



**Proyecto Previo a la Obtención del Título de Ingeniero
Automotriz**

INGENIERÍA AUTOMOTRIZ

Autor: Franklin Junior Fienco Parrales

Tutor: Ing. Erasmo García Ochoa, M.Sc.

**Elaboración de un Sensor IBS (Intelligent Battery Sensor) con
Arduino Para Baterías Plomo-Ácido de 12 Voltios**

Certificado de Autoría

Yo, Franklin Junior Fienco Parrales, declaro bajo juramento, que el trabajo aquí descrito es de mi autoría; que no ha sido presentado anteriormente para ningún grado o calificación profesional y que se ha consultado la bibliografía detallada. Cedo mis derechos de propiedad intelectual a la Universidad Internacional del Ecuador, para que sea publicado y divulgado en internet, según lo establecido en la Ley de Propiedad Intelectual, reglamento y leyes.

Franklin Junior Fienco Parrales

C.I.: 0954240081

Aprobación del Tutor

Yo, Erasmo Garcia Ochoa certifico que conozco al autor del presente trabajo siendo responsable exclusivo tanto de su originalidad y autenticidad, como de su contenido.

Ing. Erasmo Garcia Ochoa, M.Sc.

C.I.: 0917118697

Director de Proyecto

Dedicatoria

A mis padres, por ser mi pilar en cada etapa de este camino. Gracias por su amor incondicional, su guía silenciosa y por enseñarme, con el ejemplo diario, que el esfuerzo y la constancia abren las puertas del futuro.

A mis hermanos, por acompañarme incluso en los momentos más difíciles, y por hacerme sentir que nunca estuve solo.

A mis compañeros de colegio y de carrera, quienes empezaron este viaje académico conmigo y con quienes compartí aprendizajes, desafíos, risas y crecimiento. Vernos avanzar juntos y convertirnos en profesionales ha sido uno de los mayores regalos de esta etapa.

A Victoria Flores, quien en el transcurso de la carrera fue un pilar fundamental y una fuente constante de motivación. Gracias por estar ahí en los momentos clave, por tu apoyo incondicional, y por ayudarme a crecer no solo como profesional, sino también como persona.

Esta meta no la alcancé solo. A todos ustedes, gracias por caminar conmigo.

Franklin Junior Fienco Parrales

Agradecimiento

Agradezco, en primer lugar, a Dios, por brindarme la salud, la fortaleza y la sabiduría necesarias para avanzar y superar cada etapa de esta carrera.

A mis padres y hermanos, por su amor incondicional, por ser mi apoyo en los momentos más desafiantes y por confiar siempre en mis capacidades.

A mis profesores, quienes a lo largo de cada semestre compartieron no solo sus conocimientos, sino también su experiencia y vocación, motivándome a ser mejor estudiante y futuro profesional.

Al Ingeniero Erasmo García, por su guía, dedicación y paciencia durante esta etapa final de titulación. Su acompañamiento fue clave para lograr culminar este proyecto con éxito.

A todos quienes, de una u otra forma, formaron parte de este camino, gracias por haber contribuido a mi crecimiento personal y académico.

Franklin Junior Fienco Parrales

Índice General

Certificado de Autoría	iii
Aprobación del Tutor.....	iv
Dedicatoria.....	v
Agradecimiento	vi
Índice General.....	vii
Índice de Figuras	x
Índice de Tablas.....	xii
Índice de Ecuaciones	xiii
Resumen	xiv
Abstract.....	xv
Capítulo I	1
Antecedentes.....	1
1.1 Tema de Investigación	1
1.2 Planteamiento, Formulación y Sistematización del Problema.....	1
1.2.1 <i>Planteamiento del Problema</i>	2
1.2.2 <i>Formulación del Problema</i>	2
1.2.3 <i>Sistematización del Problema</i>	3
1.3 Objetivos de la Investigación.....	3
1.3.1 <i>Objetivo General</i>	3
1.3.2 <i>Objetivos Específicos</i>	3
1.4 Justificación y Delimitación de la Investigación	4
1.4.1 <i>Justificación Teórica</i>	5
1.4.2 <i>Justificación Metodológica</i>	7
1.4.3 <i>Justificación Práctica</i>	8

1.4.4	<i>Delimitación Temporal</i>	8
1.4.5	<i>Delimitación Geográfica</i>	9
1.4.6	<i>Delimitación del Contenido</i>	9
Capítulo II.....		11
Marco Referencial		11
2.1	Marco Teórico.....	11
2.1.1	<i>Conceptos Preliminares</i>	13
2.1.2	<i>Sensor Inteligente de Batería (IBS)</i>	13
2.1.3	<i>Funcionamiento del Sensor Inteligente de Batería</i>	14
2.1.4	<i>Parámetros Monitoreados por el IBS</i>	15
2.1.5	<i>Estado de Carga (SOC - State of Charge)</i>	16
2.1.6	<i>Estado de Salud (SOH - State of Health):</i>	16
2.1.7	<i>Plataforma Arduino</i>	17
2.2	Marco Conceptual.....	18
2.2.1	<i>Implementación del Sensor IBS con Arduino</i>	18
2.2.2	<i>Arduino Nano</i>	19
2.2.3	<i>Conversor Analógico Digital ADS1115</i>	20
2.2.4	<i>Sensor de Voltaje FZ0340</i>	21
2.2.5	<i>Resistencia Shunt</i>	22
2.2.6	<i>Sensor de Temperatura DS18B20</i>	23
2.2.7	<i>Cálculo del Estado de Carga (SOC)</i>	24
2.2.8	<i>Cálculo del Estado de Salud (SOH)</i>	25
Capítulo III		27
Implementación Física del Sensor IBS con Arduino.....		27
3.1	Esquema Electrónico de la Implementación Física	27

3.2	Conexión de Componentes	28
3.2.1	<i>Resistencia Shunt a Módulo ADC ADS1115</i>	28
3.2.2	<i>Módulo ADC ADS1115 a Arduino Nano</i>	28
3.2.3	<i>Sensor de Temperatura DS18B20 a Arduino Nano</i>	29
3.2.4	<i>Sensor de Voltaje FZ0340 a Modulo ADC ADS1115</i>	29
3.3	Programación del Sistema.....	29
3.3.1	<i>Configuración de Ganancias del Módulo ADS1115</i>	29
3.3.2	<i>Voltaje Sensor IBS</i>	31
3.3.3	<i>Corriente Sensor IBS</i>	32
3.3.4	<i>Temperatura Sensor IBS</i>	33
3.3.5	<i>Estado de Carga Sensor IBS</i>	35
3.3.6	<i>Estado de Salud del Sensor IBS</i>	36
3.4	Encapsulado del IBS	39
	Capítulo IV	40
	Análisis de Resultados.....	40
4.1	Instalación del Sensor IBS en el Vehículo.....	40
4.2	Muestra de Voltaje	43
4.3	Muestra de Corriente.....	44
4.4	Muestra de Temperatura	46
4.5	Muestra de Estado de Carga (SOC)	47
4.6	Muestra de Estado de Salud (SOH)	49
	Conclusiones.....	51
	Recomendaciones	52
	Bibliografía	53

Índice de Figuras

Figura 1 <i>Ejemplo de un Sensor Inteligente de Batería Automotriz de 12V</i>	4
Figura 2 <i>Modelo de Arduino con Conexión a una Pantalla LCD para Visualización de Información</i>	6
Figura 3 <i>Esquema Inicial de Cómo se Vería el Sensor IBS con Arduino</i>	12
Figura 4 <i>Diagrama General de Conexión del Sensor Inteligente de Batería</i>	14
Figura 5 <i>Esquema de Pines de un Arduino Nano</i>	18
Figura 6 <i>Módulo ADC ADS1115</i>	21
Figura 7 <i>Sensor de Voltaje FZ0340</i>	21
Figura 8 <i>Esquema de Ubicación de la Resistencia Shunt</i>	22
Figura 9 <i>Sensor de Temperatura DS18B20</i>	24
Figura 10 <i>Esquema Electrónico del Sensor IBS</i>	27
Figura 11 <i>Código de Programación para Sensor de Voltaje</i>	31
Figura 12 <i>Código de Programación para Corriente del Sensor IBS</i>	32
Figura 13 <i>Código de Programación para el Sensor de Temperatura DS18B20</i>	34
Figura 14 <i>Código de Programación para el Cálculo del SOC según Voltaje</i>	35
Figura 15 <i>Código de Programación para SOC por Consumo de Corriente</i>	36
Figura 16 <i>Código de Programación para el SOH por Consumo de Corriente</i>	37
Figura 17 <i>Código de Programación del SOH por Caída de Voltaje</i>	38
Figura 18 <i>Vehículo Chevrolet Dmax 2014 para Prueba de Batería con Sensor IBS</i>	40
Figura 19 <i>Habitáculo de Motor con el Sensor IBS Instalada en la Batería de 12V</i>	41
Figura 20 <i>Sensor FZ0340 Físico con su Cableado</i>	41
Figura 21 <i>Sensor FZ0340, Módulo ADS1115 y Arduino Nano conectados</i>	42
Figura 22 <i>Módulo ADS1115 Conectado Junto al Arduino Nano</i>	42
Figura 23 <i>Arduino Nano con sus Conexiones</i>	42

Figura 24 Pantalla LCD 2004 Instalada Con el Sistema IBS.....	43
Figura 25 Sensor de Temperatura DS18B20 Con sus Cableado al Arduino Nano	43
Figura 26 Pantalla LCD del Sensor IBS con Medición de Voltaje.....	44
Figura 27 Voltaje con Multímetro KaiWeets	44
Figura 28 Pantalla LCD del Sensor IBS con Medición de Corriente con Luces Bajas y Medias.....	45
Figura 29 Corriente de Comparación (1) con Multímetro KaiWeets	45
Figura 30 Pantalla LCD del Sensor IBS con Medición de Corriente con Luces Altas	46
Figura 31 Corriente de Comparación (2) con Multímetro KaiWeets	46
Figura 32 Pantalla LCD del Sensor IBS con Medición de Temperatura	47
Figura 33 Pantalla LCD del Sensor IBS con Medición de SOC.....	48
Figura 34 Comprobación de Estado de Batería con Equipo TOPDON	48
Figura 35 Pantalla LCD con Valores de Estado de Salud SOH.....	49
Figura 36 Impresión de Estado de Batería con Equipo TOPDON	50

Índice de Tablas

Tabla 1 <i>Descripción de los Puntos de la Figura 4</i>	15
Tabla 2 <i>Especificaciones del Arduino Nano</i>	19
Tabla 3 <i>Especificaciones del ADS1115</i>	20
Tabla 4 <i>Especificaciones del Sensor FZ0340</i>	22
Tabla 5 <i>Especificaciones del Sensor de Temperatura DS18B20</i>	23
Tabla 6 <i>Tabla de Voltajes de Referencia para el Cálculo del SOC</i>	24
Tabla 7 <i>Conexión de pines de Shunt a Módulo ADS1115</i>	28
Tabla 8 <i>Conexión de Módulo ADS1115 a Arduino Nano</i>	28
Tabla 9 <i>Conexión de Sensor Temperatura DS18B20 a Arduino Nano</i>	29
Tabla 10 <i>Conexión de Sensor de Voltaje FZ0340 a Módulo ADS1115</i>	29
Tabla 11 <i>Configuración de Ganancias del Módulo ADS1115</i>	30
Tabla 12 <i>Descripción de Variables y Constantes del Código para Corriente del Sensor IBS33</i>	
Tabla 13 <i>Detalles Técnicos del Sensor de Temperatura DS18B20</i>	34

Índice de Ecuaciones

Ecuación 1 <i>Integración de la Corriente en el Tiempo</i>	25
Ecuación 2 <i>Estado de Carga en Función del Consumo de Corriente</i>	25
Ecuación 3 <i>Estado de Salud en Función de la Variación del Voltaje</i>	26
Ecuación 4 <i>Estado de Salud en Función de la Capacidad de la Batería</i>	26

Resumen

Este proyecto presenta el diseño e implementación de un Sistema Inteligente de Monitoreo de Baterías (IBS) para baterías de plomo-ácido de 12V, utilizando la plataforma Arduino Nano. El sistema integra sensores para medir en tiempo real el voltaje, la corriente y la temperatura de la batería, y emplea una pantalla LCD 20x4 para la visualización de datos. Para la estimación del Estado de Carga (SOC), se implementaron dos métodos: uno basado en el voltaje y otro mediante el conteo de Coulomb, integrando la corriente a lo largo del tiempo. El Estado de Salud (SOH) se calcula aplicando las siguientes consideraciones: la relación entre el consumo de corriente y la variación del SOC, y el análisis de la caída de voltaje bajo carga controlada. El sensor inteligente de batería fue instalado y validado en una camioneta Chevrolet DMax 2014, demostrando precisión y confiabilidad en condiciones reales de descarga, carga y degradación de la batería. Los resultados evidencian la viabilidad y el potencial de los sistemas IBS desarrollados con hardware de código abierto, facilitando la experimentación, el diagnóstico y el aprendizaje en instrumentación automotriz.

Palabras Clave: Sensor Inteligente de Batería (IBS), Estado de Carga (SOC), Estado de Salud (SOH).

Abstract

This project presents the design and implementation of an Intelligent Battery Monitoring System (IBS) for 12V lead-acid batteries, using the Arduino Nano platform. The system integrates sensors to measure battery voltage, current, and temperature in real time, and uses a 20x4 LCD screen for data visualization. For State of Charge (SOC) estimation, two methods were implemented: one based on voltage and another using Coulomb counting, which integrates current over time. The State of Health (SOH) is calculated using two complementary approaches: the ratio between current consumption and SOC variation, and the analysis of voltage drop under controlled load. The prototype was installed and validated in a 2014 Chevrolet DMax, demonstrating accuracy and reliability under real-world conditions of battery discharge, charge, and degradation. The results highlight the feasibility and potential of IBS systems developed with open-source hardware, facilitating experimentation, diagnosis, and learning in automotive instrumentation.

Keywords: Intelligent Battery Sensor (IBS), State of Charge (SOC), State of Health (SOH).

Capítulo I

Antecedentes

1.1 Tema de Investigación

Elaboración de un sensor IBS (Intelligent Battery Sensor) con Arduino para baterías plomo-ácido de 12 voltios.

1.2 Planteamiento, Formulación y Sistematización del Problema

Conocer información importante como el estado de carga, y salud de la batería de 12 voltios en los vehículos es crucial para garantizar su buen funcionamiento. Esta información permite calcular de manera precisa el estado de carga (SOC) y el estado de salud (SOH), lo que garantiza que el sistema eléctrico funcione de manera eficiente (Lohmeier & Veik, 2021).

El sensor inteligente de batería por sus siglas en inglés (IBS) monitorea el desempeño de la batería brindando información como corriente, voltaje e incluso temperatura. Con esto se busca obtener datos para el entendimiento del correcto desempeño de la batería y el sistema de gestión de carga, y como resultado extender la vida útil de estos componentes.

Sin embargo, una parte considerable del parque automotriz, especialmente algunos vehículos de gama media o más antiguos no cuentan con este tipo de sensores, dejando a los usuarios sin información crítica para el mantenimiento de la batería de 12V.

Ante esta necesidad, un sistema IBS compatible con Arduino permite a estos vehículos contar con una tecnología alternativa que es económica y funcional, que puede dar como resultado mejor sistema de gestión de energía con su bajo costo y desarrollo.

1.2.1 Planteamiento del Problema

La electrónica en la industria automotriz ha tenido una evolución significativa, enfocada hacia el desarrollo de vehículos más eficientes, inteligentes y sostenibles. Ante este punto, la batería automotriz de 12V sigue desempeñando un papel importante para cada componente eléctrico y electrónico.

La correcta gestión energética de la batería de 12V es crucial en vehículos que no tienen algunos sistemas de monitoreo inteligentes, ya que los fallos de batería siguen siendo una de las principales razones de las averías. Esta ausencia resulta en que los conductores no pueden detectar el deterioro o problemas de la batería de manera oportuna, lo que resulta en una vida útil de batería más corta y gastos imprevistos.

En este contexto, se propone la aplicación de un sistema IBS desarrollado con Arduino, que permite la monitorización de voltaje, corriente y temperatura como una solución práctica y de bajo costo que se puede aplicar a vehículos que no disponen de esta tecnología. Se espera que la ejecución del presente proyecto permita el diagnóstico preventivo de fallos de batería, y mitigue las consecuencias del desconocimiento al tener información crucial para la rápida acción ante los fallos de este componente.

1.2.2 Formulación del Problema

¿Cómo puede desarrollarse un sistema IBS replicado con Arduino como una alternativa tecnológica y funcional para monitorear el estado de baterías automotrices de 12V de plomo-ácido en vehículos que no cuentan con este sistema, con el fin de extender su vida útil y prevenir problemas eléctricos?

1.2.3 Sistematización del Problema

- ¿Cuáles son las principales causas que contribuyen a la disminución de la vida útil de las baterías automotrices y cómo puede un sensor IBS monitorearlas de manera efectiva?
- ¿Qué métodos y análisis matemáticos son más eficaces para determinar el estado de carga (SOC) y el estado de salud (SOH) de una batería utilizando una plataforma Arduino?
- ¿Qué beneficios prácticos puede ofrecer el uso de un sensor IBS desarrollado con Arduino en comparación equipos de diagnóstico comerciales existentes en función de costo, y precisión?

1.3 Objetivos de la Investigación

1.3.1 Objetivo General

- Elaborar un sensor IBS (Intelligent Battery sensor) con Arduino para baterías plomo-ácido de 12 voltios.

1.3.2 Objetivos Específicos

- Diseñar los circuitos electrónicos que permitan la integración de sensores para voltaje, corriente y temperatura.
- Desarrollar un algoritmo para el estado de carga (SOC) y el estado de salud (SOH) de la batería.
- Comprobar la implementación del sensor IBS en la visualización de datos en tiempo real de los parámetros de SOC, SOH, voltaje, corriente y temperatura.

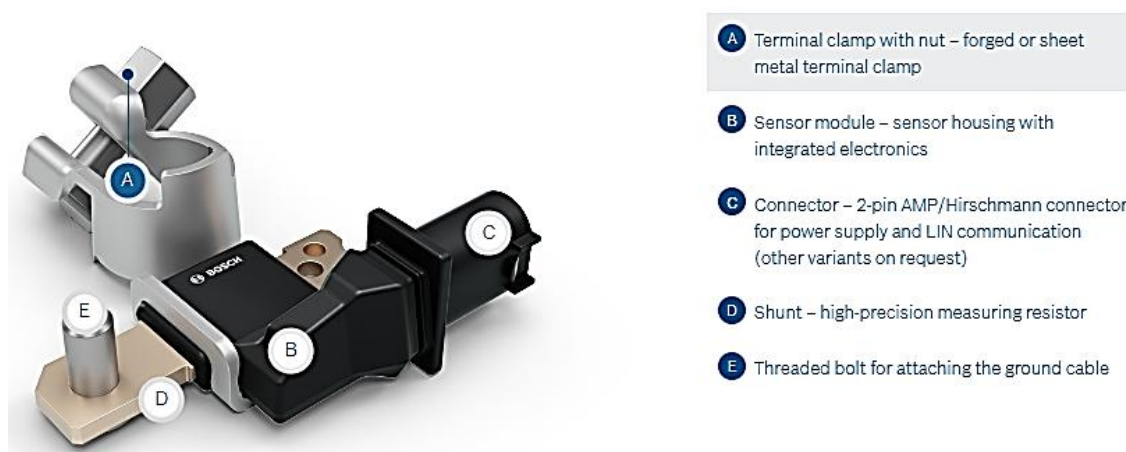
1.4 Justificación y Delimitación de la Investigación

El presente proyecto busca responder a una necesidad dentro del parque automotriz, particularmente en vehículos que carecen de sistemas inteligentes de monitoreo de baterías. Las baterías de 12V son imprescindibles en la alimentación de una variedad de sistemas electrónicos, pero debido a la ausencia de sensores específicos como el IBS (Sensor Inteligente de Batería), no siempre se gestionan de manera efectiva.

Con la ayuda de este sensor, se pueden recopilar y procesar en tiempo real parámetros como el voltaje, la corriente y la temperatura. El análisis de estos datos se utiliza para evaluar el estado de carga (SOC) y la salud (SOH) de la batería, para evitar fallos repentinos y prolongar la vida útil de la batería.

Figura 1

Ejemplo de un Sensor Inteligente de Batería Automotriz de 12V



Fuente: (BOSCH, 2025)

Las limitantes son el costo de los sistemas comerciales de IBS y la falta de integración en muchos modelos de vehículos. Por lo tanto, este trabajo presenta el desarrollo de un sistema IBS alternativo y funcional basado en la plataforma Arduino, buscando promover una alternativa de bajo costo, educativa y accesible para estudiantes, técnicos y usuarios de vehículos.

El objetivo principal de Arduino es poder darle diferentes tipos de uso a esta placa. Este hardware y software fue desarrollado con el fin crear proyectos a un bajo costo, siendo para el presente proyecto la base de lo que se necesita como alternativa, ya que por el amplio campo de desarrollo nos permite ejecutar la programación y adaptación por los diferentes sensores y gadgets que la plataforma tiene disponible (UAMedia, 2021).

1.4.1 Justificación Teórica

La justificación teórica del presente proyecto se sustenta en la necesidad de contar con sistemas de monitoreo alternativos que permitan gestionar eficientemente el estado de las baterías automotrices de 12V. Se ha demostrado que la cuantificación y la gestión adecuada de estos parámetros mejoran el rendimiento energético, evitando fallos eléctricos. Sin embargo, los sensores IBS comerciales son limitados por su costo o equivalencia en aplicaciones de acuerdo con la marca y modelo de vehículos.

Para conservar cuidadosamente la energía de la batería del vehículo es necesario conocer el estado de carga, el envejecimiento y cualquier cambio que exista, ya que las baterías débiles son la causa principal de avería del vehículo en más del 50% de los casos (HELLA, 2012).

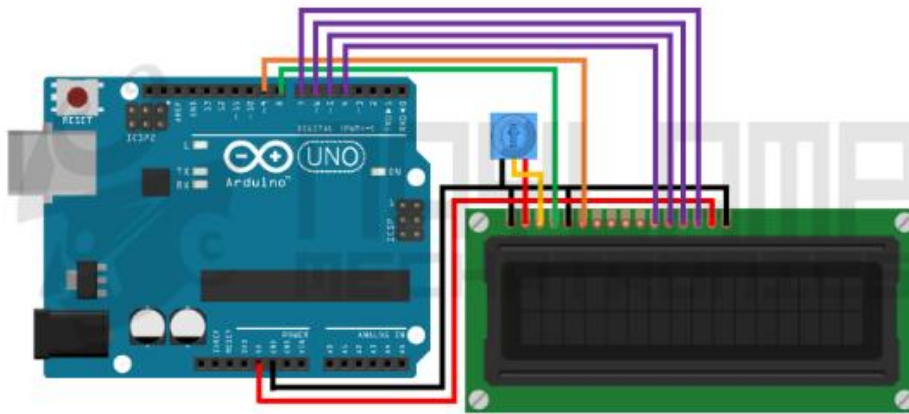
En otras palabras, el presente proyecto está destinado a ser una base teórica para algo que podría replicarse en plataformas abiertas, como Arduino, para el análisis práctico y proporcionar información sobre la batería y también la adquisición de datos eléctricos en tiempo real.

- Esta investigación se centra en el diseño, desarrollo y validación de un sistema IBS replicado para baterías de 12V de plomo-ácido, utilizando la plataforma Arduino. El proyecto se delimita a:
- Componentes utilizados: Se utilizarán sensores para medir voltaje, corriente y temperatura, un microcontrolador Arduino y una pantalla LCD para la visualización de

datos de acuerdo como se muestra en la Figura 2. La selección de componentes se realizará considerando su disponibilidad en el mercado y su compatibilidad con el sistema.

Figura 2

Modelo de Arduino con Conexión a una Pantalla LCD para Visualización de Información



Fuente: (NayLapMechatronic, 2023)

- **Parámetros monitoreados:** Los parámetros clave serán el estado de carga (SOC), el estado de salud (SOH), la temperatura, el voltaje y la corriente de la batería.
- **Aplicaciones:** El sistema está diseñado para pruebas en baterías automotrices de 12V y no contempla otros tipos de baterías como las de litio o de mayor voltaje.
- **Entorno de prueba:** Las pruebas se realizarán en condiciones controladas para evaluar la precisión y confiabilidad del sistema replicado, comparándolo con equipos de diagnósticos comerciales.
- **Adquisición de datos.** El sensor IBS a desarrollarse estará delimitado únicamente a la recolección de datos para que puedan ser vistos desde la propia interface de Arduino o una pantalla LCD integrada al sensor que muestre la información en tiempo real. No se abarca al trato de los datos hacia alguna comunicación con la computadora del vehículo.

Con estas delimitaciones, se busca garantizar que el proyecto sea viable dentro de los recursos disponibles, asegurando resultados concretos y aplicables que aporten valor al área de la ingeniería automotriz. Con esto se iniciaría el desarrollo de otros trabajos de investigación, en donde se pueda crear la comunicación de este sensor IBS a la ECU del motor mediante los diferentes tipos de comunicación para la gestión adecuada del alternador en función de los datos que el sensor IBS obtenga.

1.4.2 Justificación Metodológica

La justificación metodológica del presente proyecto está diseñado con respecto a su desarrollo metodológico lógico. Abarca el diseño del sistema, la construcción física, el modelado matemático y la validación experimental. Con esto se permite la recolección de datos reales para el análisis, lo que puede usarse para demostrar el desempeño y viabilidad del dispositivo.

La metodología consiste en integrar sensores que miden voltaje, corriente y temperatura con un microcontrolador Arduino, junto con una pantalla LCD para la visualización de los datos. Para realizar esto, se ejecutarán pruebas en un entorno controlado, emulando baterías de plomo-ácido de 12V que servirán como base práctica mientras se asegura que el enfoque siga siendo aplicable en escenarios del mundo real. La recopilación de datos cuantitativos durante las pruebas experimentales permitirá evaluar parámetros críticos de estados de la batería, proporcionando una base para posibles comparaciones con equipos de diagnóstico comerciales y validando la precisión del sistema desarrollado. La revisión de literatura técnica y la consulta de manuales de usuario y documentos de fabricantes complementarán el análisis, enriqueciendo la base teórica, técnica y matemática del proyecto.

Este enfoque metodológico asegura la robustez del proyecto, permitiendo no solo demostrar la funcionalidad del sistema, sino también generar información útil para mejorar

futuros diseños y aplicaciones en el campo de la ingeniería automotriz con el uso de la plataforma Arduino.

1.4.3 Justificación Práctica

El presente proyecto proporciona una solución alternativa a un problema tangible: la falta de monitoreo del estado de la batería de 12V en muchos vehículos, especialmente en aquellos que carecen de sistemas IBS instalados de fábrica. Esto puede resultar en fallos eléctricos no deseados, menor vida de la batería y mayores costos de mantenimiento.

Este sistema de monitoreo en tiempo real se crea combinando sensores de voltaje, corriente y temperatura con el microcontrolador Arduino y mostrando los datos en una pantalla LCD. Este sistema de bajo costo se presenta como una posibilidad y una alternativa accesible para técnicos automotrices, estudiantes y propietarios de vehículos que desean optimizar el rendimiento de la batería de sus vehículos.

A nivel práctico, el dispositivo será probado en un entorno controlado con baterías de plomo-ácido de 12V, simulando condiciones reales de operación. Los datos recopilados servirán no solo para validar el funcionamiento del sistema, sino también para demostrar su utilidad como herramienta de diagnóstico preventivo.

Además, su diseño programable permite su futura adaptación a distintos tipos de vehículos o sistemas energéticos, lo que amplía su campo de aplicación y le otorga un valor agregado en el ámbito de la ingeniería automotriz, el mantenimiento vehicular y la formación técnica.

1.4.4 Delimitación Temporal

El proyecto "Elaboración de un Sensor IBS (Intelligent Battery Sensor) con Arduino para Baterías Plomo-Ácido de 12 Voltios" se llevará a cabo en un período de seis meses, comenzando abril de 2025 y finalizando en agosto 2025. Durante este tiempo, se desarrollarán investigaciones teóricas y experimentales para diseñar, implementar y validar el

funcionamiento del sensor replicado, asegurando que los datos obtenidos sean precisos y útiles para el monitoreo del estado de la batería.

1.4.5 Delimitación Geográfica

El desarrollo y validación del proyecto "Elaboración de un Sensor IBS (Intelligent Battery Sensor) con Arduino para Baterías Plomo-Ácido de 12 Voltios" se llevará a cabo en la ciudad de Guayaquil – Ecuador, en las instalaciones de la Universidad Internacional del Ecuador (UIDE), en un entorno controlado para garantizar la precisión de las mediciones y pruebas. La investigación se enfocará en vehículos de pasajeros con sistemas eléctricos de baterías automotrices de 12V plomo-ácido, descartando baterías de otros tipos o voltajes superiores.

1.4.6 Delimitación del Contenido

La delimitación del contenido de esta investigación define los límites del proyecto “Elaboración de un Sensor IBS (Intelligent Battery Sensor) con Arduino para Baterías Plomo-Ácido de 12 Voltios ”, permitiendo que el desarrollo se base en la viabilidad metodológica y técnica.

En consecuencia, el presente proyecto tiene una perspectiva teórico-práctica, es decir, abarca el estudio de los sensores IBS, además de desarrollar e implementar un sistema funcional que permitiría la adquisición de datos pertinentes sobre una batería automotriz de 12V.

Los principales aspectos considerados en esta delimitación son:

- Tipos de baterías: Solo se soportarán baterías de plomo-ácido de 12V, porque es el tipo más común de batería utilizada en la industria automotriz.
- Parámetros para analizar: Voltaje de la batería y corriente, temperatura, estado de carga (SOC) y estado de salud (SOH).

- Componentes electrónicos: Sensores específicos para medir las variables mencionadas anteriormente, un microcontrolador Arduino y una pantalla LCD para visualizar los resultados.
- Entorno de prueba: Las pruebas se llevarán a cabo en un laboratorio o taller controlado.
Elementos fuera del alcance del estudio de investigación:
- Baterías de litio u otros voltajes: No se investigarán sistemas sofisticados de gestión de baterías (BMS) ni baterías con químicas distintas al plomo-ácido.
- Aplicaciones en vehículos eléctricos: El estudio no cubrirá vehículos híbridos o eléctricos debido a las diferencias en la estructura de gestión.
- Análisis de datos avanzados: No se integrarán algoritmos de inteligencia artificial o aprendizaje automático para la predicción de fallos.
- Gestión de comunicación con otros módulos del vehículo: Este estudio solo llega hasta la obtención de datos y no se contempla a la comunicación con alguna módulo o computadora del vehículo para el actuar de los diferentes sistemas que estos componentes puedan controlar.

Capítulo II

Marco Referencial

2.1 Marco Teórico

La siguiente investigación proporciona los conceptos que sustentan el desarrollo de una alternativa de sensor inteligente de batería (IBS) replicado con Arduino para baterías de 12V de plomo-ácido en vehículos. A continuación, se presentan los ejes temáticos para comprender la importancia y viabilidad del presente proyecto.

Importancia del Monitoreo de Baterías en Vehículos Automotrices: Actualmente, el sistema eléctrico de un vehículo depende en gran medida del estado y rendimiento de la batería, esta no solo es responsable del arranque del motor, sino que también alimenta muchos sistemas electrónicos, como la inyección electrónica, y sistemas de seguridad. Con un monitoreo preciso, se pueden prevenir fallas inesperadas y optimizar el rendimiento del vehículo.

Funcionamiento y Principios de los Sensores IBS (Intelligent Battery Sensor): Los sensores IBS son dispositivos electrónicos avanzados que monitorean parámetros clave de la batería, como:

- **Voltaje:** Permite determinar en tiempo real el Estado de Carga (SOC).
- **Amperaje:** Permite medir el flujo de energía para identificar ciclos de carga y descarga.
- **Temperatura:** Los factores térmicos afectan directamente la descarga y la vida útil de la batería.
- **Estado de Salud (SOH):** Indica la capacidad restante de la batería en comparación con la capacidad de diseño.

Implementación de Sensores con Arduino: Arduino es una plataforma versátil y accesible que permite la integración de sensores para la medición de parámetros eléctricos. Para la replicación del sensor IBS, se emplearán los siguientes componentes:

- Microcontrolador, como Arduino UNO o NANO para procesar los datos.
- Sensores de voltaje, como el FZ0430
- Sensor de temperatura, como el DS18B20 para monitoreo térmico.
- Pantalla LCD, para visualización de datos en tiempo real.

Retos y limitaciones en la implementación de sensores IBS con Arduino: Si bien el desarrollo de un sensor IBS basado en Arduino representa una alternativa accesible y funcional, también implica una serie de desafíos técnicos, entre los cuales se incluyen:

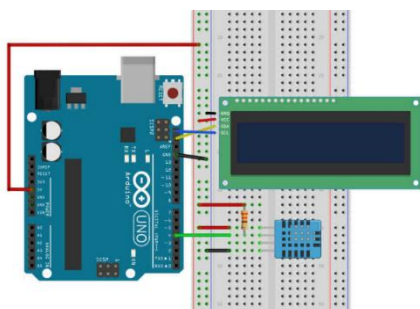
- Precisión en comparación a los equipos de diagnósticos comerciales y sensores de bajo costo.
- Ruido e interferencias en la medición de corriente y voltaje.
- Implementación de algoritmos para cálculo preciso de SOC y SOH.

Estas dificultades se abordarán mediante pruebas experimentales y uso de hardware y software, garantizando un desempeño confiable y replicable.

Este marco teórico proporciona la base conceptual para la investigación, abordando los principios eléctricos y electrónicos necesarios para el diseño y validación de un sensor IBS basado en Arduino, como se muestra en la Figura 3.

Figura 3

Esquema Inicial de Cómo se Vería el Sensor IBS con Arduino



Fuente: (ParaArduino, 2023)

2.1.1 Conceptos Preliminares

En el monitoreo de las baterías automotrices es necesario para asegurar su eficiencia, vida útil y seguridad. Desde esta perspectiva, el Sensor Inteligente de Batería (IBS) tiene un papel clave en la gestión energética de los vehículos modernos, permitiendo conocer la salud y estado de la batería.

El estado de la batería varía con la temperatura, la demanda eléctrica del vehículo y la antigüedad del acumulador. Para prevenir fallas y optimizar la vida útil de la batería, los fabricantes de vehículos han introducido sensores de batería inteligentes (IBS) que permiten medir y estimar el estado de la batería en tiempo real. También tiene diversas aplicaciones en vehículos con sistemas Start-Stop e híbridos, donde la eficiencia energética y la gestión del consumo son importantes.

2.1.2 Sensor Inteligente de Batería (IBS)

El IBS (Intelligent Battery Sensor) es un dispositivo electrónico que se instala en el terminal negativo de la batería y mide con fiabilidad y precisión los parámetros de tensión, corriente y temperatura de la batería. La información sobre el estado de carga (SOC) y el estado de salud (SOH) de la batería se calcula algorítmicamente a partir de estas mediciones (Veratron, 2024).

Su propósito es proporcionar información precisa sobre el desempeño de la batería, facilitando la optimización del sistema de carga.

El sensor IBS recopila datos a través de los módulos electrónicos y envía la información a la Unidad de Control Electrónica (ECU) del vehículo a través de una línea de comunicación LIN bus (Local Interconnect Network) (autodemiaonline.com, 2025). Con esto, se permite al sistema ajustar la carga del alternador, en función de la demanda que se está teniendo en la batería de 12V y prevenir descargas excesivas y mejorar la eficiencia energética del automóvil.

2.1.3 Funcionamiento del Sensor Inteligente de Batería

El IBS (Sensor Inteligente de Batería) está conectado directamente al borne negativo de la batería mediante el terminal de conexión. Junto al terminal, la parte mecánica del sensor de batería consiste en una resistencia llamada shunt, la cuál está conectada a la línea de carga del vehículo y se utiliza como una resistencia de medición para medir la corriente de forma indirecta.

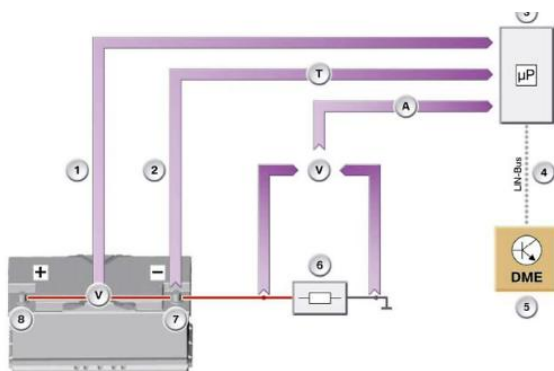
Al estar conectada en serie a la carga, se mide la caída de voltaje (V) y con este valor, junto con el valor conocido de la resistencia (Ohmios), se puede calcular la corriente (I) que está teniendo la batería en ese momento; esto con la ley de Ohm que ya se conoce.

Adicional al circuito de corriente, existen dos circuitos, uno para la medición del voltaje y otro para la toma de temperatura de la batería. Con la información recopilada por los sensores, el microprocesador interno del IBS toma estos datos y gracias a los algoritmos matemáticos se puede calcular el estado de carga (SOC) y el estado de salud (SOH) de la batería.

La electrónica está ubicada en una carcasa sellada con un conector, que funciona como interfaz con el sistema de gestión de energía. El protocolo LIN es la interfaz de comunicación con la unidad de control (ECM).

Figura 4

Diagrama General de Conexión del Sensor Inteligente de Batería.



Fuente: (Bimmerfest, 2015)

Tabla 1*Descripción de los Puntos de la Figura 4*

Índice	Explicación
1	Medición del voltaje de la batería entre el terminal positivo y negativo
2	Medición de temperatura de la batería (T)
3	Microprocesador del sensor inteligente de batería (IBS)
4	Bus de red LIN (Local Interconnect Network) para transmisión de valores
5	Electrónica Digital del Motor (DME) o Electrónica Digital Diésel (DDE)
6	Medición de corriente (A) [indirecta, mediante la caída de voltaje proporcional (V) en la resistencia de medición (shunt)]
7	Terminal negativo de la batería
8	Terminal positivo de la batería

2.1.4 Parámetros Monitoreados por el IBS

El sensor IBS mide diferentes variables esenciales para evaluar el estado y desempeño de la batería.

- **Voltaje (V):** El voltaje es una de las principales medidas utilizadas para determinar el estado de carga (SOC) de la batería. Se mide en voltios (V) y proporciona información sobre la cantidad de energía almacenada.
- **Corriente (A):** El sensor IBS mide el flujo de corriente (expresado en amperios A) que entra y sale de la batería. El análisis de la corriente permite detectar consumos eléctricos fuera de lo normal o fallos en el alternador.
- **Corriente de carga:** Hace referencia a la cantidad de energía suministrada por el alternador para recargar la batería. En un vehículo en funcionamiento, esta corriente suele tener rango entre 5A y 50A, dependiendo del sistema de carga del vehículo.

- Corriente de descarga: Indica cuánta energía está consumiendo el vehículo. Con esta variable podemos calcular el consumo en función del tiempo (Amperios-hora) que está teniendo la batería y de igual manera hacer cálculos para obtener el estado de salud.
- Temperatura (°C): Es un factor crítico en el desempeño y vida útil de la batería. Un sensor de temperatura integrado en el IBS permite detectar variaciones térmicas y ajustar el régimen de carga para evitar daños (Battery University, 2022). En temperaturas mayores a 50°, pueden acelerar la degradación del electrolito y reducir la vida útil de la batería. En cambio, en temperaturas muy menores a 0°C, pueden afectar la capacidad de arranque del motor, ya que el rendimiento de la batería disminuye significativamente en frío.

2.1.5 Estado de Carga (SOC - State of Charge)

El SOC indica cuánta energía está disponible en la batería en un momento determinado. Se expresa en porcentaje (0% - 100%) y permite evaluar si la batería tiene suficiente carga para arrancar el motor y alimentar los sistemas eléctricos (Augeri, 2012).

- Un SOC del 100% indica que la batería está cargada.
- Un SOC por debajo del 50% sugiere que la batería podría estar descargándose.
- Un SOC menor al 20% indica un alto riesgo de fallo, ya que el acumulador no tendrá suficiente energía para arrancar el motor.

La importancia del estado de carga radica en que los Sistemas de Gestión de Baterías (BMS) dependen de este dato para monitorear el nivel energético. Esta información es crucial para evitar descargas profundas que pueden degradar la batería y acortar su vida útil.

2.1.6 Estado de Salud (SOH - State of Health):

El SOH es un parámetro que evalúa la capacidad de la batería en comparación con su estado original de fábrica. Se expresa en porcentaje y mide el desgaste y envejecimiento del acumulador a lo largo del tiempo (Plaza, 2022).

- Un SOH del 100% indica que la batería está en buen estado.
- Un SOH por debajo del 80% sugiere que la batería ha perdido su capacidad original y podría necesitar reemplazo.
- Un SOH menor al 50% indica una batería en estado crítico, con un alto riesgo de fallo.

El SOH se calcula mediante un algoritmo que analiza datos obtenidos de la batería, su historial de carga y descarga, y su capacidad de retención de energía.

2.1.7 Plataforma Arduino

Arduino es una plataforma de desarrollo de código abierto basada en microcontroladores, ampliamente utilizada para proyectos de electrónica aplicada y automatización.

Su uso se justifica por su facilidad de programación, gran comunidad de soporte, y compatibilidad con múltiples sensores y módulos de comunicación (Pedamkar, 2023).

Las características que hacen de Arduino una herramienta versátil para el desarrollo de proyectos que integren varios sensores y componentes electrónicos, como se observa en la Figura 5, son las siguientes:

Características clave:

- Código abierto en hardware y software.
- Permite comunicación serial por USB.
- Requiere solo 5V de alimentación.
- Compatible con señales digitales/analógicas, sensores y actuadores.
- Voltaje recomendado: 7-12V (entrada) / 5V (operación en la mayoría de modelos).

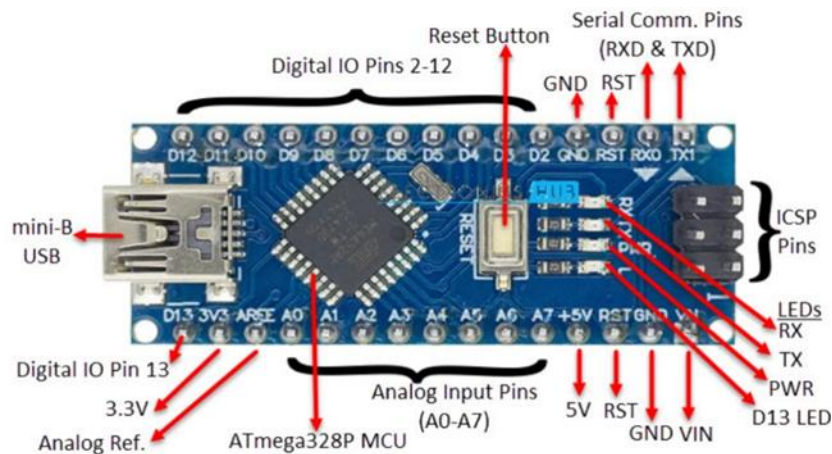
Ámbitos de aplicación más comunes:

- Creación rápida de prototipos.
- Gestión de motores.
- Robótica y sistemas de control.

- Desarrollo de esquemas electrónicos básicos.
- Automatización doméstica e industrial.
- Proyectos autónomos.

Figura 5

Esquema de Pines de un Arduino Nano



Fuente: (NextPCB, 2025)

2.2 Marco Conceptual

2.2.1 Implementación del Sensor IBS con Arduino

En el presente proyecto, se busca replicar un Sensor Inteligente de Batería utilizando la plataforma Arduino como microcontrolador principal, junto con la electrónica amigable y compatible que conlleva. Para ello, se emplean las siguientes partes para su desarrollo:

- Módulo conversor de señal analógica a digital ADS1115
- Sensor FZ0430 para la medición de voltaje
- Resistencia shunt de 400 amperios
- Sensor DS18B20 para la medición de temperatura
- Pantalla LCD 2004 para la visualización en tiempo real de los datos.

2.2.2 *Arduino Nano*

En el presente proyecto, se emplea un Arduino NANO como unidad principal de procesamiento para adquirir, calcular y visualizar los datos de los sensores que replican el sensor IBS.

El uso de Arduino permite una implementación flexible y programable, facilitando la integración de algoritmos para el cálculo del SOC y SOH. Se utilizarán protocolos de comunicación serial para la transferencia de datos entre los sensores y el microcontrolador, asegurando una recolección eficiente y precisa de la información.

Tabla 2

Especificaciones del Arduino Nano

Especificación	Valor
Microcontrolador	ATmega328P (8-bit)
Chip USB	CH340G
Conector USB	mini-USB tipo B
Voltaje de alimentación	7V - 12V DC
Voltaje nivel lógico I/O	5V DC
Pines digitales I/O	14 (6 PWM)
Memoria SRAM	2KB
Memoria EEPROM	1KB
Frecuencia de Reloj	16MHz
Salida de voltaje 3.3V	máx. 1mA
Dimensiones	18.5 x 43.2 mm
Peso	6.5 gramos

Fuente: (NayLampMechatronics, 2023)

2.2.3 Conversor Analógico Digital ADS1115

En el presente proyecto, el módulo ADS1115 es un conversor analógico-digital (ADC) de 16 bits que se comunica con el Arduino mediante I2C.

El protocolo I2C usa dos cables para enviar y recibir datos. SCL (reloj serial), que es una línea que el Arduino activa a intervalos regulares para sincronizar la comunicación, y SDA (datos seriales) que es el cable por el que se transmiten los datos entre los dispositivos conectados (Arduino Docs, 2025).

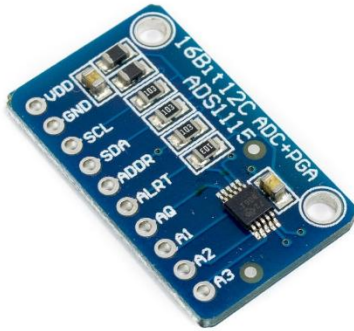
Los conversores ADS111x operan con una tasa de muestreo máxima de 860 muestras/segundo (SPS). Su amplificador programable (PGA) admite rangos de entrada ajustables, desde +/- 256 mV hasta +/- 6,144 V, facilitando mediciones precisas tanto de señales débiles como de mayor amplitud.

Tabla 3

Especificaciones del ADS1115

Especificación	Valor
Voltaje de Alimentación	2.0-5.5 VDC
Corriente de alimentación	200uA (lectura continua)
Interfaz de comunicación	I2C
Chip principal	ADS1115
Muestras por segundo	8 - 860 samples/sec
Canales disponibles	4 individuales / 2 diferenciales
Dirección I2C	0x48 - 0x4B (según PIN ADDR)
Rango de ganancia programable	1-128

Fuente: (NayLampMechatronic, 2023)

Figura 6*Módulo ADC ADS1115*

Fuente: (NayLapMechatronic, 2023)

2.2.4 Sensor de Voltaje FZ0340

Este módulo se basa en el principio de divisor de voltaje en los puntos de resistencia, y puede hacer que el voltaje de entrada se reduzca hasta 5 veces.

Utiliza tecnología de 10 Bits AD, por lo que la resolución de este módulo es de aproximadamente 0.00489V (5V/1023), por lo que el mínimo valor de tensión a detectar es de 0.02445V.

Figura 7*Sensor de Voltaje FZ0340*

Fuente: (FERRETRONICA, 2025)

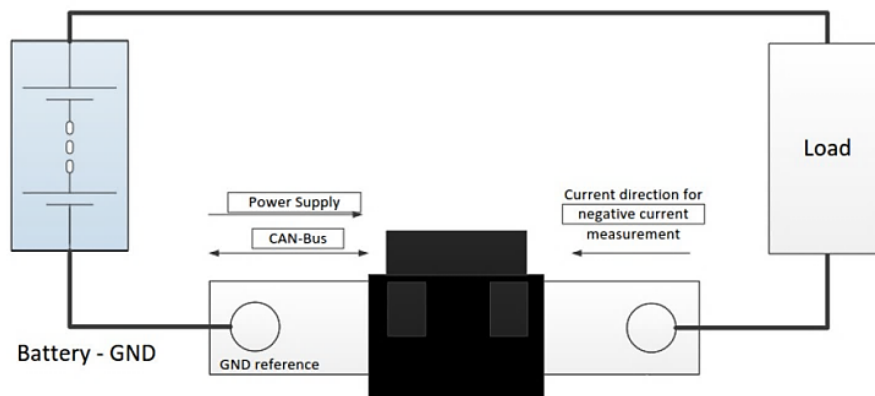
Tabla 4*Especificaciones del Sensor FZ0340*

Especificación	Valor
Rango de alimentación VCC	3.3V ~ 5V
Tipo de salida	Salida Análoga
Rango de voltaje a medir	0.02445V ~ 25V
Resolución	0.00489V
Dimensiones	25 mm x 13 mm

2.2.5 Resistencia Shunt

Es una resistencia con un valor de ohmios muy bajo, para la aplicación del proyecto se utiliza una resistencia de 400A a 75mV, de acuerdo con esta información la resistencia del shunt es de 0,00011875 ohmios.

La resistencia shunt está fabricada principalmente con un material que tiene un bajo coeficiente de temperatura de resistencia. Para la aplicación del presente proyecto se conecta en serie con la carga cuya corriente se quiere medir como se muestra en la Figura 8.

Figura 8*Esquema de Ubicación de la Resistencia Shunt*

Fuente: (Pickering, 2025)

2.2.6 Sensor de Temperatura DS18B20

El DS18B20 es un sensor de temperatura en el cual se lleva la conversión analógica a digital facilitando el resultado de dicha conversión a través de una interfaz digital llamada 1-wire (GeekFactory, 2025). Se encuentra dentro de una cápsula metálica de acero inoxidable, lo que le proporciona resistencia mecánica y protección frente a la humedad.

Como se muestra en la Figura 9. Este puede encontrarse en diversas formas, una de ellas es el que cuenta con una cubierta impermeable de acero inoxidable, la cual nos permite utilizarlo en el presente proyecto ya que evita que existan novedades con factores externos en el funcionamiento normal del vehículo, como sulfato, electrolito de batería o agua.

Tabla 5

Especificaciones del Sensor de Temperatura DS18B20

Especificación	Valor
Tipo de sensor	Sensor digital de temperatura
Protocolo de comunicación	1-Wire
Encapsulado estándar	TO-92
Rango de temperatura	-55°C a 125°C
Resolución	9 a 12 bits (programable)
Dirección única	64 bits (única por sensor)
Métodos de alimentación	Parasite power y alimentación externa
Parasite Power	Sí, usando línea de datos y condensador interno
Alimentación externa recomendada	Sí, vía pin VDD con mayor estabilidad
Número de pines	3 (VDD, GND, DQ)
Encapsulado resistente al agua	Sí, en tubo de acero inoxidable

Figura 9*Sensor de Temperatura DS18B20*

Fuente: (GeekFactory, 2025)

2.2.7 Cálculo del Estado de Carga (SOC)

Para estimar el SOC de la batería, se pueden utilizar los siguientes métodos:

Medición de voltaje: Se compara el voltaje medido con una tabla de referencia de baterías de 12V para estimar el porcentaje de carga, como se muestra en la Tabla 6.

Tabla 6*Tabla de Voltajes de Referencia para el Cálculo del SOC*

Voltaje de Reposo (V)	SOC Estimado (%)
≥ 12.65	100
12.55 - 12.64	90
12.50 - 12.54	80
12.45 - 12.49	75
12.38 - 12.44	70
12.31 - 12.37	60
12.24 - 12.30	50
12.06 - 12.23	25

Fuente: (NaturesGenerator, 2023)

Medición de corriente (Coulomb Counting): Se mide la corriente de carga y descarga en el tiempo para calcular el SOC de manera dinámica. Es ampliamente utilizado y consiste simplemente en medir la corriente que entra y sale de la batería. Este método se basa en la integración pura de la corriente a lo largo del tiempo (DiVA Portal, 2014).

Para definir el Estado de Carga (SOC), si consideramos que la batería está completamente cargada en el tiempo, de acuerdo con la Ecuación 1 t_0 con $I_b(t)$ como la corriente de descarga, la carga entregada por la batería en el tiempo (t) es:

Ecuación 1

Integración de la Corriente en el Tiempo

$$\int_{t_0}^t I_b(t) d(t)$$

Con Q_0 denotando la carga total que la batería puede entregar, el SOC (%) de la batería se define como:

Ecuación 2

Estado de Carga en Función del Consumo de Corriente

$$SOC = \frac{Q_0 - \int_{t_0}^t I_b(t) d(t)}{Q_0} \times 100$$

2.2.8 Cálculo del Estado de Salud (SOH)

Para estimar el SOH, se emplean los siguientes enfoques:

Por variación o caída de voltaje: Se mide la caída de voltaje cuando la batería está sometida a una carga conocida (Spangberg, 2023). Teniendo un voltaje de referencia inicial, se compara con el voltaje de caída luego de haber aplicado una carga, el tiempo mínimo que se puede considerar es de 30 segundos con un rango de carga de entre 10 a 30 amperios.

Ecuación 3

Estado de Salud en Función de la Variación del Voltaje

$$SOH_v = \frac{V_{ref}}{\Delta V_{actual}} \times 100$$

Donde:

- **V_{referencia}:** Se predetermina un valor de 0,50V. Este valor es para tener referencia de una batería nueva que normalmente tiene esta variación de voltaje bajo una descarga controlada en un tiempo determinado.
- **ΔV_{actual}:** Es la variación del voltaje al momento de realizar la medición, por ejemplo, si antes de aplicar la descarga, hay un voltaje de 12,65V y luego de aplicar la descarga en el tiempo que se establezca, el voltaje es de 12,10V el valor de variación será de 0,55V. Realizando el cálculo con la ecuación, nos daría un SOH de 90,90%

Por variación en el estado de carga: Compara la capacidad actual y la capacidad nominal inicial, en este cálculo existe relación con el estado de carga por consumo de corriente, ya que nos permite ya tener el valor del consumo y que se relaciona con un valor inicial para de esta manera obtener la variación.

Ecuación 4

Estado de Salud en Función de la Capacidad de la Batería

$$SOH = \frac{Capacidad\ actual}{Capacidad\ nominal} \times 100$$

Estos cálculos se implementan en Arduino a través de un código que recopile datos de los sensores y realice los cálculos en tiempo real, partiendo de las ecuaciones presentadas permitiendo al usuario visualizar esta información en una pantalla LCD 2004.

Capítulo III

Implementación Física del Sensor IBS con Arduino

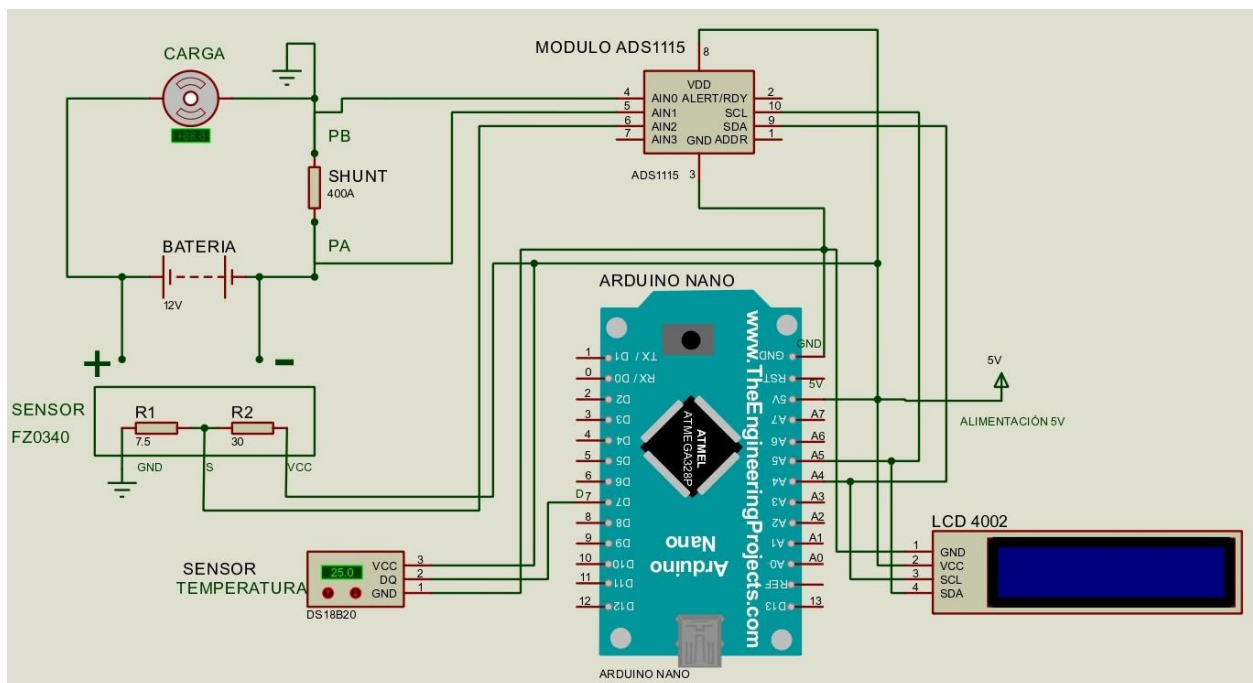
3.1 Esquema Electrónico de la Implementación Física

En la Figura 10, se detallan los componentes utilizados para el desarrollo del Sensor Inteligente de Batería (IBS)

- Arduino Nano
- Módulo ADC ADS1115
- Resistencia Shunt 400A – 75mV
- Sensor de Voltaje – FZ0340
- Sensor de Temperatura DS18B20

Figura 10

Esquema Electrónico del Sensor IBS



3.2 Conexión de Componentes

3.2.1 Resistencia Shunt a Módulo ADC ADS1115

A continuación, se detallan las conexiones empleadas en el sistema para integrar tanto el resistor shunt como el módulo ADC ADS1115 al microcontrolador Arduino Nano. Estas tablas permiten visualizar de forma clara la correspondencia entre los terminales de cada componente, facilitando así una implementación correcta y segura del circuito. Se incluyen también breves descripciones funcionales que justifican cada conexión dentro del esquema general del proyecto.

Tabla 7

Conexión de pines de Shunt a Módulo ADS1115

Componente	Terminal	Conexión a	Descripción Técnica
R.Shunt	Punto Alto (PA)	Negativo de la batería	Entrada de corriente
R.Shunt	Punto Alto (PB)	Negativo al chasis	Corriente hacia la tierra
ADS1115	AIN1 (5)	PA del shunt	Entrada diferencial positiva
ADS1115	AIN (0)	PB del shunt	Entrada diferencial negativa

3.2.2 Módulo ADC ADS1115 a Arduino Nano

Tabla 8

Conexión de Módulo ADS1115 a Arduino Nano

ADS1115	Terminal	Arduino Nano	Función
VDD (8)	3.3V o 5V	5V (Nano)	Alimentación del módulo
GND (3)	GND	GND	Referencia de tierra
SDA (9)	SDA	A4	Comunicación I2C – datos
SCL (10)	SCL	A5	Comunicación I2C – reloj

3.2.3 Sensor de Temperatura DS18B20 a Arduino Nano

Tabla 9

Conexión de Sensor Temperatura DS18B20 a Arduino Nano

Componente	Pin	Arduino Nano	Descripción
DS18B20	GND (1)	GND	Tierra común
DS18B20	VCC (3)	5V	Alimentación
DS18B20	DQ (2)	Pin D7	Comunicación digital

3.2.4 Sensor de Voltaje FZ0340 a Módulo ADC ADS1115

Tabla 10

Conexión de Sensor de Voltaje FZ0340 a Módulo ADS1115

FZ0340	Conecta a	Descripción
Señal (S)	Pin A2 ADS1115	Entrada de voltaje ADS1115
Positivo (+)	(+) Batería	Positivo batería
Negativo (-)	(-) Batería	Negativo batería
VCC	5V Arduino	Voltaje de alimentación
GND	GND Arduino	Tierra común Arduino

3.3 Programación del Sistema

3.3.1 Configuración de Ganancias del Módulo ADS1115

Las ganancias hacen referencia a cómo el ADS1115 amplifica la señal de entrada antes de convertirla en datos digitales. Como se observa en la Tabla 11, hay 6 configuraciones predefinidas, como GAIN_TWOTHIRDS, GAIN_ONE, etc.

Cada ganancia está asociada a un rango de voltaje máximo que el ADS puede leer en modo diferencial. Este es el rango de entrada analógica permitido para cada ganancia. Por ejemplo:

- **GAIN_TWOTHIRDS:** puedes medir señales de hasta ± 6.144 V, ideal si usas divisores resistivos, como en el presente proyecto que utiliza el sensor de voltaje FZ0340.
- **GAIN_SIXTEEN:** solo permite hasta ± 0.256 V, pero con una alta precisión.

Esto te permite seleccionar una relación entre rango y resolución, según lo que estés midiendo (corriente o voltaje).

Tabla 11

Configuración de Ganancias del Módulo ADS1115

Configuración de Ganancia (PGA)	Voltaje Máximo de Entrada ($\pm$V)	Resolución por bit (en mV)
GAIN_TWOTHIRDS	± 6.144 V	0.1875 mV
GAIN_ONE	± 4.096 V	0.125 mV
GAIN_TWO	± 2.048 V	0.0625 mV
GAIN_FOUR	± 1.024 V	0.03125 mV
GAIN_EIGHT	± 0.512 V	0.015625 mV
GAIN_SIXTEEN	± 0.256 V	0.0078125 mV

En el contexto del presente proyecto, una alta resolución es fundamental para captar caídas de tensión muy pequeñas generadas por el shunt de corriente, especialmente durante eventos transitorios como el arranque del motor, donde se pueden presentar picos negativos mayores a -200A (dependiendo del vehículo) durante fracciones de segundo.

Para ello, el ADS puede detectar estas ganancias, de acuerdo a la selección que se detalle en la programación.

Por ejemplo:

- GAIN_TWOTHIRDS detecta cambios de 0.1875 mV
- GAIN_SIXTEEN detecta cambios de solo 0.0078125 mV

Cuanto menor esta cifra, mayor precisión se tiene.

3.3.2 Voltaje Sensor IBS

Para la medición del voltaje de batería, se utiliza una entrada analógica del módulo ADC ADS1115, específicamente el canal A2, el cual recibe la señal de un divisor resistivo (Módulo FZ0340) conectado directamente a los bornes de la batería de 12 V. Este divisor reduce el voltaje de entrada para que se mantenga dentro del rango de medición del ADS1115, hasta 6.144 V con ganancia 2/3, como se observa en la Tabla 11.

En el código, se establece la ganancia del ADS1115 con GAIN_TWOTHIRDS, que permite una resolución de 0.1875 mV por bit. La lectura cruda (*adc_voltage*) se multiplica por este factor para obtener el voltaje en milivoltios, y posteriormente se ajusta con el factor de escalado del divisor resistivo para recuperar el voltaje real de batería.

Figura 11

Código de Programación para Sensor de Voltaje

```
ads.setGain(GAIN_TWOTHIRDS);
int16_t adc_voltage = ads.readADC_SingleEnded(2);
float battery_voltage = (adc_voltage * multiplier_voltage / 1000.0) * voltage_scale;
```

Donde:

- *adc_voltage* es la lectura cruda de 16 bits del canal A2.
- *multiplier_voltage* es el factor de resolución del ADS1115 (0.1875 mV/bit).
- *voltage_scale* es el factor de corrección del divisor resistivo (4,96 para el presente proyecto) (determinado empíricamente en pruebas).

Este valor calculado de *battery_voltage* representa el voltaje actual de la batería bajo cualquier condición (reposo, carga o descarga). Este dato se utiliza en:

- La visualización en LCD.
- El cálculo del SOC por voltaje a través de la Tabla 6.
- El cálculo del SOH por voltaje, al analizar la caída de voltaje bajo carga.

3.3.3 Corriente Sensor IBS

Para medir la corriente que fluye hacia o desde la batería, se implementa un resistor shunt de precisión conectado en serie entre la carga y el negativo de batería. La caída de voltaje a través del shunt es proporcional a la corriente según la ley de Ohm ($V = I \times R$). Esta señal en milivoltios, es leída de forma diferencial por el módulo ADC ADS1115 utilizando los canales AIN 0 (4) y AIN 1 (5) . Se repite esta lectura varias veces (ejemplo 15 o 30) para promediar y reducir el ruido en la señal medida. El resultado promedio se almacena en *promedio_adc*, esto se detalla en el código de programación de la Figura 12.

El valor ADC promedio se multiplica por el factor *multiplier_current*, que representa la resolución del ADS1115 configurado en GAIN_SIXTEEN (7.8125 $\mu\text{V/bit}$), convirtiendo el valor digital en milivoltios reales de caída de tensión en el shunt. Luego, se convierte a voltios para facilitar el cálculo de corriente, esta configuración es ideal para detectar variaciones pequeñas en el voltaje generado por el shunt.

En la Tabla 12, se detalla las variables utilizadas en el código de programación para mayor detalle de como se realiza el cálculo para la corriente que descarga o carga la batería.

Figura 12

Código de Programación para Corriente del Sensor IBS

```
ads.setGain(GAIN_SIXTEEN);
long suma_adc = 0;
for (int i = 0; i < muestras; i++) {
    suma_adc += ads.readADC_Differential_0_1();
    delay(2);
}
float promedio_adc = suma_adc / (float)muestras;
float voltage_shunt_mV = promedio_adc * multiplier_current;
float voltage_shunt_V = voltage_shunt_mV / 1000.0;
float current_A = voltage_shunt_V / shunt_resistance;
```

En esta sección se enumeran las principales variables y constantes utilizadas en el código de lectura de corriente del sensor IBS. Cada uno de estos elementos cumple una función específica en el proceso de adquisición, conversión y tratamiento de los datos eléctricos provenientes del shunt, permitiendo obtener valores precisos de corriente. La siguiente tabla resume su propósito dentro del sistema de medición.

Tabla 12

Descripción de Variables y Constantes del Código para Corriente del Sensor IBS

Elemento	Valor / Propósito
<i>GAIN_SIXTEEN</i>	Configura el ADS1115 con ± 0.256 V de rango diferencial
<i>multiplier_current</i>	Factor de resolución del ADS1115 (7.8125 μ V/bit)
<i>shunt_resistance</i>	Valor del shunt (ej. 0.00025 ohmios)
<i>voltage_shunt_mV</i>	Caída de tensión medida en milivoltios
<i>current_A</i>	Corriente real calculada ($A = V / R$)
<i>muestras</i>	Promedio de múltiples lecturas para mayor estabilidad

El valor *current_A* obtenido se usa en:

- La visualización de corriente en pantalla LCD.
- El cálculo de SOC por integración de corriente (método Coulomb Counting).
- La estimación del SOH por corriente, al observar cuánta capacidad efectiva entrega la batería durante una descarga.

3.3.4 Temperatura Sensor IBS

El sensor DS18B20 se emplea para medir la temperatura del entorno de la batería, utilizando el protocolo *OneWire*, como se muestra en la Figura 13, lo que permite una comunicación eficiente mediante un solo pin de datos. La lectura se realiza a través del objeto de control *DallasTemperature*, que permite el acceso a la medición en Celsius $^{\circ}\text{C}$. Esta variable

es esencial para monitorear condiciones térmicas y puede ser utilizada como factor de corrección en los algoritmos de cálculo de SOC y SOH.

Figura 13

Código de Programación para el Sensor de Temperatura DS18B20

```
OneWire oneWire(7);           // Pin digital 7 para datos del DS18B20
DallasTemperature sensors(&oneWire); // Objeto para controlar el sensor

sensors.begin();              // Inicializa el sensor

// Lectura de temperatura:
sensors.requestTemperatures(); // Solicita la lectura al sensor
temperatura_bateria = sensors.getTempCByIndex(0); // Obtiene la lectura en °C
```

En el presente proyecto no se lo integra de la manera antes mencionada, su aplicación es más de seguimiento o alerta visual ya que la temperatura incide directamente en el comportamiento eléctrico de la batería, sin embargo, no se descarta en futuros proyectos su integración directa para el cálculos de los estados de la batería. En la Tabla 13, se detallan los aspectos técnicos en la configuración del sensor de temperatura DS18B20.

Tabla 13

Detalles Técnicos del Sensor de Temperatura DS18B20

Elemento	Valor / Descripción
Modelo del sensor	DS18B20
Protocolo	OneWire
Precisión	± 0.5 °C entre -10 °C y +85 °C
Resolución configurable	De 9 a 12 bits (por defecto 12 bits)
Dirección única	Cada sensor posee un ID único, útil si se usan múltiples sensores en el mismo bus

3.3.5 Estado de Carga Sensor IBS

En el presente proyecto, el SOC se calcula de dos formas complementarias:

SOC por Voltaje de Reposo: Este método estima el SOC basándose en el voltaje de la batería en reposo, es decir, cuando no hay carga ni descarga significativa. Las baterías de plomo-ácido presentan una relación conocida entre el voltaje de terminal y su nivel de carga. Se utiliza la Tabla 6 de referencia, con la cual se realiza el código de programación como se muestra en la Figura 14 para estimar el SOC.

Figura 14

Código de Programación para el Cálculo del SOC según Voltaje

```
float calcularSOCporVoltaje(float v) {
    if (v >= 12.65) return 100.0;
    else if (v >= 12.55) return 90.0;
    else if (v >= 12.50) return 80.0;
    else if (v >= 12.45) return 75.0;
    else if (v >= 12.38) return 70.0;
    else if (v >= 12.31) return 60.0;
    else if (v >= 12.24) return 50.0;
    else if (v >= 12.06) return 25.0;
    else if (v >= 11.89) return 0.0;
    else return 0.0;
}
```

Consideraciones:

- Requiere que la batería esté en reposo al menos 5-10 minutos para mayor precisión.
- Es útil como referencia inicial al encender el sistema.

SOC por Consumo de Corriente (Coulomb Counting): Este método se basa en integrar la corriente consumida a lo largo del tiempo, restando o sumando carga en función del uso.

Se considera que una batería con capacidad nominal C_{nom} en Ah se descarga con una corriente $I(t)$. El SOC en el tiempo se estima usando la Ecuación 2 antes mencionada.

Figura 15

Código de Programación para SOC por Consumo de Corriente

```
float intervalo_horas = intervalo_ms / 3600000.0;
consumed_Ah += -current_A * intervalo_horas;

if (consumed_Ah < 0) consumed_Ah = 0;
if (consumed_Ah > battery_capacity_Ah) consumed_Ah = battery_capacity_Ah;

SOC = ((battery_capacity_Ah - consumed_Ah) / battery_capacity_Ah) * 100.0;
```

Consideraciones:

- Permite un seguimiento en tiempo real del consumo.
- Es más preciso durante carga/descarga activa, especialmente con corrientes altas.
- El cálculo es acumulativo, por lo tanto, pequeños errores pueden sumarse con el tiempo.

3.3.6 Estado de Salud del Sensor IBS

En el presente proyecto, se calcula el SOH mediante dos métodos:

SOH en función del consumo de corriente: Este método compara la cantidad de energía extraída de la batería durante una descarga parcial (en Ah) con el cambio real observado en el SOC. Este enfoque permite calcular la capacidad efectiva de la batería y compararla con su valor nominal, de acuerdo a la Ecuación 3 antes mencionada.

La lógica está basada en el seguimiento continuo de la corriente durante la descarga, aplicando la Ley de Coulomb e integrando en intervalos de tiempo predefinidos. La estimación se restringe al intervalo 0-100% para garantizar consistencia. Esta metodología ofrece una evaluación práctica del estado de salud sin requerir una descarga completa del sistema, por ejemplo, en las pruebas se puede considerar un consumo de 20A por un tiempo de 45

aproximadamente, para que con la programación que se configure en el Arduino, calcule el SOH.

Figura 16

Código de Programación para el SOH por Consumo de Corriente

```
if (!descarga_iniciada && SOC < 99.98) {
  SOC_inicio = SOC;
  Ah_descargados_SOHi = 0.0;
  descarga_iniciada = true;
}

if (descarga_iniciada) {
  float delta_SOC = SOC_inicio - SOC;

  if (delta_SOC > 0.15) {
    Ah_descargados_SOHi += -current_A * intervalo_horas;
    float capacidad_estimada = Ah_descargados_SOHi / (delta_SOC / 100.0);
    SOH_corriente = constrain((capacidad_estimada / battery_capacity_Ah) * 100.0, 0.0, 100.0);
  }

  if (SOC <= 10.0 || delta_SOC >= 10.0) {
    descarga_iniciada = false;
  }
}
```

Explicación de variables:

- *Ahdescargados*: Corriente integrada (negativa) durante la descarga.
- ΔSOC : Diferencia entre el SOC inicial y el SOC actual.
- $C_{nominal}$: Capacidad teórica de la batería (por ejemplo, 42 Ah).
- Se usa *constrain()* para limitar el resultado entre 0% y 100%.

Consideraciones:

- Se adapta bien a baterías con uso real bajo carga.
- Es útil cuando se dispone de medición precisa de corriente.
- Se recomienda realizar la medición en tramos de descarga moderados (2-10%).
- Los valores extremos o fluctuaciones de corriente pueden distorsionar los resultados.

SOH en función del voltaje: Durante una descarga controlada (por ejemplo, 15 segundos con carga constante), se mide la caída de voltaje. Mientras mayor sea esta caída respecto a un voltaje de referencia predeterminado para una batería en buen estado, menor es

el SOH. Cuando se detecta una condición de descarga significativa (por ejemplo, una corriente inferior a -3.00 A), se inicia un temporizador (*tiempo_descarga*) que acumula el tiempo transcurrido. Tras un lapso mínimo predefinido (por ejemplo, 15 segundos) o una reducción mínima del SOC, se evalúa el SOH.

De acuerdo a la Ecuación 4, se compara el voltaje final medido (V_{final}) con un voltaje de referencia (V_{ref_SOHi}) correspondiente a una batería en buen estado. Cuanto mayor sea la caída observada (ΔV_{actual}), menor será el SOH estimado. Este método permite una evaluación rápida del deterioro interno de la batería sin requerir descargas profundas, siendo especialmente útil para verificar la capacidad de sostener carga en condiciones reales de uso.”

Explicación de variables:

- V_{ref} : Voltaje promedio bajo carga de una batería sana (Ejemplo: 12.65 V).
- V_{final} : Voltaje medido tras la descarga.
- $\Delta V_{actual} = V_{ref} - V_{final}$
- ΔV_{ref} : Se define un umbral (Ejemplo: 0.50 V) como referencia de máxima caída aceptable.

Figura 17

Código de Programación del SOH por Caída de Voltaje

```
if (!fase_descargando && current_A < -3.00) {
    tiempo_descarga = 0.0;
    fase_descargando = true;
}

if (fase_descargando) {
    tiempo_descarga += intervalo_ms / 1000.0;

    if (tiempo_descarga >= 15.0 || SOC <= SOC_inicio - 0.1) {
        V_final = battery_voltage;
        float delta_V_actual = V_ref_SOHi - V_final;

        if (delta_V_actual > 0) {
            float nuevo_SOH = (delta_V_ref / delta_V_actual) * 100.0;
            nuevo_SOH = constrain(nuevo_SOH, 0.0, 100.0);
            if (nuevo_SOH < SOH_voltaje) {
                SOH_voltaje = nuevo_SOH;
            }
        }
    }
}
```

3.4 Encapsulado del IBS

En la Figura 19, se muestran los componentes ya con la carcasa instalada en el sistema. Para la protección del prototipo, se diseña y se imprime en 3D una carcasa, la cual está comprendida de dos partes, la primera es la base, donde internamente va el circuito con el Arduino Nano, módulo ADS1115 y sensores. Y la segunda es la carcasa superior que sirve de tapa y la pantalla LCD está fijada a esta para que todos los componentes se encuentren guardados y no queden expuestos. En la Figura 18 se muestra el diseño en 3D.

Figura 18

Carcasa del Sensor Inteligente de Batería en 3D



Figura 19

Sistema Integrado en el vehículo con la Carcasa Impresa



Capítulo IV

Análisis de Resultados

La presente sección tiene como finalidad interpretar los datos obtenidos durante las pruebas del sensor inteligente de batería (IBS) desarrollado con Arduino para baterías plomo ácido de 12 V.

Los valores registrados a través de los distintos sensores conectados al Arduino Nano permiten evaluar el comportamiento eléctrico de la batería en condiciones reales de operación, así como validar la funcionalidad del algoritmo de estimación de parámetros como el voltaje, la corriente, la temperatura, el estado de carga (SOC) y el estado de salud (SOH).

El análisis se estructura por variable medida, comenzando por el voltaje y corriente como parámetros base; con estos valores se realizan los cálculos respectivos del SOC y el SOH, cuyas variaciones están directamente relacionadas por las condiciones de descarga, carga.

4.1 Instalación del Sensor IBS en el Vehículo

Se procede con la instalación del sensor IBS en un camioneta Chevrolet Dmax 2014, como se observa en la Figura 20.

Figura 20

Vehículo Chevrolet Dmax 2014 para Prueba de Batería con Sensor IBS

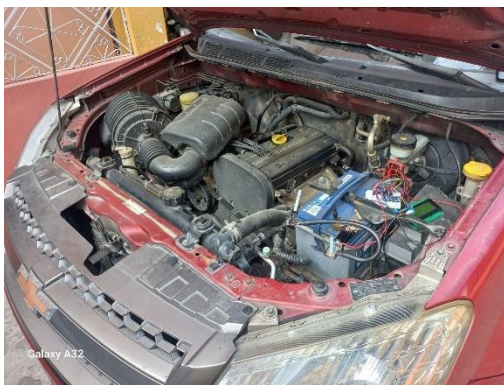


En los siguientes puntos se ejecuta el procedimiento de revisión previo al funcionamiento , donde se confirman los siguientes componentes:

- Habitáculo donde se encuentra ubicada la batería de 12V. En la Figura 21 se puede observar que ya se encuentran ubicados los componentes del sensor IBS en la batería de 12V del vehículo.

Figura 21

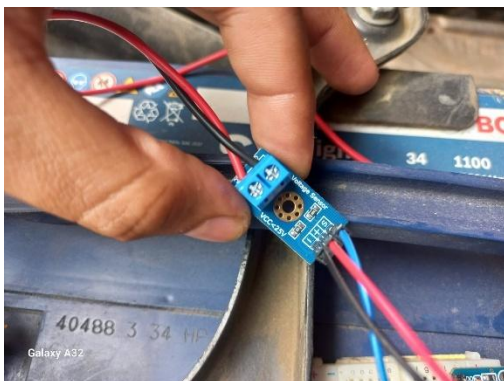
Habitáculo de Motor con el Sensor IBS Instalada en la Batería de 12V



- El sensor de voltaje FZ0340, se encuentra conectado con sus pines superior a los terminales positivo y negativo de la batería de 12V para que pueda medir el voltaje de esta al momento de que se active el sensor IBS. En la Figura 20, también se observa los pines inferiores conectados al sistema, que, de acuerdo al esquema presentado, van a las conexiones, Vcc, Gnd, y señal (S) a A2 del módulo ADS1115.

Figura 22

Sensor FZ0340 con su Cableado



En las Figuras 23, 24 y 25 se observa al Arduino Nano, módulo ADS1115 y sensor de voltaje FZ0340 ya conectados de acuerdo al esquema presentado.

Figura 23

Sensor FZ0340, Módulo ADS1115 y Arduino Nano conectados

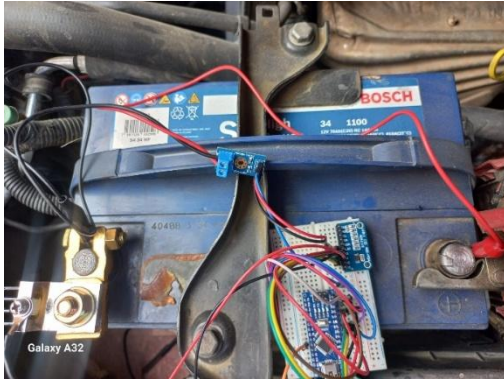


Figura 24

Módulo ADS1115 Conectado Junto al Arduino Nano

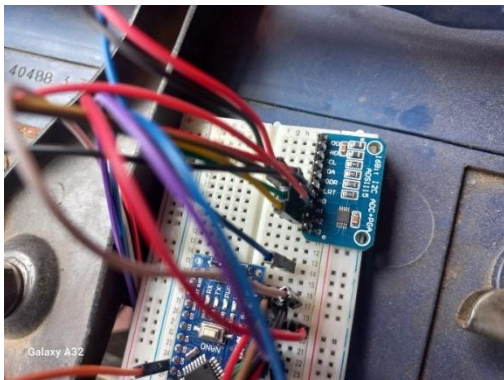
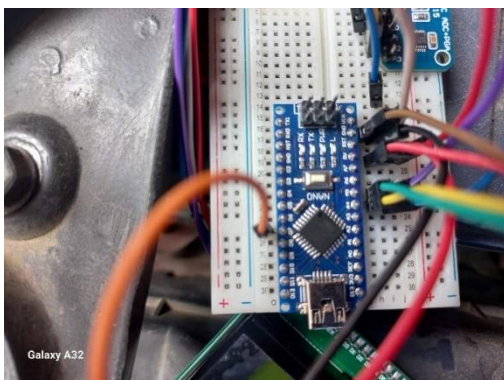


Figura 25

Arduino Nano con sus Conexiones



- En la Figura 26, la pantalla LCD 2004 también se encuentra conectada en el sistema para la visualización de datos ya mencionados en el capítulo III.

Figura 26

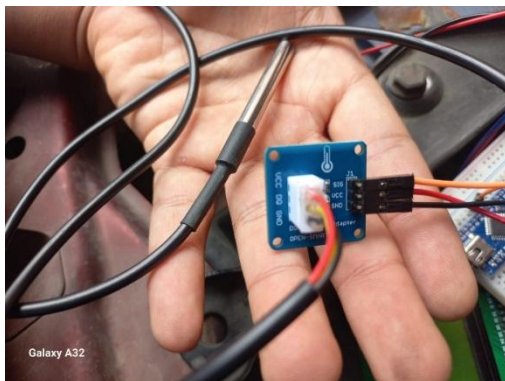
Pantalla LCD 2004 Instalada Con el Sistema IBS



- En la Figura 27, se muestra el sensor de temperatura DS18B20 con el cableado ya conectado al Arduino Nano.

Figura 27

Sensor de Temperatura DS18B20 Con sus Cableado al Arduino Nano



4.2 Muestra de Voltaje

Para las muestras se realiza la comparación con equipos de medición, para este caso un multímetro de la marca KaiWeets.

En el presente punto podemos comparar el voltaje que muestra la pantalla LCD (12,43 V) , como se observa la Figura 28 , con los 12,45V que nos evidencia el multímetro de la Figura

29. Dando como margen de error máximo de $\pm 0,02$ V, siendo bastante precisa la medida obtenida por el sensor IBS.

Figura 28

Pantalla LCD del Sensor IBS con Medición de Voltaje



Figura 29

Voltaje con Multímetro KaiWeets



4.3 Muestra de Corriente

Para la medición de corriente, realizamos la prueba bajo dos descargas diferentes, la primera, como se observa en la Figura 30, la corriente es de -13,43 A, siendo negativo ya que se está realizando un proceso de carga con las luces bajas y medias encendidas.

En la Figura 31, el multímetro muestra un valor de 13,20 A, obteniendo un margen de error de $\pm 0,23$ A.

Figura 30

Pantalla LCD del Sensor IBS con Medición de Corriente con Luces Bajas y Medias



Figura 31

Corriente de Comparación (1) con Multímetro KaiWeets

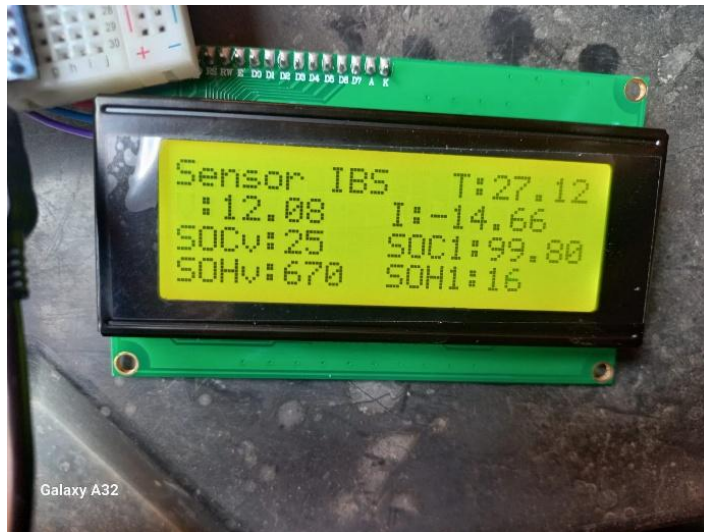


La siguiente toma de corriente fue con las luces altas encendidas, en donde da como resultado en la pantalla LCD un valor de -14,66 A, y lo podemos comparar con los 14,47 A que nos muestra el multímetro de confirmación

En este caso se evidencia un margen de error de $\pm 0,19$ A, siendo cercano a los $\pm 0,23$ A del punto anterior.

Figura 32

Pantalla LCD del Sensor IBS con Medición de Corriente con Luces Altas

**Figura 33**

Corriente de Comparación (2) con Multímetro KaiWeets



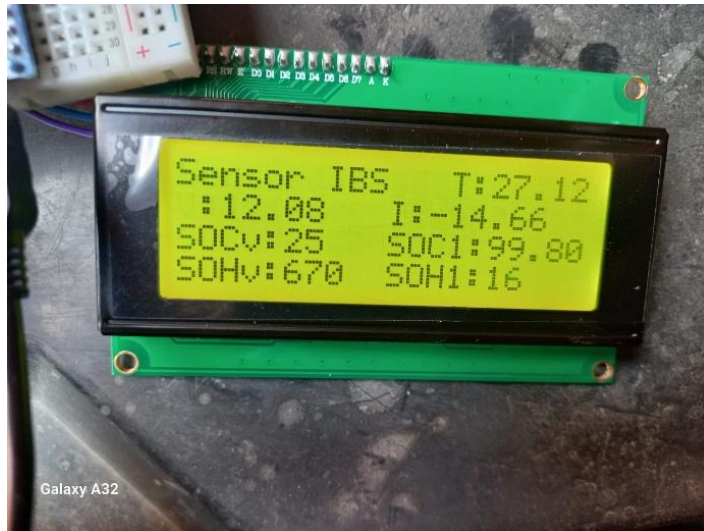
4.4 Muestra de Temperatura

Se registró la temperatura de la batería a través del sensor DS18B20. Como se observa en la Figura 34, el valor mostrado en la pantalla LCD corresponde a 27,12 °C, temperatura medida en el entorno inmediato de la caja de la batería, donde se ubica físicamente el sensor.

En las figuras anteriores también se podrá observar este valor considerando que la condición de la medición se comparte para varias de las tomas.

Figura 34

Pantalla LCD del Sensor IBS con Medición de Temperatura



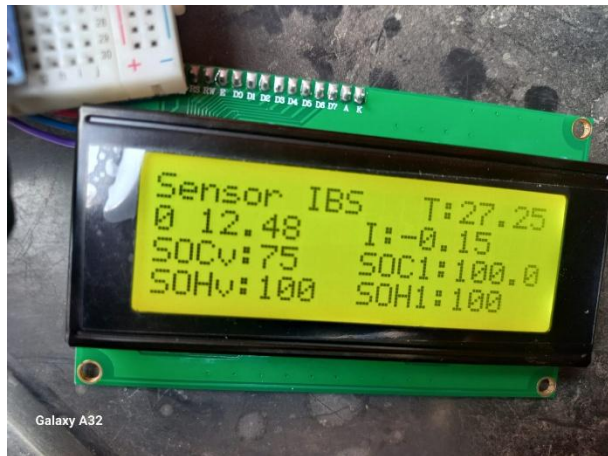
4.5 Muestra de Estado de Carga (SOC)

En la Figura 35, se muestra un SOCv de 75%, este valor representa el estado de carga estimado en función del voltaje de la batería, que en este caso es 12.48 V.

Según la tabla de referencia de voltaje para el cálculo del SOCv un voltaje de 12.48 V corresponde aproximadamente a un SOC del 75%. Esto indica que la batería no está completamente cargada y ha sufrido una leve descarga en reposo o bajo una carga baja.

El SOC1 representa el estado de carga calculado por integración de corriente (método de Coulomb Counting), el cual indica que no se ha consumido energía significativa desde el último reinicio del contador por esta razón nos muestra un valor de 100%.

Dado que la corriente mostrada es -0.15 A, y que el sistema no ha registrado una descarga prolongada o profunda, el contador de amperios-hora (Ah) no ha reducido la carga teórica.

Figura 35*Pantalla LCD del Sensor IBS con Medición de SOC*

En la Figura 36 , se muestra la prueba de batería de un comprobador de la marca TOPDON , en donde nos muestra un estado de carga (CARGO) de 85%. Con este valor y con el valor obtenido en el sensor IBS , podemos hacer una relación con los dos valores de SOC, en este caso el SOCv (75%) que es relacionado a la tabla de voltaje y el SOC1 , relacionado con el consumo de Ah que tiene la batería, si se promedian ambos valores 75% y 100%, obtenemos un SOC total de 87,5% el cuál es un valor cercano a lo que el equipo de comprobación muestra.

Figura 36*Comprobación de Estado de Batería con Equipo TOPDON*

4.6 Muestra de Estado de Salud (SOH)

En la Figura 37 se muestra un SOH_v de 73%, esto en función de la caída de voltaje bajo carga. Este método compara la diferencia entre un voltaje de referencia, toma como referencia un valor de 12.70 V como valor inicial. Un 73% indica que la batería tiene una caída de voltaje más alta de lo esperado bajo carga. Esto con una descarga de -20A por 45 segundos.

Por otro lado, el SOH₁ = 61% se estima por integración de corriente, este valor se calcula monitoreando cuántos Ah se entregan durante una descarga parcial, en base a la variación de SOC y Ah entregados. Un SOH de 61% indica que, durante la descarga monitoreada, la capacidad útil estimada es solo el 61% de la capacidad nominal, esto refleja un deterioro mucho más crítico que el mostrado por el método de voltaje.

Figura 37

Pantalla LCD con Valores de Estado de Salud SOH



De igual manera, tomando como referencia la prueba que se realizó con el equipo TOPDON, se observa que el estado de salud (SALUD) es de 46% y en los resultados de SOH se muestran dos valores, 73% y 61%, como en el SOC, estos también se pueden promediar para obtener un SOH total de 67%, tal como se evidencia en la Figura 38, que muestra un valor cercano, de 61% .

Figura 38

Impresión de Estado de Batería con Equipo TOPDON



Conclusiones

Se logró construir y validar un sensor IBS funcional para el monitoreo de baterías de ácido-plomo de 12 V, empleando componentes de bajo costo como el microcontrolador Arduino Nano, el ADC ADS1115 y sensores específicos para voltaje, corriente y temperatura. La arquitectura desarrollada permitió realizar mediciones en tiempo real, con una visualización clara de los parámetros mediante una pantalla LCD.

El sensor IBS demostró una precisión adecuada en la lectura de voltaje, corriente, temperatura, estado de carga (SOC) y estado de salud (SOH), resultados que fueron validados a través de la comparación con equipos de diagnóstico automotriz, evidenciando su confiabilidad en condiciones reales de operación.

La incorporación de dos métodos de estimación para el SOC —por voltaje y por conteo de Coulomb (integración de corriente)— permitió comparar el comportamiento de la batería bajo distintas condiciones de carga y descarga. Se verificó que el método de Coulomb Counting brinda mayor precisión en pruebas prolongadas, mientras que la estimación basada en voltaje es útil para obtener referencias rápidas en intervalos cortos.

Asimismo, se implementaron algoritmos para la estimación del SOH de la batería: uno basado en la relación entre la energía consumida (Ah descargados y variación del SOC) y otro mediante la medición de la caída de tensión bajo carga. Las pruebas controladas evidenciaron que ambos métodos ofrecen resultados consistentes y permiten identificar tendencias claras en la degradación y el estado de la batería, fortaleciendo la utilidad del sistema para el diagnóstico preventivo.

En síntesis, el sistema IBS desarrollado demuestra la viabilidad de emplear hardware de código abierto para la instrumentación automotriz, facilitando prácticas experimentales, el aprendizaje técnico y la optimización del mantenimiento en vehículos convencionales.

Recomendaciones

Se recomienda la inclusión de una memoria SD o un módulo de almacenamiento externo para registrar el historial de mediciones. Esta funcionalidad permitiría la trazabilidad de los datos, facilitando el análisis de tendencias, el monitoreo del desgaste a largo plazo y la detección anticipada de fallos en la batería.

Como desarrollo futuro, se propone la integración del sensor IBS con el sistema de control del alternador del vehículo. Esta vinculación permitiría que el sensor envíe señales al alternador para optimizar los ciclos de carga en función del estado real de la batería, contribuyendo a una gestión energética más eficiente y prolongando la vida útil tanto de la batería como del propio sistema de carga.

Para versiones futuras, se sugiere rediseñar el sistema con una arquitectura basada en protocolos de comunicación vehicular como LIN o CAN, con el objetivo de permitir una integración directa con las ECU del vehículo y alcanzar un mayor grado de automatización en la gestión de la energía de los diferentes sistemas eléctricos y electrónicos del automóvil.

Bibliografía

- Arduino Docs.* (2025). Obtenido de What Is I2C?:
<https://docs.arduino.cc/learn/communication/wire/>
- Augeri, F. (2012). *CISE Electronica*. Obtenido de Estado de carga (State of Charge):
<https://www.cise.com/portal/notas-tecnicas/item/583-estado-de-carga-soc-state-of-charge.html>
- autodemiaonline.com.* (2025). Obtenido de El sensor IBS:
<https://autodemiaonline.com/leccion/3-sensor-ibs/#:~:text=Su%20funci%C3%B3n%20es%20monitorizar%20el,vida%20%C3%BAtil%20de%20la%20bater%C3%ADa.>
- Barreiro, S. (2020). *Repositorio UIDE*. Obtenido de Metodología de prueba para análisis de fallas en baterías automotrices con tecnología plomo-ácido grupo 42.:
<https://repositorio.uide.edu.ec/handle/37000/4033>
- Battery University.* (2022). Obtenido de BU-410: Charging at High and Low Temperatures:
<https://batteryuniversity.com/article/bu-410-charging-at-high-and-low-temperatures>
- Bimmerfest. (2015). *Bimmerfest*. Obtenido de Real time Intelligent Battery Sensor current indication?: <https://www.bimmerfest.com/threads/real-time-intelligent-battery-sensor-current-indication.1459900/>
- BOSCH. (2025). *Bosch Movility*. Obtenido de Electronic battery sensor: <https://www.bosch-mobility.com/en/solutions/sensors/electronic-battery-sensor/>
- De la Rosa, C., & Marino, M. (2024). *Repositorio UIDE*. Obtenido de Rediseño del Banco de entrenamiento del Sistema de Carga y Arranque para su Implementación de los

Sistemas Start & Stop y Carga controlada por la Unidad Electrónica Fase I:
<https://repositorio.uide.edu.ec/handle/37000/7299>

DiVA Portal. (2014). *DiVA Portal*. Obtenido de Implementation of an Algorithm For Estimating Lead-Acid Battery State of Charge: <https://www.diva-portal.org/smash/get/diva2:1720726/FULLTEXT01.pdf>

FERRETRONICA. (2025). *FERRETRONICA*. Obtenido de Modulo Sensor de Voltaje FZ0430 0V ~ 25V: <https://ferretronica.com/products/modulo-sensor-de-voltaje-fz0430-0v-25v?srsltid=AfmBOor1lJk3yre00UFWmQXIzCcWVndJ96QAc3Bxm7U1x4Al7wwyZhor>

GeekFactory. (2025). *GeekFactory*. Obtenido de DS18B20 con Arduino: Sensor de temperatura digital: <https://www.geekfactory.mx/tutoriales-arduino/ds18b20-con-arduino-sensor-de-temperatura-digital/?srsltid=AfmBOoqCxCfowmJZfYTMd4laq2vrpupx1vZfC-a9liJn4nOKIoCHaGt2>

HELLA. (2012). *Hella*. Obtenido de Intelligent battery sensor (IBS) 12 V: [https://www.hella.com/hella-sg/assets/media/J01213_Intelligent%20battery%20sensor%20\(IBS\)%2012%20V.pdf](https://www.hella.com/hella-sg/assets/media/J01213_Intelligent%20battery%20sensor%20(IBS)%2012%20V.pdf)

Komsiyska, L., Buchberger, T., Diehl, S., & Ehrensberger, M. (2021). *Energies*. Obtenido de Critical Review of Intelligent Battery Systems: Challenges, Implementation, and Potential for Electric Vehicles: <https://www.mdpi.com/1996-1073/14/18/5989>

Lohmeier, C., & Veik, T. (2021). *Battery Power*. Obtenido de Battery Management With an Intelligent Battery Sensor is Vital to the Success of Future Automotive Designs: <https://www.batterypoweronline.com/articles/battery-management-with-an-intelligent-battery-sensor-is-vital-to-the-success-of-future-automotive-designs/>

NaturesGenerator. (2023). *NaturesGenerator*. Obtenido de Lead Acid Battery Voltage Chart:

https://naturesgenerator.com/blogs/news/lead-acid-battery-voltage-chart?srltid=AfmBOorquN9HOpT3R6D_YSwUAOJwGAY6BRrk_ow3p75dyJkRXu3jvy4v

NayLampMechatronic. (2023). Obtenido de Módulo ADC ADS1115:

<https://naylampmechatronics.com/conversores/394-modulo-adc-ads1115.html>

NayLampMechatronic. (2025). *NayLampMechatronic*. Obtenido de Tutorial sensor digital de

temperatura DS18B20: https://naylampmechatronics.com/blog/46_tutorial-sensor-digital-de-temperatura-ds18b20.html

NayLampMechatronics. (2023). Obtenido de Módulo ADC ADS1115:

<https://naylampmechatronics.com/conversores/394-modulo-adc-ads1115.html>

NayLapMechatronic. (2023). *NayLapMechatronic*. Obtenido de Tutorial LCD, conectando tu

arduino a un LCD1602 y LCD2004: https://naylampmechatronics.com/blog/34_tutorial-lcd-conectando-tu-arduino-a-un-lcd1602-y-lcd2004.html

NextPCB. (2025). Obtenido de The Ultimate Guide to Arduino Nano Pinout:

<https://www.nextpcb.com/blog/arduino-nano-pinout>

ParaArduino. (2023). *ParaArduino*. Obtenido de Temperatura y humedad en LCD 1602:

<https://paraarduino.com/displays/temperatura-y-humedad-en-lcd-1602/>

Pedamkar, P. (2023). *EDUCBA*. Obtenido de What is Arduino?:

<https://www.educba.com/what-is-arduino/>

Pickering, P. (2025). *ISABELLENHUETTE*. Obtenido de Measuring the differences: Shunt-based sensing technology vs. magnetic: https://www.isabellenhuettesusa.com/wp-content/uploads/2020/12/Isabellenhutte_EV-Shunt-Resistors_tech-article_final.pdf

Plaza, D. (2022). *ForoCochesElectricos*. Obtenido de Estado de carga (SOC) y estado de salud (SOH).

Spangberg, C. (2023). *DTE Research*. Obtenido de State of Health estimation of battery Systems:
https://orbit.dtu.dk/files/326540074/State_of_Health_estimation_of_LFP_battery_systems.pdf

UAMedia. (2021). *UAMedia Blog*. Obtenido de Arduino y sus Aplicaciones:
<https://uamedia.org/blog/arduino-y-sus-aplicaciones/#:~:text=El%20objetivo%20principal%20de%20Arduino,funcionando%20con%20la%20misma%20base.>

Veratron. (2024). *Productimageserver*. Obtenido de LINK UP INTELLIGENT BATTERY SENSOR 12V: <https://productimageserver.com/literature/quickGuide/101795QG.pdf>

Anexos

Anexo 1

Código de Programación del Sensor Inteligente de Batería con Arduino

```
#include <Wire.h>
#include <Adafruit_ADS1X15.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <LiquidCrystal_I2C.h>

Adafruit_ADS1115 ads;
LiquidCrystal_I2C lcd(0x27, 20, 4);

#define TEMPERATURE_PIN 7
OneWire oneWire(TEMPERATURE_PIN);
DallasTemperature sensors(&oneWire);

// Constantes
const float multiplier_current = 0.0078125F;
const float shunt_resistance = 0.000179;
const int muestras = 10;

const float multiplier_voltage = 0.1875F;
const float voltage_scale = 4.96;

const float battery_capacity_Ah = 70.0;
const float V_ref_SOHi = 12.70;
const float delta_V_ref = 0.40;

// Variables principales
float consumed_Ah = 0.0;
float SOC = 100.0;
float SOC_por_voltaje = 100.0;

float SOC_inicio = 100.0;
float Ah_descargados_SOHi = 0.0;
float SOH_corriente = 100.0;
bool descarga_iniciada = false;

float V_final = 0.0;
float tiempo_descarga = 0.0;
bool fase_descargando = false;
float SOH_voltaje = 100.0;

float temperatura_bateria = 0.0;

unsigned long previousMillis = 0;
```

```

const unsigned long intervalo_ms = 125;

void setup() {
  Serial.begin(9600);
  ads.begin();
  sensors.begin();
  lcd.init();
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Sensor IBS      T: ---");

  // Encabezado para Excel
  Serial.println("Voltaje (V),Corriente (A),Temperatura (C),SOC Voltaje
(%),SOC Integrado (%),SOH Voltaje (%),SOH Corriente (%),Ah Consumidos");
}

void loop() {
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= intervalo_ms) {
    previousMillis = currentMillis;

    // --- Corriente promedio ---
    ads.setGain(GAIN_SIXTEEN);
    long suma_adc = 0;
    for (int i = 0; i < muestras; i++) {
      suma_adc += ads.readADC_Differential_0_1();
      delay(2);
    }
    float promedio_adc = suma_adc / (float)muestras;
    float voltage_shunt_mV = promedio_adc * multiplier_current;
    float voltage_shunt_V = voltage_shunt_mV / 1000.0;
    float current_A = voltage_shunt_V / shunt_resistance;

    // --- Voltaje batería ---
    ads.setGain(GAIN_TWOTHIRDS);
    int16_t adc_voltage = ads.readADC_SingleEnded(2);
    float battery_voltage = (adc_voltage * multiplier_voltage / 1000.0) *
voltage_scale;

    // --- SOC por integración ---
    float intervalo_horas = intervalo_ms / 3600000.0;
    consumed_Ah += -current_A * intervalo_horas;
    consumed_Ah = constrain(consumed_Ah, 0, battery_capacity_Ah);
    SOC = ((battery_capacity_Ah - consumed_Ah) / battery_capacity_Ah) * 100.0;

    if (SOC >= 100.0) {
      SOC = 100.0;
    }
  }
}

```

```

    SOC_inicio = 100.0;
    Ah_descargados_SOHi = 0.0;
    SOH_corriente = 100.0;
    descarga_iniciada = false;
}

// --- SOC por voltaje ---
SOC_por_voltaje = calcularSOCporVoltaje(battery_voltage);

// --- SOH por corriente ---
if (!descarga_iniciada && SOC < 99.99) {
    SOC_inicio = SOC;
    Ah_descargados_SOHi = 0.0;
    descarga_iniciada = true;
}

if (descarga_iniciada) {
    float delta_SOC = SOC_inicio - SOC;

    if (delta_SOC > 0.12) {
        Ah_descargados_SOHi += -current_A * intervalo_horas;
        float capacidad_estimada = Ah_descargados_SOHi / (delta_SOC / 100.0);

        float nuevo_SOH = constrain((capacidad_estimada) * 100.0, 0.0, 100.0);
        if (nuevo_SOH < SOH_corriente) {
            SOH_corriente = nuevo_SOH;
        }
    }
}

if (SOC <= 10.0 || delta_SOC >= 10.0) {
    descarga_iniciada = false;
}
}

// --- SOH por voltaje (caída) ---
if (!fase_descargando && current_A < -3.00) {
    tiempo_descarga = 0.0;
    fase_descargando = true;
}

if (fase_descargando) {
    tiempo_descarga += intervalo_ms / 1000.0;
    if (tiempo_descarga >= 20.0 || SOC <= SOC_inicio - 0.15) {
        V_final = battery_voltage;
        float delta_V_actual = V_ref_SOHi - V_final;

        if (delta_V_actual > 0.05) { // Evita divisiones peligrosas
            float nuevo_SOH = (delta_V_ref / delta_V_actual) * 100.0;

```

```

        nuevo_SOH = constrain(nuevo_SOH, 0.0, 100.0);
        if (nuevo_SOH < SOH_voltaje) {
            SOH_voltaje = nuevo_SOH;
        }
    }
    fase_descargando = false;
}

// --- Temperatura ---
sensors.requestTemperatures();
temperatura_bateria = sensors.getTempCByIndex(0);

// --- LCD Actualizaciones ---
// --- Limpieza de líneas antes de imprimir nuevos datos ---
lcd.setCursor(0, 0); lcd.print(" "); // Borra línea 0
lcd.setCursor(0, 1); lcd.print(" "); // Borra línea 1
lcd.setCursor(0, 2); lcd.print(" "); // Borra línea 2
lcd.setCursor(0, 3); lcd.print(" "); // Borra línea 3

// --- Ahora sí, imprime el contenido limpio ---
lcd.setCursor(0, 0);
lcd.print("Sensor IBS ");
lcd.setCursor(13, 0);
lcd.print("T:");
lcd.print(temperatura_bateria, 1);
lcd.print(" ");

lcd.setCursor(0, 1);
lcd.print("V:");
lcd.print(battery_voltaje, 2);
lcd.setCursor(10, 1);
lcd.print("I:");
lcd.print(current_A, 2);
lcd.print(" ");

lcd.setCursor(0, 2);
lcd.print("SOCv:");
lcd.print(SOC_por_voltaje, 0);
lcd.setCursor(10, 2);
lcd.print("SOC1:");
lcd.print(SOC, 2);
lcd.print(" ");

lcd.setCursor(0, 3);
lcd.print("SOHv:");
lcd.print(min(SOH_voltaje, 100.0), 0);
lcd.setCursor(10, 3);

```

```

    lcd.print("SOH1:");
    lcd.print(SOH_corriente, 0);
    lcd.print(" ");

    // --- Serial para Excel ---
    Serial.print(battery_voltage, 2); Serial.print(",");
    Serial.print(current_A, 2); Serial.print(",");
    Serial.print(temperatura_bateria, 2); Serial.print(",");
    Serial.print(SOC_por_voltaje, 0); Serial.print(",");
    Serial.print(SOC, 2); Serial.print(",");
    Serial.print(SOH_voltaje, 0); Serial.print(",");
    Serial.print(SOH_corriente, 0); Serial.print(",");
    Serial.println(consumed_Ah, 4);
}
}

float calcularSOCporVoltaje(float v) {
    if (v >= 12.65) return 100.0;
    else if (v >= 12.55) return 90.0;
    else if (v >= 12.50) return 80.0;
    else if (v >= 12.45) return 75.0;
    else if (v >= 12.38) return 70.0;
    else if (v >= 12.31) return 60.0;
    else if (v >= 12.24) return 50.0;
    else if (v >= 12.06) return 25.0;
    else if (v >= 11.89) return 0.0;
    else return 0.0;
}

```