

MÓDULO 4: PROYECTO 10"Zoótrofo"

CURSO PROGRAMACIÓN DE PLACAS ROBÓTICAS

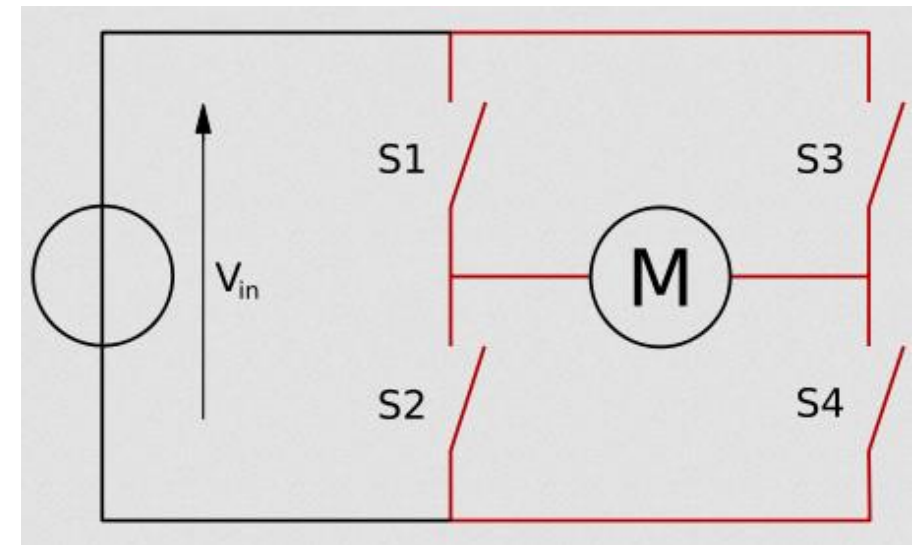
A solid orange horizontal bar at the bottom of the slide.

Introducción

- En este proyecto, se hará una imagen en movimiento a partir de una serie de imágenes fijas que giran alrededor de un eje. No sólo va a utilizar un motor para hacer girar la serie de imágenes, sino que también será capaz de funcionar el motor al revés usando un circuito integrado llamado Puente-H.
- Un zoótropo no es más que un cilindro que incluye una serie de ranuras en su superficie. Cuando dicho cilindro gira y miramos a través de las ranuras, nuestros ojos perciben las imágenes estáticas que hay en el interior como una única imagen en movimiento. Las ranuras impiden que las imágenes del interior se mezclen y las veamos borrosas, y el movimiento del cilindro es el que provoca la sensación de imagen animada. Normalmente estos aparatos se hacían girar con la mano, aunque otros incluían un sistema de engranajes que facilitaba dicha tarea.

Introducción

- En nuestro proyecto del molinillo de papel motorizado, hacíamos girar el motor en una única dirección. Si cogemos e invertimos la alimentación y la tierra del motor, lograremos que el motor gire en sentido contrario. Aunque efectivo, no es muy práctico estar haciendo esto cada vez que queramos cambiar el sentido de giro del motor, aunque por suerte para nosotros existe un componente que nos va a facilitar dicha tarea: el **Puente en H**.



Introducción

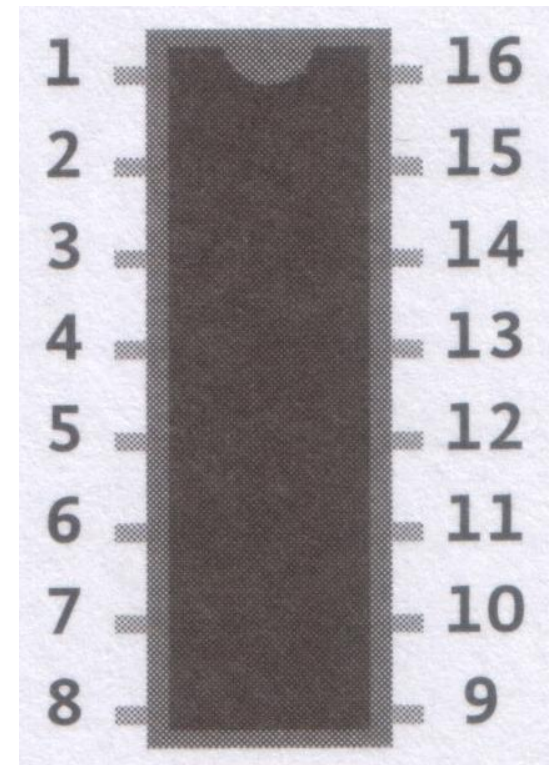
- **El Puente en H**

- Es un elemento que podemos encontrar en forma de **circuito integrado** (CI). Los circuitos impresos son componentes que pueden llegar a contener una gran cantidad de circuitos en una pastilla muy pequeña. Esto ayuda a simplificar circuitos mucho más complejos ya que se pueden reducir a un único componente que es fácilmente sustituible.
- A modo de ejemplo, el Puente en H que vamos a utilizar es nuestro proyecto contiene cierto número de transistores. Si quisiéramos montarlo a mano, necesitaríamos otra protoboard solo para él.



Introducción

- En un circuito integrado, podemos acceder a sus circuitos internos mediante los pines que hay a ambos lados del mismo.
- Diferentes CI tienen diferentes números de pines, y no todos ellos se utilizan en todos los circuitos. Normalmente se hace referencia a los pines mediante un número en lugar de hacerlo por su función. Cuando miremos un circuito integrado, debemos hacerlo posicionando hacia arriba una pequeña marca en forma de semicírculo que tiene en un extremo.
- Colocando el CI de esta manera, podremos identificar el número de cada pin comenzando por el que se encuentra en la parte superior izquierda y haciendo un recorrido en forma de “U”.

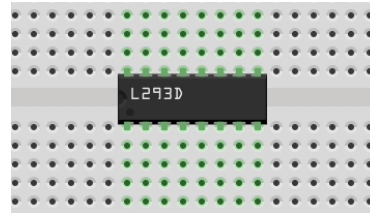


Montando el circuito

1. Conectamos la alimentación y la tierra de nuestro Arduino en la protoboard.
2. Añadimos un pulsador conectando la patilla de entrada a la alimentación. Seguidamente, una de las patillas de salida la conectamos a tierra mediante una resistencia de $10K\Omega$ y la otra patilla al pin número 5 de nuestro Arduino. Este pulsador será el que nos permita encender o apagar el motor.
3. Repetimos el paso anterior con el otro pulsador, pero esta vez, en lugar de conectarlo al pin 5 lo conectaremos al pin número 4. Este pulsador nos permitirá controlar el sentido de giro del motor.
4. A continuación conectaremos el potenciómetro. Dicho componente tiene tres patillas. La de un lado la conectaremos a la alimentación de 5V proporcionada por Arduino y la del otro lado, la uniremos a tierra. La patilla central la conectaremos al pin de entrada analógica 0 de nuestro Arduino. Conectado de esta forma, este potenciómetro nos permitirá controlar la velocidad del motor.

Montando el circuito

5. El siguiente componente que colocaremos será el Puente en H. Lo posicionaremos horizontalmente sobre la línea media de la protoboard dejando la muesca de dicho componente orientada hacia el lado izquierdo, de manera que desde nuestro punto de vista, la mitad de las patillas quedan en la mitad superior de la protoboard y la otra mitad en la inferior. La siguiente figura ilustra como debería de quedar.



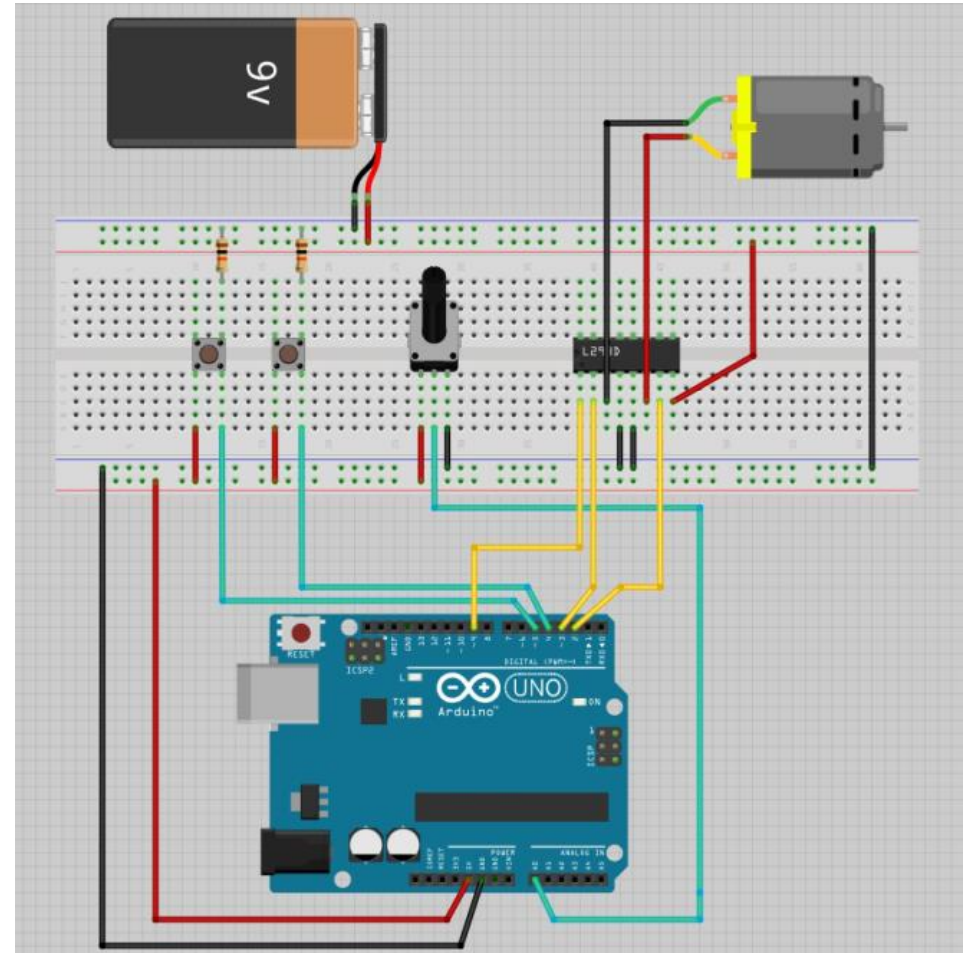
6. Una vez colocado el Puente en H, conectamos el pin 1 del puente al pin digital 9 del Arduino. Este es el pin de activación del Puente en H. Cuando a través de dicho pin reciba 5V, el motor se encenderá y cuando reciba 0V, lo apagará. Este pin del Puente en H lo podemos usar con la técnica PWM y ajustar de esta forma la velocidad del motor.
7. Seguidamente, conectamos el pin 2 del puente con el pin digital 3 del Arduino, y el pin 7 al pin digital 2. Estos pines serán los que utilizemos para comunicarnos con el Puente en H y decirle cual será el sentido de giro del motor. Si el pin 3 tiene un nivel **LOW** y el pin 2 un nivel **HIGH**, el motor girará en un sentido. Pero si el pin 3 tiene un nivel **HIGH** y el 2 un nivel **LOW**, el motor girará en sentido contrario. Por contra, si ambos pines tienen el mismo estado de forma simultánea, ya sea **HIGH** o **LOW**, el motor dejará de girar.

Montando el circuito

5. Ahora debemos alimentar a nuestro CI. Para ello conectaremos el pin 16 a la alimentación de 5V y los pines 4 y 5 a tierra.
6. A continuación añadiremos el motor. Dicho motor lo uniremos a los pines 3 y 6 del Puente en H, ya que estos serán los encargados de encender o apagar el motor en función de las señales recibidas a través de los pines 2 y 7.
7. El paso siguiente será conectar el adaptador de la batería de 9V, pero no la propia batería. La tierra de la alimentación de 9V irá unida a la tierra de nuestro Arduino y el positivo, a la línea de alimentación de la protoboard que nos quede libre. Para mi gusto, las dos líneas inferiores de la protoboard las uso para el Arduino y las dos superiores para la alimentación de 9V. El pin 8 del Puente en H lo conectamos al positivo de la batería, ya que este pin será el encargado de alimentar al motor. Sobre todo debemos asegurarnos de que las líneas de alimentación tanto del Arduino (5V) como de la batería (9V) no estén unidas, solo sus tierras deben estarlo.

Montando el circuito

Componentes	
Pulsadores	2
Potenciómetro	1
circuito integrado L293D	1
Batería de 9V	1
Resistencias 10 K Ω	2
Conector para enlazar la batería a la protoboard	1
Motor DC	1



Comentario del código

- Comencemos por comentar paso a paso el código fuente con el que vamos a programar nuestro Arduino.
- Para empezar, crearemos unas constantes para los pines de entrada y salida.
- A continuación, usaremos variables para almacenar el valor de las entradas. En nuestro programa, tendremos que detectar los cambios de estado que se produzcan en los pulsadores de una iteración del bucle a otra, similar a lo que hicimos en el proyecto del reloj de arena digital. Por tanto, además de tener que almacenar el estado actual de los pulsadores, también tendremos que guardar el estado anterior de cada uno de ellos.
- Seguidamente, emplearemos dos variables para el motor. **motorDirection** nos indicará en que dirección está girando el motor y, **motorEnabled**, nos dirá si el motor está girando o no.

Comentario del código

- En la función **setup()**, nos encargaremos de indicar que pines son de entrada y cuales de salida.
- Al pin de activación le daremos un valor **LOW**, de esta manera, empezaremos teniendo al motor parado.
- A continuación, en la función **loop()**, leeremos el estado del pulsador de On/Off y lo guardaremos en la variable **onOffSwitchState**.
- Si existe alguna diferencia entre el estado actual del pulsador y su estado previo y además, el pulsador esta actualmente en un nivel **HIGH**, le daremos a la variable **motorPower** el valor 1. Si el estado es **LOW**, le daremos el valor 0.
- Leeremos los valores del pulsador de dirección y del potenciómetro y los guardaremos en sus respectivas variables.

Comentario del código

- Seguidamente, comprobaremos si el pulsador de dirección está actualmente en una posición diferente a la anterior. Si es diferente, cambiaremos el valor de la variable de dirección del motor. Solo existen dos sentidos de giro del motor, por lo que la variable tendrá que alternar entre uno u otro. Una manera de conseguir esto es utilizando el operador de inversión de forma similar a esta: **motorDirection = !motorDirection**.
- La variable **motorDirection** determina el sentido de giro del motor. Para establecer la dirección, debemos marcar uno de los pines de control con valor **HIGH** y el otro con valor **LOW**. Cuando la variable **motorDirection** cambia, debemos invertir el estado de dichos pines.
- Si el pulsador de dirección ha sido pulsado, debemos hacer girar el motor en sentido contrario invirtiendo el estado de **controlPins**.

Comentario del código

- Si el valor de la variable `motorEnabled` es 1, encenderemos el motor utilizando la función **`analogWrite()`** en el pin de activación. Si **`motorEnabled`** es 0, apagaremos el motor indicándole a **`analogWrite`** un valor 0.
- Antes de salir de la función **`loop()`**, guardaremos el estado actual de los pulsadores como estado anterior para poder utilizar dichos valores en la próxima iteración del programa.
- Referencias
 - Más información acerca del **Puente en H**:
[https://es.wikipedia.org/wiki/Puente_H_\(electrónica\)](https://es.wikipedia.org/wiki/Puente_H_(electrónica))
 - Más información acerca del **potenciómetro**:
<https://es.wikipedia.org/wiki/Potenciometro>