5. Robótica

Robótica

En esta capítulo vamos a dar movimiento a nuestro Arduino.

Si tienes dudas técnicas en este capítulo pon un ticket a http://soporte.catedu.es/ y te ayudaremos:

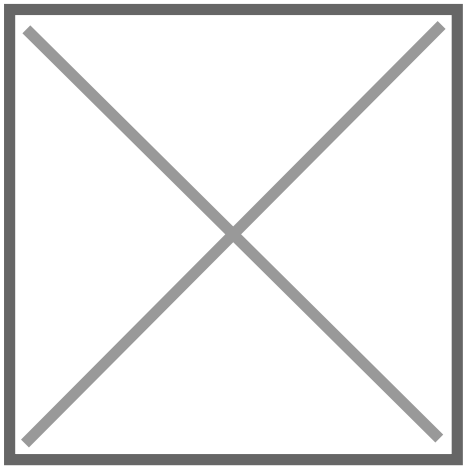
Servomotores

Una de las aplicaciones más utilizadas de los sistemas de control por ordenador y en la robótica están asociados con los motores, que permiten accionar o mover otros componentes, como puertas, barreras, válvulas, ruedas, etc. Uno de los tipos que vamos a ver en este capítulo son lo servos, hay de dos tipos:

- El **servomotor** o **servos convencionales** que posee la capacidad de posicionar su eje en un ángulo determinado entre 0 y 180 grados en función de una determinada señal.
- **Servo de rotación continua** Son servos por fuera igual que los anteriores, pero pueden girar 360º y se controlan por tiempo

> Por defecto cuando se dice **servo**, es un **servomotor o servo convencional**

Servos de rotación continua



Para controlar un servo de rotación continua, las instrucciones a realizar son :

- Incluyes la librería de servos **#include <Servo.h>**
- Declaras una variable servo **Servo myservo;** //puedes poner el nombre que quieras p.e. miservo
- En *setup()* tienes que decir a qué pin está conectado **myservo.attach(9);** //por ejemplo pin 9
- Y en *loop()*
  - **myservo.write(90)**; //significa servo parado

- **myservo.write(180);** //significa servo funcionando al 100% en el sentido de las agujas del reloj
- **myservo.write(0);** //significa servo funcionando al 100% en el sentido contrario de las agujas del reloj

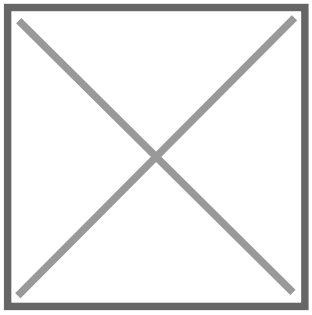
Mira el vídeo, esta realizado con otra shield ECHIDNA y con bloques mBlock (curso Echidna https://libros.catedu.es/books/echidna/) fíjate como:

- Los extremos 0º y 180º es a máxima velocidad, pero un sentido u otro.
- 90º es parado.
- Un valor intermedio es menos velocidad (se ve el ejemplo 80º y 100º) -
- Si tiene deriva, (cosa frecuente) tienen un potenciómetro para ajustar.

https://www.youtube.com/embed/Z-5SerXmRY0

> Si quieres saber más sobre servomotores te recomendamos estas paginas del Zaragozano Luis LLamas: Servomotores convencionales y Servomotores de rotación continua

Servomotores o servos convencionales



Los servos son un tipo especial de motor en el que se añade una circuito lógico electrónico que permite un control mucho más preciso que a un motor normal de corriente continua. Esto les permite posicionar el eje en un ángulo determinado.

El hardware interno se compone de un potenciómetro y un circuito integrado que controlan en todo momento los grados que gira el motor. De este modo, en nuestro caso, desde Arduino, usando las salidas digitales PWM podremos controlar fácilmente un servo. Lo ideal es conectarlo a 6V pero trabajan bien en los 5V del Arduino.

Hay muchos modelos, en robótica educativa cuestan entre 1-5€, el más común es el SG90, muy barato, pero tiene muy poca fuerza, el MG90S tiene algo más, si queremos algo más, ya tiene que

ser el MG996R pero ya este modelo **NO se puede conectar directamente al Arduino o Raspberry,** el pico de energía que necesita, provoca el reinicio de la placa. Incluso varios pequeños SG90.

Las instrucciones son las mismas que los servos de rotación continua, pero los valores que se proporcionan son los grados que se desean.

- Incluyes la librería de servos **#include <Servo.h>**
- Declaras una variable servo **Servo myservo;** //puedes poner el nombre que quieras p.e. miservo
- En *setup*() tienes que decir a qué pin está conectado **myservo.attach(9);** //por ejemplo pin 9
- Y en *loop*()
  - **myservo.write(90)**; //Posición 90º (posición por defecto)
  - **myservo.write(180);** //Posición 180º
  - **myservo.write(0);** // Posición 0º

La instrucción *myservo.write(angulo)* envía por el pin digital declarado en *myservo.attach()* pulsos cuadrados de 50Hz y de anchura el estado alto proporcional al ángulo que se desea.

- Un pulso de 0.5-1ms es 0º
- Un pulso de 1.5 ms es 90º
- Un pulso de 2-2.5ms es 180º

Si quieres saber más, te recomendamos https://www.luisllamas.es/controlar-un-servo-con-arduino/

https://sketchfab.com/models/6e5ff0e57708426b87ea8cf2edfbb2cc/embed

Arduino Servomotor by **Marco De Simone** on **Sketchfab**

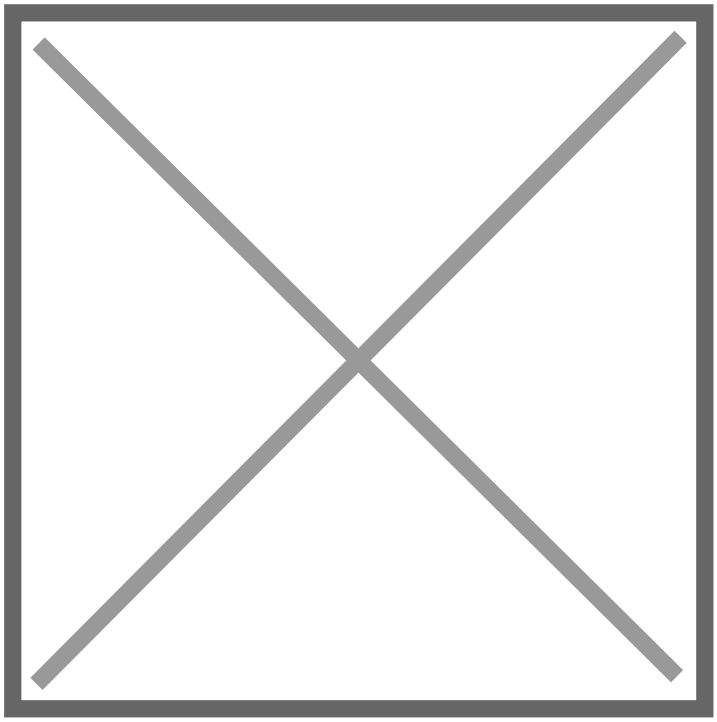
Control de servomotores

Montaje 25: Testea tu servo (servomotor)

En el siguiente programa de testeo (fuente: forum arduino) vamos a probar el servo.

- Conecta el servo al pin 7
- Utiliza el puerto serie para teclear el ángulo que quieras con el teclado de tu ordenador.
- No queremos que entiendas todo el código, pues el puerto serie lee es caracteres ASCII y tiene que convertir el carácter a ángulos.
- Si tecleas un valor más grande de 500 se le indica al servo no el ángulo que se tiene que mover, sino cuanto tiempo en ms se tiene que mover.

Montaje 25: Conexión sin Edubásica



La conexión se realiza mediante 3 cables: 2 de alimentación (+5V/GND) y un tercero, conectado por ejemplo el 7 , donde indicaremos los grados que queremos que gire a través de un programa en Arduino.

Montaje 25: Conexión con Edubásica

En Edubásica tenemos una forma muy sencilla de conectar un servo a la tarjeta. Lo puedes hacer mediante las clavijas identificadas con JP3. De arriba abajo las conexiones son:

- Señal (pin7)
- +Vin
- GND

Recuerda que siempre puedes utilizar los pines analógicos como E/S digitales, del pin 14 al 19.

Por ejemplo, puedes conectar el servo al pin analógico 5, pero declarado como digital en el 19.

Montaje 25: video

https://www.youtube.com/embed/b7JWiL-tucg?rel=0

Por ejemplo en este Servo **HD-1440A** con el anterior programa se ve que es un servo barato:

- No puede hacer ángulos de +180º luego es un servo convencional
- No puede hacer ángulos de menos de 10º no llega a parar, o sea tiene deriva.

Si eliges uno un poco más caro como el **MG90S** no tiene estos problemas en los extremos. Ver

Montaje 25: simulación

Aquí lo tienes simulado en Tinkercad https://www.tinkercad.com/things/4FQNFOdoS8Z-25testeatuservo

https://www.tinkercad.com/embed/4FQNFOdoS8Z?editbtn=1

Pincha en **simulación** y luego **código**, y encontrarás el cuadro **monitor seri**e donde puedes teclear el ángulo que quieras. Observa al servo cuando pulses Intro

servo1arduino.png

Montaje 25: Programa

https://create.arduino.cc/editor/javierquintana/1d59ad76-dc4a-4b6d-8238-66ae593ce728/preview

https://create.arduino.cc/editor/javierquintana/1d59ad76-dc4a-4b6d-8238-66ae593ce728/preview?embed

Montaje 26 controlando el servo

Vamos a jugar un poco, montar este circuito de tal manera que el servo este haciendo este ciclo

- Un segundo en su posición inicial 90º
- Un segundo en 180º
- Un segundo a 45º

El programa lo puedes conseguir en el botón Code de la siguiente simulación

https://www.tinkercad.com/embed/3sAd2FZ2opr?editbtn=1

Montaje 27 controlando el servo

Vamos a simular un motor paso a paso. El servo ahora tiene que pasar de 10º a 180º de 10 en 10 parando 1 segundo en cada paso

https://www.youtube.com/embed/lfpJzziVuFw

El programa lo puedes conseguir en el botón Code de la siguiente simulación

https://www.tinkercad.com/embed/bJlRTA5Lsha?editbtn=1

Montaje 28 servo y potenciómetro

Podemos probar una aplicación muy importante que está basada en mover el servo según una determinada entrada analógica. Este nos puede ser muy útil si queremos controlar servos por medio de joysticks por ejemplo o cualquier dispositivo que cuente con potenciómetros para realizar un movimiento.

El código está obtenido desde los ejemplos que vienen incluido en la IDE de de Arduino (Knob) que encontrarás en: Archivo->Ejemplos->Servo, sólo hemos cambiado esta línea: **myservo.attach(7)** ;

Lo que hace este programa es variar la posición del servo enfunción de la posición del potenciómetro que leemos de manera analógica.

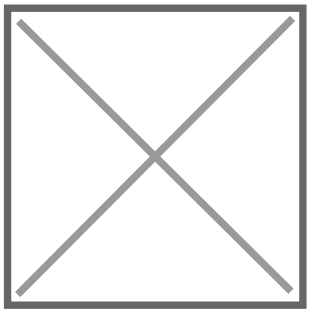
Sólo nos queda mapear la lectura para que se mueva de 0 a 180º.

https://www.youtube.com/embed/_knaXSFBF_M

En la siguiente simulación, puedes mover el potenciómetro y ver el resultado

https://www.tinkercad.com/embed/gl6syqapJHe?editbtn=1

Motores DC



Arduino (UNO) no proporciona corriente suficiente para hacer funcionar, en condiciones "dignas", un motor de corriente continua de los que solemos usar en el aula-taller de Tecnologías (motor DC). Conviene alimentar Arduino con una fuente externa (7 -12 V) para poder proporcionar intensidad necesaria para el par motor requerido en los proyectos.

Existen "shields" o tarjetas que se encajan sobre Arduino y le añaden funciones específicas como mover motores DC.

Alimentación

Como vimos en la sección de Alimentación eléctrica de Arduino (ver https://libros.catedu.es/books/programa-arduino-mediante-codigo/page/hardware) no es recomendable alimentar Arduino, cuando se trabaja con elementos de "alto" consumo como pueden ser los motores DC, con el cable USB. Tenemos la posibilidad de proporcionar más corriente (mA) a través de la conexión jack de Arduino. En el pin Vin tendremos una salida del voltaje que apliquemos por el jack que servirá para alimentar a los motores. Si el motor es pequeño y trabaja sin carga, sí que se puede conectar diréctamente al Arduino.

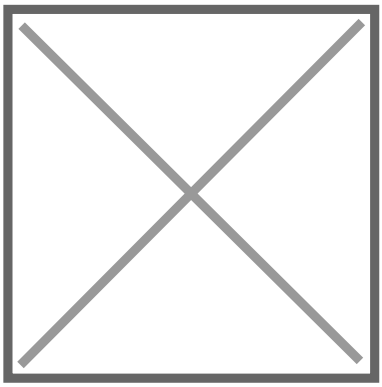
Recomendamos las siguientes páginas de Luis Llamas: Tipo de motores y Motores paso a paso

Montajes que vamos a realizar

Vamos a controlar los motores de 4 formas:

- Montaje 29 motor con transistor con/sin la shield Edubásica
- Montaje 30 circuito integrado L293 con/sin la shield Edubásica
- Montaje 31 un coche loco con/sin la shield Edubásica
- Montaje 32 un coche teledirigido con el móvil con/sin la shield Edubásica
- Montaje 33 un coche teledirigido con la voz con/sin la shield Edubásica

Montaje 29 motor con transistor



El transistor es un componente analógico con infinidad de aplicaciones. Como se observa en la imagen, es un dispositivo con tres pines llamados emisor (E), colector (C) y base (B). El funcionamiento a groso modo es sencillo: Si suministramos "cierta" cantidad de corriente al terminal llamado base, entre los terminales emisor y colector circulará una corriente proporcional a la que entra por la base. Puede ser unas 100 veces mayor (este valor dependerá del transistor y se llama ganancia), por lo tanto, lo que tenemos es un amplificador de corriente, o sea, que con una intensidad pequeña podemos obtener una más grande entre los terminales emisor y colector. Si a estos dos terminales conectamos un motor DC conseguiremos hacerlo girar con garantías. Cuando no hay corriente en la base (o es muy pequeña) el colector y emisor están desconectados y no circulará corriente entre ellos.

> Nuestra propuesta es que realices un montaje de mover un motor a través de un transistor. Que funcione 1 segundo, luego 1 segundo parado y así sucesivamente.

Montaje 29 Programa motor con transistor

Este es el programa que te proponemos: https://create.arduino.cc/editor/javierquintana/c2d6681b-76d7-4231-9f8b-d0110348b7d4/preview

https://create.arduino.cc/editor/javierquintana/c2d6681b-76d7-4231-9f8b-d0110348b7d4/preview?embed

Montaje 29 Esquema de montaje motor con transistor con Edubásica

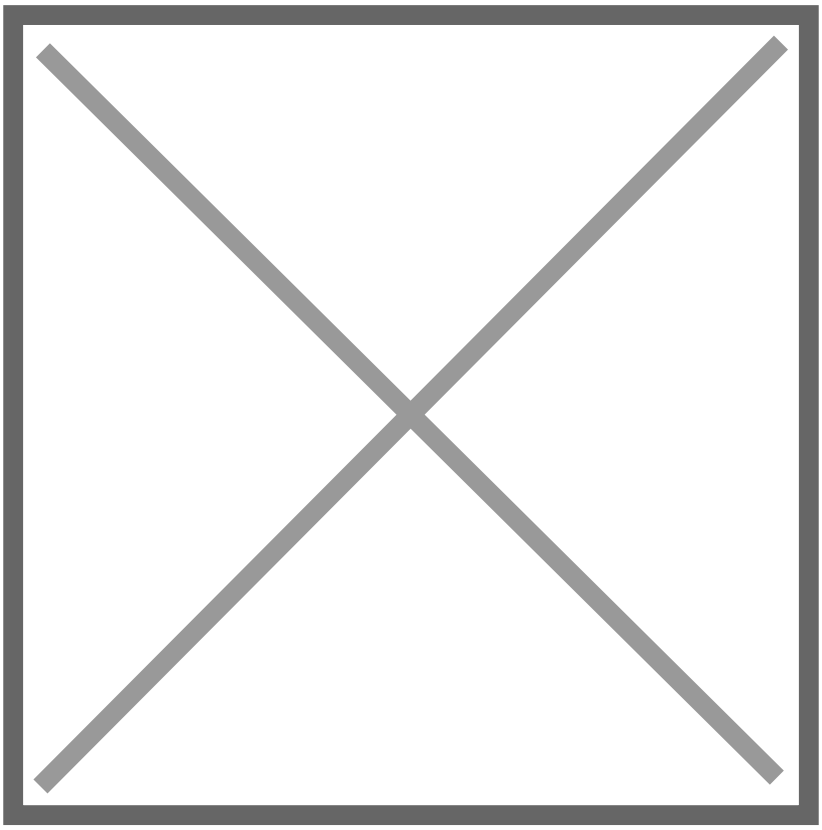
La shield de Edubásica ya tiene incorporado el transistor y los diodos de protección para funcionar el motor, para ello tenemos que hacer:

- Poner el interruptor en ON
- Conectar el motor en la salida del transistor y Vin
- Utilizar D6 como pin de control del motor

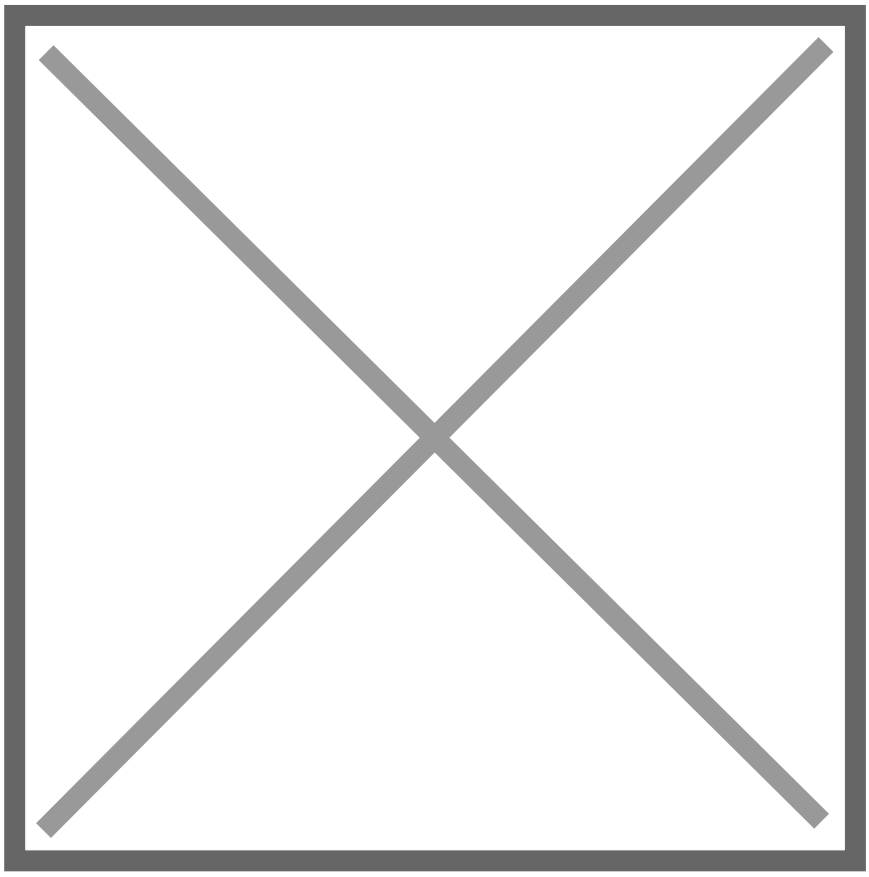
https://www.youtube.com/embed/enflaqwsDhg

Montaje 29 Esquema de montaje motor con transistor sin Edubásica

Desde Arduino (salida digital) actuamos sobre la base si enviamos un HIGH al pin digital donde la conectemos (pin 6 en el esquema). El transistor tiene que tener suficiente ganancia. Se conecta un diodo de protección en antiparalelo (1N004). Cuando el motor se para las bobinas se desmagnetizan y se descargan de energía eléctrica. El diodo proporciona un camino para su descarga (la energía se disipa en forma de calor en el diodo) y así se evita que sufra el transistor.



Es decir, sería esta vista :



> Importante: NO CONECTES EL MOTOR DIRÉCTAMENTE AL ARDUINO, PODRÍAS DAÑARLO

Montaje 29 Simulación de montaje motor con transistor sin Edubásica

En el momento de realizar este circuito, Tinkercad no tiene un Jack de conexión, luego hemos conectado directamente la pila al Vin

https://www.tinkercad.com/embed/5jwVKqxbivr?editbtn=1

¿Te atreves?

Otra posibilidad que protege el transistor es, en vez de conectar directamente el motor al colector del transistor, ubicar un relé en esa posición y accionar el motor con alimentación independiente con la conmutación del relé. ¿Te atreves a montar un montaje que active el motor DC mediante un relé conectado al colector del transistor.?

Montaje 30 motor con Circuito L293

El circuito integrado L293 permite controlar motores DC de pequeña potencia, pero en general bobinas, o sea cualquier elemento que precise pico de potencia, como los relés, etc.. Para utilizarlo hay que hacer un montaje externo a Arduino, en una placa de pruebas, y alimentar a los motores a través de este circuito integrado.

El CI L293 tiene las siguientes características:

- Se pueden controlar hasta 2 motores.
- Proporciona 1A a los motores (en total) y permite cambiar el sentido de giro.
- Utiliza un puente en H que funciona según se observa en las figuras (internamente utiliza transistores para conmutar*) :

 Modos de operación para invertir el sentido de giro:

"H bridge operating" by Cyril BUTTAY - own work, made using inkscape. Licensed under CC BY-SA 3.0 via Wikimedia Commons.

> Nuestra propuesta es que montes el motor utilizando el CI L293 que realice el siguiente ciclo :
> - Giro en un sentido durante 1seg.
> - Paro durante 0.5seg.
> - Giro en sentido contrario durante 1 seg
> - Paro durante 0.5seg.

Montaje 30 motor con Circuito L293 con Edubásica

Edubásica es una tarjeta diseñada para facilitar la tarea en el aula. Elimina gran parte de cableado y conexiones en la placa de pruebas lo que evita muchos errores en las prácticas.

En esta placa disponemos de las dos opciones vistas anteriormente. Edubásica lleva montados, entre otros componentes, un circuito integrado L293 y un transistor. Por lo tanto podemos activar motores **usando ambas opciones**, aunque lo recomendable es utilizar el L293 que permite el cambio de sentido de giro y regular la velocidad actuando con una señal PWM sobre los dos pines de habilitación (dependiendo de la hoja de datos viene como ENABLE o CHIP INHIBIT) del circuito L293. Los pines de Arduino que pueden regular la velocidad por PWM correspondientes a esas

patillas de habilitación serán: D10 para el motor A y D11 para el motor B.

Para hacer funcionar dos motores DC con Edubásica sólo tenemos que conectar en las clemas indicadas (serigrafiados en la placa como Motor A y Motor B) los dos cables de cada motor. Según se observa en la imagen, tenemos 4 conexiones para los dos motores. Desde Arduino y con la tabla de verdad del CI L293 indicada en la sección anterior, podemos regular el sentido de giro y velocidad de cada motor.

Edubásica **lleva un interruptor que permite tomar el voltaje de la salida Vin de Arduino (alimentación externa), necesaria para dar la corriente suficiente para accionar los motores.** Cuando el piloto Vin está encendido significa que la alimentación de Edubásica viene de Vin de Arduino, o bien, directamente desde una fuente externa conectada a la clema Vin de Edubásica de la regleta de alimentación (en la imagen la regleta de la parte inferior).

Es muy fácil, podemos conectar hasta dos motores en los pines dispuestos para ello, y utilizaremos:

- Para el motor A el control de velocidad por el pin 10 y las direcciones por 8 y 9
- Para el motor B el control de velocidad por el pin 11 y las direcciones por 12 y 13.
- Interruptor en ON

Conectando los dos terminales del motor a la clema del transistor de Edubásica, también podríamos hacerlo funcionar enviando un nivel HIGH al pin digital 6 de Arduino. Este pin (D6) está conectado directamente a la base del transistor de Edubásica. La desventaja respecto al CI L293 es que, en este caso, no podríamos cambiar el sentido de giro.

Montaje 30 motor con Circuito L293 sin Edubásica

Vamos a ver de qué manera se pueden activar los motores DC para hacer una secuencia sencilla de giro. Primero de todo vamos a ver cómo se conecta todo. Las conexiones del circuito integrado según podemos ver en la hoja de datos del fabricante son las siguientes:

- Vcc2(VC), pin 8 del L293: El voltaje que se introduzca aquí alimentará a los motores (Vin de Arduino).
- Vcc1(VSS), pin 18 del L293: El voltaje que se introduzca aquí alimentará al propio circuito integrado (+5V de Arduino).

Las conexiones del circuito con Arduino las podemos ver en el siguiente esquema:

Para saber la orientación del sentido de giro disponemos de la siguiente tabla de verdad:

Simulación 30 motor con Circuito L293 sin Edubásica

https://www.tinkercad.com/embed/k1f3QuzDboV?editbtn=1

Montaje 31 Coche loco

Vamos a darle un poco de emoción

https://giphy.com/embed/p5P3aRq6wimsM

via GIPHY

> Te proponemos que realices unos movimientos locos en el coche (los que quieras)

https://www.youtube.com/embed/j6Z9botrgdo

Montaje 31 chasis coche

Puedes ver en esta página de Luis Llamas cómo hacer un coche teledirigido puede salir por menos de 20€, nosotros como ya tenemos el L298N integrado en **Edubasica**, sólo hemos comprado el chasis por 7€ y no vamos a poner sensor ultrasonidos y sensor de línea:

Si quieres hacerlo todo terreno Luis Llamas te explica cómo hacerlo con cadenas.

Montaje 31 coche loco con Edubásica: Conexiones

Las conexiones son muy sencillas:

- Pines motor A al conector pines motor A de Edubásica
- Pines motor B al conector pines motor B de Edubásica
- Cables de las pilas a Vin y GND

> NOTA: Si funciona pero ves que las ordenes están cambiadas (es decir que queremos que gire a la izquierda y va hacia delante, queremos que vaya hacia delante y gira a la izquierda...) es debido a que tenemos los cables rojo y negro de un motor intercambiados.

Ordenes

Estos son los pines que hay que activar:

Orden	Motor A	Motor B
Velocidad	10	11
Dirección 1	8	12
Dirección 2	9	13

La tabla de verdad es muy fácil:

MOTOR A

Pin 10	Pin 8	Pin 9	Motor A
HIGH	HIGH	LOW	giro
HIGH	LOW	HIGH	giro contrario
LOW	X	X	STOP

MOTOR B

Pin 11	Pin 12	Pin 13	Motor B
HIGH	HIGH	LOW	giro
HIGH	LOW	HIGH	giro contrario
LOW	X	X	STOP

Ampliaciones posibles

Recuerda que en el transistor aún puedes poner otro motor. Respecto a terminales digitales, te quedan pines para poner servos, sensor de ultrasonidos, sensor de líneas...

Montaje 31 coche loco sin Edubásica: Conexiones

Tenemos que comprar el chasis anterior y además un L298N, recomendamos ver esta página de Luis Llamas

embedded-image-5fb9ciab.png

La circuitería se complica, pues necesitamos cablear los pines de control:

> Recuerda: Siempre **pon en común las masas GND.** En caso contrario puedes romper la placa Arduino y también el L298

Ordenes

Para respetar los mismos programas que Luis Llamas, estos son los pines que hay que activar:

Orden	MotorA	MotorB
Velocidad	6	11
Dirección 1	7	9
Dirección 2	8	10

La tabla de verdad es muy fácil:

MOTOR A

Pin 6	Pin 7	Pin 8	MotorA
HIGH	HIGH	LOW	giro
HIGH	LOW	HIGH	giro contrario
LOW	X	X	STOP

MOTOR B

Pin 11	Pin 9	Pin 10	MotorB
HIGH	HIGH	LOW	giro
HIGH	LOW	HIGH	giro contrario
LOW	X	X	STOP

Montaje 31 coche loco con Edubásica: Código

El código es sencillo pero engorroso. Sólo hemos utilizado la función inicial SETUP de tal manera que sólo lo ejecuta una vez. LOOP lo hemos dejado vacío

https://create.arduino.cc/editor/javierquintana/761d52b7-9cef-4af9-9061-8c405888da7d/preview

https://create.arduino.cc/editor/javierquintana/761d52b7-9cef-4af9-9061-8c405888da7d/preview?embed

Montaje 31 coche loco sin Edubásica: Código

Cambia el principio de la cabecera, y omitiendo los LEDs:

```
////////////// SIN EDUBASICA ///////////////
#define ENABLEA 6
#define DIR1A 7
#define DIR2A 8
#define ENABLEB 11
#define DIR1B 9
#define DIR2B 10
```

https://create.arduino.cc/editor/javierquintana/af7ee63f-e189-4ca2-85e7-09fa3c9abfdc/preview

https://create.arduino.cc/editor/javierquintana/af7ee63f-e189-4ca2-85e7-09fa3c9abfdc/preview?embed

> **Reto**: ¿Qué tal si en vez de activarse con el botón, que simplemente cuando este oscuro (utiliza el valor del pin analógico 2 = LDR), se ponga a bailar?

Montaje 32 Coche teledirigido

Sé que lo estabas pensando... ponerle el Bluetooth, vamos allá:

> Realizar un coche teledirigido por Bluetooth con las siguientes órdenes
> - U = Up hacia delante
> - D = Down hacia atrás
> - L = Left giro a la izquierda
> - R = Right giro a la derecha
> - S = Stop paro

https://www.youtube.com/embed/_cn9sSBwZXA

> OJO, para realizar este ejercicio tienes que vincular el HC-06 con la APP, vincularlo, poner el HC-06 pero quitarlo cuando se carga el programa... etc. ¿no te acuerdas? : A repasar lo anterior.

> Las conexiones son las mismas que el montaje 31

Montaje 32 Coche teledirigido código con Edubásica

https://create.arduino.cc/editor/javierquintana/d89d0168-fd8a-42ff-ad5f-f0c9e0da9266/preview

https://create.arduino.cc/editor/javierquintana/d89d0168-fd8a-42ff-ad5f-f0c9e0da9266/preview?embed

Montaje 32 Coche teledirigido código sin Edubásica

Cambia el principio de la cabecera, y omite los LEDs:

https://create.arduino.cc/editor/javierquintana/38a32560-c6ea-499f-8579-87edf60ddd55/preview

https://create.arduino.cc/editor/javierquintana/38a32560-c6ea-499f-8579-87edf60ddd55/preview?embed

https://giphy.com/embed/l3vQYPi2ow7YWXQFW

via GIPHY

https://giphy.com/embed/bXgimR7bxcHAc

via GIPHY

Coche teledirigido con voz

¿Y si ahora lo controlamos por voz?

https://www.youtube.com/embed/fj2cwYS2KvY

Solución

¡¡ Es el mismo código que el anterior !!!

Lo que pasa es en la aplicación Arduino Blue Control utilizamos en vez del mando con flechas, el control de voz. **Easy-peasy** !!

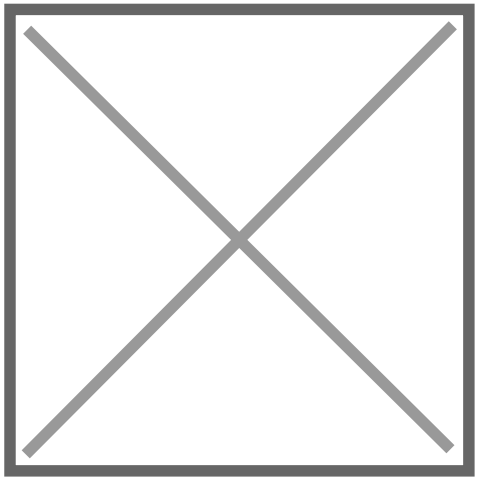
Y previamente hemos configurado los comandos de voz:

Orden de voz	Comando a enviar
adelante	U
atrás	D
derecha	R
izquierda	L
stop	S

Total 5 comandos de voz a configurar:

Barrera

Vamos ahora a realizar UN PROYECTO donde englobamos varios de los elementos que hemos visto en este curso, algo que visualmente tenga un sentido práctico y motivador en el alumnado



Utilizaremos:

- Placa Shield de Edubásica (optativo) por facilitar las conexiones
- Servo motor
- Dos sensores de ultrasonidos
- Módulo Bluetooth
- Imaginación y maña

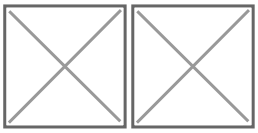
Montaje 33 Barrera por Bluetooth

> Retp 33
>
> El propósito es que cuando se pulse la flecha arriba de la APP DEL MOVIL la barrera suba y se enciende la luz verde, y cuando se pulsa la flecha abajo, baje la barrera y se enciende la luz roja, esta es una manera eficaz de que nadie entre en el recinto si no está autorizado, y que mejor que con una aplicación móvil.

Montaje 33 Barrera por Bluetooth sin Edubásica

Hay que utilizar el esquema del servo

y el esquema del Bluetooth **a la vez**



Montaje 33 Barrera por Bluetooth con Edubásica

Nos simplifica el cableado, conectando el módulo Bluetooth en el zócalo correspondiente tal y como vimos

Con piezas de lego fijamos el servo y le añadimos un cartón que simule una barrera. El pin del servo lo conectaremos **en el 7 de Edubásica**, el Vcc y G a Vin y masa.

La configuración de los ángulos de abierto y cerrado depende en qué posición atornillamos la barrera, luego lo mejor es probarlo con el MONTAJE 25 TESTEA TU SERVO y en nuestro caso nos sale que 40º es abierto y 140º es cerrado.

Detalle de la unión barrera hecha con cartón y el servo, sujetado con piezas de lego y la barrera util

Montaje 33 Barrera por Bluetooth VIDEO

- No hagas caso de los sensores de ultrasonidos por ahora, corresponde al siguiente montaje
- No desmontes las conexiones, te servirán para el siguiente montaje.

https://www.youtube.com/embed/tg0k7ZHvmCg?rel=0

Montaje 33 Barrera por Bluetooth CODIGO

El programa en el Arduino es el siguiente:

https://create.arduino.cc/editor/javierquintana/e8bdf8fc-b5b8-4e27-8207-07ab3ccf6222/preview

https://create.arduino.cc/editor/javierquintana/e8bdf8fc-b5b8-4e27-8207-07ab3ccf6222/preview?embed

Montaje 34 Barrera por sensores ultrasonidos

Ahora **le añadimos** *(esperamos que no hayas desmontado el montaje anterior)* dos sensores de ultrasonidos, si detecta el coche a la entrada de la barrera, se enciende la luz amarilla en espera que el coche pueda abrir con el móvil.

> Reto 34
> -Una vez recibido el código de abrir barrera, se abre y se enciende la luz verde.
> Una vez cruzado el coche, lo detecta el ultrasonido de la salida que cerrará la barrera poniendo el semáforo en rojo otra vez.

https://www.youtube.com/embed/nlnxai_u360?rel=0

La configuración de pines de los ultrasonidos que hemos elegido: (se omite la alimentación +5V y GND por simplificar la ilustración)

arduinobarrerabtus.png

- ULTRASONIDOS DE ENTRADA
  - Trig = 4
  - echo = 2
- ULTRASONIDOS DE SALIDA
  - Trig = 6
  - echo = 5
- Servo
  - Pin = 7

Los sensores de ultrasonidos ocupan algunos pines de los semáforos, pero no hay problemas, si te fijas en el código, en un momento dado se manda un pulso y en otro momento se recoge, se calcula la distancia y se visualiza el semáforo, sin entrar en contradicción.

> El programa por supuesto es mejorable (tiene fallos a ver si los adivinas).

https://create.arduino.cc/editor/javierquintana/5e4fd64c-bbbc-4878-bbcf-8acb61871040/preview

https://create.arduino.cc/editor/javierquintana/5e4fd64c-bbbc-4878-bbcf-8acb61871040/preview?embed

FINNNN

Esperamos que este curso, no sólo te has formado, sino que has disfrutado. Cualquier sugerencia, cambio, propuesta, fallos... puedes hacerlo en www.catedu.es en la sección de SOPORTE o INFORMACÓN ¡¡gracias!!!