

INSTITUTO TECNOLÓGICO DE SALTILLO

REPORTE DE PRACTICA 2: MOTOR PASO A PASO

Presenta:

Kevin Alejandro Fuentes Martínez

Orlando Ávila Céspedes

Eduardo Ernesto Castillo Cordero

Rogelio Pérez Guevara

Materia:

Sistemas Programables

Profesor:

ING. Mona Peña Luis Javier

11/10/2024

Índice

Objetivo general.....	3
Marco teórico.....	3
Desarrollo de la práctica	3
Materiales y Equipos	3
Programación del Arduino	3
Montaje del Circuito.....	7
Ejecución	7
Resultados	7
Imágenes del circuito	8
Conclusiones.....	8

Objetivo general

- Implementar un control de motor paso a paso utilizando un Arduino.
- Crear un programa que permita controlar el motor en diferentes modos: pasos completos, torque aumentado, y una combinación de ambos, con control sobre la dirección de movimiento y el estado del motor.

Marco teórico

Los motores paso a paso son dispositivos electromecánicos que convierten impulsos eléctricos en movimientos discretos y controlables. Cada paso corresponde a un pequeño movimiento del motor, lo que permite un control preciso de la posición. En esta práctica, se implementa un motor paso a paso que puede operarse en diferentes modos, como el de pasos completos o el de torque aumentado, con la posibilidad de cambiar la dirección (horaria o antihoraria) y detener o continuar el movimiento a través de comandos enviados por consola.

El motor paso a paso utilizado aquí es controlado mediante un Arduino UNO y se conectan cuatro pines de salida a los terminales del motor. El código permite la configuración del motor en varios modos de operación que se definen mediante secuencias de pasos específicos. El uso de una comunicación serial permite que el usuario controle el comportamiento del motor en tiempo real.

Desarrollo de la práctica

Materiales y Equipos

- 1 Arduino UNO
- 1 Motor paso a paso
- 1 Driver de motor o puente H
- Resistencias de 330 Ω
- Protoboard y cables de conexión
- Fuente de alimentación USB para el Arduino
- Procedimiento

Programación del Arduino

Se desarrolló un código en el que se definen las secuencias de pasos para controlar el motor en modo de pasos completos y en modo de torque aumentado. El programa también permite cambiar la dirección del motor (horario o antihorario) y alternar entre diferentes modos a través de comandos enviados desde la consola serie.

Código:

```
// Definición de pines para las conexiones del motor
const int IN1 = 8; // Pin de control IN1 conectado al pin 8 de Arduino
const int IN2 = 9; // Pin de control IN2 conectado al pin 9 de Arduino
```

```

const int IN3 = 10; // Pin de control IN3 conectado al pin 10 de Arduino
const int IN4 = 11; // Pin de control IN4 conectado al pin 11 de Arduino

// Variables para la consola
String command = ""; // Variable para almacenar el comando recibido de la consola
int stepDelay = 10; // Tiempo de espera entre pasos (en milisegundos), ajustado a 10ms

// Secuencia de pasos completa para el motor (8 pasos)
int stepsFull[8][4] = {
    {1, 0, 0, 0},
    {1, 1, 0, 0},
    {0, 1, 0, 0},
    {0, 1, 1, 0},
    {0, 0, 1, 0},
    {0, 0, 1, 1},
    {0, 0, 0, 1},
    {1, 0, 0, 1}
};

// Secuencia de pasos para modo de torque aumentado (4 pasos)
int stepsTorque[4][4] = {
    {1, 0, 0, 1},
    {0, 1, 0, 1},
    {0, 1, 1, 0},
    {1, 0, 1, 0}
};

// Variables para controlar el modo, dirección y estado del motor
bool fullStepMode = true; // Modo de pasos completos (true) o torque aumentado (false)
bool clockwise = true;    // Dirección de movimiento (true = horario, false = antihorario)
bool isMoving = true;     // Estado del motor (true = en movimiento, false = detenido)

void setup() {
    // Configuración de los pines como salidas
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    pinMode(IN3, OUTPUT);
    pinMode(IN4, OUTPUT);

    // Iniciar la comunicación serial a 9600 baudios

```

```

Serial.begin(9600);
// Mensaje inicial en la consola para indicar los comandos disponibles
Serial.println("Ingrese: ");
Serial.println("'Horario PasoCompleto'");
Serial.println("'Antihorario PasoCompleto'");
Serial.println("'Horario Torque'");
Serial.println("'Antihorario Torque'");
Serial.println("'Combinado Horario'");
Serial.println("'Combinado Antihorario'");
Serial.println("'Parar'");
Serial.println("'Continuar'");
}

void loop() {
    // Comprobar comandos de la consola
    if (Serial.available()) {
        command = Serial.readString(); // Leer el comando enviado
        command.trim(); // Eliminar espacios en blanco al principio y al final
        handleCommand(command); // Llamar a la función para manejar el comando
    }

    // Ejecutar el movimiento solo si el motor está en movimiento
    if (isMoving) {
        // Mover el motor según el modo de operación
        moveMotor(fullStepMode ? stepsFull : stepsTorque, fullStepMode ? 8 : 4);
    }
}

// Función para manejar los comandos recibidos
void handleCommand(String cmd) {
    if (cmd == "Horario PasoCompleto") {
        clockwise = true;           // Dirección horaria
        fullStepMode = true;        // Paso completo
        Serial.println("Modo: Paso Completo, Dirección: Horario");
    }
    else if (cmd == "Antihorario PasoCompleto") {
        clockwise = false;          // Dirección antihoraria
        fullStepMode = true;        // Paso completo
        Serial.println("Modo: Paso Completo, Dirección: Antihorario");
    }
    else if (cmd == "Horario Torque") {
        clockwise = true;           // Dirección horaria
        fullStepMode = false;       // Torque aumentado
        Serial.println("Modo: Torque aumentado, Dirección: Horario");
    }
}

```

```

else if (cmd == "Antihorario Torque") {
    clockwise = false;          // Dirección antihoraria
    fullStepMode = false;       // Torque aumentado
    Serial.println("Modo: Torque aumentado, Dirección: Antihorario");
}
else if (cmd == "Combinado Horario") {
    clockwise = true;           // Dirección horaria
    fullStepMode = false;       // Modo combinado
    Serial.println("Modo: Combinado (Paso Completo + Torque), Dirección:
Horario");
}
else if (cmd == "Combinado Antihorario") {
    clockwise = false;          // Dirección antihoraria
    fullStepMode = false;       // Modo combinado
    Serial.println("Modo: Combinado (Paso Completo + Torque), Dirección:
Antihorario");
}
else if (cmd == "Parar") {
    isMoving = false;           // Detener el movimiento
    Serial.println("Motor detenido");
}
else if (cmd == "Continuar") {
    isMoving = true;            // Continuar el movimiento
    Serial.println("Motor en movimiento");
}
else {
    Serial.println("Comando no reconocido");
}
}

// Función para mover el motor según la secuencia de pasos
void moveMotor(int steps[][4], int stepCount) {
    // Iterar sobre la cantidad de pasos especificada
    for (int i = 0; i < stepCount; i++) {
        // Calcular el índice según la dirección del movimiento
        int idx = clockwise ? i : (stepCount - 1 - i);

        // Establecer los pines de salida según la secuencia de pasos
        digitalWrite(IN1, steps[idx][0]);
        digitalWrite(IN2, steps[idx][1]);
        digitalWrite(IN3, steps[idx][2]);
        digitalWrite(IN4, steps[idx][3]);

        // Esperar un tiempo determinado antes de continuar al siguiente paso
        delay(stepDelay);
    }
}

```

```

    // Comprobar si hay un nuevo comando antes de completar el ciclo de
    pasos
    if (Serial.available()) {
        command = Serial.readString(); // Leer el nuevo comando
        command.trim(); // Eliminar espacios en blanco
        handleCommand(command); // Manejar el nuevo comando
        break; // Salir del bucle si hay un nuevo comando
    }
}
}
}

```

Montaje del Circuito

1. Conectar los pines IN1, IN2, IN3 e IN4 del motor a los pines 8, 9, 10 y 11 del Arduino, respectivamente.
2. Conectar el motor a través de un controlador (como un puente H) que permita el control adecuado de las bobinas del motor paso a paso.
3. Alimentar el Arduino mediante el puerto USB.

Ejecución

Al cargar el código en el Arduino y abrir la consola serial, se permite al usuario enviar comandos para controlar el motor. Los comandos disponibles son:

'Horario PasoCompleto': Mueve el motor en dirección horaria en modo de pasos completos.

'Antihorario PasoCompleto': Mueve el motor en dirección antihoraria en modo de pasos completos.

'Horario Torque': Mueve el motor en dirección horaria en modo de torque aumentado.

'Antihorario Torque': Mueve el motor en dirección antihoraria en modo de torque aumentado.

'Combinado Horario': Combina ambos modos en dirección horaria.

'Combinado Antihorario': Combina ambos modos en dirección antihoraria.

'Parar': Detiene el motor.

'Continuar': Reanuda el movimiento del motor.

Resultados

El código implementado permitió controlar el motor paso a paso de manera efectiva. Los comandos enviados desde la consola serial permitieron alternar entre los

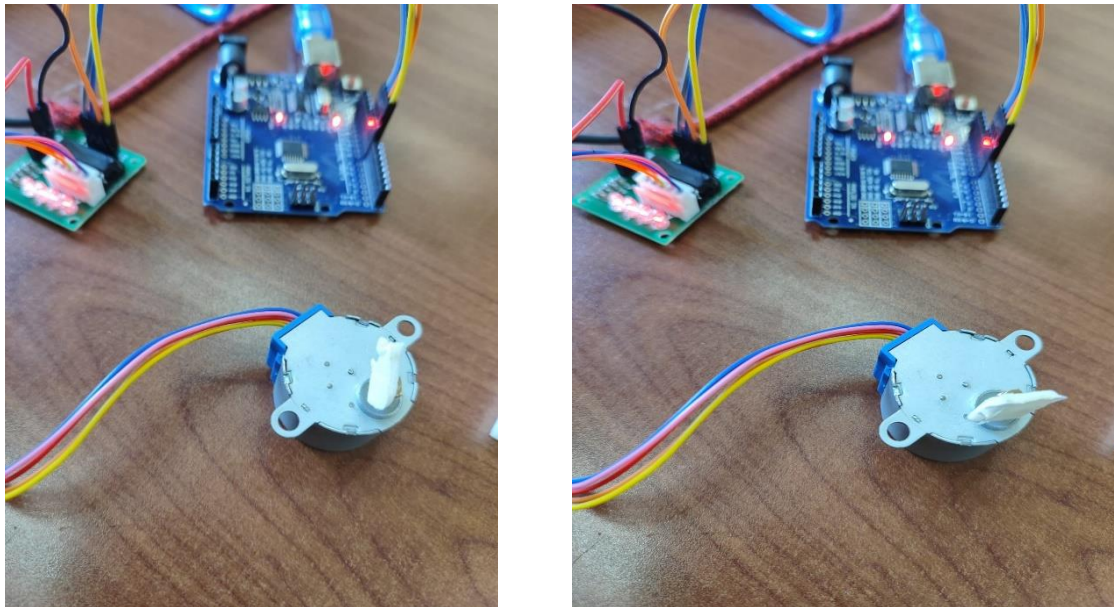
diferentes modos de operación y la dirección de movimiento. A continuación, se describen algunos de los comportamientos observados:

Modo Paso Completo Horario: El motor giró en sentido horario, activando los pines de salida en secuencia completa.

Modo Torque Aumentado Antihorario: El motor giró en sentido antihorario, utilizando una secuencia reducida para aumentar el torque.

Modo Combinado: Al alternar entre ambos modos, el motor exhibió un comportamiento mixto, mejorando el control de velocidad y fuerza.

Imágenes del circuito



Conclusiones

La implementación de un sistema de control de motor paso a paso en Arduino fue exitosa. El sistema responde de manera eficiente a los comandos enviados desde la consola, permitiendo una manipulación versátil del motor en diferentes modos. Esta práctica reforzó los conceptos de control de motores y programación de microcontroladores. El uso de comandos de consola serie también facilitó la interacción en tiempo real, permitiendo cambios dinámicos en la operación del motor sin necesidad de modificar el código directamente.