**Control de un motor DC mediante el**

**Integrado L293D – Puente H**

1. Descripción

Controlar el movimiento de un motor DC de 9v usando un integrado L293D

1. Objetivo

* Controlar la dirección de movimiento de un motor.
* Controlar la velocidad del motor mediante el integrado L293D y moviendo el potenciómetro

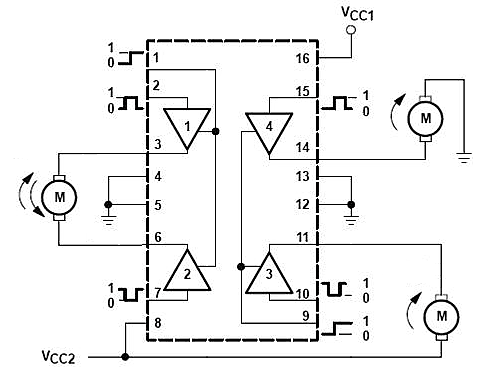
1. Fundamento teórico

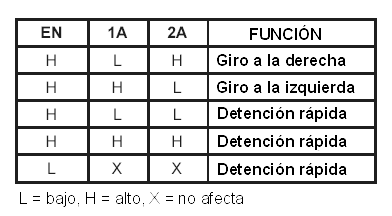
Es un integrado de gran utilidad para controlar motores pequeños y actuadores de corriente directa, este circuito es bastante utilizado en robótica para controlar motores de paso y de corriente directa.

El circuito integrado L293D incluye en su interior 4 drivers o medio puente H, la corriente máxima que puede manejar es 1 amperio a voltajes desde 4 a 36 volts.

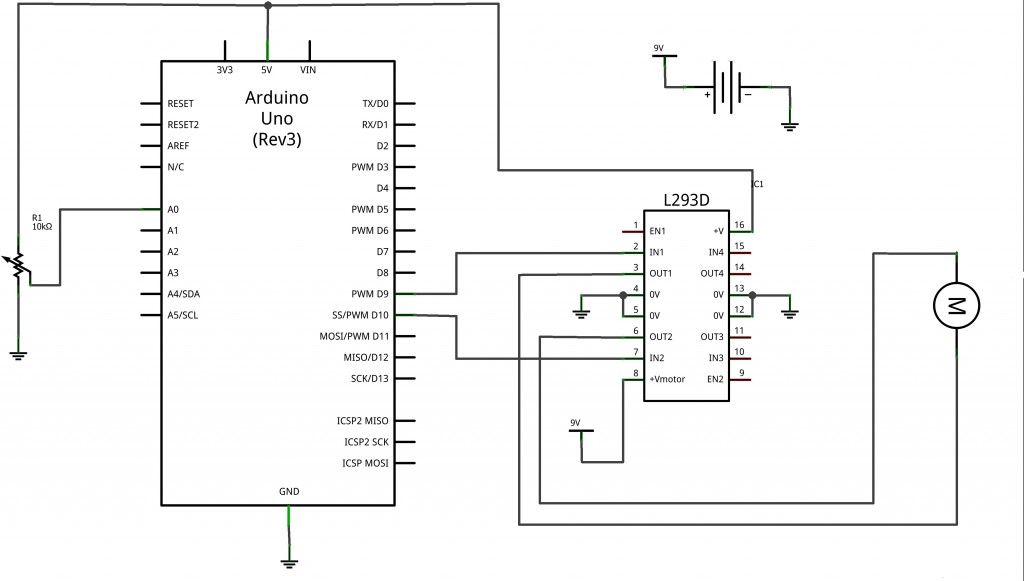
Gracias a que este circuito integrado es capaz de producir corrientes bidimensionales es útil en aplicaciones que requieran controlar la dirección de giro y potencia de motores DC, motores de paso bipolares, solenoides y cualquier otro dispositivo electrónico de corriente continua.

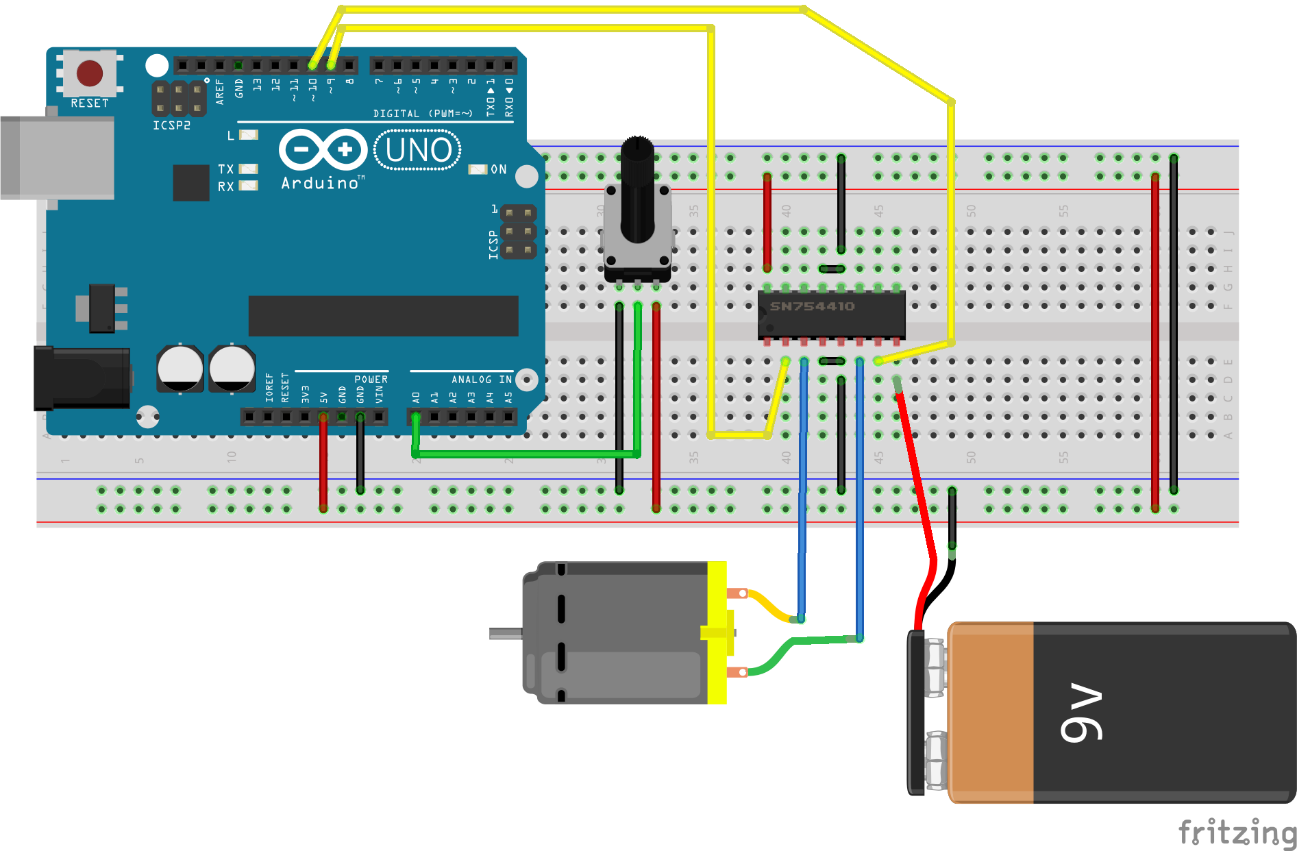






1. Procedimiento



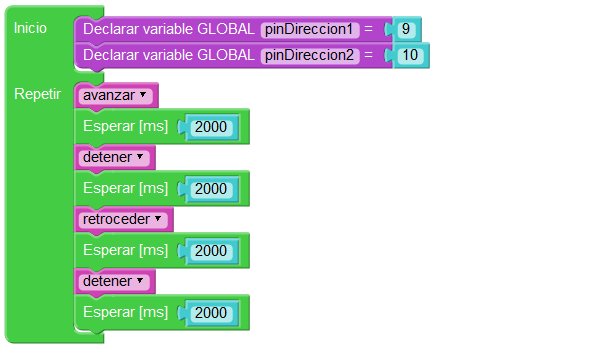


1. Código

int pinDireccion1=9;  
int pinDireccion2=10;

void setup(){  
 pinMode(pinDireccion1,OUTPUT);   
 pinMode(pinDireccion2,OUTPUT);   
}

void loop(){  
 /\*Avanzar\*/  
 analogWrite(pinDireccion1,255);  
 analogWrite(pinDireccion2,0);  
 delay(2000);   
 /\*Detener\*/  
 analogWrite(pinDireccion1,0);  
 analogWrite(pinDireccion2,0);  
 delay(2000);  
 /\*Retroceder\*/  
 analogWrite(pinDireccion1,0);  
 analogWrite(pinDireccion2,255);  
 delay(2000);   
 /\*Detener\*/  
 analogWrite(pinDireccion1,0);  
 analogWrite(pinDireccion2,0);  
 delay(2000);  
 /\*Incognhita\*/  
 analogWrite(pinDireccion1,255);  
 analogWrite(pinDireccion2,255);  
 delay(2000);   
 /\*Detener\*/  
 analogWrite(pinDireccion1,0);  
 analogWrite(pinDireccion2,0);  
 delay(2000);   
}





1. Actividad 1

Utilizar el ese y del joystick para indicar que el motor avance o retroceda.

1. Actividad 2

Controlar la velocidad del motor utilizando un potenciómetro.

int pinDireccion1 = 9;  
int pinDireccion2 = 10;  
int pinPotenciometro = A0;

void setup() {  
 pinMode(pinDireccion1,OUTPUT);  
 pinMode(pinDireccion2,OUTPUT);  
}

void loop() {  
 int valor = analogRead(pinPotenciometro);  
 int pwm1 = map(valor,0,1023,0,255);  
 int pwm2 = map(valor,0,1023,255,0);  
 analogWrite(pinDireccion1,pwm1);  
 analogWrite(pinDireccion2,pwm2);  
}

1. Actividad3

Conectar un Servo que gire a la derecha o a la izquierda cuando conecte eje X y conectar en el botón una sirena.

1. Actividad4

Conectar un pulsador para encender o apagar el motor.

int pinDireccion1 = 9;  
int pinDireccion2 = 10;  
int pinPotenciometro = A0;  
int pinPulsador = 3;  
int valor = 0;  
int valor2 = 0;  
int valor2\_ant=0;  
int estado = 0;

void setup() {  
 // put your setup code here, to run once:  
 Serial.begin(9600);  
 pinMode(pinDireccion1,OUTPUT);  
 pinMode(pinDireccion2,OUTPUT);  
 pinMode(pinPulsador,INPUT);  
}

void loop() {  
 int valor = analogRead(pinPotenciometro);  
 valor2 = digitalRead(pinPulsador);  
 if(valor2==HIGH && valor2\_ant==LOW){  
 estado = 1-estado;  
 }  
 if(estado==1){  
 int pwm1 = map(valor,0,1023,0,255);  
 int pwm2 = map(valor,0,1023,255,0);  
 analogWrite(pinDireccion1,pwm1);  
 analogWrite(pinDireccion2,pwm2);  
}  
 else{  
 analogWrite(pinDireccion1,0);  
 analogWrite(pinDireccion2,0);  
 }  
 valor2\_ant = valor2;  
}