# Índice de contenido

[Índice de contenido I](#_Toc159486050)

[Índice de imágenes I](#_Toc159486051)

[Objetivo II](#_Toc159486054)

[Material y equipo II](#_Toc159486055)

[Introducción: II](#_Toc159486056)

[Marco teórico: III](#_Toc159486057)

[PRÁCTICA 21 II](#_Toc159486052)i

[PRÁCTICA 22 iv](#_Toc159486052)

[PRÁCTICA 23 v](#_Toc159486052)

[PRÁCTICA 24 v](#_Toc159486052)

[Imágenes: vii](#_Toc159486063)

[Conclusiones: X](#_Toc159486064)

# Índice de imágenes

[**Ilustración 1: Simulación P25. IV**](https://d.docs.live.net/2bc8815b9c9f3c83/Escritorio/Documentos/PRACTICA2%20(1).docx#_Toc159455219)

[**Ilustración 2: Simulación P26. V**](https://d.docs.live.net/2bc8815b9c9f3c83/Escritorio/Documentos/PRACTICA2%20(1).docx#_Toc159455219)

[**Ilustración 3: Simulación P27. VII**](https://d.docs.live.net/2bc8815b9c9f3c83/Escritorio/Documentos/PRACTICA2%20(1).docx#_Toc159455219)

[**Ilustración 4: Simulación P28. VII**](https://d.docs.live.net/2bc8815b9c9f3c83/Escritorio/Documentos/PRACTICA2%20(1).docx#_Toc159455219)

[**Imagen1 P25. VII**](https://d.docs.live.net/2bc8815b9c9f3c83/Escritorio/Documentos/PRACTICA2%20(1).docx#_Toc159455219)

[**Imagen2 P26. VII**](https://d.docs.live.net/2bc8815b9c9f3c83/Escritorio/Documentos/PRACTICA2%20(1).docx#_Toc159455219)

[**Imagen3 P27. VII**](https://d.docs.live.net/2bc8815b9c9f3c83/Escritorio/Documentos/PRACTICA2%20(1).docx#_Toc159455219)

[**Imagen4 P28 VII**](https://d.docs.live.net/2bc8815b9c9f3c83/Escritorio/Documentos/PRACTICA2%20(1).docx#_Toc159455219)

# PRÁCTICA 25, 26, 27 y 28

# Motor CD y motor

# Objetivo:

# El objetivo de esta práctica es

# Material y equipo

* Protoboard.
* Arduino UNO.
* Cable de conexión para Arduino.
* Resistencias.
* Motor DC
* H-Bridge
* Sensor de temperatura.
* Cables macho-macho.
* IDE Arduino 2.3.2.

# Introducción:

# E

# Marco teórico:

# C

# Desarrollo:

P25: Motor CD Básico

En esta práctica se nos pidió aprender a controlar la dirección de giro de un motor DC utilizando una placa Arduino. Para lograrlo, se utilizarán dos pines digitales de la placa para cambiar la polaridad del voltaje aplicado al motor, lo que permitirá cambiar su dirección de rotación.

# Procedimiento:

1. Primeramente el motor DC al driver de motor.
2. Conectar las entradas del módulo a los pines pinA (9) y pinB (10) de la placa Arduino.
3. Conectamos el H-Bridge según las especificaciones del fabricante.
4. Escribir el código en el IDE.

# Código:

int pinA = 9;

int pinB = 10;

void setup() {

pinMode (pinA, OUTPUT);

pinMode (pinB, OUTPUT);

}

void loop() {

digitalWrite (pinA, HIGH);

digitalWrite (pinB, LOW);

delay (2000);

digitalWrite (pinA, LOW);

digitalWrite (pinB, HIGH);

delay(2000);

}

# Simulación:

**Diagrama

Descripción generada automáticamente**

Ilustración 1. Simulación P25

P26: Motor CD Sentido y velocidad variable

En esta práctica se nos pidió diseñar y programar un sistema para controlar el sentido de giro y la velocidad de un motor DC utilizando una placa Arduino. El sistema debe ser capaz de recibir comandos a través de la comunicación serial para cambiar la dirección del motor.

# Procedimiento 2:

1. Primeramente, conectar el motor DC directamente a los pines digitales de la placa Arduino.
2. Conectar las entradas del driver de motor a los pines pinA (9) y pinB (10) de la placa Arduino.
3. Conectar la fuente de alimentación adecuada al motor y al driver de motor según las especificaciones del fabricante.
4. Escribir el código en el IDE para inicializar la comunicación serial y configurar los pines de control del motor.
5. Enviar los comandos + y - para controlar la dirección del motor y observar el comportamiento del motor.

# Código 2:

int pinA = 9;

int pinB = 10;

int caracter;

void setup() {

Serial.begin(9600);

pinMode (pinA, OUTPUT);

pinMode (pinB, OUTPUT);

}

void loop() {

  if (Serial.available()){

    caracter = Serial.read();

    if(caracter == '+'){

      digitalWrite(pinA, HIGH);

      digitalWrite(pinB, LOW);

      delay (2000);

    }

    else if (caracter == "-"){

     digitalWrite(pinA, LOW);

      digitalWrite(pinB, HIGH);

      delay (2000);

    }

  }

}

# Simulación 2:

Diagrama

Descripción generada automáticamente

Ilustración 2. Simulación P26

P27: Reto botones

En esta práctica se nos pidió

# Procedimiento 3:

1. Primeramente
2. .

# Código 3:

// Definimos los pines de los botones

const int buttonPin1 = 2; // Sentido horario

const int buttonPin2 = 3; // Sentido antihorario

const int buttonPin3 = 4; // Parar motor

// Pines para el L293D

const int motorPin1 = 5; // Entrada 1 del L293D

const int motorPin2 = 6; // Entrada 2 del L293D

const int enablePin = 9; // Pin de habilitación

void setup() {

  // Configurar los pines de los botones como entradas

  pinMode(buttonPin1, INPUT);

  pinMode(buttonPin2, INPUT);

  pinMode(buttonPin3, INPUT);

  // Configurar los pines del L293D como salidas

  pinMode(motorPin1, OUTPUT);

  pinMode(motorPin2, OUTPUT);

  pinMode(enablePin, OUTPUT);

  // Inicializar el motor detenido

  digitalWrite(motorPin1, LOW);

  digitalWrite(motorPin2, LOW);

  analogWrite(enablePin, 0);

}

void loop() {

  // Leer el estado de los botones

  int buttonState1 = digitalRead(buttonPin1);

  int buttonState2 = digitalRead(buttonPin2);

  int buttonState3 = digitalRead(buttonPin3);

  // Si el botón 1 está presionado, el motor gira en sentido horario

  if (buttonState1 == HIGH) {

    digitalWrite(motorPin1, HIGH);

    digitalWrite(motorPin2, LOW);

    analogWrite(enablePin, 255); // Velocidad máxima

  }

  // Si el botón 2 está presionado, el motor gira en sentido antihorario

  else if (buttonState2 == HIGH) {

    digitalWrite(motorPin1, LOW);

    digitalWrite(motorPin2, HIGH);

    analogWrite(enablePin, 255); // Velocidad máxima

  }

  // Si el botón 3 está presionado, el motor se detiene

  else if (buttonState3 == HIGH) {

    digitalWrite(motorPin1, LOW);

    digitalWrite(motorPin2, LOW);

    analogWrite(enablePin, 0); // Motor detenido

  }

}

# Simulación 3:

**Diagrama

Descripción generada automáticamente**

Ilustración 3. Simulación P27

P28: Motor a pasos básico

En esta práctica se nos pidió

# Procedimiento 4:

1. Primeramente
2. .

# Código 4:

int retardo=5;

int dato\_rx;

int numero\_pasos =0;

String cadenaLeida;

void setup() {

  Serial.begin(9600);

  for (int g=8;g<12;g++){

    pinMode(g,OUTPUT);

  }

}

void loop() {

  while (Serial.available()) {

    delay(retardo);

    char caracter=Serial.read();

    cadenaLeida +=caracter;

  }

  if(cadenaLeida.length()>0){

    dato\_rx = cadenaLeida.toInt();

    Serial.print (dato\_rx);

    Serial.print (" Grados");

    delay(retardo);

    dato\_rx = (dato\_rx\*1.422222222);

  while(dato\_rx > numero\_pasos){

    paso\_izq();

    numero\_pasos =numero\_pasos + 1;

  }

   while(dato\_rx <= numero\_pasos){

    paso\_der();

    numero\_pasos =numero\_pasos - 1;

  }

  cadenaLeida="";

  apagado();

}

}

void paso\_der(){

  digitalWrite(11,  LOW);

  digitalWrite(10,  LOW);

  digitalWrite(9,  HIGH);

  digitalWrite(8,  HIGH);

     delay(retardo);

  digitalWrite(11,  LOW);

  digitalWrite(10, HIGH);

  digitalWrite(9,  HIGH);

  digitalWrite(8,   LOW);

     delay(retardo);

  digitalWrite(11, HIGH);

  digitalWrite(10, HIGH);

  digitalWrite(9,   LOW);

  digitalWrite(8,   LOW);

     delay(retardo);

  digitalWrite(11, HIGH);

  digitalWrite(10,  LOW);

  digitalWrite(9,   LOW);

  digitalWrite(8,  HIGH);

     delay(retardo);

}

void paso\_izq(){

  digitalWrite(11, HIGH);

  digitalWrite(10, HIGH);

  digitalWrite(9,   LOW);

  digitalWrite(8,   LOW);

     delay(retardo);

  digitalWrite(11,  LOW);

  digitalWrite(10, HIGH);

  digitalWrite(9,  HIGH);

  digitalWrite(8,   LOW);

     delay(retardo);

  digitalWrite(11,  LOW);

  digitalWrite(10,  LOW);

  digitalWrite(9,  HIGH);

  digitalWrite(8,  HIGH);

     delay(retardo);

  digitalWrite(11, HIGH);

  digitalWrite(10,  LOW);

  digitalWrite(9,   LOW);

  digitalWrite(8,  HIGH);

     delay(retardo);

}

void apagado(){

  digitalWrite(11,  LOW);

  digitalWrite(10,  LOW);

  digitalWrite(9,   LOW);

  digitalWrite(8,   LOW);

     delay(retardo);

}

# Simulación 4:

Diagrama, Esquemático

Descripción generada automáticamente

Ilustración 4. Simulación P28

# Imágenes:

Imagen3 – P26

Imagen2 – P25

Imagen5 - P28

Imagen4 – P27

**Conclusiones:**

**Conclusión Jimena García Torres:** Las prácticas nos permititeron conocer el funcionamiento de los motores que se nos presentaron, así como la manipulación de estos y sus polaridades, con sus respectivas conexiones a la protoboard.

**Conclusión Diana Carolina Plascencia Rodríguez:**

**Conclusión Juan Ricardo Barrón González:**

**Bibliografías:**