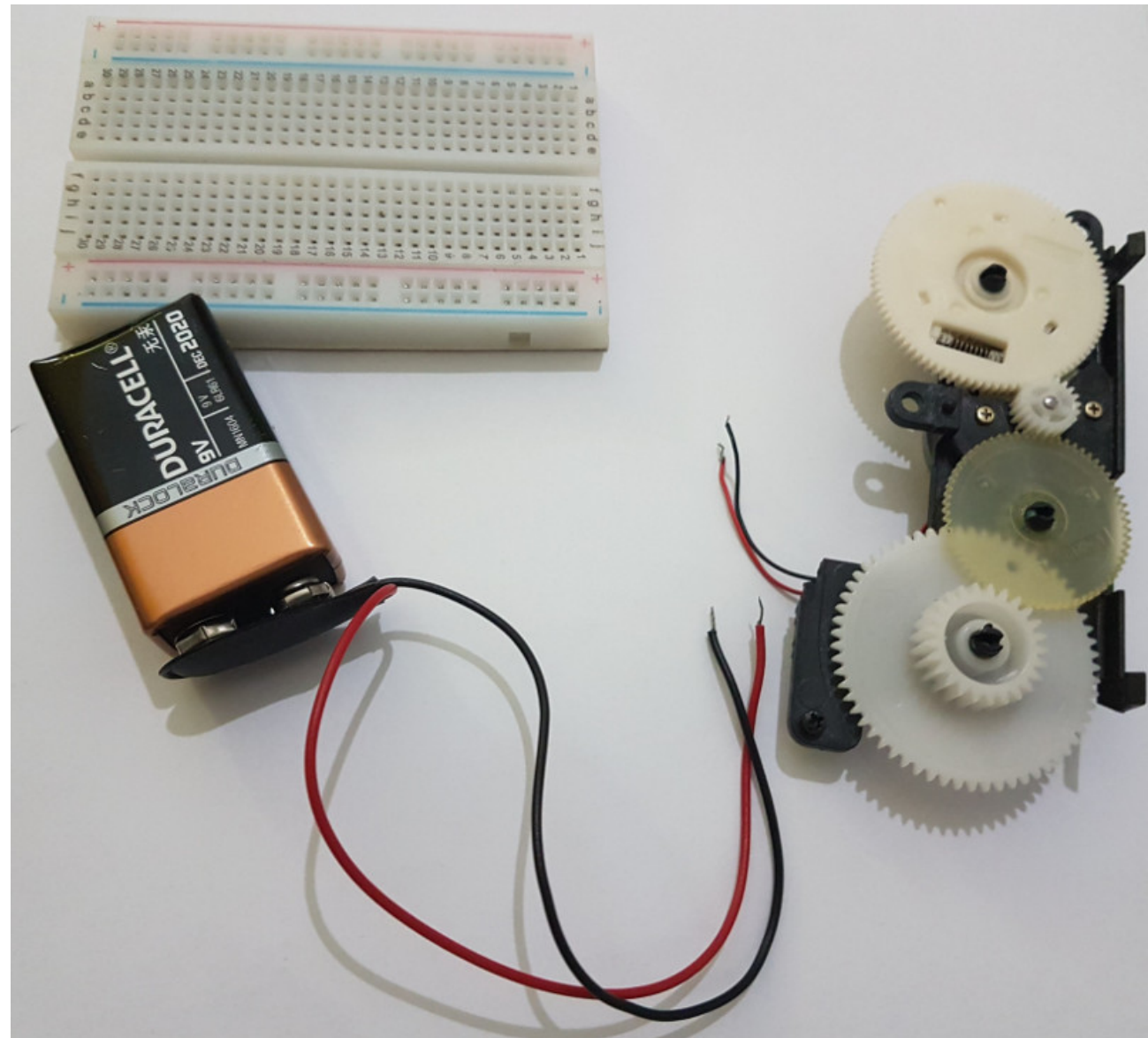


Las líneas rojas muestran cómo están interconectados los pines.

Protoboard

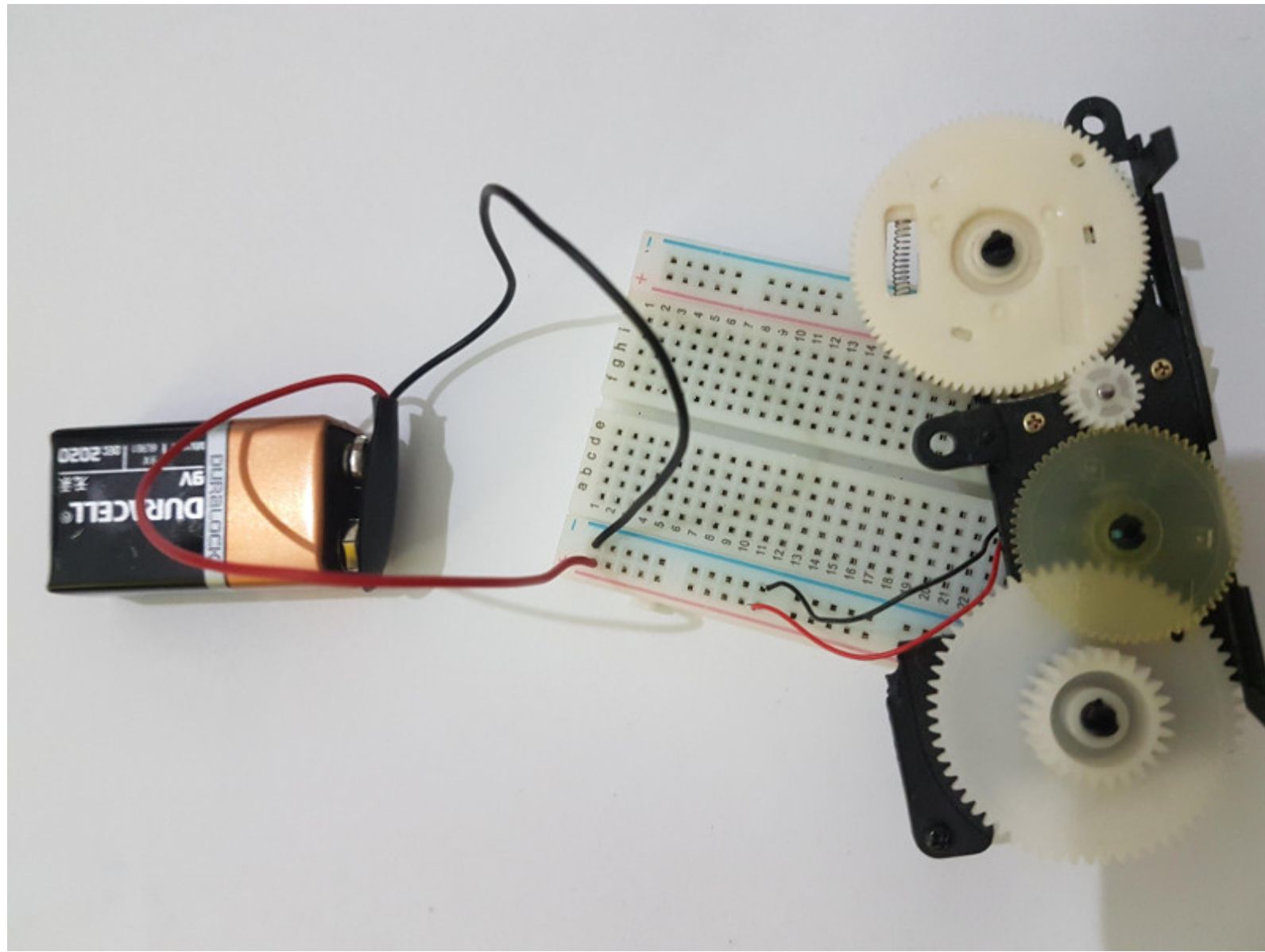
Vamos a probar
la protoboard
con una batería
de 9v y un
motor DC
tomado de una
lectora de Cds.

Video 1: batería y motor



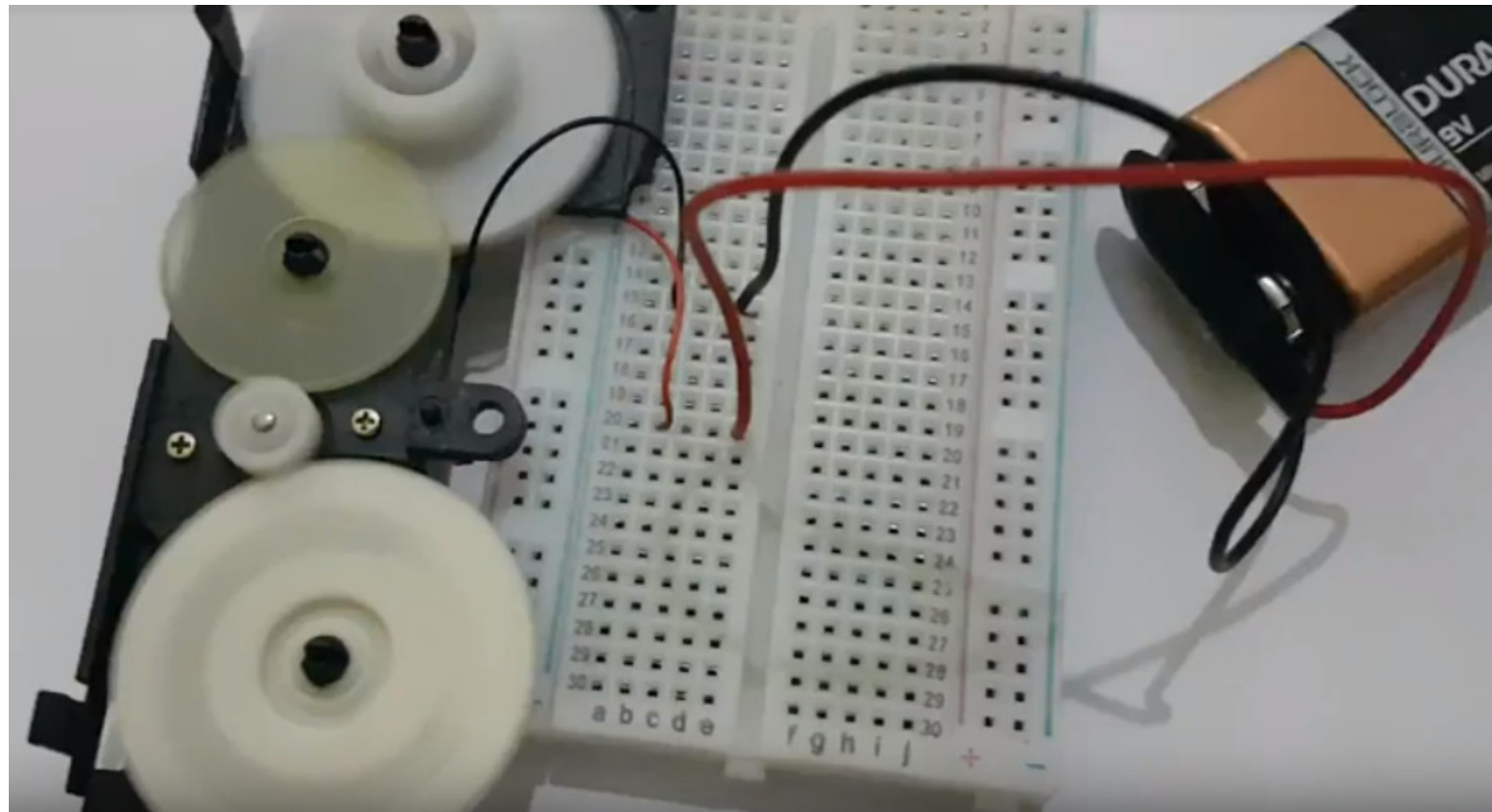
Protoboard

Conectamos en una fila el positivo y en otra fila el negativo. Podemos elegir cualquier pin de la misma hilera.



Video 2: batería y motor

Protoboard

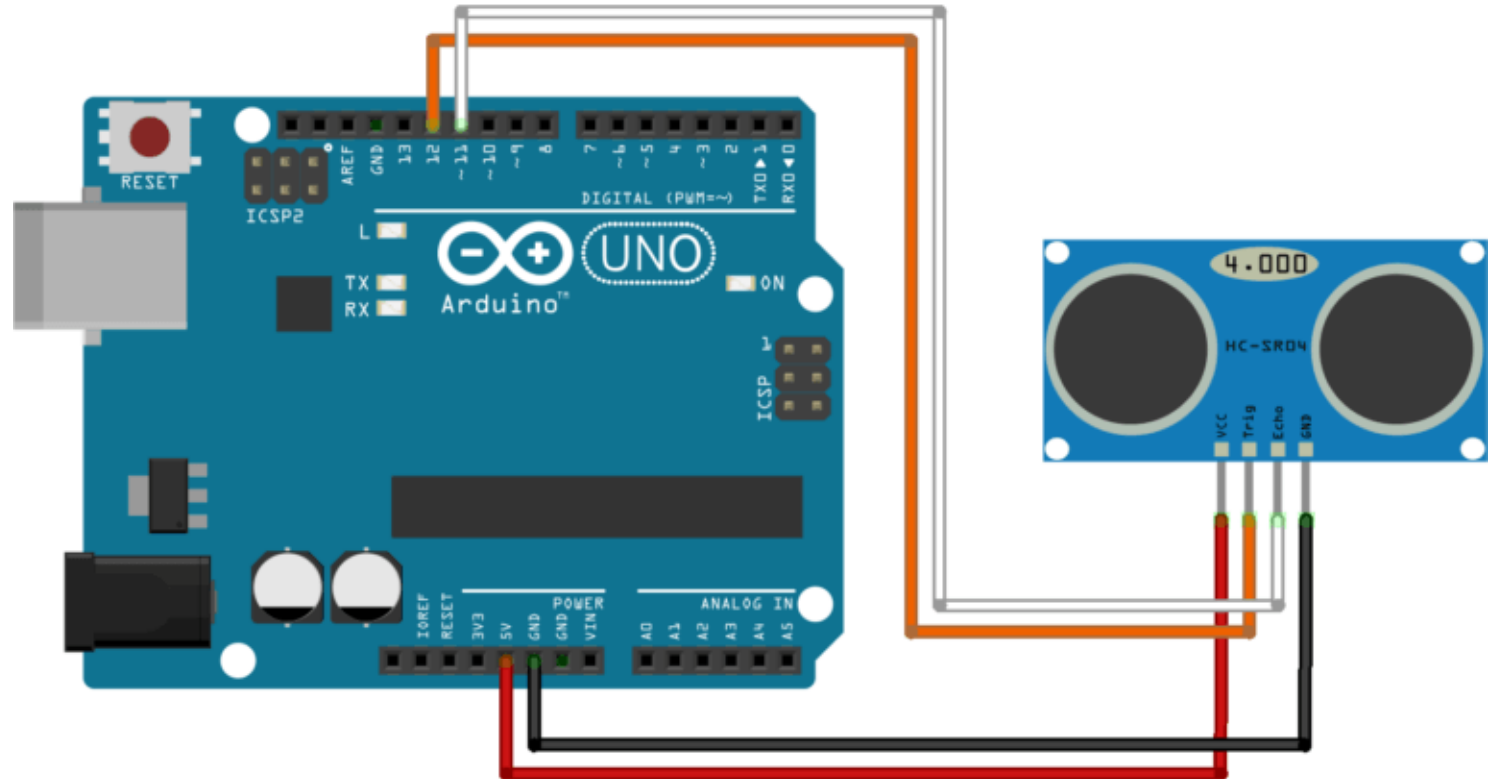


Conectamos en una fila el positivo y en otra fila el negativo. Podemos elegir cualquier pin de la misma hilera.

Video 3: batería y motor

Sensor de distancia

Sensor ultrasónico: envía una señal sonora y mide el tiempo que esa señal tarda en volver, al rebotar contra un objeto. Es un sensor de **distancia**.



<https://www.geekfactory.mx/tutoriales/tutoriales-arduino/sensor-ultrasonico-hc-sr04-y-arduino/>

fritzing

Vamos a abrir un programa de ejemplo: blink

<https://www.geekfactory.mx/tutoriales/tutoriales-arduino/sensor-ultrasonico-hc-sr04-y-arduino/>

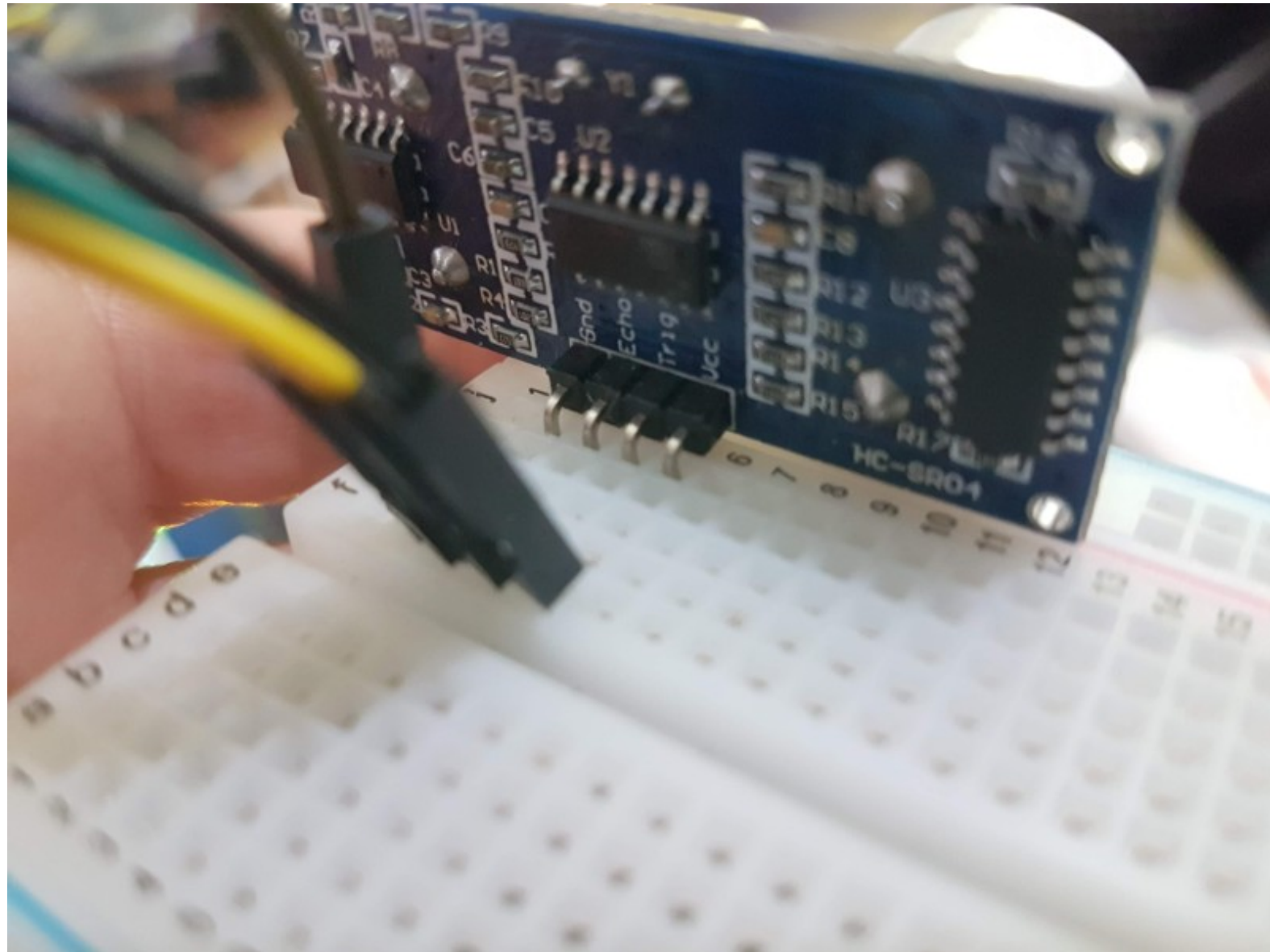
Lic. Iris Fernández
2019



Sensor de distancia



Sensor ultrasónico: Lo conectamos desde la protoboard.



<https://www.geekfactory.mx/tutoriales/tutoriales-arduino/sensor-ultrasonico-hc-sr04-y-arduino/>

Lic. Iris Fernández
2019



Sensor de distancia

Como se trata de software libre, podemos ver ejemplos de programas, descargarlos, modificarlos y compartirlos.

```
//Codigo de https://www.geekfactory.mx
// DECLARACION DE VARIABLES PARA PINES
const int pinecho = 8;
const int pintrigger = 9;
const int pinled = 13;

// VARIABLES PARA CALCULOS
unsigned int tiempo, distancia;

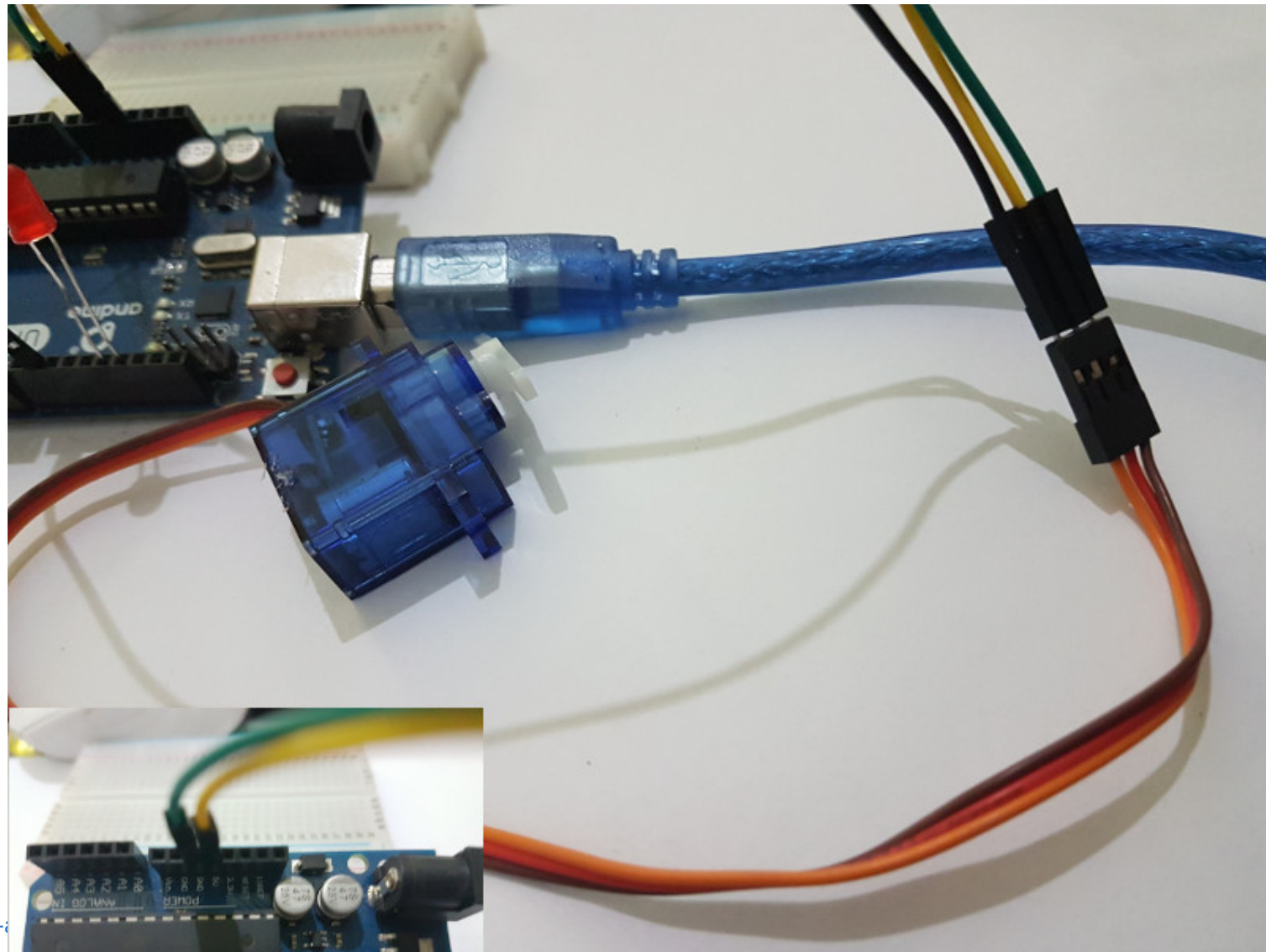
void setup() {
  // PREPARAR LA COMUNICACION SERIAL
  Serial.begin(9600);
  // CONFIGURAR PINES DE ENTRADA Y SALIDA
  pinMode(pinecho, INPUT);
  pinMode(pintrigger, OUTPUT);
  pinMode(13, OUTPUT);
}

void loop() {
  // ENVIAR PULSO DE DISPARO EN EL PIN "TRIGGER"
  digitalWrite(pintrigger, LOW);
  delayMicroseconds(2);
  digitalWrite(pintrigger, HIGH);
  // EL PULSO DURA AL MENOS 10 uS EN ESTADO ALTO
```

<https://www.geekfactory.mx/tutoriales/tutoriales-arduino/sensor-ultrasonic-o-hc-sr04-y-arduino/>

Servomotor

El cable rojo se conecta a 5v.
El cable marrón, a GND.
El cable anaranjado, al pin número 9.



<https://www.geekfactory.mx/tutoriales/tutoriales-4>

Servomotor

```
#include <Servo.h>

Servo myservo; // create servo object to control a servo
int pos = 0; // variable to store the servo position

void setup()
{
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}

void loop()
{
  myservo.write(90); // tell servo to go to position in variable
  delay(15); // waits 15ms for the servo to reach the
  myservo.write(0); // tell servo to go to position in variable
  delay(15); // waits 15ms for the servo to reach the
}
}
```

// Sweep
// by BARRAGAN
<<http://barraganstudio.com>>
// This example code is
// in the public domain.