

Mejoras propuestas para el sistema de motores paso a paso del proyecto UPFEET

A continuación se detallan las mejoras solicitadas para optimizar el rendimiento del sistema mecánico del escáner UPFEET. Estas mejoras apuntan a aumentar la velocidad de desplazamiento, reducir la latencia del programa y lograr un funcionamiento más eficiente de los motores NEMA 17.

1. Implementación de control por PWM para los motores paso a paso

Objetivo: mejorar eficiencia y control de velocidad del motor.

Descripción:

Los motores paso a paso habitualmente se controlan mediante señales digitales de paso (STEP) y dirección (DIR). Sin embargo, se puede integrar PWM para modular la corriente entregada al motor a través del driver.

Esto permite:

- Un movimiento más fluido.
- Mejor control de aceleración y desaceleración.
- Mayor eficiencia energética.
- La Raspberry Pi Pico posee temporizadores y canales PWM suficientes para generar señales precisas sin saturar el CPU.

Impacto: mayor velocidad útil sin perder torque en bajos pasos.

2. Eliminar comandos de escritura (prints) para evitar ralentización

Objetivo: aumentar la velocidad de ejecución del programa.

Descripción

Las instrucciones `print()` generan un cuello de botella importante, especialmente en microcontroladores. Cuando el programa imprime constantemente datos por USB/UART, la ejecución se detiene hasta que el buffer se libera.

Esto provoca:

- Movimientos más lentos.
- Saltos entre pasos.
- Bloqueos cuando hay muchos prints consecutivos.

Solución:

- Eliminar todos los prints innecesarios en el código final.
- Mantener únicamente prints condicionados, por ejemplo, solo para depurar en modo "debug".
- Implementar una variable booleana `debug = True/False` para desactivar toda la salida por consola.

Impacto: el programa se vuelve más dinámico, fluido y sin interrupciones.

3. Configurar los microsteps del driver para aumentar la velocidad del motor

Objetivo: lograr mayor velocidad angular del motor NEMA 17 sin perder precisión.

Descripción:

Los drivers (A4988, DRV8825 u otros) permiten configurar el número de microsteps mediante pines MS1, MS2 y MS3.

A menor microstepping → mayor velocidad del motor, ya que requie