

## Resumen del Proyecto

Este documento presenta un resumen detallado del trabajo realizado durante el curso, enfocado en el diseño, implementación y control de diversos motores mediante una Raspberry Pi. El objetivo principal fue desarrollar sistemas eficientes para controlar motores con diferentes configuraciones, explorando conceptos como aceleración gradual, multithreading y el uso de pantallas LCD para interacción en tiempo real.

## Descripción General del Proyecto

Durante el curso, trabajamos con tres tipos principales de motores: un motor de corriente continua, un motor paso a paso unipolar (28BYJ-48) y un motor paso a paso bipolar (NEMA 17). Cada motor requería un enfoque específico de control, lo que permitió desarrollar habilidades en la gestión de diferentes configuraciones de hardware y software. A continuación, se describen las actividades principales realizadas durante el curso.

## Actividades Realizadas

### 1. \*\*Control de un motor de corriente continua:\*\*

Se implementó un sistema para controlar la velocidad y dirección de un motor de corriente continua mediante señales PWM provenientes de la Raspberry Pi. Este ejercicio inicial permitió familiarizarnos con el control básico de motores y la interacción con hardware externo.

### 2. \*\*Control de un motor paso a paso unipolar (28BYJ-48):\*\*

Utilizando el driver ULN2003, se diseñó un sistema para controlar el motor unipolar. Se implementaron funciones para ajustar la velocidad y el sentido de giro de forma dinámica, utilizando inputs del usuario. Este proyecto permitió comprender la lógica detrás del control por pasos y la importancia de los retrasos entre pulsos.

### 3. \*\*Control de un motor paso a paso bipolar (NEMA 17):\*\*

Este fue uno de los proyectos más avanzados. Se diseñaron múltiples versiones de scripts (Nema3, Nema4 y Nema6) para implementar funcionalidades como el aumento gradual de la velocidad mediante una función sigmoide y el ajuste dinámico de microstepping. El driver utilizado fue el DRV8825, configurado para soportar diferentes niveles de microstepping según las necesidades del proyecto.

### 4. \*\*Proyecto del Posicionador:\*\*

Este proyecto consistió en utilizar un motor paso a paso lineal para dividir una distancia total de 5 mm en 20 posiciones discretas. El sistema estaba diseñado para asumir cada posición en 0,5 segundos, ajustando la velocidad del motor según la distancia a recorrer. El input principal era una lista de 100 posiciones discretas, procesadas en tiempo real.

### 5. \*\*Implementación de multithreading:\*\*

Para optimizar los sistemas y permitir la interacción dinámica con los usuarios, se

implementó multithreading. Esto permitió que el sistema respondiera a inputs mientras los motores estaban en operación, mejorando significativamente la eficiencia.

#### 6. **\*\*Integración de una pantalla LCD:\*\***

Se añadió una pantalla LCD al sistema para mostrar información en tiempo real, como velocidad, dirección y estado actual de los motores. Esto mejoró la usabilidad y brindó al usuario una interfaz visual para interactuar con el sistema.

### **Conclusiones**

El curso permitió desarrollar una comprensión profunda sobre el control de motores utilizando Raspberry Pi. Las habilidades adquiridas incluyeron la configuración y uso de hardware específico, el diseño de algoritmos de control eficientes, y la implementación de técnicas avanzadas como el aumento gradual de velocidades y multithreading. Los proyectos realizados sentaron una base sólida para aplicaciones futuras en automatización y robótica.