

MÓDULO 4:

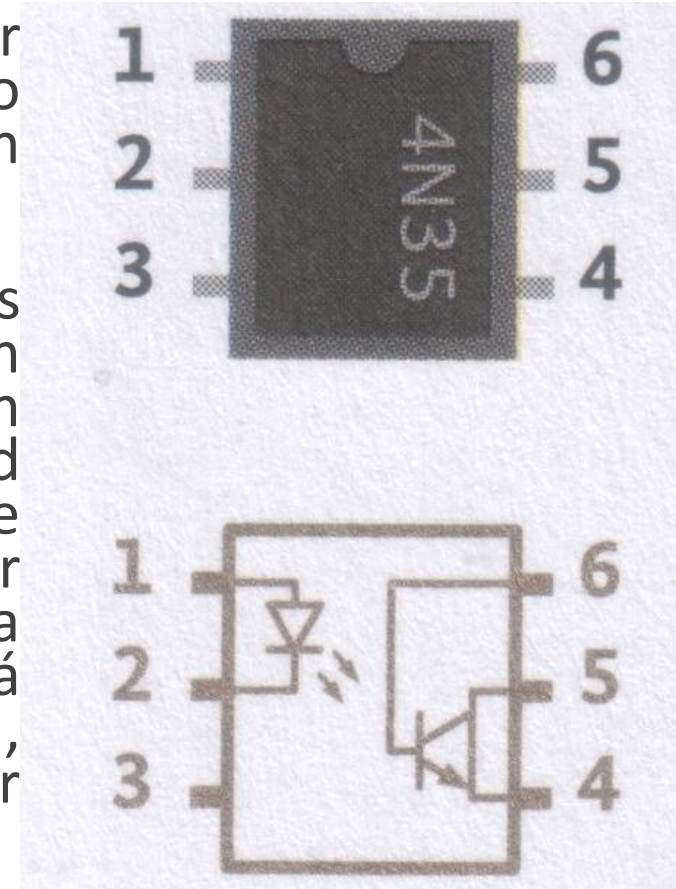
PROYECTO 14 "Retocar el logotipo de Arduino"

CURSO PROGRAMACIÓN DE PLACAS ROBÓTICAS

A solid orange horizontal bar at the bottom of the slide.

Introducción

- Obtén el control de otros dispositivos a tu alrededor presionando sus botones con Arduino. Este proyecto aprovecha un interruptor especial llamado un **optoacoplador**.
- Los optoacopladores son circuitos integrados que nos permiten controlar un circuito desde otro diferente sin ninguna conexión eléctrica entre ambos. Dentro de un optoacoplador hay un led y un sensor de luz. Cuando el led que hay dentro del optoacoplador se enciende por orden de nuestro Arduino, el sensor de luz cierra un interruptor interno. El interruptor está conectado a dos pines de salida (patillas 4 y 5) del optoacoplador. Cuando el interruptor está abierto, dichas patillas no están conectadas. De esta manera, es posible cerrar interruptores en otros dispositivos sin tener que conectarlos con nuestro Arduino.



Montando el circuito

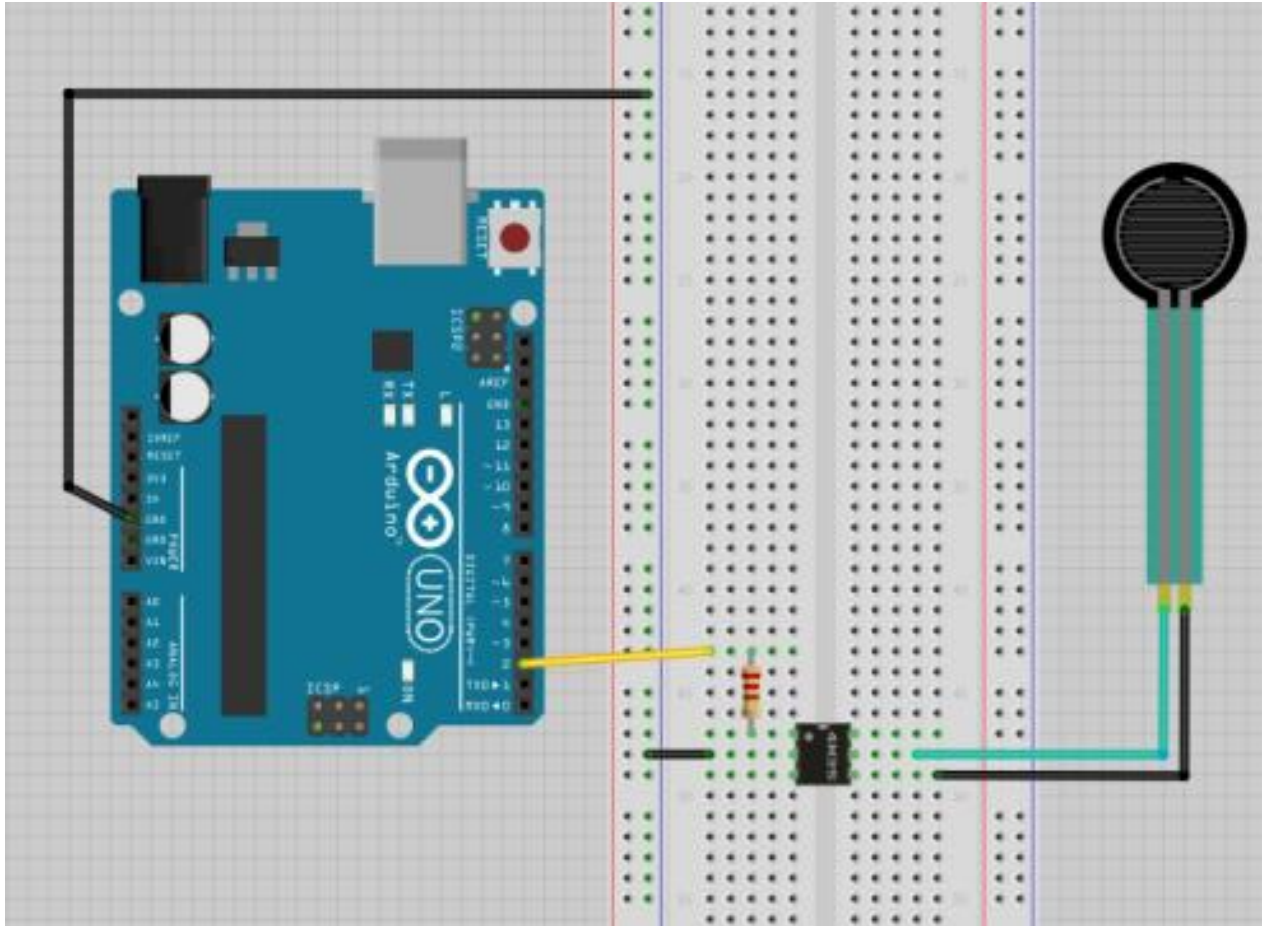
1. Lo primero de todo será conectar la tierra de nuestro Arduino a la protoboard.
2. Colocamos el optoacoplador sobre la línea central que recorre la protoboard, dejando la marca indicadora al lado izquierdo.
3. Seguidamente, unimos el pin 1 del optoacoplador al pin 2 del Arduino en serie con la resistencia de 220Ω .
4. Conectamos el pin 2 del optoacoplador a tierra.
5. En la placa del módulo de sonido hay toda una serie de componentes eléctricos, incluido un botón de reproducción. Para controlar el pulsador, tenemos que quitar el botón. Giraremos la placa del circuito y buscaremos la lengüeta o pestaña que mantienen al botón sobre el pulsador. Con cuidado, doblaremos las pestañas y sacaremos el botón de la placa del circuito.
6. Bajo el mencionado botón, existen dos pequeñas placas metálicas. Este patrón de placas es muy típico de los dispositivos electrónicos con pulsadores. Las dos “horquillas” de dicho patrón son los dos terminales del pulsador. Una pequeña lámina metálica dentro del mismo conecta ambas horquillas cuando pulsamos el botón.

Montando el circuito

7. Cuando las dos horquillas se unen, el pulsador queda cerrado. Nosotros nos encargaremos de cerrarlo mediante el optoacoplador.
8. Este método de cerrar un pulsador utilizando un optoacoplador, solo funciona si uno de los dos lados del pulsador está conectado a masa en nuestro dispositivo. Si no estamos seguros, cogeremos un multímetro y mediremos la tensión entre una de las horquillas y la masa del dispositivo. Esta medición debemos llevarla a cabo con el dispositivo encendido, por lo que tendremos que prestar mucha atención en no tocar nada más de la placa.
9. Una vez que ya sabemos que la horquilla está conectada a tierra, desconectamos la alimentación de nuestro dispositivo.
10. A continuación, conectamos un cable a cada una de las horquillas. Si los vamos a soldar, tenemos que tener cuidado para no unir ambas partes del pulsador. Si por el contrario, optamos por unirlos con una cinta, debemos asegurarnos de que la sujeción sea firme, o de lo contrario el pulsador no se cerrará. Tenemos que asegurarnos de que ninguno de los dos cables toca ninguna otra horquilla, o nuestro pulsador siempre estará cerrado.
11. El siguiente paso será conectar los dos cables anteriores a los pines 4 y 5 del optoacoplador. El lado del pulsador que está conectado a tierra lo uniremos al pin 4 del optoacoplador y el otro al pin 5.

Montando el circuito

Componentes	
Potenciómetro	1
Optoacoplador 4N35	1



Comentario del código

- La parte más divertida de este proyecto se encuentra en el circuito y en el optoacoplador. Vamos a reproducir un sonido cada 20 segundos dándole al pin 2 un valor HIGH. Así que lo primero que haremos es crear una constante para el pin de control del optoacoplador.
- En la función **setup()**, definiremos el pin del optoacoplador como salida.
- A continuación, entramos en la función **loop()**. Dicha función establece el valor de optoPin en **HIGH** durante unos milisegundos, tiempo suficiente para que el optoacoplador cierre el pulsador del dispositivo. A continuación, optoPin tomará el valor **LOW**.
- Para finalizar, esperamos 21 segundos para que el sonido se reproduzca y el bucle **loop()** comience de nuevo.