

Guía para controlar un relé con Arduino



Grupo de trabajo *Monitorización de un huerto con raspberry pi y arduino*
IES Arroyo de la Miel

Índice

Presentación del proyecto	2
Justificación	2
¿Qué es necesario?	2
Guía del desarrollo	3
Sensor de distancia	3
¿Cómo funciona el sensor de distancia?	3
Características técnicas del sensor	4
Probando el sensor de distancia	4
El código de prueba	5
Añadiendo el relé	7
¿Cómo funciona el relé?	7
Probando el relé	7
El código final	10
Resultado final	11
Enlaces de interés	12
Contacto	12

Presentación del proyecto

Dentro de la cultura Maker siempre es importante hacer algo que llame la atención de aquellos interesados en los proyectos **STEAM** (**S**cience, **T**echnology, **E**ngineering, **A**rts and **M**aths) y **DIY** (**D**o **I**t **Y**ourself) y de aquellos que no tienen ningún interés pero que algo creativo puede llamar su atención y poco a poco, introducirlos en este mundillo.

Con todo esto en mente en clase hemos decidido que podemos mostrar como se puede controlar cualquier cosa que conectemos a un enchufe, en este caso luces, utilizando un sensor de distancia. Todo ello estaría controlado por un Arduino. Esa es la idea de este proyecto.

Justificación

En los últimos cursos lectivos nos estamos encontrando con un creciente interés de los alumnos por las **TICs** y la **robótica**. En años anteriores se han hecho pequeños talleres sobre arduino y la acogida ha sido muy buena y el interés creciente.

Por lo tanto, hemos detectado una necesidad latente de aprender más sobre proyectos **STEAM** con **Arduino**. De ahí la principal justificación de crear un proyecto como el introducido en este documento.

El uso de la plataforma de arduino se debe a que están diseñadas con el principal objetivo de que sean utilizadas para la formación y se pueden encontrar muchos recursos para tal fin.

¿Qué es necesario?

Siempre que vamos a empezar un proyecto tenemos que planificar, realizar una proyección mental y ver qué es lo que nos va a ser necesario para completar el proyecto. En ocasiones nos equivocamos y pedimos cosas demás, nos faltan cosas o sencillamente nos confundimos al pedir algún elemento y pedimos uno que no nos sirve.

Para aquellos que quieran replicar este proyecto tenéis la tranquilidad de que alguien ya lo ha hecho antes y sabe exactamente qué es lo que os hace falta. Ese material que os hace falta lo vamos a listar a continuación.

¿Qué es necesario para realizar este proyecto?

- **Arduino uno**. Otras versiones de Arduino también podrían valer pero para este proyecto hemos utilizado la versión Arduino uno.
- **Cable usb**, para alimentar al Arduino.
- **Sensor de distancia** (en este proyecto se ha utilizado el sensor HC-SR04)
- **Cables** para hacer el conexionado entre el arduino, los sensores y la protoboard.

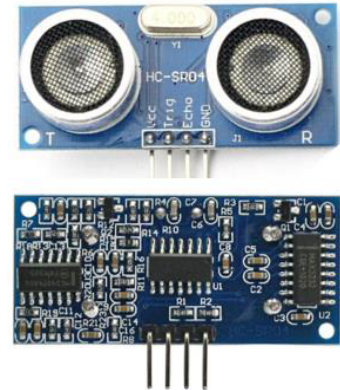
- **Protoboard**, para hacer el conexionado. También se puede soldar si se quiere hacer un proyecto que sea más definitivo.
- Un **relé**. En este proyecto se ha utilizado SRD-05-VDC-SL-C.
- Un **ordenador** con Arduino IDE para pegar el código y subirlo al Arduino.

Guía del desarrollo

En esta sección vamos a intentar describir los pasos que estamos dando por si alguien estuviese interesado en replicar el proyecto.

Sensor de distancia

El sensor de distancia que vamos a utilizar es el HC-SR04. Podemos ver en la imagen la parte frontal del sensor, donde están el emisor y el receptor de ultrasonidos, en la parte superior y la parte trasera del sensor, donde está la circuitería, en la parte inferior de la imagen.

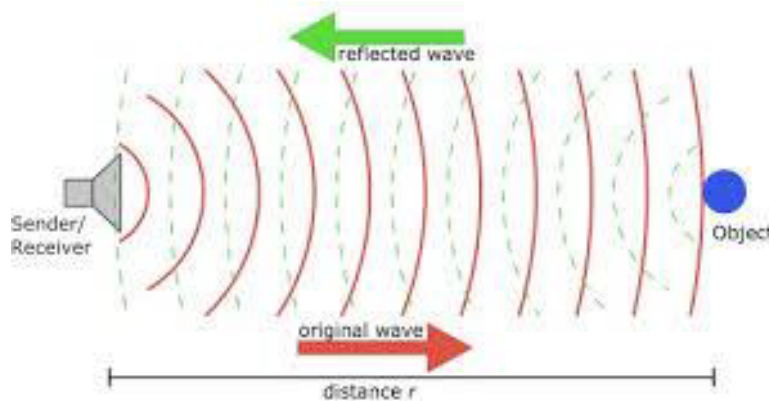


Es un sensor bastante común y que se puede encontrar fácilmente pero ¿cómo funciona?

¿Cómo funciona el sensor de distancia?

El sensor de distancia funciona utilizando el efecto eco, al igual que los sonars.

El sensor de distancia emite una señal de ultrasonido. El ultrasonido llega a un objeto y rebota. El sensor de distancia recibe de vuelta la onda de ultrasonido. En función del tiempo que haya tardado en ir y volver el ultrasonido, así de lejos estará el objeto.



Características técnicas del sensor

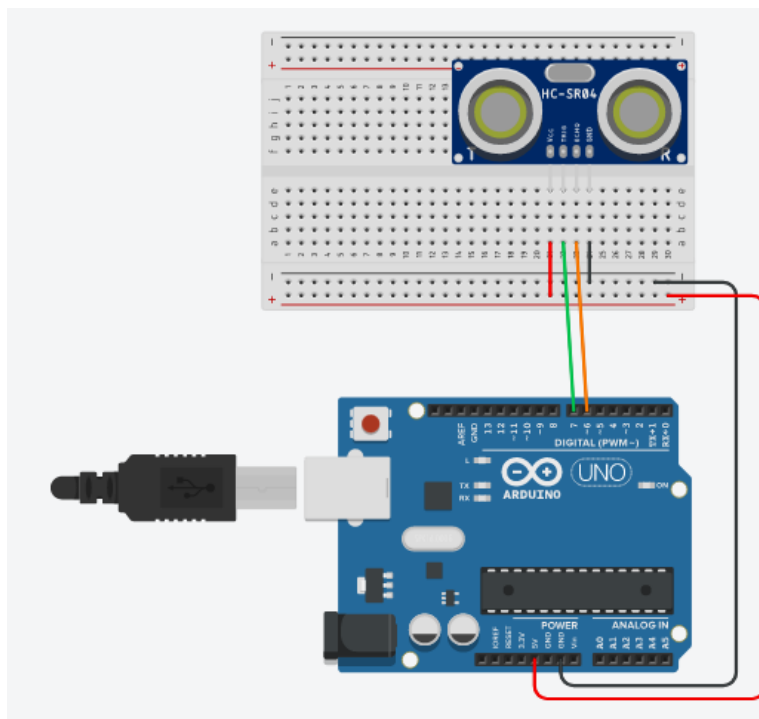
Aquí tenemos las características técnicas del sensor:

Característica	Valor
Alimentación	+5V DC
Corriente en reposo	<2mA
Corriente de trabajo	15mA
Ángulo efectivo	<15°
Rango de distancia	2cm – 400 cm
Resolución	0.3 cm

Probando el sensor de distancia

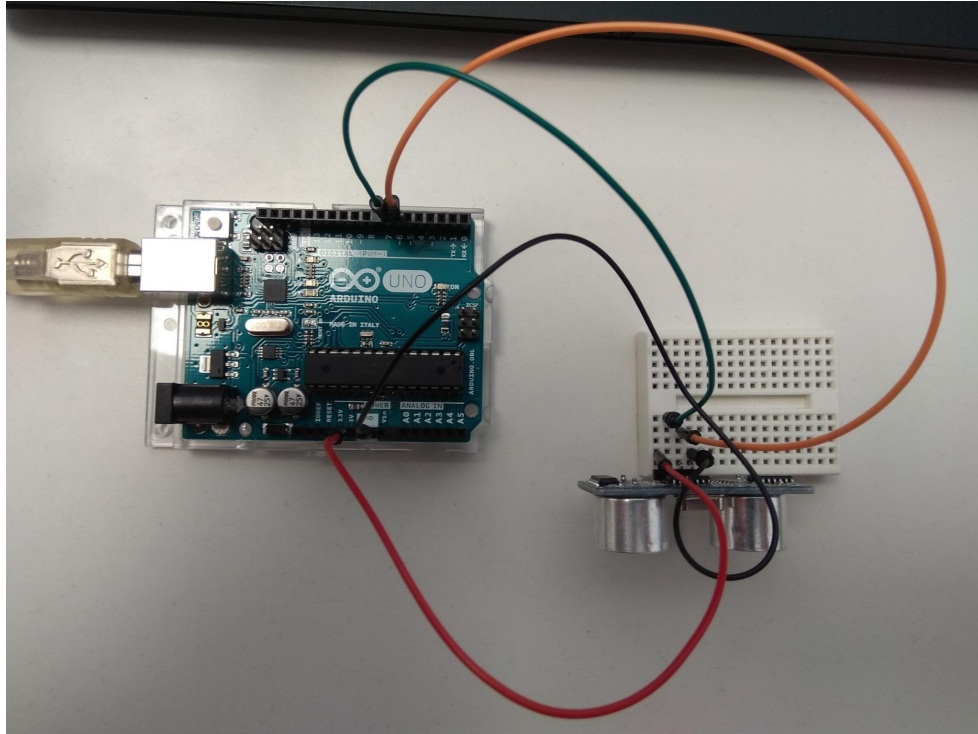
Una vez tenemos todos los materiales vamos a ir haciendo el proyecto poco a poco. En primer lugar vamos a probar el sensor de distancia y comprobar que realmente funciona bien.

En este caso vamos a conectar el sensor en la protoboard siguiendo el diseño que tenemos a continuación



El diseño muestra como realizar el conexionado de forma teórica pero hacerlo de verdad es otra cosa.

Por si hay alguna duda sobre cómo hacer el conexionado a continuación podemos ver el circuito real conectado, siguiendo el mismo patrón de colores que el que aparece en el diseño conceptual que hemos mostrado.



El código de prueba

Muy bien, ya tenemos nuestro arduino conectado al sensor de distancia utilizando la protoboard y los cables pero ¿esto no hace nada?

Efectivamente, para que esto empiece a hacer algo tenemos que cargarle un programa al microcontrolador. El programa que le vamos a cargar tendrá que indicar los pines de entrada y de salida que se utilizan y qué información vamos a mandar/leer de cada uno de ellos.

En este caso lo vamos a hacer es medir la distancia de cualquier objeto que haya delante del sensor de distancia cada 300 milisegundos y vamos a enviar al puerto serie (la forma más sencilla que tenemos de comunicarnos con el arduino) la distancia que está leyendo el sensor.

En el IDE de arduino debemos activar la ventana del puerto serie para que se muestre por ahí la información que estamos mandando.

El código fuente resultante es el siguiente.

```
const int pingPin = 7; // Trigger Pin of Ultrasonic Sensor
const int echoPin = 6; // Echo Pin of Ultrasonic Sensor

void setup() {
  Serial.begin(9600); // Starting Serial Terminal
}

void loop() {
  long duration, cm;
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(pingPin, LOW);
  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH);
  cm = microsecondsToCentimeters(duration);
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();
  delay(300);
}

long microsecondsToCentimeters(long microseconds) {
  return microseconds / 29 / 2;
}
```

Si activamos el puerto serie del arduino podemos ver una salida como la siguiente:

```
105cm
105cm
106cm
9cm
11cm
125cm
8cm
9cm
125cm
```

```
126cm
125cm
218cm
20cm
31cm
104cm
105cm
34cm
25cm
```

Podemos ver que hay distancias más cortas (cuando ponemos la mano por delante) y más lejos (cuando el límite es por ejemplo armario que hay detrás).

El sensor de distancia puede llegar hasta unos 3-4 metros de distancia.

Añadiendo el relé

De momento somos capaces de medir la distancia que hay desde el sensor hasta un objeto pero todavía no somos capaces de hacer que se enciendan las luces del árbol de Navidad cuando eso ocurra.

Para ello vamos a utilizar un relé. Para este proyecto hemos utilizado el relé SRD-05-VDC-SL-C pero cualquier relé nos puede valer.

¿Cómo funciona el relé?

Un relé nos permite controlar la activación o no de algún circuito que por lo general tiene más voltaje e intensidad que un circuito electrónico de 3,5 o 5 V. En este caso queremos encender y apagar las luces del árbol de Navidad que están conectadas a 220V.

El relé permitirá que la corriente pase cuando en la entrada de servicio tenga 5V. En ese caso los cables que estén en el otro extremo del relé se conectarán. Cuando eso ocurra también se encenderá un pequeño led del relé que nos permite ver si el relé está conectado o no. También se escuchará como el relé se activa y se desactiva.

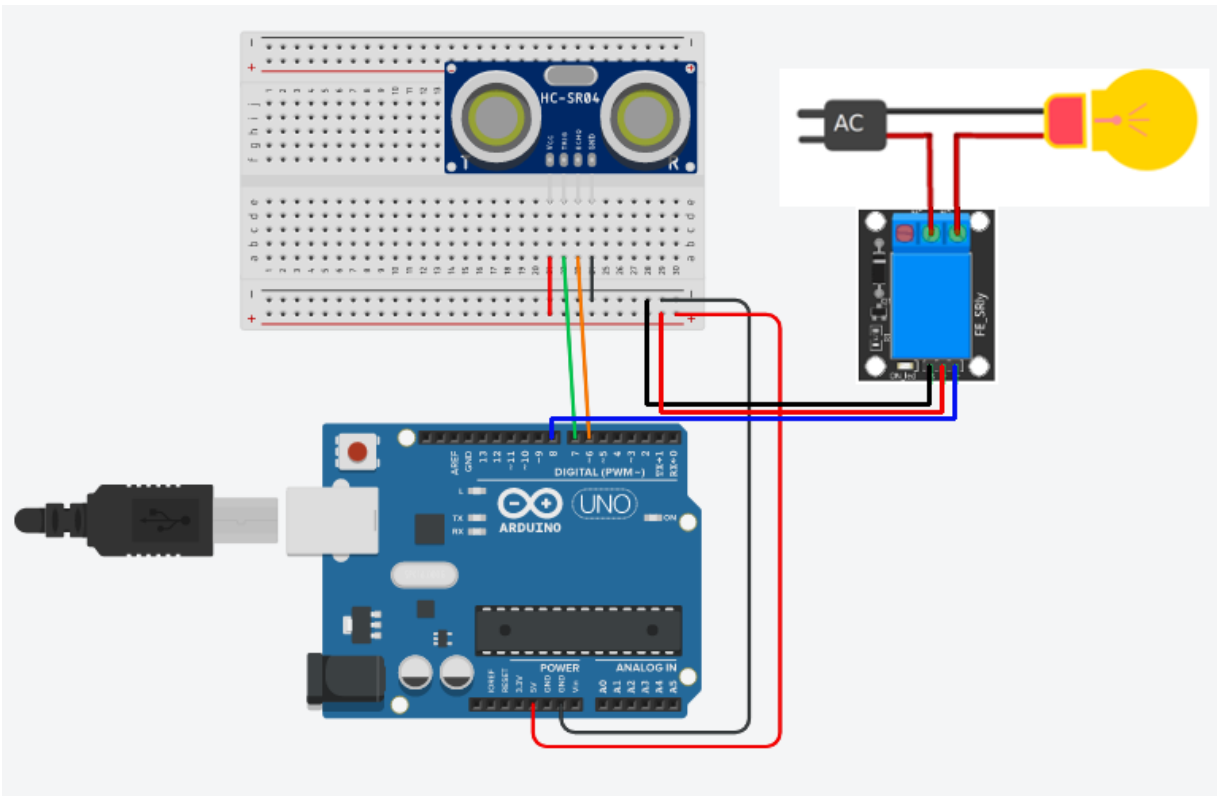
Probando el relé

Para probar el relé tenemos que hacer el conexionado de cables y pines. En este caso vamos a utilizar el pin 8 para indicar la activación o desactivación del relé. Evidentemente al relé también le tiene que llevar un cable con una toma de tierra y con una toma de alimentación de 5V para la circuitería que sacaremos de la protoboard.

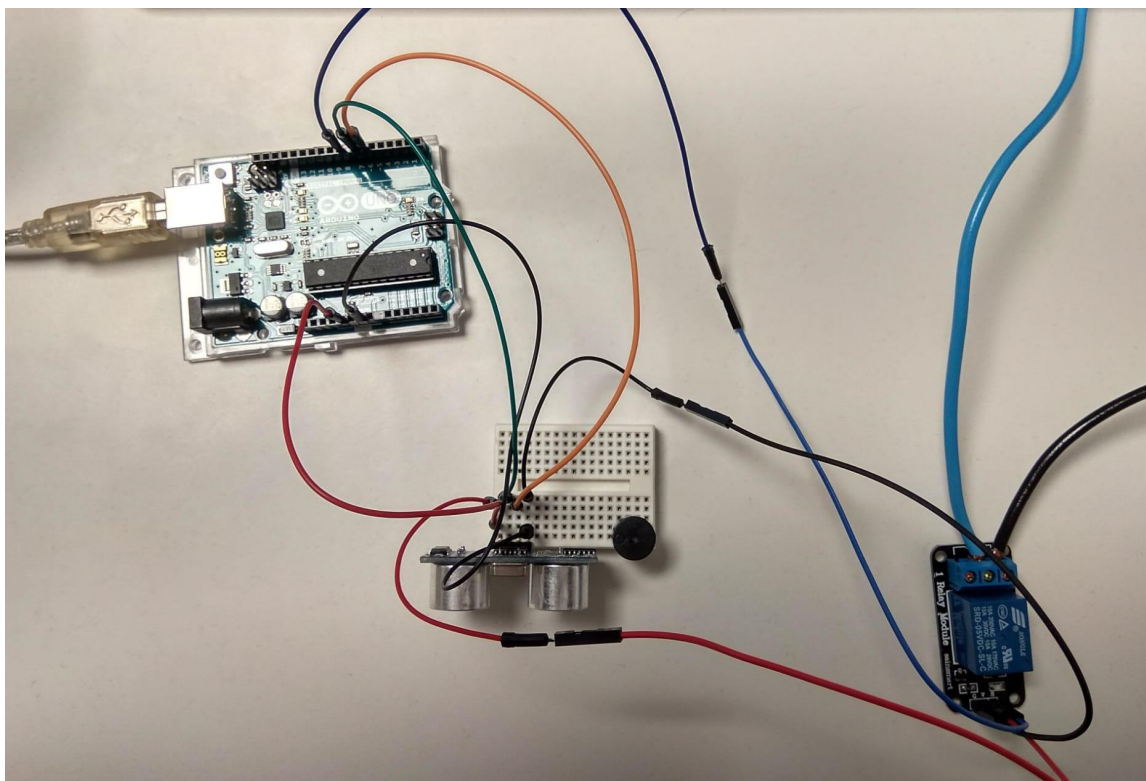
El otro extremo del relé está conectado a un cable que llevará la alimentación de las luces. Cuando la señal de activación del pin 8 esté en su valor alto el relé se accionará y comunicará ambos extremos del cable de alimentación de las luces, lo que provocará que las luces del árbol de Navidad se enciendan.

Para probar el relé no nos hace falta conectar todavía nada al otro extremo del relé. Podemos probar sin nada conectado hasta que veamos que funciona y cuando funciona le conectamos la alimentación de la luz. Esto lo podemos probar así porque el relé viene con un led que nos indica cuando está activo y cuando no. Además podremos escuchar que el relé se activa con una especie de chasquido.

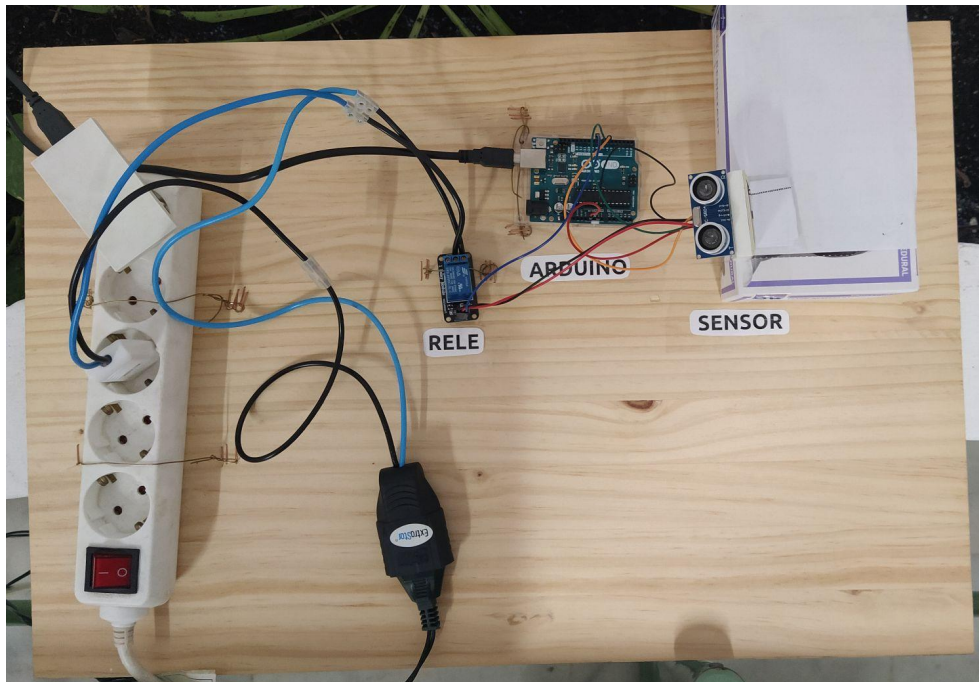
El diseño del circuito con el relé quedaría tal y como muestra el siguiente diseño.



Haciendo las conexiones en nuestro circuito real lo que veremos es algo como lo siguiente



Aquí tenemos el proyecto completo con las luces funcionando.



Y aquí el proyecto completo funcionando en el aula y con el árbol de Navidad.



El código final

Pero todavía nos falta añadir el código fuente final para controlar el relé que se enciendan las luces. En este caso lo vamos a hacer es medir la distancia de cualquier objeto que haya delante del sensor de distancia y en el caso de que sea menor que un metro (100 cm) vamos a activar el relé.

Y este es el código fuente del proyecto completo con las luces funcionando.

```
const int pingPin = 7; // Trigger Pin of Ultrasonic Sensor
const int echoPin = 6; // Echo Pin of Ultrasonic Sensor
const int relePin = 8; // Rele Pin

void setup() {
  pinMode(pingPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(relePin, OUTPUT);
  Serial.begin(9600); // Starting Serial Terminal
}

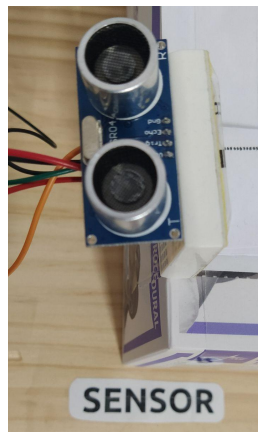
void loop() {
  long duration, cm;
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(pingPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  cm = microsecondsToCentimeters(duration);
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();
  if (cm<100) {
    digitalWrite(relePin, HIGH);
    delay(5000);
    digitalWrite(relePin, LOW);
  }
  delay(300);
}

long microsecondsToCentimeters(long microseconds) {
  return microseconds / 29 / 2;
}
```

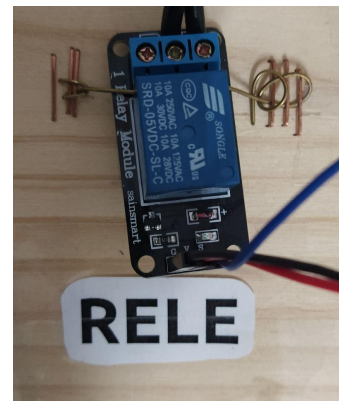
Resultado final

El resultado final del proyecto podéis verlo en las siguientes fotos.

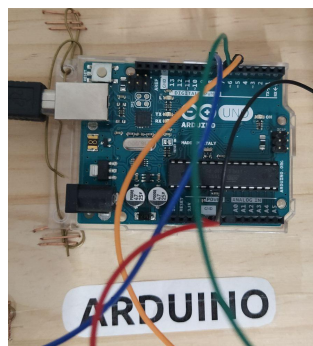
Sensor de distancia



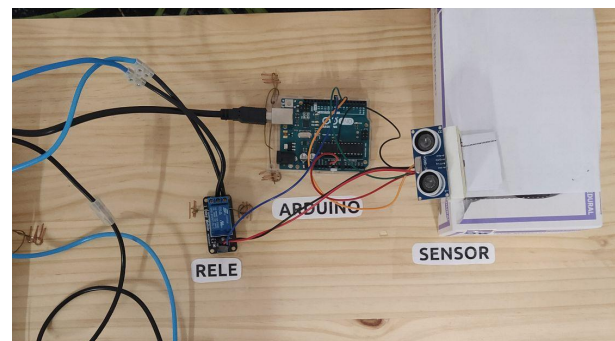
Relé



Arduino



Arduino, sensor y relé



Proyecto montado luces encendidas



Enlaces de interés

- [Cómo funciona el sensor de distancia ultrasónico HC-SR04](#)
- [Video del proyecto funcionando](#)
- [Tutorial arduino](#)
- [Página de proyectos de arduino](#)

Contacto

Si quieres comentarnos cualquier cosa del proyecto o de cualquier otro tema mandarnos un correo a profesor.diego.rosado(at)gmail.com