

# DISEÑO E IMPLEMENTACION DE UN ROBOT LIMPIA-VIDRIOS (HOMERO V1)

Juan David Alfonso

Universidad de los Andes

jd.alfonso2@uniandes.edu.co

202214487

**Resumen**—El proyecto **HOMERO VI** consiste en el diseño e implementación de un robot limpia-vidrios, orientado a la limpieza de superficies verticales y horizontales. Este sistema integra diseño mecánico, electrónica y control inteligente. El corazón del sistema es un microcontrolador ESP32, que gestiona múltiples subsistemas, incluyendo un EDF de 64 mm para succión, motores controlados mediante puentes H con transistores BJT y MOSFET, y diversos sensores para monitoreo y control.

Se emplearon optoacopladores para garantizar aislamiento eléctrico, protegiendo el microcontrolador de picos de voltaje y ruido, mientras que un sistema de luces LED RGB y un módulo OLED proporcionan retroalimentación visual. Además, el robot cuenta con un sistema de monitoreo de batería y refrigeración automática, mejorando la eficiencia operativa.

Este informe detalla el desarrollo del sistema, incluyendo esquemas eléctricos y diseño de circuitos elaborados en KiCad, así como modelos 3D generados en Fusion 360.

**Index Terms**—BJT, NMOS, NPN, PNP optoacoplacion, ESP32, PWM, I-BUS, FLYSKY

## I. INTRODUCCIÓN

En el desarrollo de sistemas robóticos orientados a tareas específicas, como la limpieza de superficies verticales, se destacan tecnologías que combinan diseño mecánico, electrónica y control inteligente. Este proyecto, titulado "**HOMERO VI**", corresponde al diseño e implementación de un robot limpia-vidrios que integra soluciones confiables basadas en fundamentos de electrónica y mecánica aplicados.

En este contexto, el sistema electrónico se fundamenta en la utilización de transistores BJT (TIP41C/TIP42C) y MOSFET (IRF44Z y 2N7000), empleados para el control de motores mediante puentes H y un controlador PWM. Estos componentes permiten gestionar de manera precisa y segura las corrientes necesarias para el funcionamiento de los actuadores, asegurando un control eficiente de velocidad y dirección. Además, se implementaron optoacopladores (4N35 y PC817) para garantizar el aislamiento eléctrico entre el microcontrolador ESP32 y las cargas inductivas, protegiendo

el sistema contra picos de voltaje y ruido eléctrico. Además, de evitar la alteración de lecturas y señales recibidas o emitidas por el microcontrolador.

El robot utiliza un EDF de 64 mm que genera succión suficiente para adherirse a paredes y techos mientras opera con un rodillo motorizado que combina microfibra y espuma, para optimizar la limpieza se usa un sistema de velcro con microfibra y goma al final. El informe incluye una detallada documentación de los esquemas eléctricos, elaborados en KiCad, que abarcan el diseño de circuitos clave como el controlador del rodillo, los puentes H, el sistema de refrigeración y el divisor de voltaje para el monitoreo de batería. Además, se describe el uso de diodos de protección en paralelo con los motores para mitigar las corrientes inversas, asegurando la durabilidad de los componentes.

El robot también incorpora un sistema de luces LED RGB para señalización visual de diferentes estados operativos, un sistema de monitoreo de batería con alertas visuales y auditivas, y un módulo de refrigeración automática activado por un sensor de temperatura MLX90614. Estos sistemas complementan la funcionalidad del robot, optimizando su seguridad y eficiencia operativa. contando el mismo con dos modos, tanto alimentación con baterías, como con una fuente DC, la usada en la prueba fue de 800W.

Este proyecto demuestra el dominio de conceptos fundamentales de electrónica, como el uso de dispositivos semiconductores, sino que también introduce una solución funcional e innovadora que combina principios de diseño mecánico, control eléctrico y programación. El código fuente se encuentra disponible en mi GitHub en el repositorio publico **Homero VI** próximamente incluiré los diseños 3D Y esquemas eléctricos, también archivos gerber para fabricación del PCB que me encuentro diseñando, por el momento la implementación fue con Baquetas soldadas. [2]

### I-A. Materiales Y Diagrama de bloques del sistema

#### MATERIALES:

- ◊ EDF 64mm 390KV, ESC 50A 3s comprado por Aliexpress [3]
- ◊ Batería 3S de LiPo para alimentación (12.6V nominales, 11.1V nominal promedio).
- ◊ Receptor 10CH FLYSKY 2.4 GHz o similar. (I-BUS necesario)
- ◊ Control FLYSKY i6 2.4GHz o similar. (I-BUS necesario)
- ◊ MDF 3mm para corte laser
- ◊ sensor de temperatura MLX90614 I2C
- ◊ Transistores BJT NPN/PNP **TIP41C/TIP42C** para el puente H.
- ◊ Rodillo de Limpieza: Controlado por controlador implementado con optoacoplador **PC817** y **IRF44Z** para control PWM en un solo sentido.
- ◊ Motoreductor amarillo.
- ◊ Velcro y Paño de Microfibra: Facilita la limpieza y permite retirar el paño para lavado.
- ◊ Sistema refrigeración 16 y soporte para el rodillo 17 impresos en 3D.
- ◊ Resistores ( $1800\Omega$  y  $600\Omega$ ): Divisor de voltaje para la lectura del nivel de batería en el ESP32.
- ◊ Resistores  $1K\Omega$ ,  $10K\Omega$ ,  $220\Omega$  para puente H, controlador PWM y otros circuitos implementados con **2n7000**.
- ◊ Diodos **4n4007** para el control de corriente inversa.
- ◊ Llantas de moto reduktor amarillo. (se realizo adaptacion al eje para que funcionara con N20)
- ◊ Micromotorreductores N20 2kg/in.
- ◊ Cautin, estaño, pasta para soldar.
- ◊ Cable diferentes AWG (8,12,18,24).
- ◊ Conectores JST (2,4,6) pines.
- ◊ Pantalla SSD1306 I2C
- ◊ ESP32 DEVKIT V1

◊ Goma limpiavidrios.

◊ Ventilador 30 \* 30mm 5V

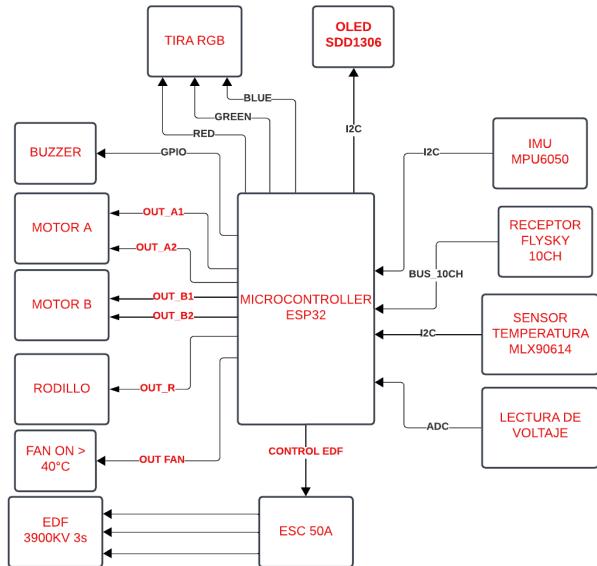


Figura 1: Diagrama de bloques del sistema

**I-A1. DIAGRAMA DE BLOQUES DEL SISTEMA:** El diagrama de bloques muestra la arquitectura general del robot limpia-vidrios \*HOMERO V1\*. El ESP32 actúa como el controlador central, gestionando entradas y salidas de múltiples subsistemas. A través de sus pines GPIO y comunicación I2C, se conecta a diversos actuadores y sensores: motores para tracción y rodillo (OUT\_A, OUT\_B, OUT\_R), una tira LED RGB, un buzzer y un ventilador EDF, controlado por un ESC de 50A. Los sensores MPU6050, MLX90614 y un divisor de voltaje proporcionan datos de orientación, temperatura y batería respectivamente, mientras que un receptor FlySky permite control remoto. Las flechas indican claramente la dirección de la señal entre cada bloque, diferenciando inputs y outputs.

## II. TRABAJO REALIZADO

### II-A. Planteamiento del sistema:

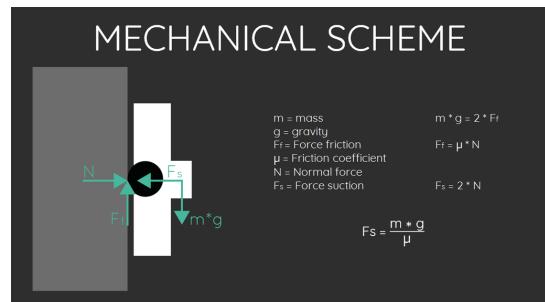


Figura 2: DCL del sistema Tomado de [1]

*II-A1. Diagrama de cuerpo libre::* Por medio del planteamiento del diagrama de cuerpo libre del sistema se obtiene la cantidad de empuje o succión necesaria para que el robot pueda mantenerse sujeto a paredes o techos, con la intención de poder adquirir un EDF con la fuerza necesaria para poder cumplir con la tarea de manera eficiente. Por lo tanto, se usó un peso ideal de 500gr, máximo 600gr según especificaciones de diseño.

Para los dos escenarios posibles se realizaron los siguientes cálculos:

CÁLCULO PARA  $m = 500 \text{ GR} = 0,5 \text{ KG}$

La fuerza de succión  $F_s$  se calcula con la siguiente fórmula:

$$F_s = \frac{m \cdot g}{\mu}$$

donde:

- ◊  $m = 0,5 \text{ kg}$  es la masa del robot,
- ◊  $g = 9,81 \text{ N/kg}$  es la aceleración debido a la gravedad,
- ◊  $\mu = 0,6$  es el coeficiente de fricción.

Sustituyendo los valores:

$$F_s = \frac{0,5 \text{ kg} \cdot 9,81 \text{ N/kg}}{0,6} = 8,175 \text{ N}$$

**Conversión a kilogramo-fuerza:**

$$\text{Kilogramo-fuerza} = \frac{8,175 \text{ N}}{9,81 \text{ N/kg}} = 0,833 \text{ kgf}$$

CÁLCULO PARA  $m = 600 \text{ GR} = 0,6 \text{ KG}$

$$F_s = \frac{0,6 \text{ kg} \cdot 9,81 \text{ N/kg}}{0,6} = 9,81 \text{ N}$$

**Conversión a kilogramo-fuerza:**

$$\text{Kilogramo-fuerza} = \frac{9,81 \text{ N}}{9,81 \text{ N/kg}} = 1 \text{ kgf}$$

Estos cálculos proporcionan el empuje necesario para que el robot se adhiera correctamente a las superficies, lo que permite seleccionar un EDF adecuado para cada escenario de peso. En este caso, el EDF adquirido posee un empuje total de 1.130 kgf [3],

Sabemos que el EDF adquirido tiene un empuje máximo de 1.130 kgf (100 % de su capacidad). Para calcular el empuje necesario para los pesos de 500 g y 600 g, utilizamos la siguiente regla de tres.

$$\text{Porcentaje de empuje (500 g)} = \left( \frac{0,833 \text{ kgf}}{1,130 \text{ kgf}} \right) \times 100 = 73,7 \%$$

Esto significa que el EDF necesitará el 73,7 % de su capacidad máxima para manejar un peso de 500 g.

Para un peso de 600g

El empuje necesario para un peso de 600 g es de 1.0 kgf. Usando la misma fórmula:

$$\text{Porcentaje de empuje (600 g)} = \left( \frac{1,0 \text{ kgf}}{1,130 \text{ kgf}} \right) \times 100 = 88,5 \%$$

Esto indica que el EDF necesitará el 88,5 % de su capacidad máxima para manejar un peso de 600 g.

De lo anterior se deduce un peso ideal de 500 gramos para que el empuje necesario sea de 0.833 kgf, lo cual mantiene el EDF dentro de un rango de operación eficiente sin forzarlo por debajo del 80 %.

*II-A2. Optoacoplación:* Se implementaron optoacopladores 4N35 y PC817 para garantizar el aislamiento eléctrico entre las señales de control (provenientes del ESP32) y las cargas inductivas controladas por BJT y MOSFET. Este aislamiento es crucial para prevenir que el ruido eléctrico generado por las cargas afecte el sistema de control y los componentes electrónicos sensibles del robot. Esto fue necesario ya que sin ellos todas las mediciones y señales se alteraban y distorsionaban gracias al ruido de los motores.

Los optoacopladores, como el 4N35 y el PC817, aseguran una separación efectiva entre las señales de baja potencia y las cargas de alta potencia. Esto permite mantener la estabilidad del sistema y evita interferencias que podrían comprometer el rendimiento. Gracias a esta implementación, se logró mejorar la fiabilidad del robot.

## II-B. Circuito del Rodillo

El **controlador PWM para el rodillo** permite ajustar su velocidad en un solo sentido. Este circuito utiliza:

- ◊ Un **MOSFET IRF44N** para el control de la carga.
- ◊ Un **optoacoplador PC817** para el aislamiento eléctrico de la señal de control.
- ◊ Un **diodo de protección** para mitigar corrientes inversas generadas por la carga inductiva del motor.
- ◊ Resistencias de polarización valores  $3K\Omega$  y  $10K\Omega$ .

El sistema es controlado por una señal PWM generada por el **ESP32**, que ajusta la velocidad del rodillo en función de los comandos recibidos del control remoto FlySky i6. En la figura 3 se observa de manera mas detallada el esquema de conexiones.

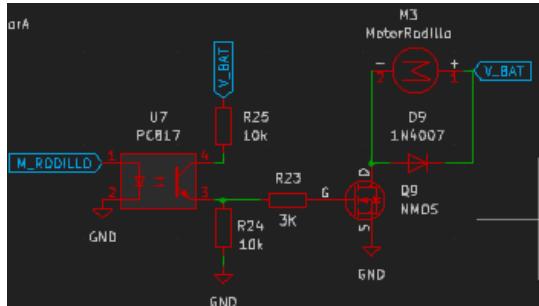


Figura 3: Esquema control rodillo PWM KICAD

*II-B1. Puente H para el Movimiento del Robot:* El robot incluye **dos puentes H bidireccionales**, diseñados con transistores **TIP41C** (NPN) y **TIP42C** (PNP), para permitir el movimiento en ambas direcciones. Estos puente H controlan los motores principales del robot y están configurados de la siguiente manera:

- ◊ **Control PWM:** Cada puente H utiliza dos señales PWM provenientes del ESP32 (cuatro señales en total).
- ◊ **Optoacopladores 4N35:** Se emplean dos optoacopladores por motor para aislar las señales de control.
- ◊ **Componentes adicionales:**
  - **4 diodos de protección** en cada puente H, para evitar daños por corrientes inversas.
  - Resistencias de polarización (valores de 1k y 10k).

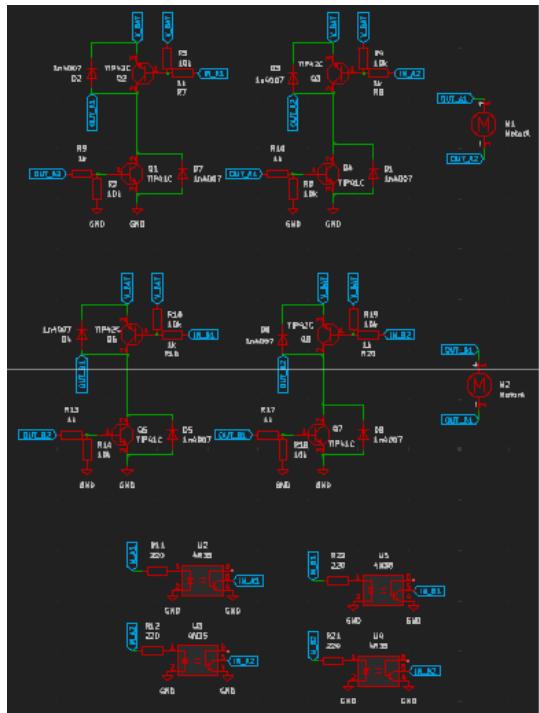


Figura 4: Esquema Puentes H Kicad

*II-B2. Sistema de refrigeración:* El sistema de refrigeración es fundamental para asegurar el funcionamiento estable y seguro del robot, especialmente en aplicaciones de alta

demandía de potencia, como la limpieza de superficies grandes. Este sistema es controlado por un sensor de temperatura MLX90614 conectado por medio del protocolo I2C, que monitoriza la temperatura del ESC (Electronic Speed Controller). Cuando la temperatura alcanza los 40°C, el sistema de refrigeración se activa automáticamente, evitando el sobrecaleamiento y protegiendo el ESC de posibles fallos térmicos. Esta función es crítica para asegurar que el ESC, que es responsable del control del motor con el mayor consumo de corriente del robot, opere de manera eficiente y sin interrupciones.

El ESC utilizado en este proyecto es un modelo de 50A a 3S, capaz de manejar potencias elevadas, alcanzando hasta 600W. Debido a la alta demanda de energía en condiciones de operación intensas, como la limpieza de superficies de gran tamaño, la disipación del calor se vuelve esencial. Para garantizar una adecuada disipación térmica, el sistema incluye un ventilador de 30x30 mm, que está montado de manera segura en una carcasa impresa en 3D, diseñada específicamente para facilitar la circulación del aire y la disipación del calor. La carcasa impresa en 3D también proporciona un soporte robusto y eficiente para el ventilador, asegurando que se mantenga en su lugar durante la operación.

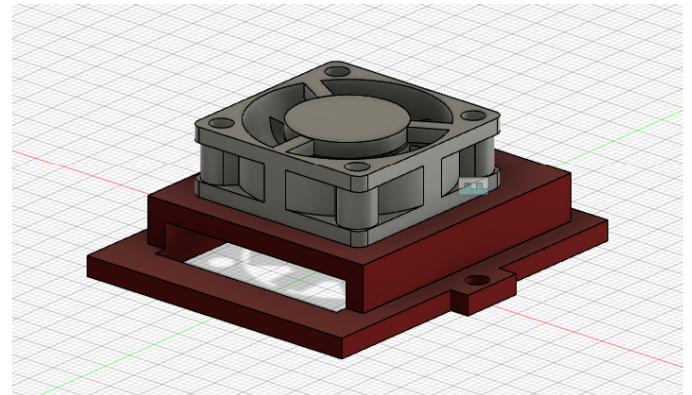


Figura 5: Sistema refrigeración

En anexos se puede visualizar parte del código de control de temperatura. 1

El sistema de refrigeración se controla electrónicamente mediante el uso de un MOSFET 2N7000 y un sensor de temperatura MLX90614. El ESP32 gestiona la activación del ventilador mediante la señal digital de su GPIO, que se conecta al Gate del 2N7000, así mismo este se encuentra con un pull down con una resistencia de 10K para asegurar que el estado siempre sea bajo a no ser que el GPIO esté activado. Cuando la temperatura del ESC supera los 40°C, el ESP32 envía una señal alta al 2N7000, lo que permite que el ventilador, alimentado por 5V, se encienda y comience a disipar el calor.

Para garantizar la seguridad y fiabilidad del sistema, se incluye un diodo de protección en el circuito. Este diodo está colocado en paralelo con el ventilador para proteger el MOSFET 2N7000 y otros componentes electrónicos de picos de voltaje y corrientes inversas que puedan generarse cuando

el ventilador se apaga.

En la figura 6 se puede observar de manera mas detallada las conexiones realizadas para el control electrónico de este subsistema.

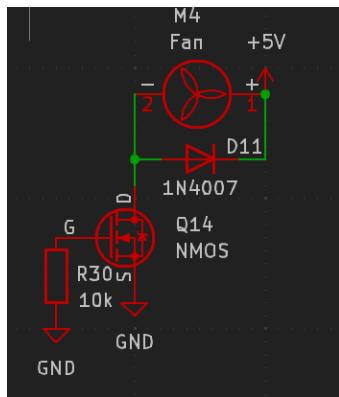


Figura 6: Esquema control de ventilador para refrigeración ESC KICAD

**II-B3. Divisor de Voltaje y Sistema de Alerta de Batería Baja:** Para monitorear el nivel de carga de la batería **3S LiPo**, se implementó un **divisor de voltaje** que reduce la tensión máxima de 12.6V (3S completamente cargada) a un rango seguro por debajo de 3.3V, compatible con el **ADC del ESP32**. Este divisor utiliza resistencias de **1800 Ω (R1)** y **600 Ω (R2)**, calculadas para mantener una corriente fija de referencia de 5 mA en el bus.

$$V_{\text{salida}} = V_{\text{entrada}} \times \frac{R2}{R1 + R2}$$

En este caso, la salida del divisor entrega un voltaje proporcional a la batería, que se conecta al **ADC del pin 25** del ESP32 para ser leído y procesado.

El voltaje reducido por el divisor se mide mediante el ADC del **pin 25** del ESP32. Este valor se convierte a un voltaje equivalente mediante un factor de calibración basado en las proporciones del divisor y las características del ADC. El software interpreta el nivel de voltaje y determina el estado de la batería en tiempo real.

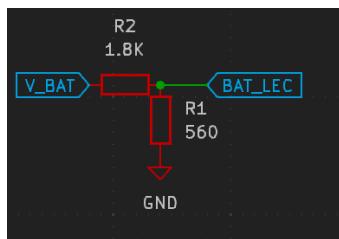


Figura 7: Esquema divisor de tensión para lectura batería KICAD

Si el voltaje de la batería cae por debajo de **9V** (umbral configurado para modo de batería baja), el sistema activa una alerta visual y auditiva que informa al usuario sobre el estado crítico de la batería. Este mecanismo asegura la protección del robot, evitando un agotamiento excesivo que podría dañar la batería o interrumpir el funcionamiento del sistema.

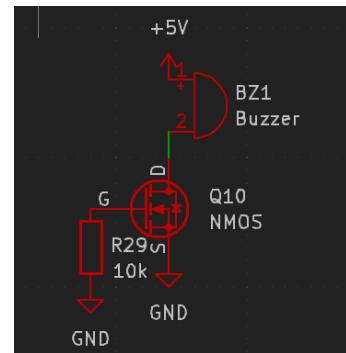


Figura 8: Esquema Buzzer pasivo KICAD

En el modo de batería baja, el sistema genera las siguientes acciones:

- ◊ Una **alerta visual** en la pantalla OLED sistema, indicando "LOW BATTERY".
- ◊ Una **alerta auditiva** mediante un buzzer pasivo conectado al sistema, emitiendo un tono periódico (300ms) a 2000Hz para informar al usuario.

En anexos se puede observar parte del código 2 usado para esta labor, para código completo revisar GitHub proyecto [2].

**II-B4. Indicador Visual con LEDs RGB y Pantalla OLED SSD1306 I2C:** Para proporcionar una señalización clara y visualmente intuitiva de los diferentes estados operativos del robot, se implementó un sistema de LEDs RGB controlado mediante **transistores MOSFET Canal N (2N7000)** que funcionan en modo interruptor (corte y satiración), en combinación con una **pantalla OLED** que muestra información adicional en tiempo real. Los LEDs RGB indican los modos de operación según la lógica mencionada, mientras que la pantalla OLED complementa la experiencia del usuario mostrando datos críticos del robot.

En la figura 9 se visualiza de manera mas clara el esquemático de conexiones.

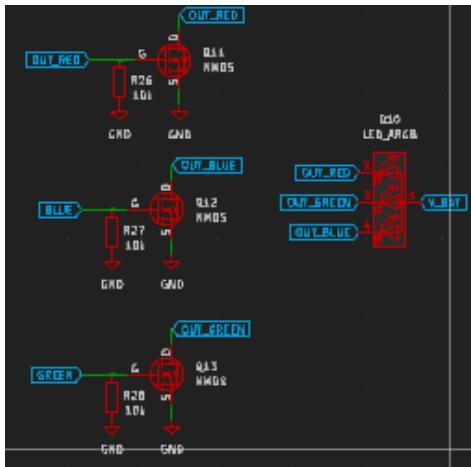


Figura 9: Esquema control RGB KICAD

Los LEDs RGB, controlados mediante transistores MOSFET y el **ESP32**, representan los siguientes estados del robot:

- ◊ **Rojo: Modo Seguro.** EDF y motores desactivados. El robot está bloqueado para evitar operaciones accidentales.
- ◊ **Azul: EDF Activo.** Solo el EDF está encendido, proporcionando adherencia a la superficie. Los motores de dirección (A y B) y el rodillo están bloqueados. (Modo para pruebas EDF).
- ◊ **Verde: Modo Operativo Completo.** EDF y motores (dirección y rodillo) están habilitados. Es el estado principal de funcionamiento.
- ◊ **Turquesa: Modo Restricción Parcial.** Solo los motores de dirección (A y B) y el rodillo están activos, mientras que el EDF permanece bloqueado. (Modo pruebas rodillo y motores)

Para el cambio entre modos de operación, se implementó el accionamiento manual mediante interruptores configurados en el control remoto FlySky, utilizando los canales 5, 6 y 10. El canal 5 permite alternar entre los modos de alimentación, seleccionando entre batería y fuente DC externa. El canal 6 controla el encendido y apagado del EDF, encargado de generar la succión necesaria para adherir el robot a las superficies. Finalmente, el canal 10 activa o desactiva los motores de tracción y del rodillo, asegurando una operación flexible y eficiente según las necesidades del entorno.

Además de los LEDs RGB, el robot incluye una pantalla OLED que proporciona información detallada y en tiempo real sobre el estado del sistema. Los datos mostrados incluyen:

- ◊ **Modo de Alimentación:** Indica si el robot está operando con **batería** o conectado a una **fuente de alimentación externa**.
- ◊ **Estado Operativo:** Según el modo del LED RGB, se muestra texto como:

- **Modo Verde:** ".EDF ON, MOTORS ON".
- **Modo Azul:** ".EDF ON, MOTORS OFF".
- **Modo Rojo:** "SAFE MODE, ALL OFF".
- **Modo Turquesa:** "MOTORS ON, EDF OFF".

- ◊ **Potencia del EDF:** En porcentaje, indicando el nivel de potencia actual del EDF.
- ◊ **Voltaje en Tiempo Real:** Lectura directa del voltaje de la batería mediante el divisor de voltaje.
- ◊ **Datos del Giroscopio:** Proporcionados por el **MPU6050**, mostrando los ángulos de inclinación en los ejes X e Y.
- ◊ **Canales de Control:** Lecturas de los canales más importantes del receptor FlySky, reflejando el estado de los joysticks para dirección y control de rodillo.

La pantalla OLED no solo cumple la función de indicar el estado operativo del robot, sino que también fue una herramienta clave durante las pruebas de validación de los módulos electrónicos. A través de ella, se monitorizaron las señales generadas con valores de -255 a 255 y procesadas por el **ESP32**, permitiendo medir, verificar y depurar el funcionamiento de los módulos de control, como los puentes H y el controlador PWM del rodillo.

*II-B5. Configuración del PWM y Librerías Utilizadas:* Durante las pruebas, se realizaron ajustes a la frecuencia y resolución del PWM en función de las necesidades de cada módulo:

- ◊ **Rodillo:** PWM configurado a una frecuencia de **5000 Hz** con una resolución de **8 bits**, permitiendo un control preciso de la velocidad.
- ◊ **Motores (puentes H):** PWM ajustado a **8000 Hz** y 8 bits de resolución, adecuado para la dinámica de los motores de dirección.
- ◊ **EDF:** Para el control del motor brushless del EDF, se utilizó la librería **ESP32Servo**, que genera pulsos PWM con una frecuencia de **50 o 60 Hz**, dependiendo del ESC. Este PWM es especial, ya que debe mantenerse constante sin interrupciones (**delay**) para evitar la pérdida de comunicación con el ESC, que de lo contrario entra en un estado de bloqueo y emite pitidos de error.

El proceso de armado del EDF se verificó durante las pruebas iniciales. Este proceso requiere emitir una serie de pulsos específicos al encenderse, indicando los valores mínimos y máximos del control. Una vez armado, el sistema mantiene pulsos constantes para ajustar la velocidad del motor, enviando un pulso equivalente a **0** cuando el EDF no está activo (por ejemplo, en los modos rojo o turquesa).

Para garantizar la comunicación constante con el ESC y evitar interrupciones en el control del EDF, se utilizó la función **millis()** en lugar de **delay()**. Esto permitió generar las señales PWM necesarias de manera eficiente, sin detener

otros procesos del sistema, asegurando una operación fluida y estable del robot en todos los modos de funcionamiento.

En el repositorio de GitHub del proyecto se puede observar de manera detallada los específicos del código [2].

**II-B6. Implementación del Protocolo iBUS:** El **protocolo iBUS** fue integrado para simplificar la comunicación entre el receptor **FlySky i6** y el **ESP32**. Este protocolo permite leer hasta **10 canales de control** mediante un único bus de datos, lo que reduce la complejidad del cableado y mejora la organización del sistema. El receptor se conectó al pin **RX2** del **ESP32**, garantizando una comunicación eficiente con el sistema.

Para implementar el iBUS, se diseñó una **clase personalizada** basada en la librería **iBUS BM**, que permite decodificar los datos recibidos del receptor y mapearlos directamente a valores PWM. Además, se desarrolló un método que facilita la lectura y manipulación de los canales de datos enviados por el receptor, optimizando la asignación de funciones en el robot. Por ejemplo, los canales se mapearon para controlar los motores de dirección, el rodillo y el EDF de forma precisa.

Toda la implementación, incluyendo el código fuente y las clases desarrolladas, está disponible en el **repositorio del proyecto** en GitHub [2].

#### II-C. Pruebas de Señales con Osciloscopio y succión:

Se empleó un osciloscopio para medir las señales en los puntos críticos del sistema, tanto en las entradas como en las salidas de los módulos:

- ◊ **Entrada a los módulos de control:** Se verificaron las señales PWM generadas por el **ESP32**, evaluando cómo se ajustaba el ciclo de trabajo en función de la posición de los joysticks del control **FlySky i6**.
- ◊ **Salidas de los módulos:** Se comprobó la amplificación y consistencia de las señales a la salida de los puentes H y del controlador PWM del rodillo.

Estas pruebas permitieron identificar y descartar errores de hardware, concentrando los ajustes necesarios únicamente en parámetros a nivel de software.

En la figura 20 se observa de manera más clara la señal PWM leída por el osciloscopio, esto se realizó con cada módulo y señal PWM de manera independiente.

### III. RESULTADOS Y ANÁLISIS DE RESULTADOS

Una vez que todos los módulos fueron soldados y estuvieron en funcionamiento, se realizó un prototipado rápido utilizando MDF de 3 mm, como se observa en la figura 11. El objetivo de este prototipo fue integrar todos los sistemas de manera similar a como se muestra en la figura 10. El prototipo final tenía unas dimensiones de 25 cm de ancho, 20 cm de largo y 6 cm de alto, sin contar con el motor ni el sistema de refrigeración. El peso final del prototipo fue de 947 g, lo que resultó ser un desafío para la sujeción en superficies verticales. Aunque en el techo funcionó de manera satisfactoria, en las paredes el robot logró mantenerse estático, pero no tenía suficiente fuerza para avanzar de abajo hacia arriba, lo que obligaba a posicionarse

de tal forma que la limpieza del vidrio se realizara de arriba hacia abajo.

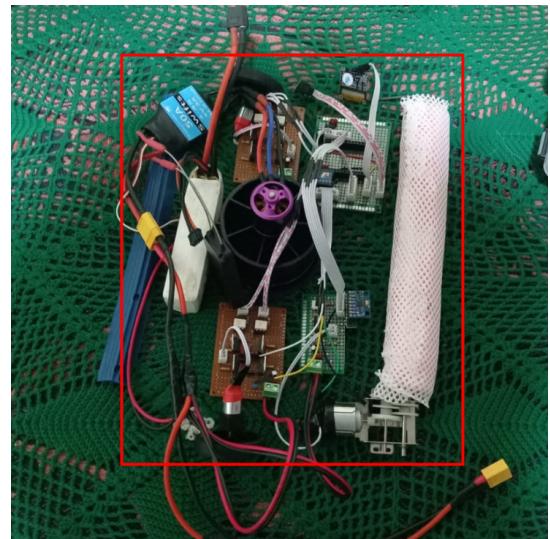


Figura 10: Planteamiento prototípico final



Figura 11: Diseño 3D

Este fallo se atribuye principalmente al peso excesivo del robot y la falta de fuerza necesaria en los motoreductores. El uso de madera en el prototipo, aunque funcional, contribuyó a un peso innecesario. Para futuras iteraciones, se podría optimizar este aspecto utilizando materiales más ligeros, como polipropileno (PP) mediante el uso de impresión 3D, ajustando las medidas del robot para mejorar la distribución del peso. La fabricación de una PCB más profesional también podría reducir el peso de la parte electrónica y permitir una distribución más eficiente del mismo, lo cual sería crucial para mejorar el rendimiento en superficies verticales.

En cuanto a los sistemas programados y gadgets implementados, como las alertas sonoras, visuales, el sistema de refrigeración y el movimiento, todos funcionaron correctamente. Esto demuestra que la parte electrónica y el software están funcionando de manera eficiente, y los problemas observados se deben más a la limitación física del prototipo actual.

En términos de mejora, se busca optimizar el diseño del robot, haciéndolo más pequeño y ligero, pero manteniendo la capacidad de adherencia y el rendimiento en la limpieza. Con los cálculos planteados en el diagrama de cuerpo libre, se puede trabajar en el desarrollo de mejores motores y en un sistema de tracción más eficiente, tal vez utilizando un sistema de orugas que distribuya mejor la fuerza sobre las superficies verticales.

En cuanto a la parte electrónica y de programación, se planea diseñar el puente H utilizando MOSFET de canal N únicamente, lo que mejoraría la eficiencia y permitiría escalar a motores de mayor corriente. Además, se pretende desarrollar una interfaz de control autónomo, que permita gestionar el robot mediante encoders, mejorando la precisión del movimiento y la autonomía del sistema. Finalmente, se investigarán estrategias de limpieza más óptimas, que permitan al robot lograr una buena durabilidad, ligereza y eficiencia en la limpieza de superficies.

El prototipo final puede ser observado en la figura 12, donde se muestra en modo seguro.



Figura 12: Prototipo Final HOMERO VI

#### IV. CONCLUSIONES

El desarrollo del proyecto *HOMERO VI* ha permitido aplicar y consolidar diversos conceptos fundamentales de la ingeniería electrónica y eléctrica en un contexto práctico. A lo largo de la implementación, se han utilizado componentes electrónicos clave, como transistores BJT, MOSFET, y optoacopladores, esenciales para controlar actuadores y gestionar la interacción entre los sistemas de control del robot. Estos conceptos son fundamentales en la industria electrónica y se emplean de manera constante en el diseño de sistemas automatizados, robots y otros dispositivos de control de precisión en la vida real.

En particular, el uso de **señales PWM, control de motores mediante puentes H, y la comunicación mediante protocolos como IBUS o radiofrecuencia** son tecnologías comunes en el mundo laboral de la ingeniería electrónica, especialmente en la robótica y sistemas de automatización industrial. La implementación del protocolo IBUS y su mapeo a señales PWM es un ejemplo de cómo los principios teóricos de control y comunicación se traducen en soluciones prácticas y eficientes que se aplican en productos y sistemas comerciales.

El proyecto también ha permitido experimentar con el **diseño de PCB** como una mejora clave en la parte electrónica. En lugar de usar una solución de prototipos basados en baquelita, se planea realizar un diseño de PCB profesional para reducir el peso, mejorar la organización y aumentar la fiabilidad de los componentes electrónicos. Este paso es fundamental en el ámbito profesional, donde el **diseño de PCB optimizado** es crucial para garantizar la durabilidad y el rendimiento de los dispositivos.

El **rediseño mecánico** también juega un papel fundamental en la mejora del proyecto. El uso de materiales más ligeros como el *polipropileno (PP)* en la *impresión 3D* permitirá optimizar el tamaño y peso del robot, lo cual mejorará la eficiencia en su movimiento sobre superficies verticales. Se planea ajustar las medidas del robot y modificar la estructura mecánica para distribuir de manera más eficiente el peso y la fuerza generada por los actuadores.

En cuanto a los sistemas programados y gadgets implementados, como las alertas sonoras, visuales, el sistema de refrigeración y el movimiento, todos funcionaron correctamente. Esto demuestra que la parte electrónica y el software están funcionando de manera eficiente, y los problemas observados se deben más a la limitación física del prototipo actual.

A futuro, se busca optimizar el diseño del robot, haciéndolo más pequeño y ligero, pero manteniendo la capacidad de adherencia y el rendimiento en la limpieza. Con los cálculos planteados en el diagrama de cuerpo libre, se puede trabajar en el desarrollo de un sistema de tracción más eficiente, tal vez utilizando un sistema de orugas que distribuya mejor la fuerza sobre las superficies verticales.

En términos de mejora en la parte electrónica y de programación, se planea diseñar el puente H utilizando **MOSFET de canal N únicamente**, lo que mejoraría la eficiencia y permitiría escalar a motores de mayor corriente. Además, se

pretende desarrollar una interfaz de control autónomo, que permita gestionar el robot mediante *encoders*, mejorando la precisión del movimiento y la autonomía del sistema. Finalmente, se investigarán estrategias de limpieza más óptimas, que permitan al robot lograr una buena durabilidad, ligereza y eficiencia en la limpieza de superficies.

## V. AGRADECIMIENTOS

Quiero agradecer profundamente a Dios, por darme las fuerzas para seguir adelante y por acompañarme en cada paso, permitiéndome seguir persiguiendo mis sueños.

A mi madre, **Martha Giraldo**, por su apoyo constante, su amor incondicional y por los valiosos consejos de limpieza y diseño, ya que trabaja en el área de aseo doméstico. Su sabiduría me ha ayudado a mejorar y aplicar los conocimientos adquiridos en este proyecto.

A mi padre, **Rodolfo Alfonso**, porque a pesar de la distancia, siempre lo tengo presente en mi vida. Él es uno de mis maestros más grandes y me ha enseñado lecciones valiosas que llevo conmigo cada día.

A la profesora **Alba Ávila**, profesora magistral del curso, por compartir sus conocimientos y por motivarme a superarme. Su orientación me ha impulsado a dar siempre lo mejor de mí y a seguir aprendiendo.

A **Sebastián Zapata**, asistente graduado del curso, por sus valiosas explicaciones y retroalimentaciones en el laboratorio. Gracias a su apoyo, pude comprender mejor los conceptos y llevarlos a la práctica de manera efectiva.

A todos ustedes, ¡muchas gracias!

## REFERENCIAS

- [1] Instructables, “GECO - Wall climbing robot,” Instructables, Sep. 21, 2017. <https://www.instructables.com/GECO-Wall-Climbing-Robot/>
- [2] Alfonso J.D., /HOMERO-V1: Repository containing information about the development of the prototype for the HOMERO V1 window-cleaning robot, final project for the subject of electronics fundamentals,” GitHub. <https://github.com/JuanD1098/HOMERO-V1>
- [3] “Hobbyocean-avión EDF de 64mm con 11 aspas, 4S de ventilador con conductos, 3500KV, 3S, 3900KV, 50A, ESC, para avión teledirigido - AliExpress,” Aliexpress. [https://es.aliexpress.com/item/1005007031481465.html?spm=a2g0o.order\\_list.order\\_list\\_main.5.3fe8194dSmznV0&gatewayAdapt=glo2esp](https://es.aliexpress.com/item/1005007031481465.html?spm=a2g0o.order_list.order_list_main.5.3fe8194dSmznV0&gatewayAdapt=glo2esp)
- [4] R. Mischianti, “ESP32: pinout, specs and Arduino IDE configuration - Part 1,” Renzo Mischianti, Dec. 17, 2021. <https://mischianti.org/esp32-pinout-specs-and-arduino-ide-configuration-part-1/>

## VI. ANEXOS

Listing 1: Control de temperatura

```
1 // Control del ventilador
2     if (temperatura >= 40.0) {
3         digitalWrite(FAN, HIGH);
4     } else {
5         digitalWrite(FAN, LOW);
6     }
```

Listing 2: Alerta Bateria Baja

```
1 if (BATT == -255 && !ESC_ON) {
```

```
3     if (voltage <= 9.5 && ...
4         !batteryAlertTriggered) {
5             // Mostrar mensaje de alerta solo la ...
6             // primera vez
7             screen.clearDisplay();
8             screen.printTextM("LOW", 40, 6);
9             screen.printTextM("BATTERY", 10, 30);
10            screen.display();
11
12            tone(BUZZER_PIN, 2000, 300); // ...
13            // Emitir tono de alerta
14            delay(300); // ...
15            // Permitir que el buzzer termine
16            noTone(BUZZER_PIN); // ...
17            // Apagar el buzzer
18            batteryAlertTriggered = true; // ...
19            // Activar la bandera de alerta
20            esc.speedESC(0); // Apagar ESC
21            return; // Salir del loop para ...
22            // evitar actualizaciones adicionales
23        }
24
25    } else {
26        batteryAlertTriggered = false; // ...
27        // Resetear bandera si el voltaje ...
28        // vuelve a niveles normales
29    }
30}
```



Figura 13: EDF 64mm 390KV ESC 50A 3s [3]

ESP32 DEV KIT V1 PINOUT

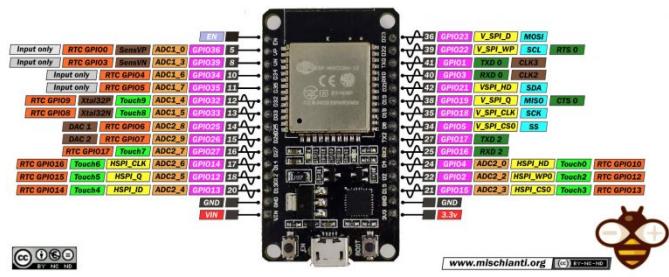


Figura 14: ESP32 Dev Kit V1 Tomado de [4]



Figura 15: Diseño 3D

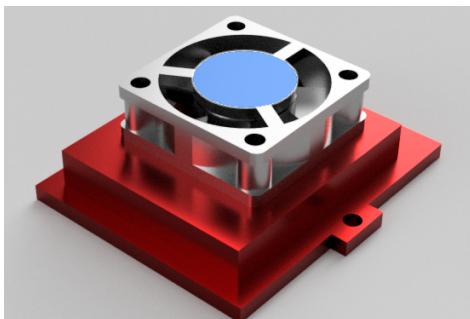


Figura 16: Sistema refrigeración

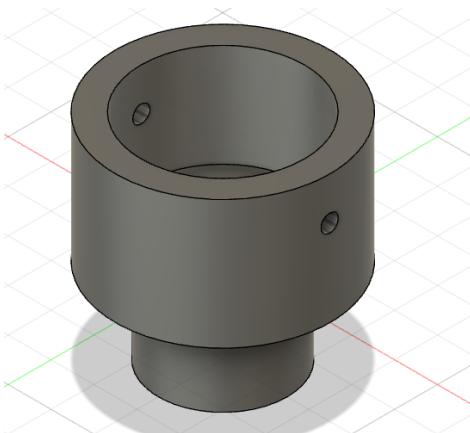


Figura 17: Pieza rodillo

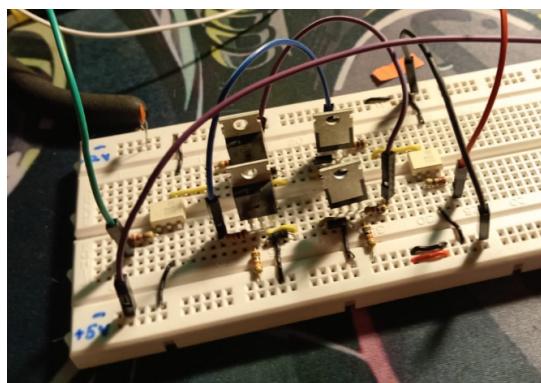


Figura 18: Puente H en protoboard

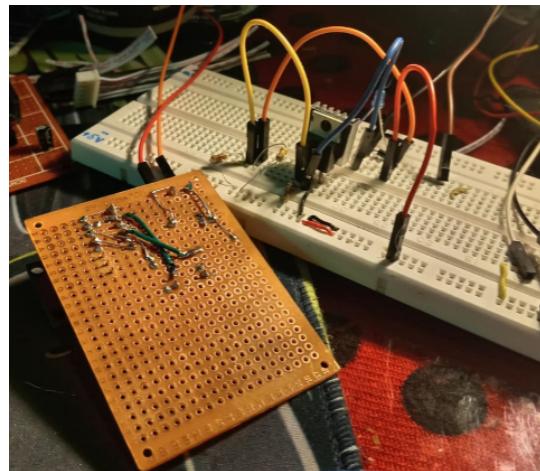


Figura 19: Soldadura en baquela y controlador rodillo en protoboard



Figura 20: prueba de señales PWM con osciloscopio

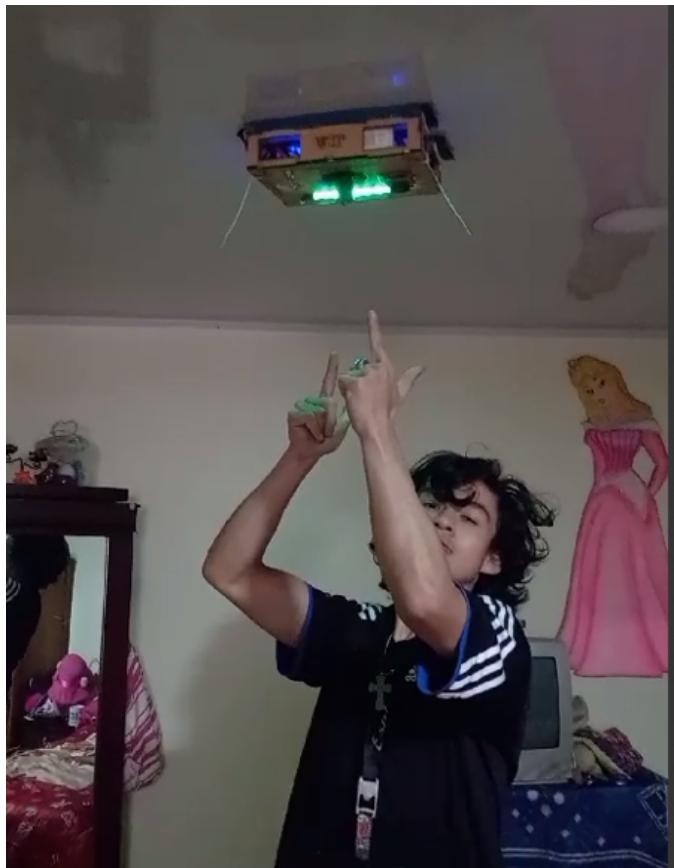


Figura 21: Prueba succión techo