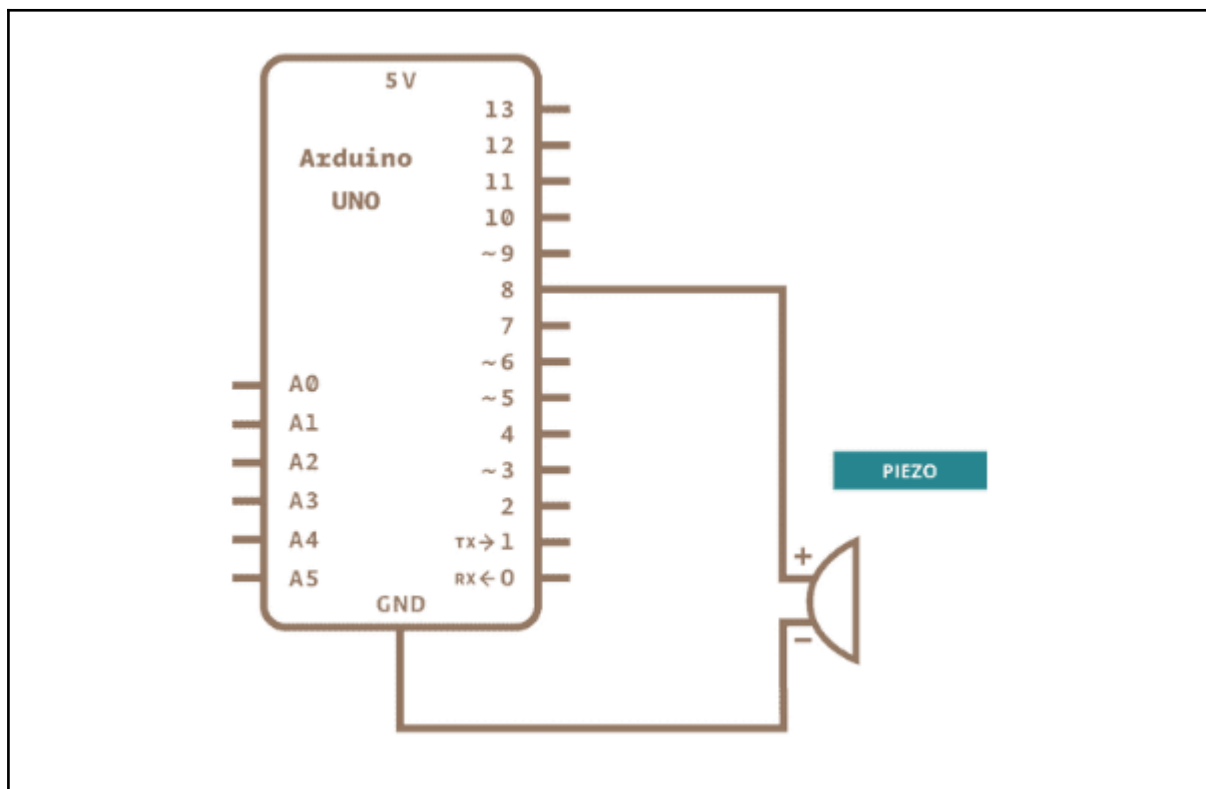
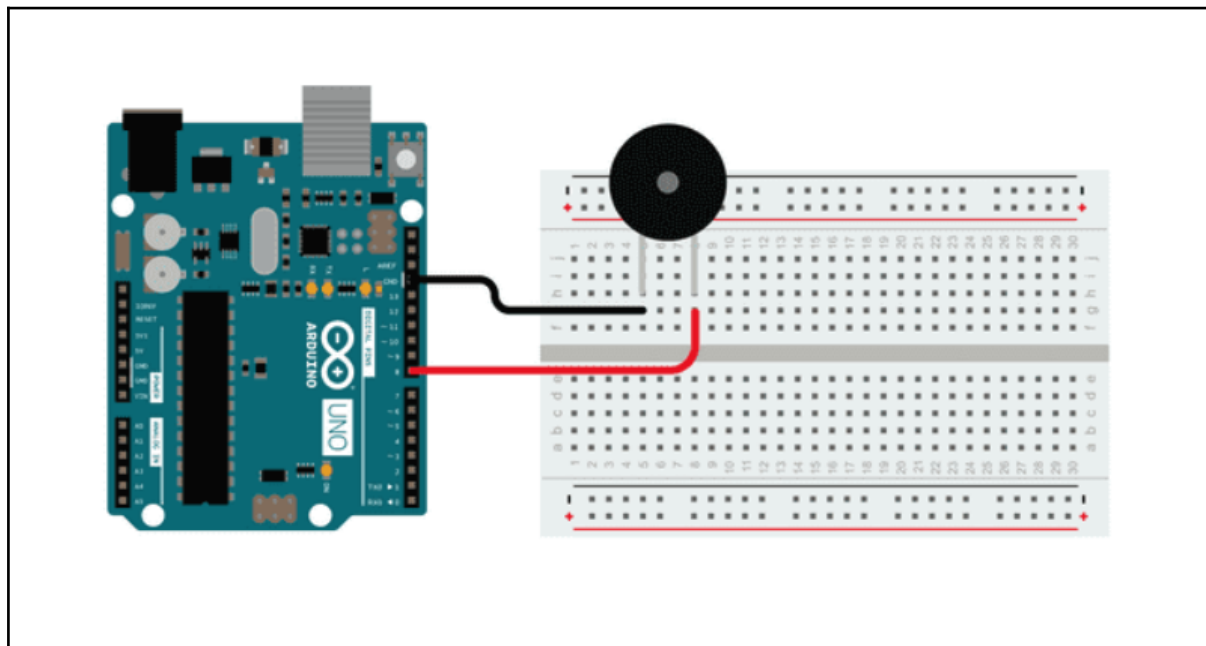


Buzzer



Buzzer

```
#include "pitches.h"

// notes in the melody:
int melody[] = {
  NOTE_C4, NOTE_G3, NOTE_G3, NOTE_A3, NOTE_G3, 0, NOTE_B3, NOTE_C4
};

// note durations: 4 = quarter note, 8 = eighth note, etc.:
int noteDurations[] = {
  4, 8, 8, 4, 4, 4, 4, 4
};

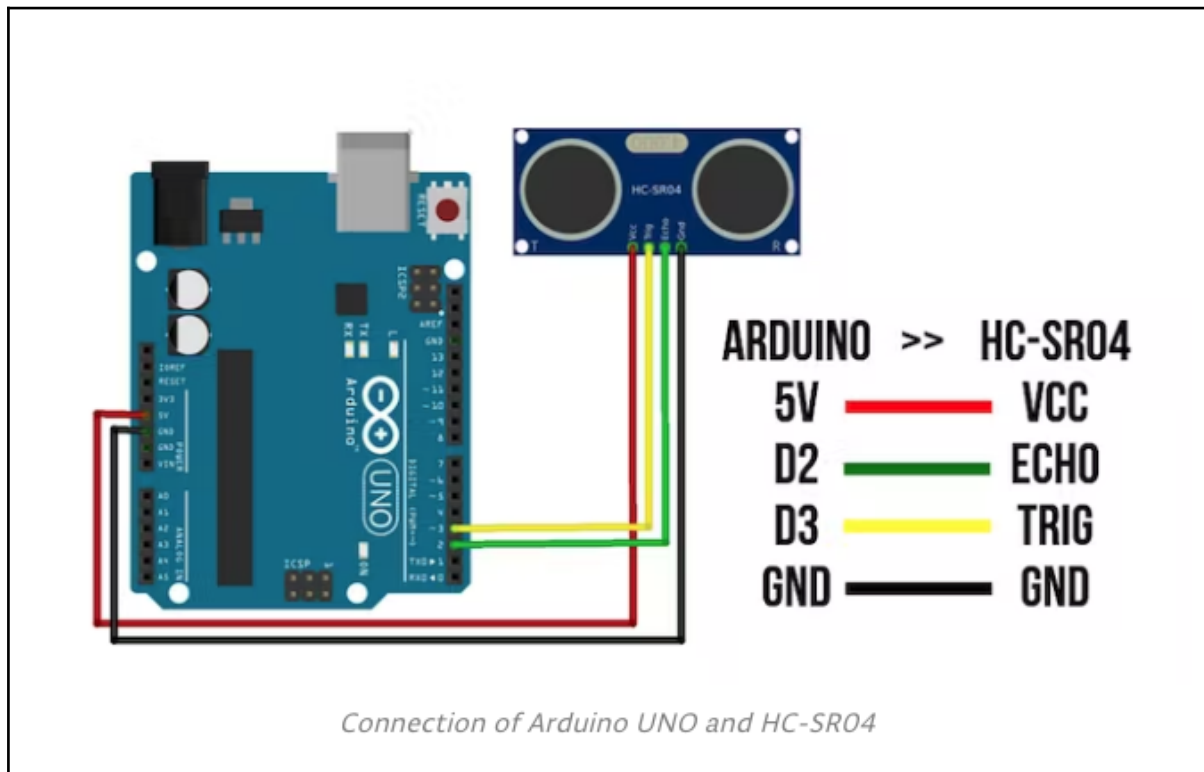
void setup() {
  // iterate over the notes of the melody:
  for (int thisNote = 0; thisNote < 8; thisNote++) {

    // to calculate the note duration, take one second divided by the note type.
    //e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.
    int noteDuration = 1000 / noteDurations[thisNote];
    tone(8, melody[thisNote], noteDuration);

    // to distinguish the notes, set a minimum time between them.
    // the note's duration + 30% seems to work well:
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);
    // stop the tone playing:
    noTone(8);
  }
}

void loop() {
  // no need to repeat the melody.
}
```

Distance



Datos

- **Velocidad del Sonido:** 340 m/s
- **Unidad de medida del sensor:** microsegundos (us). 1 segundo = 1000000 us
- velocidad = distancia/tiempo
 - distancia = velocidad/tiempo

Distance

```
const int Trigger = 6; //Pin digital 2 para el Trigger del sensor
const int Echo = 7; //Pin digital 3 para el Echo del sensor

void setup() {

    Serial.begin(9600); //iniciailzamos la comunicación
    pinMode(Trigger, OUTPUT); //pin como salida
    pinMode(Echo, INPUT); //pin como entrada
    digitalWrite(Trigger, LOW); //Inicializamos el pin con 0
}

void loop(){

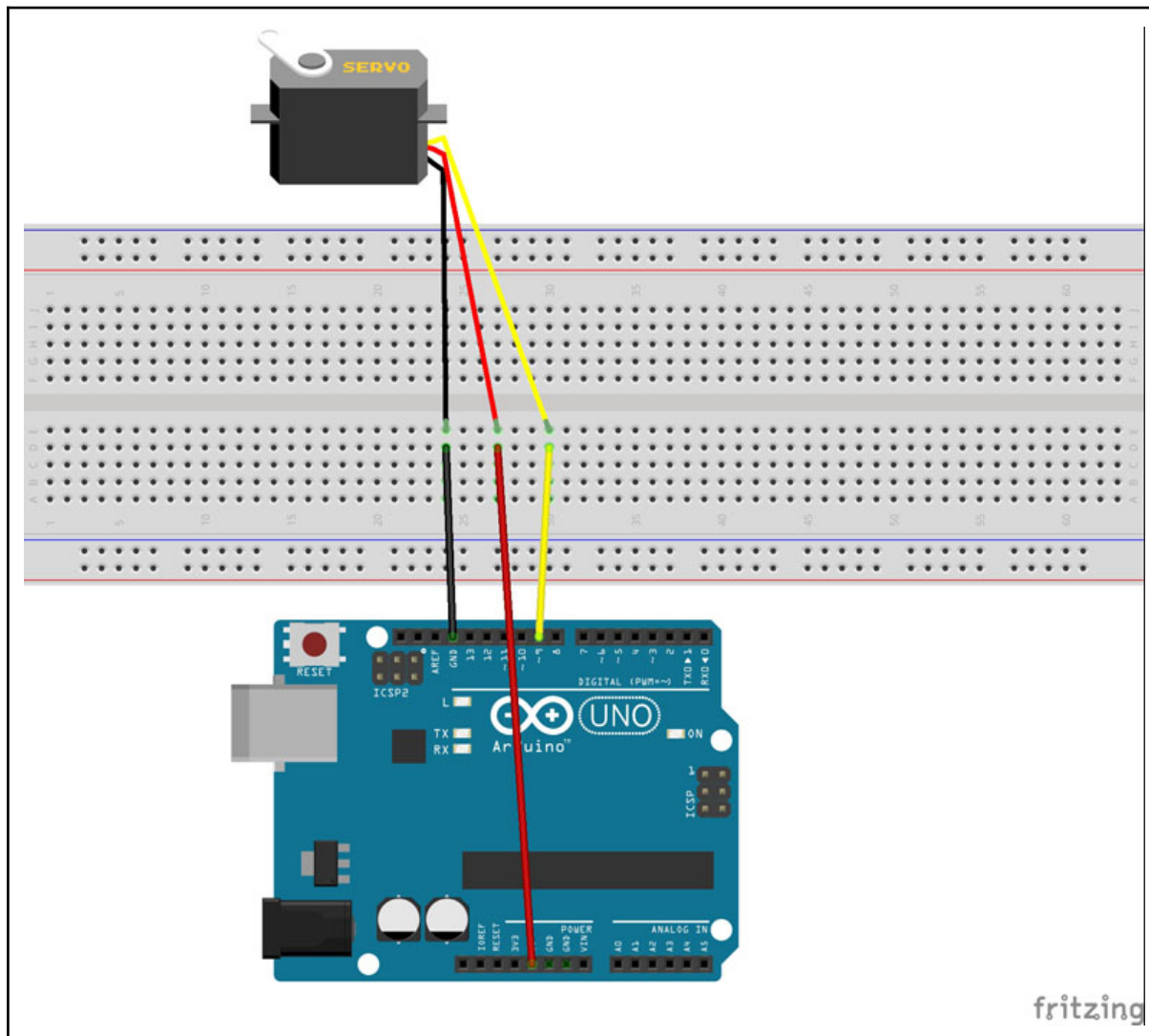
    long t; //timepo que demora en llegar el eco
    long d; //distancia en centimetros

    digitalWrite(Trigger, HIGH);
    delayMicroseconds(10); //Enviamos un pulso de 10us
    digitalWrite(Trigger, LOW);

    t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
    d = t/59; //escalamos el tiempo a una distancia en cm

    Serial.print("Distancia: ");
    Serial.print(t); //Enviamos serialmente el valor de la distancia
    Serial.print("cm");
    Serial.println();
    delay(1000); //Hacemos una pausa de 100ms
}
```

Servo



Servo

```
#include <Servo.h>

Servo myservo; // crea el objeto servo

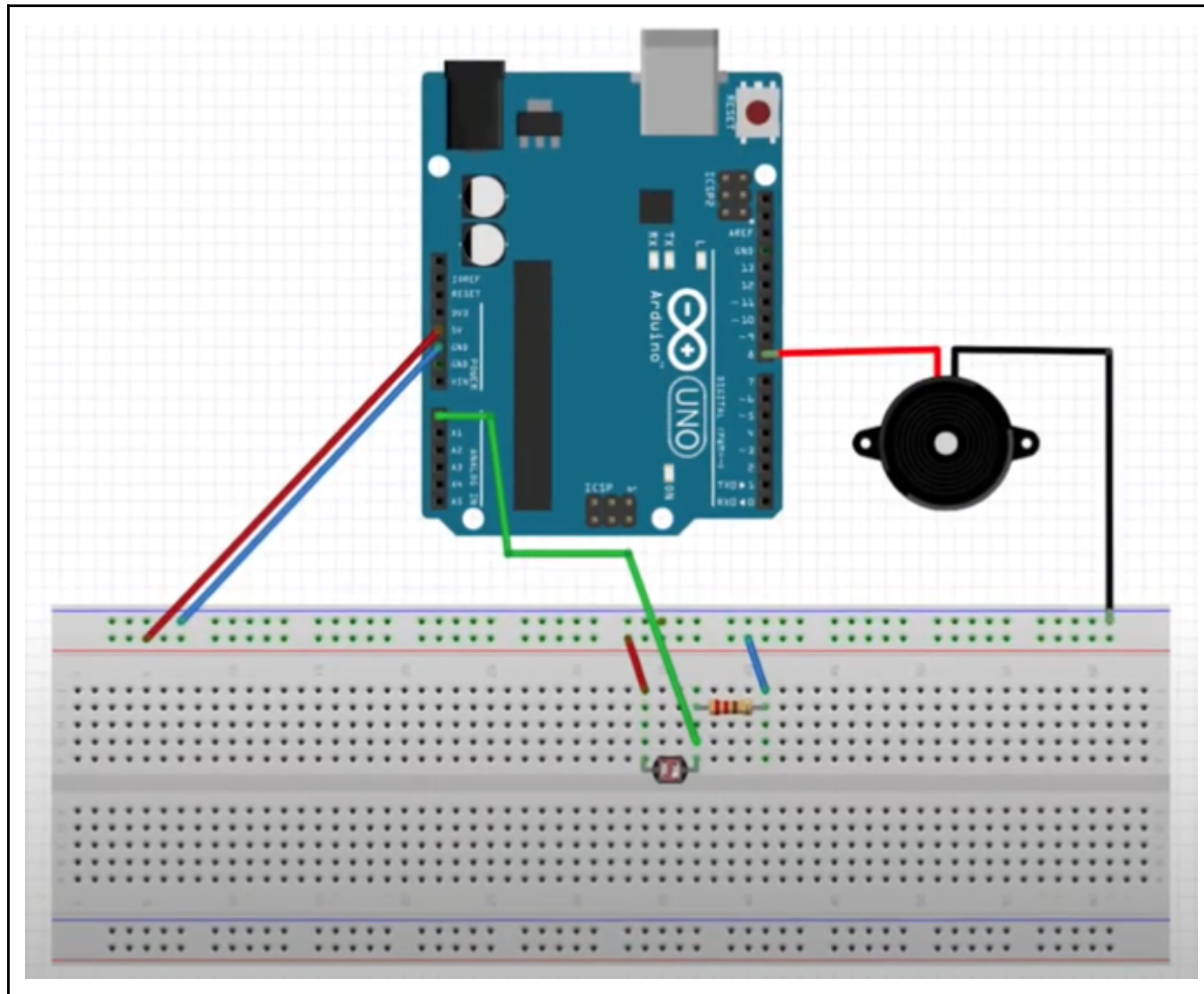
int pos = 0; // posicion del servo

void setup() {
  myservo.attach(9); // vincula el servo al pin digital 9
}

void loop() {
  //varia la posicion de 0 a 180, con esperas de 15ms
  for (pos = 0; pos <= 180; pos += 1)
  {
    myservo.write(pos);
    delay(15);
  }
  delay(1000);

  //varia la posicion de 180 a 0, con esperas de 15ms
  for (pos = 180; pos >= 0; pos -= 1)
  {
    myservo.write(pos);
    delay(15);
  }
  delay(1000);
}
```

Theremino



Theremino

```
int valorSensor; // Definimos una variable para las lecturas del sensor
int minimoSensor = 1023; // Umbral mínimo para las lecturas del sensor
int maximoSensor = 0; // Umbral máximo para las lecturas del sensor
const int ledPin = 13; // Definimos el pin del LED incorporado en la placa Arduino
```

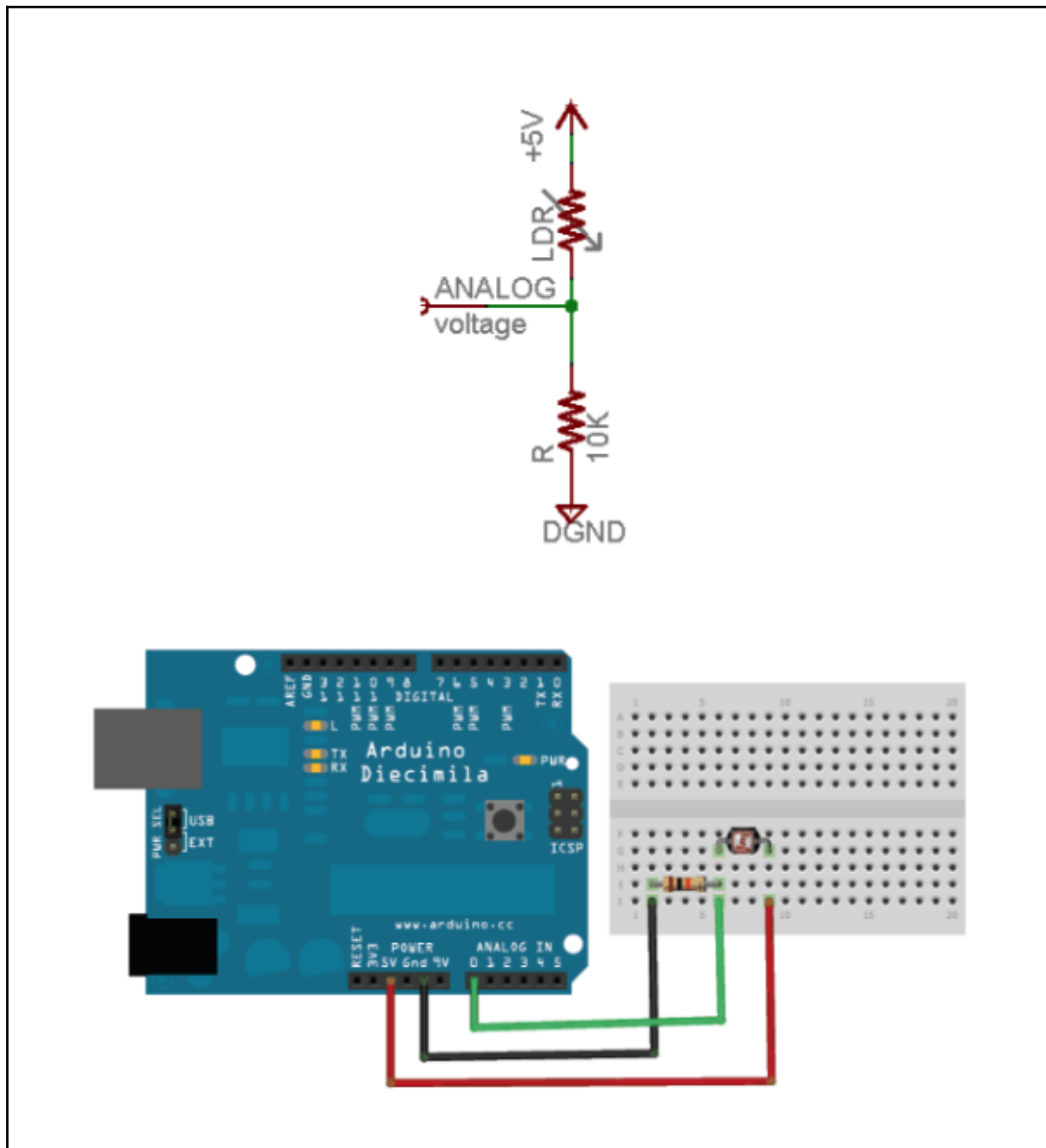
void setup() {

```
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, HIGH);
  while (millis() < 5000) {
    valorSensor = analogRead(A0);
    if (valorSensor > maximoSensor) {
      maximoSensor = valorSensor;
    }
    if (sensorValue < sensorLow) {
      minimoSensor = valorSensor;
    }
  }
  digitalWrite(ledPin, LOW);
}
```

void loop() {

```
  valorSensor = analogRead(A0); // Leemos el valor del sensor a través del pin A0
  int nota = // Definimos una variable para la nota a reproducir por el buzzer
    map(valorSensor,minimoSensor,maximoSensor, 50, 4000);
  tone(8,nota,20); // Hacemos que el buzzer ejecute la nota
  delay(10); // Esperamos 10 milisegundos antes de reiniciar el loop
}
```


Fotoresistencia



Fotoresistencia

```
int photocellPin = 0; // the cell and 10K pulldown are connected to a0
int photocellReading; // the analog reading from the sensor divider

void setup(void) {
  // We'll send debugging information via the Serial monitor
  Serial.begin(9600);
}

void loop(void) {
  photocellReading = analogRead(photocellPin);

  Serial.print("Analog reading = ");
  Serial.println(photocellReading); // the raw analog reading

  delay(100);
}
```