



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO
CARRERA DE INGENIERÍA AUTOMOTRIZ

DISEÑO DE UN PANEL DE CONTROL PARA LA VISUALIZACIÓN DE LOS
DATOS DE OPERACIÓN DEL VEHÍCULO JAC T8 MEDIANTE LA INTEGRACIÓN
DEL PROTOCOLO DE COMUNICACIÓN CAN BUS, EL SOFTWARE LABVIEW Y
LA TARJETA NI 9862

Trabajo de titulación previo a la obtención del
Título de Ingeniero Automotriz

AUTORES: JIMMY DARÍO CÓNDOR TASINCHANO

LEONARDO CHURO LUGMAÑA

TUTOR: CARLOS ALBERTO CARRANCO QUIÑONEZ

Quito - Ecuador

2024

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Jimmy Darío Córdor Tasinchano con documento de identificación N° 1720669397 y Leonardo Churo Lugmaña con documento de identificación N° 1725326837 manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 5 de agosto del año 2024

Atentamente,



Jimmy Darío Córdor Tasinchano
1720669397



Leonardo Churo Lugmaña
1725326837

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Jimmy Darío Córdor Tasinchano con documento de identificación No. 1720669397 y Leonardo Churo Lugmaña con documento de identificación No. 1725326837, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: “Diseño de un panel de control para la visualización de los datos de operación del vehículo JAC T8 mediante la integración del protocolo de comunicación CAN BUS, el software LabVIEW y la tarjeta NI 9862”, el cual ha sido desarrollado para optar por el título de Ingenieros Automotrices, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana

Quito, 5 de agosto del año 2024

Atentamente,



Jimmy Darío Córdor Tasinchano
1720669397



Leonardo Churo Lugmaña
1725326837

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Carlos Alberto Carranco Quiñónez con documento de identificación N° 1713629564, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DISEÑO DE UN PANEL DE CONTROL PARA LA VISUALIZACIÓN DE LOS DATOS DE OPERACIÓN DEL VEHÍCULO JAC T8 MEDIANTE LA INTEGRACIÓN DEL PROTOCOLO DE COMUNICACIÓN CAN BUS, EL SOFTWARE LABVIEW Y LA TARJETA NI 9862, realizado por Jimmy Darío Córdor Tasinchano con documento de identificación N° 1720669397 y por Leonardo Churo Lugmaña con documento de identificación N° 1725326837, obteniendo como resultado final el trabajo de titulación bajo la opción: Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 5 de agosto del año 2024

Atentamente,



Ing. Carlos Alberto Carranco Quiñónez, MsC.

1713629564

DEDICATORIA

Dedico el presente proyecto a mis padres, por su amor incondicional, apoyo constante y por enseñarme el valor del esfuerzo y la dedicación. A mis hermanos, por ser mi inspiración y por su compañía en este viaje académico.

A mis amigos, quienes con su comprensión y motivación han sido un pilar fundamental en momentos de dificultad. A mis profesores y mentores, por su valiosa orientación, paciencia y por compartir su conocimiento, que ha sido crucial para mi crecimiento personal y profesional.

Finalmente, dedico este trabajo a todas las personas que han contribuido a mi formación y desarrollo de esta tesis. Gracias por creer en mí y por ser parte de este logro.

Jimmy Darío Cóndor Tasinchano

Dedico el presente proyecto a mi madre y padre porque me han inspirado y me han dado ejemplo con sus propias vidas, por lo que quiero hacer una mención especial a mi padre, que siempre me ha dicho “quiero que seas mejor que yo”, siempre me ha dicho “se observador”, siempre me ha dicho “escoge algo técnico”, siempre lo repetía una y otra vez, no solo a mí, si no, también a mis hermanos, a ninguno le gustaba y llegaba a ser tedioso. Pero a medida que pasaba el tiempo y crecimos, todos acabamos diciendo “tenías razón papá”, y eso quiero decirte hoy papá, tenías razón, a lo largo de toda la carrera me sirvieron todos los conocimientos que aprendí en el colegio por seleccionar “algo técnico”, cuando entre a trabajar a un taller nunca había sido tan necesario tu frase de “ser observador”, tu frase de “ser mejor que tú” ha sido una constante inspiración y me ha ayudado a mejorar como persona, no te había dicho pero una de las motivaciones para seleccionar la carrera de automotriz fue por ti, porque sé que en el fondo siempre quisiste que alguien siga tus pasos en lo que te apasiona, por todo esto quiero dedicarte mi tesis.

Leonardo Churo Lugmaña

AGRADECIMIENTO

Mis sinceros agradecimientos a Luis Cóndor y Blanca Tasinchano, por su inquebrantable apoyo, paciencia y amor. Gracias por ser una fuente constante de inspiración y por creer en mí en cada paso de este camino. Este logro es tanto suyo como mío.

Con gratitud y cariño,

Jimmy Darío Cóndor Tasinchano

Agradezco a mi madre la licenciada Elsa Lugmaña y a mi padre el licenciado Fabián Churo porque sin ellos no hubiera sido posible lograrlo, he llegado a este punto de mi vida por todas las oportunidades que me ha brindado, me siento en deuda por todos los sacrificios que han hecho por mí, quiero que sepan que algún día voy a devolverles todo lo que me han dado, mi deseo es hacerles sentir tan orgullosos de mi como yo me siento afortunado de tenerlos como padres.

Leonardo Churo Lugmaña

ÍNDICE GENERAL

RESUMEN.....	1
ABSTRACT.....	2
INTRODUCCIÓN	3
PROBLEMA.....	5
ANTECEDENTES.....	6
Objetivo General.....	8
Objetivos Específicos.....	8
CAPÍTULO 1.....	9
1. MARCO TEÓRICO.....	9
1.1. Red CANbus.....	9
1.2. Topología de la red.....	9
1.3. Componentes de la red.....	10
1.3.1. Controlador CAN.....	10
1.3.2. Transceptor (transmisor – receptor).....	10
1.3.3. Cables del bus de datos.....	11
1.3.4. Terminadores o elementos de cierre.....	12
1.4. Protocolo de transmisión.....	12
1.5. Funciones de los campos del mensaje CAN.....	12
1.6. OBD.....	14
1.6.1. OBD II.....	14
1.7. PIDs.....	16
1.7.1. PIDs Estándar.....	16
1.8. LabVIEW.....	17
1.9. NI 9862.....	18
1.10. Librería Automotive Diagnostic Command Set Toolkit.....	19
CAPÍTULO 2.....	20
2. METODOLOGÍA.....	20
2.1. Esquema de metodología.....	21
2.2. Selección de Herramientas y Materiales.....	22
2.2.1. Software.....	22
2.2.2. Hardware.....	24
2.3. Estrategias de Recolección de Datos.....	24

2.3.1.	Iniciador de sesión de diagnóstico CAN	24
2.3.2.	Identificación de variables.....	26
2.3.3.	Diseño del Sistema de PID	28
CAPÍTULO 3.....		31
3.	DESARROLLO DE PANEL DE CONTROL	31
3.1.	Conexión de la Tarjeta NI-9862	31
3.2.	Configuración de parámetros para apertura de sesión de diagnóstico en la interfaz CAN	34
3.3.	Extracción de información del vehículo	35
3.4.	Diseño de diagrama de bloques	36
3.4.1.	Diseño diagrama de bloques sensor ECT	38
3.4.2.	Diseño diagrama de bloques tacómetro.....	38
3.4.3.	Diseño diagrama de bloques MAF	39
3.4.4.	Diseño diagrama de bloques MAP	40
3.4.5.	Diseño diagrama de bloques RPM	41
3.4.6.	Diseño diagrama general	42
3.5.	Diseño interfaz de usuario	43
3.6.	Validación de datos.....	47
CAPÍTULO 4.....		53
4.	ANÁLISIS E INTERPRETACIÓN DE RESULTADOS	53
4.1.	Efectividad del Protocolo CAN 2.0 y LabVIEW	53
4.2.	Importancia de la Validación de Datos.....	53
4.3.	Mejora en la Diagnóstico y Monitoreo	53
4.4.	Capacidad de Adaptación y Futuras Mejoras	53
5.	CONCLUSIONES.....	54
6.	RECOMENDACIONES	56
7.	REFERENCIAS BIBLIOGRÁFICAS.	57
8.	ANEXOS.....	59

ÍNDICE DE FIGURAS

Figura 1	<i>Estructura multiplexada en paralelo</i>	9
Figura 2	<i>Ubicación del controlador CAN</i>	10
Figura 3	<i>Representación de línea CAN High y CAN Low</i>	11
Figura 4	<i>Terminadores repartidas entre las diferentes UCE</i>	12
Figura 5	<i>Campos de mensaje de datos CAN</i>	14
Figura 6	<i>Pines de conexión Diagnostic Link Conector (DLC) ISO J1962 tipo A</i>	15
Figura 7	<i>Tarjeta NI-9862 y accesorios</i>	18
Figura 8	<i>NI-9862 Descripción general del hardware</i>	19
Figura 9	<i>Esquema de metodología</i>	21
Figura 10	<i>Visualización de entorno de LabVIEW</i>	22
Figura 11	<i>Librería de Automotive Diagnostic Command Set de LabVIEW</i>	23
Figura 12	<i>Función “Open Diagnostic on CAN FD”</i>	25
Figura 13	<i>Función “OBD Request Current Powertrain Diagnostic”</i>	27
Figura 14	<i>Interfaz de obtención de datos</i>	28
Figura 15	<i>Diseño en el block diagrama para obtención de datos</i>	30
Figura 16	<i>Conexión puerto OBD con tarjeta NI-9862</i>	31
Figura 17	<i>Ventana de administrador de dispositivos de Windows</i>	32
Figura 18	<i>Ventana NI-XNET Bus Monitor identificación de puerto</i>	33
Figura 19	<i>Selección de puerto CAN en la interfaz de LabVIEW</i>	34
Figura 20	<i>Interfaz de configuración de “Open Diagnostico CAN FD”</i>	35
Figura 21	<i>Función “OBD Request Vehicle Information”</i>	36
Figura 22	<i>Funciones para tratamiento de datos</i>	37
Figura 23	<i>Diagrama de bloques sensor ECT</i>	38
Figura 24	<i>Diagrama de bloques tacómetro</i>	39
Figura 25	<i>Diagrama de bloques MAF</i>	40
Figura 26	<i>Diagrama de bloques MAP</i>	41
Figura 27	<i>Diagrama de bloques velocidad real de motor</i>	42
Figura 28	<i>Diagrama de bloques completo para visualizar datos de sensores referentes al motor del vehículo</i>	43
Figura 29	<i>Modelo de interfaz para visualización de datos PID</i>	44
Figura 30	<i>Diseño de la interfaz de usuario en LabVIEW</i>	45

Figura 31 <i>Panel de control "Scanner"</i>	46
Figura 32 <i>Panel de control "Selector PID"</i>	46

ÍNDICE DE TABLAS

Tabla 1 <i>Valor de PID's con descripción</i>	17
Tabla 2 <i>Identificadores CAN de 11 bits para OBD</i>	29
Tabla 3 <i>Información de PID para ECT</i>	38
Tabla 4 <i>Información de PID para tacómetro</i>	39
Tabla 5 <i>Información de PID para MAF</i>	40
Tabla 6 <i>Información de PID para MAP</i>	41
Tabla 7 <i>Información de PID para RPM</i>	42
Tabla 8 <i>Proceso de validación de datos</i>	48
Tabla 9 <i>Proceso de validación de graficas</i>	50

ÍNDICE DE ANEXOS

Anexo 1 <i>Selección de puerto CAN dentro del panel de control</i>	59
Anexo 2 <i>Archivo Excel generado en la extracción de datos</i>	59

RESUMEN

El proyecto de tesis se centra en desarrollar una interfaz de diagnóstico y mantenimiento para la camioneta JAC T8, mediante la interpretación de datos del bus de comunicación CANbus. Se emplean herramientas como LabVIEW y la tarjeta NI 9862 para la adquisición y análisis de datos de sensores y actuadores del vehículo. Estrategias detalladas de recolección de datos, como la función "Open Diagnostic on CAN FD" y la identificación de variables a través de "Request Current Powertrain Diagnostic", son implementadas.

El proyecto aborda aspectos técnicos como el sistema CAN en vehículos, terminadores, protocolo de transmisión, OBD II, PIDs, LabVIEW, la tarjeta NI-9862, y la librería Automotive Diagnostic Command Set Toolkit. Se resalta la importancia de integrar tecnologías avanzadas para visualizar y analizar datos en tiempo real en el desarrollo de un panel de control vehicular.

Se ha diseñado un sistema de monitoreo de datos de sensores de la camioneta JAC T8 utilizando LabVIEW, validando la precisión de los datos obtenidos mediante un escáner de diagnóstico y gráficas. Los resultados confirman la exactitud y representatividad de los datos recopilados en diversas condiciones de funcionamiento del vehículo.

El uso de LabVIEW junto con la tarjeta NI 9862 y el protocolo CAN 2.0 ha demostrado ser altamente efectivo para la recolección y análisis en tiempo real de datos del vehículo JAC T8. La extracción y decodificación eficiente de Identificadores de Parámetro (PID) ha mejorado significativamente el diagnóstico y monitoreo del vehículo. La infraestructura desarrollada sienta una estructura sólida para posteriores expansiones y progresos en el ámbito automotriz, aunque se han identificado desafíos relacionados con la comunicación con otros módulos del vehículo que requieren un estudio más profundo para mejorar su integración y funcionalidad.

Palabras Claves: PID, CANbus, LabVIEW, NI 9862, OBD II, ADCS.

ABSTRACT

The thesis project focuses on developing a diagnostic and maintenance interface for the JAC T8 truck through the interpretation of data from the CANbus communication bus. Tools such as LabVIEW and the NI 9862 card are employed for data acquisition and analysis from the vehicle's sensors and actuators. Detailed data collection strategies, such as the "Open Diagnostic on CAN FD" function and the identification of variables through "Request Current Powertrain Diagnostic," are implemented.

The project addresses technical aspects such as the CAN system in vehicles, terminators, transmission protocol, OBD II, PIDs, LabVIEW, the NI-9862 card, and the Automotive Diagnostic Command Set Toolkit library. The importance of integrating advanced technologies to visualize and analyze real-time data in the development of a vehicle control panel is highlighted.

A data monitoring system for the sensors of the JAC T8 truck has been designed using LabVIEW, validating the accuracy of the data obtained through a diagnostic scanner and graphs. The results confirm the accuracy and representativeness of the data collected under various operating conditions of the vehicle.

The use of LabVIEW along with the NI 9862 card and the CAN 2.0 protocol has proven to be highly effective for the real-time collection and analysis of data from the JAC T8 vehicle. The efficient extraction and decoding of Parameter Identifiers (PID) have significantly improved vehicle diagnostics and monitoring. The developed infrastructure lays a solid foundation for future expansions and progress in the automotive field, although challenges related to communication with other vehicle modules have been identified, requiring further study to improve integration and functionality.

Keywords: PID, CANbus, LabVIEW, NI 9862, OBD II, ADCS

INTRODUCCIÓN

La tecnología de diagnóstico y monitoreo de vehículos ha avanzado en la industria automotriz. avanzado significativamente, especialmente con el uso de sistemas de comunicación como el CANbus (Controller Area Network). Este sistema es una red de red en serie confiable y eficiente diseñada para permitir la interacción entre diferentes unidades de control electrónico (UCE) dentro de un vehículo. Desarrollado en la década de 1980, Bosch, el CANbus se ha transformado en un estándar global para la transmisión de datos en vehículos debido a su alta fiabilidad y la tasa de transmisión de datos.

La ingeniería inversa, en este contexto, juega un papel importante al permitir la descomposición y análisis de los sistemas tanto de control y comunicación de los vehículos para entender su funcionamiento interno. Esta técnica es fundamental no solo para diagnosticar y solucionar problemas, sino también para la innovación y mejora continua de los sistemas automotrices. La ingeniería inversa permite identificar y solucionar problemas en los vehículos, mejorar la eficiencia de los sistemas y desarrollar nuevas tecnologías y aplicaciones que optimicen el rendimiento y la seguridad de los automóviles.

El capítulo 1 ofrece una visión general del sistema CANbus y su relevancia en la industria automotriz. Se describen los componentes y funcionamiento del protocolo CANbus, la topología de la red, y los elementos esenciales como el controlador CAN, el transceptor y los cables del servidor de datos. Además, se abordan los protocolos de transmisión y las funciones de los campos de mensaje CAN, junto con una introducción al software LabVIEW y la tarjeta NI 9862.

El capítulo 2 se detalla el proceso de desarrollo del proyecto, que incluye la selección de herramientas y materiales, estrategias de recopilación de datos, y el diseño del sistema de PID (Parámetros de Identificación del Diagnóstico). Se proporciona una explicación del proceso de configuración. de la tarjeta NI 9862 y la implementación del software LabVIEW, así como las técnicas empleadas para recopilar y analizar datos de vehículos.

El capítulo 3 describe el proceso de desarrollo del panel de control que permite la visualización de datos de operación del vehículo JAC T8. Se incluyen detalles sobre la configuración de la tarjeta NI 9862, la apertura de sesiones de diagnóstico en la interfaz CAN, y la extracción de información del vehículo. Además, se aborda la implementación de

LabVIEW para crear interfaces gráficas intuitivas y el uso de la librería Automotive Diagnostic Command Set para el procesamiento de datos.

El capítulo 4 examina y discute los resultados obtenidos durante el desarrollo del proyecto. Se evalúa la efectividad del protocolo CAN 2.0 y el software LabVIEW en el diagnóstico y monitoreo vehicular, destacando la importancia de la validación de datos. También se discuten las mejoras en la precisión del diagnóstico y monitoreo, y la capacidad de adaptación del sistema para futuras mejoras.

La combinación de ingeniería inversa, protocolo CANbus y herramientas avanzadas de software como LabVIEW que ofrece una estructura sólida para generar sistemas de diagnóstico y monitoreo vehicular más avanzados y eficientes. La continua evolución y mejora de estas tecnologías promete optimizar aún más el rendimiento y la seguridad de los automóviles, beneficiando tanto a los fabricantes como a los usuarios finales.

PROBLEMA

En el campo automotriz, existen diversas herramientas para la adquisición de PID's del motor, como escáneres, módulos y aplicaciones que permiten extraer información del vehículo. Estos datos se utilizan para diagnóstico, lectura de sensores y postratamiento de datos, siendo de vital importancia para investigadores, mecánicos y estudiantes automotrices conocer en tiempo real los sistemas automotrices.

Sin embargo, las herramientas actuales presentan inconvenientes. Por ejemplo, algunas marcas de scanners solo permiten guardar datos en un formato preestablecido propio de la marca, y solo almacenan los fragmentos de datos visibles en la pantalla en tiempo real, de igual forma para el acceso de los datos se necesita un conocimiento básico previo para el manejo del equipo. En contraste, los módulos y aplicaciones permiten una extracción de datos más sencilla en archivos de texto o formato CSV, pero presenta el inconveniente de necesitar una configuración inicial para la extracción de datos

El problema de estudio identificado es la falta de una herramienta integral y eficiente para la monitorización en tiempo real de vehículos diésel, específicamente en el modelo JAC T8. Se requiere un panel de control intuitivo que permita al usuario seleccionar los datos y su representación visual, que genere gráficos y permita guardar los datos en formato de hoja de cálculo para su análisis correspondiente.

Delimitación del problema.

La delimitación geográfica de este estudio se reside en Quito, en Ecuador., la capital del país. Quito alberga una parte significativa del parque automotor de Ecuador, lo que genera una alta demanda de herramientas para el diagnóstico y monitoreo de vehículos.

La Universidad Politécnica Salesiana (UPS) sirve como base académica y de investigación para desarrollar una interfaz capaz de visualizar datos utilizando una tarjeta NI 9862 y el software LabVIEW, con la librería Automotive Diagnostic Command Set como base para el desarrollo de la solución. Esta plataforma se aprovechará por sus capacidades en el proceso de integración de software y hardware para la recolección de datos y la visualización de resultados.

ANTECEDENTES

La creación de un panel de control utilizando LabVIEW y la tarjeta NI-9862 para adquirir datos del CANbus es crucial en el contexto actual de la industria automotriz, que requiere herramientas productivas y precisas para el análisis y diagnóstico de sistemas electrónicos en vehículos. El protocolo CAN, ampliamente utilizado para el control distribuido en tiempo real, permite una comunicación robusta entre los módulos electrónicos de un vehículo. El trabajo de Pinguil (2023), destaca la importancia de analizar y diagnosticar la comunicación CAN en vehículos eléctricos, utilizando la tarjeta NI-9862 y la aplicación NI XNET Bus Monitor. Su estudio demostró que este enfoque facilita una comprensión detallada de la comunicación CAN, mejorando el diagnóstico de fallos y el mantenimiento predictivo.

De manera similar, el trabajo de Rojas (2019) implementó un modelo dinámico para analizar las tramas del bus de datos CAN en un vehículo eléctrico Kia Soul 2015, utilizando herramientas como Simulink y LabVIEW, junto con la tarjeta NI-9862. Su investigación contribuyó significativamente al monitoreo eficiente de la red de comunicación del vehículo, proporcionando una herramienta eficaz para identificar y analizar posibles fallos.

Por otro lado, el trabajo de Cervantes (2023) presentó una alternativa rentable a las interfaces de protocolo USB, desarrollando una interfaz USB-CAN personalizada que reduce la dependencia de interfaces con licencia. Esta investigación avanzó en el desarrollo de hardware para interfaces USB-CAN, ofreciendo una opción asequible para los procesos de prueba y desarrollo de software automotriz.

Adicionalmente, el trabajo de (Arévalo y Ortega ,2016) subraya la importancia de desarrollar una interfaz gráfica para la visualización y análisis de datos de sensores del motor y emisiones de gases. Utilizando LabVIEW, lograron una herramienta más precisa y detallada para el análisis de datos automotrices, demostrando una mayor eficiencia en comparación con la instrumentación externa tradicional.

En conjunto, estos antecedentes fundamentan la necesidad y relevancia de la presente investigación, cuyo objetivo es desarrollar un panel de control eficiente utilizando LabVIEW y la tarjeta NI-9862 para adquirir y analizar datos del CANbus. Este proyecto no solo busca

mejorar la precisión en el diagnóstico y monitoreo de vehículos, sino también ofrecer una herramienta accesible e integral que potencie los procesos de prueba y desarrollo automotriz. Los métodos y resultados de estudios previos respaldan la eficacia de esta aproximación, subrayando su potencial para contribuir significativamente al avance tecnológico en el ámbito automotriz.

Objetivo General.

Desarrollar un panel de control en LabVIEW con el módulo NI 9862 mediante la adquisición e interpretación de datos del sistema CAN BUS para obtener y monitorizar en tiempo real las variables de motor de una JAC T8, presentando la información de la unidad de control en formato visual y numérico.

Objetivos Específicos.

- Establecer comunicación con la línea CAN BUS del vehículo a través del puerto OBD II y la tarjeta NI 9862, para capturar datos en tiempo real del estado de variables de motor.
- Implementar una metodología para relacionar el PID con la variable de motor para su posterior extracción de payloads y su respectivo tratamiento con ecuaciones o escalares.
- Desarrollar algoritmos en LabVIEW para adquirir y procesar datos en tiempo real de los sensores MAP, MAF, RPM, Velocímetro y ECT mediante librerías provistas por el software.
- Diseñar una interfaz de usuario intuitiva en LabVIEW que proporcione representaciones gráficas y numéricas claras de los datos de los sensores mencionados, permitiendo un monitoreo efectivo del rendimiento del vehículo.

CAPÍTULO 1

1. MARCO TEÓRICO

1.1. Red CANbus

El Controller Area Network (CANbus) es un sistema de comunicación de red diseñado para conectar dispositivos distribuidos, como sensores, controladores y actuadores, a través de un bus de datos serial de alta velocidad. Fue desarrollado por Robert Bosch en colaboración con BMW y Mercedes, y se presentó como estándar internacional en 1994. Las normas ISO que componen el CANbus son la ISO 11898-1 para la capa de enlace de datos, la ISO 11898-2 para la capa física de alta velocidad y la ISO 11898-3 para la capa física de baja velocidad y tolerancia a fallos (Nyce, 2016).

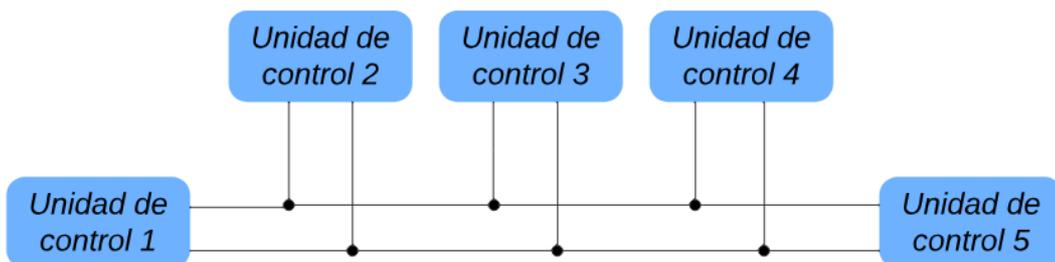
El sistema CAN bus se utiliza en vehículos debido a su capacidad para transmitir información entre diferentes componentes electrónicos de manera eficiente y confiable. Permite una comunicación de alta velocidad y es ampliamente utilizado en redes de vehículos modernos para garantizar una comunicación efectiva entre los distintos sistemas del automóvil.

1.2. Topología de la red

La conexión consiste en un bus de dos cables con un retorno común, el cual se termina en ambos extremos con resistencias que coinciden con la impedancia del bus característica. Generalmente es de 120Ω y proporciona una impedancia controlada que protege contra las interferencias causadas por las reflexiones de señal (Manzanares Caamal, 2020). La estructura utilizada se observa en la Figura 1

Figura 1

Estructura multiplexada en paralelo



Nota. Elaboración propia.

1.3. Componentes de la red

1.3.1. Controlador CAN

El dispositivo está ubicado en la ECU del vehículo y recibe datos que deben ser enviados al bus, convirtiéndolos a un formato comprensible para el transceptor CAN. Esta acción también puede ser inversa: obtiene y transfiere los datos del bus al transceptor y los traduce para que la unidad de control electrónico los entienda. (Sánchez Fernández, 2024). La ubicación y disposición con respecto a la UCE (Unidad de Control Electrónico) y transceptor se observa en la Figura 2.

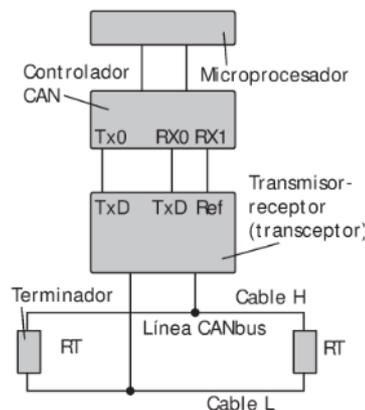
1.3.2. Transceptor (transmisor – receptor)

El transceptor es un circuito integrado ubicado en cada unidad de control, que está conectada al sistema y tiene una conexión directa con las líneas CAN High y CAN Low. El transceptor realiza las siguientes funciones en el sistema de comunicación del vehículo:

- Recibe datos del controlador, los convierte al formato necesario y los transmite al bus CAN.
- Toma datos del bus CAN, los procesa y los envía al controlador.
- Ajusta los niveles de tensión entre el bus y el controlador, amplificando la señal al enviarla al bus y reduciéndola al recibirla, garantizando la integridad y eficiencia de la transmisión de datos.

Figura 2

Ubicación del controlador CAN



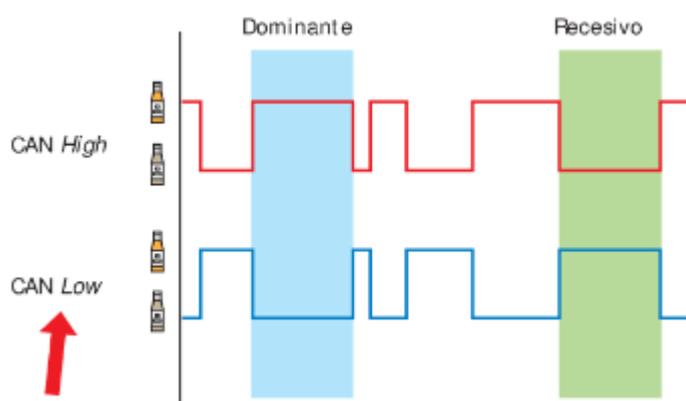
Nota. Adaptado de *Circuitos eléctricos auxiliares del vehículo* {p. 247}, por Sánchez Fernández, 2024, Macmillan Iberia, S.A.

1.3.3. Cables del bus de datos

El bus de datos está compuesto por dos cables que actúan como la vía común para la circulación de información entre las unidades de control electrónico (UCE) conectadas en paralelo. La información se transmite mediante la diferencia de tensión entre estos cables, un método conocido como transmisión diferencial. El cable de tensión alta se denomina H (High) y el de tensión baja L (Low). Este método minimiza las perturbaciones en la transmisión, ya que la diferencia de tensión se mantiene constante. Los datos en ambos cables son idénticos, aunque se expresan mediante impulsos de tensión opuestos. La transmisión diferencial del sistema CAN es resistente a interferencias electromagnéticas, ya que ambos cables reciben las mismas perturbaciones, manteniendo la señal diferencial intacta (Sánchez Fernández, 2024). En la Figura 3 se observa los estados que posee cada línea de comunicación.

Figura 3

Representación de línea CAN High y CAN Low



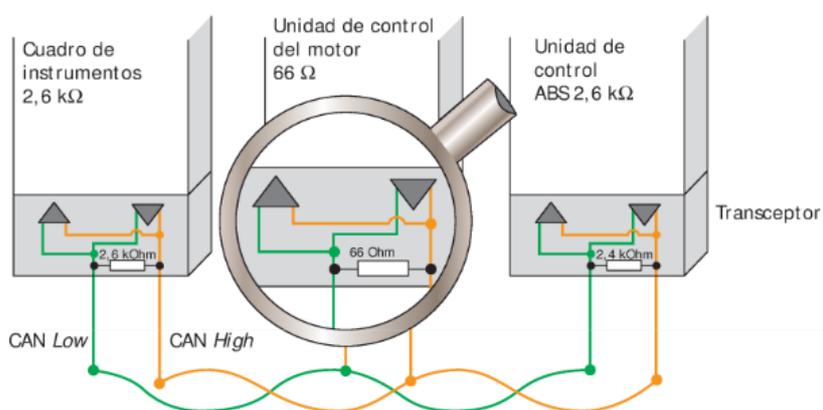
Nota. Adaptado de *Circuitos eléctricos auxiliares del vehículo* {p. 247}, por Sánchez Fernández, 2024, Macmillan Iberia, S.A.

1.3.4. Terminadores o elementos de cierre

Los terminadores son resistencias situadas al principio y al final de los cables del bus de datos, normalmente dentro de las UCE en los extremos del bus. Su propósito es ajustar el bus para diferentes longitudes de cables y número de centralitas conectadas, evitando la reflexión de los mensajes al final del bus para asegurar un funcionamiento sin interferencias. A veces, esta función se distribuye entre varias resistencias en las diferentes UCE como se observa en la Figura 4.

Figura 4

Terminadores repartidas entre las diferentes UCE



Nota. Adaptado de *Circuitos eléctricos auxiliares del vehículo* {p. 249}, por Sánchez Fernández, 2024, Macmillan Iberia, S.A.

1.4. Protocolo de transmisión

En el sistema CAN, todas las UCE actúan como maestras, lo que significa que una emite un mensaje y todas las demás lo reciben simultáneamente. Cada una decide si la información recibida es relevante y, en caso contrario, la ignora. Este método de transmisión se llama broadcast (radiodifusión). Los mensajes o tramas enviados, conocidos como protocolos, están compuestos por numerosos bits organizados en siete campos, cada uno con una función específica.

1.5. Funciones de los campos del mensaje CAN

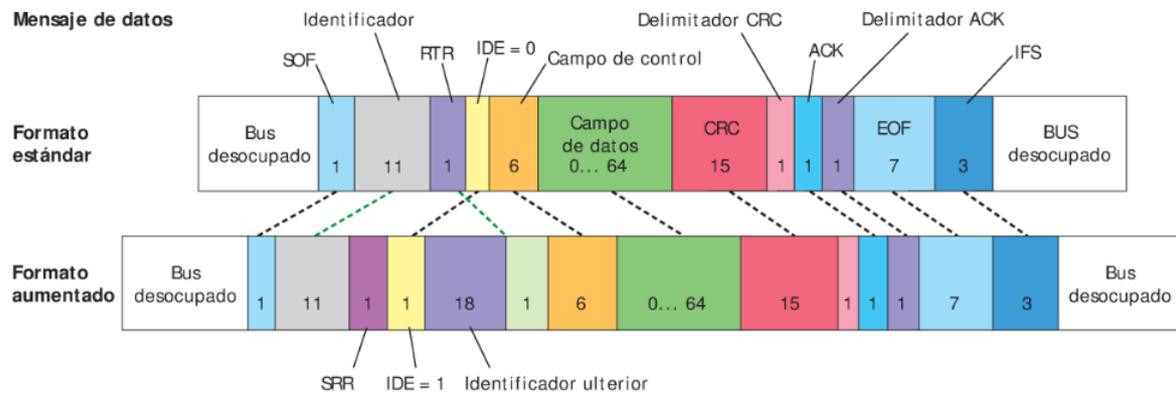
Los mensajes de datos CAN se componen de siete campos, en la Figura 5 se observa la estructura y división de cada sección.

Se procede a explicar el campo y sus funciones específicas:

- Campo de inicio, SOF (1 bit): Marca el comienzo de la trama colocando un bit dominante 0 en la línea, indicando que está a nivel alto y libre.
- Campo de arbitrio o de estado (12-30 bits): Tiene dos partes:
 - Identificador ID (11-29 bits): Identifica al elemento que está transmitiendo y determina la prioridad dominante (el primero que pone un 0 en este campo).
 - RTR (Remote Transmission Request - 1 bit): Indica si envía un 0 o recibe un 1.
- Campo de comando o control (6 bits): Incluye:
 - RES (2 bits): Reservado para futuras evoluciones.
 - DLC (4 bits): Indica la longitud del campo de datos con combinaciones binarias que representan el número de bytes (por ejemplo, 0001 = 1 byte; 0100 = 4 bytes).
- Campo de datos (0-64 bits): Contiene el mensaje a enviar, cuyo tamaño se determina por el valor de DLC. Si RTR es 1, este campo será reemplazado por otro identificador.
- Campo de verificación (16 bits): Detecta errores en la transmisión, compuesto por:
 - CRC (15 bits): Verifica y localiza errores en los bits.
 - CRC DEL (1 bit): Delimitador que indica el final del CRC y siempre vale 1.
- Campo de acuse de recibo o confirmación (2 bits): Informa al transmisor sobre la correcta recepción de los datos. Si la recepción no es correcta, se indica al transmisor que repita el mensaje. Incluye:
 - ACK (1 bit): Bit de acuse de recibo (0 = ok; 1 = no ok).
 - ACK DEL (1 bit): Delimitador que indica el final del ACK y siempre vale 1.
- Campo de fin de trama (7 bits): Termina el mensaje y proporciona la última oportunidad para señalar un error y repetir el mensaje si es necesario. EOF (7 bits), siete bits a nivel alto, indica que la línea está libre. IFS (3 bits) es el espacio intertrama, una trama libre de al menos 3 bits.

Figura 5

Campos de mensaje de datos CAN



Nota. Adaptado de *Circuitos eléctricos auxiliares del vehículo* {p. 250}, por Sánchez Fernández, 2024, Macmillan Iberia, S.A.

1.6.OBD

El OBD I (On-Board Diagnostics) era un sistema primitivo de diagnóstico a bordo para vehículos. Permitía identificar componentes deteriorados, pero era rudimentario y no cubría todas las necesidades de diagnóstico.

1.6.1. OBD II

El OBD II, desarrollado en 1996, es una evolución significativa del OBD I. Este sistema no solo detecta fallas en el motor y la transmisión, sino que también supervisa sistemas de emisión de gases (EVAP, NOx y EGR), sensores de oxígeno, aire acondicionado, sensores de presión de neumáticos, y más. Es capaz de emitir alertas y almacenar datos críticos para el mantenimiento y diagnóstico del vehículo. Desde los años 2000, el OBD II también se encarga de monitorear la presión de los neumáticos, y, desde 2008, todos los vehículos en EE.UU. deben usarlo, promoviendo un control unificado en mercados como Europa y Asia. Este sistema usa códigos de falla estandarizados y una interfaz de conexión (ISO J1962) para la comunicación entre la ECU y dispositivos externos, facilitando el mantenimiento predictivo y la detección de fallos. Además, el OBD II contribuye a la reducción de emisiones, asegurando que los vehículos cumplan con las regulaciones ambientales (Concepcion, 2011).

El OBD II es crucial para el diagnóstico y mantenimiento de vehículos modernos, proporcionando datos precisos y detallados que ayudan en la identificación y resolución de

problemas, mejorando la eficiencia y seguridad del vehículo. Además, facilita pruebas de emisiones y mantiene el cumplimiento de normativas ambientales.

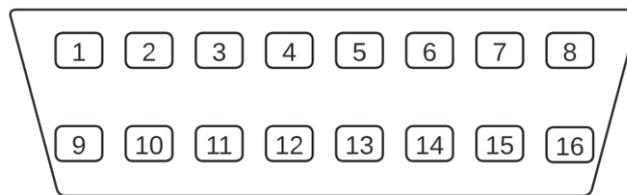
Los protocolos que conforman el sistema OBD-II incluyen

- ISO 14230-4 (KWP2000)
- SAE J1850 PWM
- SAE J1850 VPW
- SAE J2284 CAN
- ISO 15765-4 CAN
- ISO 9141-2

En la Figura 6 se observa un Diagnostic Link Conector y su distribución de pines enumerados.

Figura 6

Pines de conexión Diagnostic Link Conector (DLC) ISO J1962 tipo A



Nota. Elaboración propia.

Asignación de los pines de conexión.

1. Libre
2. Línea positiva SAE-J1850 (BUS)
3. Libre
4. Chasis
5. Masa de señal
6. CAN alto
7. Línea K (ISO 9141-2 / ISO 14230-4).
8. Libre
9. Libre
10. Línea negativa SAE-J1850
11. Libre

- 12. Libre
- 13. Libre
- 14. CAN bajo
- 15. Línea L
- 16. Positivo de batería

1.7.PIDs

Los PIDs (Identificadores de Parámetros) son códigos utilizados para solicitar datos de un vehículo, sirviendo como herramienta de diagnóstico.

1.7.1. PIDs Estándar

Los PIDs estándar de OBD-II, según lo definido por el SAE J1979, proporcionan una respuesta esperada para cada PID junto con información sobre cómo interpretar esos datos de manera significativa. No todos los vehículos soportan todos los PIDs, y pueden existir PIDs personalizados definidos por el fabricante que no están incluidos en el estándar OBD-II. En la investigación sobre los PIDs de MAP, MAF, RPM, tacómetro y ECT, se consultaron blogs, foros y páginas web. Por lo que las fuentes de información son variadas, de esta manera se recopilaron los siguientes identificadores con sus respectivas fórmulas, representándolos en la Tabla 1.

Tabla 1*Valor de PID's con descripción*

PID (Hex)	PID (Dec)	N° Byte	Descripción	Valor Min.	Valor Max.	Unidades	Ecuación
0C	12	2	Velocidad real de motor (RPM)	0	16383,75	rpm	$\frac{256A + B}{4}$
0B	11	1	Intake manifold absolute pressure MAP	0	255	kPa	A
05	5	1	Temperatura del refrigerante del motor ECT	-40	215	°C	A - 40
0D	13	1	Tacómetro Sensor de	0	255	km/h	A
10	16	2	flujo de masa de aire MAF	0	655.35	g/s	$\frac{256A + B}{32}$

1.8.LabVIEW

LabVIEW se enfoca en programación grafica la cual ofrece aceleradores de productividad exclusivos para el desarrollo de sistemas de prueba Destaca por su enfoque intuitivo, conectividad con cualquier instrumento e interfaces de usuario integradas. Simplifica la integración de hardware, permitiendo adquirir datos de dispositivos de NI y terceros, y facilita la visualización inmediata resultado de interfaces de usuario que funcionan con clic y arrastre. Además, permite producir algoritmos de análisis de datos y control avanzado, y es compatible con otras herramientas de ingeniería y lenguajes de programación como

C/C++ y Python, integrando y reutilizando bibliotecas de otros software (National Instrument, 2024).

1.9.NI 9862

El NI-9862 es un módulo de interfaz CAN de alta velocidad con razón de datos flexible, diseñado para aplicaciones que requieren manipulación en tiempo real de numerosos marcos y señales CAN, como simulaciones hardware-in-the-loop, prototipos de control rápido, monitoreo de bus y automatización.

La NI 9862 es compatible con las plataformas NI CompactDAQ y NI CompactRIO, lo que permite realizar mediciones de adquisición de datos adicionales en paralelo mientras se manipulan tramas y señales CAN a alta velocidad. Además, su integración con el controlador Bosch DCAN y el transceptor NXP TJA1041AT facilita su uso en diversas aplicaciones automotrices y de control embebido, en la Figura 7 se puede observar la tarjeta junto con el cable USB y el adaptador para el puerto OBD.

Figura 7

Tarjeta NI-9862 y accesorios

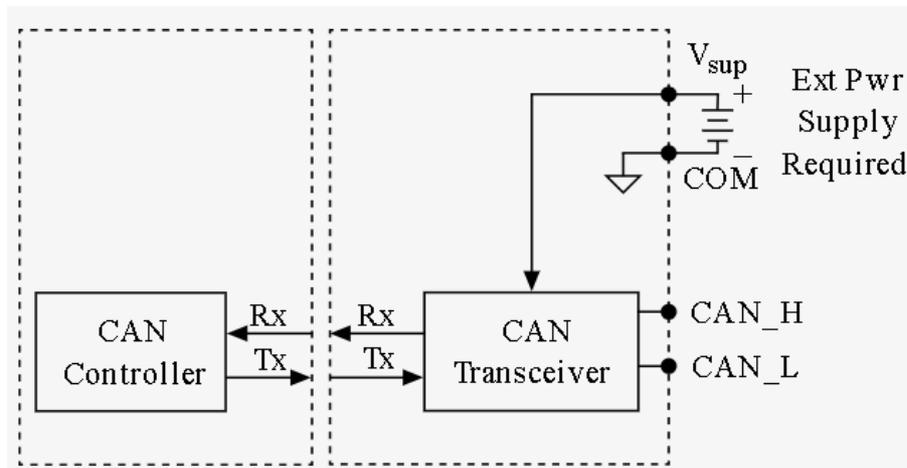


Nota. Elaboración propia.

La tarjeta NI-9862 está equipada con un controlador Bosch DCAN compatible con CAN 2.0B (11 y 29 bits) y un transceptor NXP TJA1041AT, cumple con la norma ISO 11898 y admite velocidades de hasta 1 Mbps (National Instrument, 2023). En la Figura 8 se puede observar su configuración interna.

Figura 8

NI-9862 Descripción general del hardware



Nota. Adaptado de *NI-9862 Getting Started, National Instrument, 2023*, <https://www.ni.com/docs/en-US/bundle/ni-9862-getting-started/page/overview.html>.

1.10. Librería Automotive Diagnostic Command Set Toolkit

El Automotive Diagnostic Command Set (ADCS) Toolkit ayuda a diseñar e implementar aplicaciones de diagnóstico automatizado para la prueba de producción de ECU. El ADCS es un complemento de software para LabVIEW. Es adecuado para aplicaciones de diagnóstico automatizado, incluyendo lectura y escritura de parámetros, acceso a códigos de error (DTC), actualización de firmware de la ECU e inicio de modos de prueba de diagnóstico de la ECU. Además, puede desarrollar e implementar una aplicación sin necesidad de licencias adicionales. Este complemento es compatible con muchos dispositivos de interfaz automatizado de NI y soporta protocolos como UDS, OBD, y WWH-OBD (National Instrument, 2024).

CAPÍTULO 2

2. METODOLOGÍA

Esta tesis analiza la integración y utilizando tecnologías sofisticadas para visualizar y analizar datos en tiempo real en el contexto del desarrollo de un panel de control para el vehículo JAC T8. Se establece un sistema robusto para la adquisición de datos a través del protocolo CANbus utilizando el entorno de programación LabVIEW y la tarjeta NI 9862. Se utilizan técnicas de diagnóstico de la librería Automotive Diagnostic Command Set (ADCS). Esta combinación de herramientas y métodos permite una interacción efectiva con los sistemas vehiculares, lo que facilita una recolección y procesamiento de datos precisos, que son esenciales para la monitorización y evaluación del rendimiento del vehículo bajo diversas condiciones operativas.

En el proyecto se usa un método estructurado para obtener datos de diagnóstico de potencia de vehículos a través de solicitudes OBD-II, destacando la importancia de elegir parámetros específicos, como el PID, para analizar y tomar decisiones. Además, se habla sobre el diseño optimizar la interfaz de usuario de LabVIEW, que facilita la visualización de datos importantes del motor del vehículo, como temperatura del refrigerante (ECT), flujo de masa de aire (MAF), velocidad del vehículo (tacómetro) y presión absoluta del múltiple de admisión (MAP).

Este proyecto utiliza la metodología heurística, mediante la ingeniería inversa se obtiene información del sistema del bus de comunicación preestablecido por el fabricante y se busca identificar la trama de datos de los sensores antes mencionados para la constitución de una interfaz con un entorno simple de manejar. El enfoque de prueba y error se utiliza para mejorar la interpretación de las señales del CANbus y la adquisición de datos, asegurando que los datos recopilados sean confiables, pertinentes y útiles para el diagnóstico. y mantenimiento de la JAC T8.

El desarrollo del proyecto inicia en la línea de comunicación CANbus del automóvil, donde se encuentra la información de todos los procesos de actuadores y sensores. Para observar el flujo de información, se utilizó en primera instancia la aplicación NI-XNET Bus Monitor, que mostró los identificadores junto con sus respectivos datos (payloads). Mediante el programa, se buscó la relación de cada identificador con los PID's obtenidos en la

recopilación de información basada en la norma SAE J1979, aplicando las ecuaciones correspondientes para calcular los valores respectivos de cada sensor.

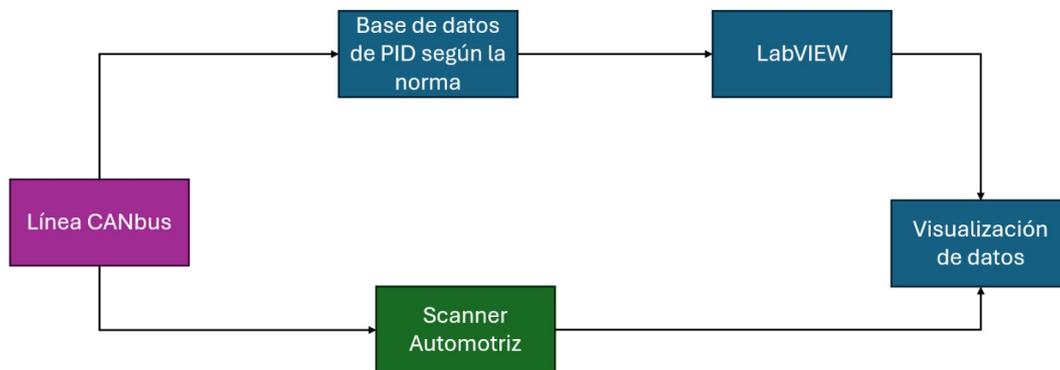
Al no encontrar coincidencias entre los valores del bus monitor y los PID's de la norma, se conectó en paralelo un escáner externo a través del puerto OBD II, el cual permite obtener los valores de los parámetros del motor. En el bus monitor se observaron dos nuevos identificadores que se relacionan con la ECU (Unidad de Control del Motor), lo que llevó a un cambio en la estrategia metodológica del proyecto.

2.1. Esquema de metodología

En el análisis de los datos CAN de la camioneta JAC T8 se procedió a usar un escenario denominado “encendido” donde el vehículo se encuentra en funcionamiento, este se empleará en ambas rutas de extracción de datos como se observa en la Figura 9 .

Figura 9

Esquema de metodología



Nota. Elaboración propia.

Se utiliza LabVIEW con sus librerías para la búsqueda manual de PID's disponibles y comparando con la información de la norma SAE J1979 haciendo uso de ecuaciones para la transformación de valores, posteriormente se utiliza un scanner comercial que extraiga datos de motor ingresando por OBD. Para validar los datos se compara los valores de los respectivos sensores de ambas rutas.

2.2. Selección de Herramientas y Materiales

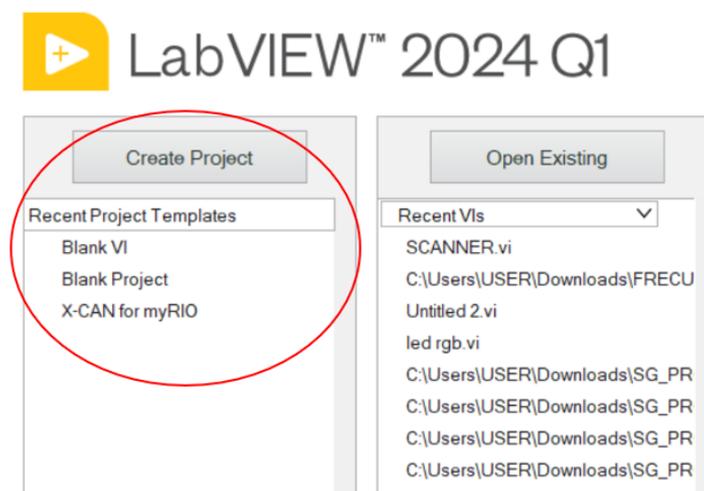
Para la selección de un entorno de visualización de datos y diseño de interfaz es necesario que cumpla con una amplia gama de herramientas para la adquisición e interpretación de datos dentro de un marco técnico. El cual, es un proceso crucial que permite convertir datos numéricos en datos útiles y pertinentes para la toma de decisiones, el análisis y la comprensión de fenómenos.

2.2.1. Software

En la Figura 10 la selección de LabVIEW 2024 Q1 como interfaz principal de visualización de datos en tiempo real es una versión específica que proporciona mejoras en el rendimiento y capacidades avanzadas que son ideales para el manejo y visualización de datos, lo que lo hace adecuado para aplicaciones en el sector automotriz. Se propone el uso de ciertas librerías que garantizan la comunicación con el puerto OBD-II dentro de las líneas de transmisión CANbus.

Figura 10

Visualización de entorno de LabVIEW



Nota. Elaboración propia.

Dentro del entorno de programación LabVIEW, la librería Automotive Diagnostic Command Set (ADCS) desempeña un papel crucial. Esta librería está diseñada para facilitar

la comunicación entre las aplicaciones de software y los sistemas de vehículos a través del protocolo OBD-II permite solicitar y recibir datos de diagnóstico en tiempo real. Además, ADCS se utiliza para comunicarse con otros sistemas, como la unidad de control del motor de vehículos, proporcionando un método estandarizado y confiable para solicitar parámetros del vehículo referentes a los planteados en el proyecto. Una visualización general de las funciones provistas por la librería se observa en la Figura 11.

Figura 11

Librería de Automotive Diagnostic Command Set de LabVIEW



Nota. Elaboración propia.

El uso de la librería ADCS dentro de la tesis permite la implementación de funciones de diagnóstico avanzado, esencial para la vigilancia y el análisis en tiempo real del estado del vehículo JAC T8. Por ejemplo, mediante el uso de esta librería, se desarrollan algoritmos específicos en LabVIEW que solicitan, interpretan y visualizan datos del CANbus en una interfaz de usuario intuitiva. Esto no solo mejora la capacidad de diagnóstico del sistema, sino que también facilita la comprensión del comportamiento del vehículo bajo diferentes condiciones de operación, lo que es vital para la evaluación de la eficiencia del vehículo y la observación temprana de posibles anomalías o fallas. Así, la librería ADCS se convierte en un componente crucial del proyecto de tesis, contribuyendo significativamente a la precisión y la funcionalidad del panel de control desarrollado.

2.2.2. Hardware

La herramienta encargada de la recolección de datos es la tarjeta NI 9862, que actúa como puente entre los programadores y la línea de comunicación CANbus del automóvil, conectándose mediante el puerto OBD II. A diferencia de otras tarjetas programables como Arduino o ESP32, se optó por la tarjeta NI debido a sus prestaciones superiores.

La tarjeta soporta una tasa de baudios máxima de 1 Mbps, lo que garantiza una rápida transmisión y recepción de datos, importante para aplicaciones que necesitan manipulación en tiempo real de múltiples tramas CAN. La NI 9862 cuenta con aislamiento completo de su puerto CAN respecto a otros módulos del sistema, mejorando la seguridad y la fiabilidad. Este aislamiento protege contra interferencias eléctricas y fallos, lo que es esencial en entornos industriales.

Además, la NI 9862 es eficiente energéticamente, con un consumo máximo de potencia de 1 W en modo activo y una disipación térmica baja. El software controlador NI-XNET facilita la integración de la tarjeta con diversas aplicaciones y herramientas de desarrollo de National Instruments, como LabVIEW, simplificando el desarrollo y poner en práctica aplicaciones de diagnóstico y control.

Finalmente, se consideró el aspecto económico, aprovechando los recursos disponibles en la Universidad Politécnica Salesiana para minimizar costos y fomentar el desarrollo académico, haciendo de la NI 9862 la opción ideal para el proyecto.

En el contexto del proyecto de tesis para el panel de control vehicular de la camioneta JAC T8, el uso de LabVIEW permite interpretar datos de forma sistemática y estructurada, y la tarjeta NI 9862 garantiza precisión y eficiencia en el proceso.

2.3. Estrategias de Recolección de Datos

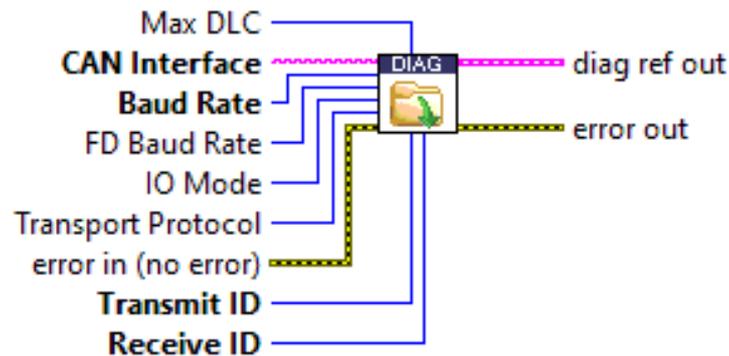
El método de recopilación de información requiere programación. en el software LabVIEW que permita la selección y visualización de datos adquiridos por la tarjeta NI 9862, esto implica varias etapas clave, cada una diseñada para asegurar la utilidad de los datos recogidos.

2.3.1. Iniciador de sesión de diagnóstico CAN

La función "Open Diagnostic on CAN FD.vi" del conjunto de comandos de diagnóstico del Automotive Diagnostic Command Set (ADCS). Esta función es esencial para iniciar una sesión de diagnóstico en una interfaz CAN (Controller Area Network). En la Figura 12 se observa la estructura, así como pines de salida y entrada

Figura 12

Función "Open Diagnostic on CAN FD"



Nota. Elaboración propia.

A continuación, se detalla cada componente de esta función y su propósito:

- **Max DLC (Data Length Code):** Define la longitud máxima de los datos que se pueden ser enviados o recibidos en un solo mensaje CAN. Este parámetro asegura que la sesión de diagnóstico pueda manejar la cantidad correcta de datos necesarios para el diagnóstico.
- **CAN Interface:** Selecciona la interfaz CAN específica que se utilizará para la sesión de diagnóstico. Esta interfaz actúa como el canal de comunicación entre el software de diagnóstico y el vehículo.
- **Baud Rate:** Establece la velocidad de transmisión de datos en comunicación de bits por segundo (bps). La tasa de baudios es crucial para garantizar que la comunicación sea lo suficientemente rápida para soportar el flujo de datos en tiempo real.
- **IO Mode (Input/Output Mode):** Define el modo de operación de entrada y salida para la sesión de diagnóstico. Este parámetro permite configurar cómo se manejarán los datos entrantes y salientes durante la sesión de diagnóstico.
- **Transport Protocol:** Selecciona el protocolo de transporte que se utilizará, como ISO-TP (ISO 15765-2), común para el diagnóstico en redes CAN. El protocolo de

transporte garantiza que los datos se transmitan de manera estructurada y entendible entre el vehículo y el sistema de diagnóstico.

- **Transmit ID:** Especifica el identificador de mensaje CAN que se utilizará para transmitir datos. Este identificador es único para cada tipo de mensaje y asegura que los datos enviados sean correctamente interpretados por el vehículo.
- **Receive ID:** Especifica el identificador de mensaje CAN que se utilizará para recibir datos. Similar al Transmit ID, este identificador asegura que los datos recibidos sean los esperados y se procesen adecuadamente.
- **Terminal Data Type:** Define el tipo de datos de los terminales de salida y entrada para la interfaz CAN. Esto asegura que los datos se manejen en el formato correcto durante la transmisión y recepción.

Flujo de Trabajo de la Función:

- **Apertura de Sesión:** La función abre una sesión de diagnóstico en la interfaz CAN especificada, permitiendo que el software de diagnóstico comience a comunicarse con el vehículo.
- **Configuración de Parámetros:** Los parámetros como la velocidad de transmisión y los identificadores de mensaje son configurados para asegurar una comunicación precisa y eficiente.
- **Inicialización del Protocolo de Transporte:** Se inicializa el protocolo de transporte seleccionado, permitiendo la correcta comunicación y transmisión de datos diagnósticos. Esto es fundamental para garantizar que los datos se transfieran sin errores y se puedan interpretar correctamente.

Se debe aclarar que a pesar de que la función tiene el nombre de CAN FD está nos permite seleccionar el protocolo de comunicación I/O mode CAN 2.0.

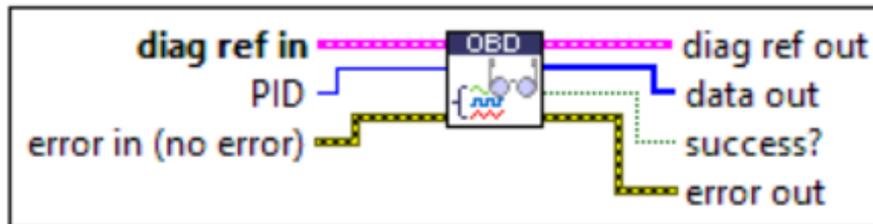
2.3.2. Identificación de variables

El diseño de la interfaz requiere la extracción de un PID específico de la trama de datos del bus. Esto es específicamente cierto para los sensores MAP (Sensor de presión absoluta de admisión múltiple), MAF (Sensor de flujo de masa de aire), ECT (Sensor de temperatura del refrigerante del motor), tacómetro (Velocidad del vehículo) y la variable de velocidad de motor en RPM (Revoluciones por minuto). Para ello, se requiere la capacidad de manipular el valor del PID y obtener en respuesta los valores contenidos (payloads). Por esta razón, se

optó por utilizar la función "Request Current Powertrain Diagnostic", que permite visualizar la información necesaria de manera eficiente. En la Figura 13 se observa la mencionada función.

Figura 13

Función "OBD Request Current Powertrain Diagnostic"



Nota. Elaboración propia.

Descripción general de los componentes y su propósito:

- **Diag ref in:** Este es el terminal de referencia de diagnóstico de entrada. Es una referencia a la sesión de diagnóstico que está utilizando la función.
- **Diag ref out:** Este es el terminal de referencia de diagnóstico de salida. Después de que la función haya procesado los datos, la referencia de diagnóstico actualizada se pasa a través de este terminal.
- **PID:** Este terminal se utiliza para especificar el "Parameter ID" (Identificador de Parámetro) que se desea leer del vehículo. Los PID son códigos que identifican datos específicos del vehículo.
- **Error in (no error):** Este terminal es una entrada de clúster de error que se usa para la gestión de errores. Se debe conectar al error out del bloque anterior en el diagrama.
- **Error out:** Este terminal es una salida de clúster de error que proporciona información sobre cualquier error que haya ocurrido durante la ejecución de la función.
- **Data out:** Este terminal proporciona los datos leídos del vehículo, basados en el PID especificado. Los datos de respuesta se devuelven en un formato "array" de una dimensión.
- **Success?:** Este terminal indica si la operación se ha completado con éxito (true) o no (false).

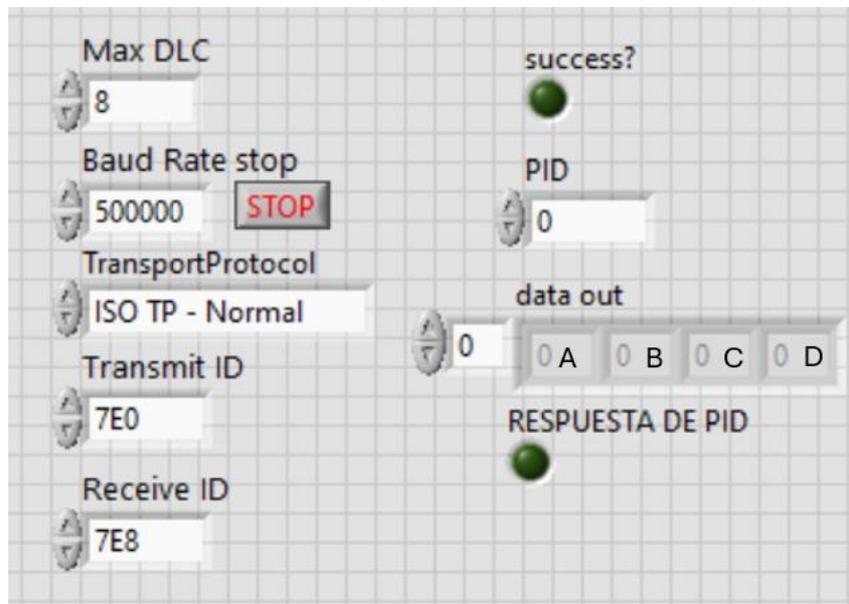
2.3.3. Diseño del Sistema de PID

Con las funciones antes descritas se diseñó dentro del entorno de LabVIEW un sistema para el acceso a la información de la línea CANbus que permite buscar manualmente todos los PID's disponibles teniendo como base la metodología heurística y haciendo uso de la ingeniería inversa. Identificado así los PID's ocupados por el fabricante para transmitir la información de motor, los cuales varían de 0 a 255 delimitados por la norma SAE J1979.

En el extremo izquierdo de la Figura 14 se muestra la configuración de la transmisión de datos a través del CANbus, destacando un Baud Rate de 500 kbps, en conformidad con la norma ISO 11898-1 contenida en la norma ISO 15765-4. El tipo de protocolo empleado está definido bajo la norma ISO TP (15765-2).

Figura 14

Interfaz de obtención de datos



Nota. Elaboración propia.

El proceso de transmisión y recepción de datos contenida en la norma ISO 15765-4, específicamente para el módulo del motor (ECU) como se observa en la Tabla 2. Cabe aclarar que la información presentada es la recopilación de varias fuentes referentes a la norma ISO 15765-4

Tabla 2*Identificadores CAN de 11 bits para OBD*

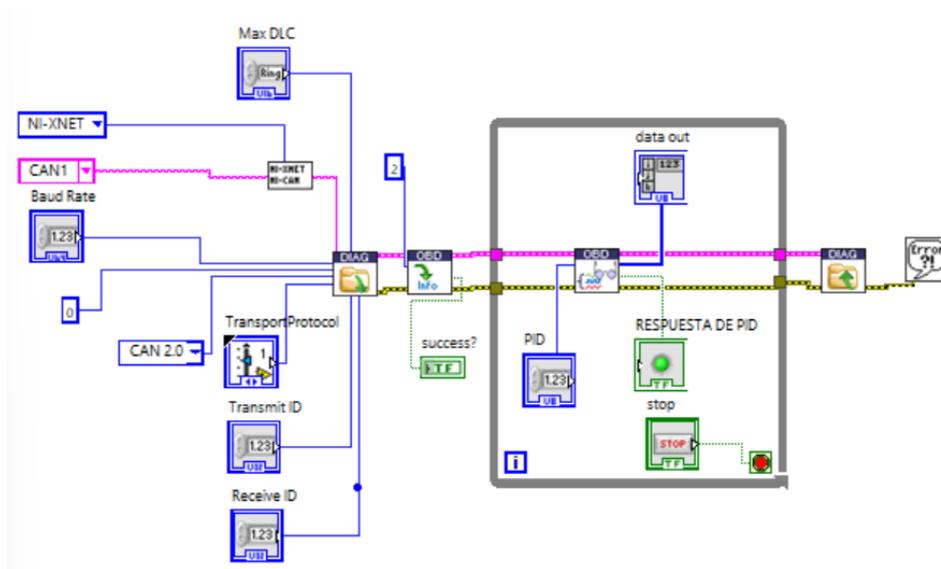
Identificador CAN {hex}	Descripción
7E0	Identificador CAN de solicitud física desde el equipo de pruebas externo hacia la ECU #1.
7E8	Identificador CAN de respuesta física desde la ECU #1 hacia el equipo de pruebas externo.

Adicionalmente, en la parte superior derecha de la Figura 14, donde se observar un controlador denominado PID. Y un indicador para verificar que los datos de salida o 'data out' estén efectivamente disponibles y accesibles en el módulo, en este caso se observa el tipo de dato es de números enteros de una dimensión y cada posición 0,1,2,3 representa la letra A, B, C, D respectivamente, esto será de utilidad en las ecuaciones. Este sistema es esencial para garantizar la correcta comunicación y funcionamiento del sistema de diagnóstico vehicular, proporcionando una interfaz vital para el análisis y monitoreo en tiempo real del estado del motor y otros sensores del módulo del motor.

En la Figura 15 se visualiza el diagrama de bloques de programación en LabVIEW del prototipo para obtener datos del módulo del motor, se puede observar todas las conexiones entre la función “Open Diagnostico on CAN FD” el cual da paso a la configuración y entrada de datos del CANbus. Abriendo una sesión de diagnóstico en la interfaz CAN que da paso a la función “OBd Request Current Powertrain Diagnostic” que solicita información al módulo y da acceso a los PID’s, esta última función dentro de una estructura “while” para evitar que se detenga el flujo de información.

Figura 15

Diseño en el block diagrama para obtención de datos



Nota. Elaboración propia.

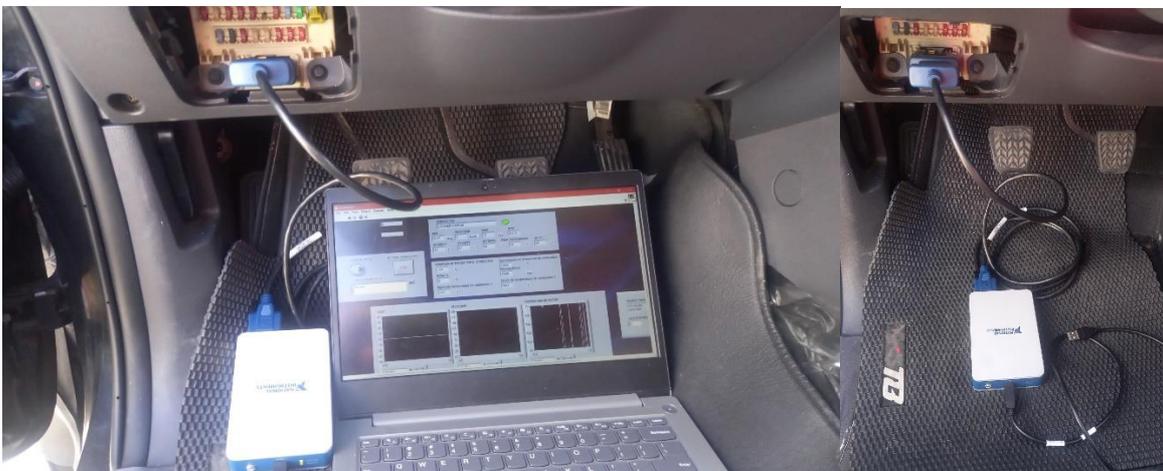
CAPÍTULO 3

3. DESARROLLO DE PANEL DE CONTROL

En este capítulo se aborda la configuración y uso de la tarjeta NI 9862 para la adquisición eficiente de datos del vehículo JAC T8 mediante el protocolo CAN BUS. En la Figura 16 se procede a mostrar la conexión con la NI 9862:

Figura 16

Conexión puerto OBD con tarjeta NI-9862



Nota. Elaboración propia.

Este sistema utiliza la interfaz “Open Diagnostic on CAN FD” de LabVIEW, optimizada para el vehículo JAC T8, recordando que a pesar que el nombre de la función es CAN FD, si permite realizar el cambio y seleccionar I/O mode CAN 2.0. La configuración implica ajustar el número de bits, específicamente a 8 bits en Max DLC, y la interfaz CAN seleccionada en CAN1 con la configuración NI-XNET. Se establece un baud rate de 500 kbps conforme a la norma ISO 11898-1, empleando el protocolo ISO TP – Normal Mode para garantizar una transmisión y recepción de datos conforme a estándares internacionales.

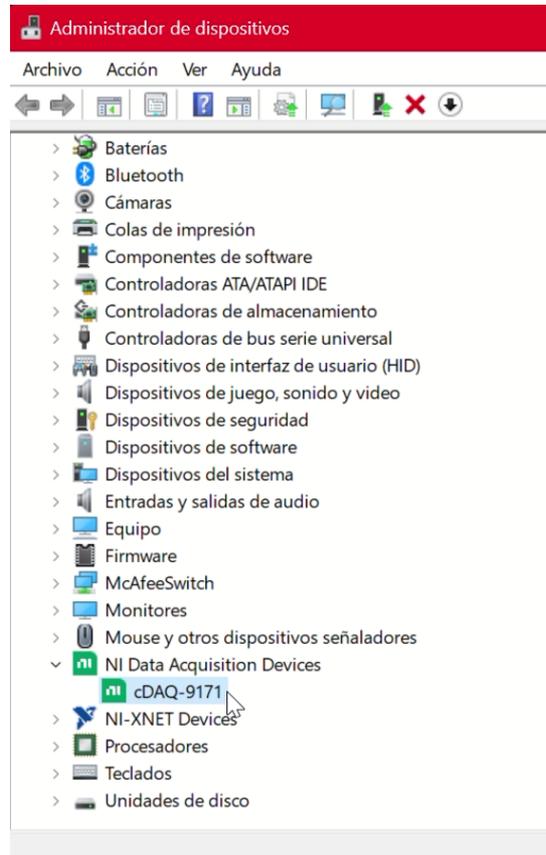
3.1. Conexión de la Tarjeta NI-9862

Para la configuración de la NI-9862 es necesario conectar el adaptador de la tarjeta al puerto OBD II del vehículo, y el otro cable tipo USB b directo al computador. Es necesario verificar

que la computadora reconozca el dispositivo mediante el administrador de dispositivos como se muestra en la Figura 17

Figura 17

Ventana de administrador de dispositivos de Windows

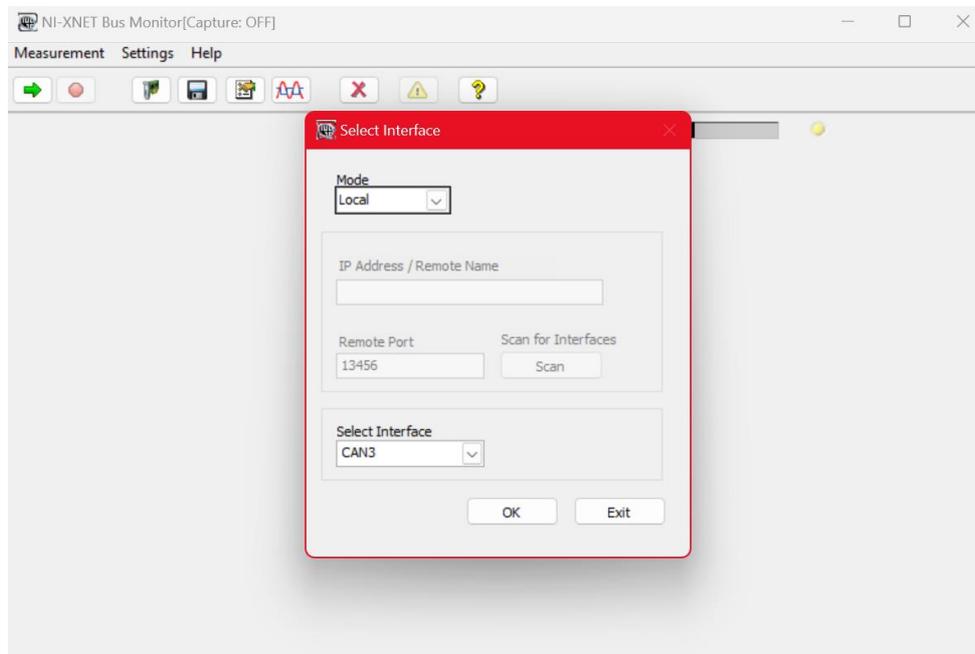


Nota. Elaboración propia.

De igual manera se procede a verificar la interfaz de CAN, la cual hace la comunicación con el vehículo y la computadora. En la Figura 18 procedemos buscar el puerto CAN en la aplicación NI-XNET Bus monitor:

Figura 18

Ventana NI-XNET Bus Monitor identificación de puerto



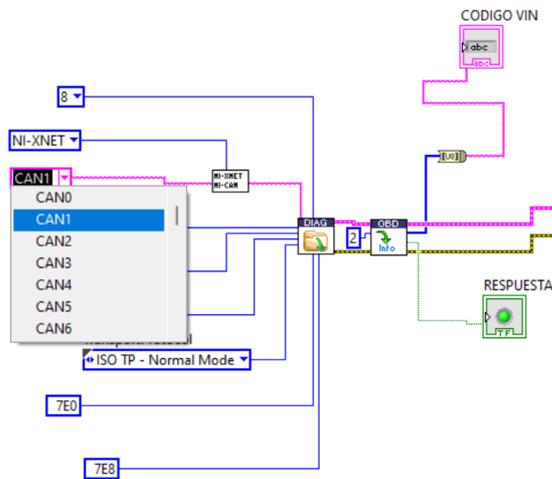
Nota. Elaboración propia.

Se hace uso de NI XNET Bus Monitor, el cual, verifica el puerto de transmisión con la NI 9862. Así se configura la tarjeta con el ordenador, el puerto de transmisión varía con diferentes computadoras al reconocer su puerto.

En la Figura 19 se procede a realizar la selección de puerto CAN de entrelazamiento con el dispositivo NI 9862:

Figura 19

Selección de puerto CAN en la interfaz de LabVIEW



Nota. Elaboración propia.

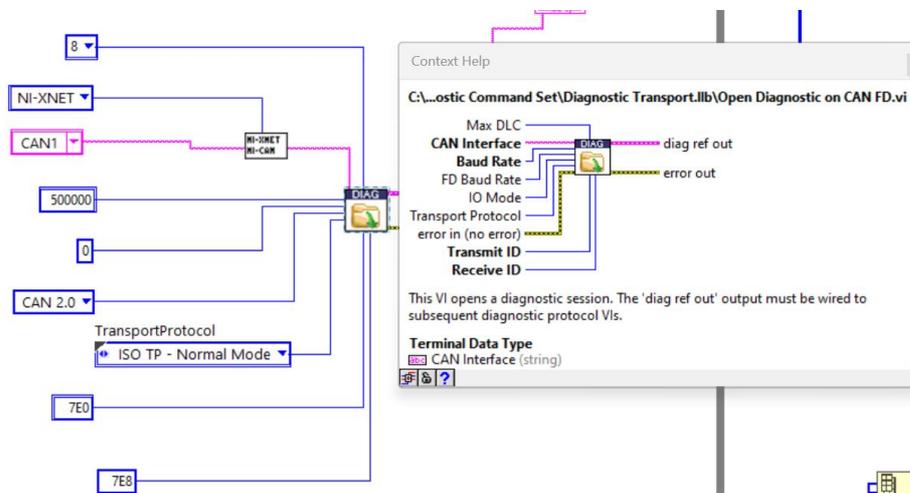
3.2. Configuración de parámetros para apertura de sesión de diagnóstico en la interfaz CAN

En la Figura 20 se visualiza la configuración estándar para el vehículo JAC T8, donde se configura el número de bits que entran, esta configuración es de 8 bits en Max DLC. Lo que respecta al “CAN Interface” se visualiza una configuración con la National Instrument (NI-XNET) y el puerto CAN de entrelazamiento con el dispositivo NI 9862.

Lo que respecta al “Baud Rate” se configuró a 500 kbps determinado por la norma ISO 11898-1, esta norma es aplicada en el protocolo de ISO TP – Normal Mode, de esta manera los códigos de transmisión y envío de datos se establece una norma internacional que garantiza la comunicación con el módulo del motor. Cabe recalcar que es la misma función y configuración empleada en la sección 2.3.3 Diseño del sistema de PID, con la diferencia que los valores serán constantes y no se verán reflejados en el panel del control que visualiza el operario a excepción de la selección de puerto CAN .

Figura 20

Interfaz de configuración de “Open Diagnostico CAN FD”



Nota. Elaboración propia.

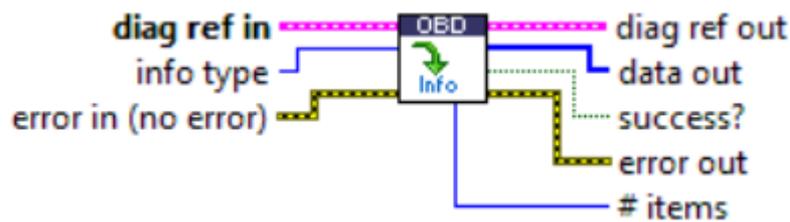
3.3.Extracción de información del vehículo

El número de identificación del vehículo, también conocido como VIN, o número de chasis del vehículo, es una información crucial para el diseño del panel de control, es un código alfanumérico de 17 caracteres asignados a cada vehículo al momento de su fabricación. Este código incluye información sobre el fabricante, el modelo, el tipo de motor y carrocería, y un número de serie único. La importancia del VIN radica en su capacidad para identificar de manera inequívoca a cada vehículo, permitiendo el acceso a su historial completo, incluyendo registros de accidentes, reparaciones, mantenimiento y cambios de propietarios. Además, el VIN ayuda a verificar la autenticidad del vehículo, asegurando que no ha sido clonado o alterado ilegalmente.

Para la obtención de esta información se usa la función “OBDD Request Vehicle Information.vi” que solicita información específica del sistema OBD del vehículo y devuelve los datos correspondientes, se observa en la Figura 21

Figura 21

Función “OBD Request Vehicle Information”



Nota. Elaboración propia.

Descripción de los terminales y su propósito:

- **diag ref in:** Este terminal recibe la referencia del diagnóstico que se está llevando a cabo.
- **error in (no error):** Este terminal de entrada de error se utiliza para manejar cualquier error que pueda haber ocurrido antes de esta función en el flujo de datos.
- **info type:** Aquí se detalla el tipo de datos que se desea obtener del sistema OBD.
- **diag ref out:** Este terminal proporciona la referencia del diagnóstico después de ejecutar la función, para ser utilizada en operaciones subsiguientes.
- **data out:** Este terminal de salida proporciona los datos solicitados del sistema OBD.
- **success?:** Este indicador booleano muestra si la operación fue exitosa.
- **error out:** Este terminal de salida de error maneja cualquier error que ocurra durante la ejecución de la función.
- **# items:** Este terminal proporciona el número de elementos obtenidos del sistema OBD.

Lo que se obtiene de la función dando como entrada un valor constante 2 en “info type” es un tipo de dato número entero (Integer Numeric) con una transformación a dato de caracteres (String) para visualizar el código único del fabricante para el vehículo y un indicador de respuesta del tipo booleano que representa el estado de extracción del número VIN, es decir; TRUE si se realizó con éxito, caso contrario FALSE.

3.4.Diseño de diagrama de bloques

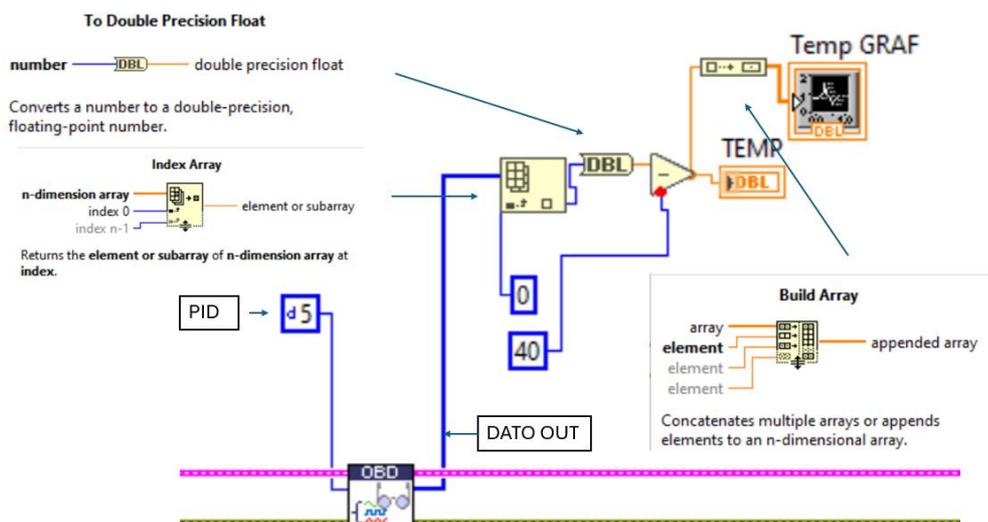
Se realiza la integración de los procesos antes descritos y se procede al tratamiento de datos mediante las ecuaciones correspondientes para cada sensor.

Después de la extracción de información del PID, la cual ya se realizó en la sección 2.3.2 Identificación de variables, se procede a el tratamiento de datos. Observe en la Figura 22 las funciones requeridas en el diagrama de bloques para separar la información en bites y usar cada uno en la ecuación. El tipo de dato que se busca tratar es un número entero (Integer Numeric) por lo que cada elemento del arreglo es un número sin decimales. Debido al tipo de línea que se observa se determina que es un arreglo unidimensional (1D Array) siendo una estructura de datos que contiene una secuencia ordenada de objetos similares, en este caso, números enteros. Un arreglo unidimensional tiene una sola dimensión, lo que significa que los elementos se almacenan en una lista lineal.

A continuación, se presentará un procedimiento estándar independientemente del tipo de PID. La primera función a emplear es un “Index Array” la cual extrae la información de cada posición que se solicite, empezado desde la posición 0. Una vez extraída la información se procede a transformar con un “DBL”, el cual transforma de un numero entero a un número de coma flotante de doble precisión. Para las ecuaciones se usa las operaciones matemáticas básicas de la sección “numeric”, con el resultado obtenido se usa un build array que permite crear un nuevo arreglo a partir de elementos individuales o de otros arreglos, se puede ingresar varios elementos de datos o varios arreglos y combinarlos en un solo arreglo, esta función servirá a futuro para mejoras en la visualización de gráficas.

Figura 22

Funciones para tratamiento de datos



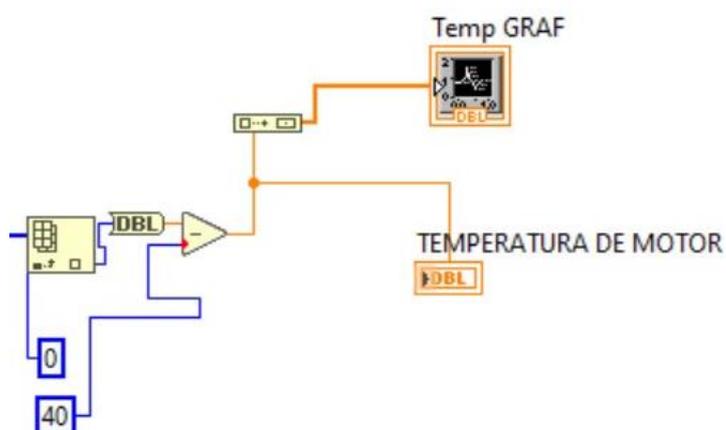
Nota. Elaboración propia.

3.4.1. Diseño diagrama de bloques sensor ECT

En la Figura 23 se observa la distribución de diagramas de bloque para la obtención del valor del sensor ECT, la ecuación que se empleó se encuentra en la Tabla 3 donde al valor A se le resta una constante de 40, y el resultado se lo representa de forma numérica y gráfica.

Figura 23

Diagrama de bloques sensor ECT



Nota. Elaboración propia.

Tabla 3

Información de PID para ECT

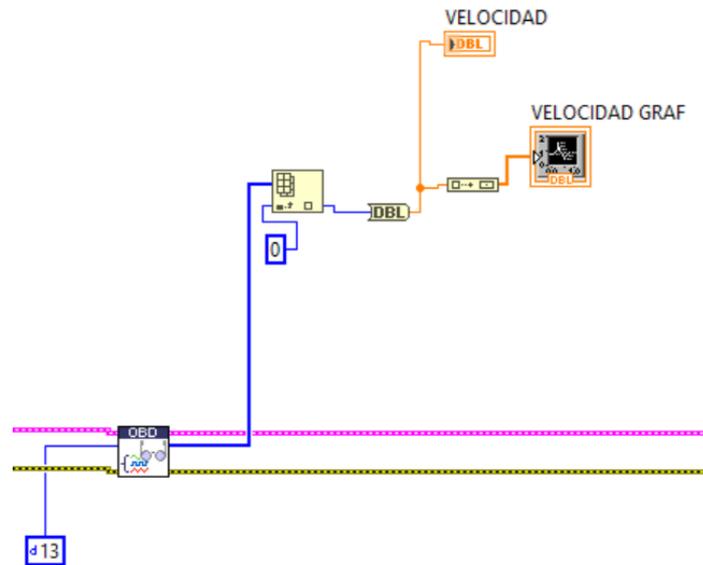
PID (Hex)	PID (Dec)	Descripción	Unidades	Ecuación
05	5	Temperatura del refrigerante del motor ECT	°C	$A - 40$

3.4.2. Diseño diagrama de bloques tacómetro

En la Figura 24 se observa la distribución de diagramas de bloque para la obtención del valor del sensor ubicado en caja para medir la velocidad (tacómetro), la ecuación que se empleó se encuentra en la Tabla 4 donde al valor A procede a mostrarse directo sin ninguna operación matemática, este valor se lo representa de forma numérica y gráfica.

Figura 24

Diagrama de bloques tacómetro



Nota. Elaboración propia.

Tabla 4

Información de PID para tacómetro

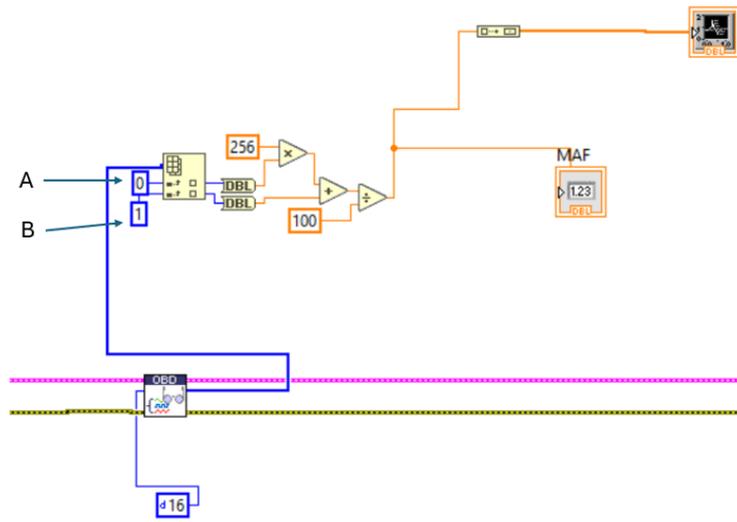
PID (Hex)	PID (Dec)	Descripción	Unidades	Ecuación
0D	13	Tacómetro	km/h	A

3.4.3. Diseño diagrama de bloques MAF

En la Figura 25 se observa la distribución de diagramas de bloque para la obtención del valor del sensor MAF, la ecuación que se empleó se encuentra en la Tabla 5 donde en la parte del numerador al valor A se le multiplica por una constante de 256 más el valor B y en el denominador una constante de 32, el resultado se lo representa de forma numérica y gráfica.

Figura 25

Diagrama de bloques MAF



Nota. Elaboración propia.

Tabla 5

Información de PID para MAF

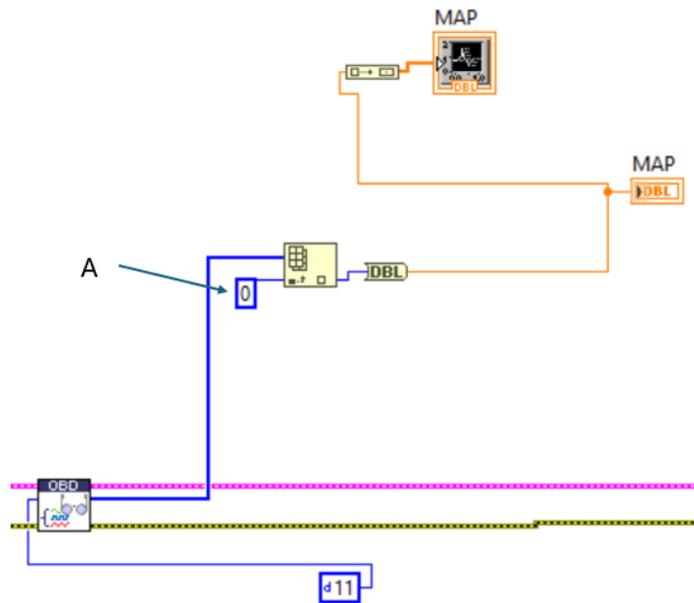
PID (Hex)	PID (Dec)	Descripción	Unidades	Ecuación
10	16	Sensor de flujo de masa de aire MAF	g/s	$\frac{256A + B}{32}$

3.4.4. Diseño diagrama de bloques MAP

En la Figura 26 se observa una distribución en diagramas de bloque para la obtención del valor del sensor MAP, la ecuación que se empleó se encuentra en la Tabla 6 donde al valor A no requiere de ninguna operación matemática, siendo un valor directo, y este valor se lo representa de forma numérica y gráfica.

Figura 26

Diagrama de bloques MAP



Nota. Elaboración propia.

Tabla 6

Información de PID para MAP

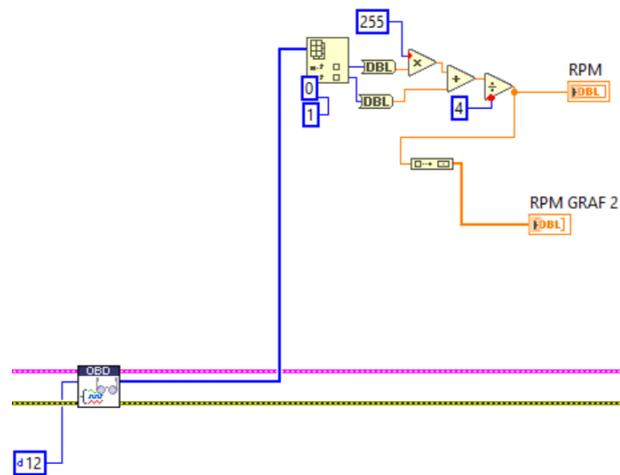
PID (Hex)	PID (Dec)	Descripción	Unidades	Ecuación
0B	11	Intake manifold absolute pressure MAP	kPa	A

3.4.5. Diseño diagrama de bloques RPM

En la Figura 27 se observa una distribución de diagramas de bloque para la obtención del valor del sensor de velocidad real de motor, la ecuación que se empleó se encuentra en la Tabla 7 donde en la parte del numerador al valor A se le multiplica por una constante de 256 más el valor B y en el denominador una constante de 4, el resultado se lo representa de forma numérica y gráfica.

Figura 27

Diagrama de bloques velocidad real de motor



Nota. Elaboración propia.

Tabla 7

Información de PID para RPM

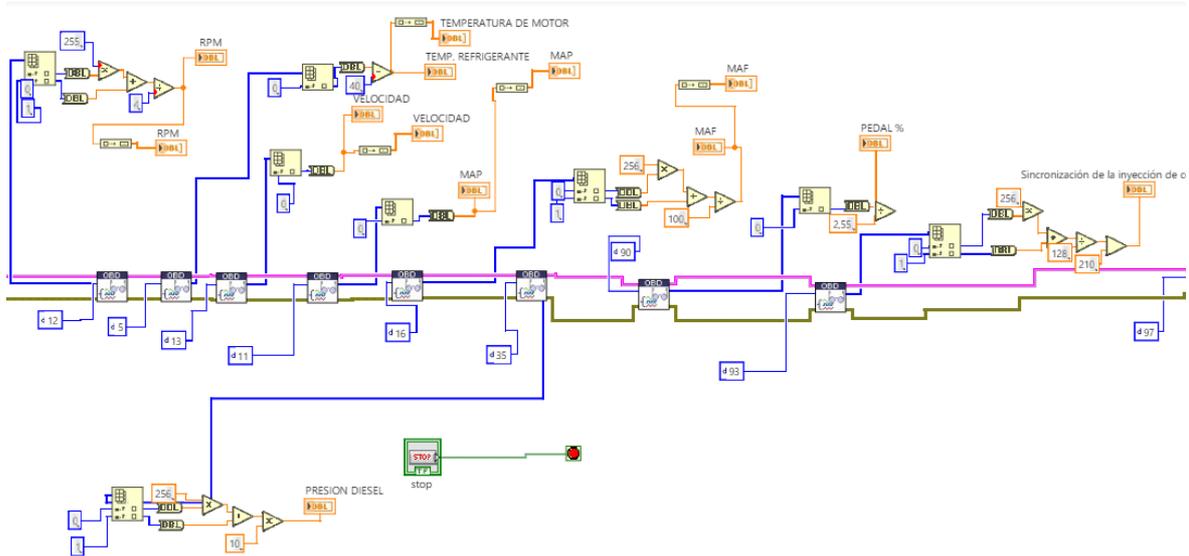
PID (Hex)	PID (Dec)	Descripción	Unidades	Ecuación
0C	12	Velocidad real de motor (RPM)	rpm	$\frac{256A + B}{4}$

3.4.6. Diseño diagrama general

Como se busca una herramienta integral el proyecto abarco la extracción de un total de 10 PID's como se muestra en la Figura 28, entre los cuales están MAP, MAF, RPM, ECT, velocidad del vehículo, demanda del torque por el conductor, % pedal de aceleración, bancos de temperatura de catalizador 1 y 2, sincronización de la inyección de combustible y presión de diésel en los apartados anteriores se muestra a detalle el proceso de 5 de ellos, para los demás no se explica pero se mantiene los mismo pasos para desarrollarlo.

Figura 28

Diagrama de bloques completo para visualizar datos de sensores referentes al motor del vehículo



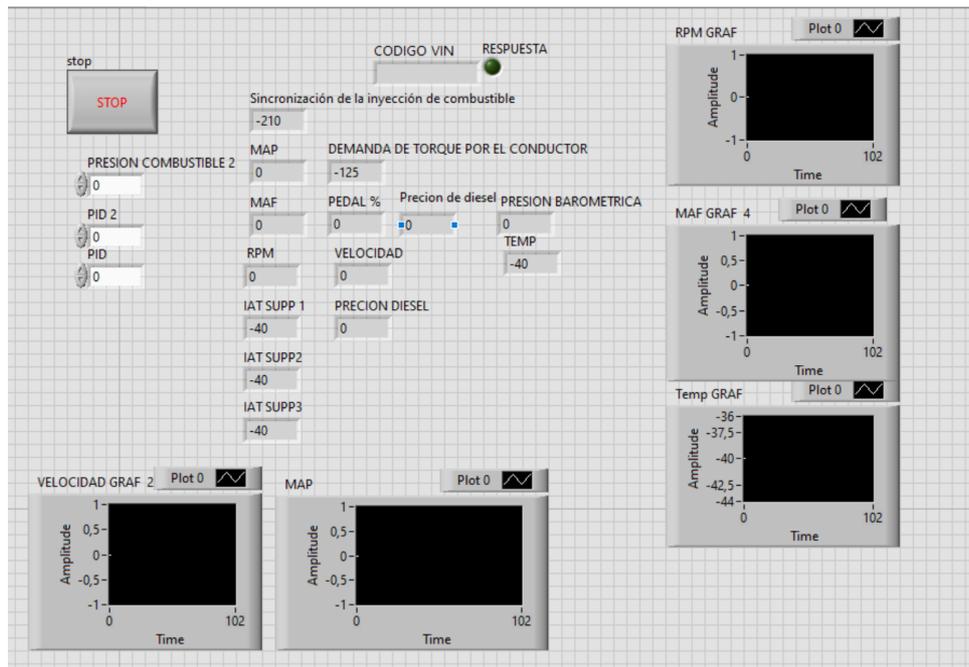
Nota. Elaboración propia.

3.5. Diseño interfaz de usuario

Para facilitar el monitoreo y análisis de los datos del vehículo, la interfaz LabVIEW incluyó elementos importantes como visualizaciones numéricas y gráficos. Asegurando que cada etapa del procesamiento de datos estuviera definida y optimizada para el rendimiento en tiempo real, se utilizaron diagramas de bloques para representar el flujo de datos y las operaciones realizadas sobre él. En la Figura 29 se observa un modelo de interfaz clara y directa que muestra los datos del vehículo sin necesidad de ningún procedimiento adicional por parte del usuario, ya que el objetivo de la interfaz es ser intuitiva.

Figura 29

Modelo de interfaz para visualización de datos PID

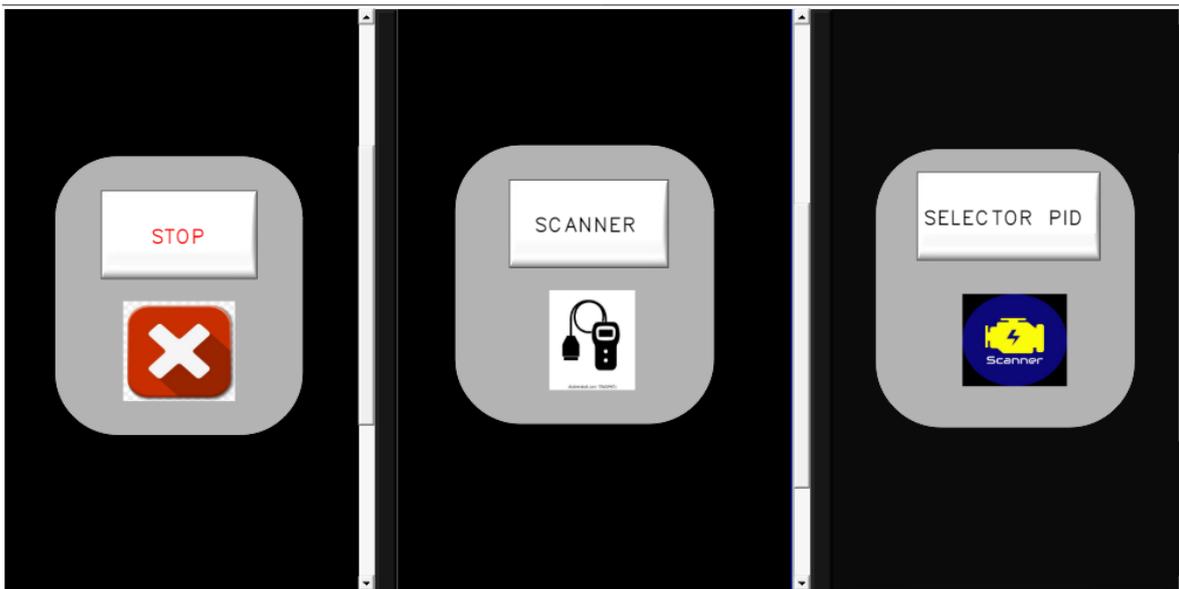


Nota. Elaboración propia.

En la búsqueda de facilitar la visualización de datos por separado, se creó dos apartados en la parte inicial de la interfaz, un apartado llamado “scanner” que dirija al usuario al panel general de todos los PID’s y otro llamado “selector PID” donde se puede seleccionar cada PID por separado, esto se lo puede observar en la Figura 30.

Figura 30

Diseño de la interfaz de usuario en LabVIEW



Nota. Elaboración propia.

Una vez ejecutado el programa el usuario podrá seleccionar 3 botones, el primero es un botón que termina el ciclo “while” y le permite detener el programa. Los otros dos son botones que redirigen a apartados, el botón denominado “scanner” abre una ventana donde se muestran todos los valores de sensores de forma numérica con sus respectivas unidades y la información de numero VIN. En este panel de control existe un selector para que el usuario escoja el puerto de comunicación, el cual varía según la tarjeta y la computadora que se vaya a usar. También se implementó la opción para guardar los valores registrados en tiempo real en una hoja de cálculo, la única operación que el usuario debe realizar es seleccionar la ubicación del archivo y aplastar el botón denominado “guardar datos” al momento que quiera registrar los datos, esto se lo puede observar en la Figura 31

Figura 31

Panel de control "Scanner"

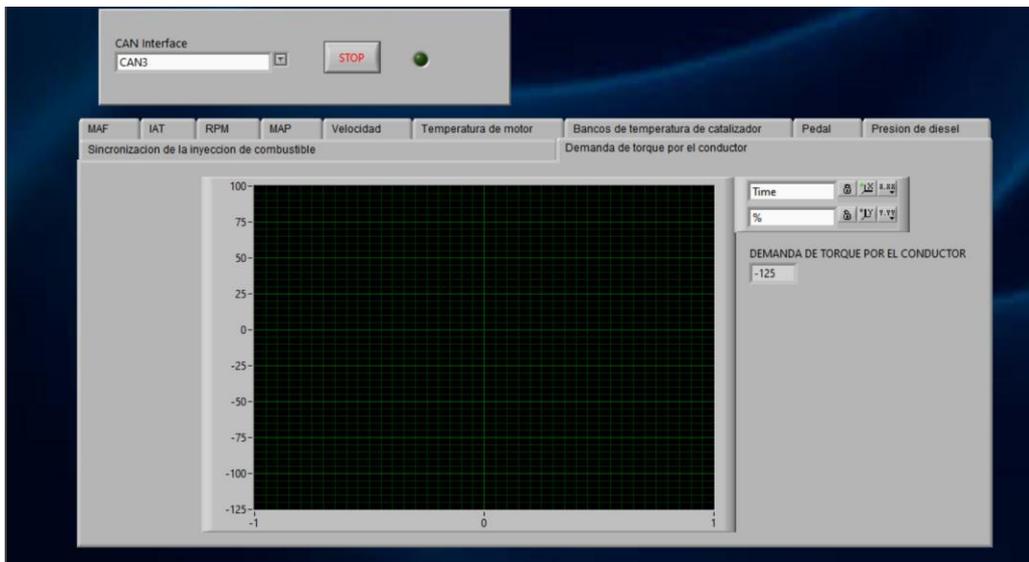


Nota. Elaboración propia.

El botón denominado "Selector PID" muestra los valores de cada sensor de forma individual con una gráfica en función del tiempo, la navegación es libre y se la realiza intercambiando entre ventanas como se observa en Figura 32

Figura 32

Panel de control "Selector PID"



Nota. Elaboración propia.

3.6. Validación de datos

En el contexto del desarrollo del panel de control en LabVIEW se procedió a realizar la comparación de los Identificadores de Parámetro (PID) obtenido con la tarjeta NI-9862 y el dispositivo de diagnóstico Autel MaxiIM, este una herramienta esencial para la validación de los datos recopilados. Se accedió al sistema del vehículo a través del escáner, donde para el modelo y marca específicos del vehículo, solo se pudo acceder mediante la normativa EURO 5, permitiendo una revisión exhaustiva de los valores.

Es importante destacar que, durante el proceso de identificación y entrada del scanner al vehículo de acuerdo con la normativa EURO 5, se encontraron diferencias en los valores obtenidos para los sensores de Temperatura de Aire de Admisión (IAT), Flujo de Masa de Aire (MAF) y Temperatura de Refrigerante (ECT). Estos valores fueron fuera del rango de funcionamiento y no coincidían con las condiciones reales del motor. En particular, los sensores ECT (Temperatura del Refrigerante del Motor) y IAT (Temperatura del Aire de Admisión) registraron valores superiores a 110 °C. Esto se debe a que el vehículo tiene un motor con tecnología EURO 3, y el acceso se estaba realizando a través de la normativa EURO 5.

En respuesta a esta situación, se decidió utilizar directamente el acceso OBD II para obtener y verificar la información de los sensores mencionados. Este enfoque garantiza la obtención de datos fiables y precisos, fundamentales para el análisis y diagnóstico correcto dentro de la interfaz desarrollada en LabVIEW. Este proceso no solo facilita la corrección de errores en la adquisición de datos, sino que también refuerza la precisión en el monitoreo del rendimiento del vehículo.

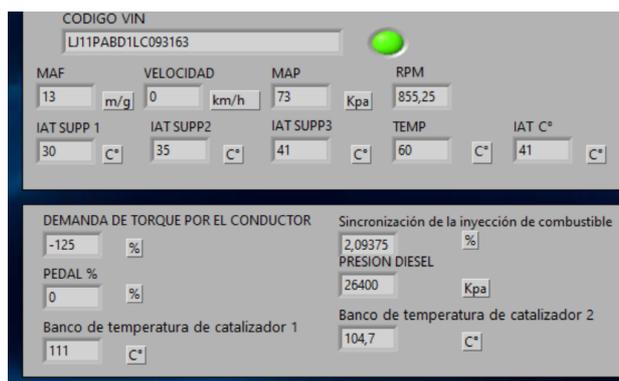
El Tabla 8, se muestra el proceso de validación de datos utilizando el escáner Autel MaxiIM y el acceso OBD-II. La validación directa permitió confirmar la precisión de los datos de los sensores, asegurando que la información utilizada para el diagnóstico y análisis fuera fiable y exacta.

Tabla 8

Proceso de validación de datos

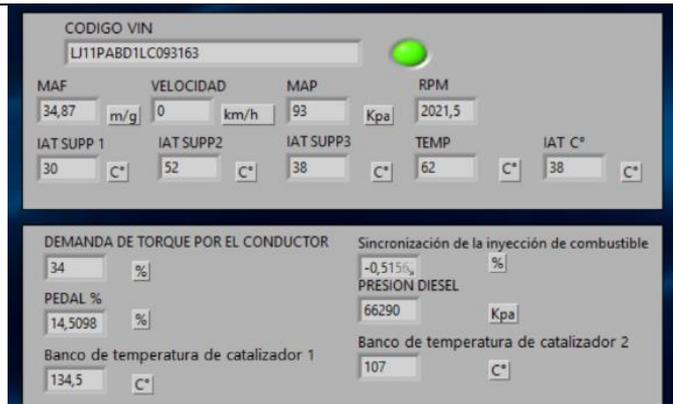
Análisis de valores respecto a la interfaz propuesta vs el scanner Autel MaxiIM

Dentro de la interfaz propuesta se visualizan datos de la camioneta JAC T8. Las ecuaciones obtenidas de diversas fuentes brindan información numérica de los sensores relevantes para la investigación.



En la interfaz, se puede ver el código VIN del vehículo y los sensores seleccionados, presentados de manera sencilla para el usuario. Por ejemplo, se muestran datos como las RPM del motor, que oscilan en 855.25 rpm, con una temperatura de refrigerante de 60°C y un valor del sensor de temperatura del aire de admisión (IAT) de 41°C. Es importante aclarar que los "supports" del sensor IAT son útiles para verificar su funcionamiento adecuado, asegurando que el sensor opere dentro de los parámetros esperados.

Los datos relacionados con la inyección del vehículo y los sensores se muestran en la parte inferior de la imagen. de temperatura del catalizador. Se entiende que el vehículo está en ralentí, con una demanda de torque por parte del conductor que muestra valores negativos, al igual que el pedal del vehículo. En cuanto a la sincronización de la inyección de combustible, se observa que el vehículo está adelantando la inyección en 2.09375°, con una presión del combustible en el riel de 26300 kPa. Los sensores de temperatura del catalizador en los bancos 1 y 2 son esenciales para detectar si la temperatura es demasiado alta o baja, lo que podría indicar problemas en el sistema de escape o en el propio catalizador.

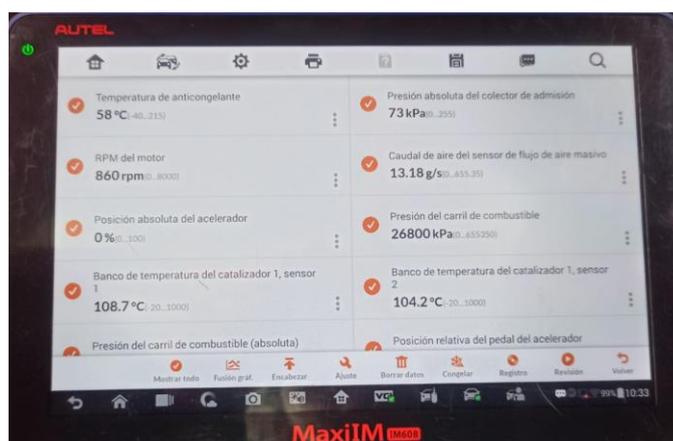


En la segunda imagen se observa un cambio en las RPM del vehículo, alcanzando 2021.5 revoluciones, lo que provoca una variación notable en los sensores. El sensor MAF muestra un valor de 34.87 g/s, mientras que la presión en el sensor MAP aumenta a 93 kPa. Este cambio se acompaña de un leve incremento en la temperatura del refrigerante, que alcanza los 62°C.

En la parte inferior de la imagen, se registra una demanda de torque del conductor del 34%, con una presión en el pedal del 14.51%. La sincronización de la inyección de combustible muestra un retraso de -0.5156°. Además, la presión en el riel de combustible aumenta aproximadamente a 66290 kPa.

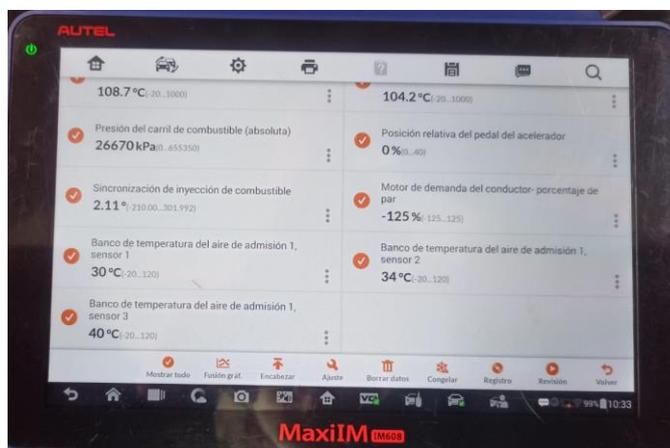
Los sensores de temperatura de los catalizadores indican un incremento, mostrando 134.5°C en el banco 1 y 107°C en el banco 2. Estos datos se muestran en la interfaz propuesta, que garantiza la verificación de la transmisión de datos, y muestran cómo varían las condiciones de operación del vehículo con el cambio en las revoluciones.

Para validar estos datos, utilizamos un escáner de la marca MaxiIM IM608.



De esta manera, podemos observar la temperatura antes de conectar con la NI9862. Se registran datos de 860 RPM, una presión en el sensor MAP de 73 kPa, y un flujo medido por el sensor MAF de 13.18 g/s, con una presión de 26800 kPa. Además, se visualizan

las temperaturas de los catalizadores en los bancos 1 y 2. Estos datos muestran una clara similitud con los valores del panel de control en LabVIEW, ya que la temperatura aumenta gradualmente a medida que el vehículo alcanza su temperatura óptima de funcionamiento.



De la misma manera, observamos que la sincronización de la inyección de combustible presenta un avance de 2.11° en ralentí. Además, se registra el torque producido por el conductor. Asimismo, La computadora realiza tres comprobaciones al sensor de temperatura del aire de admisión para garantizar que funcione correctamente.

Para asegurar la fiabilidad de los datos, se realizó un análisis detallado de las gráficas generadas durante la operación del vehículo. Estas representaciones visuales proporcionaron una visión profunda de cómo los sensores respondieron en tiempo real a diversas condiciones operativas. Por ejemplo, al variar las RPM del motor, se observó cómo los sensores MAP y MAF reaccionaron ante cambios en la presión y el flujo de aire dentro de la cámara de combustión. El procedimiento de validación de gráficas se lo detalla en la Tabla 9.

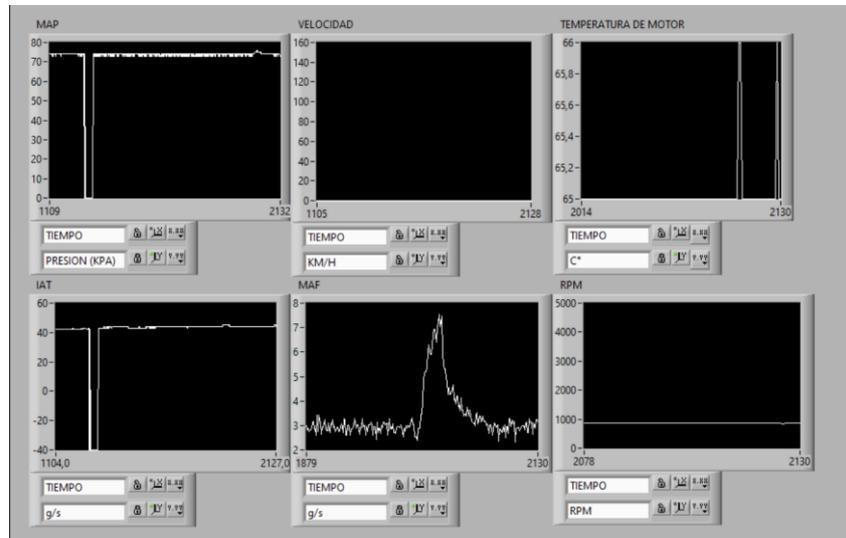
Tabla 9

Proceso de validación de gráficas

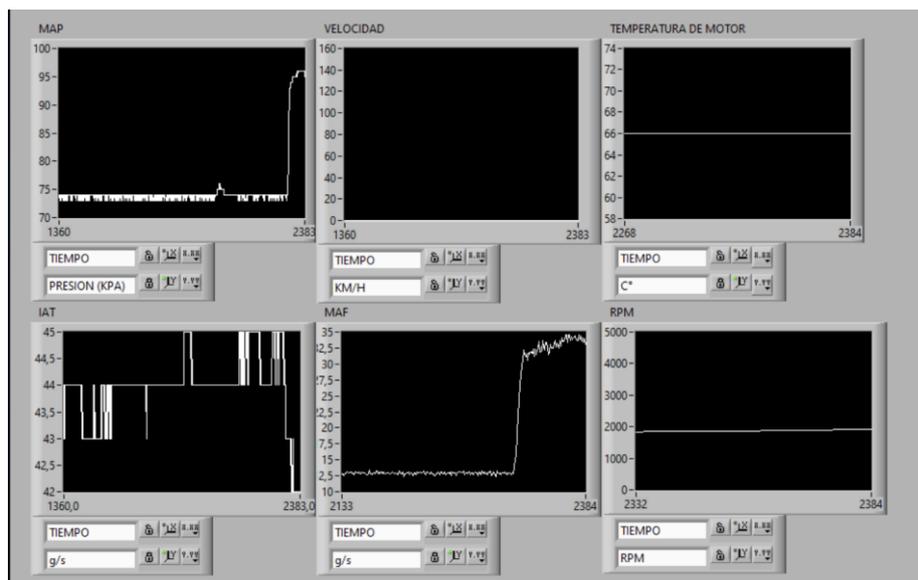
Verificación de gráficas

Se procede a encender el vehículo donde visualizamos el funcionamiento de los sensores en gráficas simples de interpretar, de esta manera validamos los datos de los sensores propuesta en la investigación. En la primera imagen se presentan las gráficas en condiciones de motor a ralentí como se observa en la gráfica de RPM a menos de 1000 rpm, existe una irregularidad en la gráfica del ECT debido a que a empezado a

calentarse y el cambio brusco representado por picos en la gráfica se debe los cambios en los valores resistivos del sensor, a medida que el motor se calienta la gráfica forma una línea con curvas suaves.



En la segunda imagen se varia aproximadamente a 2000 rpm donde es evidente la variación de los sensores MAP y MAF, debido a la aceleración el flujo de aire que atraviesa el depurador y la presión del múltiple de admisión varían., existiendo mayor flujo y mayor presión, por otro lado se visualiza que la gráfica IAT y ECT no existe cambio significativo ya que el parámetro que mide el IAT es la temperatura del aire que depende de las condiciones climatológicas y el ECT la temperatura de motor, el cual al llegar a su temperatura de trabajo se mantiene en el mismo rango.



Estos datos no solo confirmaron la coherencia de los valores obtenidos a través de la interfaz de LabVIEW, sino que también permitieron identificar patrones y tendencias importantes en el comportamiento del vehículo.

Las gráficas también muestran cómo cambiaron las temperaturas del aire de admisión (IAT) y del refrigerante en respuesta a los cambios en las condiciones de funcionamiento del motor. Este análisis detallado validó la precisión de los datos al mostrar cómo los sensores respondieron dinámicamente a las variaciones en la carga del motor y las demandas del conductor. Estos hallazgos son cruciales para asegurar que los datos utilizados en el diagnóstico y análisis del rendimiento del vehículo sean no solo precisos, sino también representativos de su funcionamiento real en condiciones variadas.

CAPÍTULO 4

4. ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

4.1.Efectividad del Protocolo CAN 2.0 y LabVIEW

El uso de LabVIEW junto con la tarjeta NI 9862 y el protocolo CAN 2.0 han sido muy efectivos para recoger datos y análisis en tiempo real del vehículo JAC T8. Esta combinación proporciona una plataforma robusta y confiable para la visualización y el tratamiento de grandes volúmenes de datos del vehículo, permitiendo una interacción precisa y eficiente con el sistema propuesto.

4.2.Importancia de la Validación de Datos

Implementar métodos para validar los datos, como el uso del scanner Autel MaxiIM y la verificación directa a través del sistema OBD2, ha sido crucial para garantizar la precisión de los datos recopilados. Esta estrategia ha permitido identificar y corregir discrepancias en los datos de sensores propuestos, asegurando que la información utilizada para diagnósticos y análisis sea fiable y exacta. De esta manera se verifico que al conectar la interfaz y el scanner automotriz con la línea CANbus al mismo tiempo afecta en la lectura de datos. Esta información podría resultar datos corruptos o incompletos

4.3.Mejora en la Diagnóstico y Monitoreo

La capacidad de extraer y decodificar eficientemente los Identificadores de Parámetro (PID) mediante fórmulas especializadas en LabVIEW ha mejorado significativamente el diagnóstico y monitoreo del vehículo. Esto facilita una comprensión más profunda del funcionamiento interno del vehículo, permitiendo intervenciones más rápidas y precisas en la resolución de problemas y el mantenimiento. Además de guardar los datos de los sensores en un archivo de Excel para analizar el campo de la ingeniería automotriz.

4.4.Capacidad de Adaptación y Futuras Mejoras

La infraestructura desarrollada proporciona una base sólida para futuras expansiones y desarrollos. La versatilidad del sistema permite adaptarse a nuevas necesidades y tecnologías emergentes en el ámbito automotriz, como la integración de más funciones de diagnóstico avanzado o la adaptación a diferentes modelos de vehículos, maximizando así su utilidad y relevancia a largo plazo. [OBJ]

5. CONCLUSIONES

- En el diseño de la interfaz garantiza al usuario un manejo sencillo e intuitivo para el manejo de datos. Mediante el uso de datos numéricos y gráficas para visualizar el comportamiento de los sensores en tiempo real cuando se varía el comportamiento del motor.
- La comunicación con el vehículo resulto efectiva usando comandos de la norma ISO 15765-4 para el acceso al módulo de motor, donde fue posible transmitir datos en tiempo real sin problemas de tráfico de datos. De esta manera se garantiza que los datos sean certeros y efectivos para un diagnóstico u otras actividades previstas para su uso.
- La metodología heurística para extraer PID es efectiva en este proceso de transmisión de datos, ya que comparar datos de la ECU del vehículo y el scanner Autel MaxiIM por medio de OBD-II garantizó de forma objetiva los datos de la interfaz hecha en LabVIEW. Cabe aclarar actualmente ciertos scanners con información de vehículos chinos como lo es la camioneta JAC T8 no son precisos. Ya que, durante la comparación de datos se presentó un problema de valoración de datos, donde el sensor IAT y MAF marcaron valores inusuales de los cuales un técnico automotriz asumiría como problemas del sensor. Esto se debe a que las normas internacionales no han sido actualizadas dentro de Ecuador, haciendo uso de normativas anteriores como lo es EURO 3 y los scanners comerciales se rigen bajo normativas actualizadas como EURO 5, por ende, los datos no muestran valores reales de funcionamiento. De esta manera se ha optado en mejorar y garantizar la información de vehículos de la marca JAC T8 para el diagnóstico automotriz.
- El desarrollo de la interfaz garantiza datos en tiempo real de los sensores MAP, MAF, RPM, Velocímetro y ECT mediante la librería Automotive Diagnostic Command Set, para manejar y tratar el flujo de datos. De esta manera se logró obtener valores numéricos de otros sensores como lo son la sincronización de la inyección de combustible, demanda de torque por el conductor, el porcentaje del pedal de aceleración, presión barométrica e identificador de código VIN del vehículo. Se logró recopilar los datos entregados por cada sensor en un archivo Excel para su respectivo análisis. La interfaz también garantiza un modo de selección de PID con su correspondiente gráfica de forma individual, permitiendo realizar un análisis detallado y específico de cada sensor.

- Se observó cierta demora al transmitir datos durante la visualización conjunta del panel de control desarrollado y el scanner, porque ambos solicitan información del vehículo, por lo que no se validaron datos con este método sino sometiendo de forma independiente de bajo las mismas condiciones de trabajo.
- Para concluir, el flujo de información expresada en PID por el vehículo e interpretada por LabVIEW ha demostrado que la funcionalidad de este programa es comparable a la de un escáner convencional disponible en el mercado. Se ha logrado visualizar datos específicos del módulo del motor, asegurando una interfaz simple e intuitiva para el usuario. Sin embargo, persisten desafíos relacionados con la comunicación con otros módulos del vehículo, lo que requiere un estudio más profundo para mejorar su integración y funcionalidad.

6. RECOMENDACIONES

- Para mejorar y expandir el número de sensores es necesario la implementación de módulos de actualización remota que permitan la actualización del software del panel de control sin necesidad de intervención manual para mejorar la funcionalidad y eficiencia del sistema de monitoreo. Esto facilitará la incorporación rápida y eficiente de nuevas funcionalidades como son el tipo de protocolo de cada vehículo actualmente.
- Crear una plataforma de datos abiertos para que los datos de diagnóstico y monitoreo se visualicen en otros computadores. Donde se visualicen datos que ayuden a la colaboración entre investigadores y profesionales del sector automotriz referente al CANBus.
- Antes de expandir el número de sensores, es crucial evaluar la capacidad actual del sistema, incluyendo la infraestructura de red, el hardware y el software. Planifica la exploración tomando en cuenta los efectos sobre el rendimiento del sistema y asegura que la infraestructura existente pueda manejar el incremento en el volumen de datos. Considera la posibilidad de actualizar los componentes del sistema, como procesadores más potentes en los dispositivos de recopilación de datos y mayor capacidad para almacenar y procesamiento en los servidores.
- Establecer planes de capacitación y soporte técnico continuo para que los usuarios del sistema puedan usar todas las funciones del panel de control. La capacitación debe incluir el uso del software LabVIEW, la configuración y el funcionamiento de la tarjeta NI 9862 y la interpretación de los datos recopilados.
- A medida que aumenta el número de sensores, también lo hace la complejidad del análisis de datos. Desarrolla o actualiza el software de análisis para que pueda procesar eficientemente grandes cantidades de datos en tiempo real. Considera la utilización de algoritmos de procesamiento paralelo y técnicas de aprendizaje automático para mejorar el procesamiento dentro del entorno de LabVIEW.

7. REFERENCIAS BIBLIOGRÁFICAS.

- Arévalo Calderón, F. E., & Ortega Ulloa, A. G. (2016). *Desarrollo de una interfaz para la visualización y adquisición de datos provenientes de la ECU a través de OBD II mediante un dispositivo de comunicación serial y del analizador de gases qrotech 6000*[Tesis de pregrado, Universidad Politécnica Salesiana]. Repositorio Institucional. <https://dspace.ups.edu.ec/handle/123456789/12029>
- Arévalo Calderón, F. E., & Ortega Ulloa, A. G. (2016). *Desarrollo de una interfaz para la visualización y adquisición de datos provenientes de la ECU a través de OBD-II mediante un dispositivo de comunicación serial y del analizador de gases Qrotech 6000*[Tesis de pregrado, Universidad Politécnica Salesiana]. Repositorio Institucional de la Universidad Politécnica Salesiana. <https://dspace.ups.edu.ec/handle/123456789/12029>
- Concepcion, M. (2011). *Estrategias de Sistemas OBD-2: (Incluyendo Inspección Estatal)*. CreateSpace Independent Publishing Platform. <https://books.google.com.ec/books?id=tsOZZwEACAAJ>
- Manzanares Caamal, A. F. (2020). *Análisis del estado actual de tecnología del Bus CAN (CONTROLLER AREA NETWORK)*[Tesis de pregrado, Universidad de Quintana Roo División de Ciencias e ingeniería]. Repositorio institucional UQROO. Retrieved 13 de Junio de 2024, from <http://risisbi.uqroo.mx/handle/20.500.12249/2614>
- National Instrument. (1 de Agosto de 2023). *NI-9862 Getting Started*. Retrieved 16 de Junio de 2024, from National Instrument: <https://www.ni.com/docs/en-US/bundle/ni-9862-getting-started/page/overview.html>
- National Instrument. (2024). *LabVIEW*. Retrieved 16 de Junio de 2024, from National Instrument: <https://www.ni.com/es-cr/shop/product/labview.html>
- National Instruments. (Diciembre de 2009). *Automotive Diagnostic Command Set User Manual*. Retrieved 12 de Junio de 2024, from National Instruments: <https://download.ni.com/support/manuals/372139d.pdf>
- Nyce, D. S. (2016). *Position Sensors : Theory and Application*. John Wiley & Sons, Incorporated. Retrieved 12 de Junio de 2024, from <https://bibliotecas.ups.edu.ec:2708/lib/upsal/detail.action?docID=4530806>
- Pinguil Peñaranda, J. L. (2023). *Implementación de un sistema de monitoreo de red can basado en identificadores para un vehículo eléctrico renault kangoo ze*[Tesis de

pregrado, Universidad Del Azuay]. Repositorio institucional. Retrieved 20 de Junio de 2024, from <http://dspace.uazuay.edu.ec/handle/datos/13303>

Rojas Hiedra, D. F. (2019). *Implementación de un modelo dinámico para una red de comunicación presente en un vehículo eléctrico Kia Soul mediante NI_XNET*[Tesis de maestría, Universidad del Azuay]. Repositorio institucional. <http://dspace.uazuay.edu.ec/handle/datos/9564>

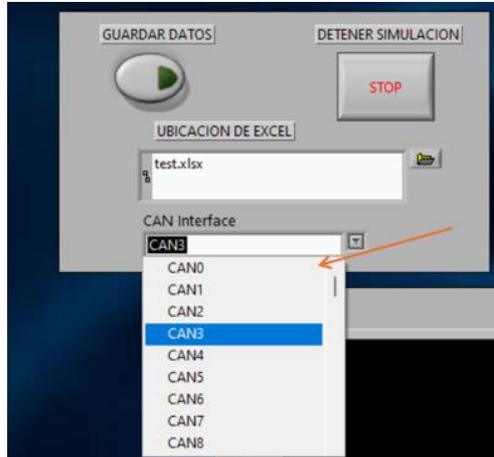
Sánchez Fernández, E. (2024). *Circuitos eléctricos auxiliares del vehículo* (Primera ed.). Macmillan Iberia, S.A. Retrieved 12 de Junio de 2024, from <https://bibliotecas.ups.edu.ec:3488/es/ereader/bibliotecaups/267618?page=1>

Cervantes-Veloz, A. A. (2023). *Cost-Effective USB-CAN Interface for Automotive Testing and Development Processes: Bridging the Gap with Software*.

8. ANEXOS.

Anexo 1

Selección de puerto CAN dentro del panel de control



Anexo 2

Archivo Excel generado en la extracción de datos.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Time	RPM	BANCO 1 CA'	BANCO 2 CA'	DEMANDA DIAT (C°)	MAF(g/s)	MAP(KPA)	PEDAL(%)	PRESION DE	SINCRONIZA	TEMPERATUF	VSS(KM/H)	
2	09:45:39,080	0	-40	-40	-125	-40	0	0	0	0	-210	-40	0
3	09:45:39,130	0	-40	-40	-125	-40	0	0	0	0	-210	-40	0
4	09:45:39,235	0	-40	-40	-125	-40	0	0	0	0	-210	-40	0
5	09:45:39,317	0	-40	-40	-125	-40	0	0	0	0	-210	-40	0
6	09:45:39,401	0	-40	-40	-125	-40	0	0	0	0	-210	-40	0
7	09:45:39,488	0	-40	-40	-125	-40	0	0	0	0	-210	-40	0
8	09:45:39,578	0	-40	-40	-125	-40	0	0	0	0	-210	-40	0
9	09:45:39,671	0	-40	-40	-125	-40	0	0	0	0	-210	-40	0
10	09:45:39,758	0	-40	-40	-125	-40	0	0	0	0	-210	-40	0
11	09:45:39,843	0	-40	-40	-125	-40	0	0	0	0	-210	-40	0
12	09:45:39,935	0	-40	-40	-125	-40	0	0	0	0	-210	-40	0
13	09:45:40,042	0	-40	-40	-125	-40	0	0	0	0	-210	-40	0
14	09:45:40,164	0	-40	-40	-125	-40	0	0	0	0	-210	-40	0
15	09:45:40,299	0	-40	-40	-125	-40	0	0	0	0	-210	-40	0
16	09:45:40,409	0	-40	-40	-125	-40	0	0	0	0	-210	-40	0
17	09:45:40,507	0	-40	-40	-125	-40	0	0	0	0	-210	-40	0
18	09:45:40,619	0	-40	-40	-125	-40	0	0	0	0	-210	-40	0
19	09:45:40,726	0	-40	-40	-125	-40	0	0	0	0	-210	-40	0
20	09:45:40,825	0	-40	-40	-125	-40	0	0	0	0	-210	-40	0
21	09:45:40,939	0	-40	-40	-125	-40	0	0	0	0	-210	-40	0
22	09:45:41,042	0	-40	-40	-125	-40	0	0	0	0	-210	-40	0