

Documentación Gustavo Petro Tangarife

Nodos ESP32

23 de agosto de 2025

Índice

1. Nodo Motor	2
1.1. Movimientos Básicos	2
1.2. Velocidad y Dirección vía MQTT	2
2. Archivos	2
2.1. Archivo <code>__functions.py</code>	2
2.2. Archivo <code>test_IP_fija.py</code>	3
2.3. Archivo <code>test_motores_vel_direc.py</code>	3
2.3.1. Funciones utilizadas	3
2.4. Archivo <code>config.py</code>	4

1. Nodo Motor

El sistema está diseñado para controlar cuatro motores de corriente continua (DC) mediante una ESP32, utilizando señales digitales y modulación por ancho de pulso (PWM).

- Cada motor se maneja con dos pines digitales: in1 y in2, que determinan la dirección del giro (adelante o atrás).
- Control de Velocidad (PWM): La velocidad de los motores no se controla simplemente encendiéndolos o apagándolos

1.1. Movimientos Básicos

Los motores están dispuestos en cuatro posiciones: A, B, C y D. Al combinar los giros de los cuatro motores, se obtienen diferentes movimientos:

- Adelante: todos los motores giran hacia adelante.
- Atrás: todos los motores giran hacia atrás.
- Izquierda: los motores del lado derecho avanzan y los del lado izquierdo retroceden, provocando un giro.
- Derecha: los motores del lado izquierdo avanzan y los del lado derecho retroceden.
- Detener: se desactivan todos los pines, parando los motores.

1.2. Velocidad y Dirección vía MQTT

El sistema recibe comandos desde un broker MQTT (por ejemplo, instalado en una Raspberry Pi con ROS).

Se usan dos tipos de mensajes:

- Dirección (adelante, atras, izquierda, derecha, detener).
- Velocidad (un número de 0 a 1023).

Cuando llega un mensaje de dirección, se activa la función que ajusta los pines de los motores. Cuando llega un mensaje de velocidad, se actualiza el valor PWM en todos los motores.

2. Archivos

2.1. Archivo `__functions.py`

- **WiFi:** funciones para conectar la ESP32 a una red con IP fija.
- **Motores:** funciones para configurar PWM, mover hacia diferentes direcciones y ajustar la velocidad.
- **Otros actuadores:** funciones para controlar un LED y un servo.

- **Ejecución de comandos:** función `ejecutar_comando` que interpreta cadenas de texto y delega la acción a motores, LED o servo.
- **MQTT:** funciones para establecer la conexión al broker, suscripción a *topics* y procesamiento de mensajes recibidos.

2.2. Archivo `test_IP_fija.py`

Este archivo prueba la conexión WiFi y el servidor web.

- Usa `conectar_wifi_fija` para conectar a la red y asignar la IP 192.168.1.50 de forma estática.

2.3. Archivo `test_motores_vel_direc.py`

El programa ejecuta un bucle principal donde escucha mensajes MQTT. En caso de interrupción manual (**Ctrl+C**), los motores se detienen y se desconecta del broker MQTT. Este archivo prueba la integración de motores y MQTT.

- Conecta a la red con `conectar_wifi_fija`.
- Configura los motores con la velocidad inicial mediante `configurar_motores`.
- Se conecta al broker MQTT con `conectar_mqtt` y se suscribe a los *topics* configurados.
- El bucle principal escucha mensajes MQTT y:
 - Ajusta la velocidad si recibe un valor en el *topic* de velocidad.
 - Cambia la dirección de movimiento (adelante, atrás, izquierda, derecha, detener) si recibe un comando.
- Si el programa se interrumpe con **Ctrl+C**, ejecuta `detener_motores` y desconecta del broker.

2.3.1. Funciones utilizadas

- `conectar_wifi_fija()`: conecta la ESP32 a la red WiFi con la configuración establecida en `config.py`.
- `configurar_motores(vel)`: inicializa el PWM de los motores con una velocidad definida.
- `conectar_mqtt()`: conecta la ESP32 al broker MQTT y se suscribe a los *topics* configurados.
- `detener_motores()`: detiene todos los motores, usada en la finalización del programa.

2.4. Archivo `config.py`

Este archivo contiene la configuración centralizada del proyecto. Define:

- Credenciales de WiFi.
- Dirección IP fija de la ESP32 y parámetros de red.
- Dirección del broker MQTT y nombres de *topics*.
- Pines asociados a cada motor, LED y servo.
- Parámetros globales como la velocidad inicial y el flag de depuración `DEBUG`.