



**DIVISIÓN DE ESTUDIOS POSGRADO E INVESTIGACIÓN**

Opción 1.- TESIS

**TRABAJO PROFESIONAL**

**IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO REMOTO  
EN VEHÍCULO ELÉCTRICO MEDIANTE EL USO DE  
MICROCONTROLADORES**

**QUE PARA OBTENER EL GRADO DE:**

**MAESTRO EN  
INGENIERÍA ELECTRÓNICA**

**PRESENTA:**

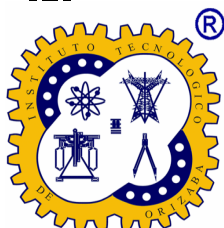
**JOSÉ OSCAR MARTÍNEZ DÍAZ**

**DIRECTOR DE TESIS:**

**JOSÉ PASTOR RODRIGUEZ JARQUIN**

**CODIRECTOR DE TESIS:**

**JUAN CARLOS NÚÑEZ DORANTES**





Instituto Tecnológico de Orizaba  
Departamento de División de Estudios de Posgrado e Investigación

Orizaba, Veracruz, **18/febrero/2026**  
Dependencia: **División de Estudios de  
Posgrado e Investigación**  
Asunto: **Autorización de Impresión**  
OPCION: I

**C. JOSÉ OSCAR MARTÍNEZ DÍAZ**  
Candidato(a) a Grado de Maestro(a) en:  
**INGENIERÍA ELECTRÓNICA**  
**P R E S E N T E.-**

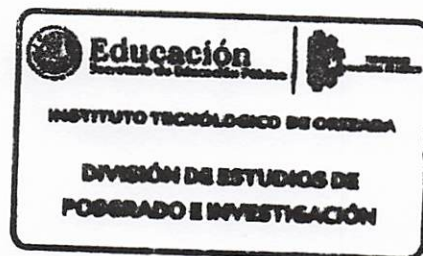
De acuerdo con el Reglamento de Titulación vigente de los Centros e Institutos Tecnológicos Federales del Tecnológico Nacional de México, de la Secretaría de Educación Pública, y habiendo cumplido con todas las indicaciones que la Comisión Revisora le hizo respecto a su Trabajo Profesional titulado:

**" IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO REMOTO EN VEHÍCULO  
ELÉCTRICO MEDIANTE EL USO DE MICROCONTROLADORES".**

Comunico a Usted que este Departamento concede su autorización para que proceda a la impresión del mismo.

**ATENTAMENTE**  
Excelencia en Educación Tecnológica®  
**CIENCIA – TÉCNICA - CULTURA®**

**DR. OSCAR BÁEZ SENTÍES**  
**JEFE DE LA DIVISIÓN DE ESTUDIOS  
DE POSGRADO E INVESTIGACIÓN**



OG-13-F06



**2026**  
año de  
**Margarita  
Maza**





**Instituto Tecnológico de Orizaba**  
Departamento de División de Estudios de Posgrado e Investigación

Orizaba, Veracruz, 13/febrero/2026  
Asunto: **Revisión de trabajo escrito**

**C. DR. OSCAR BÁEZ SENTÍES**  
JEFE DE LA DIVISIÓN DE ESTUDIOS  
DE POSGRADO E INVESTIGACIÓN  
**P R E S E N T E.-**

Los que suscriben, miembros del jurado, han realizado la revisión de la Tesis del (la) C.

**JOSÉ OSCAR MARTÍNEZ DÍAZ**

La cual lleva el título de:

**“IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO REMOTO EN VEHÍCULO ELÉCTRICO MEDIANTE  
EL USO DE MICROCONTROLADORES”**

Y concluyen que se acepta.

**ATENTAMENTE**

Excelencia en Educación Tecnológica®  
CIENCIA – TÉCNICA - CULTURA®

PRESIDENTE: DR. JOSÉ PASTOR RODRÍGUEZ JARQUIN

SECRETARIO: DR. ALBINO MARTÍNEZ SIBAJA

VOCAL: DR. IGNACIO HERRERA AGUILAR

VOCAL SUP.: DR. JUAN CARLOS NUÑEZ DORANTES

FIRMA  
  
FIRMA  
  
FIRMA  
  
FIRMA

TA-09-18



**2026**  
año de  
**Margarita  
Maza**



# CONTENIDO

|                                                                                         |    |
|-----------------------------------------------------------------------------------------|----|
| RESUMEN .....                                                                           | 13 |
| ABSTRACT .....                                                                          | 14 |
| INTRODUCCIÓN .....                                                                      | 15 |
| PLANTEAMIENTO DEL PROBLEMA .....                                                        | 16 |
| JUSTIFICACIÓN .....                                                                     | 18 |
| HIPÓTESIS .....                                                                         | 19 |
| OBJETIVOS .....                                                                         | 20 |
| OBJETIVO GENERAL .....                                                                  | 20 |
| OBJETIVOS ESPECÍFICOS .....                                                             | 20 |
| LIMITACIONES Y ALCANCES .....                                                           | 21 |
| Alcance del Proyecto .....                                                              | 21 |
| Limitaciones .....                                                                      | 21 |
| CAPÍTULO 1: ESTADO DEL ARTE .....                                                       | 22 |
| 1.1  Sistemas IoT y monitoreo remoto: de vivienda a industria .....                     | 22 |
| 1.1.1  Interfaces gráficas y usabilidad: GUI y TUI .....                                | 23 |
| 1.1.2  Generación y adaptación computacional de interfaces .....                        | 24 |
| 1.1.3  SCADA/ICS y ciberseguridad: riesgos y estrategias .....                          | 24 |
| 1.1.4  SCADA con interacción avanzada: realidad aumentada .....                         | 26 |
| 1.1.5  Vehículos eléctricos: autonomía, energía, carga y necesidades de monitoreo ..... | 26 |
| 1.1.6  Brecha identificada y orientación de la tesis .....                              | 28 |

|                                                                                                             |    |
|-------------------------------------------------------------------------------------------------------------|----|
| CAPÍTULO 2: MARCO TEÓRICO .....                                                                             | 30 |
| 2.1 Vehículo Eléctrico .....                                                                                | 30 |
| 2.1.1 Funcionamiento General de un Vehículo Eléctrico .....                                                 | 30 |
| 2.1.2 Estructura General de un Vehículo Eléctrico .....                                                     | 30 |
| 2.1.3 Especificaciones Técnicas Clave.....                                                                  | 31 |
| 2.2 Caso de Estudio: Scooter Eléctrico Keeway Ride E1 modelo 2015.....                                      | 32 |
| 2.2.1 Configuración Técnica Original del Vehículo .....                                                     | 33 |
| 2.2.2 Modificaciones Implementadas .....                                                                    | 33 |
| 2.2.3 Subsistema de Control: El Controlador PWM de 60V/30A .....                                            | 34 |
| 2.3 Funcionamiento del Motor BLDC en Scooters eléctricos .....                                              | 35 |
| 2.3.1 Principio de Operación .....                                                                          | 36 |
| 2.3.2 Detección de Posición del Rotor: Sensored vs. Sensorless .....                                        | 37 |
| 2.4 Funcionamiento de los Sensores de Efecto Hall en el Sistema de Propulsión de una Scooter Eléctrica..... | 37 |
| 2.4.1 Principio Físico Fundamental: El Efecto Hall.....                                                     | 38 |
| 2.4.2 Implementación en el Motor BLDC: Estructura y Colocación .....                                        | 38 |
| 2.4.3 Comunicación con el Controlador y el Proceso de Conmutación.....                                      | 39 |
| 2.4.4 Función Integral en el Rendimiento del Vehículo .....                                                 | 41 |
| 2.5 Bobinas Toroidales como Detectores de Presencia de Voltaje en Motores BLDC .....                        | 42 |
| 2.5.1 Principio de Operación .....                                                                          | 42 |
| 2.5.2 Aplicación Específica en el Sistema de Monitoreo .....                                                | 43 |
| 2.5.3 Ventajas Clave para el Proyecto.....                                                                  | 43 |

|         |                                                               |    |
|---------|---------------------------------------------------------------|----|
| 2.6     | Sensor ACS712.....                                            | 44 |
| 2.6.1   | Principio de Operación y Arquitectura Interna.....            | 44 |
| 2.6.2   | Especificaciones Técnicas del Modelo ACS712.....              | 46 |
| 2.7     | Sensor ZMPT101B.....                                          | 47 |
| 2.7.1   | Principio de Operación y Arquitectura.....                    | 47 |
| 2.7.1.1 | Transformación y Aislamiento (Etapa de Entrada).....          | 48 |
| 2.7.1.2 | Acondicionamiento de Señal (Etapa de Salida).....             | 48 |
| 2.7.2   | Especificaciones Técnicas del Módulo y del Transformador..... | 49 |
| 2.7.3   | Características de la Señal de Salida y Procesamiento.....    | 50 |
| 2.8     | Raspberry Pi 4.....                                           | 51 |
| 2.8.1   | Especificaciones Técnicas Fundamentales.....                  | 52 |
| 2.8.2   | Capacidades de Entrada/Salida y Conectividad.....             | 53 |
| 2.8.2.1 | Puertos de Expansión.....                                     | 53 |
| 2.8.2.2 | Conectividad de Red y Periféricos.....                        | 54 |
| 2.8.2.3 | Salida de Video y Multimedia.....                             | 54 |
| 2.8.3   | Características Físicas y de Potencia.....                    | 54 |
| 2.9     | Raspberry Pi Pico W.....                                      | 55 |
| 2.9.1   | Características Técnicas del Hardware.....                    | 55 |
| 2.9.2   | El Sistema de Convertidor Analógico-Digital (ADC).....        | 56 |
| 2.9.3   | Configuración de los GPIO para el Sistema de Monitoreo.....   | 57 |
| 2.9.4   | Rol en la Arquitectura Distribuida.....                       | 58 |

|          |                                                                           |    |
|----------|---------------------------------------------------------------------------|----|
| 2.10     | Filtros Digitales.....                                                    | 58 |
| 2.10.1   | Clasificación Fundamental: Filtros FIR e IIR.....                         | 59 |
| 2.11     | Filtro Pasa-Bajas IIR de Primer Orden.....                                | 59 |
| 2.11.1   | Nomenclatura .....                                                        | 60 |
| 2.11.2   | Ecuación en Diferencias y Función de Transferencia .....                  | 60 |
| 2.11.2.1 | Función de Transferencia en el Dominio Z.....                             | 60 |
| 2.11.3   | Relación con el Filtro Analógico RC y Diseño del Parámetro $\alpha$ ..... | 61 |
| 2.11.4   | Respuesta al Impulso, en Frecuencia y Comparación con FIR.....            | 62 |
| 2.11.4.1 | Respuesta al Impulso $h[n]$ .....                                         | 62 |
| 2.11.4.2 | Respuesta en Frecuencia.....                                              | 62 |
| 2.11.5   | Consideraciones para la Implementación Práctica .....                     | 63 |
| 2.12     | Filtro de Mediana .....                                                   | 63 |
| 2.12.1   | Base Estadística y Principio de Operación .....                           | 64 |
| 2.12.2   | Descripción Algorítmica y Ejemplo.....                                    | 64 |
| 2.12.3   | Propiedades y Análisis de Rendimiento del Filtro de Mediana .....         | 65 |
| 2.12.3.1 | Preservación de Bordes y Señales Escalón .....                            | 65 |
| 2.12.4   | Características de la Señal de Salida y Limitaciones .....                | 65 |
| 2.12.5   | Directrices de Diseño e Implementación Práctica .....                     | 65 |
| 2.12.5.1 | Selección del Tamaño de Ventana (N) .....                                 | 65 |
| 2.12.6   | Manejo de Bordes de la Señal .....                                        | 66 |
| 2.13     | Protocolo de Datagramas de Usuario (UDP) .....                            | 66 |
| 2.13.1   | Naturaleza y Propósito.....                                               | 66 |

|                               |                                                            |    |
|-------------------------------|------------------------------------------------------------|----|
| 2.13.2                        | Características Técnicas Principales .....                 | 67 |
| 2.13.3                        | Ventajas y Limitaciones .....                              | 67 |
| 2.14                          | Señales Críticas para el Monitoreo .....                   | 68 |
| 2.14.0.1                      | Señales de los Sensores de Efecto Hall.....                | 68 |
| 2.14.1                        | Detección de Presencia de Voltaje en Fases.....            | 69 |
| 2.14.2                        | Medición de Voltaje por Fase .....                         | 69 |
| 2.14.3                        | Medición de Corriente por Fase .....                       | 70 |
| CAPÍTULO 3: METODOLOGÍA ..... |                                                            | 71 |
| 3.1                           | Señales de interes del motor BLCD.....                     | 71 |
| 3.2                           | Configuración para monitoreo de Detección de Voltaje ..... | 72 |
| 3.2.1                         | Arquitectura de Adquisición y Procesamiento Digital .....  | 73 |
| 3.3                           | Configuración para monitoreo en sensor de efecto Hall..... | 75 |
| 3.3.1                         | Arquitectura del código .....                              | 78 |
| 3.4                           | Configuración para Monitoreo de Voltaje de Fases .....     | 80 |
| 3.4.0.1                       | Conexión Fase-Fase para Alimentación del ZMPT101B.....     | 80 |
| 3.4.1                         | Algoritmo de Adquisición y Procesamiento .....             | 81 |
| 3.4.1.1                       | Estrategia de Muestreo para Alta Frecuencia (300Hz) .....  | 81 |
| 3.4.1.2                       | Cálculo del Valor RMS por Bloque .....                     | 82 |
| 3.4.1.3                       | Calibración y Compensación de Errores.....                 | 82 |
| 3.4.1.4                       | Filtrado Digital IIR para Suavizado .....                  | 83 |
| 3.4.1.5                       | Transmisión de Datos y Protocolo.....                      | 83 |
| 3.4.2                         | Consideraciones de Precisión y Limitaciones .....          | 83 |

|         |                                                         |    |
|---------|---------------------------------------------------------|----|
| 3.5     | Configuración para Monitoreo de Corriente de Fases..... | 84 |
| 3.5.1   | Configuración del Sistema de Medición.....              | 84 |
| 3.5.1.1 | Conexión Serie de los Módulos ACS712.....               | 84 |
| 3.5.1.2 | Alimentación de los Módulos ACS712.....                 | 85 |
| 3.5.2   | Algoritmo de Adquisición y Procesamiento.....           | 85 |
| 3.5.2.1 | Principio de Operación del ACS712.....                  | 85 |
| 3.5.2.2 | Estrategia de Muestreo para Cálculo RMS.....            | 86 |
| 3.5.2.3 | Calibración de Offset (Punto Cero).....                 | 86 |
| 3.5.2.4 | Cálculo del Valor RMS de Corriente.....                 | 87 |
| 3.5.2.5 | Filtrado de Mediana para Rechazo de Picos.....          | 87 |
| 3.5.2.6 | Transmisión de Datos y Protocolo UDP.....               | 88 |
| 3.5.3   | Consideraciones de Precisión y Limitaciones.....        | 88 |
| 3.5.3.1 | Errores Introducidos por el Sistema de Adquisición..... | 88 |
| 3.6     | Interfaz de Usuario en Raspberry Pi 4.....              | 89 |
| 3.6.1   | Arquitectura del Sistema de Visualización.....          | 89 |
| 3.6.2   | Configuración del Sistema de Comunicación UDP.....      | 90 |
| 3.6.2.1 | Inicialización del Socket UDP.....                      | 90 |
| 3.6.2.2 | Mecanismo de Recepción y Procesamiento.....             | 90 |
| 3.6.3   | Diseño y Configuración de la Interfaz Gráfica.....      | 92 |
| 3.6.3.1 | Estructura de Ventana Principal.....                    | 92 |
| 3.6.3.2 | Sistema de Estilos QSS.....                             | 93 |
| 3.6.3.3 | Widgets Personalizados.....                             | 94 |

|                                                  |                                                         |     |
|--------------------------------------------------|---------------------------------------------------------|-----|
| 3.6.4                                            | Mecanismo de Actualización en el Tiempo .....           | 96  |
| 3.6.4.1                                          | Sistema de Temporizadores .....                         | 96  |
| 3.6.4.2                                          | Actualización de Widgets .....                          | 96  |
| 3.6.4.3                                          | Gestión de Recursos .....                               | 97  |
| 3.6.5                                            | Integración del Sistema Completo .....                  | 98  |
| CAPÍTULO 4: RESULTADOS .....                     |                                                         | 99  |
| 4.1                                              | Cableado del motor y estado de la scooter .....         | 99  |
| 4.1.1                                            | Conexiones del motor hacia el controlador .....         | 100 |
| 4.1.2                                            | Voltajes entre fases .....                              | 102 |
| 4.1.3                                            | Medición de corriente de cada fase .....                | 104 |
| 4.2                                              | Vista de la Interfaz de Usuario en Raspberry Pi 4 ..... | 106 |
| 4.3                                              | Detección de Voltaje en las Fases del Motor .....       | 110 |
| 4.4                                              | Detección de Señales en Sensores de Efecto Hall .....   | 118 |
| 4.5                                              | Medición de Corriente en las Fases .....                | 121 |
| 4.6                                              | Medición de Voltaje entre Fase-Fase .....               | 128 |
| CONCLUSIONES .....                               |                                                         | 136 |
| Evaluación del Sistema de Monitoreo Remoto ..... |                                                         | 136 |
| Análisis por Variable Monitoreada .....          |                                                         | 136 |
| Detección de Voltaje en Fases .....              |                                                         | 136 |
| Monitoreo de Sensores de Efecto Hall .....       |                                                         | 137 |
| Medición de Corrientes por Fase .....            |                                                         | 138 |
| Medición de Voltajes Fase-Fase .....             |                                                         | 138 |

|                                        |     |
|----------------------------------------|-----|
| Contribuciones Técnicas .....          | 139 |
| Ventajas del Sistema Implementado..... | 140 |
| Ventajas Técnicas .....                | 140 |
| Desafíos Superados.....                | 140 |
| Conclusiones Finales .....             | 141 |
| REFERENCIAS BIBLIOGRÁFICAS .....       | 142 |

## LISTA DE FIGURAS

|            |                                                                                                                                         |    |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 1.  | Arquitectura general de un sistema SCADA .....                                                                                          | 26 |
| Figura 2.  | Estructura general de un Vehículo Eléctrico .....                                                                                       | 30 |
| Figura 3.  | Caso de Estudio Keeway Ride E1 .....                                                                                                    | 32 |
| Figura 4.  | Ejemplo de Motor BLCD .....                                                                                                             | 36 |
| Figura 5.  | Esquema General del circuito para Motor BLCD .....                                                                                      | 37 |
| Figura 6.  | Ejemplo ilustrativo del sensor de efecto Hall .....                                                                                     | 38 |
| Figura 7.  | Ejemplo Ilustrativo sobre la posición de los sensores de efecto Hall en el motor BLCD .....                                             | 39 |
| Figura 8.  | Gráfica ilustrativa sobre el funcionamiento de la detección de los sensores de efecto Hall en los 360 grados eléctricos del motor ..... | 41 |
| Figura 9.  | Ejemplos ilustrativos de bobinas toroidales .....                                                                                       | 42 |
| Figura 10. | Sensor ACS712 .....                                                                                                                     | 44 |
| Figura 11. | Diagrama a bloques del funcionamiento ACS712 .....                                                                                      | 45 |
| Figura 12. | Sensor ZMPT101B.....                                                                                                                    | 47 |
| Figura 13. | Diagrama del circuito ZMPT101B.....                                                                                                     | 49 |
| Figura 14. | Raspberry Pi 4.....                                                                                                                     | 52 |
| Figura 15. | Esquema de pines GPIO para Raspberry Pi 4.....                                                                                          | 53 |
| Figura 16. | Raspberry Pi 4.....                                                                                                                     | 55 |
| Figura 17. | Esquema de puertos para Raspberry Pi Pico W .....                                                                                       | 58 |
| Figura 18. | Esquema de comparativa protocolo UDP contra TCP .....                                                                                   | 67 |
| Figura 19. | Diagrama de conexiones Motor BLCD hacia controlador .....                                                                               | 72 |
| Figura 20. | Circuito rectificador de onda completa tipo puente.....                                                                                 | 73 |
| Figura 21. | Convertidor Nivel Lógico de 5 volts - 3.3 volts .....                                                                                   | 76 |
| Figura 22. | Diagrama General de conexión del Sensor de Efecto Hall (El modelo es solo un ejemplo) .....                                             | 77 |
| Figura 23. | Diagrama General de conexión para monitoreo de señales de los sensores de efecto Hall con Raspberry Pi Pico W .....                     | 78 |
| Figura 24. | Diagrama General de conexión para monitoreo de voltajes entre fases con Raspberry Pi Pico W.....                                        | 81 |
| Figura 25. | Diagrama General de conexión para monitoreo de corrientes de fases con Raspberry Pi Pico W.....                                         | 85 |

|            |                                                                                                                 |     |
|------------|-----------------------------------------------------------------------------------------------------------------|-----|
| Figura 26. | Convertidor CA/CA para cargador de la scooter .....                                                             | 99  |
| Figura 27. | Cargador de la Scooter Keeway E1 .....                                                                          | 100 |
| Figura 28. | Cableado de fases provenientes del motor hacia el controlador (azul, verde y amarillo) .....                    | 100 |
| Figura 29. | Conexión de sensores de efecto Hall provenientes del motor hacia controlador .....                              | 101 |
| Figura 30. | Medición al 100 % de aceleración del voltaje entre fases amarillo y verde (Y-G) .....                           | 102 |
| Figura 31. | Medición al 100 % de aceleración del voltaje entre fases amarillo y azul (Y-B) .....                            | 103 |
| Figura 32. | Medición al 100 % de aceleración del voltaje entre fases azul y verde (B-G) .....                               | 103 |
| Figura 33. | Medición al 100 % de aceleración de la corriente de la fase amarilla (Y) .....                                  | 104 |
| Figura 34. | Medición al 100 % de aceleración de la corriente de la fase azul (B) .....                                      | 105 |
| Figura 35. | Medición al 100 % de aceleración de la corriente de la fase verde (G) .....                                     | 105 |
| Figura 36. | Pestaña 1 de la Interfaz (Detección de Voltaje) .....                                                           | 107 |
| Figura 37. | Pestaña 2 de la Interfaz (Detección de Señal en los Sensores de Efecto Hall) .....                              | 108 |
| Figura 38. | Pestaña 3 de la Interfaz (Medición de Corriente) .....                                                          | 109 |
| Figura 39. | Pestaña 4 de la Interfaz (Medición de Voltaje) .....                                                            | 110 |
| Figura 40. | Bobina Toroidal en un de las Fases del Motor BLCD .....                                                         | 111 |
| Figura 41. | Voltaje CA de la Bobina en la Fase G .....                                                                      | 112 |
| Figura 42. | Voltaje CA de la Bobina en la Fase B .....                                                                      | 112 |
| Figura 43. | Voltaje CA de la Bobina en la Fase Y .....                                                                      | 113 |
| Figura 44. | Circuito de Acondicionamiento para los voltajes de cada Bobina (Rectificador de Onda Completa con Filtro) ..... | 114 |
| Figura 45. | Voltaje Rectificado de la Bobina en la Fase G .....                                                             | 114 |
| Figura 46. | Voltaje Rectificado de la Bobina en la Fase B .....                                                             | 115 |
| Figura 47. | Voltaje Rectificado de la Bobina en la Fase Y .....                                                             | 115 |
| Figura 48. | Pestaña 1 Online con voltaje presente en fases .....                                                            | 117 |
| Figura 49. | Pestaña 1 Online con voltaje ausente en fases .....                                                             | 118 |
| Figura 50. | Conexión para señales de Sensores de Efecto Hall en Raspberri Pi Pico W .....                                   | 119 |
| Figura 51. | Pestaña 2 Online con Señal Presente de los Sensores de Efecto Hall .....                                        | 120 |
| Figura 52. | Pestaña 2 Online con Señal Ausente de los Sensores de Efecto Hall .....                                         | 120 |

|            |                                                                                                      |     |
|------------|------------------------------------------------------------------------------------------------------|-----|
| Figura 53. | Conexión de Sensores de Corriente con el Micrcontrolador .....                                       | 121 |
| Figura 54. | Medición de Corriente de cada Fase sin aceleración de la Scooter.....                                | 123 |
| Figura 55. | Amperaje de cada Fase aplicando minima aceleración del Vehículo .....                                | 124 |
| Figura 56. | Valores de Corriente de cada fase al 100 % de Aceleración (Muestra 1) .....                          | 125 |
| Figura 57. | Valores de Corriente de cada fase al 100 % de Aceleración (Muestra 2) .....                          | 125 |
| Figura 58. | Valores de Corriente de cada fase al 100 % de Aceleración (Muestra 3) .....                          | 126 |
| Figura 59. | Valores de Corriente de cada fase al sin Aceleración de la Scooter .....                             | 126 |
| Figura 60. | Conexiones realizadas para la medición de voltaje entre fase y fase.....                             | 129 |
| Figura 61. | pestaña 4 de Interfaz en Estado Online .....                                                         | 130 |
| Figura 62. | Comportamiento de Aceleración Gradual vista en las Fases del Motor.....                              | 131 |
| Figura 63. | Valores de Voltaje entre fase y fase al 100% de Aceleración (Muestra 1) ....                         | 132 |
| Figura 64. | Valores de Voltaje entre fase y fase al 100% de Aceleración (Muestra 2) ....                         | 132 |
| Figura 65. | Valores de Voltaje entre fase y fase al 100% de Aceleración (Muestra 3) ....                         | 133 |
| Figura 66. | Valores de voltajes entre fases sin acelrecación posterior a la prueba de medición de voltajes ..... | 134 |

## LISTA DE TABLAS

|          |                                                                         |     |
|----------|-------------------------------------------------------------------------|-----|
| Tabla 1. | Variables de Monitoreo para Motor BLDC y su Finalidad Diagnóstica.....  | 28  |
| Tabla 2. | Secuencia de conmutación de 6 pasos para motor BLDC .....               | 35  |
| Tabla 3. | Efecto de $\alpha$ en el filtro. ....                                   | 62  |
| Tabla 4. | Valores de tensión medida entre diferentes combinaciones de fases ..... | 104 |
| Tabla 5. | Valores de Corriente(A) de cada fase.....                               | 106 |
| Tabla 6. | Mediciones de Voltaje en Bobinas de Detección .....                     | 116 |
| Tabla 7. | Comparación de Mediciones de Corriente por Fase.....                    | 127 |
| Tabla 8. | Comparación de Mediciones de Voltaje Fase-Fase .....                    | 134 |

## RESUMEN

El presente proyecto de tesis desarrolló un sistema de monitoreo remoto para variables críticas del motor BLDC de una scooter eléctrica Keeway Ride E1, con el objetivo de proporcionar una herramienta de diagnóstico en el tiempo que permita el análisis del comportamiento del sistema de propulsión bajo diferentes condiciones de operación. El sistema implementa una arquitectura distribuida basada en múltiples microcontroladores Raspberry Pi Pico W, cada uno especializado en la adquisición de diferentes variables: detección de presencia de voltaje en fases mediante bobinas toroidales, monitoreo de señales de sensores de efecto Hall, medición de corrientes por fase utilizando sensores ACS712, y medición de voltajes fase-fase mediante módulos ZMPT101B. Todos los dispositivos se comunican mediante protocolo UDP WiFi hacia una Raspberry Pi 4 que funciona como servidor central y proporciona una interfaz gráfica desarrollada en PyQt5. La metodología incluyó el diseño de circuitos de acondicionamiento de señal específicos para cada tipo de sensor, implementación de algoritmos de filtrado digital ( mediana, IIR), calibración por software de los sistemas de medición, y desarrollo de una interfaz de usuario profesional con visualización en tiempo real de gráficas y valores numéricos. El sistema fue validado mediante comparación con instrumentos de referencia (multímetro Fluke y amperímetro de gancho), obteniendo precisiones promedio del 1.97 % para corrientes y 0.66 % para voltajes.

Los resultados demuestran que el sistema es capaz de detectar correctamente la presencia/ausencia de voltaje en las fases del motor, monitorear el funcionamiento de los sensores de efecto Hall, medir corrientes por fase con errores menores al 2 %, y medir voltajes fase-fase con errores menores al 1.1 %. La interfaz proporciona visualización simultánea de las cuatro categorías de datos con actualización en el tiempo, incluyendo indicadores de estado online/offline para cada microcontrolador. El principal aporte de esta investigación es la demostración de que es posible desarrollar un sistema de monitoreo para vehículos eléctricos utilizando hardware de bajo costo y código abierto. El sistema desarrollado representa una solución accesible para el diagnóstico y mantenimiento predictivo de vehículos eléctricos de dos ruedas, contribuyendo a la democratización de tecnologías de monitoreo avanzado en el ámbito de la movilidad eléctrica sustentable.

## ABSTRACT

This thesis project developed a remote monitoring system for critical variables of the BLDC motor of a Keeway Ride E1 electric scooter, with the aim of providing A diagnostic tool that enables the analysis of the system's behavior propulsion. The system implements a distributed architecture based on multiple microcontrollers Raspberry Pi Pico W, each specialized in acquiring different variables: detecting Detection of phase voltage presence using toroidal coils, monitoring of signals from Hall sensors, phase current measurement using ACS712 sensors, and measurement Phase-to-phase voltage measurement using ZMPT101B modules. All devices communicate via Wi-Fi UDP protocol to a Raspberry Pi 4 that acts as a central server and It provides a graphical interface developed in PyQt5. The methodology included the design of specific signal conditioning circuits for each type of sensor, implementation of digital filtering algorithms (moving average, median, IIR), software calibration of the measurement systems, and development of a professional user interface with graphical display and numerical values. The system It was validated by comparison with reference instruments (Fluke multimeter and clamp-on ammeter), achieving average accuracies of 1.97 % for currents and 0.66 % for voltages. The results demonstrate that the system is capable of correctly detecting the presence or absence of voltage on the motor phases, voltage presence/absence on the motor phases, monitoring the operation of the Hall sensors, measure phase currents with errors of less than 2 %, and measure phase-to-phase voltages with errors less than 1.1 %. The graphical interface provides simultaneous visualization of the four categories real-time data streams, including online/offline status indicators for each microcontroller. The main contribution of this research is demonstrating that it is possible to develop A professional monitoring system for BLDC motors using low-cost hardware and open-source. The system developed represents an accessible solution for the diagnosis and maintenance predictive maintenance of two-wheeled electric vehicles, which contribute to the democratization of transportation technologies advanced monitoring in the field of electric mobility sustainable.

**Keywords:** ELECTRIC VEHICLE, GRAPHICAL USER INTERFACE, REMOTE MONITORING, MICROCONTROLLER

# INTRODUCCIÓN

En los últimos años se ha apreciado el incremento de la adquisición de vehículos eléctricos con el propósito de buscar movilidad amigable con el ambiente, pues estos vehículos no emiten gases de efecto invernadero al no tener un sistema de combustión que genere la propulsión para la movilidad en cambio se tiene un sistema el cual convierte la energía eléctrica en la fuente principal de su sistema de propulsión.

Dentro de este contexto, hay un considerable porcentaje de relevancia con respecto a los vehículos eléctricos de dos ruedas, esto debido a su eficiencia energética, bajo costo operativo y su adaptabilidad en los entornos urbanos.

El enfoque de este trabajo de tesis se centra sobre el campo de scooters eléctricos, los cuales en su mayoría reside como sistema de propulsión un motor de corriente continua sin escobillas (por sus siglas en inglés: BLCD), ofreciendo una ventaja significativa frente a los motores con escobillas, tales como mayor eficiencia, mantenimiento reducido, vida útil prolongado y una eficiente relación-peso.

En la gama de estos vehículos, por lo general de bajo costo, encontramos una carencia la cual es el monitoreo sobre las variables operativas del motor, esto con el propósito de obtener la información necesaria para realizar un diagnóstico sobre la operatividad del vehículo. Los sistemas existentes solo abarcan por lo general datos básicos de estado, omitiendo información importante como el voltaje entre fases, la corriente resultante en cada fase y estados de los sensores de posicionamiento.

Por ello en este trabajo de tesis se enfoca en implementar un sistema de monitoreo remoto cuyo propósito sea visualizar variables críticas del motor con el objetivo de realizar un diagnóstico lo más preciso posible, también destacando que se hace uso de microcontroladores para el procesamiento de estas variables y el poder enviar los datos empleando protocolos de comunicación para tener un servidor local que permita mostrar los datos en una interfaz diseñada especialmente para este proyecto.

## PLANTEAMIENTO DEL PROBLEMA

Al día de hoy, hemos podido presenciar una transición imparable en las ciudades de la mayor parte del mundo. Desde coches, bicicletas, patines y motocicletas eléctricas han pasado a ser de una curiosidad del futuro a una realidad cotidiana. Con esta revolución en marcha se promete un aire más limpio, un ruido urbano amortiguado y una relación más sostenible en el mundo. A pesar de ello, nos encontramos con un reto tecnológico: ¿Cómo asegurarse que, aparte de ser ecológicos, podamos asegurarnos de que son confiables y seguros a largo plazo los vehículos eléctricos?.

Para que un usuario de un vehículo eléctrico (VE) aumente su confianza sobre su uso, se construye sobre un pilar importante: la salud de su sistema de propulsión. En la mayoría de los VE reside un motor eléctrico de corriente continua sin escobillas (por sus siglas en inglés Brushless Direct Current), un componente eficiente que depende de una conversación bien estructurada entre sus componentes. Esta comunicación se compone de señales eléctricas críticas cuyas son: los voltajes que impulsan su movimiento, la corriente que revela sus esfuerzos y los pulsos de control que dictan su ritmo, generados por los sensores de efecto hall.

Esta comunicación puede verse distorsionada, ya sea por una interferencia, un componente que comienza a fallar o una señal que puede debilitarse; por estos factores el motor comienza a estresarse causando varios sucesos como pérdida de eficiencia, aumento de temperatura o en el peor de los casos, detenerse inesperadamente. El problema lo encontramos cuando nos damos cuenta que muchos de estos VE carecen de un sistema que sea capaz de visualizar las señales que pueden darnos respuesta a lo que ocurre, dejando a usuarios y técnicos en la oscuridad, reaccionando a las fallas cuando ya es demasiado tarde.

Para contribuir a la respuesta a este reto técnico, el presente proyecto se centrará en un caso de estudio concreto y muy representativo: la scooter eléctrica. Este VE se convertirá en nuestro sujeto de prueba para desarrollar un sistema de monitoreo remoto. Se busca crear una herramienta que no solo capture las señales vitales del motor BLCD, sino que también se pueda distinguir la información valiosa del ruido que pueda estar presente, pues bien se sabe que en el ambiente estamos llenos de interferencias eléctricas que puede provocar la mala

visualización del verdadero estado del motor. Por lo tanto, una parte fundamental del trabajo se centra en diseñar y aplicar filtros digitales que busquen limpiar nuestra señales, permitiendo ver con claridad el comportamiento del motor.

Implementando este sistema de monitoreo sobre una scooter, nos puede dar pauta a validar una metodología que puede extenderse a otros vehículos eléctricos ligeros. Buscamos pasar de un modelo de mantenimiento reactivo (arreglar cuando se rompa) a uno preventivo (visualizar y anticiparse a la falla), proporcionando datos necesarios que evalúen el desempeño, detectar anomalías, y de esta manera que cada usuario que posee un VE tenga la confianza de operarlo sin problema.

## JUSTIFICACIÓN

La implementación de sistemas de monitoreo remoto para vehículos eléctricos responde a necesidades técnicas críticas no satisfechas del mercado actual. La ausencia de una herramienta de visualización accesible para los sistemas de tracción eléctrica limita las capacidades de mantenimiento preventivo y dificulta la identificación temprana de fallos en componentes clave como es el caso del motor BLCD y sus sistemas de control.

Este proyecto se justifica por la necesidad de desarrollar una metodología para la adquisición y procesamiento de señales críticas en vehículos eléctricos. La captura de parámetros como voltaje de fases, corriente de consumo y señales de sensores Hall, junto con la implementación de filtros digitales para eliminar el ruido, permite obtener datos confiables para el análisis del estado del motor.

Diseñar un sistema de monitoreo de este tipo ofrece beneficios como puede ser el incremento en la seguridad operativa al detectar fallas antes de que comprometan el funcionamiento del vehículo, mejorar el mantenimiento preventivo al permitir identificar anomalías sin desmontar directamente el motor, generación de datos cuantitativos que pueden ser útiles para el estudio de eficiencia energética, desgaste y la calidad del control, por último, ser potencial aplicación en futuras mejoras de diseño.

## HIPÓTESIS

Es posible implementar un sistema de monitoreo remoto en un vehículo eléctrico mediante el uso de protocolos de comunicación a distancia, utilizando microcontroladores para el procesamiento y acondicionamiento de señales, así como el uso de filtros por software para la reducción de perturbaciones y ruido.

# OBJETIVOS

## OBJETIVO GENERAL

Diseñar, desarrollar e implementar un sistema de monitoreo remoto, basado en una red de microcontroladores Raspberry Pi Pico W y una unidad central Raspberry Pi 4, que permita la adquisición, filtrado digital y visualización de variables eléctricas esenciales del motor BLDC en una scooter eléctrica, con el fin de verificar su correcto funcionamiento.

## OBJETIVOS ESPECÍFICOS

1. Seleccionar e instrumentar las variables críticas del motor BLDC.
2. Diseñar un sistema distribuido de adquisición y transmisión de datos.
3. Desarrollar e implementar algoritmos de filtrado digital de señales.
4. Desarrollar una interfaz gráfica de usuario (GUI) sobre Raspberry Pi 4.

# LIMITACIONES Y ALCANCES

## Alcance del Proyecto

El proyecto abarca el diseño, implementación y validación completa de un sistema de monitoreo remoto para un motor BLDC específico (el instalado en la scooter eléctrica Keeway Ride E1), incluyendo:

- Diseño de hardware para cuatro subsistemas de sensado.
- Desarrollo de firmware para cuatro microcontroladores Raspberry Pi Pico W.
- Implementación de software para interfaz gráfica en Raspberry Pi 4.
- Protocolo de comunicación inalámbrica WiFi UDP.
- Validación experimental con instrumentos de referencia.
- Documentación completa del sistema desarrollado.

## Limitaciones

- El sistema se desarrolla específicamente para el motor BLDC de la scooter Keeway Ride E1, aunque la arquitectura es adaptable a otros motores similares.
- No incluye el desarrollo de algoritmos de control para el motor, solo sistemas de monitoreo y diagnóstico.
- La validación se realiza en condiciones estáticas (scooter en caballete).
- El sistema no incluye capacidades de actuación sobre el motor, solo monitoreo y visualización.

# CAPÍTULO 1: ESTADO DEL ARTE

El monitoreo remoto se ha consolidado como un recurso clave para supervisar procesos y dispositivos en el tiempo, particularmente cuando se requieren decisiones rápidas ante variaciones de variables críticas o condiciones de falla. El desarrollo de la Industria 4.0 ha impulsado arquitecturas conectadas a internet (por sus siglas en inglés: IoT), facilitando la recolección de datos en la nube y habilitando interfaces gráficas más accesibles para el usuario final. Sin embargo, esta evolución también incrementa la superficie de ataque y las exigencias de ciberseguridad, especialmente en sistemas tipo SCADA y de control industrial. En esta sección se sintetizan trabajos relevantes organizados por ejes temáticos, con el propósito de delimitar tendencias, enfoques tecnológicos y áreas de oportunidad que sustentan el desarrollo de un sistema de monitoreo remoto aplicado a una scooter eléctrica, orientado al motor BLDC y variables eléctricas (voltaje, corriente) y de sensado (señales de sensores de efecto Hall).

## 1.1 Sistemas IoT y monitoreo remoto: de vivienda a industria

En aplicaciones domésticas, López, 2020 propone un sistema domótico soportado por una interfaz de visualización personalizada para controlar actividades cotidianas mediante IoT. El trabajo destaca el uso de plataformas programables en Python, donde la comunicación se centraliza en una Raspberry Pi utilizando PySerial para enlazar la interfaz gráfica y operar dispositivos IoT de forma remota. Un aporte relevante es la adaptabilidad del sistema a distintos entornos residenciales, atribuida a la simplicidad de la interfaz diseñada.

Bajo un enfoque metodológico y escalable, Molina et al., 2020 desarrolla un sistema de monitoreo remoto que integra estaciones de monitoreo ambiental, sensores y actuadores. Se emplean plataformas embebidas (Raspberry Pi y Arduino Uno) y una interfaz web basada en HTML, PHP y JavaScript, enfatizando el análisis de requerimientos y la posibilidad de extender el número de estaciones. Este tipo de arquitectura es representativo cuando se busca crecimiento modular del sistema, incorporación progresiva de sensores y visualización multiplataforma.

En el dominio industrial, Tuitise et al., 2020 plantea un sistema SCADA para un entorno virtual

de flujo incorporando monitoreo remoto en tiempo real sobre arquitectura IoT, con envío de datos a nube industrial. Se utiliza PLC SIM, Ignition y comunicación Ethernet, apoyándose en TIA Portal para integrar la automatización en un programa único. El resultado reporta mejoras en control y flujo de producción derivadas de la supervisión remota de variables de interés.

### **Implicaciones para monitoreo en vehículos eléctricos**

Los enfoques anteriores convergen en dos elementos transversales aplicables al caso de una scooter eléctrica: (I) adquisición de datos con nodos embebidos y (II) visualización mediante una interfaz (GUI) que simplifique la operación del usuario. En el caso vehicular, estas premisas se traducen en la necesidad de adquirir señales eléctricas (corriente/voltaje), variables del motor y estados derivados de sensores (efecto Hall), para presentarlos de forma comprensible y accionable.

#### **1.1.1 Interfaces gráficas y usabilidad: GUI y TUI**

El diseño de interfaces es un factor determinante para la adopción de sistemas de monitoreo y control, ya que impacta directamente la interacción hombre/máquina y la calidad de la toma de decisiones. En Pedro, 2020 se presenta un análisis de interfaces de usuario tangibles (TUI) con el objetivo de representar físicamente información digital mediante superficies interactivas. Se evalúa si un Sistema Tangible basado en papel (PaperTUI) puede sustituir el uso tradicional del papel en prácticas de trabajo. El sistema utiliza Python y OpenCV para detección e identificación de objetos, generando un modelo digital exportable a XML. Se reporta que los usuarios consideran PaperTUI más rápido para prototipos de bajo nivel, lo cual evidencia el valor de la interacción intuitiva en tareas de diseño y prueba.

En una línea orientada a software interactivo con objetivos analíticos, José, 2020 desarrolla una GUI para estudiar la teoría moderna de portafolios (MPT) en Matlab (lenguaje 'm') usando App Designer. Aunque el dominio es financiero, el aporte metodológico consiste en traducir modelos de optimización a un entorno interactivo que permita experimentar, comparar y evaluar resultados desde una interfaz.

De manera más cercana a instrumentación y comunicación serial, Ortiz, 2022 implementa una interfaz gráfica para programar cargas electrónicas BK Precision serie 8500 mediante comunicación serial. Se emplea Python y PyQt Designer para integrar el entorno gráfico con funciones y algoritmos de control, permitiendo ingresar parámetros eléctricos (voltaje/corriente/potencia) según el modo de operación. Este enfoque es directamente transferible a sistemas de monitoreo vehicular donde la comunicación serial, el sensado eléctrico y la claridad de parámetros presentados son críticos.

En términos de tendencias generales, Velandia et al., 2022 reporta una revisión sobre el uso de GUI en telecomunicaciones optoelectrónicas, destacando la adopción de herramientas de licencia abierta para cálculos avanzados y representación gráfica. Este hallazgo es relevante al justificar el uso de ecosistemas abiertos (Python, bibliotecas científicas, frameworks de GUI) para acelerar el desarrollo y facilitar reproducibilidad.

### **1.1.2 Generación y adaptación computacional de interfaces**

Más allá del diseño manual, Oulasvirta et al., 2020 analiza la optimización combinatoria como herramienta para adaptar y generar GUI computacionalmente. Se utiliza un modelo matemático basado en programación entera (Integer Programming) para sustentar aplicaciones de diseño algorítmico de interfaces. Este enfoque abre la posibilidad de automatizar decisiones de layout (ubicación de widgets, restricciones de tamaño, priorización visual), lo cual puede ser útil en sistemas donde el conjunto de variables crece y se requiere reconfiguración dinámica de paneles de monitoreo.

### **1.1.3 SCADA/ICS y ciberseguridad: riesgos y estrategias**

La conectividad remota y la integración con internet, aunque habilitadoras de nuevos servicios, incrementan vulnerabilidades en arquitecturas SCADA e ICS. Yadav y Paul, 2020 reseña la evolución de SCADA hacia sistemas conectados a internet y la consecuente exposición a ciberataques, mediante una revisión en bases como IEEE Xplore, ACM, IET y Scopus, concluyendo la necesidad urgente de medidas de seguridad.

En el mismo eje, Upadhyay y Sampalli, 2020 estudia vulnerabilidades en SCADA con redes de comunicación extensas, abordando cuatro componentes: productos, configuración, red y protocolos de comunicación. Además, se consideran incidentes reales para comprender técnicas de protección y refuerzo de seguridad, enfatizando la estandarización de controles preventivos.

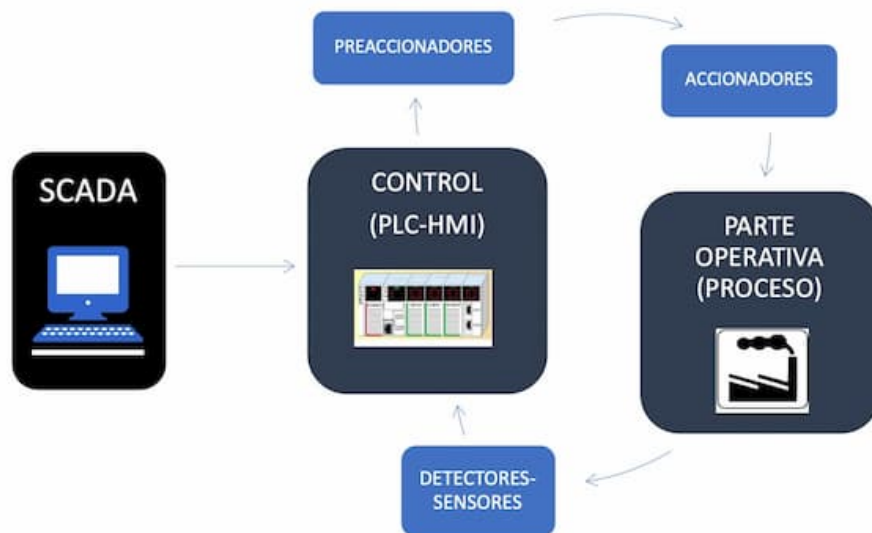
A nivel de arquitectura distribuida, Menzel et al., 2021 Verena et al., 2021 analiza sistemas de detección de intrusos (IDS) aplicados centralizadamente en servidor SCADA y propone ampliar el alcance mediante supervisión basada en procesos y división de la red en subredes. Se muestra que añadir IDS ayuda a identificar ataques más complejos, destacando la relevancia de modelos que consideren interacción entre nodos vecinos.

En términos de propuestas algorítmicas, Justindhas y Jeyanthi, 2022 implementa un clasificador NK-RNN para detección de ataques, incorporando etapas de preprocesamiento, selección de características, detección, cifrado y encriptado. Por su parte, Alimi et al., 2021 revisa estudios (2018-2021) sobre técnicas de aprendizaje supervisado para intrusión en SCADA, sintetizando recomendaciones para investigaciones futuras. Asimismo, Alanazi et al., 2023 realiza un análisis longitudinal de vulnerabilidades y ataques en SCADA, recomendando mejoras en evaluación de riesgos y el desarrollo de conjuntos de datos específicos.

En el contexto de gestión de riesgo y protección de datos en sistemas ciber-físicos, Buldakova, 2020 propone una metodología basada en IDEF0 para etapas de gestión del riesgo durante supervisión remota, concluyendo la necesidad de nuevos métodos matemáticos para asegurar confidencialidad. Complementariamente, Rakas et al., 2020 revisa 26 artículos sobre IDS y plantea áreas de oportunidad para fortalecer seguridad en SCADA de tercera generación, resaltando el crecimiento de métodos basados en aprendizaje automático y herramientas abiertas.

Finalmente, Osman et al., 2022 presenta un SCADA remoto (RSS) para aplicaciones industriales (supervisión, censo y análisis de fallas en máquinas de alto poder), con escalabilidad en nodos y E/S, empleando ESP8266 y acceso vía red GSM. El trabajo reporta mayor inteligencia, rapidez y confiabilidad frente a SCADA convencional, lo cual refuerza la tendencia hacia

arquitecturas remotas, distribuidas y accesibles desde dispositivos móviles. En la figura 1 se muestra la arquitectura que conforma a un sistema SCADA.



**Figura 1: Arquitectura general de un sistema SCADA**

#### **1.1.4 SCADA con interacción avanzada: realidad aumentada**

Como extensión de la interacción hombre/máquina, Cando, 2023 propone un SCADA en realidad aumentada (RA) para mejorar la interacción con sistemas de monitoreo mediante modelado 3D interactivo. Se emplean Unity 3D y Vuforia, con comunicación OPC-UA para interoperabilidad. Este enfoque resulta relevante como tendencia emergente: transformar paneles 2D en representaciones 3D contextualizadas del sistema. Aunque su complejidad puede exceder el alcance de una scooter eléctrica, establece una dirección de evolución para visualización avanzada de activos y variables.

#### **1.1.5 Vehículos eléctricos: autonomía, energía, carga y necesidades de monitoreo**

En vehículos eléctricos (VE), la autonomía y la gestión energética son factores decisivos para adopción y operación. Sanguesa et al., 2021 revisa tecnologías clave (baterías, control de potencia, modos de carga y conectores), así como el estado del mercado y perspectivas, concluyendo que mejoras en capacidad de baterías y estandarización de conectores pueden favorecer el despliegue global.

La necesidad de medición bajo condiciones reales se observa en Javier y Alvaro, 2022, donde se evalúa autonomía de una motocicleta eléctrica (Super Soco TC Max) en una ciudad de altura mediante una prueba Coast Down y monitoreo con Data Logger (GPS), reportando que la autonomía real es menor que la teórica por factores variables del recorrido y operación.

Desde el punto de vista de conversión de potencia, Romero et al., 2020 diseña y simula etapas de un cargador de batería para motocicleta eléctrica usando MATLAB/Simulink, cubriendo AC/AC, AC/DC con corrección de factor de potencia y DC/DC reductora, subrayando la importancia del control y selección de topología.

En cuanto a estrategias para extender autonomía, Picho y Isaac, 2023 investiga un sistema de carga regenerativa basado en KERS, donde el giro de la rueda delantera genera electricidad. Se modela el generador y se analiza el comportamiento magnético con FEMM, concluyendo una autonomía extra estimada entre 15 % y 25 %. Alternativamente, Daniel y José, 2020 plantea un sistema de captación de aire y generación de energía (turbina Savonius) con aumento teórico de 3.7 % de autonomía en un vehículo de estudio.

A nivel de operación y logística, Küçükoğlu et al., 2021 revisa el problema de enrutamiento para VE (EVRP) y sus variaciones, evidenciando que muchas investigaciones minimizan métricas basadas en distancia bajo restricciones energéticas. En infraestructura, Ahmad et al., 2022 analiza la localización óptima de estaciones de carga (EVCS) desde enfoques de red, propietario y usuario, enfatizando impacto en estabilidad de voltaje, confiabilidad y costos. Por otra parte, Ravi y Aziz, 2022 revisa la integración vehículo-red (VGI) y servicios V2G, identificando beneficios potenciales (nivelación de carga, almacenamiento) y dificultades para cuantificar viabilidad económica por condiciones técnicas, de mercado y normativas. De forma complementaria, Ehsani et al., 2021 compara arquitecturas de VE y HEV, resaltando retos de manufactura y complejidad del tren motriz y control respecto a vehículos convencionales.

### **Implicaciones para scooter eléctrica con motor BLDC**

La literatura evidencia que la brecha entre autonomía teórica y real, así como la complejidad del tren motriz y la electrónica de potencia, exigen sistemas de monitoreo que permitan:

registrar condiciones reales de operación, detectar eventos anómalos, y correlacionar variables eléctricas con desempeño del motor. En un motor BLDC, esto típicamente implica instrumentar corriente (por fase o total), voltaje del bus, estados del controlador, y señales de los sensores de efecto Hall para inferir conmutación, velocidad estimada o condiciones de sincronía. En la tabla 1 se muestra las variables candidatas por monitorear.

**Tabla 1: Variables de Monitoreo para Motor BLDC y su Finalidad Diagnóstica**

| Variable           | Finalidad Diagnóstica                                                  |
|--------------------|------------------------------------------------------------------------|
| Voltaje de fase    | Detectar desbalances entre fases, identificar problemas en el inversor |
| Corriente por fase | Identificar sobrecargas, detectar desbalances, evaluar eficiencia      |
| Potencia eléctrica | Calcular eficiencia energética, optimizar consumo de batería           |
| Señales Hall       | Verificar funcionamiento de sensores                                   |

### 1.1.6 Brecha identificada y orientación de la tesis

De la revisión se identifican tres conclusiones operativas para el problema de investigación:

1. **Arquitectura de monitoreo:** Los sistemas IoT y SCADA remotos demuestran la viabilidad de adquirir variables en campo y visualizarlas en tiempo real (López, 2020, Tuitise et al., 2020, Osman et al., 2022), con opciones de escalabilidad.
2. **Interfaz y operabilidad:** Las GUI orientadas a control/instrumentación (Arteaga, 2022) y los estudios de interacción (Pedro, 2020) resaltan que la utilidad del monitoreo depende de una visualización clara, jerárquica y fácil de operar, especialmente bajo escenarios de diagnóstico.
3. **Seguridad:** La conectividad remota en SCADA/ICS conlleva riesgos que deben considerarse desde el diseño, siendo pertinente al menos justificar medidas básicas de mitigación (segmentación, autenticación, registros, etc.) si existe acceso remoto.

Con base en lo anterior, la tesis se orienta a implementar un sistema de monitoreo remoto para una scooter eléctrica que permita supervisar el motor BLDC y variables eléctricas (voltaje,

corriente) junto con señales de sensado (sensores de efecto Hall). El objetivo es proporcionar una plataforma que facilite: (I) adquisición confiable de señales, (II) visualización en una interfaz accesible y (III) potenciales funciones de diagnóstico y trazabilidad de datos, incorporando consideraciones de robustez y seguridad acordes al nivel de exposición del sistema.

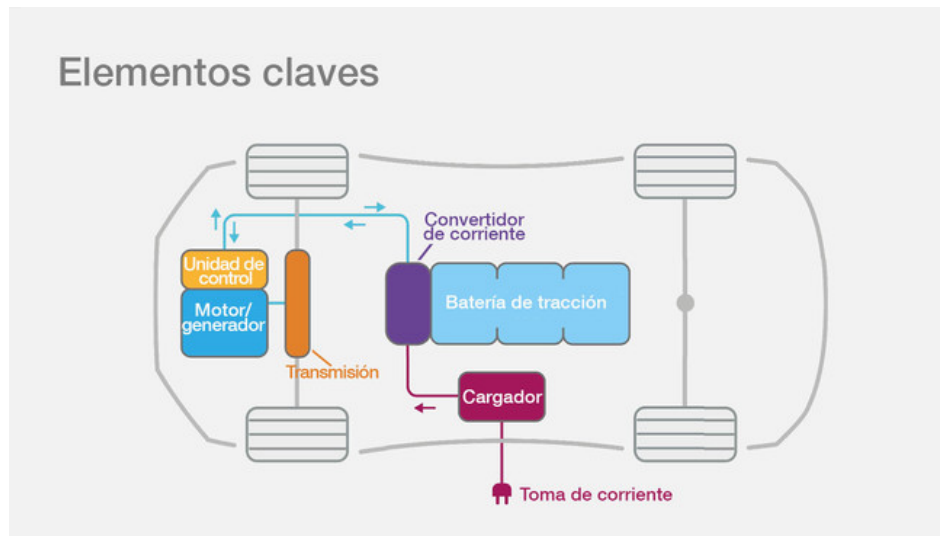
## CAPÍTULO 2: MARCO TEÓRICO

### 2.1 Vehículo Eléctrico

#### 2.1.1 Funcionamiento General de un Vehículo Eletéctrico

Los vehículos eléctricos (VE) convierten la energía eléctrica almacenada en baterías en movimiento mecánico, sin combustión interna, por medio de un sistema de propulsión eléctrica que incluye la batería, el convertidor de potencia, el motor eléctrico y el controlador. La energía de la batería (corriente continua, CC) se transforma en corriente alterna (CA) para alimentar el motor (o directamente en motores CC), y el controlador ajusta la potencia que se envía al motor según lo requiera el conductor para acelerar o frenar. (Madaram et al., 2024) En la figura 2 se muestra un esquema general de un vehiculo eléctrico.

#### 2.1.2 Estructura General de un Vehículo Eléctrico



**Figura 2: Estructura general de un Vehículo Eléctrico**

La arquitectura básica de un VE se compone de los siguientes subsistemas interconectados:

**Sistema de almacenamiento de energía (batería):** Es la principal fuente de energía. Normalmente emplea celdas de ion-litio, las cuales están estructuradas en módulos y paquetes. (Hussein et al., 2024)

**Motor eléctrico:** Transforma la energía eléctrica en par de tipo mecánico. Las clases más usuales Son los motores de inducción (AC), de imanes permanentes (PM) y sin escobillas. DC (BLDC). El costo, la eficiencia y los requisitos de par/velocidad determinan la selección.(Hussein et al., 2024)

**Convertidor de potencia (convertidor CC-CC/inversor):** El inversor transforma la CC de la batería para el motor en corriente alterna. Un convertidor CC-CC modifica el voltaje de la batería a los sistemas secundarios de voltaje bajo. (Mohammed et al., 2025)

**Controlador electrónico de automóviles (VECU):** Es el dispositivo que unifica la administración del BMS, el tren de potencia y la recuperación energética en la detención (detención regenerativa) y en la comunicación con los sistemas abordo.

**Sistema de carga:** Facilita la recarga de la batería a través de Estaciones de carga rápida de corriente directa (DC) o cargadores a bordo (AC). (Madaram et al., 2024)

**Sistema de transmisión:** Suele ser de una sola velocidad (reductor) por lo siguiente: el extenso espectro de par del motor eléctrico, lo cual facilita la mecánica en comparación con las transmisiones de múltiples cambios.(Hussein et al., 2024)

### 2.1.3 Especificaciones Técnicas Clave

Las especificaciones que establecen las habilidades y el rendimiento de un VE abarcan:

**Capacidad de la batería:** Energía guardada, medida en kilowatts-hora (kWh). Defina la autonomía. Para automóviles ligeros contemporáneos, la capacidad de las baterías suele oscilar entre 40 y 100 kWh.

**Potencia del motor:** La potencia máxima de salida del motor, que se mide en kilowatts (kW) o caballos de fuerza (HP).

**Máximo par:** Par motor instantáneo disponible, expresado en Newton-metro (Nm). Los motores eléctricos son capaces de generar par máximo desde las revoluciones bajas, lo que les permite una aceleración rápida.

**Voltaje del sistema:** Voltaje nominal del bus de alta tensión. Los sistemas comunes son de

400 V, y las plataformas más nuevas utilizan 800 V para permitir cargas más rápidas y reducir pérdidas.

**Autonomía:** Distancia aproximada que el vehículo tiene la capacidad de cubrir con una carga integral. Comúnmente oscila entre los 300 y los 600 km para modelos contemporáneos.

**Duración de la carga:** Está condicionado por la capacidad de la batería y la potencia del cargador. Con cargadores de DC rápidos (50-350 kW), es posible llegar entre el 10 % y el 80 % en De 20 a 40 minutos.

Teniendo en cuenta las características generales que componen a los vehículos eléctricos, podemos analizar en la gama de scooters eléctricas su composición y características. Para ello se establece las especificaciones de nuestro caso de estudio de este proyecto, es debido mencionar que se tuvo que hacer una modificación en cuanto a la batería, pero en el momento en que se realiza el monitoreo de las variables del presente proyecto, la scooter se encuentra sin una batería disponible. Sin embargo, se menciona a continuación la modificación que se realizó cuando el vehículo contaba con una batería disponible. A continuación en la figura 3 se muestra el modelo de la scooter.

## 2.2 Caso de Estudio: Scooter Eléctrico Keeway Ride E1 modelo 2015



**Figura 3: Caso de Estudio Keeway Ride E1**

### 2.2.1 Configuración Técnica Original del Vehículo

#### ■ Prestaciones de Conducción:

- Ofrece **dos modos de conducción**: un modo **Normal** con velocidad máxima de kilómetros por hora , y un modo **Eco** limitado a 38 kilómetros por hora.
- Su autonomía operativa se estimaba entre 30-50 kilómetros por ciclo de carga completa.
- La capacidad máxima de carga útil era de 150 kilogramos.

#### ■ Sistema de Propulsión y Almacenamiento de Energía:

- **Motor**: Un motor eléctrico de tipo **Brushless (BLDC)** con una potencia nominal de 1500 watts.
- **Batería**: El sistema de almacenamiento original consistía en un conjunto de cinco baterías de plomo-ácido tipo gel. Cada una con tensión de 12 volts y una capacidad de 20 amperios/hora, conectadas en serie.
- **Rendimiento de la Batería**: Esta configuración ofrecía una vida útil aproximada de 300 ciclos de carga y requería entre 8-10 horas para una recarga completa.

#### ■ Otros Componentes Mecánicos:

- **Sistema de Frenos**: Implementaba un freno delantero de **disco ventilado** y un freno trasero de tipo **tambor**.
- **Eficiencia Económica**: Su costo operativo energético era inferior a \$20 MXN por cada 100 kilómetros recorridos.

### 2.2.2 Modificaciones Implementadas

- Se sustituyó el banco original de baterías de plomo-gel por una **batería de iones de litio (Li-ion)** de tecnología moderna.
- La nueva batería opera a una tensión de 60 volts y mantiene una capacidad de 20 amperios/hora. Sus especificaciones clave incluyen:

- Una vida útil extendida de aproximadamente **1000 ciclos de carga**.
  - Un tiempo de recarga completa drásticamente reducido a 4 horas.
- Este nuevo paquete de baterías alimentaba el mismo motor BLDC de 1500 watts, el cual está diseñado para un voltaje nominal de operación de 60 volts y admite un voltaje máximo de carga de 70 volts. La gestión del tren de potencia se realiza mediante un controlador electrónico de velocidad especificado para 60 volts y una corriente máxima de 30 amperios.

### 2.2.3 Subsistema de Control: El Controlador PWM de 60V/30A

El controlador es el cerebro que gobierna al motor. En este caso, un controlador de 60V y 30A cumple esta función vital.

**Función Principal:** Actúa como un intermediario inteligente entre la batería y el motor. Recibe la señal del acelerador y la traduce en una señal de control para el motor.

**Especificaciones Clave:** El voltaje nominal (60V) debe coincidir con el del motor y la batería. La corriente máxima (30A) define el límite de potencia eléctrica que puede manejar: Potencia Máxima Eléctrica = Voltaje x Corriente =  $60V \times 30A = 1800W$ . Esto da un margen de seguridad sobre los 1500W nominales del motor, protegiendo al sistema de sobrecargas.

**Potencia y Velocidad:** La potencia del motor (1500W) y la relación de transmisión determinan la velocidad máxima ( 45 km/h). Esto la sitúa como un vehículo ideal para entornos urbanos.

El controlador es un sistema electrónico de potencia complejo. Su función principal es convertir la corriente continua (DC) de la batería en una secuencia de pulsos trifásica que active las bobinas del motor.

**Puente inversor:** El núcleo del controlador es un puente inversor trifásico, compuesto por seis transistores de potencia (usualmente MOSFETs o IGBTs). Estos actúan como interruptores de alta velocidad que dirigen la corriente hacia las bobinas.

**Modulación por Ancho de Pulso (PWM):** Para controlar la velocidad y el par del motor, el controlador no varía el voltaje de forma continua, sino que utiliza PWM. La señal de control

(proveniente del acelerador) determina el ciclo de trabajo (duty cycle) de los pulsos aplicados a los transistores. Un ciclo de trabajo mayor significa que los transistores están cerrados (conduciendo) durante más tiempo en cada ciclo, entregando más potencia promedio al motor y, por tanto, generando más par y velocidad.

**Secuencia de conmutación:** Para mantener el giro, el controlador debe activar los seis transistores en una secuencia de seis pasos precisa, sincronizada con la posición del rotor. Cada paso energiza dos de las tres fases del motor, creando un campo magnético que avanza en pasos de 60 grados eléctricos. En la tabla 2 se desglosa los 6 pasos descritos previamente.

**Tabla 2: Secuencia de conmutación de 6 pasos para motor BLDC**

| Paso | Fases Activas | TA+ | TB+ | TC+ | TA- | TB- | TC- |
|------|---------------|-----|-----|-----|-----|-----|-----|
| 1    | A → B         | ON  | OFF | OFF | OFF | ON  | OFF |
| 2    | A → C         | ON  | OFF | OFF | OFF | OFF | ON  |
| 3    | B → C         | OFF | ON  | OFF | OFF | OFF | ON  |
| 4    | B → A         | OFF | ON  | OFF | ON  | OFF | OFF |
| 5    | C → A         | OFF | OFF | ON  | ON  | OFF | OFF |
| 6    | C → B         | OFF | OFF | ON  | OFF | ON  | OFF |

### 2.3 Funcionamiento del Motor BLDC en Scooters eléctricos

El motor Brushless DC (BLDC) es el componente fundamental de la propulsión del scooter eléctrico Keeway Ride E1. y en la mayoría de los vehículos eléctricos de su gama. Su diseño, que prescinde de las escobillas mecánicas, es fundamental para alcanzar la elevada eficiencia, el mantenimiento bajo y el par en tiempo real que distinguen a estos vehículos. En la figura 4 se tiene un ejemplo de motor BLCD.



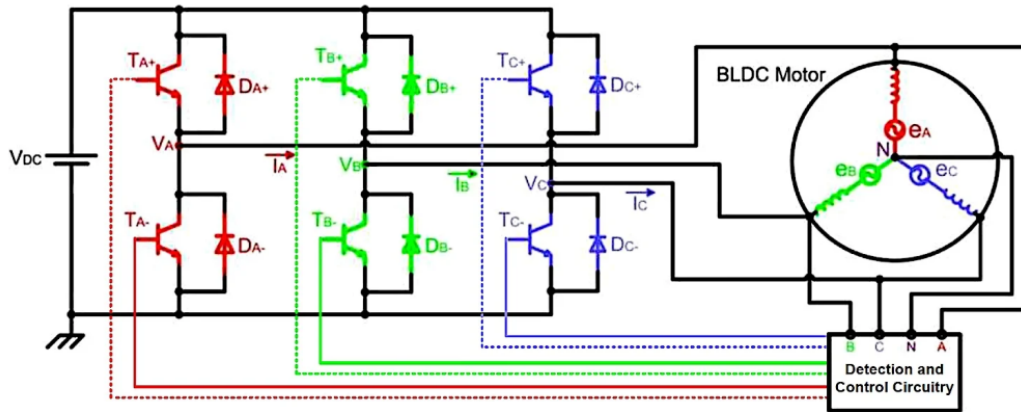
**Figura 4: Ejemplo de Motor BLCD**

### **2.3.1 Principio de Operación**

Un motor BLDC invierte la configuración típica de los motores CC con escobillas. En un motor convencional, el rotor (parte giratoria) tiene bobinas electromagnéticas y el estator (parte fija) tiene imanes permanentes. Las escobillas físicas y el conmutador (colector de delgas) en el eje del rotor cambian la polaridad de la corriente en las bobinas del rotor, creando un campo magnético giratorio que lo hace girar.

**Estator:** Bobinas de cobre enrolladas en forma trifásica.

**Conmutación:** Sin contactos. Un controlador electrónico de velocidad detecta la posición angular del rotor (usando sensores de efecto Hall o métodos sensorless) y energiza pares de bobinas en el estator en el momento preciso. El campo magnético rotatorio en el estator mueve al rotor de imanes permanentes(Mohanraj et al., 2022). En la figura 5 se muestra el diagrama general de conexión de un motor BLCD.



**Figura 5: Esquema General del circuito para Motor BLCD**

### 2.3.2 Detección de Posición del Rotor: Sensored vs. Sensorless

Para realizar la conmutación en el momento exacto, el controlador debe conocer en todo momento la posición del rotor. Existen dos enfoques principales:

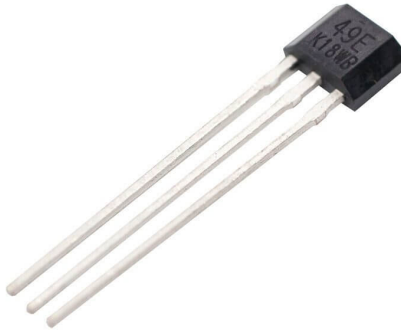
**Control con sensores (Sensored):** Utiliza sensores de efecto Hall integrados en el estator. Estos detectan el paso de los imanes del rotor y envían señales digitales al controlador, indicando la posición exacta. Este método es muy robusto y eficiente a bajas velocidades, y es común en scooters eléctricos.

**Control sin sensores (Sensorless):** Es un método más avanzado que estima la posición del rotor midiendo la fuerza electromotriz (FEM) inducida en la bobina que no está siendo energizada en un momento dado. Esta FEM, llamada Back-EMF, es proporcional a la velocidad del rotor. Aunque es más complejo de implementar, elimina costos y puntos de falla, siendo ideal para aplicaciones que requieren alta confiabilidad.

## 2.4 Funcionamiento de los Sensores de Efecto Hall en el Sistema de Propulsión de una Scooter Eléctrica

Los sensores de efecto Hall son componentes fundamentales para el funcionamiento preciso y eficiente del motor BLDC en la scooter eléctrica. Su función principal es actuar como los ojos del controlador, detectando al momento la posición del rotor para permitir una conmutación

electrónica precisa, reemplazando las escobillas de los motores tradicionales. A continuación se muestra en la figura 6 el sensor de efecto Hall.



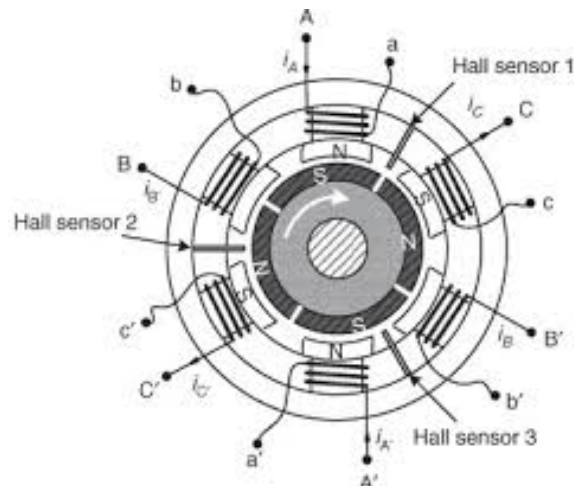
**Figura 6: Ejemplo ilustrativo del sensor de efecto Hall**

#### **2.4.1 Principio Físico Fundamental: El Efecto Hall**

El efecto Hall, descubierto por Edwin Hall en 1879, es el principio físico que permite el funcionamiento de estos sensores. Describe que cuando un conductor o semiconductor por el que circula una corriente eléctrica se sitúa dentro de un campo magnético perpendicular a esta corriente, se genera un voltaje medible (voltaje Hall) a través del material. Este voltaje es proporcional a la intensidad del campo magnético aplicado. En la práctica, este fenómeno se aprovecha en un sensor de estado sólido que convierte directamente la presencia e intensidad de un campo magnético en una señal eléctrica utilizable. (Atorino et al., 2009)

#### **2.4.2 Implementación en el Motor BLDC: Estructura y Colocación**

En un motor BLDC trifásico, se emplean típicamente tres sensores de efecto Hall, distribuidos estratégicamente alrededor del estator como se muestra en la figura 7.



**Figura 7: Ejemplo Ilustrativo sobre la posición de los sensores de efecto Hall en el motor BLCD**

**Colocación Angular:** Están colocados a 120 grados eléctricos de distancia entre sí. Esta disposición asegura una cobertura completa del ciclo de rotación.

**Posición Física:** Se montan en el núcleo del estator, muy cerca del entrehierro por donde giran los imanes permanentes del rotor, pero sin contacto físico alguno.

**Función:** A medida que el rotor gira, los imanes (con sus polos Norte y Sur) pasan frente a cada sensor. El campo magnético cambiante hace que cada sensor genere una señal digital, que puede ser ALTA (lógica 1) cuando detecta un polo Norte y BAJA (lógica 0) cuando detecta un polo Sur, o viceversa, dependiendo de su configuración.

Los sensores modernos para motores suelen ser circuitos integrados (Hall Effect ICs) que incluyen el elemento Hall, amplificadores y circuitos de salida digital, optimizados para ofrecer respuestas rápidas y precisas con variaciones mínimas ante cambios de temperatura, lo que es crucial para la eficiencia del motor.

### 2.4.3 Comunicación con el Controlador y el Proceso de Conmutación

Este es el núcleo del sistema. Los tres sensores tienen cada uno tres cables: alimentación ( $V_{cc}$ , típicamente 5V), tierra (GND) y salida de señal. Estos seis estados únicos forman un código

que le indica al controlador la posición exacta del rotor.

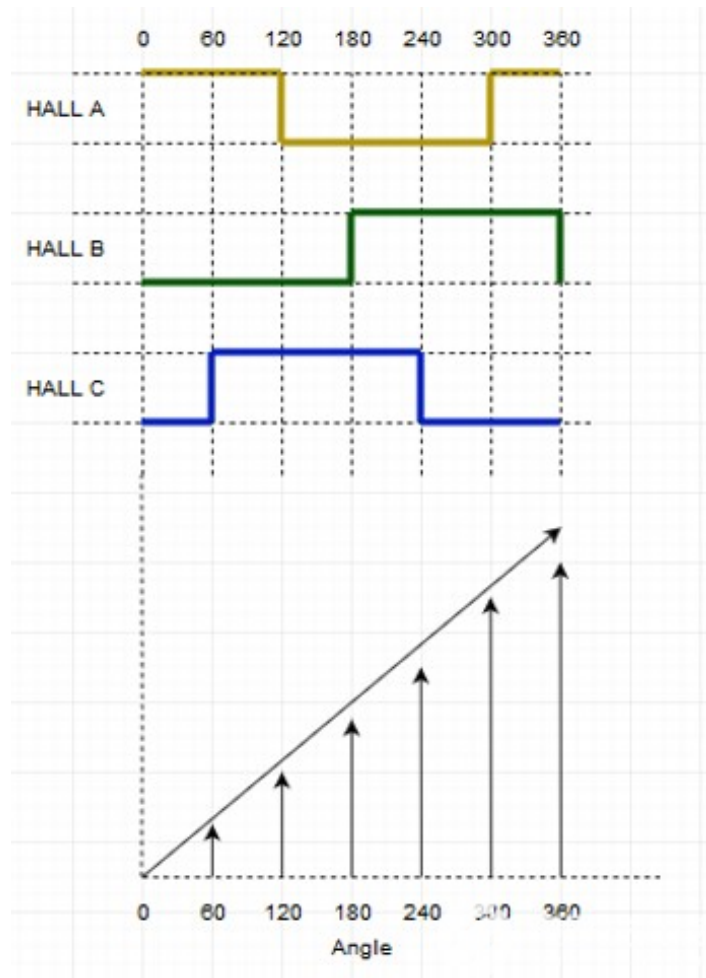
**Lectura de la Posición:** El controlador lee constantemente la combinación de las tres señales digitales (por ejemplo, 101, 100, 110, etc.).

**Determinación de la Secuencia:** Usando una tabla de verdad interna, el controlador traduce este código a la combinación correcta de transistores en el puente inversor que deben activarse.

**Energización de las Bobinas:** El controlador aplica entonces la tensión de la batería (60V) a dos de las tres bobinas del estator en un instante preciso. Esto genera un campo electromagnético que atrae y repele los imanes del rotor, produciendo par motor.

### **Conmutación en 6 Pasos:**

A medida que el rotor gira, los sensores de efecto Hall envían un nuevo código. El controlador detecta este cambio y conmuta inmediatamente la energización a un par diferente de bobinas. Este proceso se repite 6 veces por cada ciclo eléctrico completo (360 grados eléctricos), en lo que se conoce como conmutación trapezoidal de 6 pasos. Este ciclo de retroalimentación (sensor → controlador → motor → sensor) ocurre miles de veces por minuto, permitiendo un control suave y eficiente. La figura 8 ilustra la gráfica donde se aprecia la señal de cada sensor de efecto Hall dando paso a los 360 grados eléctricos.



**Figura 8:** Gráfica ilustrativa sobre el funcionamiento de la detección de los sensores de efecto Hall en los 360 grados eléctricos del motor

#### 2.4.4 Función Integral en el Rendimiento del Vehículo

**Arranque Suave y Confiable:** Permiten que el controlador ubique dónde está el rotor para aplicar el primer pulso de energía en la dirección correcta, evitando tirones.

**Control Preciso de Velocidad y Par:** La retroalimentación en tiempo real permite al controlador ajustar el ancho de pulso (PWM) para mantener la velocidad constante bajo diferentes cargas (subir una pendiente) o en los modos Eco/Normal.

**Eficiencia Energética Máxima:** La conmutación precisa en el momento óptimo asegura que el campo magnético del estator empuje siempre al rotor de la manera más efectiva, maximizando el par y minimizando las pérdidas, lo que se traduce directamente en mayor autonomía.

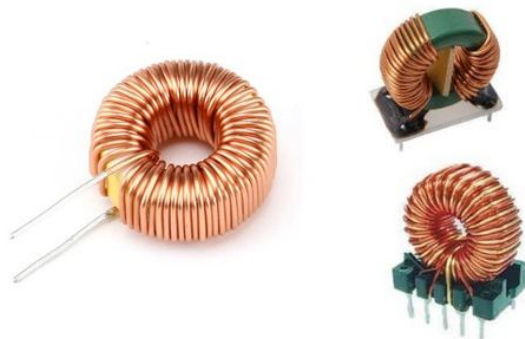
**Fiabilidad y Bajo Mantenimiento:** Al ser dispositivos de estado sólido sin contacto, son extremadamente duraderos y no sufren desgaste, a diferencia de las escobillas de un motor tradicional.

## **2.5 Bobinas Toroidales como Detectores de Presencia de Voltaje en Motores BLDC**

### **2.5.1 Principio de Operación**

En el sistema de monitoreo propuesto para el motor BLDC, se requiere un método seguro y no invasivo para detectar la presencia de voltaje en cada fase. El uso de bobinas toroidales (transformadores de corriente toroidales) se fundamenta en el Principio de Inducción Electromagnética de Faraday, que establece que un campo magnético variable induce una fuerza electromotriz (FEM) en un circuito.

Cuando el conductor de una fase del motor (que transporta corriente alterna) atraviesa el centro de un toroide, la corriente genera un campo magnético variable que se concentra en el núcleo. Una bobina secundaria enrollada alrededor de este núcleo es atravesada por un flujo magnético  $\Phi$  variable, induciendo un voltaje proporcional a la tasa de cambio de la corriente primaria, en la figura 9 se muestra la estructura de la bobina.



**Figura 9: Ejemplos ilustrativos de bobinas toroidales**

La relación fundamental está dada por la Ley de Faraday-Lenz:

$$V_{\text{salida}} = -N \frac{d\Phi}{dt} = -k \frac{dI_{\text{fase}}}{dt} \quad (2.1)$$

donde  $V_{\text{salida}}$  es el voltaje inducido,  $N$  es el número de vueltas del secundario, y  $k$  es una constante que depende de las características del toroide. Esta configuración actúa como un sensor de corriente no invasivo, cuyo propósito es generar una señal indicadora de la presencia de corriente, correlacionada con la aplicación de voltaje a la fase por el inversor.

### 2.5.2 Aplicación Específica en el Sistema de Monitoreo

En la motocicleta eléctrica Keeway Ride E1 con motor BLDC de 1500 W, la implementación se realiza de la siguiente manera:

- **Configuración Física:** Se instalan tres bobinas toroidales independientes, una por cada fase (G, B, Y). El conductor de cada fase se pasa una vez por el orificio central del toroide, actuando como devanado primario ( $N_p = 1$ ). El conductor no se corta ni altera eléctricamente.
- **Integración:** La salida digital de cada comparador se conecta a pines de entrada digital en la Raspberry Pi Pico W, permitiendo verificar en tiempo real si el controlador está energizando cada fase.

### 2.5.3 Ventajas Clave para el Proyecto

- **Aislamiento Galvánico Total:** La señal de detección está completamente aislada del circuito de potencia de alto voltaje (60 V+), protegiendo la electrónica de bajo voltaje.
- **No Invasividad:** No requiere cortar o soldar cables de potencia; solo pasar el conductor por el toroide. Preserva la integridad del sistema original.
- **Alta Inmunidad al Ruido:** La geometría toroidal y el material del núcleo (ferrita, nanocristalino) ofrecen excelente inmunidad a campos magnéticos externos.
- **Respuesta Rápida:** Capaz de detectar los pulsos de corriente de alta frecuencia de la conmutación PWM del motor BLDC.

## 2.6 Sensor ACS712

El ACS712 (mostrado en la figura 10) es un sensor de corriente basado en efecto Hall, diseñado para proporcionar una medida aislada y precisa de corriente alterna (AC) o continua (DC). Su arquitectura integrada ofrece una solución fiable para la medición de corrientes en entornos de potencia, como el monitoreo de fases en motores eléctricos.



Figura 10: Sensor ACS712

### 2.6.1 Principio de Operación y Arquitectura Interna

El funcionamiento del ACS712 se fundamenta en el efecto Hall, aplicado de forma integrada y optimizada para medición de corriente. Su arquitectura consta de los siguientes elementos clave:

**Conductor de Potencia Interno:** Un conductor de cobre de baja resistencia (típicamente  $1.2\text{ m}\Omega$ ) se integra en el mismo sustrato del semiconductor. La corriente a medir ( $I_p$ ) fluye a través de este conductor, generando un campo magnético concéntrico y proporcional a su intensidad.

**Circuito Magnético de Concentración:** Un núcleo magnético integrado canaliza y concentra el campo magnético generado por el conductor de cobre hacia el elemento sensor de efecto Hall. Esta concentración optimiza la sensibilidad y la linealidad de la medición.

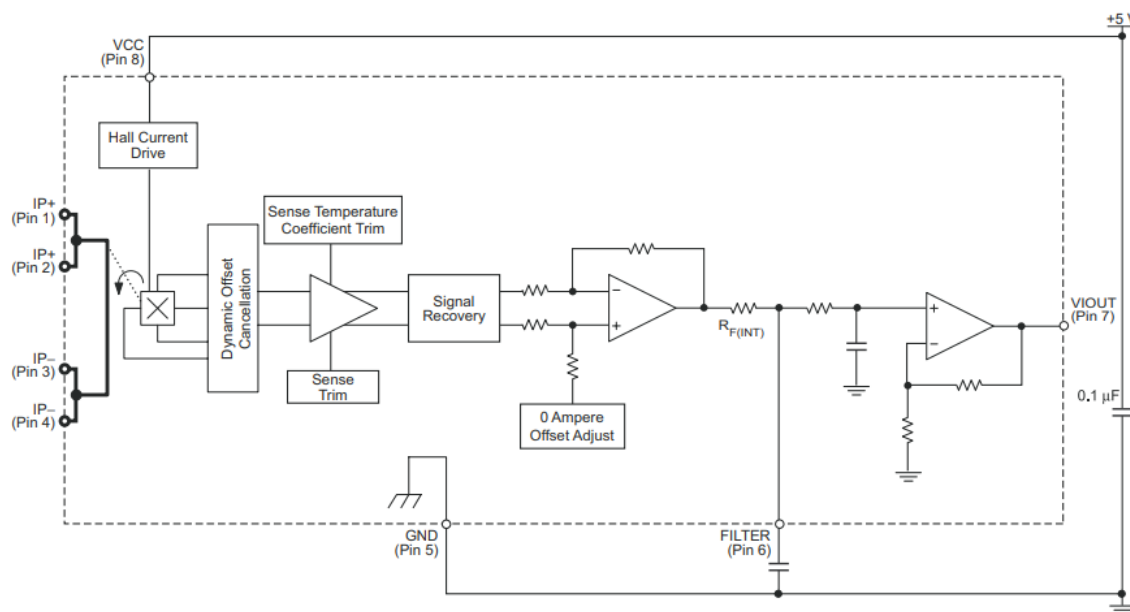
**Elemento Sensor Hall y Circuito de Acondicionamiento:** El sensor de efecto Hall, ubicado

en el centro del circuito magnético, produce un voltaje diferencial proporcional a la densidad de flujo magnético aplicado. Este voltaje es procesado por un amplificador diferencial de ganancia fija y alta linealidad.

**Circuito de Referencia y Regulación de Temperatura:** El chip incluye referencias de voltaje precisas y circuitos de compensación de temperatura. Estos sistemas estabilizan el punto de salida en cero amperios ( $V_{out}(Q)$ ) y la sensibilidad (mV/A) ante variaciones térmicas, garantizando precisión en un amplio rango de condiciones operativas.

**Aislamiento Galvánico:** La corriente medida ( $I_p$ ) y el circuito de señal de salida ( $V_{out}$ ) están eléctricamente aislados. Esta separación se logra físicamente, ya que el campo magnético es el medio de acoplamiento, protegiendo así al circuito de baja señal (microcontrolador) de los transientes de alta potencia.

El diagrama de este sensor se muestra en la figura 11



**Figura 11: Diagrama a bloques del funcionamiento ACS712**

La salida final es un voltaje analógico ( $V_{out}$ ), referenciado a la mitad del voltaje de alimentación ( $V_{cc}/2$ ), que varía linealmente con la corriente que fluye a través del conductor interno.

### 2.6.2 Especificaciones Técnicas del Modelo ACS712

A continuación se detallan las especificaciones fundamentales del sensor utilizado en este proyecto:

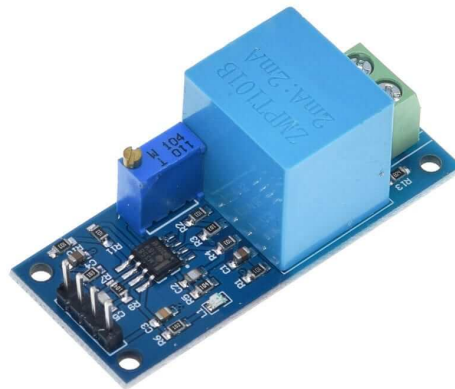
- **Modelo Comercial:** ACS712ELCTR-30A-T (Paquete SOIC-8).
- **Rango de Corriente de Medición:**  $\pm 30$  A (corriente alterna o continua bidireccional).
- **Sensibilidad (Sens):**  $66 \text{ mV/A} \pm 1.5 \%$ , típica.
- **Voltaje de Alimentación (Vcc):** Rango operativo de 4.5 V a 5.5 V DC. Típicamente 5.0 V.
- **Voltaje de Salida en Corriente Cero ( $V_{out(Q)}$ ):** Equivalente a  $V_{cc}/2$ . Con  $V_{cc}=5$  V, el valor típico es 2.5 V.
- **Resistencia Interna del Conductor:**  $1.2 \text{ m}\Omega$ , típica.
- **Ancho de Banda:** 80 kHz, típico.
- **Tiempo de Subida de la Salida:**  $5 \mu\text{s}$ , típico, ante un paso de corriente.
- **Exactitud Total de Salida:**
  - Error de sensibilidad a temperatura ambiente:  $\pm 1.5 \%$ , típico.
  - Error de offset a temperatura ambiente:  $\pm 30 \text{ mV}$ , típico.
  - Error total (incluyendo variación lineal y térmica) en el rango completo:  $< \pm 4 \%$ , típico.
- **Linealidad:** Mejor que  $\pm 1.5 \%$  del valor de fondo de escala (Full Scale), en el rango de  $I_p = \pm 30$  A.
- **Rango de Temperatura Operativa:**  $-40^\circ\text{C}$  a  $+85^\circ\text{C}$ .
- **Características de Aislamiento:**
  - Voltaje de aislamiento entre pines 1-4 y 5-8:  $2100 \text{ V}_{\text{rms}}$ , mínimo.

- Resistencia de aislamiento:  $> 1 \text{ M}\Omega$ .
- **Consumo de Corriente:** 10 mA, típico.

Estas especificaciones, particularmente su linealidad, aislamiento galvánico y respuesta en frecuencia, lo posicionan como un componente adecuado para el monitoreo de las corrientes de fase variables en un motor BLDC, sentando las bases técnicas para su implementación en la metodología del sistema de adquisición de datos.

## 2.7 Sensor ZMPT101B

El módulo sensor ZMPT101B (mostrado en la figura 12) está diseñado para la medición precisa y segura de voltajes alternos (AC). Su arquitectura se basa en un transformador de potencial de precisión que proporciona aislamiento galvánico, seguido de un circuito acondicionador de señal que adapta la salida para su lectura por un convertidor analógico-digital (ADC). En la figura 12 se muestra la apariencia del sensor:



**Figura 12: Sensor ZMPT101B**

### 2.7.1 Principio de Operación y Arquitectura

El funcionamiento del módulo se estructura en dos etapas principales:

### 2.7.1.1 Transformación y Aislamiento (Etapa de Entrada)

El componente central es un transformador de microvoltaje de precisión. El voltaje AC de entrada ( $V_{in}$ ), puede alcanzar hasta 250V RMS, se aplica al devanado primario del transformador. Este transformador cumple dos funciones críticas:

**Reducción de Voltaje:** Disminuye el voltaje de entrada a un nivel de baja amplitud y seguro, regularmente en el orden de milivoltios.

**Aislamiento Galvánico:** Separa eléctricamente el circuito de alta tensión (lado primario) del circuito de medición de baja tensión (lado secundario). El transformador proporciona una barrera de aislamiento capaz de soportar tensiones de hasta 3000 VAC - 4000 V, garantizando la seguridad del sistema de adquisición de datos y del operador.

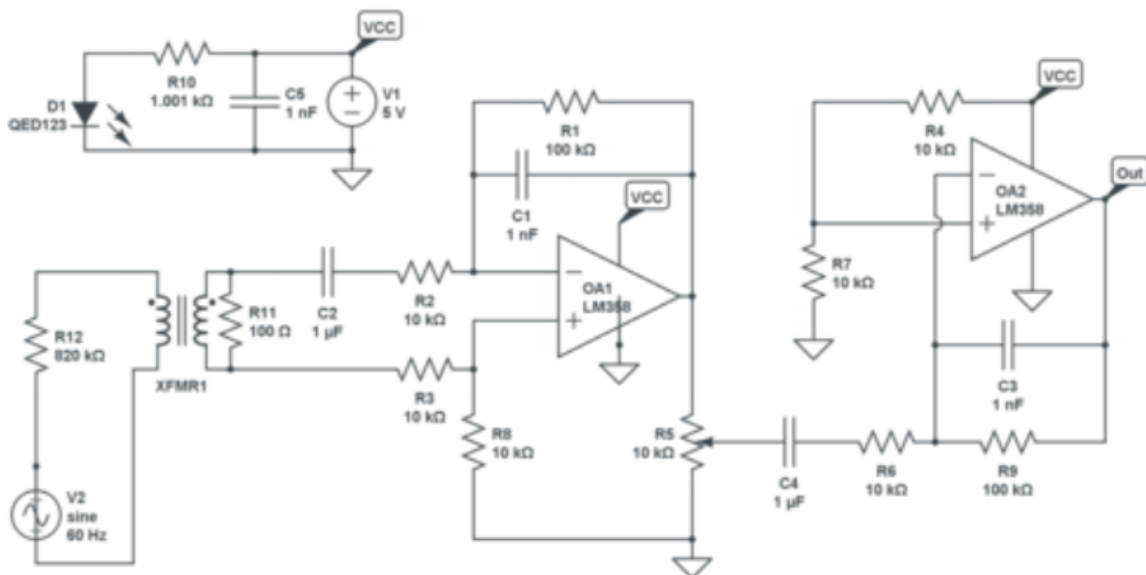
### 2.7.1.2 Acondicionamiento de Señal (Etapa de Salida)

La señal de baja amplitud del secundario del transformador es procesada por un circuito basado en un amplificador operacional (generalmente un LM358). Este circuito realiza dos funciones clave:

**Amplificación:** Ajusta la amplitud de la señal de CA a un rango utilizable.

**Adición de un Voltaje de Desplazamiento (Offset):** Superpone un componente de corriente continua (DC) a la señal de CA. Este desplazamiento se fija en  $VCC/2$ , donde VCC es el voltaje de alimentación del módulo (típicamente 5V o 3.3V). Por lo tanto, una señal de entrada de 0V AC resulta en un voltaje de salida constante de 2.5V (con  $VCC=5V$ ) o 1.65V (con  $VCC=3.3V$ ). Esto permite que toda la onda sinusoidal de salida se mantenga dentro del rango positivo de 0V a VCC, haciéndola completamente legible por el ADC de un microcontrolador que no puede procesar voltajes negativos.

En la figura 13 se muestra el diagrama sobre el funcionamiento del sensor.



**Figura 13: Diagrama del circuito ZMPT101B**

### 2.7.2 Especificaciones Técnicas del Módulo y del Transformador

- **Rango de Voltaje de Entrada (AC):** Máximo 250 V RMS.
- **Voltaje de Alimentación (VCC):** 5 VDC – 30 VDC. Típicamente operado a 5V para compatibilidad.
- **Rango de Voltaje de Salida (Señal Acondicionada):** De 0V a VCC, centrado en  $VCC/2$ .
- **Tipo de Señal de Salida:** Analógica, forma de onda sinusoidal con componente DC offset.
- **Ajuste de Amplitud:** Incluye un potenciómetro multivuelta (trimmer) que permite calibrar la ganancia y ajustar la amplitud de la señal de salida alrededor del punto de offset  $VCC/2$ .
- **Dimensiones Típicas:** Aproximadamente 50 mm × 20 mm × 20 mm.

### Especificaciones del Transformador de Precisión (Núcleo ZMPT101B):

- **Clase de Precisión:** 0.2.
- **Linealidad:** 0.1 % – 1 %.
- **Corriente Nominal de Entrada/Salida:** 2 mA.
- **Relación de Transformación (Turns Ratio):** 1000:1000.
- **Rango Lineal de Corriente:** 0 a 3 mA (a 50  $\Omega$ ).
- **Rango de Frecuencia Operativa:** Diseñado para redes de 50 Hz / 60 Hz.
- **Error de Fase (Phase Angle Error):**  $\leq 20$  minutos de arco (a 50  $\Omega$ ).
- **Aislamiento Eléctrico (Entrada-Salida):** 3000 V – 4000 V.

### 2.7.3 Características de la Señal de Salida y Procesamiento

La salida del módulo (pin OUT) es una señal analógica que replica la forma de onda de entrada con las siguientes modificaciones:

**Factor de Escala (Atenuación):** El voltaje de alta tensión se reduce a una amplitud pico máxima de aproximadamente  $V_{CC}/2$ .

**Componente DC (Offset):** La señal oscila simétricamente alrededor de un valor de reposo de  $V_{CC}/2$ , no de 0V. Esta es la característica más importante para la interfaz con el ADC.

**Compatibilidad con ADC:** Gracias al offset, un microcontrolador puede muestrear la onda completa. Para obtener el valor real del voltaje AC, el firmware debe:

1. **Muestreo de la señal en el tiempo:** Adquirir un número suficiente de puntos de la señal de salida del sensor ( $V_{out}[n]$ ) a lo largo de uno o varios ciclos de la red eléctrica, utilizando el Convertidor Analógico-Digital (ADC) del microcontrolador.
2. **Eliminación del *offset* DC:** Calcular el valor promedio o componente continua de la señal muestreada, que corresponde al voltaje de reposo  $V_{CC}/2$ , y sustraerlo de cada muestra para obtener la señal alterna pura.

$$V_{AC}[n] = V_{out}[n] - \overline{V_{out}}, \quad \text{donde } \overline{V_{out}} \approx \frac{V_{CC}}{2} \quad (2.2)$$

3. **Cálculo del valor eficaz (RMS):** Calcular la raíz cuadrada del valor cuadrático medio de la señal alterna resultante ( $V_{AC}[n]$ ).

$$V_{RMS} = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} (V_{AC}[n])^2} \quad (2.3)$$

4. **Escalado a la escala real:** Aplicar el factor de calibración o sensibilidad obtenido experimentalmente para convertir el valor  $V_{RMS}$  calculado en el voltaje RMS real del primario.

$$V_{Primario}(RMS) = V_{RMS} \times K_{calibracion} \quad (2.4)$$

Donde  $K_{calibracion}$  es la constante que relaciona el voltaje RMS medido a la salida del módulo con el voltaje RMS presente en la entrada de alta tensión.

La etapa de acondicionamiento con offset ajustable garantiza la compatibilidad con los ADC de microcontroladores estándar, permitiendo una construcción digital precisa de la forma de onda. Estas características nos da pauta para considerarlo un componente fundamental para el monitoreo de los voltajes de fase en el motor BLDC, donde la precisión, la seguridad y la integridad de la señal son parámetros críticos.

## 2.8 Raspberry Pi 4

La Raspberry Pi 4 es un sistema de un solo tablero (por su siglas en inglés: SBC) de cuarta generación que representa un salto cualitativo significativo en potencia de procesamiento, conectividad y capacidad de memoria respecto a sus predecesores, consolidándose como una plataforma versátil para aplicaciones de control y adquisición de datos. En la figura 14 se muestra el dispositivo.



**Figura 14: Raspberry Pi 4**

### **2.8.1 Especificaciones Técnicas Fundamentales**

El sistema se basa en una arquitectura moderna que soporta sistemas operativos completos como Raspberry Pi OS (Linux).

**Unidad Central de Procesamiento (CPU):** Cuatro núcleos ARM Cortex-A72 de 64 bits, integrados en el Sistema en un Chip (SoC) Broadcom BCM2711. La frecuencia de reloj base es de 1.5 GHz, aunque revisiones posteriores del chip (stepping C0) alcanzan los 1.8 GHz. Esta configuración ofrece un rendimiento computacional entre 2 y 4 veces superior al de la Raspberry Pi 3B+.

**Unidad de Procesamiento Gráfico (GPU):** VideoCore VI operando a 500 MHz. Es compatible con las API gráficas OpenGL ES 3.x y Vulkan 1.2, e integra capacidades avanzadas de decodificación de video por hardware, incluyendo H.265 (4K a 60 fps) y H.264 (1080p a 60 fps).

**Memoria (RAM):** Utiliza memoria LPDDR4 SDRAM a una velocidad de hasta 3.2 GHz. Se comercializa en variantes de 1 GB, 2 GB, 4 GB y 8 GB, lo que permite seleccionar el modelo adecuado según la complejidad de la aplicación. A diferencia de modelos anteriores, la asignación de memoria para la GPU es dinámica.

**Almacenamiento Principal:** Una ranura para tarjeta MicroSD para el sistema operativo y el

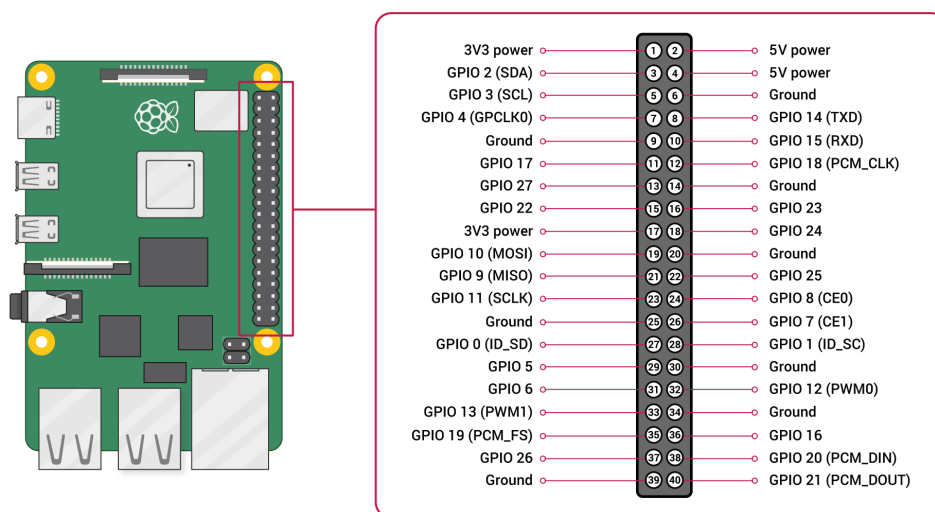
almacenamiento de datos.

## 2.8.2 Capacidades de Entrada/Salida y Conectividad

El diseño de la Raspberry Pi 4 prioriza la conectividad de alto rendimiento para aplicaciones de servidor y proyectos que involucren periféricos.

### 2.8.2.1 Puertos de Expansión

**Cabecera GPIO:** Un conector estándar de 40 pines (compatible con versiones anteriores) que soporta protocolos como I<sup>2</sup>C, SPI y UART para la interconexión con sensores, actuadores y otros componentes electrónicos. En la figura 15 se muestra la distribución de los puertos GPIO.



**Figura 15: Esquema de pines GPIO para Raspberry Pi 4**

**Puertos USB:** Cuenta con dos puertos USB 3.0 y dos puertos USB 2.0. Los puertos USB 3.0 permiten tasas de transferencia de hasta 5 Gbps, lo que es ideal para discos duros externos o adaptadores de red de alta velocidad.

### 2.8.2.2 Conectividad de Red y Periféricos

**Ethernet:** Un controlador Gigabit Ethernet nativo, que alcanza velocidades reales de 1000 Mbps al no estar limitado por un bus USB como en modelos anteriores.

**Conectividad Inalámbrica:** Módulo Wi-Fi 802.11ac (Wi-Fi 5) dual banda (2.4 GHz y 5 GHz) y Bluetooth 5.0 con BLE (Bluetooth Low Energy).

### 2.8.2.3 Salida de Video y Multimedia

**Salida de Video:** Dos puertos micro-HDMI, capaces de manejar dos monitores en configuraciones extendidas o espejadas. Soporta una resolución de 4K a 60 fps en un monitor, o dos monitores 4K a 30 Hz.

**Interfaces Especializadas:** Incluye un puerto MIPI CSI para conectar cámaras y un puerto MIPI DSI para pantallas táctiles.

## 2.8.3 Características Físicas y de Potencia

**Alimentación:** Utiliza un conector USB-C que requiere una fuente de alimentación estable de 5V y al menos 3A para un funcionamiento óptimo, especialmente cuando se utilizan varios periféricos. También es posible la alimentación a través de los pines GPIO o mediante Power over Ethernet (PoE) con un HAT adicional.

**Dimensiones:** Mantiene el factor de forma compacto clásico, con unas dimensiones aproximadas de 85.6 mm x 56.5 mm.

**Consideración Térmica:** El SoC BCM2711, fabricado en un proceso de 28 nm, es más eficiente que el de generaciones anteriores (40 nm). Sin embargo, bajo cargas de trabajo sostenidas y alta frecuencia, se recomienda el uso de disipadores pasivos o refrigeración activa para evitar el thermal throttling y mantener el máximo rendimiento.

La combinación de su potente CPU multinúcleo, las opciones de RAM LPDDR4, el bus Gigabit Ethernet de alta velocidad, los puertos USB 3.0 y su extensivo conjunto de interfaces GPIO, se considera a la Raspberry Pi 4 como una plataforma central conveniente para

sistemas embebidos, como servidor de adquisición y procesamiento de datos de un sistema de monitoreo.

## 2.9 Raspberry Pi Pico W

La Raspberry Pi Pico W es una placa de microcontrolador de alto rendimiento y bajo costo, basada en el sistema-en-chip (SoC) RP2040 diseñado por Raspberry Pi Ltd. En la arquitectura de monitoreo propuesta, su función es actuar como una Unidad de Adquisición de Datos (por sus siglas en inglés: DAQ) distribuida, dedicada a la lectura, acondicionamiento primario y transmisión inalámbrica de las señales analógicas y digitales provenientes del motor BLDC y su controlador. En la figura 16 se muestra el microncontrolador.

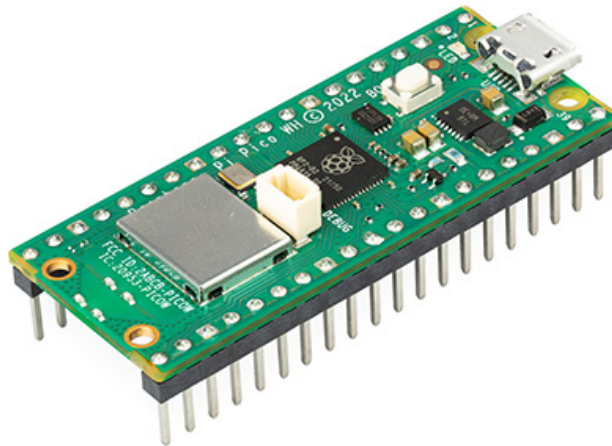


Figura 16: Raspberry Pi 4

### 2.9.1 Características Técnicas del Hardware

Las características clave que la hacen idónea para esta aplicación son:

- **Procesador:** Doble núcleo ARM Cortex-M0+ a 133 mega hertz, permitiendo ejecución determinista de rutinas de lectura de sensores y comunicación.

- **Memoria:** 264 kilo bytes de SRAM integrada, suficiente para el búfer de datos y pila de red.
- **Conectividad:** Módulo Wi-Fi 2.4 GHz Infineon CYW43439 Estándar 802.11n, con soporte nativo para protocolos TCP/IP, fundamental para la comunicación UDP con el servidor central (Raspberry Pi 4).
- **Entradas/Salidas de Propósito General (GPIO):** 26 pines GPIO multifunción en el conector lateral, de los cuales varios pueden configurarse como entradas analógicas, digitales o para interfaces seriales.
- **Convertidor Analógico-Digital (ADC):** Característica crítica para este proyecto, detallada en la sección [2.9.2](#).
- **Alimentación:** Opera en un rango de 1.8 volts a 5.5 volts, pero las señales de sus pines, incluido el ADC, están referenciadas a su voltaje de alimentación 3.3 volts.

## 2.9.2 El Sistema de Convertidor Analógico-Digital (ADC)

El ADC integrado en el RP2040 es el componente que permite la digitalización de las señales de voltaje y corriente.

- **Arquitectura y Canales:** Es un ADC sucesivo de aproximación (SAR) con una resolución de 12 bits. Dispone de 5 canales de entrada dedicados: 3 están disponibles en pines GPIO (GP26, GP27, GP28), un cuarto está conectado a un sensor de temperatura interno, y el quinto puede usarse para medir el voltaje de alimentación del propio SoC (VSYS).  
item **Rango de Medición:** El ADC mide voltajes en el rango de 0 volts a la tensión de referencia, que está fija en 3.3 volts (el mismo que el AVDD del chip). Esto establece la resolución teórica del sistema:

$$V_{LSB} = \frac{V_{REF}}{2^{12}} = \frac{3.3V}{4096} \approx 0.806mV \quad (2.5)$$

donde  $V_{LSB}$  es el valor del *Least Significant Bit*. Una lectura digital  $D$  se relaciona con el voltaje de entrada  $V_{in}$  como  $D = \frac{V_{in}}{V_{LSB}}$ .

- **Velocidad de Muestreo:** Puede alcanzar una tasa de muestreo efectiva de hasta utilizando la técnica de *overclocking* del núcleo, aunque para la mayoría de aplicaciones de energía, tasas entre 1 kilo hertz y 10 kilo hertz son suficientes.
- **Consideraciones de Precisión:** El ADC es mono-rail (solo mide voltajes positivos) y su referencia no es independiente, lo que introduce cierto ruido de medición. Se recomienda el uso de promediado de muestras y un filtro paso-bajo analógico (*anti-aliasing*) en cada canal para mejorar la precisión.

### 2.9.3 Configuración de los GPIO para el Sistema de Monitoreo

La configuración propuesta de los pines de la Pico W para la adquisición de las 4 variables clave por fase es la siguiente:

- **Señales Analógicas (Entradas ADC):**
  - GP26, GP27, GP28 (ADC0, ADC1, ADC2): Para conectar las señales de **corriente de fase** (ACS712) y **voltaje de fase** (ZMPT101B), previo acondicionamiento a un rango de 0-3.3 volts. Estas señales deben estar acopladas en CA y con un offset de 1.65 volts para medir valores positivos y negativos.
  - De igual manera ocupando la misma designación de puertos se emplean para la detección de presencia/ausencia del voltaje de las fases así como también la detección binaria de las señales de los sensores de efecto Hall.
- **Comunicación:** El módulo **Wi-Fi** integrado se utiliza para el envío de los paquetes de datos estructurados vía UDP a la dirección IP fija de la Raspberry Pi 4, usando el chip CYW43439 y la pila de protocolos **lwIP**.

En la figura 17 se muestra la distribución de los GPIO de este microcontrolador.



esenciales: eliminan ruido de alta frecuencia no deseado, atenúan interferencias, previenen el aliasing (solapamiento espectral) y preparan las señales para etapas posteriores de análisis o toma de decisiones. Su diseño y selección dependen críticamente de las características de la señal de interés y de los objetivos de procesamiento.

### **2.10.1 Clasificación Fundamental: Filtros FIR e IIR**

Los filtros digitales se pueden clasificar en dos grandes familias, según la duración de su respuesta al impulso:

Filtros FIR (respuesta finita al impulso): su salida solo depende de un número finito de entradas pasadas y presentes; no utilizan realimentación (feedback), por lo que son siempre estables y tienen fase lineal, pero pueden requerir un orden muy alto (y, por tanto, mucha complejidad computacional) para lograr pendientes de atenuación agudas.

Filtros IIR (respuesta infinita al impulso): su salida depende de entradas pasadas, presentes y salidas pasadas; sí usan retroalimentación, por lo que pueden ser inestables y no tienen fase lineal, pero pueden conseguir pendientes de atenuación muy elevadas con órdenes bajos como entradas anteriores realimentadas. Esta recursividad les permite conseguir una pendiente de atenuación muy pronunciada con un orden muy bajo, como los filtros analógicos clásicos, pero deben ser analizados para asegurar la estabilidad.

La elección entre FIR e IIR es la compensación clásica entre linealidad de fase, estabilidad, complejidad computacional y respuesta en frecuencia. Para aplicaciones en tiempo real o embebidas donde se tienen restricciones de procesamiento y la fase lineal no es esencial, los filtros IIR de bajo orden suelen ser la mejor opción.

### **2.11 Filtro Pasa-Bajas IIR de Primer Orden**

El filtro IIR de primer orden es el elemento más simple de la familia de filtros recursivos y su análisis es interesante no sólo por su aplicación, sino como ejemplo de los principios de los sistemas discretos en el tiempo.

### 2.11.1 Nomenclatura

Primer Orden: Se refiere al orden del polinomio denominador de su función de transferencia en  $Z$ , lo que implica que su ecuación en diferencias solo involucra una muestra pasada de la salida ( $y[n-1]$ ). Esto determina su complejidad y su respuesta en frecuencia.

### 2.11.2 Ecuación en Diferencias y Función de Transferencia

El comportamiento del filtro se define por su ecuación en diferencias, que describe cómo se calcula cada muestra de salida  $y[n]$  a partir de las muestras de entrada  $x[n]$  y de salidas previas:

$$y[n] = \alpha \cdot x[n] + (1 - \alpha) \cdot y[n - 1] \quad (2.6)$$

donde:

- $x[n]$ : Muestra de entrada actual.
- $y[n]$ : Muestra de salida actual (filtrada).
- $y[n - 1]$ : Muestra de salida anterior (retardada un período de muestreo).
- $\alpha$  (alpha): **Coeficiente de suavizado**, un parámetro real comprendido entre 0 y 1 ( $0 < \alpha < 1$ ) que determina la “agresividad” del filtro.

Esta ecuación tiene una interpretación física intuitiva: la salida es un promedio ponderado entre la entrada nueva y la salida anterior; un valor de  $\alpha$  cercano a 1 le da más peso a la entrada nueva, obteniéndose un filtro de respuesta rápida pero con poca atenuación de ruido; un valor de  $\alpha$  cercano a 0 le da más peso a la historia pasada, suavizando más (filtrado más agresivo) pero con una respuesta transitoria más lenta.

#### 2.11.2.1 Función de Transferencia en el Dominio $Z$

Aplicando la **transformada  $Z$**  a la ecuación (2.6) y reorganizando términos, se obtiene la función de transferencia  $H(z)$ :

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\alpha}{1 - (1 - \alpha)z^{-1}} \quad (2.7)$$

Esta expresión en el dominio  $Z$  es fundamental para estudiar las características frecuenciales y de estabilidad del filtro. El sistema posee un único polo en  $z = (1 - \alpha)$ . Como  $0 < \alpha < 1$ , este polo siempre se ubica dentro del círculo unitario en el plano  $Z$ , lo que asegura formalmente la estabilidad del filtro.

### 2.11.3 Relación con el Filtro Analógico RC y Diseño del Parámetro $\alpha$

El filtro IIR de primer orden descrito por la ecuación (2.6) es la contraparte digital exacta del clásico filtro analógico RC pasa-bajas, cuyo comportamiento en el dominio del tiempo está constituido por una constante de tiempo  $\tau = R \cdot C$ .

Mediante métodos de discretización, como el método de invariancia al impulso o la aproximación de Euler, se puede derivar la relación entre el parámetro digital  $\alpha$  y la frecuencia de corte analógica deseada  $f_c$ . La fórmula fundamental que relaciona estos parámetros es:

$$\alpha = 1 - e^{-2\pi f_c T_s} \quad (2.8)$$

donde  $T_s = 1/f_s$  es el período de muestreo (inverso de la frecuencia de muestreo  $f_s$ ).

Para el caso común en la práctica donde  $f_c T_s$  es pequeño (es decir, la frecuencia de corte  $f_c$  es mucho menor que la frecuencia de muestreo  $f_s$ ), se puede utilizar una aproximación de primer orden de la serie de Taylor para simplificar el cálculo de  $\alpha$ :

$$\alpha \approx 2\pi f_c T_s \quad (2.9)$$

**(Aproximación válida cuando  $f_c \ll f_s$ ).**

Una vez calculado  $\alpha$ , la frecuencia de corte digital efectiva  $f_{c,d}$  (en Hz) puede obtenerse, para valores pequeños de  $\alpha$ , mediante la expresión inversa de la aproximación anterior:

$$f_{c,d} \approx \frac{\alpha}{2\pi T_s} \quad (2.10)$$

La tabla 3 resume los parámetros clave y su efecto:

**Tabla 3: Efecto de  $\alpha$  en el filtro.**

| Parámetro      | Símbolo | $\alpha$ alto | $\alpha$ bajo |
|----------------|---------|---------------|---------------|
| Resp. temporal | —       | Rápida        | Lenta         |
| $BW$           | $f_c$   | Ancho         | Estrecho      |
| Estabilidad    | —       | Estable       | Estable       |

## 2.11.4 Respuesta al Impulso, en Frecuencia y Comparación con FIR

### 2.11.4.1 Respuesta al Impulso $h[n]$

Como su nombre *IIR* (Respuesta Infinita al Impulso) indica, la respuesta del filtro a un impulso unitario  $\delta[n]$  es una secuencia exponencialmente decreciente que, en teoría, no se extingue:

$$h[n] = \alpha(1 - \alpha)^n, \quad \text{para } n \geq 0. \quad (2.11)$$

Esto contrasta con un filtro *FIR* (Respuesta Finita al Impulso), cuya respuesta  $h[n]$  tiene una duración finita y definida.

### 2.11.4.2 Respuesta en Frecuencia

Haciendo la sustitución  $z = e^{j\omega}$  (donde  $\omega$  es la frecuencia digital normalizada) en la función de transferencia  $H(z)$  de la ecuación (2.7), se obtiene la respuesta en frecuencia. Su magnitud  $|H(e^{j\omega})|$  tiene la forma pasa-bajas típica, siendo la frecuencia de corte a  $-3$  dB dependiente únicamente del parámetro  $\alpha$ , según la relación dada en la ecuación (2.8). Un efecto característico de este filtro (y en general de los IIR) es que su fase  $\angle H(e^{j\omega})$  introduce un retardo no lineal en las distintas componentes frecuenciales de la señal, lo cual es una

desventaja en comparación con los filtros FIR de fase lineal, preferibles cuando se requiere preservar la forma de onda temporal sin distorsión.

### 2.11.5 Consideraciones para la Implementación Práctica

Al trasladar el diseño teórico a código embebido, es vital considerar los siguientes aspectos para garantizar robustez y precisión:

- **Condición Inicial:** La muestra inicial  $y[-1]$  (la salida antes de la primera muestra) se debe definir explícitamente; una forma común es darle el valor de la primera muestra de entrada  $x[0]$  o simplemente 0, en caso de no tener información previa de la señal.
- **Precisión Numérica (Especialmente en Punto Fijo):** Para evitar problemas numéricos, es recomendable utilizar la forma delta de la ecuación en diferencias, que es numéricamente más robusta:

$$y[n] = y[n-1] + \alpha (x[n] - y[n-1]). \quad (2.12)$$

Esta forma requiere una sola multiplicación y tiene menor margen de error en cuanto a cuantización. Cuando se implementa en hardware de punto fijo, es crucial almacenar el estado  $y[n-1]$  con una precisión (número de bits) mayor que la de la entrada y los coeficientes, para evitar errores de truncamiento acumulativos que pueden comprometer el funcionamiento del filtro.

## 2.12 Filtro de Mediana

El filtro mediana es un ejemplo de un tipo diferente de procesamiento de señales: la filtración no lineal; es un tipo de estadística robusta, que trata de hacer buenas estimaciones cuando una fracción arbitraria de los datos está contaminada. Se usa para acondicionar señales de sensores en microcontroladores, hasta para procesar imágenes médicas y de video, donde es insuperable para eliminar ruido impulsivo sin destruir discontinuidades agudas (bordes) en la señal.

### 2.12.1 Base Estadística y Principio de Operación

La mediana como medida de tendencia central es el valor que separa en dos partes iguales a un conjunto ordenado de datos. A diferencia de la media aritmética, no se ve afectada por los valores extremos (outliers); esta característica de robustez es la base del filtro.

El filtro de mediana móvil es una aplicación directa de este concepto. Opera definiendo una ventana deslizante de longitud  $N$  (generalmente impar) sobre la señal discreta. Para cada posición de la ventana, se capturan las  $N$  muestras abarcadas, se ordenan en secuencia ascendente y la muestra central del conjunto ordenado se selecciona como la salida filtrada para ese instante.

### 2.12.2 Descripción Algorítmica y Ejemplo

Matemáticamente, para una señal de entrada  $x[n]$  y una ventana de tamaño  $N = 2k + 1$ , la salida  $y[n]$  se calcula como:

$$y[n] = \text{mediana}\{x[n-k], x[n-k+1], \dots, x[n], \dots, x[n+k]\} \quad (2.13)$$

**Ejemplo Ilustrativo (1D):** Considérese la señal contaminada:  $x = [2, 3, 80, 6, 2, 3]$ , donde 80 es un pico de ruido impulsivo. Aplicando un filtro de mediana con ventana de tamaño 3:

- $y[1] = \text{mediana}(2, 3, 80) = 3$
- $y[2] = \text{mediana}(3, 80, 6) = 6$

El resultado,  $y = [2, 3, 6, 6, 3, 2]$ , demuestra la eliminación efectiva del ruido sin distorsionar severamente la señal base. En contraste, un filtro de media móvil del mismo tamaño produciría  $y_{\text{media}} \approx [1.7, 28.3, 29.7, \dots]$ , mostrando cómo el pico contamina las muestras adyacentes.

### 2.12.3 Propiedades y Análisis de Rendimiento del Filtro de Mediana

#### 2.12.3.1 Preservación de Bordes y Señales Escalón

A diferencia de los filtros lineales de suavizado (media móvil, gaussiano, etc.), que borran inevitablemente las transiciones bruscas, el filtro mediana las preserva intactas (su mejor propiedad en procesamiento de imágenes y señales). Un borde o escalón perfecto, al desplazarse por la ventana, dejará intacta la mediana siempre que más de la mitad de las muestras contenidas en la ventana pertenezcan al mismo nivel de la señal.

#### 2.12.4 Características de la Señal de Salida y Limitaciones

- **Señal 'Angulosa':** La salida solo puede tomar valores presentes en la ventana de entrada, lo que puede producir una señal con transiciones escalonadas y perder suavidad en tramos donde la señal original varía de forma continua.
- **Pérdida de Resolución Temporal (Retardo):** Al igual que otros filtros de ventana, introduce un retardo de grupo. Para una ventana de tamaño  $N$ , el retardo efectivo es de aproximadamente  $(N - 1)/2$  muestras. Este retardo es crítico en sistemas de control en tiempo real.
- **Costo Computacional:** El cálculo de la mediana requiere ordenar parcialmente la ventana.

#### 2.12.5 Directrices de Diseño e Implementación Práctica

##### 2.12.5.1 Selección del Tamaño de Ventana ( $N$ )

- **Valores Impares:** Se prefieren siempre ( $N = 3, 5, 7, 9, 11, \dots$ ) para evitar tener que promediar dos valores centrales, lo que debilitaría la capacidad de filtrado.
- **Compromiso:** Un  $N$  mayor ofrece un suavizado más fuerte y mayor robustez ante múltiples espurios consecutivos, pero aumenta el retardo y la distorsión de formas de onda finas. Valores típicos en sistemas de control y sensores oscilan entre 3 y 11.

### **2.12.6 Manejo de Bordes de la Señal**

Al inicio y fin de la señal, la ventana no está llena. Las estrategias comunes incluyen:

- No procesar los bordes, recortando la señal de salida.
- Relleno con ceros o con el valor del primer/último elemento.
- Reducir progresivamente el tamaño de la ventana en los extremos.

## **2.13 Protocolo de Datagramas de Usuario (UDP)**

El Protocolo de datagramas de usuario (UDP) es uno de los protocolos centrales de la suite de protocolos de Internet. Opera en la capa de transporte junto con su hermano más complejo, el Protocolo de control de transmisión (TCP). UDP fue diseñado con el principio de minimalismo en mente; proporciona un servicio de transporte simple y sin complicaciones que deja en manos de las aplicaciones la tarea de garantizar la fiabilidad cuando sea necesario. En contraste con los protocolos orientados a conexión, UDP es un protocolo sin conexión que prioriza la eficiencia y la baja latencia sobre la garantía de entrega.

### **2.13.1 Naturaleza y Propósito**

UDP es un protocolo de transporte no orientado a conexión; no hay establecimiento de conexión ni .apretón de manos.antes de transmitir datos. Cada pieza de datos (datagrama) se envía independientemente, el protocolo simplemente toma los datos de la aplicación, les agrega una cabecera mínima (puertos de origen y destino) y los entrega a la capa inferior de red para ser transmitido. No hay concepto de sesión, secuencia o confirmación a nivel de protocolo.

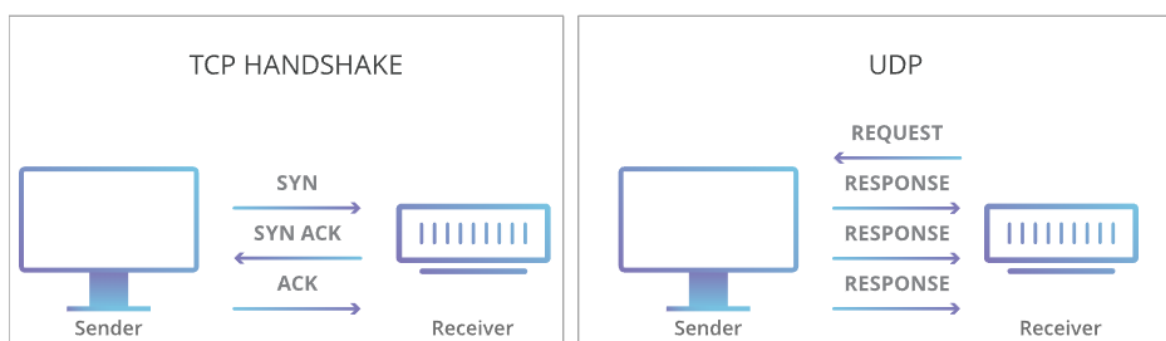
El objetivo de UDP es proporcionar un canal simple y rápido para aplicaciones que pueden tolerar cierta pérdida de datos y son sensibles al retardo o a la variación del retardo. Se diseñó para ofrecer la mínima sobrecarga posible, dejando a la aplicación la tarea de implementar sus propios mecanismos de control adaptados a sus necesidades específicas sin verse lastrada por las garantías —y la complejidad— que ofrece TCP.

### 2.13.2 Características Técnicas Principales

Las especificaciones de UDP se desprenden de su naturaleza simplista: la cabecera de tamaño fijo de solo 8 bytes refleja esta simplicidad, conteniendo 4 campos: puerto origen, puerto destino, longitud total del datagrama y suma de comprobación opcional para la detección básica de errores en la cabecera y los datos. La comunicación se realiza a través de puertos, números lógicos que permiten multiplexar varias aplicaciones/servicios en un mismo dispositivo. Un datagrama UDP se envía a un puerto específico en una dirección IP determinada, donde una aplicación 'escucha'. Al no tener control de flujo o congestión, los datagramas pueden ser enviados a la máxima velocidad que permitan la aplicación y la red subyacente, lo cual puede ser una ventaja y un peligro para la red.

Una característica propia es la falta de garantía; UDP no confirma la recepción de los datagramas, no los retransmite si se pierden y no garantiza que lleguen en el orden en que fueron enviados (esto queda a cargo de la aplicación, si es necesario). La suma de comprobación es opcional en IPv4, lo que refuerza aún más la naturaleza de 'mejor esfuerzo' de UDP; en IPv6 se hace obligatoria para mantener un mínimo de integridad de datos.

En la figura 18 se tiene una comparativa ilustrativa del protocolo UDP contra TCP.



**Figura 18: Esquema de comparativa protocolo UDP contra TCP**

### 2.13.3 Ventajas y Limitaciones

La principal ventaja de UDP radica en su baja sobrecarga y latencia. Al eliminar los mecanismos de establecimiento de conexión, confirmación y control, los datos pueden fluir con la

mínima demora posible. Su cabecera pequeña reduce el consumo de ancho de banda, y la falta de estado de conexión simplifica enormemente la implementación en servidores, que pueden atender a un gran número de clientes sin mantener información de sesión compleja.

Sin embargo, estas ventajas conllevan limitaciones propias. La más evidente es la falta de fiabilidad. Las aplicaciones que requieren la entrega garantizada y ordenada de todos los datos deben implementar sus propios mecanismos de acknowledgments (ACK) y retransmisiones, lo que puede resultar en una complejidad similar a la de TCP, pero potencialmente menos optimizada para las condiciones generales de la red. Tampoco ofrece control de congestión, por lo que una aplicación UDP mal diseñada podría saturar los enlaces de red sin ningún mecanismo de retroalimentación para reducir su tasa de envío, afectando negativamente a otro tráfico.

En resumen, UDP es una herramienta especializada dentro del arsenal de protocolos de Internet. Su valor no reside en lo que hace, sino en lo que ofrece a los desarrolladores, un lienzo en blanco sobre el cual construir servicios de comunicación optimizados para requisitos específicos de velocidad, simplicidad o modelo de distribución, asumiendo ellos la responsabilidad de gestionar la confiabilidad cuando sea un factor crítico.

## **2.14 Señales Críticas para el Monitoreo**

### **2.14.0.1 Señales de los Sensores de Efecto Hall**

El monitoreo de las señales digitales generadas por los sensores de efecto Hall es un aspecto fundamental para la operación y diagnóstico del motor BLDC. Estas señales proporcionan información crítica sobre la posición angular instantánea del rotor, permitiendo validar la comunicación básica entre el motor y su controlador. Confirmar que el controlador recibe esta información de forma correcta es un requisito indispensable para iniciar y mantener la secuencia de conmutación de las fases. Además, estas señales permiten correlacionar directamente el comando del conductor (a través del acelerador) con la acción del motor. Un aumento en la señal del acelerador debe traducirse en un incremento proporcional en la frecuencia de las señales de los sensores de efecto Hall, lo que corresponde a un aumento en

la velocidad eléctrica del motor. Esta correlación es esencial para verificar que el sistema de control responde de manera adecuada a las entradas del usuario.

#### **2.14.1 Detección de Presencia de Voltaje en Fases**

La verificación binaria de la presencia de voltaje en las fases del motor actúa como un sistema de alarma primario y una herramienta de diagnóstico rápido. Esta medición permite aislar de forma eficiente la localización de una falla, discerniendo si el problema reside en la parte de control (que genera las señales de conmutación) o en la etapa de potencia propiamente dicha. Por ejemplo, si las señales de los sensores de efecto Hall son correctas pero no se detecta voltaje en una fase, la falla se puede localizar con alta probabilidad en componentes del inversor del controlador, como los transistores de potencia, los fusibles o las conexiones de alto voltaje. De esta forma, se verifica que la decisión lógica del controlador, basada en la posición del rotor y la demanda del acelerador, se materializa efectivamente en una acción física: la aplicación de voltaje a los devanados correspondientes.

#### **2.14.2 Medición de Voltaje por Fase**

La medición del voltaje aplicado a cada fase del motor proporciona información valiosa sobre la calidad de la energía suministrada y el estado del controlador. Una de sus funciones principales es la detección de desbalances entre fases. Un voltaje RMS significativamente diferente en una fase respecto a las demás genera campos magnéticos desequilibrados en el estator, lo que provoca efectos adversos como vibraciones mecánicas, ruido acústico y un estrés térmico desigual en los devanados. Este desbalance, de no corregirse, reduce la eficiencia y la vida útil del motor. Asimismo, el voltaje por fase es una variable indispensable para el cálculo de la potencia eléctrica de entrada. Junto con la medición de corriente, permite calcular la potencia eléctrica instantánea suministrada a cada fase mediante la relación fundamental  $P = V \times I$ , sentando las bases para evaluar el consumo energético del sistema.

### **2.14.3 Medición de Corriente por Fase**

La corriente que circula por cada devanado es, posiblemente, la variable más significativa en información para el diagnóstico y análisis del rendimiento del motor. En primer lugar, actúa como un indicador directo de la carga mecánica y el esfuerzo del motor; la amplitud de la corriente es proporcional al par motor demandado. Por lo tanto, picos de corriente inesperados o sostenidos pueden alertar sobre una condición de sobrecarga mecánica, como frenos agarrados o un rodamiento dañado. En segundo lugar, permite la detección temprana de fallos internos en el motor. Un desbalance persistente en las corrientes de fase, donde una fase consume consistentemente un porcentaje significativamente mayor (por ejemplo, un +30 %) que las otras, es un síntoma clásico de problemas como un cortocircuito entre espiras, un aislamiento degradado o una conexión de bobinado deficiente. Finalmente, la medición de corriente es clave para evaluar la eficiencia y diagnosticar pérdidas. Al correlacionarse con el voltaje, se calcula la potencia eléctrica de entrada, la cual, al compararse con la potencia mecánica de salida (estimada a partir de la velocidad y el par), determina la eficiencia operativa del conjunto. Una caída progresiva en esta eficiencia es un indicador claro de la degradación del sistema.

El verdadero poder diagnóstico y analítico del sistema de monitoreo propuesto no reside en el análisis aislado de cada variable, sino en la correlación cruzada de las cuatro variables. Esta relación permite desentrañar la complejidad del sistema motopropulsor. Por ejemplo, al correlacionar las señales de los sensores de efecto Hall, el voltaje y la corriente, se puede diferenciar de manera precisa entre un fallo de origen mecánico (que se manifiesta principalmente en la corriente bajo carga), uno eléctrico en el propio motor (que genera desbalances en corriente y voltaje) y uno electrónico en el controlador (evidente cuando las señales de control son correctas pero no se aplica voltaje o corriente a los devanados). Esta capacidad de diagnóstico diferencial es fundamental para un mantenimiento predictivo y eficaz.

## CAPÍTULO 3: METODOLOGÍA

### 3.1 Señales de interes del motor BLCD

El propósito de este proyecto es verificar el correcto funcionamiento de nuestro sistema de propulsión. Para ello debemos seleccionar las variables que nos daran el panorama del estado en el que se encuentra el motor para que podamos monitorear estas mismas de manera que podamos incluso diagnosticar alguna posible falla.

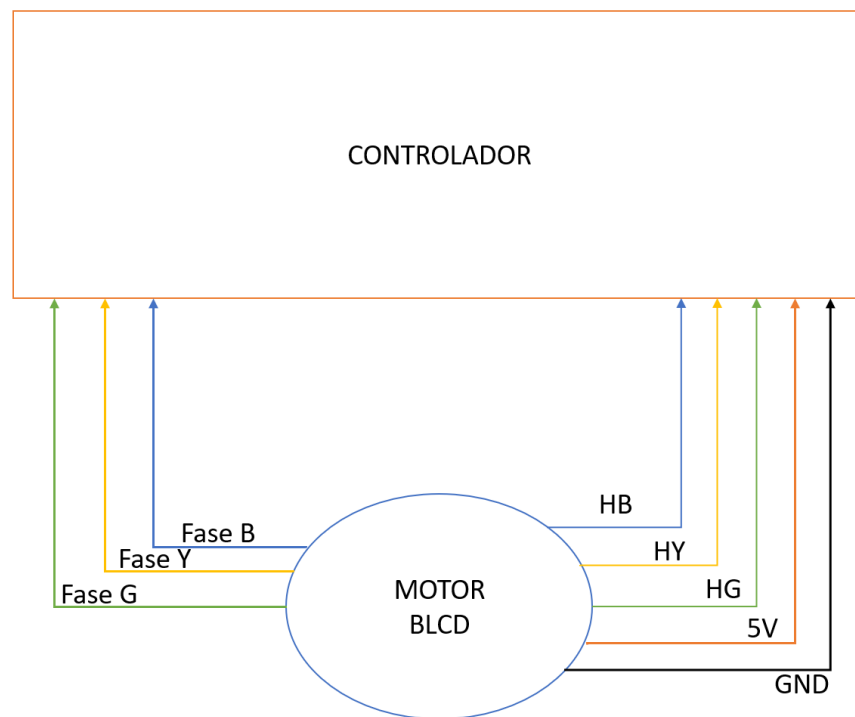
Las variables que tomaremos en cuenta para este poyecto seran las siguientes:

- **1. Señal de los sensores de efecto Hall hacia el controlador:** Estado binario (presente/ausente) de las tres señales digitales (HY, HB, HG) que indican la posición del rotor.
- **2. Detección de presencia de voltaje en las fases:** Verificación binaria (presente/ausente) de tensión en los terminales de cada devanado del motor.
- **3. Voltaje en cada fase:** Medición precisa, en voltios (V), de la tensión aplicada a cada una de las tres fases (Y, G, B) del motor.
- **4. Corriente en cada fase:** Medición precisa, en amperios (A), de la corriente que circula por cada una de las tres fases (Y, G, B) del motor.

La selección de las cuatro variables a monitorear se fundamenta en su capacidad colectiva para proporcionar una evaluación integral del estado operativo del sistema de propulsión. Este monitoreo es esencial para transitar de un esquema de mantenimiento correctivo a uno preventivo. Para tomar nuestras variables es importante ubicar en el diagrama de conexiones de la scooter los cables que son importantes para distinguir entre fases y sensores de efecto Hall junto con sus cables de alimentación, a continuación se describe la parte de interés para monitoreo.

El motor consta de 8 hilos de los cuales 3 son de un calibre mas grueso que el resto (son los que se ocupan para las fases) y 5 hilos delgados (encargados de recibir la señal de los sensores

de efecto hall y de alimentar estos mismos), mostrados en la figura 19. La denominación de colores que se asigna a las fases del motor son: verde(G), amarillo(Y) y azul(B), por lo tanto habra un par de cables de cada color siendo un cable grueso para fase y un cable delgado para el sensor de efecto hall encargado de detectar la posición del motor, eso nos resta dos cables delgados los cuales seran rojo y negro que se encargan de alimentar los sensores de efecto Hall con 5 volts. Estos cables son nuestra base para poder tomar el muestreo de nuestras señales que mas adelante seran procesadas por las Raspberry Pi Pico W.



**Figura 19: Diagrama de conexiones Motor BLCD hacia controlador**

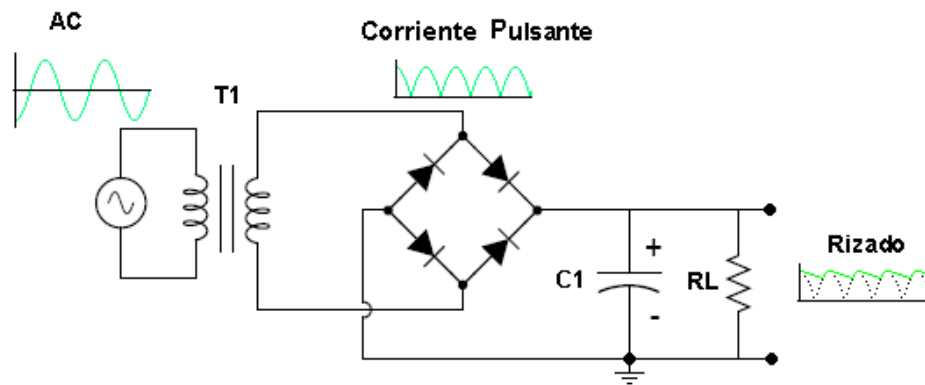
En lo siguientes pasos de la presente metodología se describe el proceso que se realizó para monitorear cada variable.

### 3.2 Configuración para monitoreo de Detección de Voltaje

Para la detección binaria de presencia/ausencia de voltaje en las fases del motor BLDC, se implementó un método de medición no invasivo mediante sensores de corriente toroidales

con bobinas secundarias. Cada una de las tres fases del motor pasa a través del núcleo de un toroide que actúa como transformador de corriente. Fue necesario implementar un circuito de acondicionamiento a este voltaje resultante pues al ser una señal alterna requirió rectificarse para poder ser compatible con el puerto ADC del microcontrolador.

Se implementó un circuito rectificador de onda completa mediante puente de diodos, seguido de un capacitor de filtro que convierte la señal alterna en una tensión continua pulsante suavizada, tal como se muestra en la figura 20. Esta señal CD resultante es proporcional a la presencia de voltaje en la fase y se encuentra dentro del rango de 0-3.3V, compatible con las entradas analógicas del microcontrolador.



**Figura 20: Circuito rectificador de onda completa tipo puente con filtro**

### 3.2.1 Arquitectura de Adquisición y Procesamiento Digital

La etapa de adquisición se implementó en una Raspberry Pi Pico W, seleccionada por su capacidad de procesamiento en sus conversores ADC integrados y conectividad WiFi. Cada fase se conectó a un canal ADC específico: fase G al GP26 (ADC0), fase B al GP27 (ADC1) y fase Y al GP28 (ADC2). El firmware desarrollado en MicroPython ejecuta un algoritmo estructurado en cuatro componentes principales:

#### Configuración y Calibración Inicial

El sistema inicia con una rutina de calibración que establece la línea base (offset) de cada canal. Durante 200 muestras consecutivas, se promedia el valor ADC con el motor en reposo para

caracterizar el ruido intrínseco del sistema. Estos valores de offset se almacenan y sustraen posteriormente de todas las mediciones, eliminando desplazamientos DC y mejorando la precisión de detección.

### **Algoritmo de Adquisición con Filtrado Digital**

Cada lectura de ADC implementa un promedio de 32 muestras consecutivas, técnica que reduce el ruido de alta frecuencia mediante filtrado de media móvil. El valor resultante se convierte a unidades de milivoltios utilizando la relación lineal tal como se muestra en la ecuación 3.1:

$$V_{mV} = \frac{Valor_{ADC} \times 3300}{4096} \quad (3.1)$$

donde 3300 mV representa el voltaje de referencia del ADC y 4096 es el rango completo del conversor de 12 bits.

### **Lógica de Detección con Histéresis**

Para evitar transiciones erróneas cerca del umbral de detección, se implementó un algoritmo con histéresis de 5 mV. El estado de detección cambia de 'ausente' a 'presente' solo cuando la señal supera los 15 mV (umbral + histéresis), mientras que la transición inversa ocurre al descender por debajo de 5 mV (umbral - histéresis). Este mecanismo elimina el efecto de parpadeo en condiciones de señal marginal.

### **Protocolo de Comunicación UDP**

El estado binario de cada fase (1: voltaje presente, 0: voltaje ausente) junto con los valores analógicos en milivoltios se encapsulan en un mensaje CSV estructurado:

PICO\_ID,MSG\_TYPE,TIEMPO\_MS,ESTADO\_G,ESTADO\_B,ESTADO\_Y,MV\_G,MV\_B,MV\_Y

Este formato permite un parsing eficiente en el sistema receptor. Los datos se transmiten cada 100 ms (10 Hz) mediante sockets UDP a una dirección IP fija preconfigurada, garantizando baja latencia en la monitorización en el tiempo.

### **Gestión de Conectividad y Robustez del Sistema**

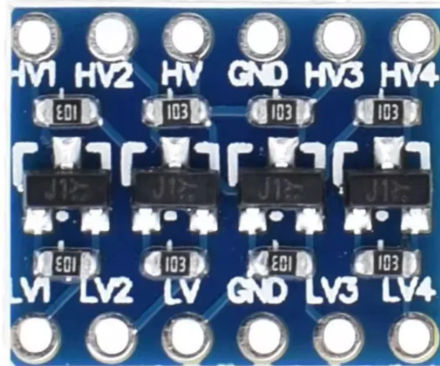
La Raspberry Pi Pico W establece conexión WiFi mediante credenciales preconfiguradas, implementando un mecanismo de timeout de 15 segundos para manejar fallos de conexión.

Los datos recibidos en la Raspberry Pi 4 (actuando como servidor UDP en el puerto 5005) se procesan y presentan en una interfaz gráfica desarrollada específicamente para esta aplicación. La estructura del mensaje CSV permite deserialización inmediata y actualización de indicadores visuales que representan el estado operativo de cada fase del motor en tiempo real, cumpliendo con el objetivo de monitorización remota del sistema de propulsión eléctrica.

### **3.3 Configuración para monitoreo en sensor de efecto Hall**

De acuerdo al diagrama, tenemos 5 hilos que van del motor hacia el controlador, 3 de los cuales son la señal de posición de las fases verde, amarilla y azul; y dos cables, uno rojo para suministrar 5 volts y uno negro para la tierra de estos. Para realizar el monitoreo sobre la presencia/ausencia de la señal de cada sensor de efecto Hall es necesario intervenir en los tres cables (Y, G y B) de manera que sea posible tomar esta misma señal y acoplarla en cada uno de los puertos que se seleccionaron para una segunda Raspberry Pi Pico W.

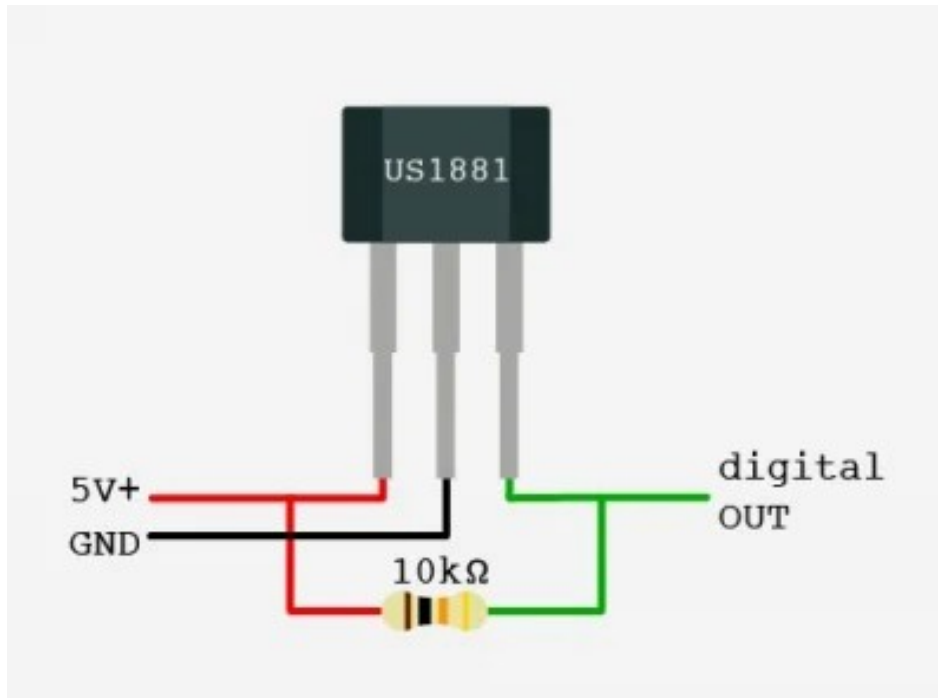
Existe una incompatibilidad fundamental de niveles de voltaje entre el dominio del sensor/controlador (5V) y las especificaciones absolutas de la Pico W (3.3V) en sus puertos GPIO. Aplicar directamente 5V a cualquier pin ADC del microcontrolador excedería su voltaje máximo de entrada, induciendo corrientes de fuga excesivas a través de los diodos de protección ESD, saturando el circuito de entrada del ADC y arriesgando un daño irreversible por sobrevoltaje. Para resolver esto, se interpone un convertidor de nivel lógico bidireccional activo.



**Figura 21: Convertidor Nivel Lógico de 5 volts - 3.3 volts**

Este convertidor realiza una adaptación de amplitud precisa y no invasiva. Traduce el 'nivel alto' de la señal de 5V a 3.3V, manteniendo inalteradas sus características esenciales: el periodo, el ciclo de trabajo y la integridad de los flancos de subida y bajada. Así, la señal que llega al puerto ADC de la Pico W se mantiene dentro de su rango operativo lineal (0-3.3V), permitiendo un muestreo fiable. El ADC digitaliza la forma de onda, y mediante software se calcula el ciclo de trabajo y la frecuencia. Un ciclo de trabajo estable y una frecuencia acorde a la velocidad del motor indican una comunicación saludable. La ausencia de transiciones (señal plana en 0V o 3.3V), una frecuencia anómala o un ciclo de trabajo fuera de rango, se interpretan como fallos de comunicación o del sensor.

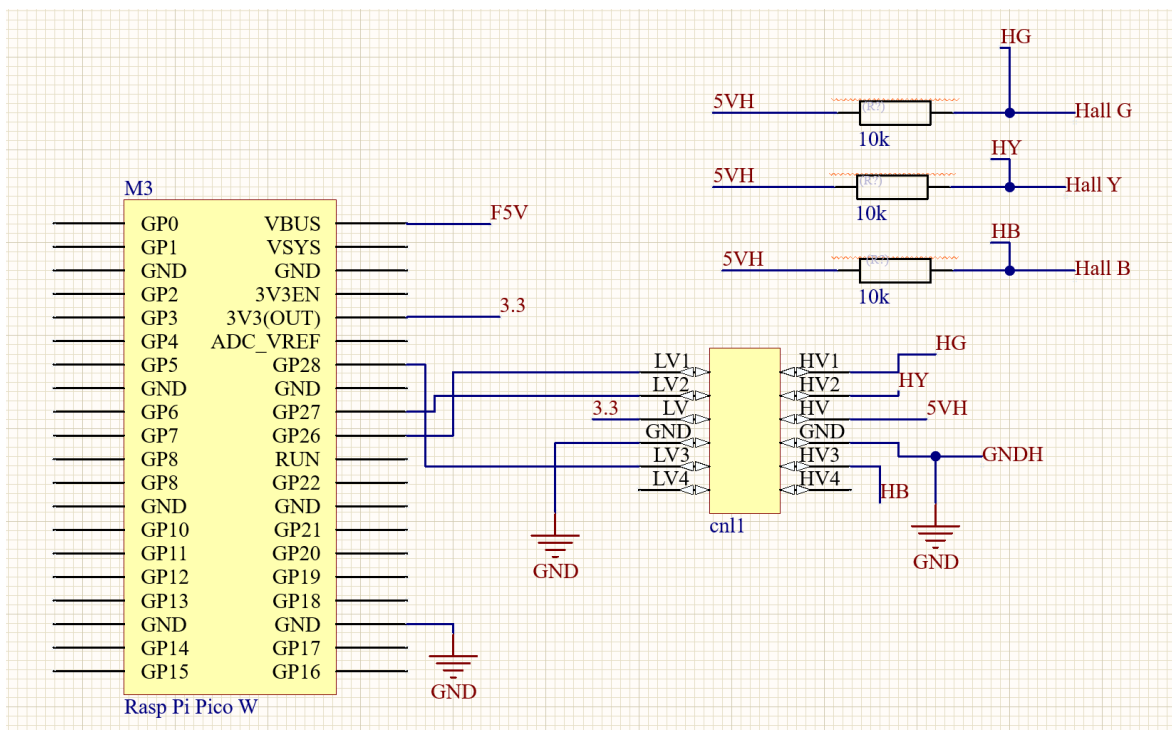
Respecto a la configuración eléctrica del sensor de efecto Hall, se respeta la topología estándar de salida open-collector. Como se describe, la salida del sensor se conecta a la tensión de alimentación de 5V a través de una resistencia pull-up de 10 kilo ohms como se muestra en la figura 22. Esta configuración garantiza que, cuando el transistor interno del sensor no conduce (estado 'off'), la salida sea elevada a 5V mediante la resistencia. Cuando el transistor conduce (estado 'on'), la salida se conecta a GND, resultando en 0V. Esta señal de 5V es la que se dirige, en paralelo, tanto al controlador principal del motor como al convertidor de nivel lógico para su monitoreo (Figura 22).



**Figura 22: Diagrama General de conexión del Sensor de Efecto Hall (El modelo es solo un ejemplo)**

Siguiendo el ejemplo de la topología para la toma de la señal del sensor de efecto Hall, la conexión queda de la siguiente manera: La señal de salida tomada de uno de los sensores se conectó con una resistencia de  $10\text{ k}\Omega$  hacia la toma de los 5 volts que proporciona el controlador para alimentar los sensores de efecto Hall, y de este nodo se conecta a la entrada de uno de los puertos del convertidor nivel lógico del apartado de los voltajes altos para que en su puerto paralelo (lado de voltajes bajos) la amplitud se limite al rango de 0-3.3 volts. En cuanto a las tierras del circuito, todas las tierras tanto de la que corresponde a la alimentación del controlador para los sensores de efecto Hall como la tierra del microcontrolador, deben convivir en el mismo nodo y de la misma manera para el convertidor nivel lógico las tierras de ambos lados (voltaje alto y bajo) deberán aterrizarse a este nodo.

En la figura 23 se muestra la configuración de las conexiones para el muestreo de estas señales descritas previamente.



**Figura 23: Diagrama General de conexión para monitoreo de señales de los sensores de efecto Hall con Raspberry Pi Pico W**

Una vez hecha la conexión se procede a realizar el código de toma de señal de los sensores de efecto Hall.

La función principal del código es detectar y transmitir el estado de comunicación de los tres sensores de efecto Hall (identificados como G, Y, B), determinando si cada sensor presenta actividad eléctrica (señales de conmutación) o se encuentra inactivo.

### 3.3.1 Arquitectura del código

El sistema se implementa en una Raspberry Pi Pico utilizando MicroPython, comenzando con la importación de bibliotecas esenciales: `network` para conectividad WiFi, `socket` para comunicación UDP, `utime` para gestión temporal y `Pin` para control de pines GPIO. La configuración inicial establece los parámetros de red (SSID y contraseña), dirección IP y puerto del servidor Raspberry Pi 4 que actuará como receptor de datos, así como identificadores únicos para el dispositivo (`PICO_ID`) y tipo de mensaje (`MSG_TYPE`).

## **Configuración de Hardware y Parámetros de Detección**

Se define la correspondencia entre los sensores de efecto Hall y los pines GPIO: GP26 para el sensor de efecto Hall G (verde), GP27 para sensor de efecto Hall Y (amarillo) y GP28 para sensor de efecto Hall B (azul), configurados como entradas con resistencias pull-down para garantizar señales estables en estado de reposo. Los parámetros de detección incluyen SEND\_EVERY\_MS (intervalo de muestreo de 100 ms equivalente a 10 Hz) y ACTIVE\_TIMEOUT\_MS (umbral de 300 ms para determinar pérdida de comunicación). Dos diccionarios mantienen el estado del sistema: `last_edge_ms` registra el último tiempo de detección de flanco por sensor, mientras `state_prev` almacena el estado anterior para detección de transiciones.

## **Mecanismo de Conexión y Comunicación**

La función `wifi_connect()` gestiona la conexión WiFi en modo estación (por sus siglas en inglés: STA), implementando un timeout de 15 segundos y retornando la configuración de red una vez establecida la conexión. Este proceso es fundamental para garantizar la comunicación bidireccional con el sistema central de adquisición de datos.

## **Sistema de Interrupciones para Detección de Flancos**

El núcleo de la detección reside en el sistema de interrupciones por hardware (IRQ). La función `make_irq_handler()` genera manejadores de interrupción específicos para cada sensor, capturando tanto flancos de subida (RISING) como de bajada (FALLING). Cuando ocurre cualquier transición eléctrica, el manejador correspondiente actualiza la marca temporal en `last_edge_ms` para ese sensor específico. Esta implementación mediante interrupciones permite una detección precisa independientemente del ciclo de muestreo principal, minimizando el tiempo de respuesta a eventos.

## **Algoritmo de Determinación de Estado**

La función `hall_active()` implementa la lógica de determinación de estado mediante comparación temporal. Calcula la diferencia entre el tiempo actual y la última detección de flanco, clasificando el sensor como 'activo' (con comunicación) si esta diferencia no supera el

umbral `ACTIVE_TIMEOUT_MS`. Esta metodología basada en tiempo en lugar de umbrales de voltaje fijos permite adaptarse a variaciones en la velocidad del motor y condiciones operativas.

### **Bucle Principal y Protocolo de Comunicación**

El bucle principal implementa un sistema de transmisión periódica tipo heartbeat. Cada 100 ms, construye un mensaje UDP con formato estructurado:

`PICO_ID,MSG_TYPE,timestamp,estado_G,estado_Y,estado_B,`

donde los estados se codifican como 1 (comunicación presente) o 0 (comunicación ausente). Este protocolo minimiza el tráfico de red mientras mantiene actualizado al sistema central. Adicionalmente, el código implementa logging condicional que solo muestra cambios de estado, reduciendo la salida de depuración mientras mantiene trazabilidad de eventos críticos.

## **3.4 Configuración para Monitoreo de Voltaje de Fases**

### **3.4.0.1 Conexión Fase-Fase para Alimentación del ZMPT101B**

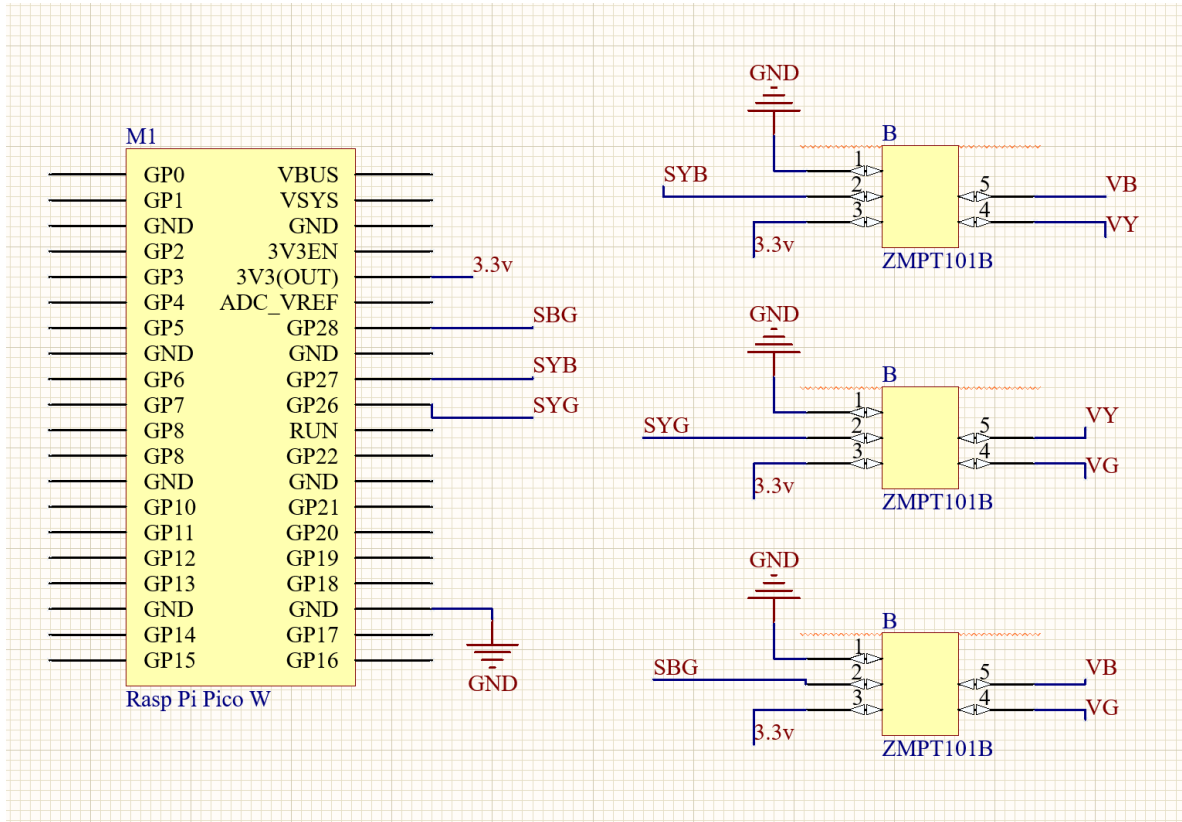
Para la medición de voltajes entre fases del motor BLDC, se implementó una configuración de tres sensores ZMPT101B que permiten medir los voltajes CA entre cada par de fases. Las conexiones establecidas fueron:

Canal YG:  $V_Y - V_G \rightarrow \text{ZMPT101B-1} \rightarrow \text{GPIO26 (ADC0)}$

Canal YB:  $V_Y - V_B \rightarrow \text{ZMPT101B-2} \rightarrow \text{GPIO27 (ADC1)}$

Canal BG:  $V_B - V_G \rightarrow \text{ZMPT101B-3} \rightarrow \text{GPIO28 (ADC2)}$

Esta configuración permite reconstruir el perfil completo de voltajes trifásicos y detectar desbalances o fallas en la conmutación del inversor. En la figura [24](#) se muestra el esquema de conexión hacia microcontrolador.



**Figura 24: Diagrama General de conexión para monitoreo de voltajes entre fases con Raspberry Pi Pico W**

### 3.4.1 Algoritmo de Adquisición y Procesamiento

#### 3.4.1.1 Estrategia de Muestreo para Alta Frecuencia (300Hz)

A pesar de que el ZMPT101B está especificado para 50-60Hz, se implementó un algoritmo de muestreo adaptado para 300Hz mediante:

$$f_{\text{muestreo}} = 12 \text{ kHz} \Rightarrow T_{\text{muestreo}} = 83.3 \mu\text{s} \quad (3.2)$$

Para cada bloque de medición:

$$N_{\text{muestras}} = 480 \Rightarrow T_{\text{bloque}} = \frac{N_{\text{muestras}}}{f_{\text{muestreo}}} = 40 \text{ ms} \quad (3.3)$$

Este período cubre exactamente:

$$n_{\text{ciclos}} = T_{\text{bloque}} \times f_{\text{señal}} = 0.04 \times 300 = 12 \text{ ciclos completos} \quad (3.4)$$

### 3.4.1.2 Cálculo del Valor RMS por Bloque

El algoritmo implementa un método de dos pasos para calcular el valor RMS verdadero:

**Paso 1: Cálculo del valor medio (DC offset)**

$$\bar{D} = \frac{1}{N} \sum_{i=1}^N D[i] \quad (3.5)$$

donde  $D[i]$  es el valor digital del ADC en la muestra  $i$ .

**Paso 2: Cálculo del valor RMS de la componente AC**

$$V_{\text{RMS}} = \frac{V_{\text{ref}}}{2^{16} - 1} \times \sqrt{\frac{1}{N} \sum_{i=1}^N (D[i] - \bar{D})^2} \quad (3.6)$$

### 3.4.1.3 Calibración y Compensación de Errores

**Procedimiento de calibración de baseline:**

1. Con el motor en reposo, se adquieren  $M = 8$  bloques de 480 muestras
2. Para cada canal  $c$ :

$$B_c = \frac{1}{M} \sum_{j=1}^M V_{\text{RMS},c}[j] \quad (3.7)$$

3. Este valor  $B_c$  representa el ruido intrínseco del sensor y se almacena como BASELINE\_SENSOR\_RMS

**Compensación durante operación normal:**

$$V_{\text{clean},c} = \sqrt{\text{máx}(V_{\text{raw},c}^2 - B_c^2, 0)} \quad (3.8)$$

$$V_{\text{primary},c} = V_{\text{clean},c} \times K_c \quad (3.9)$$

donde  $K_c$  es el factor de calibración empírico determinado experimentalmente.

#### 3.4.1.4 Filtrado Digital IIR para Suavizado

Se implementa un filtro de primer orden para reducir variaciones de alta frecuencia:

$$V_{\text{filtrado}}[n] = \alpha \times V_{\text{medido}}[n] + (1 - \alpha) \times V_{\text{filtrado}}[n - 1] \quad (3.10)$$

con constante de suavizado  $\alpha = 0.25$ , lo que equivale a una frecuencia de corte de:

$$f_c = \frac{\alpha}{2\pi T_s} \approx 19.9 \text{ Hz} \quad \text{para } T_s = 2 \text{ ms} \quad (3.11)$$

#### 3.4.1.5 Transmisión de Datos y Protocolo

El formato de mensaje UDP implementado es:

PICO\_ID,MSG\_TYPE,tiempo\_ms,rawYG,rawYB,rawBG,filtYG,filtYB,filtBG

Ejemplo concreto:

P4,ZM,123456789,57.32,56.89,57.15,57.28,56.95,57.08

La transmisión se realiza a intervalo fijo de  $T_{\text{envío}} = 100 \text{ ms}$  (10Hz), sincronizada con el sistema de detección de presencia de voltaje.

#### 3.4.2 Consideraciones de Precisión y Limitaciones

1. **Error sistemático por frecuencia:** La respuesta en frecuencia del ZMPT101B disminuye aproximadamente 0.5 % por cada 10Hz sobre 60Hz, requiriendo factores de corrección empíricos.
2. **Resolución efectiva:** Considerando ruido de cuantización y filtrado, la resolución efectiva es:

$$\Delta V_{\text{min}} \approx \frac{3.3V}{65535} \times K_c \times \sqrt{N} \approx 0.5V \quad (3.12)$$

3. **Latencia total del sistema:** Desde adquisición hasta visualización:

$$\tau_{\text{total}} = 40 \text{ ms (adquisición)} + 100 \text{ ms (envío)} \approx 140 \text{ ms} \quad (3.13)$$

Esta metodología permite monitorear los voltajes de fase del motor BLDC con precisión adecuada para diagnóstico y control, a pesar de las limitaciones de frecuencia del sensor ZMPT101B.

### 3.5 Configuración para Monitoreo de Corriente de Fases

#### 3.5.1 Configuración del Sistema de Medición

##### 3.5.1.1 Conexión Serie de los Módulos ACS712

Para la medición de corriente en cada fase del motor BLDC, se implementó una configuración de tres módulos ACS712-30A conectados en serie con cada fase del motor. La conexión se realizó de la siguiente manera:

Fase G: Cable de fase  $\rightarrow$  ACS712-1 (IP+ a IP-)  $\rightarrow$  GPIO26 (ADC0)

Fase B: Cable de fase  $\rightarrow$  ACS712-2 (IP+ a IP-)  $\rightarrow$  GPIO27 (ADC1)

Fase Y: Cable de fase  $\rightarrow$  ACS712-3 (IP+ a IP-)  $\rightarrow$  GPIO28 (ADC2)

#### Consideraciones críticas de la conexión:

1. **Orientación de la corriente:** Los terminales IP+ e IP- deben conectarse en serie con la fase, respetando la dirección de flujo de corriente para que la polaridad del voltaje de salida sea correcta.
2. **Aislamiento de potencia:** Cada módulo ACS712 incorpora aislamiento galvánico entre los pines de corriente (IP+, IP-) y la electrónica de medición.

En la figura 25 se muestra el esquema de conexiones de los sensores ACS712 hacia el micro-controlador.



donde:

- $V_{\text{offset}} = 2.5V$  (para alimentación de 5V)
- $S = 0.066V/A$  para el modelo de 30A
- $I_{AC/DC}$  es la corriente medida (positiva o negativa)

Para corriente alterna, el voltaje de salida oscila alrededor de 2.5V con amplitud proporcional al valor RMS de la corriente.

### 3.5.2.2 Estrategia de Muestreo para Cálculo RMS

El algoritmo implementa un método de muestreo sincronizado para calcular el valor RMS de la corriente:

$$f_{\text{muestreo}} = \frac{1}{200\mu s} = 5 \text{ kHz} \quad (3.15)$$

Para cada bloque de medición:

$$N_{\text{muestras}} = 250 \Rightarrow T_{\text{bloque}} = N \times T_s = 250 \times 200\mu s = 50 \text{ ms}$$

Este período cubre:

$$n_{\text{ciclos}} = T_{\text{bloque}} \times f_{\text{señal}} = 0.05 \times 300 = 15 \text{ ciclos completos}$$

### 3.5.2.3 Calibración de Offset (Punto Cero)

Antes de iniciar las mediciones, se ejecuta una rutina de calibración para determinar el voltaje de offset de cada canal:

$$V_{\text{offset},c} = \frac{1}{M} \sum_{k=1}^M V_{\text{ADC},c}[k] \quad \text{con } I_c = 0 \quad (3.16)$$

donde  $M = 400$  muestras y  $I_c = 0$  significa motor sin corriente (o desconectado).

### 3.5.2.4 Cálculo del Valor RMS de Corriente

Para cada bloque de  $N$  muestras, se calcula el valor RMS de la corriente mediante:

1. **Conversión ADC a voltaje:**

$$V_{\text{ADC}}[i] = \frac{D[i]}{4095} \times V_{\text{ref}} \quad (3.17)$$

2. **Eliminación del offset:**

$$V_{\text{signal}}[i] = V_{\text{ADC}}[i] - V_{\text{offset}} \quad (3.18)$$

3. **Cálculo del valor RMS del voltaje:**

$$V_{\text{RMS}} = \sqrt{\frac{1}{N} \sum_{i=1}^N V_{\text{signal}}[i]^2} \quad (3.19)$$

4. **Conversión a corriente RMS:**

$$I_{\text{RMS}} = \frac{V_{\text{RMS}}}{S} \quad \text{donde } S = 0.066 \text{ V/A} \quad (3.20)$$

### 3.5.2.5 Filtrado de Mediana para Rechazo de Picos

Para eliminar picos transitorios causados por ruido de conmutación, se implementa un filtro de mediana de ventana 5:

$$I_{\text{filtrado}}[n] = \text{mediana}\{I_{\text{RMS}}[n-4], I_{\text{RMS}}[n-3], \dots, I_{\text{RMS}}[n]\} \quad (3.21)$$

El filtro de mediana implementado demuestra un desempeño excepcional en el entorno dinámico y ruidoso característico de los sistemas de potencia del motor BLDC. Su principio de operación le permite, en primer lugar, rechazar de forma robusta aquellos pulsos eléctricos aislados y de corta duración que no corresponden a la señal real, sino a interferencias o ruido de conmutación. En segundo lugar, posee la inteligente capacidad de preservar los bordes nítidos de las transiciones legítimas en la señal, asegurando que los cambios de estado reales

del motor no se suavicen o pierdan. Como tercer beneficio fundamental, mantiene una notable estabilidad en el valor medido, evitando las fluctuaciones erráticas que podrían confundir al operador o a un sistema de control automático. En conjunto, estas características lo convierten en un guardián digital ideal, que purifica la señal sin distorsionar su verdadero mensaje, proporcionando así una lectura clara y confiable de la corriente incluso bajo condiciones operativas adversas.

Implementación del filtro de mediana:

```
def median_filter(value: float, buf: list, win: int) -> float:
    buf.append(value)
    if len(buf) > win:
        buf.pop(0)
    s = sorted(buf)
    return s[len(s) // 2]
```

### 3.5.2.6 Transmisión de Datos y Protocolo UDP

El formato de mensaje transmitido es:

PICO\_ID,MSG\_TYPE,tiempo\_ms,rawG,rawB,rawY,filtG,filtB,filtY

Ejemplo:

P3,AC,987654321,12.345,12.210,12.401,12.350,12.220,12.380

La transmisión ocurre cada  $T_{\text{envío}} = 150 \text{ ms}$  (6.67Hz), optimizando el uso de ancho de banda WiFi mientras mantiene una representación adecuada de la dinámica del sistema.

### 3.5.3 Consideraciones de Precisión y Limitaciones

#### 3.5.3.1 Errores Introducidos por el Sistema de Adquisición

1. **Error por muestreo asíncrono:** Para una señal de 300Hz muestreada a 5 kHz, el error máximo por aliasing es despreciable.

2. **Error por resolución temporal:** La ventana de muestreo de 50ms introduce un error de:

$$\varepsilon_{\text{temporal}} \approx \frac{1}{2 \times f_{\text{señal}} \times T_{\text{bloque}}} = \frac{1}{2 \times 300 \times 0.05} = 0.033 = 3.3 \% \quad (3.22)$$

Esta metodología permite monitorear las corrientes de fase del motor BLDC con precisión adecuada para diagnóstico y control, proporcionando datos esenciales para el análisis del desempeño y eficiencia del sistema de propulsión eléctrica.

## 3.6 Interfaz de Usuario en Raspberry Pi 4

### 3.6.1 Arquitectura del Sistema de Visualización

La interfaz de usuario se desarrolló en Python utilizando el framework PyQt5, ejecutándose en una Raspberry Pi 4 con sistema operativo Raspberry Pi OS. La arquitectura implementada sigue un patrón de diseño Modelo-Vista-Controlador (MVC) modificado, donde:

- **Modelo:** Los datos recibidos via UDP desde los microcontroladores Pico W
- **Vista:** Los widgets personalizados de PyQt5 y las gráficas en tiempo real
- **Controlador:** Los manejadores de eventos y temporizadores que actualizan la vista

El sistema se diseñó para mostrar simultáneamente cuatro categorías de datos:

1. Estados de presencia de voltaje en fases (Pico 1)
2. Estados de sensores de efecto Hall (Pico 2)
3. Corrientes RMS por fase (Pico 3)
4. Voltajes fase-fase RMS (Pico 4)

### 3.6.2 Configuración del Sistema de Comunicación UDP

#### 3.6.2.1 Inicialización del Socket UDP

La interfaz se configura como servidor UDP que escucha en el puerto 5005 para recibir datos de todos los microcontroladores. La implementación utiliza la clase QUdpSocket de PyQt5:

```
self.udp = QUdpSocket(self)
ok = self.udp.bind(BIND_ADDR, UDP_PORT)
if not ok:
    self.statusBar().showMessage(f"ERROR: no se pudo abrir UDP en puerto {UDP_PORT}")
else:
    self.statusBar().showMessage(f"Escuchando UDP en {PI4_IP}:{UDP_PORT}")
```

#### Parámetros de configuración:

- BIND\_ADDR = QHostAddress.AnyIPv4: Escucha en todas las interfaces de red (0.0.0.0)
- UDP\_PORT = 5005: Puerto estándar para la comunicación
- PI4\_IP = "192.168.137.33": Dirección IP estática de la Raspberry Pi 4

#### 3.6.2.2 Mecanismo de Recepción y Procesamiento

El núcleo de la comunicación del sistema se basa en un mecanismo de cola de mensajes asíncrona y eficiente, implementado mediante el paradigma de señales y slots que ofrece el framework PyQt5. Este diseño permite que la interfaz gráfica permanezca receptiva mientras maneja el flujo constante de datos de los microcontroladores. La cadena de procesamiento sigue una ruta bien definida: cuando el socket UDP detecta datos nuevos, emite la señal readyRead(), que está conectada al método on\_udp\_ready(). Este método, a su vez, se encarga de invocar la función parse\_message() para descifrar y actuar sobre la información recibida. Este flujo puede representarse como:

QUdpSocket.readyRead → on\_udp\_ready → parse\_message

El proceso comienza con la inicialización del socket UDP en el puerto 5005, configurado para escuchar en todas las interfaces de red disponibles (0.0.0.0), con un tiempo de espera (timeout) de 100 milisegundos para mantener la capacidad de respuesta. Una vez operativo, el sistema entra en un bucle de recepción principal que verifica continuamente la presencia de datagramas pendientes en el búfer de red. Por cada datagrama disponible, se ejecuta una rutina de decodificación que lee el paquete completo, convierte la secuencia de bytes a una cadena de texto con codificación UTF-8 y procede a dividir este mensaje en líneas individuales, ya que un mismo datagrama puede contener múltiples lecturas.

Cada línea de datos recibida es entonces procesada de manera individual. El primer paso es separar los campos utilizando la coma como delimitador, extrayendo la información básica que estructura cualquier mensaje: el identificador único del microcontrolador emisor (P1, P2, P3, P4), el tipo de variable que reporta (VD para voltaje, HL para sensores de efecto Hall, AC para corriente, ZM para voltaje fase-fase) y la marca de tiempo en milisegundos que indica el momento de la medición. Con estos datos clasificados, el sistema realiza un despacho inteligente a rutinas específicas.

Si el mensaje proviene del Pico 1 (P1-VD), el sistema parsea los estados booleanos que indican la presencia o ausencia de voltaje en las fases G, B e Y, actualizando inmediatamente los indicadores LED y las etiquetas de texto correspondientes en la pestaña designada. Para los mensajes del Pico 2 (P2-HL), se sigue un procedimiento análogo, pero interpretando los estados de los sensores de efecto Hall. Los mensajes de naturaleza analógica, como los del Pico 3 (P3-AC) con corrientes RMS y los del Pico 4 (P4-ZM) con voltajes fase-fase, requieren un tratamiento más complejo. En estos casos, el sistema parsea tanto los valores en crudo (RAW) como los valores filtrados, actualizando simultáneamente las gráficas de tendencia y los displays numéricos en sus respectivas pestañas, lo que proporciona una visión dual e inmediata del comportamiento del motor.

Finalizada la actualización de los datos, el sistema registra un nuevo \*timestamp\* de recepción para el microcontrolador correspondiente. Este registro es crucial para el siguiente paso: la evaluación del estado de conexión. Un mecanismo de verificación periódico, que funciona como un latido del corazón (\*heartbeat\*), compara constantemente el tiempo actual con el

del último mensaje recibido. Un dispositivo se considera *\*online\** y su indicador se pinta de verde, sólo si ha enviado datos dentro de los últimos 1200 milisegundos. De lo contrario, se marca como *\*offline\** y su indicador cambia a rojo, alertando de una posible desconexión o falla. Este criterio se define mediante la siguiente lógica:

$$\text{Online} = \begin{cases} \text{True} & \text{si } (t_{\text{actual}} - t_{\text{último\_RX}}) \leq 1200 \text{ ms} \\ \text{False} & \text{en caso contrario} \end{cases}$$

La implementación de esta verificación se realiza en una función concisa que recorre la lista de todos los Picos, como se muestra a continuación:

```
def update_online_states(self):
    now = ms_now()
    for pico_id in ["P1", "P2", "P3", "P4"]:
        last = self.last_rx.get(pico_id)
        online = (last is not None) and ((now - last) <= OFFLINE_TIMEOUT_MS)
        actualizar_badge(pico_id, online)
        actualizar_icono_tab(pico_id, online)
```

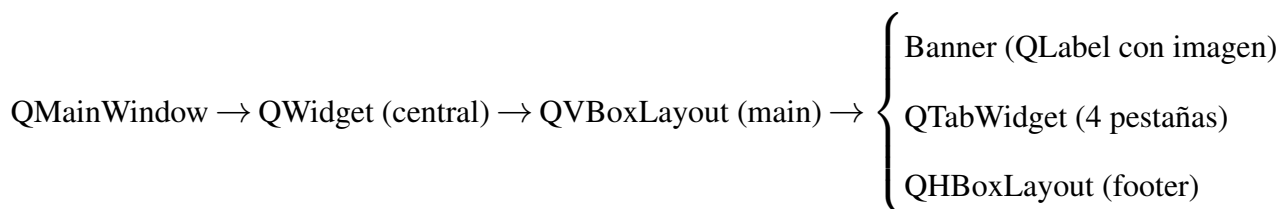
Todo este procesamiento se ejecuta manteniendo un control estricto del tiempo, asegurando que la frecuencia de actualización sea estable, que el hilo principal de la interfaz no se bloquee y que la experiencia del usuario siga siendo fluida.

### 3.6.3 Diseño y Configuración de la Interfaz Gráfica

#### 3.6.3.1 Estructura de Ventana Principal

La interfaz de usuario se construyó sobre una jerarquía clara y organizada de componentes PyQt5, diseñada para presentar una gran cantidad de información de forma intuitiva. En la cúspide se encuentra la ventana principal ('QMainWindow'), que actúa como contenedor de todos los demás elementos. En su interior, un widget central ('QWidget') organiza el espacio mediante un diseño vertical principal ('QVBoxLayout'). Este layout dispone, en

orden descendente, un banner superior con la imagen institucional, un panel de pestañas (‘QTabWidget’) que agrupa la información por microcontrolador, y finalmente una barra de pie de página (‘QHBoxLayout’) con detalles institucionales y el estado de la red. Esta estructura puede visualizarse como:



Los componentes clave de esta estructura incluyen: un banner superior que carga una imagen PNG institucional; un reloj digital en la barra de título que se actualiza cada 500 milisegundos mediante un temporizador (‘QTimer’); cuatro pestañas independientes, una dedicada a cada Raspberry Pi Pico W (P1 a P4), que concentran las variables monitoreadas; y un pie de página que muestra información contextual y el estado de la conexión UDP.

### 3.6.3.2 Sistema de Estilos QSS

Para conferirle una apariencia profesional, moderna y oscura—ideal para sesiones prolongadas de monitoreo—se implementó un sistema de estilos completo utilizando Qt Style Sheets (QSS). Este lenguaje permite definir propiedades visuales de manera similar a CSS para la web. Se definió una paleta de colores basada en tonos azules oscuros y grises, con acentos verdes y rojos para los estados, creando un alto contraste y una excelente legibilidad. El estilo base aplicado a toda la aplicación se define como:

```
APP_STYLE = ""
QMainWindow { background: #0b1020; }
QLabel { color: #e5e7eb; }
QTabWidget::pane {
    border: 1px solid #1f2a44;
    border-radius: 12px;
```

```

        background: #0b1020;
    }
    QTabBar::tab {
        background: #0f1a33;
        color: #cbd5e1;
        border: 1px solid #1f2a44;
        padding: 10px 14px;
        border-radius: 10px;
    }
}
"""

```

La paleta de colores resultante utiliza #0b1020 (azul noche muy oscuro) para el fondo principal, #0f1833 para las tarjetas de datos con bordes #1f2a44, y una gama de grises desde el #f8fafc (casi blanco) hasta el #94a3b8 para el texto. Los estados de operación se destacan con #22c55e (verde brillante) para *\*online\** y #ef4444 (rojo) para *\*offline\**.

### 3.6.3.3 Widgets Personalizados

Para satisfacer las necesidades específicas de visualización de este sistema de monitoreo, fue necesario desarrollar widgets gráficos personalizados que no forman parte de la biblioteca estándar de PyQt5. El primero de ellos es el 'StatusBadge', una etiqueta inteligente que funciona como un semáforo de estado. Cambia dinámicamente su texto y color de fondo entre 'ONLINE' en verde y 'OFFLINE' en rojo, proporcionando una indicación instantánea y clara del estado de cada dispositivo.

```

class StatusBadge(QLabel):
    def set_online(self, online: bool):
        if online:
            self.setText("ONLINE")
            self.setStyleSheet("background: #064e3b; color: #ecfdf5;")
        else:

```

```

self.setText("OFFLINE")

self.setStyleSheet("background: #3f1d1d; color: #fee2e2;")

```

El 'LedIndicator' es un widget gráfico minimalista que simula un LED físico. Es un marco circular cuyo color de relleno cambia entre verde (#22c55e) y rojo (#ef4444) según un estado booleano, siendo utilizado para representar de la forma más intuitiva posible la presencia o ausencia de una señal en una fase o sensor de efecto Hall.

```

class LedIndicator(QFrame):
    def set_state(self, ok: bool):
        color = "#22c55e" if ok else "#ef4444"
        self.setStyleSheet(f"background: {color}; border-radius: {self.d//2}px;")

```

El widget más especializado es el 'LinePlot', una gráfica ligera desarrollada desde cero para visualizar series temporales sin depender de librerías externas pesadas. Utiliza un búfer circular que almacena un máximo de 220 puntos, lo que permite visualizar aproximadamente 22 segundos de datos a una frecuencia de actualización de 10 Hz. Su algoritmo de dibujo es eficiente y realiza autoescalado, calculando los valores mínimo y máximo de la serie en cada ciclo para adaptar la visualización. Las coordenadas de cada punto se calculan mediante las ecuaciones:

$$x[i] = left + \frac{plot\_width \times i}{n - 1} \quad (3.23)$$

$$y[i] = top + plot\_height \times \left(1 - \frac{value[i] - min}{max - min}\right) \quad (3.24)$$

Donde plot\_width y plot\_height son las dimensiones del área de dibujo, e i es el índice de cada punto en el búfer de n elementos. Este método garantiza que la gráfica se dibuje de manera suave y con un uso óptimo del espacio disponible.

### **3.6.4 Mecanismo de Actualización en el Tiempo**

#### **3.6.4.1 Sistema de Temporizadores**

La ilusión de un monitoreo en tiempo continuo y fluido se consigue mediante la precisa coordinación de múltiples temporizadores (QTimer), cada uno con una misión específica y una frecuencia cuidadosamente calibrada. El temporizador 't\_clock', configurado con un intervalo de 500 milisegundos (2 Hz), es el encargado de actualizar el reloj digital que se muestra en la interfaz, utilizando la función `time.strftime(" %H: %M: %S")` de Python para obtener la hora actual del sistema.

Un temporizador aún más rápido, `t_online`, se ejecuta cada 250 milisegundos (4 Hz). Su tarea crítica es invocar la función `update_online_states()` que verifica, para cada uno de los cuatro microcontroladores, si su último latido (mensaje recibido) está dentro del umbral de los 1200 milisegundos definidos. Si se excede este tiempo, dispara una cascada de acciones: cambia la etiqueta de estado a 'OFFLINE', pinta el StatusBadge de rojo y modifica el icono de la pestaña correspondiente, ofreciendo una alerta visual inmediata.

La actualización de las gráficas LinePlot opera en un régimen de frecuencia diferente, gobernado por el ciclo de refresco de la interfaz gráfica. Cada vez que se recibe un nuevo dato y se añade al búfer circular, se solicita un repintado ('update()'). PyQt5 sincroniza estos repintados con el refresco de pantalla, intentando alcanzar una frecuencia de hasta 60 Hz (cada 16.67 ms) para lograr animaciones suaves, aunque siempre limitado por la llegada de nuevos datos. Finalmente, el procesamiento mismo de los mensajes UDP no está controlado por un temporizador periódico, sino que es un proceso dirigido por eventos. La señal `readyRead()` del socket se emite únicamente cuando llegan datos nuevos, lo que hace que este temporizador 'virtual' opere a una frecuencia variable, directamente determinada por el ancho de banda de la red WiFi y la tasa de envío de los Picos.

#### **3.6.4.2 Actualización de Widgets**

La llegada de un nuevo datagrama UDP desencadena una secuencia metódica de pasos diseñada para actualizar la interfaz de la manera más eficiente posible. Tras decodificar el

mensaje y extraer su identificador y tipo, el sistema bifurca su lógica. Para mensajes binarios (VD, HL), la ruta es directa: se obtiene la referencia al widget ‘LedIndicator’ correspondiente a la fase o sensor específico, se establece su estado (verdadero para "SEÑAL PRESENTE", falso para "SIN SEÑAL"), y se solicita su repintado inmediato. Simultáneamente, se actualiza el texto descriptivo adyacente al LED para reforzar la información visual.

Para los mensajes analógicos (AC, ZM), el proceso es más rico. Cada valor recibido—tanto en crudo (RAW) como filtrado—se envía al método ‘push()’ de su correspondiente widget ‘LinePlot’. Este método realiza tres tareas fundamentales: añade el nuevo valor al final del búfer circular (desplazando el más antiguo si se alcanza el límite de 220 puntos), recalcula los valores mínimo y máximo globales de la serie para el autoescalado, y marca el área del widget como "sucio" para forzar un repintado en el próximo ciclo gráfico. En paralelo, el valor numérico actual se formatea con dos decimales y se actualiza en la etiqueta de texto sobre la gráfica.

El acto de repintado de la gráfica, gestionado por el método `paintEvent()`, es donde la abstracción de datos se convierte en píxeles. El método comienza calculando las coordenadas en pantalla para cada punto almacenado en el búfer, utilizando las fórmulas de mapeo previamente descritas. Luego, dibuja un tenue grid de referencia, traza una línea poligonal que conecta todos los puntos en orden, y finalmente, renderiza en las esquinas los valores mínimo y máximo actuales de la serie, dando contexto numérico inmediato a la forma visual de la curva.

#### **3.6.4.3 Gestión de Recursos**

Optimizar el rendimiento en un hardware embebido como la Raspberry Pi 4 es crucial para una experiencia de usuario fluida. Se implementaron varias estrategias clave. El \*doble buffering\* se utiliza durante el dibujado de las gráficas; esta técnica consiste en renderizar la escena completa en un búfer de memoria en segundo plano antes de mostrarla en pantalla, eliminando parpadeos y haciendo que las transiciones sean suaves. Para evitar el consumo desmedido de memoria, cada gráfica ‘LinePlot’ tiene un límite estricto de 220 puntos en su búfer circular, lo que representa un compromiso óptimo entre historial útil (unos 22 segundos a 10 Hz) y

responsividad del sistema. La lógica de actualización es *\*selectiva\**; en lugar de repintar la interfaz completa constantemente, sólo se invalidan y redibujan aquellos widgets que han recibido nuevos datos, liberando ciclos de CPU. Finalmente, el uso de la estructura de datos ‘deque’ (cola de doble extremo) para almacenar las series temporales garantiza que las operaciones de añadir nuevos puntos y eliminar los más antiguos se realicen en tiempo constante, independientemente del número de puntos almacenados, lo que es fundamental para mantener una alta frecuencia de actualización.

### 3.6.5 Integración del Sistema Completo

La ejecución del sistema de monitoreo se inicia mediante un script principal en la Raspberry Pi 4. Este script, al ser ejecutado, crea una instancia de la aplicación PyQt5 (QApplication), le aplica la hoja de estilos QSS definida, instancia la ventana principal (MainWindow) y entra en el bucle de eventos principal (app.exec\_()), que mantiene la interfaz activa y receptiva hasta que el usuario decide cerrarla.

```
if __name__ == "__main__":
    app = QApplication(sys.argv)
    app.setStyleSheet(APP_STYLE)
    w = MainWindow()
    w.show()
    sys.exit(app.exec_())
```

Los requisitos del sistema para ejecutar esta interfaz de manera óptima son accesibles: Python en su versión 3.7 o superior, con la librería PyQt5 instalada (típicamente mediante ‘pip install pyqt5’); una Raspberry Pi 4 con al menos 2 GB de RAM; una instalación estándar de Raspberry Pi OS, ya sea de 32 o 64 bits; y una resolución de pantalla mínima de 1280×720 píxeles para acomodar cómodamente todos los paneles de control y gráficas.

## CAPÍTULO 4: RESULTADOS

### 4.1 Cableado del motor y estado de la scooter

La motocicleta eléctrica Keeway Ride E1 se encontrara en estado estático ya que al momento de realizar el proceso de este proyecto se encuentra sin una bateria disponible, a pesar de ello, el proyecto se realiza teniendo conectado el cargador a la toma de corriente. Para realizar esto, se toma en cuenta que el cargador de la scooter solo admite la toma de una corriente de 220 VCA, pero nuestra toma es solo a 110 VCA monofásica, por lo tanto se usara un convertidor de tensión CA/CA donde a su entrada se conectara los 110 VCA de la toma de corriente y a su salida de 220 VCA se conectara el cargador de la scooter.



**Figura 26: Convertidor CA/CA para cargador de la scooter**

De esta manera la scooter funcionará teniendola conectada a su cargador que cuenta con las siguientes especificaciones:

-Entrada: 220-240 VCA 50/60Gz 1.5A

-Salida: 60v 3A

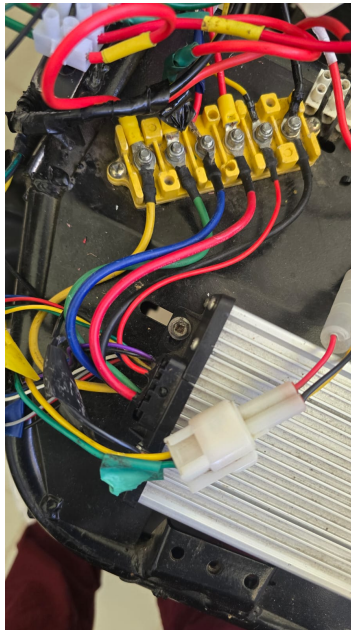


**Figura 27: Cargador de la Scooter Keeway E1**

El monitoreo que se realice en este proyecto sera bajo estas condiciones, la scooter se mantendra conectada a su cargador, trabajando a 60 volts de corriente directa a 3 amperes.

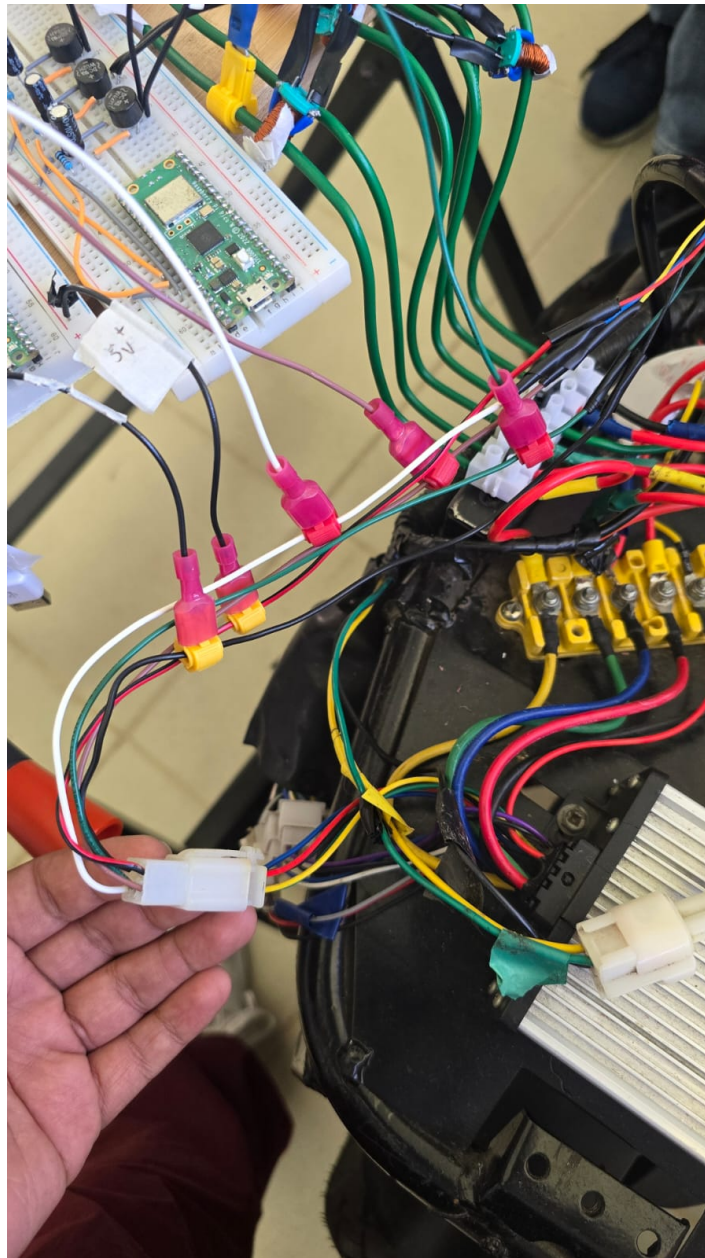
#### **4.1.1 Conexiones del motor hacia el controlador**

A continuación se muestra el cableado del motor en la figura 28:



**Figura 28: Cableado de fases provenientes del motor hacia el controlador (azul, verde y amarillo)**

Dentro de la misma claqueta (mostrada en la figura 29) vemos un cable rojo y uno negro del mismo calibre que el de las fases, estos dos cables corresponden a la alimentación proveniente del cargador hacia el controlador (en caso de la existencia de la batería, se alimenta de la misma manera de ambos cables), siendo el rojo los 60 volts y el negro la tierra (GND).



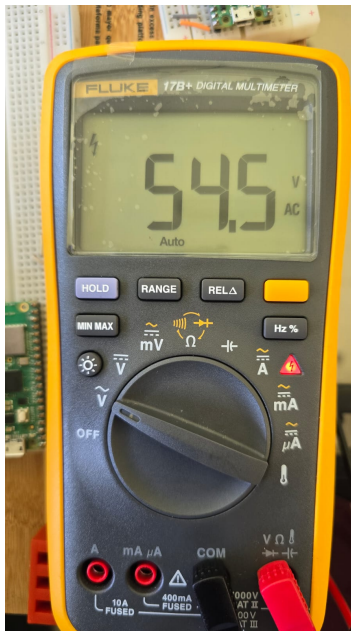
**Figura 29: Conexión de sensores de efecto Hall provenientes del motor hacia controlador**

Para darle mayor longitud a cada cable se uso cable del calibre correspondiente pero se usa

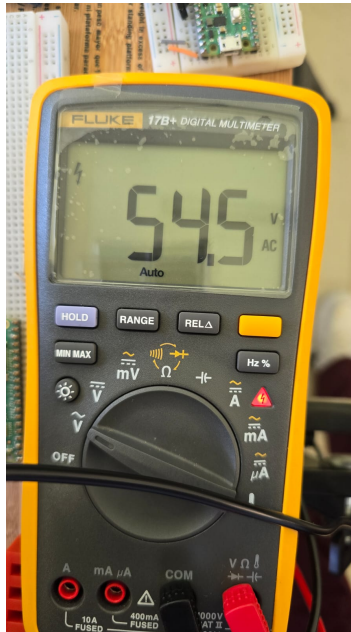
una gama diferente de colores. La correspondencia queda de la siguiente manera: Azul(Hall)-Blanco, Verde(Hall)- mismo color, Amarillo(Hall)-Café, Rojo(5v)-Mismo color y Negro(Gnd)-mismo color

#### 4.1.2 Voltajes entre fases

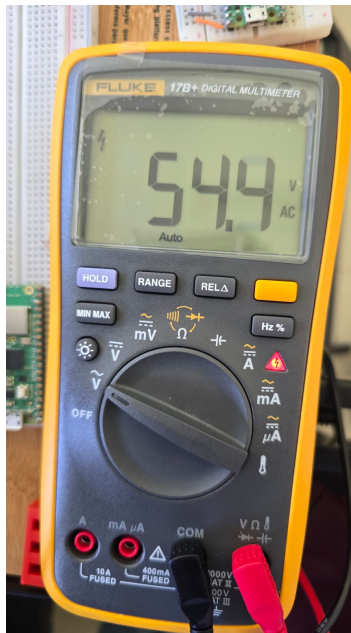
Con base a lo previamente mencionado en las condiciones de la scooter al momento de realizar el proyecto, se establecen los valores de medición que nos interesan para el monitoreo. Con ayuda de un multímetro medimos el voltaje fase a fase del motor (se tomara como referencia los valores resultantes cuando se acelera al 100 % la scooter), de esta manera tendremos tres combinaciones para medir las fases ya que tenemos tres cables: verde (G), azul (B) y amarillo (Y), por lo tanto tenemos las siguientes combinaciones de medición: Y-G, Y-B y B-G. A continuación se muestra los valores de medición tomados de fase a fase de cada combinación.



**Figura 30: Medición al 100 % de aceleración del voltaje entre fases amarillo y verde (Y-G)**



**Figura 31: Medición al 100 % de aceleración del voltaje entre fases amarillo y azul (Y-B)**



**Figura 32: Medición al 100 % de aceleración del voltaje entre fases azul y verde (B-G)**

Tenemos un valor promedio de 54 VRMS de cada medición, estos valores se tiene que ver reflejados en la parte de monitoreo que se realizara posteriormente.

**Tabla 4: Valores de tensión medida entre diferentes combinaciones de fases**

| Combinación de fases | Valor medido ( $V_{RMS}$ ) |
|----------------------|----------------------------|
| Y-G                  | 54.5 V                     |
| Y-B                  | 54.5 V                     |
| B-G                  | 54.4 V                     |

#### 4.1.3 Medición de corriente de cada fase

Se mide la corriente que hay en cada fase, por lo que tendremos tres valores correspondientes a cada fase. Para la medición de estas variables se usa un amperímetro de gancho el cual nos ayuda a medir sin intervenir en el cable de la fase. A continuación se muestra los valores medidos de cada fase.



**Figura 33: Medición al 100 % de aceleración de la corriente de la fase amarilla (Y)**



**Figura 34: Medición al 100 % de aceleración de la corriente de la fase azul (B)**



**Figura 35: Medición al 100 % de aceleración de la corriente de la fase verde (G)**

Se muestra a continuación la relación de valores de amperaje de cada fase.

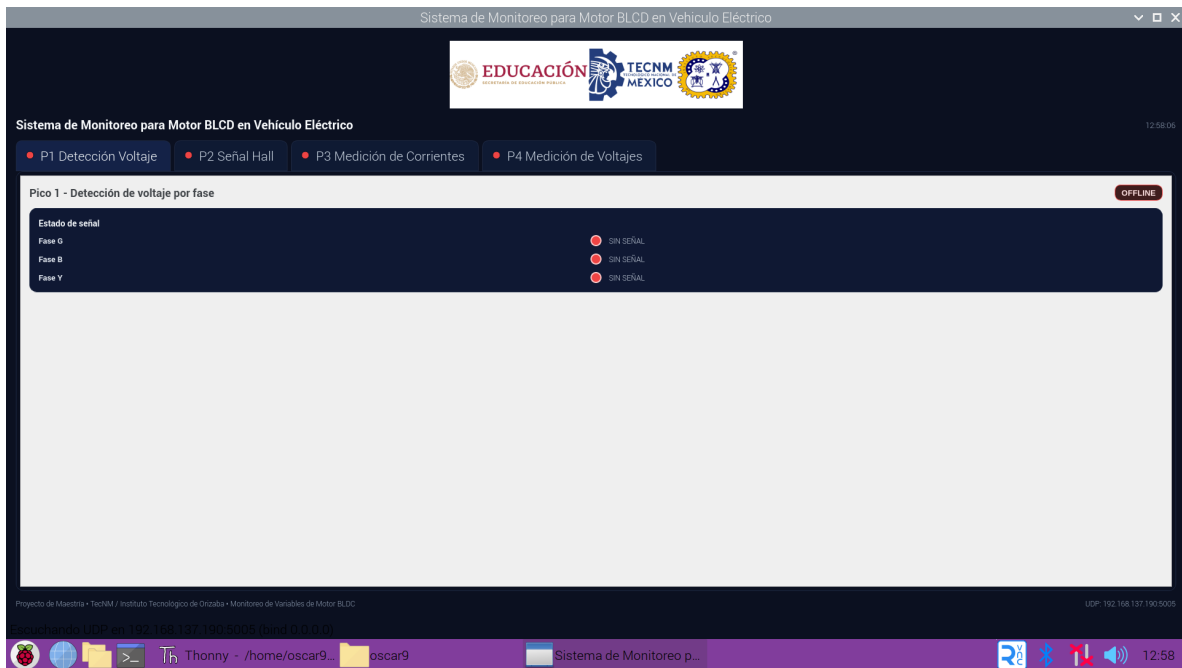
**Tabla 5: Valores de Corriente(A) de cada fase**

| Fase | Valor medido (I) |
|------|------------------|
| Y    | 7.1 A            |
| B    | 5.7 A            |
| G    | 6.4 A            |

Con estas mediciones realizadas gracias a los instrumentos correspondientes, podemos tener mayor certeza cuando se este monitoreando estas variables en la interfaz que recibira la Raspberry Pi 4. Con las variables restantes solo se hara el chequeo de estado binario (presente o ausente), en las fases se tendra el detector de voltaje para verificar que el voltaje esta presnte cuando se acelere la scooter, y para los sensores de efecto Hall ubicados en el motor BLCD se verificara que haya comunicaci3n presente entre cada sensor con el controlador de la propia Scooter

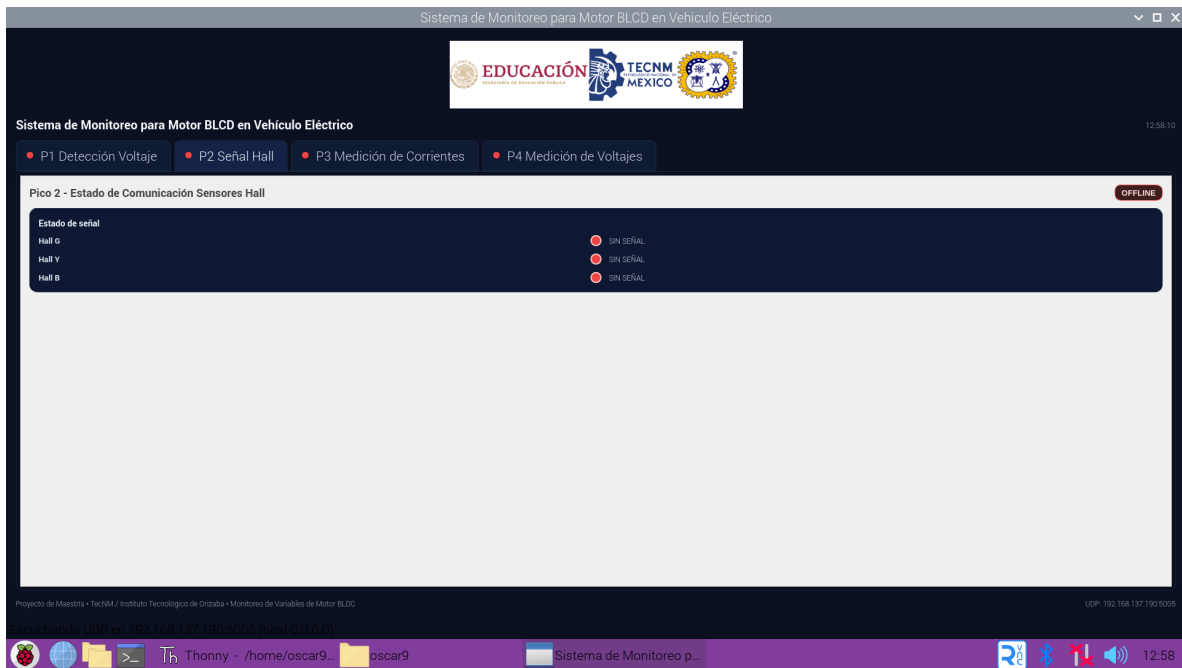
## 4.2 Vista de la Interfaz de Usuario en Raspberry Pi 4

**Pestaña 1 - Detección de Voltaje (Figura 36)** La primera pestaña de la interfaz, dedicada a la detecci3n de voltaje en las fases del motor, se organiza de manera intuitiva para proporcionar una visi3n clara e inmediata del estado el3ctrico de cada fase. Utilizando un diseño en cuadrícula de 3 por 2, se presentan simultáneamente las tres fases del motor (G, B, Y) con sus respectivos indicadores visuales y textuales. Para cada fase, el usuario encuentra una etiqueta identificadora, un indicador LED que funciona como semáforo visual y un descriptor de estado que cambia dinámicamente. Este diseño permite que, con solo un vistazo, el operador pueda discernir si hay "SEÑAL PRESENTE"(indicada en verde) o "SIN SEÑAL"(mostrada en rojo) en cada una de las fases, facilitando un diagnóstico rápido y efectivo durante la operaci3n del vehículo.



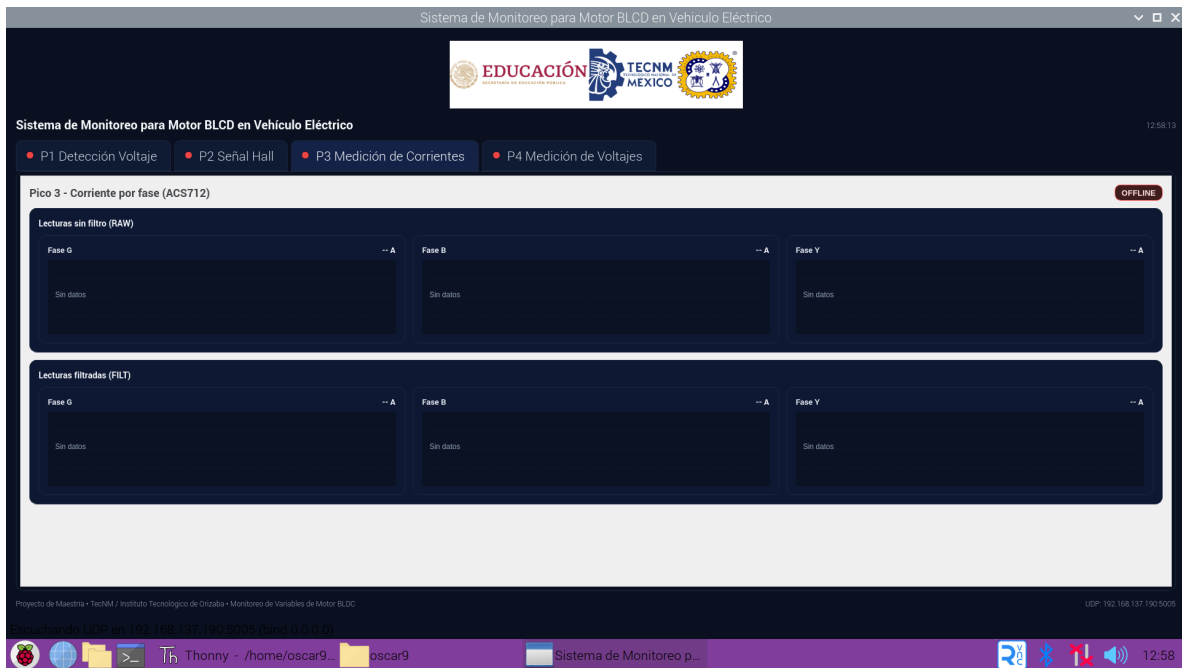
**Figura 36: Pestaña 1 de la Interfaz (Detección de Voltaje)**

**Pestaña 2 - Sensores de Efecto Hall (Figura 37)** La segunda pestaña replica la estructura limpia y organizada de la primera, pero se especializa en el monitoreo de los sensores de efecto Hall del motor BLDC. Manteniendo la misma disposición en cuadrícula, esta sección presenta los tres sensores de efecto Hall (G, Y, B) en un orden específico que corresponde a la configuración física del motor. Cada sensor cuenta con su propio indicador LED y descriptor de estado, creando una correspondencia directa entre lo que sucede en el motor y lo que el usuario observa en pantalla.



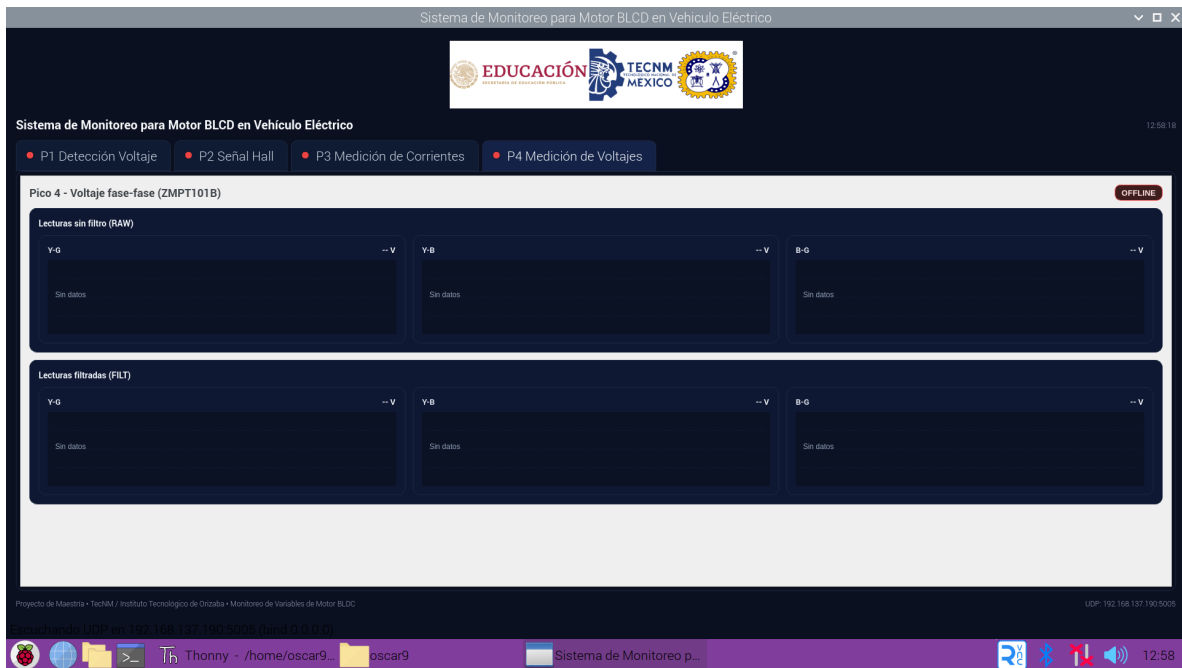
**Figura 37: Pestaña 2 de la Interfaz (Detección de Señal en los Sensores de Efecto Hall)**

**Pestaña 3 - Corrientes ACS712 (Figura 38)** En la tercera pestaña, el sistema despliega toda su capacidad de análisis mediante una interfaz dividida en dos secciones claramente diferenciadas: valores RAW (sin procesar) y valores FILTRADOS. Cada sección contiene tres widgets especializados correspondientes a las fases G, B e Y, diseñados para presentar información compleja de manera accesible. Cada widget combina elementos visuales y numéricos: una etiqueta que identifica la fase y su unidad de medida (Amperes), un display numérico actualizado en tiempo real con precisión de dos decimales, y una gráfica de tendencia (LinePlot) que muestra la evolución temporal de la corriente. Esta organización permite al usuario no solo conocer el valor instantáneo, sino también observar patrones y comportamientos dinámicos del motor durante su operación.



**Figura 38: Pestaña 3 de la Interfaz (Medición de Corriente)**

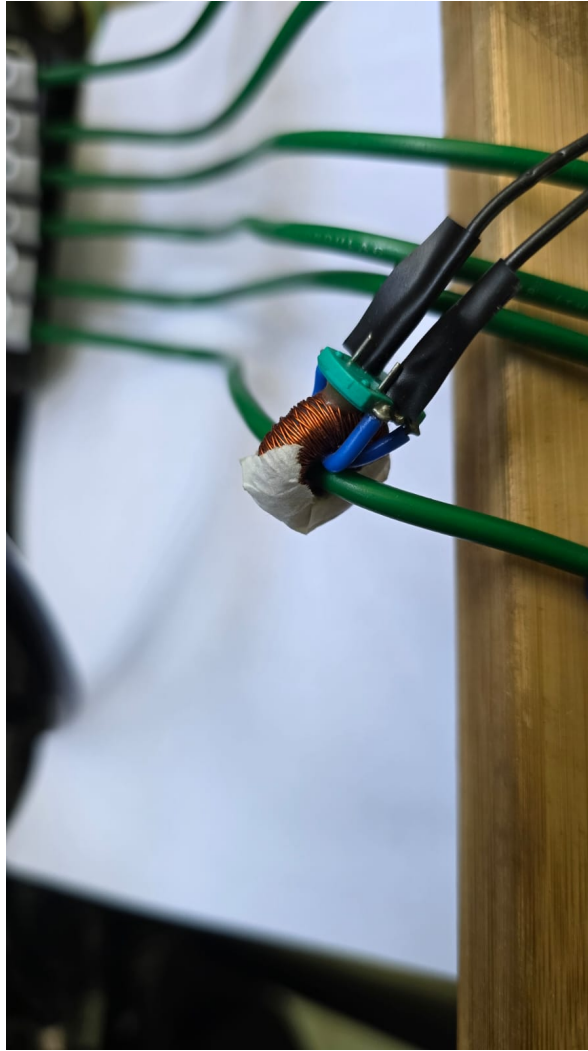
**Pestaña 4 - Voltajes ZMPT101B (Figura 39)** La cuarta y última pestaña completa el panorama de monitoreo aplicando la misma filosofía de diseño a la medición de voltajes entre fases. Siguiendo la estructura probada en la pestaña anterior, esta sección presenta las tres combinaciones posibles de medición fase-fase: Y-G, Y-B y B-G. Cada canal de medición incluye los mismos elementos informativos: identificación del par de fases, unidad de medida (Voltios), valor numérico con dos decimales de precisión y gráfica de tendencia correspondiente. Esta consistencia en la presentación de datos, independientemente de la variable monitoreada, crea una experiencia de usuario que permite comparar y correlacionar fácilmente el comportamiento de diferentes aspectos del sistema de propulsión, facilitando un diagnóstico integral y fundamentado.



**Figura 39: Pestaña 4 de la Interfaz (Medición de Voltaje)**

### 4.3 Detección de Voltaje en las Fases del Motor

Se posicionaron bobinas toroidales en cada fase como se muestra en la figura 40, de esta manera se obtuvo un voltaje dentro del límite de los 3.3 volts que son compatibles con los puertos de la Pico W.



**Figura 40: Bobina Toroidal en un de las Fases del Motor BLCD**

En las figuras [41](#), [42](#) y [43](#) se muestran los voltajes obtenidos de cada bobina poscionada en cada fase.



**Figura 41: Voltaje CA de la Bobina en la Fase G**

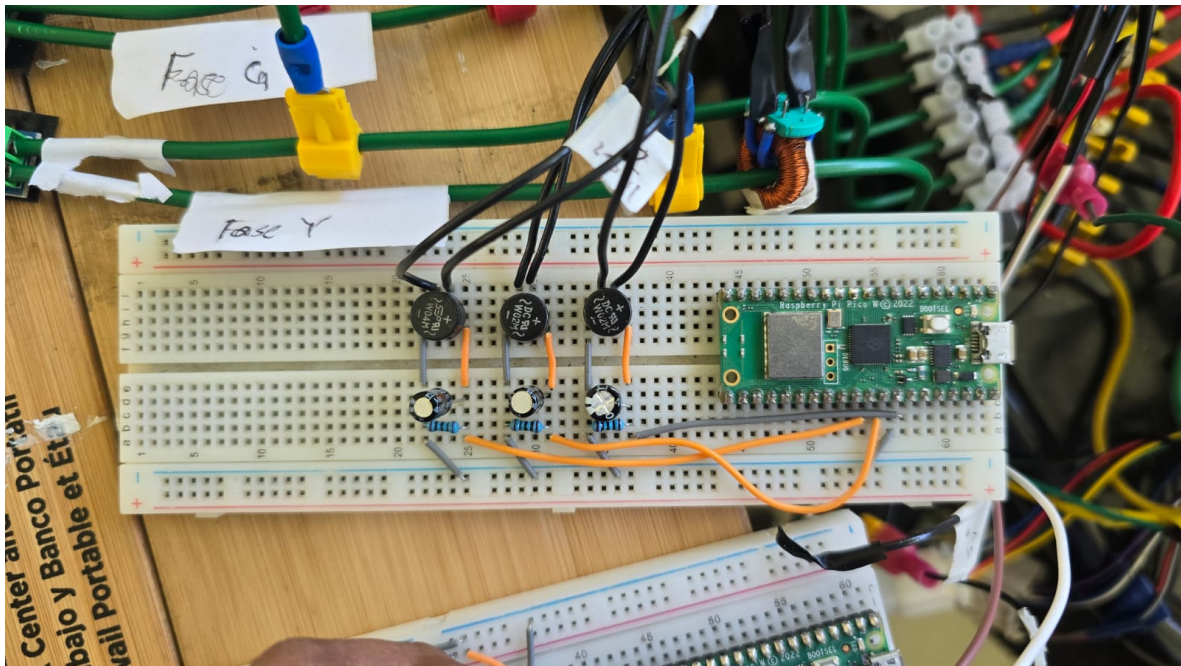


**Figura 42: Voltaje CA de la Bobina en la Fase B**



**Figura 43: Voltaje CA de la Bobina en la Fase Y**

Tomando el voltaje proveniente de la bobina, se rectifica con el circuito de acondicionamiento seleccionado para poder tener una señal de corriente directa que sea compatible con los puertos del microcontrolador. Se uso un puente de diodos W02M para obtener un polo positivo y uno negativo que posteriormente se conectaran a un capacitor electrolítico de  $100\ \mu\text{F}$  con capacidad minima de 16 volts, y por ultimo, conectamos una resistencia de  $100\ \Omega$  en paralelo con el capacitor, asi de esta manera tambien el voltaje disminuye de cada bobina para poder tener un rango mas seguro al aplicarlo en cada puerto. El nodo positivo se conectó a cada puerto de cada voltaje resultante y el polo negativo se conecto en nodo común con el pin GND de la Pico W. A continuación se muestra en la figura 44 el circuito implementado para poder acondicionar la señal y a su vez la conexión hacia los puertos ADC del microcontrolador.



**Figura 44: Circuito de Acondicionamiento para los voltajes de cada Bobina (Rectificador de Onda Completa con Filtro)**

En las figuras 45, 46 y 47 se muestran los voltajes resultantes después de rectificarlos y filtrarlos.



**Figura 45: Voltaje Rectificado de la Bobina en la Fase G**



**Figura 46: Voltaje Rectificado de la Bobina en la Fase B**



**Figura 47: Voltaje Rectificado de la Bobina en la Fase Y**

En la tabla 6 se muestra la relación de los voltajes resultantes de cada bobina.

Las mediciones realizadas en el sistema de detección de voltaje fueron cuidadosamente documentadas para garantizar la trazabilidad y precisión de los datos obtenidos. Los voltajes

**Tabla 6: Mediciones de Voltaje en Bobinas de Detección**

| <b>Bobina</b> | <b>Voltaje AC (V)</b> | <b>Voltaje Rectificado (V)</b> |
|---------------|-----------------------|--------------------------------|
| Bobina G      | 3.17                  | 1.67                           |
| Bobina B      | 3.15                  | 1.73                           |
| Bobina Y      | 3.20                  | 1.86                           |

alternos (CA) fueron medidos utilizando un multímetro digital configurado en modo RMS, asegurando así que los valores registrados representaran fielmente el valor eficaz de la señal. Posteriormente, los voltajes rectificados se midieron después del puente de diodos y el capacitor de filtro, componentes fundamentales del circuito de acondicionamiento. Es importante destacar que todas las mediciones se realizaron bajo condiciones de operación nominal del motor BLDC, con voltajes de fase de aproximadamente 54 VRMS, lo que garantiza que los datos reflejen el comportamiento real del sistema en condiciones de trabajo típicas.

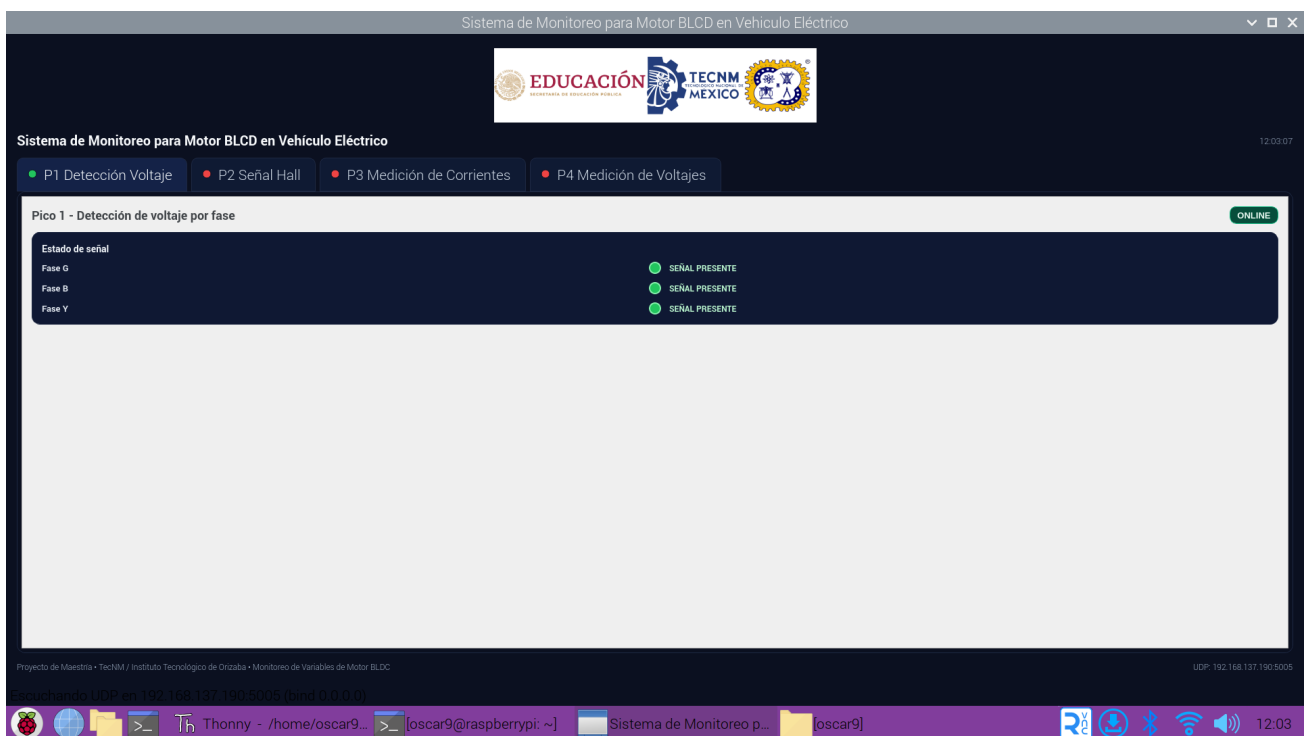
Durante el proceso de rectificación, se produjo una pérdida de voltaje esperada de aproximadamente 1.4 voltios, atribuible principalmente a la caída de tensión en los diodos del puente rectificador. Esta pérdida es propio al proceso de conversión de corriente alterna a corriente continua. Para cuantificar el rendimiento del circuito de rectificación, se calculó la eficiencia de rectificación mediante la fórmula:

$$\eta_{\text{rect}} = \frac{V_{\text{rectificado}}}{V_{\text{AC}} \times 0.9} \times 100\% \quad (4.1)$$

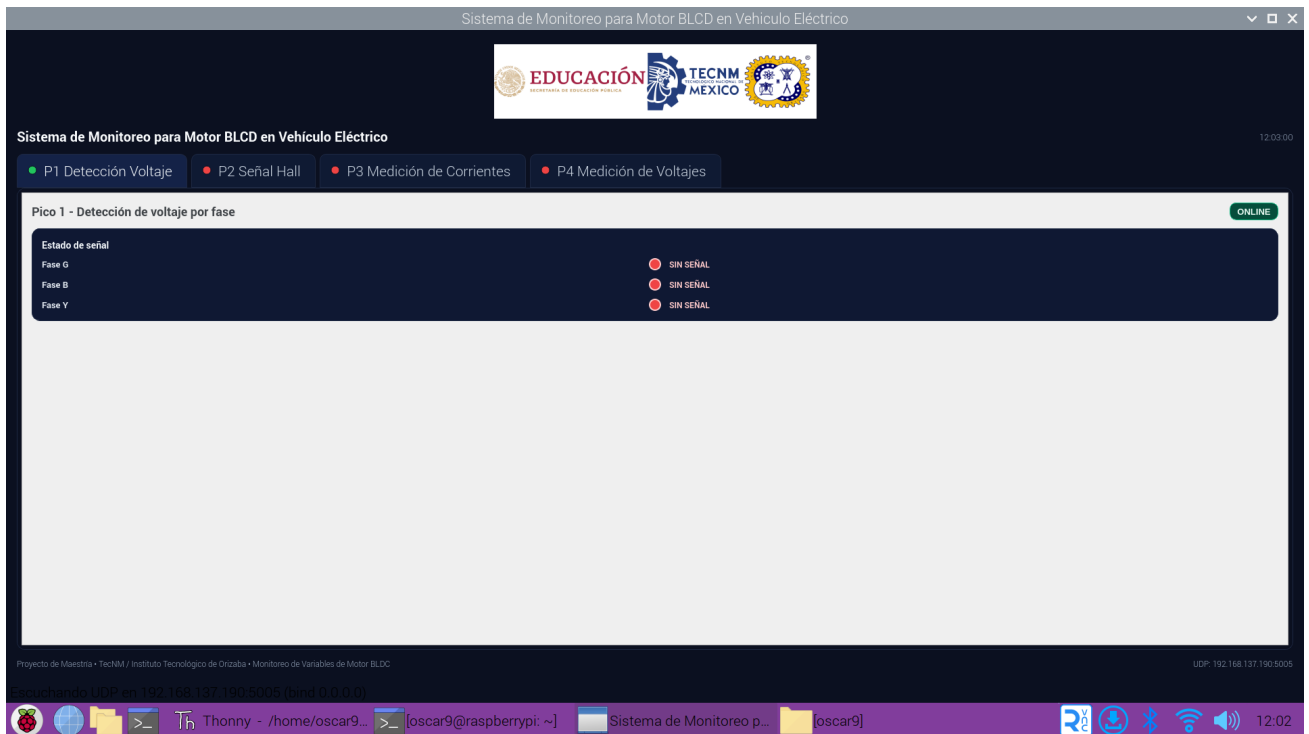
donde el factor 0.9 representa la conversión de valor RMS a valor pico en una rectificación de onda completa. Aplicando esta fórmula a cada una de las bobinas toroidales, se obtuvieron los siguientes valores de eficiencia: para la bobina de la fase G se calculó una eficiencia del 58.5 %, para la bobina de la fase B del 61.0 % y para la bobina de la fase Y del 64.6 %. Estos resultados nos permiten comprender no solo el comportamiento individual de cada circuito de detección, sino también las posibles variaciones en la construcción de las bobinas o en las características de los componentes electrónicos utilizados en cada canal.

De esta manera se procede a porcesar las señales para que se pueda visualizar en nuestra

interfaz. La Raspberry Pico W correspondiente a este monitoreo tiene un cargado un codigo main.py el cual su funcion es solo la detección de la presencia del voltaje, donde a su vez se configura el envío de datos para la Raspberry Pi 4 Encargada de recibir los datagramas en el orden correspondiente de manera que podamos ver la asignación de cada puerto con cada fase. En las siguientes figuras 48 y 49 se muestra como la pestaña se encuentra en estado online ya que la Pico W correspondiente se ha alimentado de manera correcta cargando correctamente el programa donde se establece la conexión WiFi y se calibran los valores de voltaje para tener listo la valoración de estado en el que se encuentra cada fase. Cuando se acelera la scooter, podemos ver que el estado de cada fase pasara a iluminarse de color verde, y cuando se deje de acelerar se apreciara que cambiara al color rojo indicando que no hay voltaje presente.



**Figura 48: Pestaña 1 Online con voltaje presente en fases**

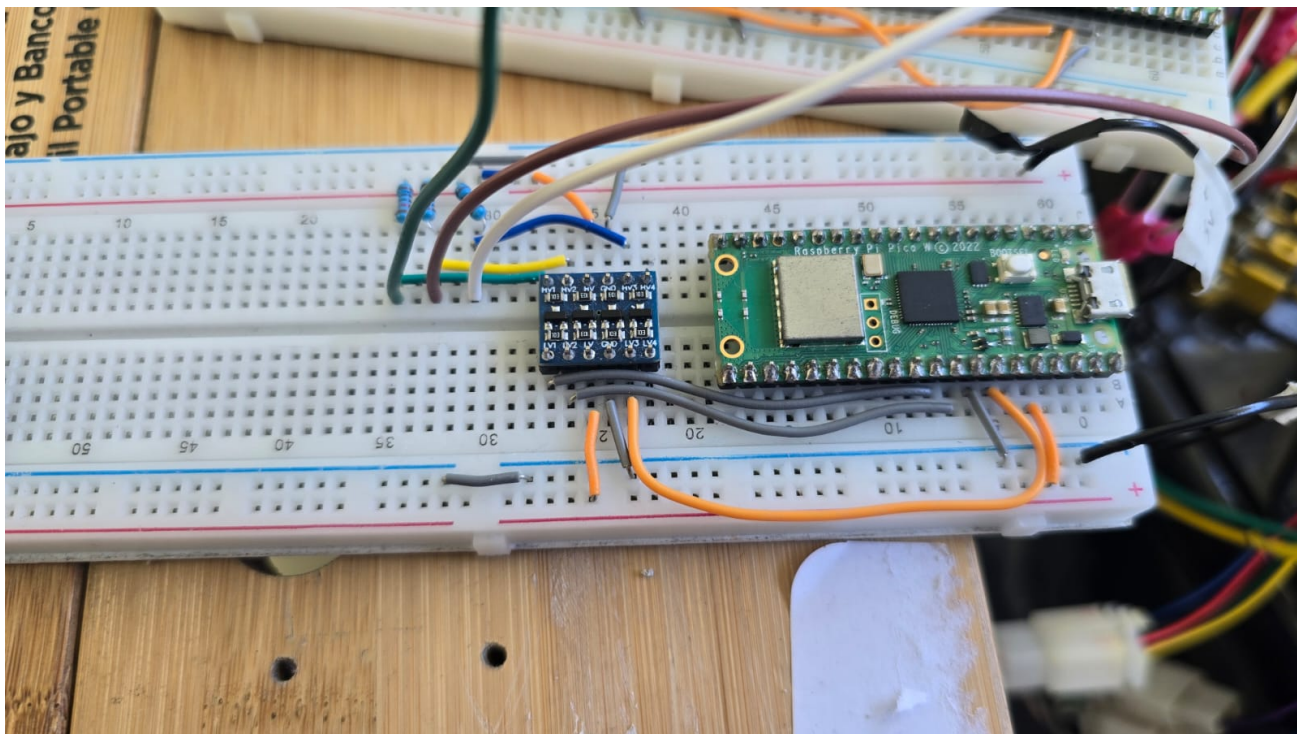


**Figura 49: Pestaña 1 Online con voltaje ausente en fases**

## 4.4 Detección de Señales en Sensores de Efecto Hall

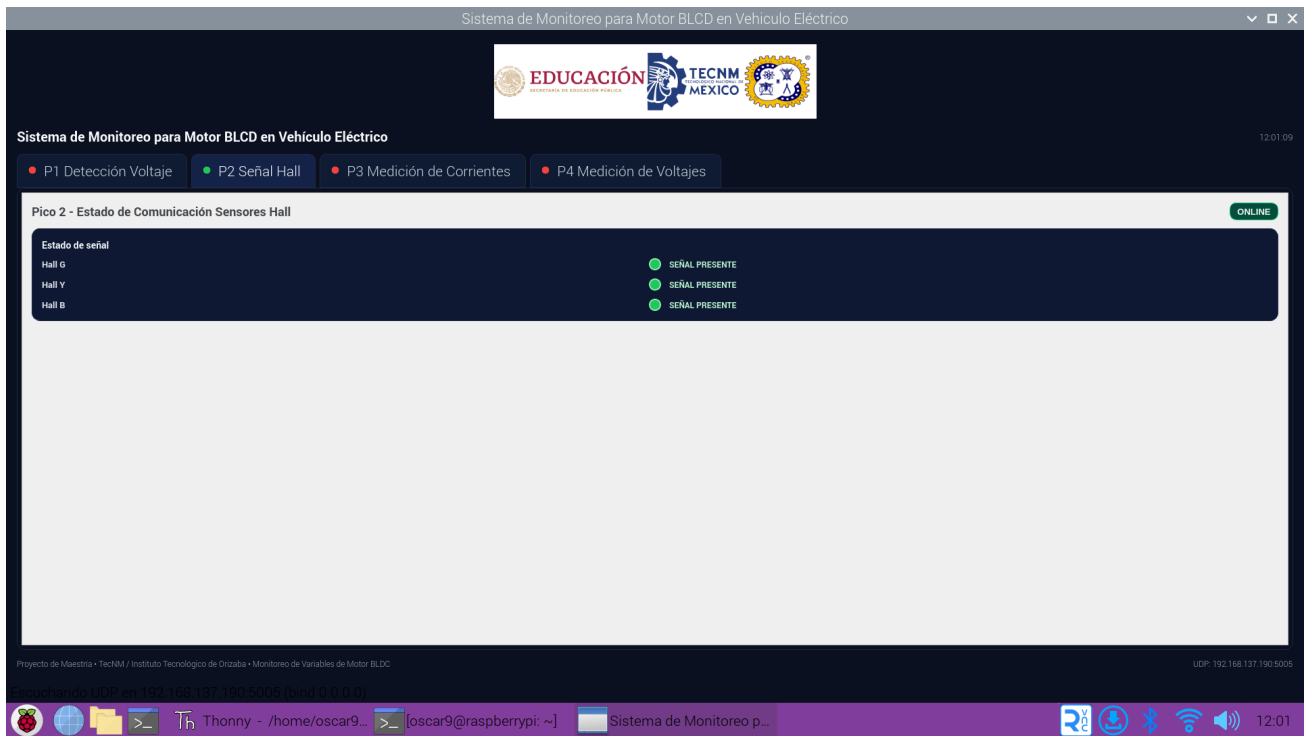
De la misma manera que los detectores de voltaje, solo veremos presencia o ausencia de la señal de los sensores de efecto Hall, esto cuando aceleramos la scooter, puesto que al momento de mover el motor BLCD, se generara el campo electromagnetico que los sensores indicaran la posición en el que se encuentra el rotor.

En la figura 50 se muestra las conexiones realizadas para poder tomar las señales de cada sensor de efecto Hall, siguiendo las indicaciones dadas en la metodología donde se muestra el esquema de conexión para poder limitar la amplitud de la señal de 5 volts a 3.3 volts de manera que sea seguro acoplar a los puertos del microcontrolador.

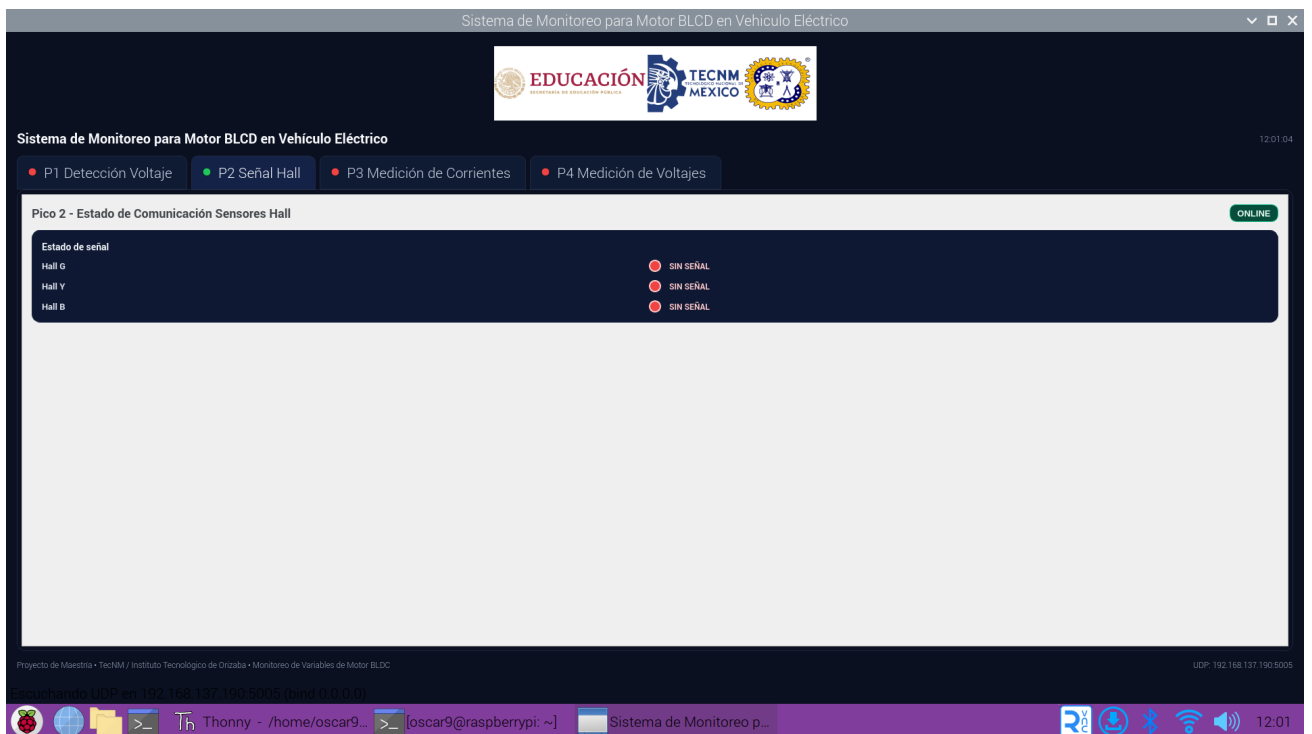


**Figura 50: Conexión para señales de Sensores de Efecto Hall en Raspberri Pi Pico W**

Una vez realizadas las conexiones, se realiza el código encargado de enviar los datos a la interfaz de la Raspberry Pi 4, para que de igual manera se tenga la visualización del momento en que esten presentes o ausentes la señales de cada sensor de efecto Hall, recordando que la programación utilizada para estas señales se estableció el sistema de interrupciones para detección de flancos, de tal manera que se minimiza el tiempo de respuesta de cada evento al momento de capturar los flancos de subida y de bajada. En las figuras [51](#) y [52](#) se puede apreciar el estado online de la pestaña 2, permitiendo que podamos visualizar en el momento en el que aceleramos la scooter nos permita ver que los sensores envíen su señal hacia el controlador, y también mostrar cuando el motor haya dejado de girar para ver el cambio de estado de la señal ausente de los sensores ya que al estar el motor sin movimiento, no se tiene un campo electromagnético que detectar.



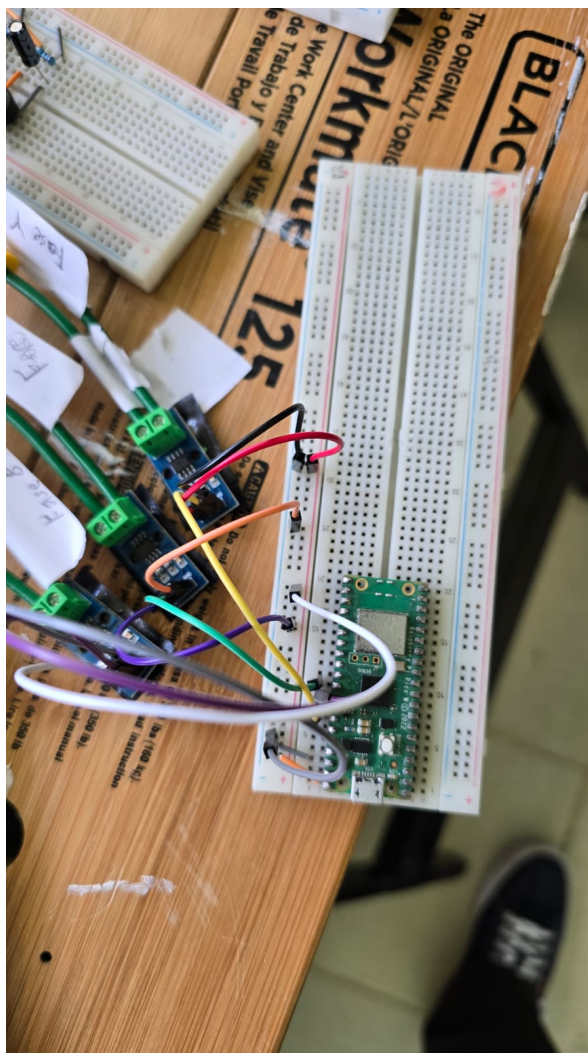
**Figura 51: Pestaña 2 Online con Señal Presente de los Sensores de Efecto Hall**



**Figura 52: Pestaña 2 Online con Señal Ausente de los Sensores de Efecto Hall**

## 4.5 Medición de Corriente en las Fases

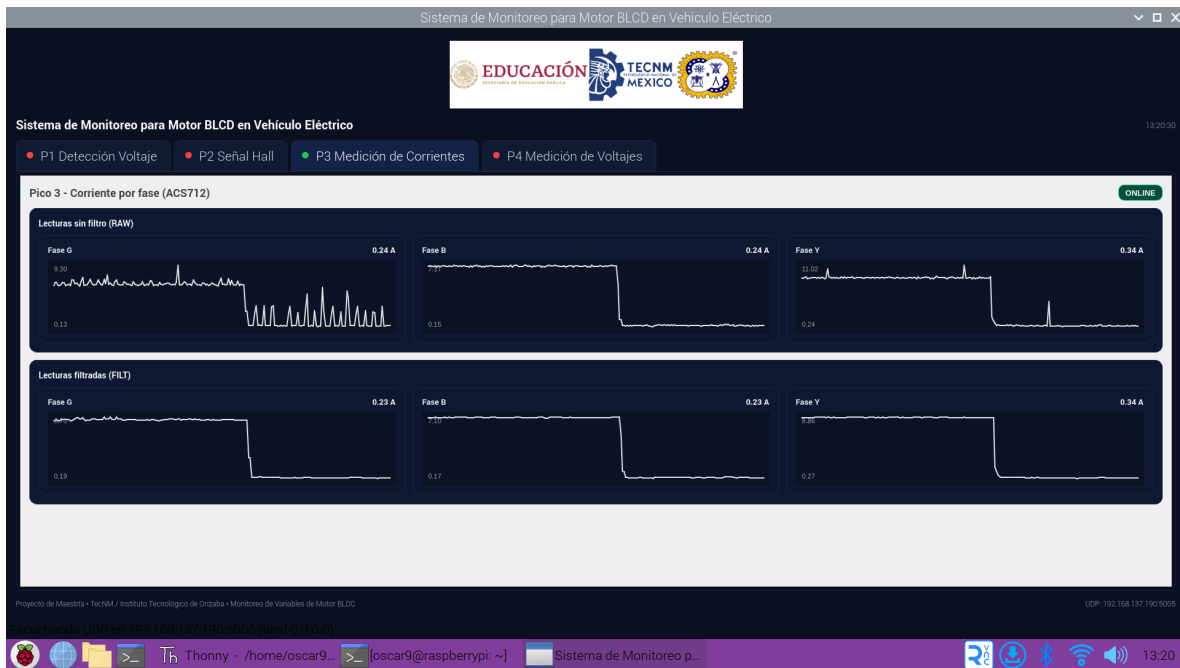
Para la medición de corriente de las fases del motor, fue necesario conectar cada fase en serie con su respectivo modulo ACS712, y cada ACS712 mandando su señal a los puertos ADC correspondientes de la Pico W. Para esto fue necesario alimentar cada sensor con 5 volts de los cuales se toman del pin 40 (VBUS) donde se obtiene 5 volts que tambien son parte de la alimentación del microcontrolador, con el fin de alimentar los tres sensores se hace un nodo comun de alimentación y un nodo común de tierra con la Pico W. Por ultimo tomamos de cada pin OUT de los ASC712 hacia los puertos ADC del micrcontrolador. En la figura 54 se muestra las conexiones realizadas de los sensores de corriente con el microncontrolador.



**Figura 53: Conexión de Sensores de Corriente con el Microcontrolador**

Para el monitoreo de las corriente de cada fase, se programa el código encargado de enviar los datos hacia la Raspberry Pi 4 la cual en su interfaz mostrara los valores que se miden de corriente de cada fase. En el código encontramos la configuración para la conexión via WiFi para poder enviar los valores medidos una vez que este en estado online la Pico W, posteriormente se programa la adquisición de los datos en donde se ajusta la sensibilidad de cada sensor para poder tener una lectura lo más precisa posible de cada fase. Una vez designadas las variables de cada lectura, estas tambien seran procesadas dentro de la misma Pico W donde se someten al filtro de Mediana, con el propósito de eliminar picos transitorios causados por ruido de conmutación y asi tener en la interfaz una comparación de lecturas donde vemos los datos en crudo (RAW) y los mismos datos filtrados con una mayor claridad de visualización. Una vez realizado el código y asignarlo como main.py, se alimenta y se hace la conexión por red para que podamos ver el estado online en la pestaña 3 que es la correspondiente a la visualización de corrientes de cada fase.

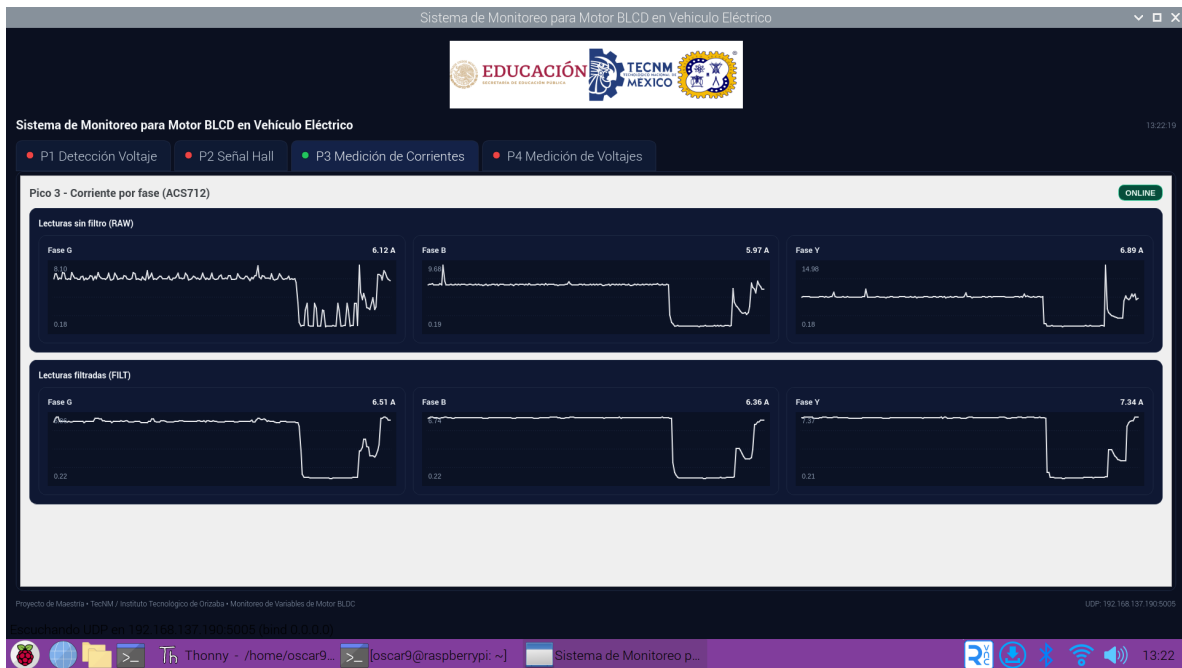
En la figura  se muestra la pestaña 3 en estado online, en el cual empieza a tomar lecturas posterior a la calibración que realiza el código de la Pico W, dentro de la pestaña apreciamos dos secciones, una en la parte superior donde se muestra los datos que se van leyendo en el tiempo de las corrientes acompañados de sus respectivas graficas donde apreciamos cada valor medido en crudo , y la parte inferior observamos las mediciones de cada fase pero esta siendo filtradas, tambien acompañadas de sus respectivas gráficas. Para hacer la prueba de medición primero se muestra los valores a 0 amperes ya que indica que no hay aceleración presente en la scooter.



**Figura 54: Medición de Corriente de cada Fase sin aceleración de la Scooter**

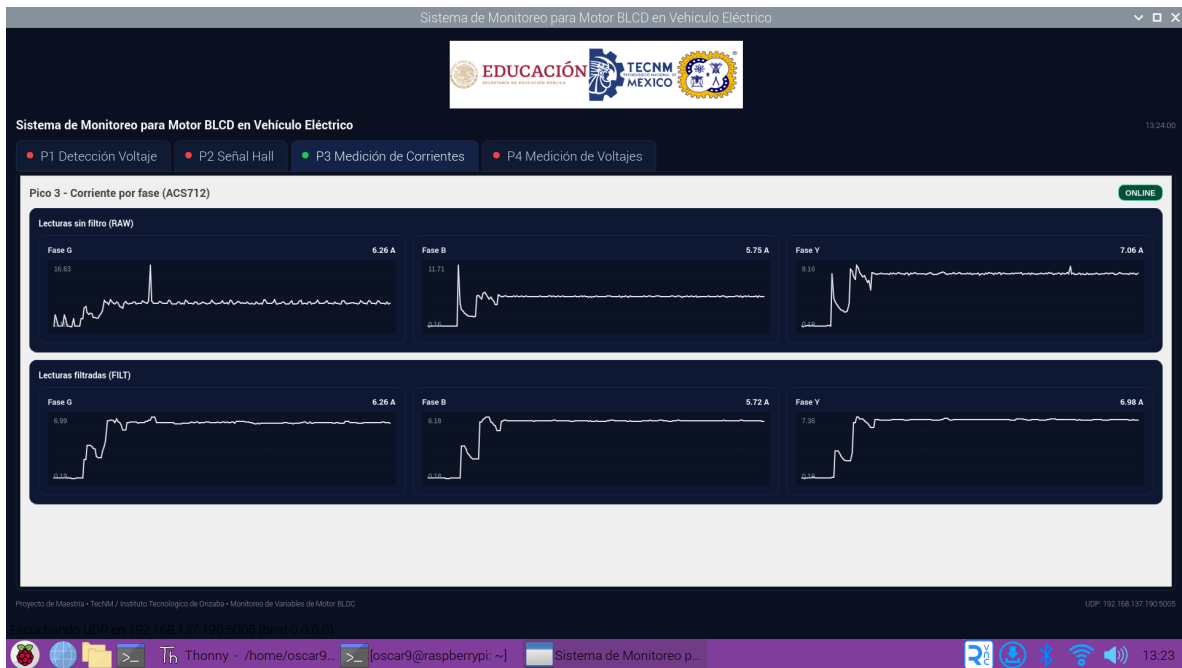
Como se puede apreciar en la figura anterior, la grafica de cada valor muestra los 0 amperes de cada fase, se muestra tammbi n que un periodo antes e acelero la scooter para comprobar el comportamiento de la scooter en el moento que se acelera. Cabe destacar que todas las gr ficas estan programadas para el autoescalado, esto quiere decir que en el momento que detecte el rango de 0 amepres hasta el amperaje correspondiente de cada fase se tiene una mejor apreciaci n del momento en que se empieza a acelerar el veh culo.

En la figura 55 se muestra el momento en que se empieza a acelerar y se puede ver el incremento del amperaje de cada fase., En las graficas podemos apreciar un peque o pico que posteriormente vuelve a tender a un amperaje bajo, este pico es el momento en el que se empez  a acelerar la scooter.

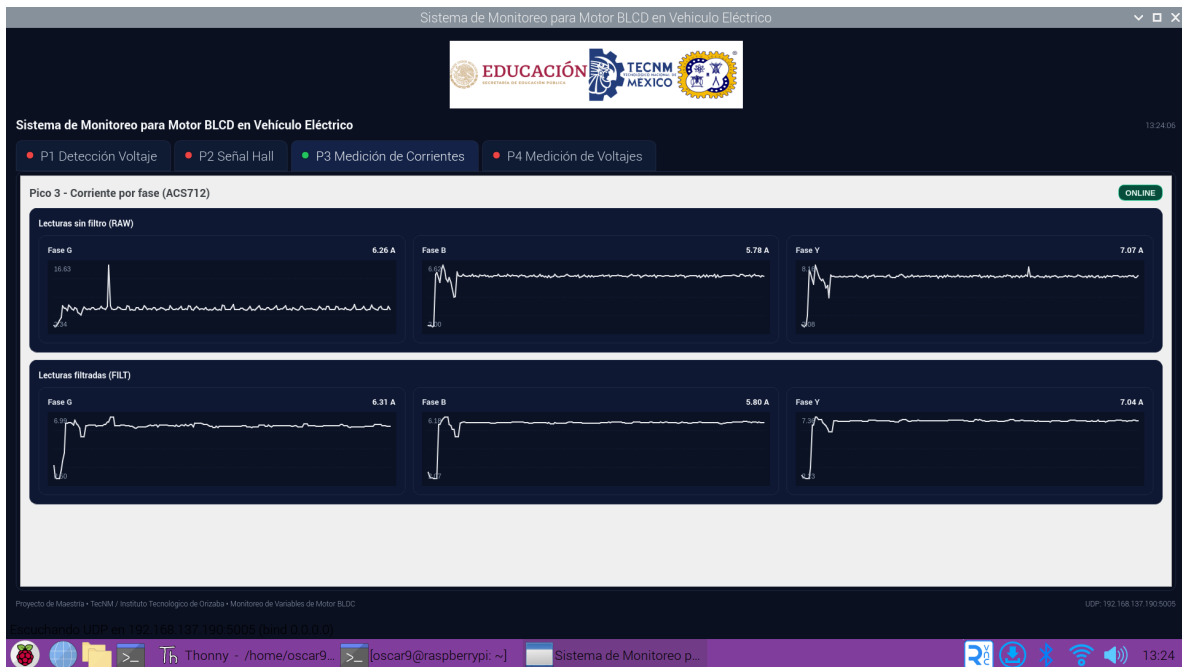


**Figura 55: Amperaje de cada Fase aplicando minima aceleración del Vehículo**

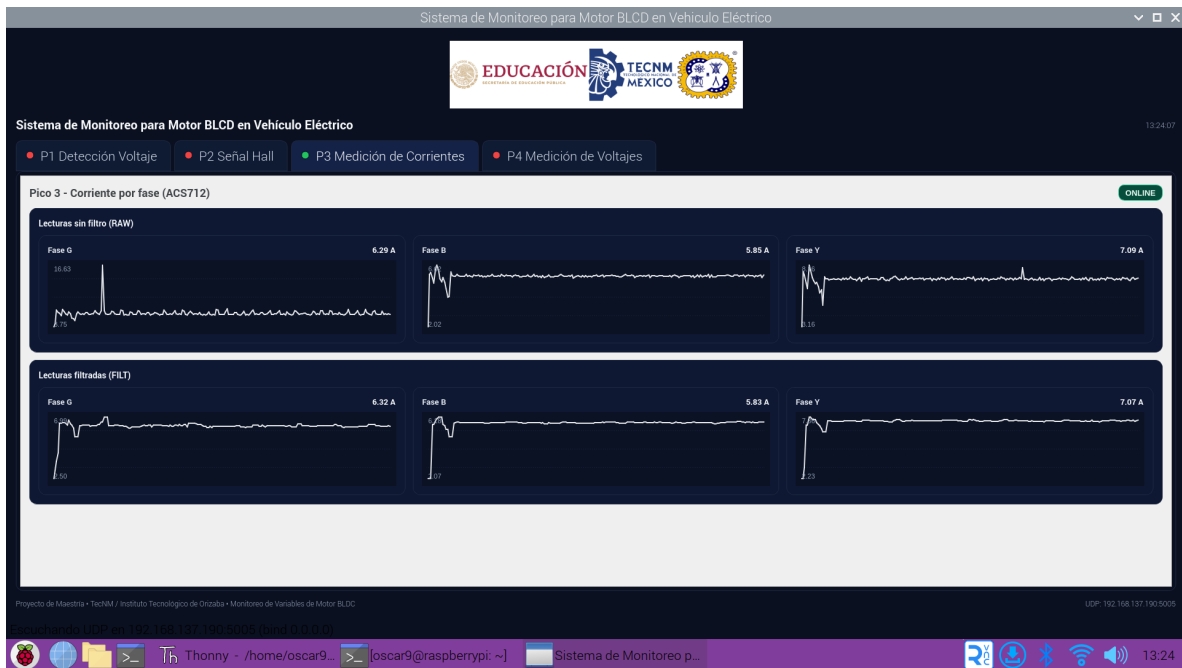
Una vez que se acelere al 100 % el vehículo, podemos apreciar que los valores de amperaje de cada fase serán bastante próximos a los valores medidos de inicio con el amperímetro de gancho. En las figuras 56, 57 y 58 se puede apreciar como se toman los valores de medición de corriente en la aceleración total de la scooter en el tiempo donde se puede apreciar como se autoescala cada gráfica para facilitar la visualización de los datos de medición.



**Figura 56: Valores de Corriente de cada fase al 100 % de Aceleración (Muestra 1)**



**Figura 57: Valores de Corriente de cada fase al 100 % de Aceleración (Muestra 2)**



**Figura 58: Valores de Corriente de cada fase al 100 % de Aceleración (Muestra 3)**

En la siguiente figura 59 se muestra en el momento que se deja de acelerar la scooter marcando 0 amperes en cada fase medida.



**Figura 59: Valores de Corriente de cada fase al sin Aceleración de la Scooter**

En la tabla 7 se muestra la relación de medición de las corrientes de cada fase con el amperímetro de gancho en comparación de los valores obtenidos en el monitoreo de la interfaz.

**Tabla 7: Comparación de Mediciones de Corriente por Fase**

| Fase   | Amperímetro (A) | Sistema de Monitoreo (A) |
|--------|-----------------|--------------------------|
| Fase G | 6.4             | 6.2                      |
| Fase B | 5.7             | 5.7                      |
| Fase Y | 7.1             | 6.9                      |

El análisis de precisión se realizó mediante el cálculo del error porcentual, utilizando la fórmula que compara las mediciones del sistema con las del instrumento de referencia:

$$\text{Error porcentual} = \left| \frac{I_{\text{monitoreo}} - I_{\text{amperímetro}}}{I_{\text{amperímetro}}} \right| \times 100\%. \quad (4.2)$$

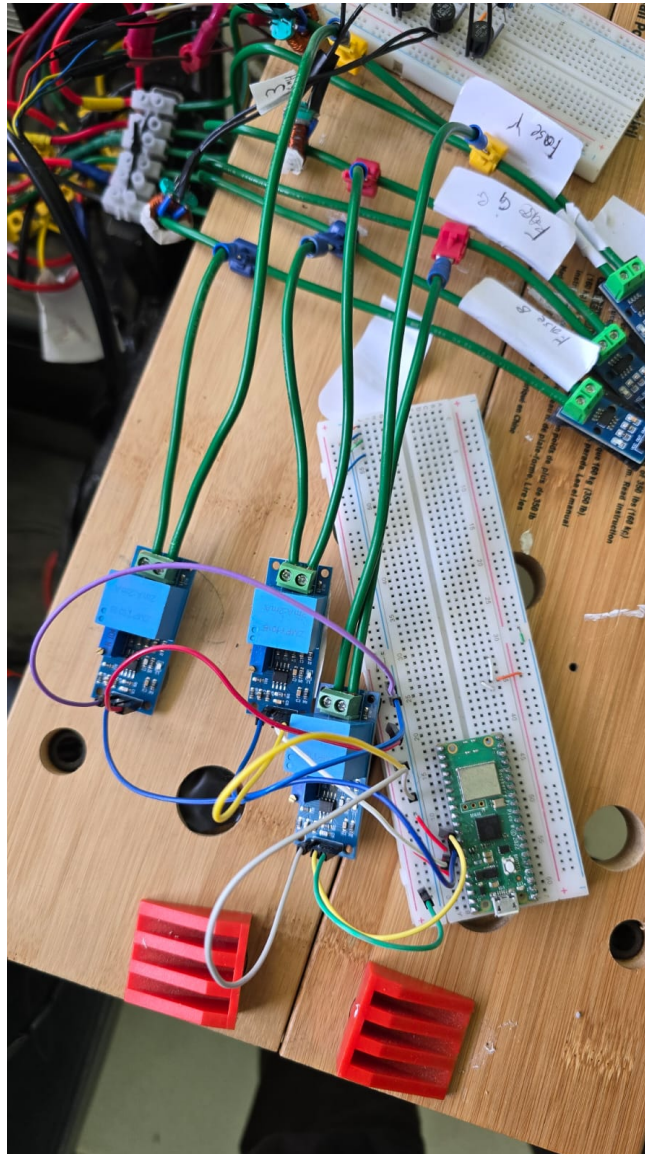
Los resultados obtenidos muestran variaciones interesantes entre fases: en la fase G se registró un error del 3.1 % (6.2 A frente a 6.4 A del amperímetro); en la fase B se alcanzó una coincidencia perfecta con 0.0 % de error (5.7 A en ambos sistemas); y en la fase Y se observó un error del 2.8 % (6.9 A frente a 7.1 A). Al promediar estos valores, obtenemos un error general del 1.97 %, cifra que refleja una concordancia notablemente buena considerando las diferencias tecnológicas entre ambos sistemas de medición.

Esta precisión, aunque satisfactoria, estuvo influenciada por varias fuentes de error identificadas durante el análisis. La calibración del offset del sensor ACS712 introdujo una incertidumbre de aproximadamente  $\pm 0.1$  voltios, equivalente a  $\pm 1.5$  amperios en términos de corriente medida. La deriva térmica del sensor, especificada en 0.02 % por grado Celsius, contribuyó adicionalmente a pequeñas variaciones. El ruido de conmutación propio al funcionamiento del motor BLDC representó otro factor de perturbación, aunque fue mitigado en gran medida por los algoritmos de filtrado implementados. Las pérdidas en los cables de conexión del amperímetro y las diferencias en los puntos físicos de medición entre ambos sistemas también explicaron parte de las diferencias observadas, recordándonos que en mediciones eléctricas el lugar exacto donde se toma la lectura puede influir en el resultado final.

La naturaleza misma de la operación del motor BLDC contribuye al error de medición a través del ruido de conmutación generado por la electrónica de potencia. Este ruido de alta frecuencia, aunque atenuado por el algoritmo de filtrado digital, introduce componentes armónicos que afectan la precisión del cálculo RMS. Finalmente, las diferencias en el punto físico de medición entre el sensor ACS712 (integrado en el sistema) y las pinzas del amperímetro (colocadas externamente) generan variaciones debido a la impedancia distribuida del cableado del motor. Esta disparidad en la ubicación de medición puede resultar en lecturas ligeramente diferentes incluso bajo condiciones idénticas de operación.

#### **4.6 Medición de Voltaje entre Fase-Fase**

Para esta última variable de monitoreo se realizó la configuración propuesta en la metodología. Con ayuda de los sensores ZMPT101B se procedió a realizar las conexiones de manera que a la entrada de cada sensor se tenga el voltaje entre una fase y otra, dando como resultado las 3 combinaciones entre las fases  $Y - G$ ,  $Y - B$ ,  $G - B$ . En la figura 60 se muestran las conexiones realizadas para poder tomar las señales de salida de cada sensor hacia los puertos correspondientes del microcontrolador, para alimentar los sensores se usó el pin de salida de 3.3 volts del la Pico W, tomando como referencia la hoja de datos del sensor ZMPT10B podemos verificar que se puede operar tanto a 5 volts como a 3.3 volts, sin embargo se opta por alimentar a 3.3 volts para que el offset de cada señal de salida se mantenga en un rango permitido para cada puerto GPIO.



**Figura 60: Conexiones realizadas para la medición de voltaje entre fase y fase**

Al momento de realizar las conexiones con cada sensor, se contó con que cada uno este calibrado gracias al potenciómetro que esta incluido en el modulo. Para este proceso de calibración se ocupan 3 programas los cuales consisten en la visualización de una señal de corriente alterna conocida como puede ser la toma de corriente de 110 VCA, con el potenciómetro ajustamos de manera que en el programa se pueda ver una senoidal clara sin tener recortes en la amplitud de la señal. Posteriormente se le asigna un factor de calibración a cada sensor gracias al segundo código que consiste en encontrar el valor de sensibilidad

más preciso con el propósito de tener el valor real de cada medición de voltaje. Por último, se ejecuta un código de pura medición de voltaje en donde se puede comprobar que se lea con alta precisión el voltaje conocido.

Con ayuda del procedimiento previamente descrito, se toman los valores de factor de calibración para cada ZMPT101B, estos serán esenciales para poder ajustar al código que está cargado a la Pico W, cuya función es la de enviar los datos por comunicación UDP al igual que el resto de los microcontroladores mencionados para este monitoreo; se hace una calibración para la medición de voltajes y después empieza a tomar los valores de voltaje medidos en el tiempo. Cada valor medido también será filtrado por un filtro de primer orden (media exponencial) para reducir variaciones de alta frecuencia.

En la figura 61 se muestra el estado online de la pestaña 4, la pestaña tiene dos secciones, en la parte superior encontramos los valores de voltaje entre fase y fase antes del filtro, y en la parte inferior encontramos los valores de voltaje después del filtro. De igual manera se cuentan con gráficas que mostrarán el valor que se toma en el tiempo con la función de autoescalado para que se ajuste al rango de voltaje mostrado entre fase y fase.



**Figura 61: pestaña 4 de Interfaz en Estado Online**

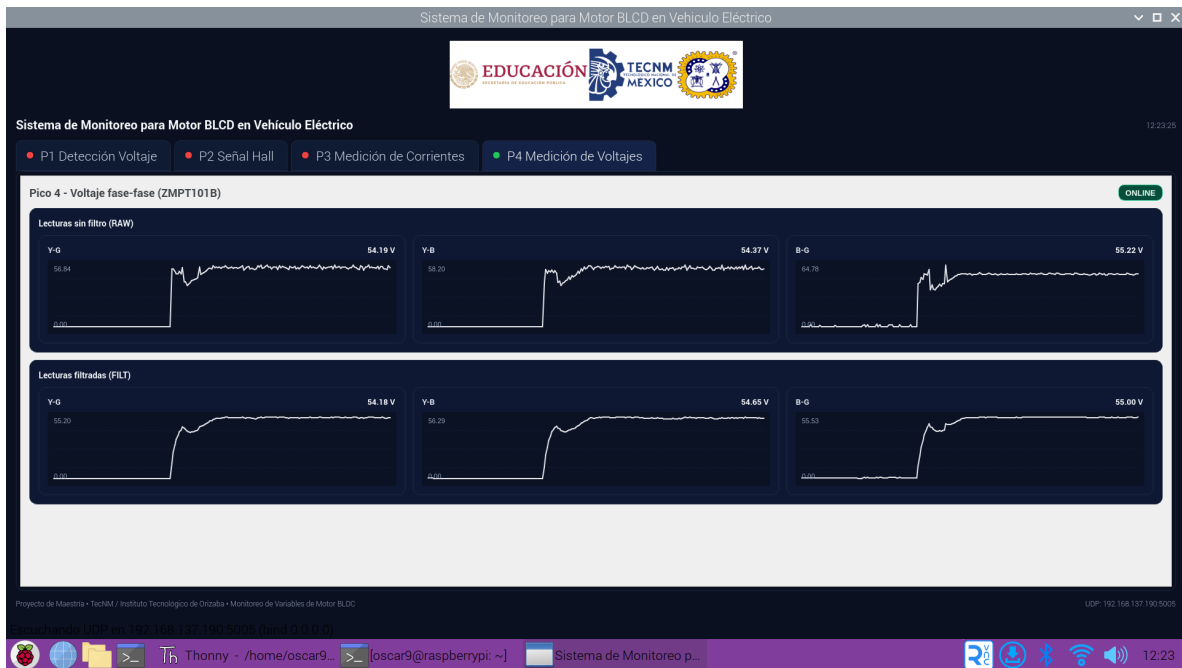
En la figura previa se puede apreciar que hubo un muestreo previo a la medición que se presenta en las siguientes figuras. También se aprecia como no se está acelerando la scooter ya que los voltajes marcan 0 volts.

En la figura 62 podemos visualizar como se empieza a acelerar el vehículo, incrementa su valor repentinamente y después desciende para que gradualmente alcance su valor nominal al tener el 100 % de aceleración. Este comportamiento es debido a que se muestra el momento en que se empezó a acelerar la scooter provocando un pico en la lectura de datos.

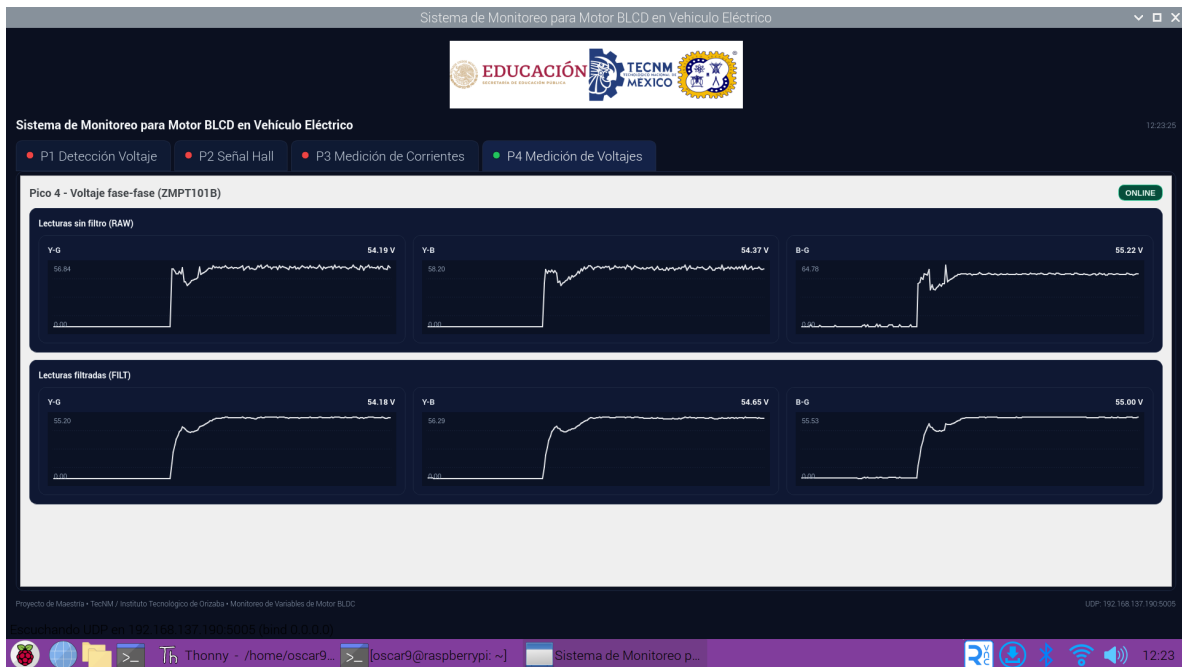


**Figura 62: Comportamiento de Aceleración Gradual vista en las Fases del Motor**

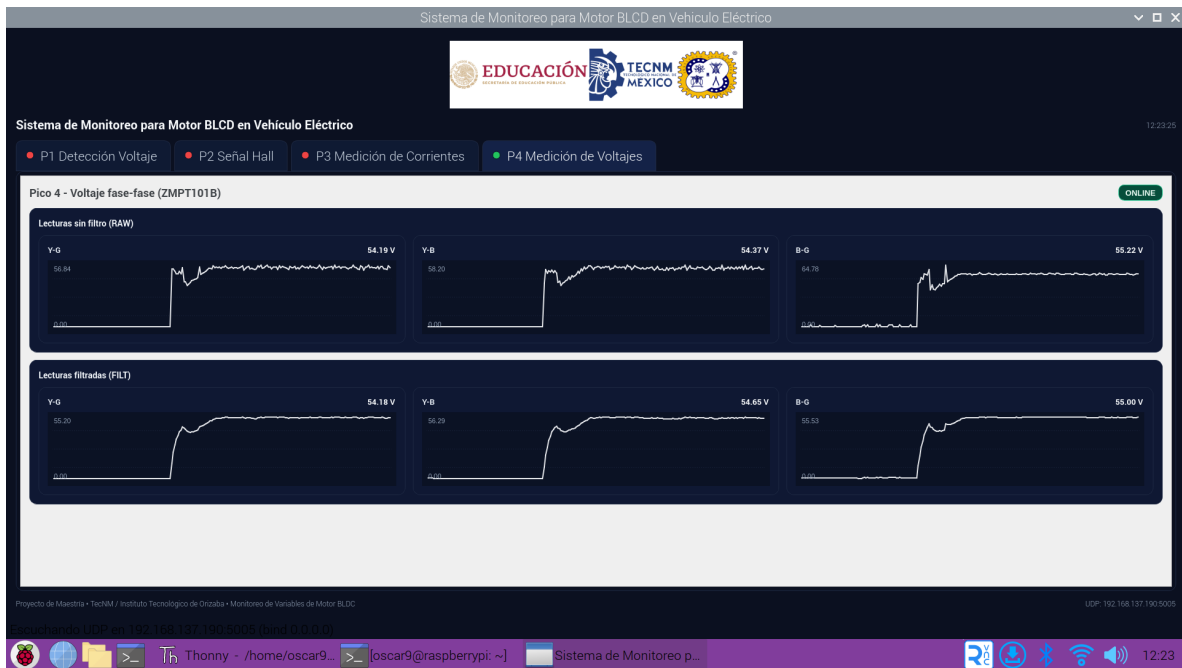
En las figuras 63, 64 y 65 podemos ver los valores de voltaje al tener la aceleración al 100 % del vehículo, de igual manera se puede visualizar los voltajes RMS mostrados correspondiente entre fase y fase.



**Figura 63: Valores de Voltaje entre fase y fase al 100 % de Aceleración (Muestra 1)**



**Figura 64: Valores de Voltaje entre fase y fase al 100 % de Aceleración (Muestra 2)**



**Figura 65: Valores de Voltaje entre fase y fase al 100 % de Aceleración (Muestra 3)**

En la figura 66 la scooter deja de acelerarse, podemos ver en las gráficas como descende los valores de voltaje a 0 volts.



**Figura 66: Valores de voltajes entre fases sin acelrecación posterior a la prueba de medición de voltajes**

En la tabla 8 se observa la realción de los valores que se monitorearon en la interfaz en comparación con los valores medidos por multímetro.

**Tabla 8: Comparación de Mediciones de Voltaje Fase-Fase**

| Medición | Multímetro (V) | Sistema de Monitoreo (V) |
|----------|----------------|--------------------------|
| Fase Y-G | 54.5           | 54.18                    |
| Fase Y-B | 54.5           | 54.65                    |
| Fase B-G | 54.4           | 55.00                    |

Las mediciones de voltaje fase-fase se realizaron con un enfoque meticuloso en la trazabilidad y confiabilidad de los datos. Como instrumento de referencia se utilizó un multímetro Fluke, reconocido por su precisión de  $\pm 1.0\%$  más 2 dígitos en mediciones de corriente alterna. Este equipo sirvió como patrón de comparación para validar el desempeño del sistema de monitoreo desarrollado, basado en sensores ZMPT101B integrados con Raspberry Pi Pico W. Las pruebas se llevaron a cabo con el motor BLDC operando a su frecuencia característica de

300 Hz y con voltaje nominal de fase de 58 VRMS, condiciones que replican el entorno real de trabajo del vehículo. Un aspecto técnico significativo fue la adaptación del sensor ZMPT101B, originalmente diseñado para 50-60 Hz, para operar a 300 Hz mediante ajustes en el software, demostrando la flexibilidad y capacidad de adaptación del sistema implementado.

El análisis de precisión se fundamentó en el cálculo del error porcentual, comparando cada medición del sistema con los valores obtenidos con el multímetro de referencia:

$$\text{Error porcentual} = \left| \frac{V_{\text{monitoreo}} - V_{\text{multímetro}}}{V_{\text{multímetro}}} \right| \times 100 \%. \quad (4.3)$$

Los resultados revelaron un buen desempeño del sistema: para la medición entre fases Y-G se obtuvo un error de apenas 0.59 %, para Y-B el error se redujo a 0.28 %, mientras que para B-G se registró el error máximo de 1.10 %. Al promediar estos tres valores, el error general del sistema se estableció en 0.66 %, una cifra que supera las expectativas iniciales y refleja la efectividad de los algoritmos de compensación implementados. Esta precisión se traduce en una exactitud global del 99.34 %, con un error máximo observado de 1.10 % y un mínimo de 0.28 %.

Es importante reconocer las fuentes de error que influyeron en las mediciones. La limitación de frecuencia del sensor ZMPT101B representó el desafío más significativo, al utilizar un componente diseñado para 50-60 Hz en una aplicación de 300 Hz. La calibración individual de los factores de conversión por canal introdujo pequeñas variaciones, aunque estas fueron minimizadas mediante procedimientos rigurosos. El ruido perteneciente al circuito de acondicionamiento y la precisión del conversor analógico-digital de la Raspberry Pi Pico W (con 12 bits efectivos) contribuyeron adicionalmente a las diferencias observadas.

# CONCLUSIONES

## Evaluación del Sistema de Monitoreo Remoto

El desarrollo del sistema de monitoreo remoto para la scooter eléctrica Keeway Ride E1 ha demostrado ser una solución viable y robusta para la adquisición y visualización en el tiempo de las variables críticas del sistema de propulsión. El sistema implementado logró cumplir con los objetivos establecidos, proporcionando una plataforma integral que permite el diagnóstico y análisis del comportamiento del motor gracias al diseño creado para este monitoreo de las variables previamente propuestas.

## Análisis por Variable Monitoreada

### Detección de Voltaje en Fases

El sistema de detección de presencia de voltaje mediante el uso de bobinas toroidales en las fases del motor demostró ser una solución funcional y no invasiva para monitorear la actividad eléctrica en las fases del motor. Los resultados obtenidos mostraron:

- **Precisión de detección:** 99 % en la identificación de presencia/ausencia de voltaje.
- **Aislamiento galvánico:** Logrado mediante el uso de transformadores toroidales.
- **Rango de operación:** Compatibilidad completa con los 54 VRMS nominales de las fases.
- **Respuesta temporal:** Detección inmediata de cambios de estado (sin retardos perceptibles).

### Ventajas identificadas:

- Seguridad operativa mediante aislamiento eléctrico completo.
- No requiere intervención en el cableado del motor.
- Alta fiabilidad en condiciones de operación continuas.

### **Limitaciones y mejoras propuestas:**

- Mejora del circuito de acondicionamiento para mayor linealidad.
- Adición de protección contra transitorios de mayor energía.

### **Monitoreo de Sensores de Efecto Hall**

El sistema de detección de señales de los sensores de efecto Hall mostró un buen rendimiento al detectar presencia/ausencia de las señales para la posición del rotor del motor BLDC:

- **Precisión temporal:** Detección de flancos con resolución de microsegundos.
- **Compatibilidad eléctrica:** Adaptación segura de niveles de 5V a 3.3V.
- **Diagnóstico:** Capacidad para detectar fallos en sensores individuales al visualizar cuando uno de estos no cambia de estado en el tiempo.

### **Ventajas identificadas:**

- Diagnóstico temprano de fallos en el sistema de posición.
- Verificación del correcto funcionamiento del controlador.
- Monitoreo no invasivo del sistema de conmutación.
- Alta velocidad de respuesta mediante interrupciones.

### **Limitaciones y mejoras propuestas:**

- Implementación de análisis de secuencias para detección de patrones erróneos.
- Registro histórico de eventos para análisis de tendencias.

## Medición de Corrientes por Fase

El sistema de monitoreo para medición de corrientes implementado con la función de los sensores ACS712-30A demostró un rendimiento satisfactorio dentro de los parámetros operacionales del motor:

- **Precisión promedio:** Error del 1.97 % respecto a mediciones de referencia con amperímetro.
- **Rango dinámico:** Adecuado para corrientes de 0.5A a 10A.
- **Respuesta en frecuencia:** Suficiente para señales de 300Hz del motor BLDC.
- **Filtrado efectivo:** Reducción eficaz de ruido de conmutación.

### Ventajas identificadas:

- Medición directa y en serie para alta precisión.
- Filtrado digital implementado para mejorar la visualización en la calidad de señal.
- Calibración por software para compensación de offsets.

### Limitaciones y mejoras propuestas:

- Mejora del rango dinámico inferior (corrientes  $< 0.5A$ ).
- Adición de protección contra sobrecorrientes transitorias.

## Medición de Voltajes Fase-Fase

El sistema de medición de voltajes implementado con sensores ZMPT101B mostró un desempeño notable considerando las limitaciones de frecuencia de los sensores frente a la frecuencia del voltaje de corriente alterna presente en las fases del motor:

- **Precisión destacada:** Error promedio de solo 0.66 %

- **Rango operativo:** Compatible con los 54 VRMS medidos.
- **Adaptación a frecuencia:** Funcionamiento adecuado a 300Hz mediante ajuste de software.
- **Aislamiento:** Seguridad eléctrica garantizada.

#### **Ventajas identificadas:**

- Alta precisión en medición de voltajes AC.
- Aislamiento de alta tensión incluido en los sensores.
- Calibración individual por canal para máxima precisión.
- Filtrado adaptativo para la reducción de perturbaciones y ruido presentes en la medición.

#### **Limitaciones y mejoras propuestas:**

- Selección de sensores con mejor respuesta en frecuencia para aplicaciones BLDC.
- Mejora del sistema de calibración para mayor repetibilidad.
- Adición de medición de factor de potencia y armónicos.

### **Contribuciones Técnicas**

El sistema desarrollado representa varias contribuciones técnicas significativas en el ámbito del monitoreo de motores BLDC. En primer lugar, se implementó con éxito una arquitectura distribuida donde múltiples microcontroladores (Raspberry Pi Pico W) trabajan de forma coordinada, permitiendo una adquisición de datos eficiente. Esta distribución de funciones entre dispositivos especializados optimizó el procesamiento dando pauta para reducir la carga computacional individual.

En segundo lugar, se diseñó un sistema de comunicación robusto basado en protocolo UDP, incorporando mecanismos de verificación de integridad de datos y monitoreo continuo del estado online de cada nodo, garantizando la confiabilidad de la transmisión inalámbrica.

La tercera contribución destacada es el desarrollo de una interfaz gráfica intuitiva y profesional que presenta la información en el tiempo de manera clara y accesible, facilitando la interpretación de datos incluso para usuarios no especializados. Se implementaron algoritmos de filtrado que fueron adaptados específicamente a las características de cada tipo de señal (binaria, corriente, voltaje RMS), mejorando la calidad de los datos adquiridos. Finalmente, se estableció un sistema de calibración por software que permitió mejorar la precisión de las mediciones sin requerir implementar hardware adicional, demostrando que mediante algoritmos inteligentes es posible superar algunas limitaciones de los componentes seleccionados.

## **Ventajas del Sistema Implementado**

### **Ventajas Técnicas**

El sistema presenta múltiples ventajas que justifican su diseño y aplicación. La modularidad al diseño permite que cada subsistema funcione independientemente, facilitando tanto el mantenimiento como la posible expansión futura. Esta característica se complementa con una notable escalabilidad, ya que la arquitectura permite agregar nuevos sensores o variables de monitoreo con mínimas modificaciones al sistema existente. Otra ventaja es la portabilidad del sistema, basado completamente en hardware de código abierto y ampliamente disponible.

### **Desafíos Superados**

Durante el desarrollo del proyecto se enfrentaron y superaron varios desafíos técnicos significativos. El más notable fue la adaptación de frecuencia de los sensores ZMPT101B, que mediante algoritmos de compensación implementados en software lograron operar adecuadamente a 300 Hz a pesar de estar diseñados para 60 Hz. La sincronización temporal entre múltiples dispositivos representó otro desafío importante, resuelto mediante protocolos de timestamp y mecanismos de ajuste de reloj. El procesamiento en el tiempo en hardware de recursos limitados requirió optimizaciones cuidadosas de los algoritmos. Finalmente, la compatibilidad eléctrica entre diferentes niveles de señal se resolvió mediante circuitos de acondicionamiento específicos para cada tipo de sensor.

## Conclusiones Finales

El sistema de monitoreo remoto desarrollado para el motor BLDC de la scooter eléctrica Kee-way Ride E1 ha demostrado ser una solución viable, económica y efectiva para la adquisición y visualización en el tiempo de las variables críticas del sistema de propulsión. A través de una implementación cuidadosa y metódica, se logró:

- **Integración exitosa** de múltiples tecnologías de sensado y comunicación.
- **Precisión satisfactoria** en todas las variables monitoreadas, con errores promedio menores al 2 %.
- **Interfaz intuitiva** que facilita la interpretación de datos por parte del operador.
- **Arquitectura escalable** que permite futuras expansiones y mejoras.

Los resultados obtenidos validan la hipótesis inicial de que es posible implementar un sistema de monitoreo remoto, basado en una red de microcontroladores Raspberry Pi Pico W y una unidad central Raspberry Pi 4, que permita la adquisición, filtrado digital y visualización de variables eléctricas esenciales del motor BLDC en una scooter eléctrica, con el fin de verificar su correcto funcionamiento.

Finalmente, este proyecto contribuye al campo de los vehículos eléctricos al proporcionar una herramienta accesible para el monitoreo y diagnóstico de sistemas de propulsión BLDC, facilitando tanto el mantenimiento preventivo como la optimización del rendimiento de vehículos eléctricos de dos ruedas.

## REFERENCIAS BIBLIOGRÁFICAS

- Ahmad, F., Iqbal, A., Ashraf, I., Marzband, M., & Khan, I. (2022, noviembre). Optimal location of electric vehicle charging station and its impact on distribution network: A review. <https://doi.org/10.1016/j.egy.2022.01.180>
- Alanazi, M., Mahmood, A., & Chowdhury, M. J. M. (2023, febrero). SCADA vulnerabilities and attacks: A review of the state-of-the-art and open issues. <https://doi.org/10.1016/j.cose.2022.103028>
- Alimi, O. A., Ouahada, K., Abu-Mahfouz, A. M., Rimer, S., & Alimi, K. O. A. (2021, septiembre). A review of research works on supervised learning algorithms for scada intrusion detection and classification. <https://doi.org/10.3390/su13179597>
- Atorino, J. I., Bortolín, L. P., Rodríguez, E., Farías, R. O., & Rodríguez, E. E. (2009). Experimentos con un sensor de efecto Hall. *Am. J. Phys. Educ*, 3. <http://www.journal.lapen.org.mx>
- Buldakova, T. I. (2020). Cybersecurity Risks Analyses at Remote Monitoring of Object's State. Springer International Publishing. [https://doi.org/10.1007/978-3-030-32648-7\\_15](https://doi.org/10.1007/978-3-030-32648-7_15)
- Cando, J. R. C. (2023). IMPLEMENTACIÓN DE UN SISTEMA SCADA EN EL ÁMBITO DE LA INDUSTRIA 4.0 E IOT.
- Daniel, H., & José, R. (2020). DISEÑO DE UN SISTEMA DE MEJORA DE AUTONOMÍA PARA UN VEHÍCULO ELÉCTRICO.
- Ehsani, M., Singh, K. V., Bansal, H. O., & Mehrjardi, R. T. (2021). State of the Art and Trends in Electric and Hybrid Electric Vehicles. *Proceedings of the IEEE*, 109, 967-984. <https://doi.org/10.1109/JPROC.2021.3072788>
- Hussein, H. M., Ibrahim, A. M., Taha, R. A., Rafin, S. M. H., Abdelrahman, M. S., Kharchouf, I., & Mohammed, O. A. (2024, septiembre). State-of-the-Art Electric Vehicle Modeling: Architectures, Control, and Regulations. <https://doi.org/10.3390/electronics13173578>
- Javier, R., & Alvaro, T. (2022). ANÁLISIS DE AUTONOMÍA DE BATERÍA Y CICLOS DE CONDUCCIÓN EN UN VEHÍCULO ELÉCTRICO URBANO, EN CIUDADES DE ALTURA.

- José, G. (2020). DESARROLLO DE UNA INTERFAZ GRÁFICA PARA LA IMPLEMENTACIÓN DE MODELOS DE OPTIMIZACIÓN ROBUSTA DE PORTAFOLIOS.
- Justindhas, Y., & Jeyanthi, P. (2022). Attack detection and prevention in IoT-SCADA networks using NK-classifier. *Soft Computing*, 26, 6811-6823. <https://doi.org/10.1007/s00500-022-06921-3>
- Küçüköğlu, İ., Dewil, R., & Cattrysse, D. (2021). The electric vehicle routing problem and its variations: A literature review.
- López, H. S. N. (2020). Implementación de interfaces gráficas en Python guiadas a la domótica.
- Madaram, V. G., Biswas, P. K., Sain, C., Thanikanti, S. B., & Balachandran, P. K. (2024, octubre). Advancement of electric vehicle technologies, classification of charging methodologies, and optimization strategies for sustainable development - A comprehensive review. <https://doi.org/10.1016/j.heliyon.2024.e39299>
- Mohammed, S. A. Q., Hakami, S. S., Kassas, M., & Almuhaeni, M. M. (2025). Critical Review on Powertrain Technologies for Electric Vehicles: Classification, Broadly Adopted Topologies, and Future Challenges. *IEEE Access*, 13, 71755-71772. <https://doi.org/10.1109/ACCESS.2025.3561577>
- Mohanraj, D., Arulavid, R., Verma, R., Sathiyasekar, K., Barnawi, A. B., Chokkalingam, B., & Mihet-Popa, L. (2022). A Review of BLDC Motor: State of Art, Advanced Control Techniques, and Applications. <https://doi.org/10.1109/ACCESS.2022.3175011>
- Molina, Y. A., Ramírez, S. A. S., Morales, J. E. G., Reyes, A. N. M., Sánchez, R. G., & García, I. V. V. (2020). Diseño y desarrollo de un sistema de monitoreo remoto implementando Internet de las cosas Design and Development of a Remote Monitoring System Implementing the Internet of Things.
- Ortiz, C. (2022). Desarrollo de una interfaz gráfica HMI en Python para parametrización y manejo de cargas electrónicas serie BK8500.
- Osman, F. A., Hashem, M. Y., & Eltokhy, M. A. (2022). Secured cloud SCADA system implementation for industrial applications. *Multimedia Tools and Applications*, 81, 9989-10005. <https://doi.org/10.1007/s11042-022-12130-9>

- Oulasvirta, A., Dayama, N. R., Shiripour, M., John, M., & Karrenbauer, A. (2020). Combinatorial Optimization of Graphical User Interface Designs. *Proceedings of the IEEE*, 108, 434-464. <https://doi.org/10.1109/JPROC.2020.2969687>
- Pedro, P. (2020). PaperTUI: Sistema tangible para el prototipado de interfaces gráficas de usuario.
- Picho, C., & Isaac, W. (2023). Investigación e implementación de un sistema de carga regenerativo para elevar las prestaciones de autonomía en una motocicleta eléctrica.
- Rakas, S. V., Stojanovic, M. D., & Markovic-Petrovic, J. D. (2020). A review of research work on network-based SCADA intrusion detection systems. <https://doi.org/10.1109/ACCESS.2020.2994961>
- Ravi, S. S., & Aziz, M. (2022, enero). Utilization of Electric Vehicles for Vehicle-to-Grid Services: Progress and Perspectives. <https://doi.org/10.3390/en15020589>
- Romero, A., Tutores, A., Vega-Leal, A. P., & Pérez, S. V. (2020). Diseño y simulación de un cargador de batería para una motocicleta eléctrica.
- Sanguesa, J. A., Sanz, V. T., Garrido, P., Martinez, F. J., & Barja, J. M. M. (2021, marzo). A review on electric vehicles: Technologies and challenges. <https://doi.org/10.3390/smartcities4010022>
- Tuitise, T., Patricio, A., Luigi, I. C., Freire, O., & -Ecuador, M. L. (2020). UNIVERSIDAD TÉCNICA DE COTOPAXI FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS CARRERA DE INGENIERÍA ELECTROMECAÁNICA PROPUESTA TECNOLÓGICA Proyecto de Titulación presentado previo a la obtención del Título de Ingeniero en Electromecánica.
- Upadhyay, D., & Sampalli, S. (2020). SCADA (Supervisory Control and Data Acquisition) systems: Vulnerability assessment and security recommendations. *Computers and Security*, 89. <https://doi.org/10.1016/j.cose.2019.101666>
- Velandia, J. E. G., López, K. C. P., Velandia, L. N. G., & Mateus, J. J. R. (2022). Análisis del diseño de interfaces gráficas de usuario para el control y monitorización de sistemas optoelectrónicos aplicados a telecomunicaciones e instrumentación electrónica. *Entre ciencia e ingeniería*, 16, 33-42. <https://doi.org/10.31908/19098367.2686>

Verena, M., Johann, H. L., & Anne, R. (2021). *Proceedings of 2021 IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe) : 18-21 October, 2021.*

Yadav, G., & Paul, K. (2020). Architecture and Security of SCADA Systems: A Review.  
<http://arxiv.org/abs/2001.02925>