



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO
CARRERA DE INGENIERÍA AUTOMOTRIZ

DISEÑO DE UN DASHBOARD PARA EL BANCO DE SENSORES, ACTUADORES Y UNIDADES DE CONTROL, MODELO PMK – TL – LOO4 “HYUNDAI GRANDEUR” PARA MONITOREO Y CONTROL BASADO EN RED CAN BUS

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

AUTORES: JOSE LUIS BARONA ZALDUMBIDE
JORGE NICOLÁS TERÁN BURGOS

TUTOR: CARLOS ALBERTO CARRANCO QUIÑÓNEZ

Quito – Ecuador
2023

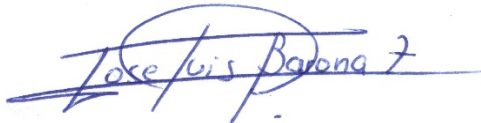
CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Jose Luis Barona Zaldumbide con documento de identificación N° 1718436403 y Jorge Nicolás Terán Burgos con documento de identificación N° 1754494373 manifestamos que:

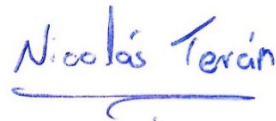
Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 11 de septiembre del año 2023

Atentamente,



Jose Luis Barona Zaldumbide
1718436403



Jorge Nicolás Terán Burgos
1754494373

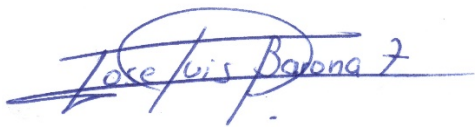
**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, Jose Luis Barona Zaldumbide con documento de identificación N° 1718436403 y Jorge Nicolás Terán Burgos con documento de identificación N° 1754494373, 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 Dashboard para el Banco de Sensores, Actuadores y Unidades de Control, modelo PMK – TL – LOO4 “Hyundai Grandeur” para monitoreo y control basado en Red Can Bus”, 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.

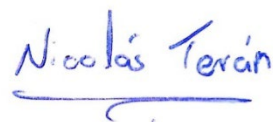
En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana

Quito, 11 de septiembre del año 2023

Atentamente,



Jose Luis Barona Zaldumbide
1718436403



Jorge Nicolás Terán Burgos
1754494373

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 DASHBOARD PARA EL BANCO DE SENSORES, ACTUADORES Y UNIDADES DE CONTROL, MODELO PMK – TL – LOO4 “HYUNDAI GRANDEUR” PARA MONITOREO Y CONTROL BASADO EN RED CAN BUS, realizado por Jose Luis Barona Zaldumbide con documento de identificación N° 1718436403 y por Jorge Nicolás Terán Burgos con documento de identificación N° 1754494373, 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, 11 de septiembre del año 2023

Atentamente,

A handwritten signature in blue ink, appearing to read 'Carlos Carranco', written over a horizontal line.

Ing. Carlos Alberto Carranco Quiñónez, MSc
1713629564

DEDICATORIA

La presente Investigación se la dedico con mucho amor, respeto y de manera muy especial a las personas que me apoyaron incondicionalmente frente a todos los obstáculos y buenos momentos que se presentaron en toda mi vida estudiantil dejando todo a un lado para verme sobresalir como profesional:

A mis Padres Rocío y Joselito por darme las fuerzas para seguir adelante, ser mi inspiración durante toda mi carrera además de brindarme todo el apoyo, confianza y cariño desde el instante que nací y a lo largo de toda mi vida, sin ellos nada de esto sería posible.

A mi hermano Diego por ser el ejemplo de profesional que deseo ser y motivarme siempre a cumplir con mis objetivos, además de velar por mí en todo momento.

A mi hermana Andrea por su cariño y apoyo constante demostrado hacia mi durante toda mi vida.

A mi sobrino Carlitos por toda la ternura y amor que he recibido de su parte y darme fe para persistir.

A toda mi familia en general, por el cariño brindado particularmente a mi Tía Paty, Tío Pablo y Tío Gonzalo los cuales me impulsaron a perseguir mis metas y me apoyaron siempre.

Y finalmente a todos mis familiares que partieron de este mundo y sé que me cuidan desde donde se encuentran, los llevo en mi mente y corazón, especialmente a mi primo Christian Gamboa.

Jose Luis Barona Zaldumbide

DEDICATORIA

Esta tesis se la dedico a mi familia.

A mi papá por enseñarme a nunca rendirme, a seguir adelante siempre con la cabeza en alto, por instruir en mí un pensamiento fuerte y real sobre el mundo en el que vivimos por mi mejor amigo y mejor ejemplo a seguir.

A mi madre quien siempre me alentó a seguir mis sueños y no soltarlos hasta alcanzarlos se la dedico por todas esas veces en las que supo guiar mi camino y apoyarme en cada decisión que he tomado.

A mis hermanos por ser un apoyo y mi más grande motivación a ser cada día una mejor persona y brindarles un buen ejemplo.

Por eso les doy mi trabajo en ofrenda por su cariño, amor y apoyo que me han brindado día a día y me han ayudado a ser quien soy.

Aún que les dedique este trabajo de titulación, siento que me faltaría la vida para agradecerles por todo lo que han hecho por mí, los amo mis Reyes.

Jorge Nicolás Terán Burgos

AGRADECIMIENTO

En el presente trabajo de investigación mis más sinceros agradecimientos:

A Dios, por darme la inteligencia, tranquilidad y fortaleza, no dejarme rendido en los malos momentos y adversidades que fueron superadas y guiarme por el camino del bien y justo.

A mis padres por ser el pilar de mi vida, estar para mí en los momentos que más necesito apoyo, brindarme su bendición y confiar plenamente en mí.

A el Ing. Carlos Alberto Carranco que fue mi docente, tutor de tesis y estuvo siempre al tanto del proyecto y sin su asistencia no sería posible llevar a cabo el presente proyecto, además de compartir su amistad y su don de gente.

A todos mis docentes en general, de la Universidad Politécnica Salesiana de la Carrera de Ingeniería Automotriz que me compartieron todo el conocimiento necesario otorgándome valiosas herramientas para mi profesión y el importante valor de la amistad.

A mi persona por persistir, nunca rendirme y dar frente a las adversidades, culminar la carrera de Ingeniería la cual siempre fue mi anhelo y encontré mi verdadera vocación, ser Ingeniero Automotriz.

Jose Luis Barona Zaldumbide

AGRADECIMIENTO

¡Gracias vida!

Por qué hoy miro atrás y tengo tanto que agradecer, gracias por los fracasos, gracias por las enseñanzas y vivencias, gracias por los días de dolor, los días de frustración y las pruebas en el camino.

Gracias familia por ese apoyo incondicional y por ese amor inmenso que día a día me brindan y llenan de luz mis días.

Gracias a mi tutor el Ingeniero Carlos Carranco por todo su apoyo en el ámbito laboral y por los consejos de vida que nos ha brindado él, por todas aquellas veces en que nos dijo "no todo es trabajo" se refería a que también debemos desarrollar la parte humana no solamente la parte teórica o técnica, ya que para ser un buen Ingeniero uno debe de ser primero un buen ser humano, agradezco.

Al director de carrera Ingeniero Angel Paucar por siempre estar pendiente de nuestras necesidades como estudiantes y preocuparse por el bienestar de nosotros, por todos aquellos docentes que en algún momento de mi vida me incentivaron y alentaron a ser un mejor profesional

Jorge Nicolás Terán Burgos

ÍNDICE DE CONTENIDOS

ÍNDICE DE CONTENIDOS.....	8
ÍNDICE DE FIGURAS.....	12
ÍNDICE DE TABLAS	0
RESUMEN	1
ABSTRACT.....	2
CAPÍTULO I	3
INTRODUCCIÓN.....	3
PROBLEMA	4
DELIMITACIÓN DEL PROBLEMA.....	5
Objetivo General.....	6
Objetivos Específicos.	6
CAPÍTULO II.....	7
Marco Teórico	7
Antecedentes.....	7
2.1 Redes Multiplexadas.....	7
2.1.1 Conceptos básicos de redes multiplexadas.....	7
2.1.2 Bit	7
2.1.3 Red.....	7
2.1.4 CAN.....	7
2.2 Can High y Can Low	8
2.3 Organización de la red multiplexada	9
2.3.1 Identificadores estándar.....	9
2.3.2 Identificadores extendidos.....	9
2.4 Red Can Bus	10
2.4.1 Breve historia del CAN y su aplicación relevante en los vehículos.....	10

2.5	Arquitectura del CAN BUS	11
2.6	Modos de medición OBDII.....	12
2.6.1	Modo 1.....	13
2.6.2	Modo 2.....	13
2.7	Mecanismo de control PID	14
2.8	Tipos de PID	14
2.8.1	PID genéricos	15
2.8.2	PID del fabricante.....	15
2.9	CAN BUS	16
2.9.1	Hexadecimal	16
2.10	Registro de datos	17
2.11	Datos del CAN BUS.	17
2.12	Trama de datos	18
2.13	Seguridad en CAN BUS.....	19
2.14	Introducción al OBD II	20
2.15	Función del OBD en los vehículos.....	21
2.16	Unidad de Control Electrónico.....	21
2.17	Protocolos de comunicación automotriz	23
2.17.1	SAEJ1939	23
2.17.2	ISO9141	24
2.17.3	Estructura maestra/esclavo	24
2.17.4	ISO 15765.....	25
CAPITULO III.....		26
HIPÓTESIS		26
3.1	Temperatura (Sensor ECT).....	27
3.1.1	Funcionamiento del Sensor ECT.....	29

3.1.2	Fórmula algebraica Sensor ECT.....	30
3.2	Revoluciones (Sensor CKP)	31
3.2.1	Fórmula algebraica Sensor CKP.....	34
3.3	Sensor MAP (Presión Absoluta del Múltiple)	35
3.3.1	Fórmula algebraica Sensor MAP.....	36
3.4	Sensor de Posición del Pedal de Acelerador (APP).....	36
3.4.1	Fórmula algebraica Sensor APP	38
3.5	Sensor de apertura del cuerpo de aceleración (TPS)	39
3.5.1	Fórmula algebraica del Sensor TPS.....	40
CAPÍTULO IV.....		41
4.1	PRESENTACION DEL PROYECTO.....	41
4.2	Presentación de componentes y configuraciones importantes.....	41
4.2.1	Arduino Uno	41
4.2.2	Configuración de variables.....	43
4.2.3	Puertos RXD y TXD.....	44
4.2.4	Puerto RXD	44
4.2.5	Puerto TXD.....	44
4.2.6	Módulo Bluetooth HC-05.....	44
4.2.7	Módulo CAN MCP 2515.....	45
4.3	Indicadores y medidores	46
4.3.1	Analógico.....	46
4.3.2	Digital	47
4.3.3	Indicadores Digitales	48
CAPÍTULO V		50
MÉTODO Y DESARROLLO DEL PROYECTO		50
	Enfoque Investigativo.....	50

Tipo de Investigación	50
Metodología analítica:	50
5.1 Presentación de la maqueta base del proyecto.....	51
5.2 Diseño del Hardware y Software para la comunicación.....	52
5.2.1 MCP2515 y Arduino Uno.....	52
5.2.2 Código para la comunicación CAN Bus con Arduino	54
5.2.3 Detalle del código 1.....	55
5.2.4 Visualización de datos entre Arduino y Computador.....	56
5.2.5 Comprobación base de datos obtenidos.....	57
5.2.6 Comprobación datos obtenidos a tiempo real comparado mediante Scanner Automotriz.....	59
5.2.7 Implementación del Módulo Bluetooth HC - 05	60
5.2.8 Filtración de datos	62
5.3 Diseño Hardware Final	64
5.4 Creación de la Interfaz de la aplicación para el Dashboard.....	64
CAPÍTULO VI.....	67
Análisis e interpretación de resultados	67
6.1 Temperatura (Sensor ECT) PID (A0).....	68
6.2 Revoluciones por minuto (Sensor CKP) PID (A0).....	69
6.3 Presión absoluta del múltiple (MAP) PID (A1).....	71
6.4 Sensor de Posición del Pedal de Acelerador (APP) PID (329).....	73
6.5 Sensor de apertura del cuerpo de aceleración (TPS) PID (329)	75
CONCLUSIONES	77
RECOMENDACIONES	78
BIBLIOGRAFÍA	79
ANEXOS	1

ÍNDICE DE FIGURAS

Figura 1 Niveles de alta velocidad CAN.....	8
Figura 2 Niveles de baja velocidad CAN.....	9
Figura 3 Diagrama básico de red multiplexada.....	10
Figura 4 Modo 1 a detalle.....	13
Figura 5 PID del fabricante	15
Figura 6 Diferentes módulos del CAN Bus.....	16
Figura 7 Trama Can en formato base con extensiones eléctricas del bus y sin bits de relleno	18
Figura 8 Diagrama comparativo entre SAEJ1939 y CAN	23
Figura 9 Conexión ISO 9141.....	24
Figura 10 Conexión ISO 15765.....	25
Figura 11 Registro datos OBD2	26
Figura 12 Distribución de PID's y Bytes en columnas.....	27
Figura 13 Sensor ECT del Banco de Sensores, Actuadores y Unidades de Control.....	27
Figura 14 Sensor NTC.....	29
Figura 15 Resistencia de las sondas de temperatura	29
Figura 16 Ejemplo forma de Onda del Sensor de Temperatura del Refrigerante del Motor	30
Figura 17 Sensor CKP obtenido del Banco de Sensores, Actuadores y Unidades de Control	31
Figura 18 Funcionamiento sensor CKP de efecto Hall	32
Figura 19 Funcionamiento sensor CKP inductivo.....	32
Figura 20 Descripción de código de avería	33
Figura 21 Sensor MAP del Banco de Sensores, Actuadores y Unidades de Control.....	35
Figura 22 Sensor APP del Banco de Sensores, Actuadores y Unidades de Control.....	36
Figura 23 Circuito del Sensor de Posición del Acelerador.....	37
Figura 24 Funcionamiento Sensor APP con representación de señales de salidas de voltaje	37
Figura 25 Sensor TPS del Banco de Sensores, Actuadores y Unidades de Control	39
Figura 26 Sensor TPS en fallo.....	40

Figura 27 Microcontrolador Arduino Uno - Vista Superior.....	41
Figura 28 Microcontrolador Arduino Uno - Vista Inferior	42
Figura 29 Microchip ATmega 328p.....	42
Figura 30 Memoria del microcontrolador Arduino Uno	43
Figura 31 Función void setup de Arduino.....	43
Figura 32 Función void loop de Arduino	44
Figura 33 Módulo Bluetooth (Hc - 05) Vista Superior	45
Figura 34 Módulo Bluetooth (Hc - 05) Vista Inferior.....	45
Figura 35 Módulo CAN MCP 2515 (Vista Superior).....	46
Figura 36 Indicadores Analógicos presentes en el tablero del Banco de Sensores, Actuadores y Unidades de Control.....	47
Figura 37 Indicadores digitales presentes en el tablero del Banco de Sensores, Actuadores y Unidades de Control	49
Figura 38 Banco De Sensores, Actuadores Y Unidades De Control, Modelo PMK – TL – LOO4 “Hyundai Grandeur”.....	52
Figura 39 Módulo CAN MCP2515 y Arduino Uno.....	53
Figura 40 Extensión del código total 1.....	54
Figura 41 Monitor Serial de la aplicación Arduino posterior a ejecutar el código 1	56
Figura 42 Transceptor individual LIN y CAN parte del frente y parte posterior.....	58
Figura 43 Aplicación CAN-Bus Monitor en desarrollo	59
Figura 44 Cable divisor OBD II	60
Figura 45 Comprobación de la variable mediante Software Can Bus Monitor y CSSElectronics.	60
Figura 46 Conexión Arduino Uno, Módulo CAN MCP2515 y Módulo HC – 05	61
Figura 47 Verificación del emparejamiento módulo HC05 – Smartphone.....	62
Figura 48 Dato filtrado PID A0.....	63
Figura 49 Dato filtrado PID A1.....	63
Figura 50 Dato filtrado PID 329.....	63
Figura 51 Diseño final del dispositivo.....	64
Figura 52 Fragmento de la programación de la aplicación	65
Figura 53 Visualización de datos recibidos en tiempo real por Red CAN BUS del Banco de Sensores.....	66

Figura 54 Análisis PID A0 (Temperatura ECT) desde App CAN-Bus Monitor a tiempo real	68
Figura 55 Análisis PID A0 (Temperatura ECT) desde Scanner Automotriz a tiempo real.	68
Figura 56 Reflejo de la temperatura del ECT a una temperatura exacta de 78 °C en la aplicación.....	69
Figura 57 Análisis PID A0 (Temperatura -ECT) desde App CAN-Bus Monitor a tiempo real	69
Figura 58 Análisis PID A0 (Revoluciones por minuto - CKP) desde Scanner Automotriz a tiempo real	70
Figura 59 Reflejo de la temperatura de las revoluciones exactas de 1260 rpm en la aplicación	71
Figura 60 Análisis PID A1 (Presión [kPa] - MAP) desde App CAN-Bus Monitor a tiempo real	71
Figura 61 Análisis PID A1 (Presión absoluta del múltiple - MAP) desde Scanner Automotriz a tiempo real.....	72
Figura 62 Reflejo de la presión absoluta del múltiple exacta de 30 kPa en la aplicación..	73
Figura 63 Análisis PID 329 (Posición del pedal de aceleración [%] - APP) desde App CAN-Bus Monitor a tiempo real	73
Figura 64 Análisis PID 329 (Posición del pedal de aceleración [%] - APP) desde Scanner Automotriz a tiempo real.....	74
Figura 65 Reflejo de la posición del pedal de aceleración exacto de 99,61% en la aplicación	75
Figura 66 Análisis PID 329 (Sensor de apertura del cuerpo de aceleración [%] - TPS) desde App CAN-Bus Monitor a tiempo real	75
Figura 67 Reflejo de la posición del pedal de aceleración exacto de 47,84% en la aplicación	76

ÍNDICE DE TABLAS

Tabla 1	Orden de la identificación de datos	11
Tabla 2	Segmento de bytes.....	12
Tabla 3	Modos de medición OBDII	12
Tabla 4	PID Genérico ejemplo	15
Tabla 5	PID Obtenido del Sensor ECT y fórmula algebraica	30
Tabla 6	Componentes del código de avería.....	33
Tabla 7	PID Obtenido del Sensor CKP y fórmula algebraica	34
Tabla 8	PID Obtenido del Sensor MAP y fórmula algebraica	36
Tabla 9	PID Obtenido del Sensor APP y fórmula algebraica.....	38
Tabla 10	PID Obtenido del Sensor TPS y fórmula algebraica	40
Tabla 11	Tabla de costos generales del proyecto	49
Tabla 12	Tabla importada en formato CSV de un tramo de la base de datos mediante el complemento de Transmisor de Datos para Excel	57

RESUMEN

Actualmente los componentes electrónicos de asistencia para el vehículo se han desarrollado a medida del paso del tiempo, es debido a que la importancia de los sensores, actuadores y unidades de control radican en la obtención y fiabilidad de datos en vivo precisos con la finalidad de poder gestionar de una manera más eficiente las funciones, el diagnóstico y control de los vehículos mediante la gestión electrónica.

El presente proyecto técnico detalla el análisis y visualización mediante un Dashboard sobre la comunicación con el Protocolo OBD – II para Red CAN BUS realizando un énfasis detallado en la obtención de variables como (Presión, Temperatura, Posición de acelerador, RPM) obtenidas desde la Unidad de Control de Motor (ECU) como códigos hexadecimales de parámetros de identificación (PID's) vinculados a mediciones precisas de los modos 1 y 2 las variables se presentarán de manera clara y entendible haciendo uso de gráficos y valores numéricos según el aplicativo por consiguiente se apreciará las señales de activación propias de cada variable (sensor o actuador) que intervienen en el momento de actuación del componente electrónico.

Mediante la examinación y el análisis de la gestión de datos transferidos desde la Red Can Bus del Banco de sensores, actuadores y unidades de control hacia el prototipo electrónico elaborado mediante una placa de microcontrolador (Arduino Uno) un módulo CAN (MCP2515) y un módulo bluetooth (HC-05) que contribuirá en el enlazamiento entre los datos de la red CAN y la salida de datos numéricos o gráficos que se monitorearán en el Dashboard, obteniendo finalmente una identificación clara de las variables así como una detección del estado óptimo de los componentes, posibles irregularidades o estado del vehículo en general de acuerdo a los datos previamente analizados.

Palabras Claves: Dashboard, Can Bus, OBD II, Arduino Uno, MCP2515, HC -05, Arduino

ABSTRACT

Currently the electronic components of assistance for the vehicle have been developed as time goes by, it is because the importance of sensors, actuators and control units lies in the obtaining and reliability of accurate live data in order to be able to manage in a more efficient way the functions, diagnosis and control of vehicles through electronic management.

The present technical project details the analysis and visualization by means of a Dashboard on the communication with the OBD - II Protocol for CAN BUS Network making a detailed emphasis on the obtaining of variables such as (Pressure, Temperature, Throttle position, RPM) obtained from the Engine Control Unit (ECU) as hexadecimal codes of identification parameters (PID's) linked to precise measurements of modes 1 and 2. The variables will be presented in a clear and understandable way making use of graphics and numerical values according to the application, therefore the activation signals of each variable (sensor or actuator) that intervene in the moment of action of the electronic component will be appreciated.

By examining and analyzing the data management transferred from the Can Bus Network of the Bank of sensors, actuators and control units to the electronic prototype developed by means of a microcontroller board (Arduino Uno), a CAN module (MCP2515) and a Bluetooth module (HC-05) that will contribute to the link between the CAN network data and the numerical or graphic data output to be monitored on the Dashboard, finally obtaining a clear identification of the variables as well as a detection of the optimal state of the components, possible irregularities or state of the vehicle in general according to the previously analyzed data.

Keywords: Dashboard, Can Bus, OBD II, Arduino Uno, MCP2515, HC -05, Arduino

CAPÍTULO I

INTRODUCCIÓN

Actualmente, existen varias empresas en el campo automotriz dedicadas a la preservación y cuidado del medio ambiente, esto con el objetivo de generar una movilidad sostenible con el menor impacto ambiental posible, de aquí nace la necesidad de crear sistemas más eficientes, este avance tecnológico genera una mayor competencia dentro del mundo automotriz por diseñar vehículos con mejores prestaciones y con menos emisiones contaminantes.

Como resultado del continuo desarrollo tecnológico, con relación a los vehículos utilizados en décadas pasadas los cuales eran íntegramente mecánicos y con el transcurso del tiempo fueron desarrollando e implementando asistencias electrónicas controladas mediante sistemas y módulos electrónicos que asisten tanto al conductor como a el vehículo, además de informar el estado, transmisión de datos de variables en tiempo real, y detección de averías mediante códigos de falla. Para conseguir el enlazamiento de los distintos sistemas con los componentes electrónicos, es clave la intervención de la comunicación por red Can Bus. La red de comunicación (CAN BUS) fue creada por la empresa Bosch en la década de 1980, esta red genera una mejor comunicación entre los módulos electrónicos del vehículo, lo cual se ve representado en una mejor gestión de componentes como motor, transmisión, sistemas de suspensión, sistemas de gestión de gases, sistemas de seguridad activa y pasiva, etc. (Zambrano et al., 2015)

La obtención de valores cada vez más precisos dentro de los sistemas automotrices para la reparación de vehículos se ha vuelto indispensable para un correcto diagnóstico y control del estado del mismo, gracias a esta necesidad se han desarrollado diferentes controles electrónicos más precisos para aquella persona que trata de realizar reparaciones dentro de un vehículo, como los odómetros digitales en nuevos modelos de vehículos. Dentro de nuestro medio automotriz existe una cantidad reducida de vehículos que cuentan con este sistema, sin embargo, los vehículos que disponen de una red de comunicación CAN BUS pueden implementar dicho sistema a su funcionamiento.

Los odómetros automotrices digitales son implementados por los fabricantes de vehículos de manera que estos sistemas son utilizados para presentar una visualización de datos más precisos, creando una perspectiva más elevada sobre el vehículo y a su vez una interfaz más amigable con el conductor.

PROBLEMA

La implementación de nuevas tecnologías dentro de la industria automotriz se encuentra cada día en crecimiento, es por eso que se deben buscar soluciones a problemáticas de obsolescencia de sistemas automotrices como la conversión de sistemas analógicos a digitales, esto con el objetivo de obtener datos más precisos y ayudar con el diseño de prototipos más eficientes.

El uso de artefactos digitales se ha visto implementados en los vehículos dentro de los últimos años y es un avance sumamente importante, ya que presenta menores limitaciones que artefactos analógicos, en este estudio se trata de reducir ciertas problemáticas como:

Brinda una mayor seguridad en los artefactos digitales pues la encriptación de la señal es posible de esta manera la decodificación, transferencia y almacenamiento de información es representada en señales de entrada y salida.

Debido a que las señales de los componentes digitales solo implican dos niveles, se contaminan de menor ruido, su reproducción es más sencilla, y el traspaso de información es instantánea respecto a las señales de los componentes analógicos.

Los dispositivos actuales centrados en sistemas digitales se orientan a un diseño más ergonómico y funcional.

Identificar los protocolos CAN Bus requeridos, ya que la mayor parte del parque automotor presentan dichos protocolos de comunicación y resulta imprescindible su conocimiento para el aplicativo en el presente proyecto y en el área automotriz, por lo que será imprescindible analizar rigurosamente la solicitud y acceso de datos de la ECU del automóvil mediante códigos PID (Parameter Identification) relacionados a mediciones específicas de los modos 1 y 2 del sistema OBD II.

Implementar una herramienta de estudio apta para el diagnóstico automotriz y monitoreo de datos de bajo costo y alta efectividad.

Conocer el correcto funcionamiento de una red multiplexada y de los caracteres que esta lleva en su interior, además de conocer el reconocimiento de la red dentro de cada módulo y su interpretación, poder manipular los datos a conveniencia del usuario y operador es algo indispensable para el mundo automotor.

Simplificar los sistemas de comunicación automotrices ha sido uno de los mayores retos dentro del mundo automotriz, ya que a medida que los sistemas de comunicación sean mayores, se pueden mejorar los sistemas electrónicos, como la seguridad, consumo de combustible, aumento de potencia, etc.

Identificar datos en vivo del vehículo de manera precisa es un dato que el usuario debería conocer para poder tomar acciones correspondientes acerca del control del automóvil.

DELIMITACIÓN DEL PROBLEMA

A partir de la década de los 80 la contaminación ambiental se ha vuelto una problemática muy grande dentro del mundo convencional, debido a la contaminación excesiva de los vehículos se implementaron sistemas de control mejorados, los fabricantes al darse cuenta de que la electrónica es indispensable para un control óptimo dentro del vehículo deciden implementar sistemas electrónicos.

Los nuevos sistemas electrónicos a partir de esa fecha han sido mejorados con el pasar de los años, esto implica que cada vez se tiene un mejor control sobre el vehículo obteniendo una mejor eficiencia, seguridad y reducir emisiones contaminantes, es por eso que si se realizan comparativas entre vehículos antiguos en comparación con vehículos modernos es indiscutible las grandes diferencias que ambos presentan.

De toda esta problemática han surgido sistemas de comunicación más eficientes como el ISO 15765 o la red CAN BUS.

En Ecuador hasta la fecha no existe una normativa que regularice los sistemas de comunicación dentro de los vehículos, es por ese motivo que se siguen vendiendo dentro del país vehículos sin este sistema, lo cual hace que estos vehículos sean más propensos a un mayor consumo de combustible, menor eficiencia del motor y a una menor seguridad activa y pasiva.

Los vehículos en la actualidad en el Ecuador cuentan mayormente con sistema de comunicación CAN BUS, esto debido a las exigencias del consumidor y la importancia de poder observar estos datos en vivo de una manera más suave para el usuario, mecánico o controlador se ha vuelto una necesidad en el mundo moderno.

Objetivo General

Diseñar un Dashboard eficiente y de manipulación sencilla con asistencia de varios módulos de programación electrónicos, logrando visualizar y controlar datos en vivo del comportamiento del vehículo mediante la Red Can Bus con protocolo OBD II.

Objetivos Específicos.

- Lograr la obtención de comunicación hacia la red Can Bus mediante un módulo electrónico que permita acceder a los datos, con el objetivo de identificar y adecuar ciertos parámetros del balance de pruebas.
- Registrar la transferencia de datos de las Unidades de Control Electrónico e identificar el modo en que trabajan electrónicamente ciertos componentes eléctricos o electrónicos al momento de entrar en funcionamiento.
- Desarrollar un sistema el cual permita visualizar de forma gráfica o numérica los datos a obtener del vehículo en tiempo real.

CAPÍTULO II

Marco Teórico

En el capítulo II referente al marco teórico se establecerán las bases teóricas y conceptuales que serán fundamentales para la comprensión de los capítulos posteriores se abordarán conceptos que relacionen a la red Can Bus, así mismo como la revisión de investigaciones y proyectos previos relacionados los cuales hayan examinado temas similares al del presente proyecto de titulación. De esta forma se proporcionará una base esencial con respecto al “Diseño de un Dashboard para el Banco de Sensores, Actuadores y Unidades de Control, modelo PMK – TL – Loo4 “Hyundai Grandeur” para monitoreo y control basado en Red Can Bus”.

Antecedentes

2.1 Redes Multiplexadas

2.1.1 Conceptos básicos de redes multiplexadas

2.1.2 Bit

Un bit es la unidad básica de información en un sistema informático. Es la abreviatura de "binary digits" (dígitos binarios en español) y puede tener dos valores posibles: 0 o 1. Los bits se utilizan para representar y almacenar información digital como números, texto, imágenes, sonidos y cualquier tipo de datos que se puedan representar en forma binaria.

2.1.3 Red

Una red es un grupo de dispositivos interconectados que pueden comunicarse entre sí para compartir recursos y transferir información.

2.1.4 CAN

es un estándar de comunicación en serie desarrollado originalmente por la empresa alemana Robert Bosch GmbH en la década de 1980. Está especialmente diseñado para una comunicación fiable y robusta entre unidades de control electrónico distribuidas (ECU) en vehículos o entornos industriales (Mallamaci et al., 2021).

Las redes multiplexadas automotrices cumplen con varios propósitos y beneficios en el desarrollo automotriz, ya sea la eficiencia en el cableado, comunicación entre distintos módulos electrónicos, mejora en el diagnóstico y monitoreo, ayuda a realizar actualizaciones de software de manera y corrobora integración de sistemas de manera más eficiente.

La organización de la red múltiplex de automóviles puede variar según los protocolos utilizados y el diseño específico del sistema del vehículo, aunque existen varios diseños de redes multiplexadas como SAEJ1939, ISO9141, ISO15765, etc. Sin embargo, lo común en los vehículos modernos es una arquitectura de red multiplexada conocida como Controller Area Network (CAN), que se usa ampliamente en la industria automotriz.

La red multiplexada (CAN) basa su diseño en una organización específica.

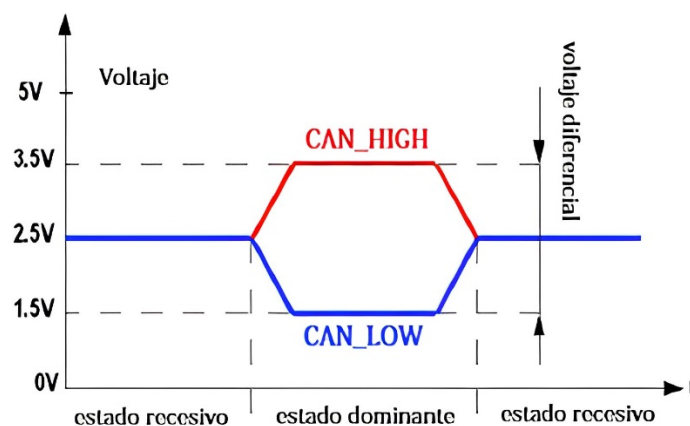
2.2 Can High y Can Low

La línea CAN High y la línea CAN Low se refieren a las dos líneas del bus CAN utilizadas para la comunicación entre dispositivos electrónicos en un sistema de red CAN. El bus CAN es un estándar de comunicación utilizado principalmente en la industria automotriz, pero también se usa en otras industrias.

CAN High y CAN Low son las dos líneas de señal que forman el bus de datos. CAN High es el cable de señal de alto voltaje y CAN Low es el cable de señal de bajo voltaje. Juntos, estos cables forman líneas diferenciales, lo que significa que transmiten señales relacionadas con la diferencia de voltaje entre ellos.

Figura 1

Niveles de alta *velocidad* CAN

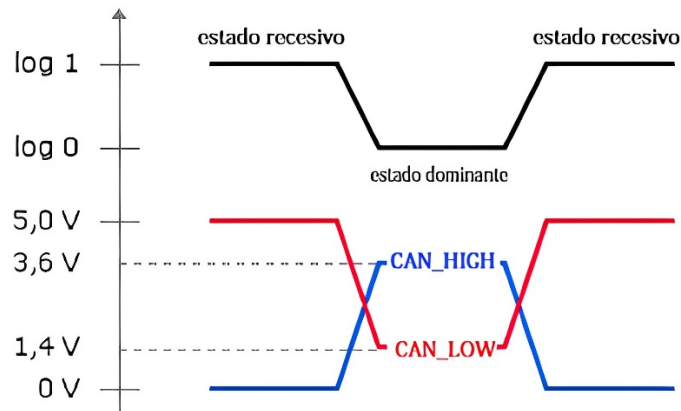


Fuente: (Autores)

Can High tiene polaridad opuesta a la línea CAN Low, lo que hace que todas las respuestas de estos dos sean opuestas entre sí, detección de errores en la línea de datos CAN BUS se puede efectuar si una de las dos genera una onda diferente a la otra. Cabe recalcar que existe una resistencia entre ambos de 120 ohmios, esto ayuda a reducir el ruido de las señales.

Figura 2

Niveles de baja velocidad CAN



Fuente: (Autores)

2.3 Organización de la red multiplexada

2.3.1 Identificadores estándar

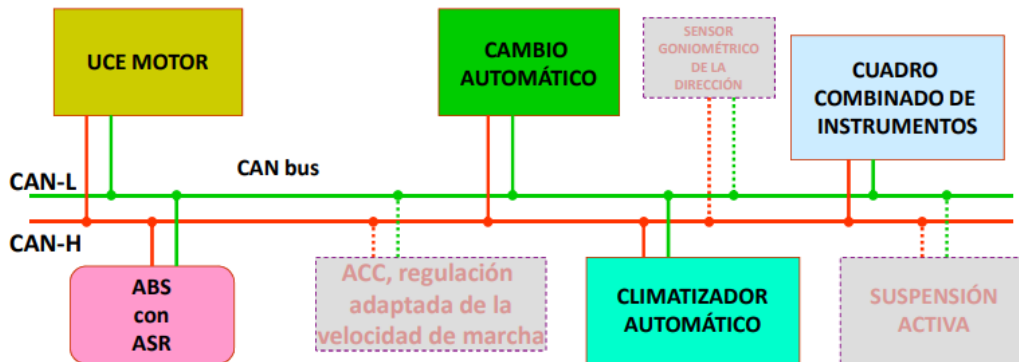
(11 bits): estos identificadores se utilizan principalmente para mensajes de baja prioridad y comunicación entre diferentes subsistemas del vehículo. El ID predeterminado varía de 0x000 a 0x7FF (0 a 2047 decimal).

2.3.2 Identificadores extendidos

(29 bits): estos identificadores se utilizan para una comunicación más compleja entre los mensajes de alta prioridad y la ECU del vehículo. Los identificadores extendidos van desde 0x00000000 a 0x1FFFFFFF (0 a 536,870,911 en decimal).

Figura 3

Diagrama básico de red multiplexada



Fuente: (UT05. Redes multiplexadas: CAN bus de datos, s/f)

2.4 Red Can Bus

El control mediante CAN BUS en la actualidad es algo indispensable en la nueva generación de vehículos, esto debido a las nuevas normativas de contaminación ambiental y debido a que las exigencias de los compradores cada vez crecen. A partir de la década de los 80 Robert Bosch crea la red de comunicación CAN BUS, un diseño increíble para esos años y algo totalmente revolucionario, este diseño renovó la industria automotriz y dio paso a nuevas tecnologías, como el control electrónico de ciertos componentes, la necesidad de los usuarios de cada vez tener mayor control sobre el vehículo es más grande, de allí nace este proyecto, la visualización de datos en vivo del vehículo siempre se han visto en vehículos de alta gama, sin embargo en los vehículos de baja gama solamente se disponen de relojes analógicos y no digitales (Martínez Requena, 2017).

2.4.1 Breve historia del CAN y su aplicación relevante en los vehículos

La tecnología empleada CAN que por sus siglas en inglés se refiere a (Controller Area Network) el cual se refiere a un protocolo de comunicaciones (Bus Automotriz) enormemente empleado en sistemas del vehículo que se requiere una gestión de datos en tiempo real, elaborado inicialmente por la empresa Bosch "Robert Bosch GmbH" durante el año de 1986, al presentarse la imprescindible exigencia de enlazar múltiples sistemas electrónicos presentes en los vehículos con un tipo de comunicación que asegure fiabilidad y solidez para una ejecución que desempeñe excelente función de activación, desactivación o variabilidad de los datos según lo requiera el comando de las variables en los distintos sistemas (Martínez Requena & García Martín, 2017).

El Can fue una implementación de comunicación de elevada relevancia en el progreso de acuerdo a la magnitud de conexiones entre los sistemas o dispositivos imprescindible para salvaguardar un enlace de elementos en comunicación. El mencionado protocolo tipo CAN hace uso de un bus que comunica y a la misma vez se comparte hacia los dispositivos disuadiendo la exigencia de una conexión del tipo punto a punto a cada sistema o elemento que así lo requiera (Martínez Requema & García Martín, 2017).

La importancia de la tecnología implementada a través del protocolo CAN y se ha mantenido prolongada en el paso del tiempo independientemente del surgimiento de más protocolos de comunicación con una superior transferencia de datos ya sea más veloz o a distancias más amplias, es fundamentalmente la fiabilidad de comunicación la cual proporciona determinante significado al momento de llevar a cabo elementos o sistemas implicados con propiedades basadas a tiempo real. (Martínez Requema & García Martín, 2017).

Inicialmente los dispositivos de control se los encontraba bajo el nombre de módulos de control de motor, mediante pasaba el tiempo mejoraron su funcionamiento a tal desarrollo del mencionado componente hasta lograr convertirse en Unidades de Control Electrónico o también conocidos bajo el nombre de ECU o inclusive como “centralitas” (Zambrano Beltrán et al., 2015).

2.5 Arquitectura del CAN BUS

La identificación de datos dentro de una red multiplexada (CAN BUS) es algo indispensable para el reconocimiento de valores dentro de la misma, es por eso que la gran mayoría siguen un patrón en común como se muestra en la Tabla 1

Tabla 1

Orden de la identificación de datos

PID	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7	BYTE 8

Fuente: (Autores)

Para lograr comprender un poco mejor la tabla debemos explicar a qué se refiere cada byte obtenido de la red, como primer requisito para entender un poco sobre el comportamiento de los bytes enviados y registrados por los módulos del vehículo debemos conocer el identificador de cada valor, mejor llamado “PID“, este es el encargado de brindar un número de

identificación al resto de bytes para dar a conocer que el resto de bytes serán los encargados de identificar a un componente en específico o a varias acciones de un componente.

El siguiente segmento de bytes es encargado de mostrar las acciones que genera dicho componente dentro del sistema electrónico del vehículo, como ejemplo podrás visualizar la Tabla 2 .

Tabla 2

Segmento de bytes.

A0	00	FF	01	16	32	00	FF	00
----	----	----	----	----	----	----	----	----

Fuente: (Autores)

Como se puede observar en la Tabla 2 el primer byte es el A0 y este es un PID, sin embargo, dependiendo del fabricante éste PID puede representar uno o más valores, los bytes a continuación son valores escritos en formato hexadecimal, estos datos para poder ser interpretados de una manera entendible se deben transformar a formato decimal y luego usados en formulas específicas para poder ser entendidos e interpretados.

2.6 Modos de medición OBDII

OBD2 (On-Board Diagnostics) es un sistema de diagnóstico a bordo estandarizado que se utiliza en vehículos modernos para monitorear e informar el funcionamiento del motor y otros sistemas relacionados. OBD2 define varios modos de operación para acceder y recuperar información de diagnóstico del vehículo. Dos de estos modos son el Modo 1 y el Modo 2.

Los objetivos principales de estos modos es guardar en la memoria del computador del vehículo los códigos de avería generados en cualquier sistema electrónico del vehículo y mostrarlos cuando el scanner ingresa dentro del sistema para poder realizar un correcto diagnóstico además de mostrar datos en vivo del vehículo (Recalde & Stefany, 2022).

Tabla 3 *Modos de medición OBDII*

Modo	Características Principales
1	Datos en vivo
2	Datos congelados
3	Códigos de falla
4	Borrado de códigos de avería

5	Resultados de pruebas
6	Resultados de pruebas
7	Muestra de códigos de avería pendientes
8	Funcionamiento de componentes
9	Información de automóvil

Fuente: (Wilson et al., 2016)

2.6.1 Modo 1

Se utiliza para solicitar y recuperar datos en tiempo real relacionados con los sensores y actuadores del motor y otros sistemas. Especifique la información en forma de "parámetro" o "PID" (ID de parámetro). Estos parámetros son códigos numéricos estandarizados que representan ciertos valores, como la velocidad del motor, la temperatura del refrigerante, la posición del acelerador, etc. Cuando se solicita un PID específico en el Modo 1, el vehículo responderá con el valor actual de ese parámetro (Zambrano et al., 2015).

Figura 4

Modo 1 a detalle

BYTE	INDICADORES	VALOR HEXADECIMAL
1	Indica que estamos realizando una petición en el Modo 1	01
2	Valor del PID requerido	FF
3	Valor hexadecimal del segundo PID	FF
4	Valor hexadecimal del tercer PID	FF
5	Valor hexadecimal del cuarto PID	FF
6	Valor hexadecimal del quinto PID	FF
7	Valor hexadecimal del sexto PID	FF

Fuente: (Autores)

2.6.2 Modo 2

Se utiliza para obtener información sobre los sistemas auxiliares de diagnóstico del vehículo, como el sistema de emisiones. Proporciona datos de estado de los componentes y sistemas relacionados con las emisiones, incluidos los resultados de las pruebas de autodiagnóstico realizadas por el vehículo. Al igual que en el modo 1, el modo 2 también utiliza códigos numéricos estandarizados para representar varios parámetros de diagnóstico (Zambrano et al., 2015).

2.7 Mecanismo de control PID

El controlador o regulador que se examinará y se hará uso es el PID (Proporcional, Integral, Derivativo) el cual se encuentra presente en la Unidad de Control electrónico (ECU) o en las unidades de control que lo requieran en el control de variables como temperatura, revoluciones por minuto, velocidad del vehículo, etc. (Molina & Oliva, 2014).

El PID se trata de un sistema de mando el cual se adecua por sí mismo, dando como resultado de un proceso la incidencia en el conjunto de la operación integrándose y adecuándose según sea necesaria, de un bucle cerrado. De esta manera el controlador cuantifica la desviación de un valor evaluado y otro de referencia recurriendo a un ajuste en el procedimiento. Hay tres factores que determinan el algoritmo de operación calculada del controlador: La cifra proporcional, integral y derivativa (Molina & Oliva, 2014).

En función a la clase de controlador hay casos que alguna cifra de dichos valores tome la cantidad de 0, el cual podría ser PI, PD o inclusive, P o I. Dichas referencias repercutirán de manera unitaria según el tipo de particularidad de la salida, evolución transitoria de la señal de salida de acuerdo a la cifra final (sobre oscilación), en consecuencia, el empleo y conjunción de distintas cifras en el sistema obtiene una salida con parámetros específicos (Molina & Oliva, 2014).

Dichos modos además de llevar un registro de datos también se encargan de la remisión de información que incorpora al software incorporado o implementado cuyo software actúa como un interpretador y nos ofrece una presentación de datos eficientes y fiables apoyando notablemente al usuario que requiere de un diagnóstico óptimo del vehículo e identificar posibles fallos tanto eléctricos como electrónicos (Sánchez Carrizo, 2017).

2.8 Tipos de PID

En una red CAN Bus, los Identificadores de Parámetros (PID) son códigos específicos utilizados para solicitar información a los diferentes módulos o dispositivos conectados a la red. Cada módulo tiene una serie de PID asignados que permiten obtener datos sobre el estado, diagnóstico y configuración del dispositivo.

Los PID están estructurados en bytes y se dividen en varios tipos:

2.8.1 PID genéricos

Son PID estándar definidos por el protocolo CAN y se utilizan para acceder a datos básicos de los vehículos, como la velocidad del motor, la temperatura del refrigerante, el estado del sistema de combustible, etc. Algunos ejemplos son el PID 0x0C (RPM del motor) y el PID 0x0D (velocidad del vehículo) (Recalde & Stefany, 2022).

Los PID genéricos están estandarizados por legislaciones y regulaciones de emisiones de vehículos en 2006, sin embargo, debido a la variedad de vehículos dentro del mercado y sus diferentes prestaciones se decidió suprimir este requisito desde el año 2001 a partir de este año los diferentes fabricantes optaron por usar sus propios PID.

Tabla 4

PID Genérico ejemplo

Nombre	Sensor	Unidad	PID	Fórmula
Velocidad del vehículo	VSS	Km/h	0D	Valor en decimal

Fuente: (Autores)

2.8.2 PID del fabricante

Son PID específicos de cada fabricante de vehículos y brindan acceso a datos más detallados y personalizados. Cada fabricante puede definir sus propios PID para obtener información específica de sus sistemas y componentes.

Los PID de fabricante obligan al técnico tener una mayor gama de equipos, ya que para poder realizar un correcto diagnóstico dentro del sistema electrónico del vehículo es necesario tener un scanner multimarca o en ciertas ocasiones es indispensable tener un scanner original, estos generan un acceso restringido dentro del sistema y niegan el acceso a cualquier técnico no autorizado.

Figura 5

PID del fabricante

Identifier	Länge	Datenbytes								Periode	Anzahl
000h	0									960	6
018h	8	00	00	00	60	00	80	00	00	200	2814

Fuente: (KMTecSoft, 2017)

2.9 CAN BUS

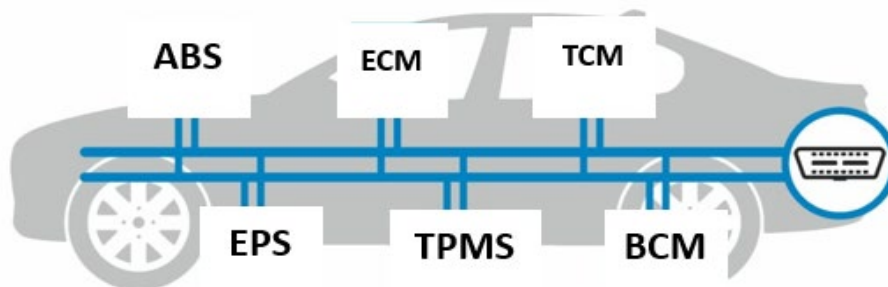
CAN bus (Controller Area Network bus) es un estándar de comunicación utilizado en vehículos y otros sistemas distribuidos para permitir la transferencia de datos entre diferentes componentes electrónicos. Fue desarrollado originalmente por Bosch en la década de 1980 y se ha convertido en el protocolo de comunicación más comúnmente utilizado en la industria automotriz.

El CAN bus permite que los diferentes módulos y dispositivos electrónicos dentro de un vehículo se comuniquen entre sí de manera eficiente y confiable. Estos componentes pueden incluir el motor, la transmisión, el sistema de frenos, el sistema de dirección, los sistemas de seguridad, el sistema de entretenimiento y muchos otros.

La principal ventaja del CAN bus es su capacidad para transmitir datos de manera rápida y confiable a través de una sola línea de comunicación compartida por todos los dispositivos conectados. Esto reduce la necesidad de cables y simplifica la arquitectura del sistema, lo que a su vez ayuda a reducir el peso y el costo del cableado en los vehículos.

Figura 6

Diferentes módulos del CAN Bus



Fuente: (DevCom, 2022)

La red CAN BUS está incorporada en todos los sistemas operativos electrónicos del vehículo con el objetivo principal de informar a todos los módulos el estado del vehículo en milésimas de segundos, sin necesidad de usar nada más que dos cables para ello (Recalde & Stefany, 2022).

2.9.1 Hexadecimal

El sistema hexadecimal es un sistema de numeración que utiliza una base de 16. A diferencia del sistema decimal (base 10) que utilizamos cotidianamente, donde tenemos 10 dígitos del 0 al 9, en el sistema hexadecimal se utilizan 16 dígitos: del 0 al 9 y las letras A, B, C, D, E, F.

Los valores de estos dígitos en el sistema hexadecimal son equivalentes a los valores decimales del 0 al 15. (Jiménez, 2020)

La secuencia de los dígitos en hexadecimal es la siguiente: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. El dígito A representa el valor decimal 10, B representa el valor decimal 11, y así sucesivamente hasta F que representa el valor decimal 15. (Jiménez, 2020)

El sistema hexadecimal es ampliamente utilizado en la informática y la programación, ya que es una forma conveniente de representar números binarios de una manera más compacta y legible. Cada dígito hexadecimal se corresponde con un grupo de 4 bits en el sistema binario, lo que facilita la conversión entre ambos sistemas. (Jiménez, 2020)

2.10 Registro de datos

Los datos enviados por los módulos del vehículo no son enviados en forma analógica, como actúan la mayoría de los sensores o actuadores, sino estos son enviados mediante una serie de códigos digitales, este reconocimiento de datos se hace mediante el uso de dispositivos complejos como lo hace el scanner automotriz, este scanner analiza el tipo de protocolo que usa un vehículo y refleja datos en vivo del automotor.

Estos datos por lo general llegan al scanner automotriz en formato hexadecimal, el scanner analiza estos datos, hace las transformaciones y los coloca en formato decimal, para que podamos entenderlos de una manera más suave.

El registro de datos dentro de un BUS de datos se desarrolla de bite en bite, esto sucede en milésimas de segundo, con el objetivo de que todos los módulos del vehículo tengan información sobre el resto de los sistemas del vehículo y de esta forma poder tomar decisiones sobre el control del automóvil.

Cada módulo del vehículo tiene un analizador de datos, este analizador recepta los datos que le interesan conocer y descarta los datos que no necesita mediante filtros.

2.11 Datos del CAN BUS.

Los datos de la BUS siguen un orden específico, el cual depende del diseñador del vehículo, sin embargo, estos siguen un patrón en común, como los PID de cada variable, estos son los encargados de brindar una identificación de cada componente del sistema electrónico del vehículo, los siguientes dígitos luego de este identificador representan al estado del sensor en

posee un identificador menor (un mayor número de bytes dominantes al inicio de las posiciones) tendrá relevancia superior con respecto a una trama con identificador mayor.(Martínez Requema & García Martín, 2017)

El formato estándar está formado por la cantidad de 11 bytes, finalizado con 1 byte agregado que se usa para distinguir entre las tramas de datos (valor 0), o tramas de petición remota (valor 1). (Martínez Requema & García Martín, 2017)

Referente a el campo de control, los 2 bytes que se identifican al inicio se reservan para un uso posterior y más adelante se halla 4 bytes, que fijan la magnitud del campo de datos referente a la trama consiguiente a esta. Posteriormente la trama que se encuentra coloreada de rojo en la Figura 7 implica un campo de datos que puede consistir de 0 a 8 bytes.(Martínez Requema & García Martín, 2017)

Finalmente, el Campo de CRC (Código de redundancia cíclica), cumple la función de asegurar la integridad de los datos transmitidos. Finaliza con un bit de contracción o recesivo para el delimitador CRC, un campo de magnitud de 2 bytes que será usado de celda de reconocimiento conocida bajo el nombre de ACK y por último el EOF que representa el fin de la trama que comprende en 7 bytes recesivos sucesivos. Al final de la trama debe existir un espaciado reglamentario en medio de las tramas, conformado de 3 bytes recesivos.(Martínez Requema & García Martín, 2017)

2.13 Seguridad en CAN BUS

La seguridad de una red multiplexada es algo muy importante dentro del sistema de control del vehículo, esto se debe a que la red controla toda la red electrónica del vehículo y si ésta sufre de ataques cibernéticos podría ser algo perjudicial para los usuarios de los vehículos, de esta necesidad nacen los firewalls, para evitar que cualquier persona pueda controlar la red del vehículo.

Las nuevas tecnologías de las redes multiplexadas CAN BUS cada vez tienen un mayor número de complejidades en sus sistemas, como los nuevos firewalls, estos son sistemas los cuales previenen accesos a usuarios no autorizados a la lectura de los datos del vehículo, así mismo como impedir que estos datos sean manipulados, las características más primordiales de los firewalls es el control de acceso, filtrado de datos, detección de anomalías y la seguridad en

capas, estos son controles de los firewalls que tienen ciertas marcas de vehículos, ya que implementar éstos dentro de un vehículo es de elevado coste, sin embargo para ciertos diseñadores es indispensable que no se copien sus datos y puedan ser utilizados en otros vehículos con prestaciones similares, cabe recalcar que la implementación de firewalls pueden variar entre los diferentes fabricantes de vehículos y modelos, sin embargo el mercado automotor debe de adaptarse a ciertas necesidades de los usuarios, como obtener valores precisos en la gestión electrónica del vehículo.

2.14 Introducción al OBD II

El modelo OBD II se refiere a un tipo de norma que con el paso del tiempo ha ido logrando acoplarse globalmente en los diferentes tipos de vehículos (Sedán, Cuv, Suv, Hatchback, Pickup, etc.) se ha estipulado con universalidad la índole de compatibilidad, requiriendo gran flexibilidad de acuerdo con la amplia gama de automóviles así mismo como las insignias automotrices (Sánchez Carrizo, 2017).

Además, dicha diversidad ha dado lugar a considerables números de softwares que tratan como tema fundamental el diagnóstico automotriz que se logra operar mediante el estándar mencionado extendiéndose hacia modelos de vehículos. Hacia un tiempo atrás, los fabricantes de cada marca hacían uso de sistemas de diagnóstico que eran propios de ellos elaborando su protocolo de comunicación autónomo que incluía el conector específico para poner en marcha el diagnóstico de los vehículos.

Posteriormente se estableció la obligación de implementar un puerto de conexión con una base de patrón de especie trapezoidal que cuente con 16 pines cuyo puerto brinde asistencia en el diagnóstico OBD II. El objetivo fue conceder la facultad de contener protocolos que logren comunicar con un solo instrumento de prueba.

Mediante los años transcurridos de la década de los años 80, la mayoría de las innovaciones efectuadas en el sector automotriz se vieron situadas mediante la aplicación de tecnología electrónica en lugar de mecánica. Se agregaron sensores nuevos a los automóviles y se perfeccionó las unidades de control implicadas en el motor. Este mencionado progreso en el mundo automotriz fue tan importante que en los años actuales en un automóvil común de gama media puede involucrar a al menos 200 sensores (Zambrano Beltrán et al., 2015)

2.15 Función del OBD en los vehículos

El mecanismo para diagnóstico vehicular bajo el nombre de OBD (On Board Diagnostics) basa su funcionalidad en primordialmente el monitoreo y control de distintas características propias de los vehículos, la identificación de posibles falencias que perjudiquen el sistema eléctrico, electrónico y mecánico de los automóviles bajo la vanguardia de la tecnología contemporánea. Integra la capacidad integral del motor, seguimiento del chasis entre otras funciones de relevancia en los sistemas electrónicos implementados indispensable para el correcto análisis del estado en que se encuentra el vehículo (Sánchez Carrizo, 2017).

OBD II supervisa de manera exhaustiva el adecuado comportamiento de los múltiples sistemas presentes en el vehículo así mismo como sensores, actuadores o unidades de control electrónico (Motor, Sistema de frenos antibloqueo, Control de estabilidad electrónico, Sistema de control de tracción, Sistema de control de presión de neumáticos, etc.). En el preciso momento que se efectuó un fallo imprevisible, el OBD se encarga de localizarlo, para subsiguientemente informar al interfaz del usuario del automóvil mediante una señal luminosa de color amarillento (testigo o indicadores) o a su vez de sonido, de acuerdo corresponda el caso o bien el procedimiento de aviso según el fabricante.

Los modos de medición del OBD II proporcionan una gran cantidad de tareas o funcionalidades, que van incluidos en los protocolos, los modos inducen los componentes de información que se requiere ingresar, a través de cada modo se puede recurrir a varios parámetros de sistemas o elementos inmiscuidos en la Red, a su vez cumple con la capacidad de llevar una contabilidad de información para una posterior comprobación, por otra parte es posible la separación de códigos de fallo, suprimirlos y llevar a cabo test de sensores y actuadores en tiempo real.

2.16 Unidad de Control Electrónico

Las actuaciones de inspección y análisis ejecutadas por el OBD II se llevan a cabo gracias a la Unidad de Control Electrónico conocida por la abreviatura comúnmente de (ECU), a la cual es posible el enlazamiento para obtención de datos recibidos por la misma. En los vehículos

actuales se hallan dispuestos más de una ECU para el tratamiento de variables autónomas con el fin de un funcionamiento apropiado de todo el automóvil en general (Sánchez Carrizo, 2017)

Hoy por hoy los sensores tienen una relevante importancia en lo que se refiere al monitoreo, control, medición de variables ambientales o que se ven implicadas en el vehículo como (temperatura del refrigerante, presión, posición del cigüeñal, posición del pedal de aceleración, nivel del combustible, golpeteo del motor, velocidad del vehículo, revoluciones por minuto, etc.) cabe destacar también la basta cuantía de factores que se ven implicados con el correcto desempeño de los motores de combustión interna (MCI) o motores de encendido por compresión (MEC) (Zambrano Beltrán et al., 2015).

Los datos que reciben los sensores por medio de esta comunicación es transmitida para posteriormente ser acumulada en la Unidad de Control Electrónico que por consiguiente a esto se confronta con valores que se conservan en la memoria. En caso de existir una estimación que no sea la correcta la ECU alerta del fallo ocurrido al conductor de la manera en que haya sido programada o de la forma que sea más oportuna según el fabricante las cuales varían pero de las cuales pueden ser (testigos de verificación o luces de indicadores, mensajes que se reflejan en el panel de instrumentos, pitidos audibles o sonidos que alerten al conductor, modos de emergencia que limiten el desempeño del motor según la situación en que se vea afectado y determine la ECU, etc.) los errores se mantendrán en la memoria con el fin de que en un futuro se puedan analizar a detalle (Zambrano Beltrán et al., 2015).

La importancia de los sistemas que se emplea para el diagnóstico a bordo o de autodiagnóstico que detectan fallas de los vehículos es una característica que dependerá de los fabricantes automotrices mediante una interfaz adecuada para obtener la información correcta y mantener un sistema de diagnóstico que garantice al conductor seguridad y brinde facilidad a la hora del mantenimiento así también como una reparación sencilla que apunte directamente a el inconveniente presentado.

2.17 Protocolos de comunicación automotriz

2.17.1 SAEJ1939

SAE J1939: Es un protocolo basado en CAN (Controller Area Network) utilizado principalmente en vehículos comerciales pesados como camiones y autobuses, define la comunicación entre varios sistemas electrónicos de un vehículo como motor, transmisión, sistema de frenos, etc. Proporciona una estructura de red jerárquica y una variedad de mensajes y parámetros estandarizados. (Control & Committee, 2023)

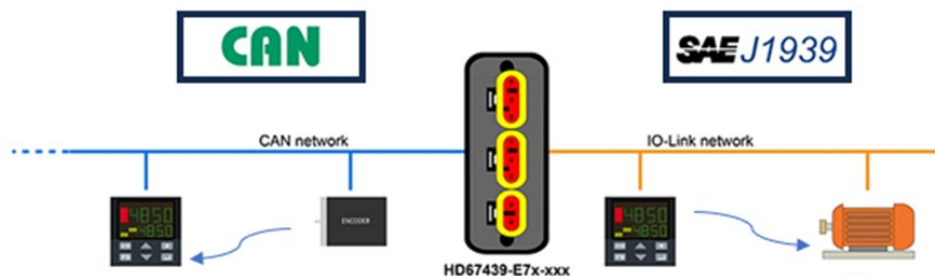
El estándar SAE J1939 define una variedad de mensajes estandarizados, denominados unidades de datos de protocolo (PDU), que permiten la transmisión de datos específicos. Cada mensaje de PDU contiene información como el identificador del mensaje, los datos asociados y el período de actualización. Además, SAE J1939 define un conjunto de parámetros estandarizados que describen varias variables y condiciones del vehículo. (Control & Committee, 2023)

Estos parámetros permiten el monitoreo en tiempo real de los sistemas del vehículo, como la velocidad del motor, la temperatura del refrigerante, la presión del aceite y más. SAE J1939 se ha convertido en un estándar importante en la industria automotriz porque permite una comunicación confiable y segura entre varios sistemas electrónicos en vehículos pesados.

Además, al adherirse a los estándares comunes, los fabricantes pueden desarrollar componentes y sistemas que sean interoperables, lo que facilita la interoperabilidad y el mantenimiento de los vehículos. (Control & Committee, 2023)

Figura 8

Diagrama comparativo entre SAEJ1939 y CAN



Fuente: (Control & Committee, 2023)

2.17.2 ISO9141

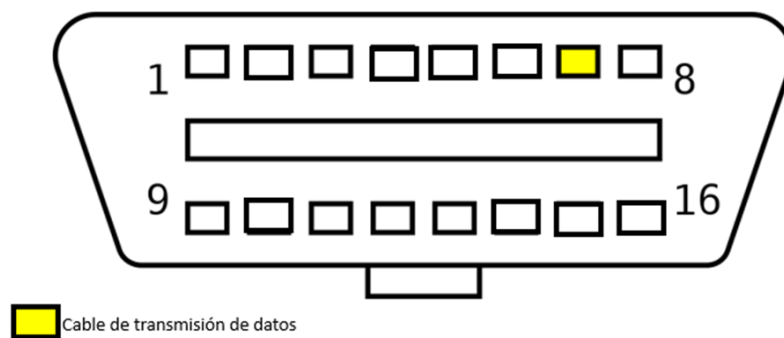
El estándar fue desarrollado por la Organización Internacional de Normalización (ISO) y se usa principalmente en vehículos más antiguos fabricados antes de la adopción generalizada del protocolo CAN (Controller Area Network), define la estructura y el protocolo de comunicación de los mensajes de diagnóstico utilizados en las redes. Utiliza una arquitectura de bus de un solo cable donde todos los dispositivos comparten una línea de comunicación común.

La comunicación ISO 9141 se basa en una estructura maestro/esclavo. El dispositivo de diagnóstico (maestro) envía solicitudes a los módulos electrónicos del vehículo (maestros) para obtener información de diagnóstico o enviar comandos de control.

El protocolo ISO 9141 se divide en dos variantes: ISO 9141-1 e ISO 9141-2. ISO 9141-1 define la capa física para la comunicación mediante la transmisión de datos basada en modulación de ancho de pulso (PWM). ISO 9141-2 define la capa de aplicación y define formatos de mensajes, códigos de diagnóstico y otros aspectos de la comunicación (Asociación Española de Normalización y Certificación, 2013).

Figura 9

Conexión ISO 9141



Fuente: (Asociación Española de Normalización y Certificación, 2013)

2.17.3 Estructura maestra/esclavo

La estructura maestra/esclavo genera una comunicación eficiente dentro de los sistemas de comunicación, permiten al módulo maestro obtener información a través de los módulos esclavos y poder controlarlos a su conveniencia, el maestro envía mensajes en código hexadecimal a los esclavos y ellos responden realizando acciones o con respuestas de características similares.(García & Tejero, 2015)

2.17.4 ISO 15765

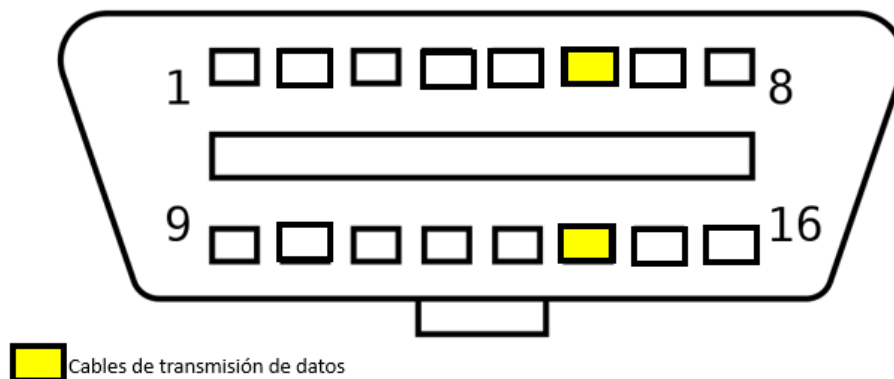
ISO 15765, también conocida como ISO 15765-2, es una norma internacional desarrollada por la Organización Internacional de Normalización (ISO) que define el protocolo de comunicación de diagnóstico en vehículos que utilizan Controller Area Network (CAN).

Es una parte esencial del sistema de diagnóstico a bordo (OBD-II) que se utiliza en los vehículos modernos. El protocolo ISO 15765 se basa en una arquitectura maestro/esclavo que permite la comunicación bidireccional entre el equipo de diagnóstico (maestro) y los módulos electrónicos del vehículo (esclavo).

A diferencia de los estándares anteriores, como ISO 9141 y SAE J1850, que usaban líneas de comunicación en serie, ISO 15765 se basa en Controller Area Network (CAN), que es una red de comunicación más rápida y eficiente (International Standard, 2005).

Figura 10

Conexión ISO 15765



Fuente: (International Standard, 2005)

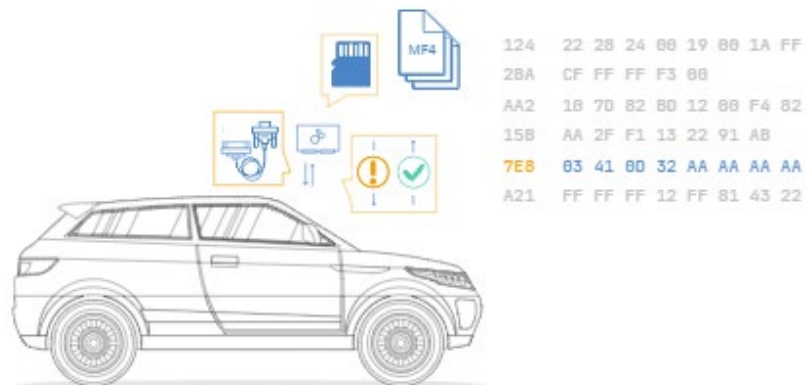
CAPITULO III

HIPÓTESIS

En este capítulo se abordará las variables tratadas como (Temperatura del refrigerante, Revoluciones por Minuto, APP, TPS, MAP, CKP) el PID encontrado de cada una de las variables y el tratamiento de la obtención del dato en código hexadecimal transformado a código decimal con el fin de poder implementar fórmulas algebraicas con el fin de calcular su valor exacto a tiempo real desde el Banco de Sensores, Actuadores y Unidad de Control.

Figura 11

Registro datos OBD2



Fuente: (CSS Electronics, 2023a)

A continuación, se dará un vistazo de los sensores que intervienen para obtener las variables requeridas para el presente proyecto además de hacer uso de la fórmula algebraica que se obtuvo mediante una herramienta de [búsqueda/ conversión, tabla, CSV, DBC] para descripción general de OBD2 PID

Figura 12

Distribución de PID's y Bytes en columnas

PID	Columna1	Columna2	Columna3	Columna4	Columna5	Columna6	Columna7	Columna8
43F	14	65	61	FF	0	0	0	0
A0	0	4E	0	0	0	25	0	0
390	3	23	EA	0	1A	0	0	2A
43F	14	65	61	FF	0	0	0	0
545	32	0	0	74	0	0	0	0
43F	14	65	61	FF	0	0	0	0
2A0	E	0	5D	0	89	F	49	6
43F	14	65	61	FF	0	0	0	0
260	1E	35	FF	0	2B	A6	64	31
43F	14	65	61	FF	0	0	0	0
390	3	23	EA	0	28	0	0	38
18F	FA	FF	0	0	0	78	0	0

Fuente: (Autores)

3.1 Temperatura (Sensor ECT)

Figura 13

Sensor ECT del Banco de Sensores, Actuadores y Unidades de Control



Fuente: (Autores)

El sensor ECT que por sus siglas en inglés se refiere a (Engine Coolant Temperature) o (Sensor de Temperatura del Refrigerante del motor) es un componente electrónico crucial de los vehículos modernos, dicho sensor monitorea la temperatura del líquido de enfriamiento que circula por el motor. De esta manera suministra datos importantes para el sistema de funcionamiento del motor, ayudando a mantener una temperatura adecuada para un rendimiento óptimo del motor.

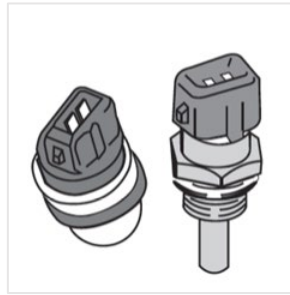
El Sensor ECT parte desde el desarrollo de los MCI o (Motores de Combustión Interna) en vista de la importancia del control de la temperatura. A medida que los motores fueron perfeccionando al paso del tiempo, se hizo evidente la relevancia de monitorear y controlar la temperatura ya que permite asegurar el funcionamiento de los motores de la manera más óptima posible evitando un sobrecalentamiento del mismo.

En los primeros años del motor de combustión interna, la temperatura del refrigerante se obtenía mediante la medición con un termómetro mecánico. El funcionamiento de dichos termómetros se basaba en la expansión y contracción de un bulbo que estaba repleto de líquido el cual se enlazaba a una barra indicadora. Este método empleado no cumplía con una eficacia adecuada por lo que era bastante impreciso y requería de una lectura visual. Los sensores actuales de temperatura los cuales son electrónicos, como el nombrado sensor ECT, fueron implementados de acuerdo a la evolución de la tecnología.

Estos sensores calculan la temperatura del refrigerante con una mayor precisión haciendo uso de métodos como la resistencia o el efecto termoelectrico. Para posteriormente transferir los datos obtenidos al sistema de gestión o funcionamiento del motor. Mediante la digitalización de los sistemas implicados como el mencionado anteriormente en la gestión del motor en las últimas décadas, los sensores ECT han ido perfeccionando. Actualmente se utilizan los termistores los cuales son aparatos que varían una resistencia eléctrica interna de acuerdo a la temperatura recibida obteniendo valores a tiempo real más precisos que son transferidos al sistema de gestión del motor.

Figura 14

Sensor NTC



Fuente: (Carranco, 2022)

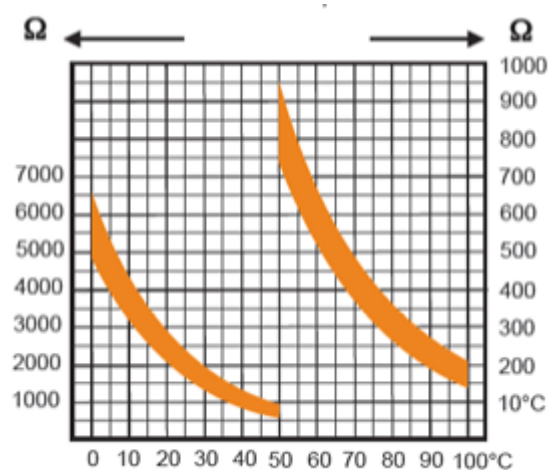
3.1.1 Funcionamiento del Sensor ECT

El Sensor ECT suele encontrarse generalmente en la culata del motor o aproximadamente cerca del termostato como se mencionó anteriormente este sensor se encarga de monitorear la temperatura del refrigerante del motor, mediante un termistor de coeficiente de temperatura negativo más conocido como (NTC), basa su funcionamiento en una resistencia la cual disminuye su ohmiaje a medida que la temperatura del refrigerante asciende.

El termistor tipo NTC consta de un material que es sensible al calor, lo que incide directamente en su resistencia eléctrica modificándola de acuerdo a las variaciones que ocurren con la temperatura.

Figura 15

Resistencia de las sondas de temperatura

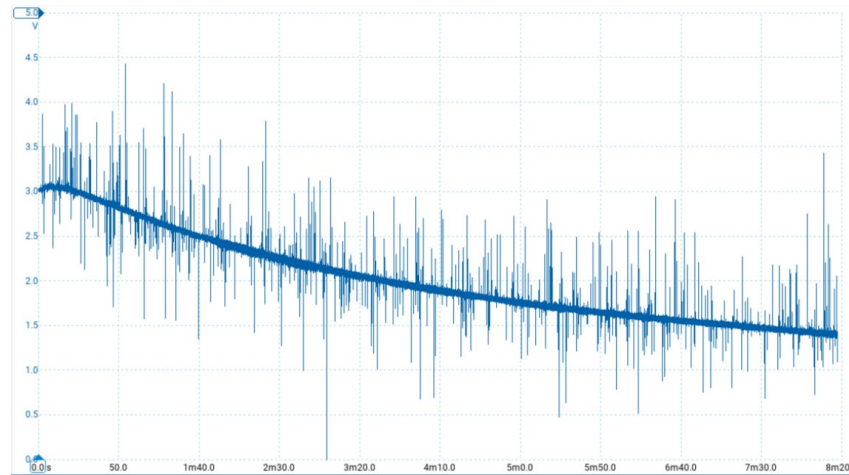


Fuente: (Carranco, 2022)

Mediante dicho funcionamiento es posible que el sensor logra emitir una señal de voltaje que es recibido por la Unidad de Control Electrónico (ECU) o ya sea el Módulo de Control del Motor (PCM). Los datos recopilados se emplean para controlar el funcionamiento del motor ejerciendo un efecto en diversos aspectos como (ajuste de la mezcla del combustible, tiempo de encendido, activación del electroventilador, etc.).

Figura 16

Ejemplo forma de Onda del Sensor de Temperatura del Refrigerante del Motor



Fuente:(Picoauto, 2022)

3.1.2 Fórmula algebraica Sensor ECT

Tabla 5

PID Obtenido del Sensor ECT y fórmula algebraica

PID	Data Byte ubicación	Fórmula
A0	Columna 2	$-40 + 1 * (\text{Valor byte HEX} \rightarrow \text{DEC})$
		Respuesta en Grados Celsius [°C]

Nota: Fórmula obtenida de (CSS Electronics, 2023b)

A partir de la base de datos obtenidas en la Placa Arduino Uno se logró obtener el PID del dato correspondiente al Sensor ECT, cuyo resultado refleja un valor alfanumérico A0 correspondiente al mismo, por consiguiente, se pudo detectar el valor da fluctuación en los

datos de la Temperatura del Refrigerante del Motor precisamente en el Data Byte que se encuentra en la columna 2.

A el valor hexadecimal detectado fue necesaria la transformación del sistema hexadecimal a decimal, una vez realizada la conversión mencionada fue necesario aplicar la fórmula de la Tabla 5 obteniendo eficazmente el valor de la temperatura del ECT una vez obtenido el dato fue indispensable la comprobación por medio de la asistencia del Scanner Automotriz cuyo dato comprobaría de manera exacta la ubicación del Data Byte encontrado y la Temperatura exacta a partir del valor hexadecimal hasta la transición por la fórmula algebraica empleada.

3.2 Revoluciones (Sensor CKP)

El sensor CKP o mejor conocido como (Crankshaft Position Sensor) es un componente piezoeléctrico utilizado en la industria automotriz que nace de la necesidad de tener un mejor control sobre el vehículo, esto con el objetivo de mejorar el consumo y la eficiencia del mismo, a partir del desarrollo tecnológico que el campo automotriz demanda es necesario buscar una mayor eficiencia dentro de los sistemas de inyección a combustible.

Figura 17

Sensor CKP obtenido del Banco de Sensores, Actuadores y Unidades de Control



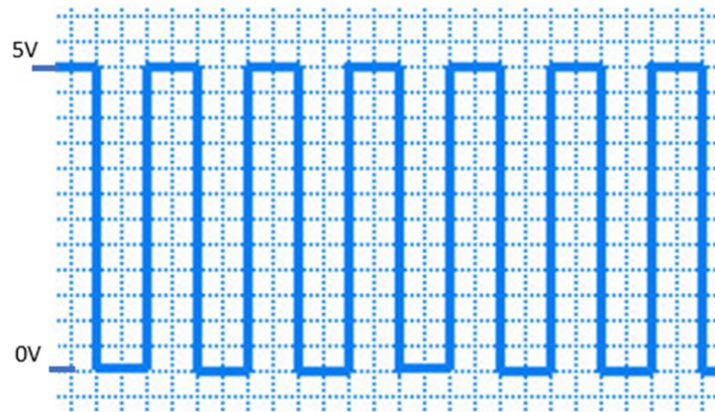
Fuente: (Autores)

El funcionamiento básico del sensor CKP es basado en detección de cambios magnéticos generados por dientes de la rueda dentada, el sensor CKP dependiendo del fabricante puede usar tecnología de efecto Hall o inductiva, este sensor por lo general se encuentra fijado al

cigüeñal, este sensor es uno de los más importantes dentro del sistema de inyección, ya que es el encargado de informar a la computadora la velocidad de giro y posición del cigüeñal.

Figura 18

Funcionamiento sensor CKP de efecto Hall

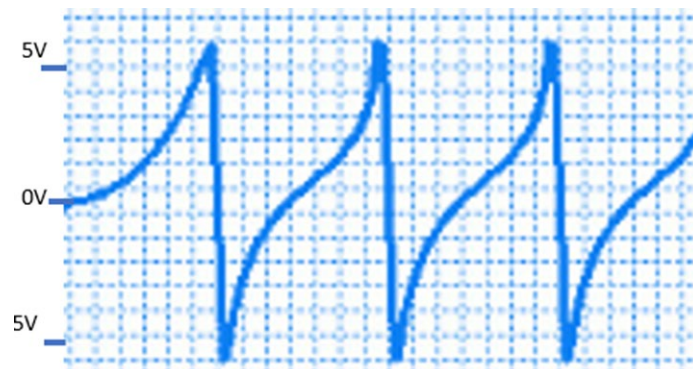


Fuente: (Picoauto, 2022)

Como se puede observar en la Figura 15 el funcionamiento del sensor CKP de efecto Hall, cuyo sensor recibe pulsaciones de acuerdo a la posición de la rueda fónica, este sensor en la actualidad no es el más preciso, ya que no se puede determinar el grado exacto en el que se encuentra el cigüeñal, de lo contrario el sensor CKP inductivo, será el encargado de generar una onda, la cual refleja con una mayor precisión el ángulo donde se identifica el cigüeñal.

Figura 19

Funcionamiento sensor CKP inductivo



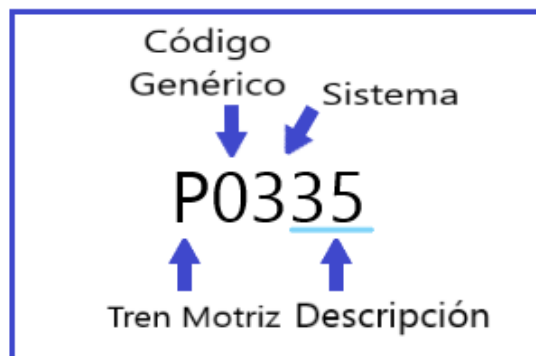
Fuente: (Picoauto, 2022)

El sensor CKP de efecto Hall muestra pulsaciones cuadradas, las cuales con sus picos de voltaje muestran cada diente de la rueda dentada, de esta manera la computadora logra interpretar cada diente de la rueda dentada, mientras que el sensor CKP de efecto inductivo muestra datos más precisos sobre la posición del cigüeñal de manera tal que podemos conocer el ángulo exacto del cigüeñal para una mejor gestión electrónica.

Los rangos de trabajo del CKP dependen del fabricante, sin embargo, en la mayoría de los sensores CKP oscila el rango de trabajo entre 0 y 5 voltios, si el rango de voltaje de estos sensores varía sobre o bajo el rango de trabajo la computadora tomará este dato como una avería en el sensor y tomará acciones para indicar al usuario que éste se encuentra fallando, para informar al usuario que este sensor se encuentra fallando el motor no encenderá, se quedará encendido el indicador del Check Engine y el código de avería que mostrará en el scanner automotriz será el siguiente:

Figura 20

Descripción de código de avería



Fuente: (Autores)

El código de avería se divide en 4 componentes, los cuales se expresan de la siguiente forma:

Tabla 6

Componentes del código de avería

P	0	3	3	5
---	---	---	---	---

Este dígito se refiere a sistema que se encuentra en falla, en este caso se refiere allí tren motriz.	Este dígito nos indica si el código de avería es un código genérico o específico del fabricante	Este dígito se refiere al subsistema que está fallando, en este caso se refiere al sistema de encendido.	Estos dos dígitos se refieren al sensor exacto en donde se encuentra la avería.
---	---	--	---

Fuente: (Autores)

3.2.1 Fórmula algebraica Sensor CKP

La señal del sensor CKP se encuentra dentro de la información de la red CAN BUS, no obstante, esta debe de tener un carácter en específico para que los módulos del sistema logren reconocer este valor este es conocido como PID, el cual para representar el valor de las rpm es el A0, una vez identificado el identificador del CKP debemos seleccionar los datos de la tercera y cuarta columna de este PID, ya que luego estos valores serán transformados a formato decimal y luego multiplicado a 0.25 para conocer el valor exacto de las revoluciones por minuto del motor.

El valor obtenido es un valor decimal el cual debe de ser comparado con asistencia de herramientas automotrices como lo es un scanner automotriz el cual ayudará a comprobar el valor exacto del dato obtenido a tiempo real.

Tabla 7

PID Obtenido del Sensor CKP y fórmula algebraica

PID	Data Byte ubicación	Fórmula
A0	Columna 3 y 4	$0 + 0.25 * (\text{Valor byte HEX} \rightarrow \text{DEC})$
	“Ubicar al revés en la fórmula los data byte primero 4 ^{to} y luego 3 ^{er} byte es un numero de 4 cifras HEX junto, entre la unión de los dos bytes”	Respuesta en [RPM]

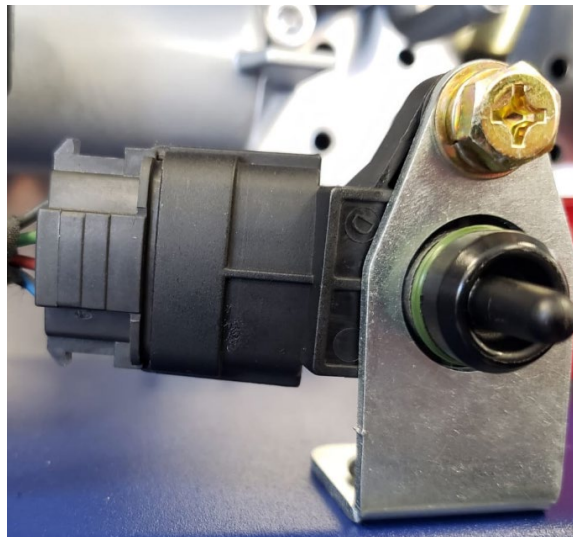
Nota: Fórmula obtenida de (CSS Electronics, 2023b)

3.3 Sensor MAP (Presión Absoluta del Múltiple)

El Sensor MAP es una pieza electrónica que usualmente se lo emplea actualmente en los sistemas de inyección de combustible que se disponen para los motores de los vehículos. Este tipo de sensores pueden remontarse hacia años atrás. No obstante, la evolución que ha experimentado en el paso de los vehículos se ha visto beneficiada por el aumento de sensores que puedan monitorear y controlar variables importantes que ocurren a tiempo real para poder enfatizar en el desenvolvimiento óptimo del motor ya sea desde la Unidad de Control Electrónico o ECU obteniendo datos precisos al instante en que se está ejecutando la activación de los actuadores, en este caso de los Inyectores.

Figura 21

Sensor MAP del Banco de Sensores, Actuadores y Unidades de Control



Fuente: (Autores)

Un punto crucial del control de la inyección por medio del aporte de Sensores como el MAP es la de poder mitigar las emisiones contaminantes generadas por el vehículo. En este apartado es importante el papel que desempeña el sensor MAP, ya que nos ayuda a regular la cantidad de aire que ingresa al sistema del motor y a realizar una mezcla óptima aire-combustible.

3.3.1 *Fórmula algebraica Sensor MAP*

Se logró la obtención del PID del dato correspondiente al Sensor MAP, cuyo resultado refleja un valor alfanumérico A1 correspondiente al mismo, por consiguiente, se pudo detectar el valor de fluctuación en los datos de la Presión precisamente en el Data Byte que se encuentra en la columna 5.

Tabla 8

PID Obtenido del Sensor MAP y fórmula algebraica

PID	Data Byte ubicación	Fórmula
A1	Columna 5	$0 + 1 * (\text{Valor byte HEX} \rightarrow \text{DEC})$
		Respuesta en kPa

Nota: Fórmula obtenida de (CSS Electronics, 2023b)

3.4 **Sensor de Posición del Pedal de Acelerador (APP)**

Figura 22

Sensor APP del Banco de Sensores, Actuadores y Unidades de Control

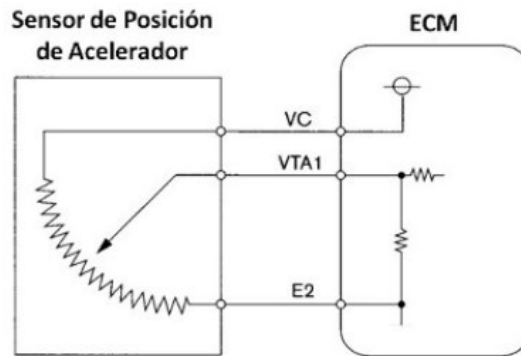


Fuente: (Autores)

El sensor de pedal de acelerador conocido comúnmente como APP es un sensor de tipo potenciómetro, su composición interna es de discos de carbono, los cuales generan una resistencia, la cual es encargada de provocar una variación de voltaje dentro del sistema.

Figura 23

Circuito del Sensor de Posición del Acelerador



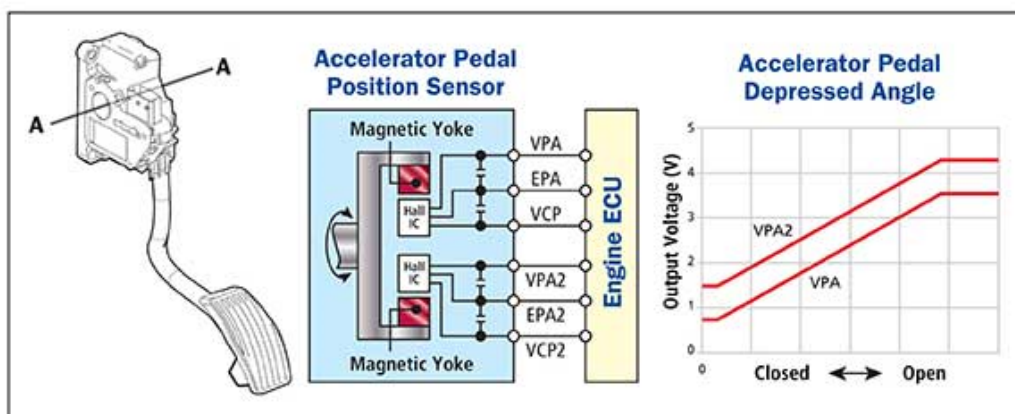
Fuente: (Ingeniería Mecánica Automotriz, 2021)

Funcionamiento del Sensor de Posición del Pedal

El funcionamiento del Sensor APP depende del fabricante, sin embargo, en su gran mayoría el funcionamiento es el de calcular la posición en que se encuentra el pedal al accionar el pedal del acelerador del vehículo y de esta manera recopilar información que se enviará directamente a la ECM (Engine Control Module) por consiguiente el módulo de control del motor se encarga del ajuste de la mezcla de aire combustible de acuerdo a la situación que se encuentre o lo amerite mediante el control del sensor de posición del pedal de aceleración.

Figura 24

Funcionamiento Sensor APP con representación de señales de salidas de voltaje



Fuente:(PremierAuto Trade, 2023)

Cabe recalcar que dentro de este sensor se encuentran colocados dos potenciómetros, los cuales envían a la ECU la misma señal, pero la primera señal que se envía hacia la ECU se trata como señal principal y la segunda señal será manejada como una señal de verificación de datos.

La verificación de datos se utiliza en sensores de gran importancia para el control del vehículo, como el APP, este en su interior consta de dos sensores para más confiabilidad del tipo potenciómetro los cuales tendrán una alimentación mediante la ECU, la cantidad de pines implementados en el sensor pueden ser hasta de 6 y su función primordial es informar a la computadora cual es la necesidad del usuario, si quiere acelerar el vehículo o desacelerar, tiene una alimentación que puede depender del fabricante si es de 12 voltios o de 5, una tierra electrónica proveniente de la computadora y dos señales, la principal y la de verificación.

3.4.1 *Fórmula algebraica Sensor APP*

El valor del PID de este sensor corresponde al 329, de dónde solamente se adquiere el séptimo valor y se divide para 2.55.

Tabla 9

PID Obtenido del Sensor APP y fórmula algebraica.

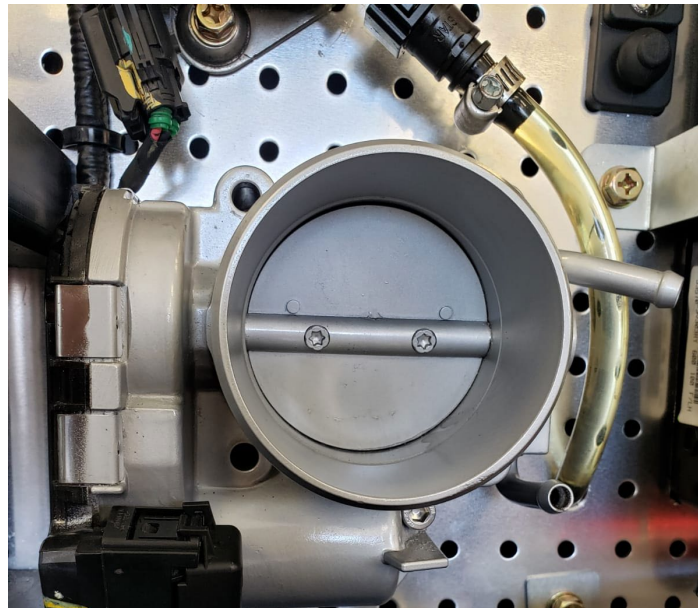
PID	Data Byte ubicación	Fórmula
329	Columna 7	$0 + \frac{1}{2.55} * (\text{Valor byte HEX} \rightarrow \text{DEC})$
		Respuesta en porcentaje [%]

Nota: Fórmula obtenida de (CSS Electronics, 2023b)

3.5 Sensor de apertura del cuerpo de aceleración (TPS)

Figura 25

Sensor TPS del Banco de Sensores, Actuadores y Unidades de Control



Fuente: (Autores)

El sensor TPS (sensor de posición del acelerador) es un componente utilizado en el sistema de control del acelerador del automóvil. Su función principal es medir la posición del acelerador, de acuerdo que tan abierto o cerrado está el acelerador. El sensor TPS generalmente se encuentra en el cuerpo del acelerador y está conectado directamente al eje de la mariposa del cuerpo de aceleración. La posición del acelerador cambia según la presión del conductor sobre el pedal del acelerador, y el sensor TPS registra estos cambios y envía la información al sistema de gestión del motor.

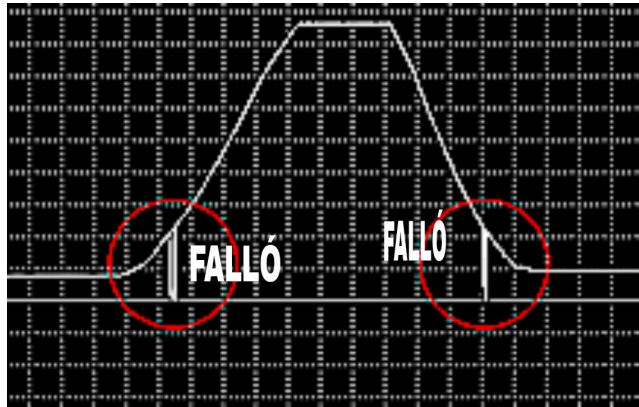
El TPS envía una señal eléctrica al sistema de gestión del motor cuyo voltaje o resistencia varía según el diseño del sensor. Esta señal es utilizada por la unidad de control del motor (ECU) para calcular la cantidad de aire que ingresa al motor y ajustar la inyección de combustible y el encendido en consecuencia. La información del sensor TPS también es utilizada por otros sistemas del vehículo, como el control de cruce, el control de tracción y el control de estabilidad, para mejorar el rendimiento y la seguridad en la conducción.

Hay que tener en cuenta que un sensor TPS dañado puede causar una serie de problemas de rendimiento del motor, como una respuesta deficiente del acelerador, dificultad para arrancar,

sacudidas o pérdida de potencia. Si sospecha que un sensor TPS está defectuoso, debe hacer que un mecánico profesional lo diagnostique y lo reemplace.

Figura 26

Sensor TPS en fallo



Fuente: (Picoauto, 2022)

El sensor TPS es un componente esencial del sistema de control del acelerador de un automóvil. Su tarea principal es medir la posición del acelerador y transmitir esta información al sistema de gestión del motor, asegurando que el motor funcione de manera correcta y eficiente.

3.5.1 Fórmula algebraica del Sensor TPS

EL sensor TPS es uno de los componentes más importantes dentro del control automotriz a inyección, de su importancia se deriva la jerarquización del PID dentro del grupo automotor.

Tabla 10

PID Obtenido del Sensor TPS y fórmula algebraica

PID	Data Byte ubicación	Fórmula
329	Columna 6	$0 + \frac{1}{2.55} * (\text{Valor byte HEX} \rightarrow \text{DEC})$
		Respuesta en porcentaje [%]

Nota: Fórmula obtenida de (CSS Electronics, 2023b)

CAPÍTULO IV

MARCO METODOLÓGICO

4.1 PRESENTACION DEL PROYECTO

En este capítulo, se presentará una visión general del desarrollo de un Dashboard automotriz, un componente esencial en la industria automotriz que proporciona información y control sobre diferentes aspectos del vehículo. Se abordarán los conceptos clave, los antecedentes y la importancia del diseño y desarrollo de Dashboards automotrices eficientes y funcionales.

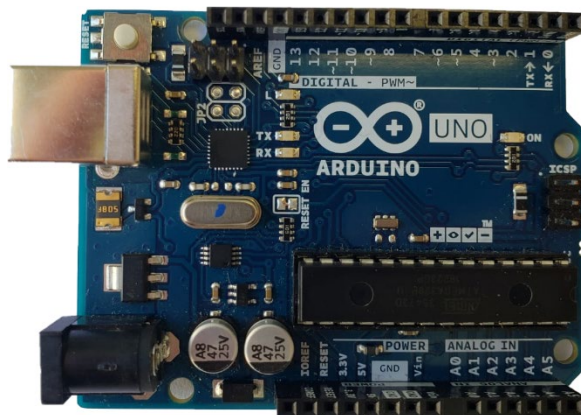
4.2 Presentación de componentes y configuraciones importantes

4.2.1 *Arduino Uno*

Arduino es una plataforma de hardware y software de código abierto diseñada para facilitar el desarrollo de proyectos electrónicos interactivos. Consiste en una placa de circuito impreso con un microcontrolador programable y un entorno de desarrollo integrado (IDE) que permite escribir y cargar programas en la placa.

Figura 27

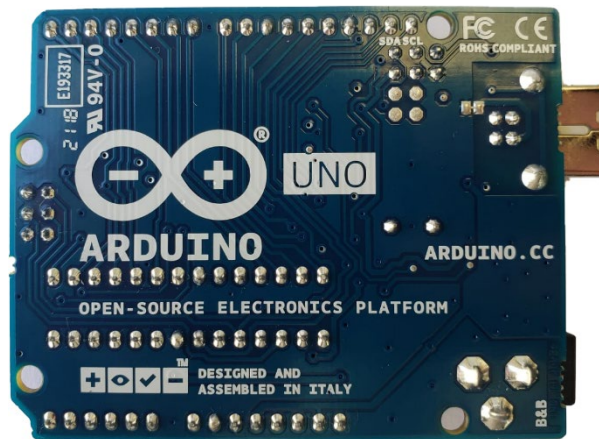
Microcontrolador Arduino Uno - Vista Superior



Fuente: (Autores)

Figura 28

Microcontrolador Arduino Uno - Vista Inferior

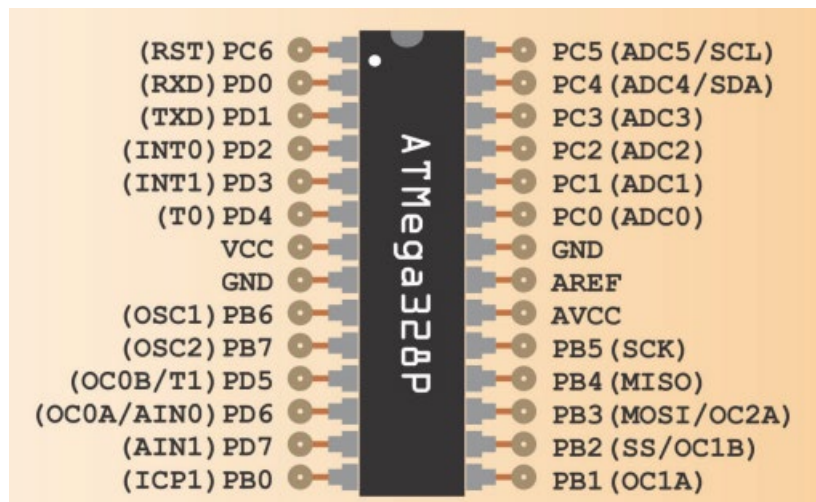


Fuente: (Autores)

Es importante explicar un poco del microchip que contiene la placa de Arduino Uno en su interior, el microchip ATmega 328p se encuentra colocado en la mayoría de las placas Arduino y es el encargado de realizar todas las acciones que el código exige.

Figura 29

Microchip ATmega 328p

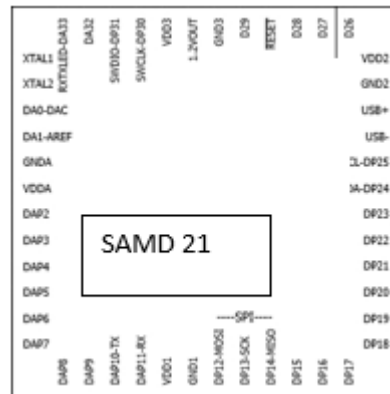


Fuente: (Espinosa, 2021)

Es indispensable darle crédito también a la memoria incorporada dentro del módulo Arduino, ya que es la encargada de guardar todo el código de programación y repetirlo las veces que el usuario crea conveniente, aunque la memoria de almacenamiento de la memoria es de 32 kb es suficiente para guardar los códigos que el usuario crea conveniente.

Figura 30

Memoria del microcontrolador Arduino Uno



Fuente: (Espinosa, 2021)

La placa de programación Arduino, consta de un proceso dentro del código de programación del mismo, a continuación, se describirá el proceso básico para que la tarjeta pueda ser programada y realizar los procedimientos que uno requiere o busca.

4.2.2 Configuración de variables

Cómo primer punto debemos configurar las variables, librerías que vamos a usar en nuestro programa, en este apartado se definen los puertos de conexión y las configuraciones que los puertos de conexión deben de tener, para poder configurar todos estos parámetros se debe de hacer dentro del apartado (void.setup).

Figura 31

Función void setup de Arduino

```
void setup()
```

Fuente: (Arduino.cc, 2023)

Una vez configurado el bloque (void setup), se debe generar un apartado llamado (void loop), el cual cumplirá la función de indicarle a Arduino que el apartado de sentencias va a iniciar, desde aquí se comienza a desarrollar la configuración de las acciones que van a realizar las sentencias, sin olvidar que para generar un código de programación se debe tener orden, ya que las acciones que el módulo Arduino va a ejecutar van a ser de acuerdo al orden en que sean escritas las líneas del código.

Figura 32

Función void loop de Arduino

```
void loop()
```

Fuente: (Arduino.cc, 2023)

4.2.3 Puertos RXD y TXD

Los puertos RXD y TXD del módulo Arduino son puertos de comunicación entre módulos, es similar al CAN BUS dentro de un vehículo, ya que estas conexiones sirven para enviar y recibir datos de cualquier tipo de una manera eficiente.

4.2.4 Puerto RXD

El puerto RXD es mejor conocido como RX0, sus siglas corresponden a Serial Receive Data, este puerto es el encargado de recibir datos, en muchas ocasiones se encuentra conectado al conector USB de Arduino, ya que es usado para enviar información al microcontrolador de Arduino.

Una vez configurado y emparejado con otro dispositivo Bluetooth, el HC-05 puede transmitir y recibir datos de manera inalámbrica a través de la conexión serial. Esto permite, por ejemplo, controlar dispositivos a distancia, transferir información o establecer una comunicación bidireccional entre dos dispositivos.

4.2.5 Puerto TXD

El puerto TXD se encarga de la transmisión de datos del tipo serial hacia otros dispositivos, se lo puede encontrar en el Arduino Uno precisamente en el pin digital 1 del Microchip ATmega 328P. Dicho puerto al encontrarse aliado al hardware de comunicación del microcontrolador Arduino, se puede hacer uso del enlace de comunicación con otros dispositivos, que incluyen Arduinos, módulos, computadoras, etc.

4.2.6 Módulo Bluetooth HC-05

El HC-05 es un módulo de comunicación Bluetooth que se utiliza comúnmente para habilitar la comunicación inalámbrica en proyectos electrónicos. El módulo HC-05 se basa en el estándar Bluetooth 2.0 y utiliza el perfil Serial Port Profile (SPP), lo que lo hace ideal para establecer una conexión serial inalámbrica entre dispositivos.

Figura 33

Módulo Bluetooth (Hc - 05) Vista Superior

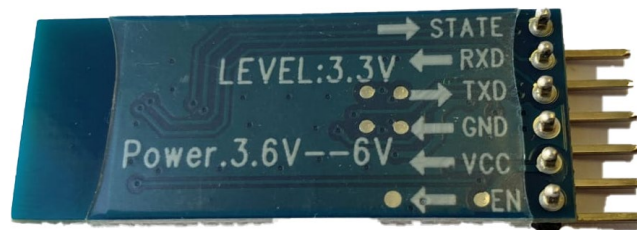


Fuente: (Autores)

El HC-05 se presenta en forma de un pequeño módulo con pines de conexión, lo que facilita su integración en proyectos electrónicos. Los pines generalmente incluyen la alimentación (VCC y GND), los pines de comunicación serial (RXD y TXD), y algunos pines adicionales para configuración, alimentación y control.

Figura 34

Módulo Bluetooth (Hc - 05) Vista Inferior



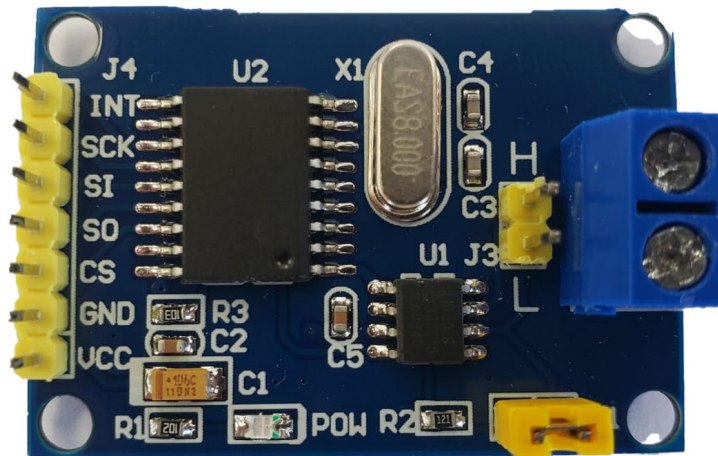
Fuente: (Autores)

4.2.7 Módulo CAN MCP 2515

El MCP2515 es un controlador de comunicación CAN (Controller Area Network) desarrollado por Microchip Technology. Es ampliamente utilizado como interfaz entre un microcontrolador y una red CAN, permitiendo la comunicación y el intercambio de datos en tiempo real en aplicaciones industriales, automotrices y de automatización.

Figura 35

Módulo CAN MCP 2515 (Vista Superior)



Fuente: (Autores)

El MCP2515 está diseñado para cumplir con la especificación CAN 2.0B, que es un protocolo de comunicación robusto y confiable utilizado en sistemas distribuidos. Este controlador permite a un microcontrolador enviar y recibir mensajes en una red CAN, actuando como un intermediario entre el microcontrolador y la red.

4.3 Indicadores y medidores

Los indicadores y medidores automotrices son aquellos que nos muestran datos en vivo de ciertos componentes del vehículo, como rpm de motor, temperatura del motor, etc. Estos indicadores nos muestran datos de distintas maneras, sin embargo, existen varios modos de presentar los datos del vehículo, en el mercado automotor existen en general dos maneras de mostrar estos datos, una forma analógica de resultados y la otra es en forma digital a continuación se abordarán explicaciones sobre ambos sistemas.

4.3.1 Analógico

En el contexto de la electrónica y las señales, "analógico" se refiere a una forma de representación o transmisión de información que es continua y varía en un rango completo de valores.

En un sistema analógico, la información se puede representar utilizando una escala continua de valores, como voltajes, corrientes o frecuencias. Estos valores analógicos pueden tomar

cualquier valor dentro de un rango determinado, lo que permite una representación precisa de la magnitud de una señal o una cantidad física.

A diferencia de los sistemas digitales, que trabajan con señales discretas y valores binarios, los sistemas analógicos operan con señales continuas en el tiempo y la amplitud. Esto significa que pueden tomar cualquier valor dentro de un rango determinado en lugar de estar limitados a valores específicos.

Los indicadores analógicos nacieron en el siglo XVIII, uno de sus creadores fue André-Marie Ampère, dice que la fuerza electromotriz es un resultado de dos efectos, la tensión y la corriente eléctrica, teniendo como resultado que si ambos fluyen en el mismo sentido o dirección al interactuar con componentes magnéticos se atraen hacia los mismos y cuando la corriente eléctrica fluye en sentido contrario se repelen estos componentes (Alvarado & Resendiz, 2018).

Figura 36

Indicadores Analógicos presentes en el tablero del Banco de Sensores, Actuadores y Unidades de Control



Fuente: (Autores)

4.3.2 Digital

"Digital" se refiere a una forma de representación y transmisión de información que utiliza valores discretos o niveles distintos. A diferencia de la representación analógica, donde la información se representa de manera continua en un rango completo de valores, la información digital se representa utilizando dígitos o bits que pueden tomar solo dos valores: 0 y 1.

El sistema digital se basa en el uso de la lógica binaria, donde cada valor o señal se representa utilizando combinaciones de ceros (0) y unos (1). Estos dígitos binarios se utilizan para representar información codificada, como números, texto, imágenes, sonidos y cualquier otra forma de datos.

Se basan en el uso de bits, que son unidades básicas de información que pueden representar los valores 0 o 1. Estos sistemas se utilizan en una amplia gama de aplicaciones, desde dispositivos electrónicos personales hasta sistemas de comunicación y control industrial.

La representación digital tiene varias ventajas sobre la analógica, como una mayor precisión, facilidad de almacenamiento, capacidad para procesar y manipular información de manera más eficiente, y una mayor resistencia al ruido y las interferencias.

Una de las características clave de los sistemas digitales es su capacidad para procesar y almacenar información de manera eficiente. Los datos se representan en forma binaria, lo que permite un procesamiento más rápido y una mayor capacidad de almacenamiento en comparación con los sistemas analógicos.

4.3.3 Indicadores Digitales

Los indicadores digitales nacieron en la década de 1960, la empresa IBM fue la encargada de crear el primer indicador digital, el cual solamente tenía la capacidad de mostrar un máximo de siete valores digitales, desde ese entonces se ha desarrollado esta tecnología para poder ser utilizada en varios campos de las industrias, estos indicadores digitales trabajan con bytes y software de programación, desde su creación se han desarrollado varios modelos diferentes de muestreo de datos, como el muestreo por puntos, barras o segmentos, cabe recalcar que de aquí se comienzan a desarrollar las nuevas tecnologías como la creación de pantallas LED, OLED y LCD.

Figura 37

Indicadores digitales presentes en el tablero del Banco de Sensores, Actuadores y Unidades de Control



Fuente: (Autores)

Tabla 11

Tabla de costos generales del proyecto

Número	Componentes	Cantidad	Precio Unitario
1	Arduino (UNO)	1	\$15
2	MCP 2515	1	\$ 7
3	HC 05	1	\$ 7
4	Conector OBDII	1	\$ 10
TOTAL		4	\$39

Fuente: (Autores)

CAPÍTULO V

MÉTODO Y DESARROLLO DEL PROYECTO

En este capítulo se llevará a cabo las metodologías de investigación ocupadas para desarrollar el dispositivo propuesto para cumplir con los objetivos antes planteados del proyecto técnico. Una vez expuestos las metodologías empleadas, se analizará de manera detallada los pasos a seguir que se cumplió para desarrollar el proyecto. A partir de los conocimientos previos de programación, electrónica, autotróica, y electricidad automotriz será posible el desarrollo del dispositivo final empleado además de la programación utilizada y la aplicación que se llevará a cabo para visualizar los datos a tiempo real del Banco de Sensores Actuadores y Unidades Electrónicas.

Enfoque Investigativo

El proyecto tiene gran relevancia académica altamente significativa en el estudio referente a la Ingeniería Automotriz y campos que se relacionen, al analizar la transición de sistemas analógicos a digitales implicando un mejor desempeño en la industria automotriz. Este enfoque implicado la comprensión de la Red Can Bus y una visión del diseño para el desarrollo del Dashboard. Esto ayuda a generar mejores proyectos, así mismo como mejorar el desempeño de los vehículos por medio del monitoreo preciso que se ven implicados los sensores, actuadores y unidades de control finalmente la implementación de sistemas digitales posibilita corregir de manera más sencilla varias problemáticas que se pueden presentar dentro del grupo automotriz ya que aplicaría dentro de un sistema factible de realizar diagnósticos.

Tipo de Investigación

Metodología analítica:

La destreza analítica se basa en lograr comprender y gestionar los datos transferidos de la red CAN BUS que se comunican entre módulos de manera hexadecimal, así también como analizar el ID de cada dato para identificar el estado de los sensores automotrices y lograr interpretar su estado mediante códigos que incorporan numerosas variables convertidas según la necesidad del uso informático que se requiera para el funcionamiento y visualización en el Dashboard.

Una vez diseñado el prototipo es esencial desarrollar la programación pertinente haciendo hincapié en los parámetros a obtener, lo que nos permitirá recibir datos a tiempo real.

Convertir los datos recibidos por la red CAN BUS, para ello se hará uso de fórmulas matemáticas recopiladas de estudios previos, las cuales nos ayudarán a transformar datos hexadecimales a decimal para posteriormente presentarlos mediante una serie de algoritmos hacia a la pantalla de visualización de datos.

Metodología experimental:

Este tipo de metodología se lleva a cabo mediante el diseño del prototipo a realizar, en esto también se involucra las conexiones entre el módulo electrónico y el banco de sensores, actuadores y unidades de control, los datos que van a ser recopilados para posteriormente ser enviados a un aplicativo el cual ilustrará los datos obtenidos del banco de pruebas.

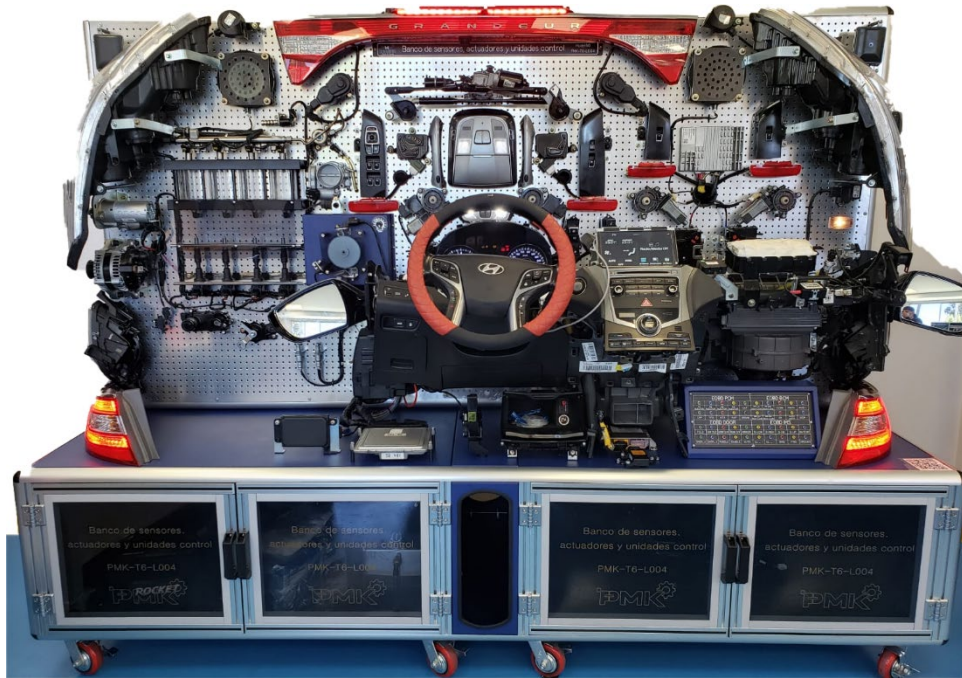
Tomando como referencia experimental mediante una prueba preliminar por medio del scanner automotriz se logró acceder a las variables de obtención que se pretende examinar, entre estos datos (Presión, Temperatura, TPS, RPM) y datos tentativos como (Nivel del combustible, Temperatura del anticongelante, Temperatura del aire de admisión, etc..) para posteriormente visualizarlo en el Dashboard.

5.1 Presentación de la maqueta base del proyecto

La maqueta la cual simula el comportamiento propio de un vehículo, en la que se realizó la obtención de todos los datos implicados en el proyecto se encuentra en el Laboratorio de Inyección de la Carrera de Ingeniería Automotriz de la Universidad Politécnica Salesiana, posee los elementos reales eléctricos, electrónicos y mecánicos necesarios para llevar a cabo la elaboración del Dashboard mediante red Can Bus, a continuación, se aprecia la parte frontal de la misma.

Figura 38

*Banco De Sensores, Actuadores Y Unidades De Control, Modelo PMK – TL – L004
“Hyundai Grandeur”*



Fuente: (Autores)

5.2 Diseño del Hardware y Software para la comunicación

En este apartado se explica la elaboración del dispositivo elaborado, con la asistencia de los microcontroladores y módulos para la placa de microcontrolador Arduino Uno que funcionarán acoplados mediante la asistencia de un protoboard. Haciendo énfasis en la previa investigación del funcionamiento de cada componente y en la comprobación independiente de cada uno, antes de ser acoplados en la placa de inserción.

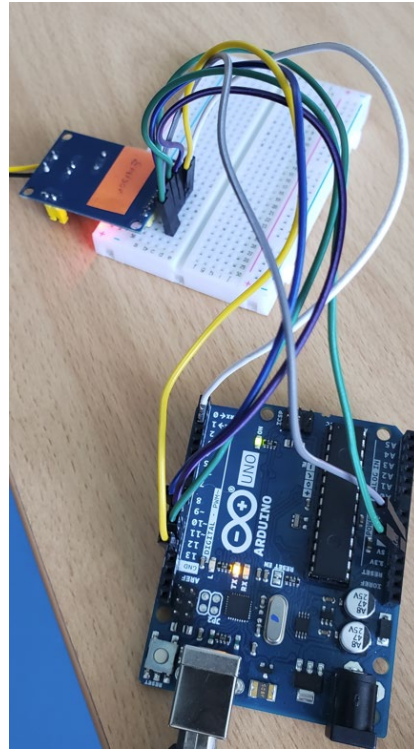
5.2.1 MCP2515 y Arduino Uno

El módulo CAN MCP2515 será un dispositivo de vital importancia pues nos permitirá conectar y comunicar a la placa Arduino Uno hacia la Red Can Bus además de ser posible adquirir datos de los nodos. Arduino Uno desempeñará el papel de interfaz entre el microcontrolador y el resto del sistema que logrará la transmisión y obtención de los datos necesarios recopilados de la red Can Bus.

Al lograr la obtención de los datos Arduino Uno será el responsable de recibir y analizar los datos transmitidos que mediante la programación apropiada se podrá procesar los datos y poder visualizarlos en un Monitor Serial en el Software propio de la marca la aplicación “Arduino”.

Figura 39

Módulo CAN MCP2515 y Arduino Uno.



Fuente: (Autores)

5.2.2 Código para la comunicación CAN Bus con Arduino

Figura 40

Extensión del código total 1

```
#include <mcp_can.h>

const int spiCS = 10; // El pin CS del mcp2515 está conectado al pin 10 de Arduino
MCP_CAN can(spiCS); // Crear un objeto MCP_CAN con el pin CS especificado

void setup() {
  Serial.begin(9600);

  if (can.begin(MCP_ANY, CAN_500KBPS, MCP_8MHZ) == CAN_OK) { // Iniciar la comunicación con el mcp2515
    Serial.println("MCP2515 iniciado correctamente!");
    can.setMode(MCP_NORMAL); // Configurar el modo de operación del mcp2515
  } else {
    Serial.println("Error al iniciar el MCP2515...");
  }
}

void loop() {
  unsigned long canId;
  unsigned char len;
  unsigned char buf[8];

  if (can.checkReceive()) { // Comprobar si hay un mensaje en el buffer de recepción
    if (can.readMsgBuf(&canId, &len, buf) == CAN_OK) { // Leer el mensaje del buffer de recepción
      //Serial.print("ID del mensaje: 0x");
      if (buf[0] == 0xA0) {
        Serial.print(canId, HEX);
        //Serial.print(" Longitud del mensaje: ");
        //Serial.print(len);
        //Serial.print(" Mensaje: ");
        Serial.print(",");
        for (int i = 0; i < len; i++) {
          Serial.print(buf[i], HEX);
          Serial.print(" ");
          Serial.print(",");
        }

        Serial.println();
      }
    }
  }
}
```

Fuente: (Autores)

5.2.3 *Detalle del código 1.*

- Se inicia por incluir la biblioteca ‘mcp_can.h’ para proporcionar las funciones fundamentales para lograr la comunicación con el módulo Can MCP2515.
- Declarar una constante ‘spiCS’ que especificará el pin CS (Chip Select) conforme al MCP2515 que en nuestro caso se encuentra conectado al pin 10 del microcontrolador Arduino uno.
- Crear un objeto ‘MCP_CAN’ llamado ‘can’ haciendo uso del pin CS especificado anteriormente.
- Mediante la función ‘setup()’, se empezará la comunicación a una velocidad agregada de 9600 baudios y se adiciona la configuración inicial del módulo MCP2515.
- La función ‘Serial.begin (9600)’ empieza la comunicación serial a la velocidad nombrada de 9600 baudios. Mediante esta estructura se logrará el envío y recibimiento de los datos desde y hacia la computadora mediante el puerto serie.
- Mediante (can.begin(MCP_ANY, CAN_500KBPS, MCP_8MHZ) inicia la comunicación con el MCP2515. Especificando una tasa de baudios o baudrate de 500 kbps y la frecuencia del oscilador del MCP215 de 8 MHz.
- Si el proceso es exitoso se imprimirá un mensaje indicando que el MCP2515 ha iniciado correctamente. De no ser así, muestra un mensaje de error.
- ‘can.setMode(MCP_NORMAL)’ determina el modo de operación en el que trabaja el MCP2515 en este caso Normal. Permitirá enviar y recibir mensajes CAN.
- Por medio de la función ‘loop()’ se lleva a cabo la recepción de mensajes CAN para luego poder imprimir los datos obtenidos por el puerto serie.
- ‘(can.checkReceive())’ se encarga de comprobar si hay algún mensaje en el buffer de recepción del MCP2515. Si hay un mensaje disponible, el código que se encuentra dentro del condicional se ejecuta.
- ‘(can.readMsgBuf(&canId, &len, buf)’ Se encarga de leer el mensaje del buffer de recepción y lo almacena en las variables ‘canId’ (ID del mensaje), ‘len’ (longitud del mensaje) y ‘buf’ (datos del mensaje).
- Dentro del bloque condicional, se verifica si el 1er byte del mensaje ‘buf[0]’ es igual a ‘0xA0’. En el caso de ser así, se procede a ejecutar el código dentro del bloque.
- ‘Serial.print(canId, HEX)’ se encarga de imprimir el ID del mensaje en formato hexadecimal. El parámetro ‘HEX’ indica que debe imprimir en base hexadecimal.

- ‘Serial.print(",")’ permite imprimir una coma (‘,’) posteriormente a imprimir el ID del mensaje
- El bucle ‘for’ que recorre los bytes del mensaje. Empieza en ‘i=0’ y se lleva a cabo hasta que ‘i’ sea menor que ‘len’ (la longitud del mensaje).
- Finalmente ‘Serial.println()’ imprime una nueva línea en el puerto serial, lo que hace que la siguiente impresión se pueda presentar en una línea nueva.

5.2.4 Visualización de datos entre Arduino y Computador

Una vez realizado el enlazamiento de comunicación con la Red Can Bus y recibir y procesar los datos por medio del microcontrolador Arduino Uno, será necesaria una previa visualización de los datos transferidos mediante la conexión del puerto serial hacia una computadora, Arduino Uno en la aplicación propia de la marca se encargará de presentar los datos a tiempo real haciendo uso de la opción Monitor Serial.

A continuación, se presenta una parte de la base de datos recibida a tiempo real:

Figura 41

Monitor Serial de la aplicación Arduino posterior a ejecutar el código 1

```

COM4
329,B2,5F,59,16,12,20,0,18,
18E,FA,FF,0,0,0,90,0,0,
5B5,0,C8,B8,4B,6A,0,0,0,
545,33,0,0,3B,0,0,0,0,
420,6,30,FF,E3,7F,1E,40,0,
329,40,5F,59,16,12,20,0,18,
A0,0,3F,0,0,0,25,0,0,
18E,FA,FF,0,0,0,90,0,0,
18E,FA,FF,0,0,0,90,0,0,
2B0,B2,FF,0,7,E,
18E,FA,FF,0,0,0,90,0,0,
2B0,B2,FF,0,7,A4,
A0,0,3F,0,0,0,25,0,0,
329,F,5F,59,16,12,20,0,18,
2B0,B1,FF,0,7,D,
2A0,4E,0,5E,8,9E,F,49,6,
4F2,28,0,52,A0,96,BC,16,E9,

```

Fuente: (Arduino.cc, 2023)

Tabla 12

Tabla importada en formato CSV de un tramo de la base de datos mediante el complemento de Transmisor de Datos para Excel

Columna1	Columna2	Columna3	Columna4	Columna5	Columna6	Columna7	Columna8	Columna9
43F	14	65	61	FF	0	0	0	0
A0	0	4E	0	0	0	25	0	0
390	3	23	EA	0	1A	0	0	2A
43F	14	65	61	FF	0	0	0	0
545	32	0	0	74	0	0	0	0
43F	14	65	61	FF	0	0	0	0
2A0	E	0	5D	0	89	F	49	6
43F	14	65	61	FF	0	0	0	0
260	1E	35	FF	0	2B	A6	64	31

Fuente: (Autores)

5.2.5 Comprobación base de datos obtenidos

La obtención de los datos es fundamental en el proyecto actual es por ello que los datos obtenidos anteriormente deben ser los precisos y no valores indiferentes a lo esperado, de ahí parte la necesidad de comprobar que los datos obtenidos no son inexactos y que la programación implementada anteriormente funciona correctamente recordemos que estos valores son enviados desde la Red Can Bus del Banco de Sensores y Arduino Uno se encargará de procesar los datos.

A través de un adaptador con transceptor LIN y CAN que registrará los datos del Banco de Sensores, Actuadores y Unidades de Control para poder observarlos en la aplicación CAN-Bus Monitor. La aplicación permite observar y registrar todos los PID's que son activados y varían conforme el usuario manipula el Banco.

Figura 42

Transceptor individual LIN y CAN parte del frente y parte posterior



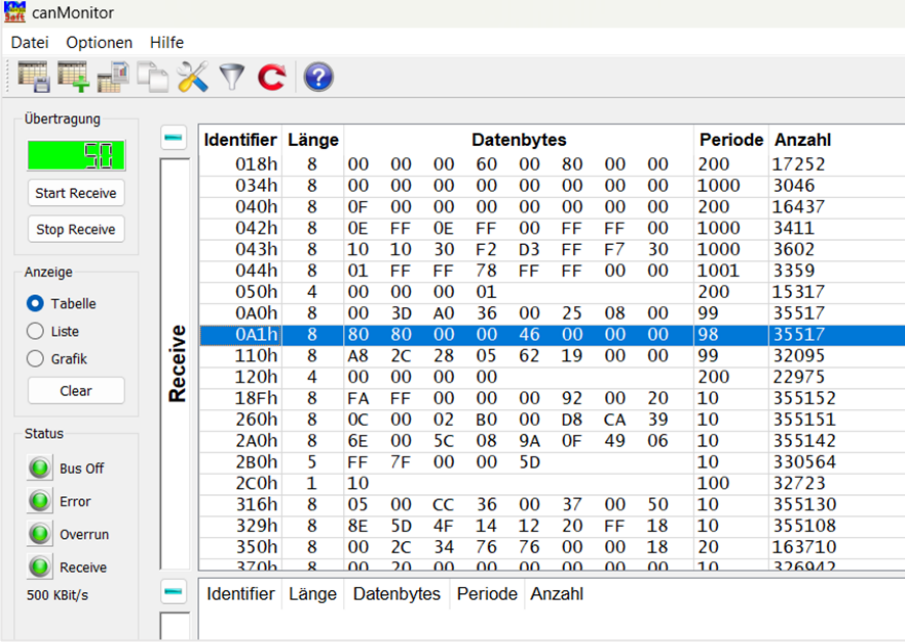
Fuente: (Autores)

Por medio del software KMTEC Soft una aplicación alemana para ingeniería, en el caso presente, para soluciones de conectividad y comunicación relacionado con CAN BUS, Ethernet y otras interfaces de comunicación que se utilizan en la industria automotriz, se logró analizar los sistemas de bus de datos exactos proporcionados por la Red Can Bus como una alternativa exacta de comprobación de la base de datos.

Una vez realizada la conexión desde el DLC OBDII del Banco de Sensores, se procede a acoplar el terminal del Transceptor LIN-CAN y por último la conexión USB hacia el terminal del Puerto USB de la computadora en donde vayamos a visualizar desde la aplicación CAN-Bus Monitor previamente ya instalada, se procede a revisar que todo esté correctamente acoplado y en dentro de la aplicación hay que verificar que todos los parámetros del Status se encuentre en color verde como señal que se encuentra sin errores y podrá transferir los datos, procedemos a dar clic en Star Receive y se logrará visualizar la transferencia de datos en tiempo real, la misma base de datos que se debería reflejar en el Monitor Serial de la aplicación Arduino Uno pero en este caso mucho más distribuido y explicado cada uno de los datos como los Identificadores de cada PID como se puede apreciar en la Figura 40 se puede apreciar 50 PID's que se encuentran en activación desde el Banco de Sensores, también se puede apreciar la longitud de los Data Bytes que servirán para identificar la activación de las variables requeridas (RPM, APP, Presión, TPS, Nivel de anticongelante).

Figura 43

Aplicación CAN-Bus Monitor en desarrollo



The screenshot shows the 'canMonitor' application window. It features a menu bar with 'Datei', 'Optionen', and 'Hilfe'. Below the menu is a toolbar with icons for file operations and help. The main interface is divided into several sections:

- Übertragung (Transmission):** Includes a green status indicator, 'Start Receive', and 'Stop Receive' buttons.
- Anzeige (Display):** Offers radio buttons for 'Tabelle' (selected), 'Liste', and 'Grafik', along with a 'Clear' button.
- Status:** Contains status indicators for 'Bus Off', 'Error', 'Overrun', and 'Receive', with a speed of '500 kBit/s'.
- Receive Table:** A table with columns for Identifier, Länge, Datenbytes, Periode, and Anzahl. The table contains 20 rows of data, with the row for Identifier '0A1h' highlighted in blue.

Identifier	Länge	Datenbytes	Periode	Anzahl
018h	8	00 00 00 60 00 80 00 00	200	17252
034h	8	00 00 00 00 00 00 00 00	1000	3046
040h	8	0F 00 00 00 00 00 00 00	200	16437
042h	8	0E FF 0E FF 00 FF FF 00	1000	3411
043h	8	10 10 30 F2 D3 FF F7 30	1000	3602
044h	8	01 FF FF 78 FF FF 00 00	1001	3359
050h	4	00 00 00 01	200	15317
0A0h	8	00 3D A0 36 00 25 08 00	99	35517
0A1h	8	80 80 00 00 46 00 00 00	98	35517
110h	8	A8 2C 28 05 62 19 00 00	99	32095
120h	4	00 00 00 00	200	22975
18Fh	8	FA FF 00 00 00 92 00 20	10	355152
260h	8	0C 00 02 B0 00 D8 CA 39	10	355151
2A0h	8	6E 00 5C 08 9A 0F 49 06	10	355142
280h	5	FF 7F 00 00 5D	10	330564
2C0h	1	10	100	32723
316h	8	05 00 CC 36 00 37 00 50	10	355130
329h	8	8E 5D 4F 14 12 20 FF 18	10	355108
350h	8	00 2C 34 76 76 00 00 18	20	163710
370h	8	00 20 00 00 00 00 00 10	10	326942

Fuente: (KMTecSoft, 2017)

5.2.6 Comprobación datos obtenidos a tiempo real comparado mediante Scanner Automotriz

Además de los datos reales que se obtuvieron y de comprobar visualmente en el tablero funcional del Banco de Sensores, Actuadores y Unidades de Control y demás componentes mecánicos de los Sensores, los cuales su activación se puede variar mediante manipulación podemos tener una idea del valor que debe obtenerse (movimiento en perilla CKP, presión en el pedal con APP, variación en la temperatura del ECT), es necesario la verificación oficial de los valores mediante un Scanner Automotriz el cuál con la ayuda de un Cable divisor OBD II 1macho y 2 adaptadores de cable puerto J1962 hembra.

Figura 44

Cable divisor OBD II



Fuente: (Autores)

Figura 45

Comprobación de la variable mediante Software Can Bus Monitor y CSSElectronics.

The screenshot shows the CSSElectronics software interface. On the left, there is a table for 'Engine coolant temperature' with columns for Name, Bit start, Bit Length, Scale, Offset, Min, Max, and Unit. The table shows a CAN ID of 7DF and 7E8, with a value of 30 in the Byte 3 column. On the right, there is a 'Receive' window showing a list of CAN messages with columns for Identifier, Länge, and Datenbyte. The message with Identifier 0A0h and Länge 8 is highlighted, showing a value of 3D in the Datenbyte column.

Name	Bit start	Bit Length	Scale	Offset	Min	Max	Unit
Engine coolant temperature (v)	39	8	1	-40	-40	215	degC

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	B
7DF	02	01	67	AA	AA	AA	AA	A
7E8	03	41	67	3D	AA	AA	AA	A

Identifier	Länge	Datenbyte
018h	8	20 00
034h	8	00 00
040h	8	0F 00
042h	8	00 FF
043h	8	08 00
044h	8	01 FF
050h	4	00 00
0A0h	8	00 3D
0A1h	8	80 80
110h	8	A8 2C
120h	4	00 00
18Fh	8	FA FF
260h	8	05 00
2A0h	8	4E 00
2B0h	5	FF 7F
2C0h	1	10
316h	8	05 00
329h	8	0F 5D
350h	8	00 24

Fuente: (Autores)

5.2.7 Implementación del Módulo Bluetooth HC - 05

Tras la comprobación de los datos obtenidos en el Arduino Uno por medio de las dos herramientas base (App CAN – Bus Monitor, Scanner Automotriz) podemos proceder a llevar a cabo el siguiente paso el cual es el envío de datos hacia un dispositivo remoto, para ello hemos propuesto la implementación de un módulo Bluetooth HC – 05 que irá acoplado directamente en el Arduino Uno.

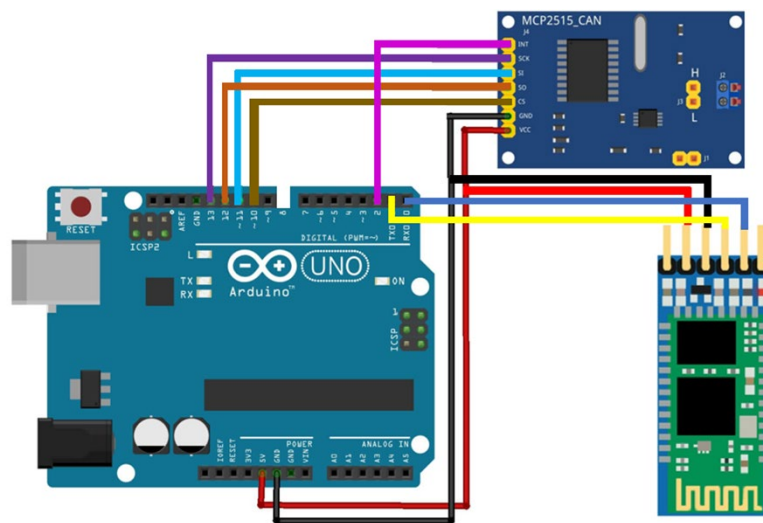
El módulo Bluetooth HC – 05 para Arduino permite enlazar una comunicación inalámbrica en este caso “Bluetooth” entre el microcontrolador Arduino Uno y otro dispositivo que sea

compatible con conexión bluetooth apto para smartphones, tablets o computadoras de acuerdo a la necesidad.

Teniendo en claro el papel que desempeña este dispositivo electrónico, se procede a acoplarlo en el Arduino Uno. La conexión es sencilla pues no requiere más que realizar el acoplamiento pertinente desde la placa Arduino Uno y el módulo CAN MCP2515.

Figura 46

Conexión Arduino Uno, Módulo CAN MCP2515 y Módulo HC – 05

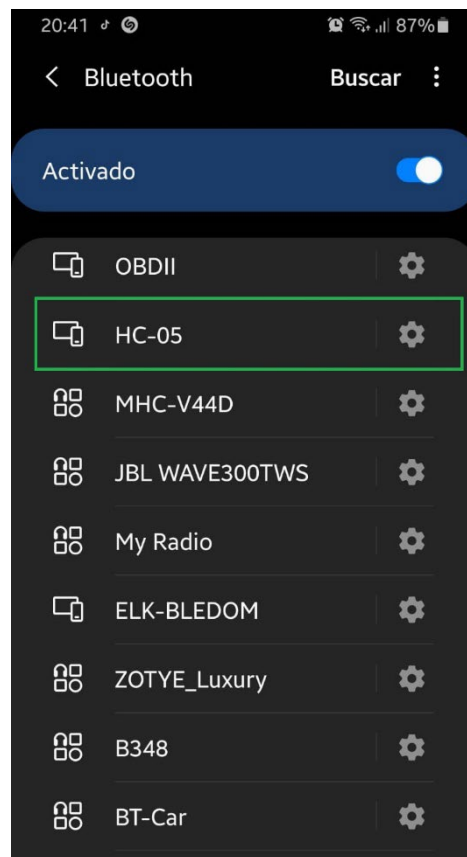


Fuente: (Autores)

Una vez obtenida la conexión de estos 3 dispositivos (Arduino Uno, Módulo CAN MCP2515 y Módulo HC - 05), se procederá a enlazar y posteriormente a realizar una verificación en el del dispositivo en que se desee llevar a cabo el emparejamiento en nuestro caso se ha utilizado un smartphone. En los ajustes de bluetooth del smartphone realizaremos la búsqueda del módulo el cual nos debería reflejarnos como la Figura 44.

Figura 47

Verificación del emparejamiento módulo HC05 – Smartphone



Fuente: (Autores)

5.2.8 Filtración de datos

Conforme se logró obtener la base de datos a tiempo real y encontrar los PID's de las variables requeridas. es necesario la filtración de los datos ya que, al recibir grandes cantidades de datos en segundos, se vuelve complicado el tratamiento de toda la base. De ahí parte la necesidad de escoger solamente las variables que utilizaremos para poder gestionar el tratamiento de sus datos para el Dashboard.

En este caso, será necesario recurrir nuevamente a modificar la programación del código 1 desde la aplicación de Arduino para lograr obtener la filtración de los datos deseados además de agregar las funciones necesarias para el envío e impresión de los datos para el módulo bluetooth y agregar la librería “SoftwareSerial.h”.

Figura 48

Dato filtrado PID A0

```
// Verificar si el PID es "A0"
if ((canId & 0x7FF) == 0xA0) {
    Serial.print("#S1,");

    // Unir los dígitos del tercer y cuarto dígito
    int value = (buf[3] << 8) | buf[2];

    // Multiplicar por 0.25
    int result = value * 0.25;
```

Fuente: (Autores)

Figura 49

Dato filtrado PID A1

```
// Verificar si el PID es "A1"
if ((canId & 0x7FF) == 0xA1) {
    Serial.print("#S3,");

    // Mostrar solo el quinto valor del PID A1
    Serial.print("01,");
    Serial.println(buf[4], DEC);
    // Después de mostrar el PID A1, mostrar el valor dividido por 2.55 del PID 329 una vez
```

Fuente: (Autores)

Figura 50

Dato filtrado PID 329.

```
// Después de mostrar el PID A1, mostrar el valor dividido por 2.55 del PID 329 una vez
if (!pid329Received) {
    // Buscar y mostrar el PID 329
    while (can.checkReceive()) {
        if (can.readMsgBuf(&canId, &len, buf) == CAN_OK && (canId & 0x7FF) == 0x329) {
            Serial.print("#S4,");

            // Dividir el valor del PID 329 entre 2.55
            int value = buf[6];
            float result = value / 2.55;
            Serial.print("01,");
            Serial.println(result);

            pid329Received = true; // Marcar que se ha recibido el PID 329
```

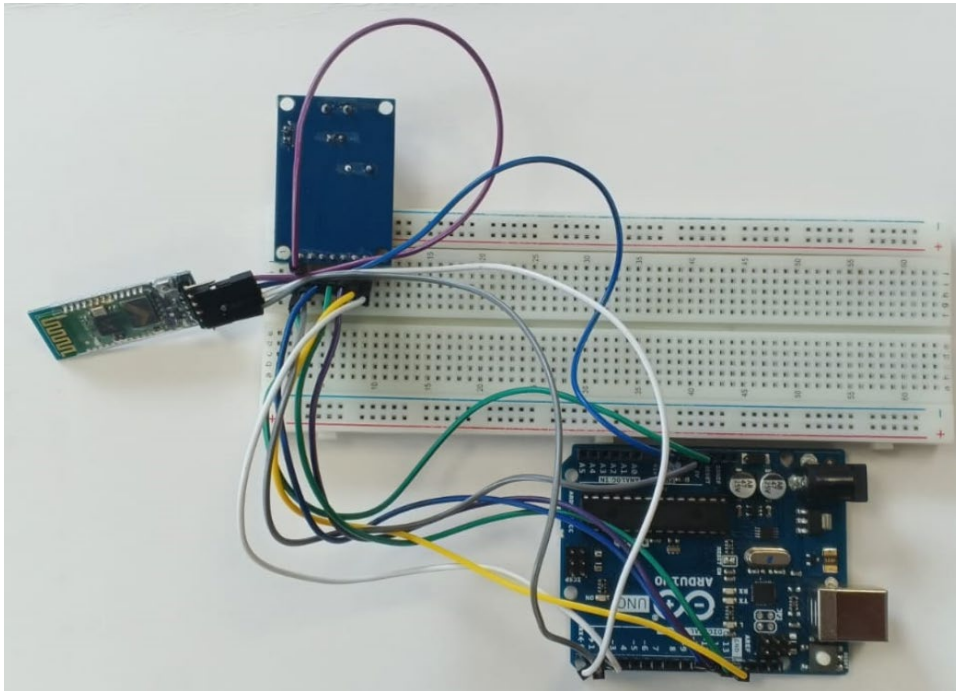
Fuente: (Autores)

5.3 Diseño Hardware Final

Una vez ensamblados todos los componentes (Arduino Uno, Módulo Can MCP2515, HC - 05) y haciendo de un protoboard, el resultado final de nuestro dispositivo electrónico para recibimiento, interpretación y envío de datos para la Red Can Bus del Banco de Sensores Actuadores y Unidades de Control queda plasmado en la Figura 48.

Figura 51

Diseño final del dispositivo



Fuente: (Autores)

5.4 Creación de la Interfaz de la aplicación para el Dashboard

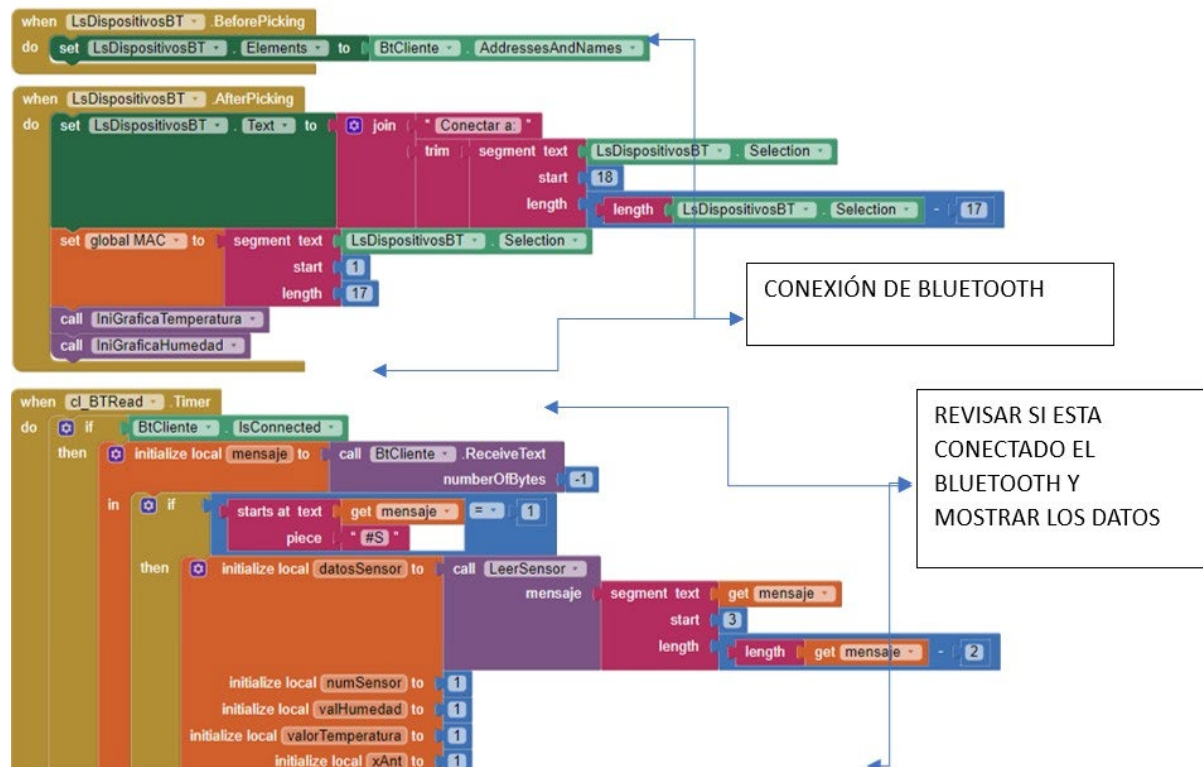
Para poder controlar y monitorear el funcionamiento del Banco de Sensores, Actuadores y Unidades de Control se requiere una aplicación en donde se pueda representar los valores enviados desde Can Bus, interpretados por el módulo CAN MCP2515 y a su vez por Arduino Uno y finalmente enviar la información por el módulo Bluetooth HC – 05.

Es por ello que el desarrollo de la aplicación para el Dashboard se vuelve esencial en el momento de poder visualizar los datos obtenidos a tiempo real. En esta aplicación de nuestra autoría hemos logrado representar de forma gráfica a manera de dato en vivo, las distintas variables.

Esta aplicación recibirá los datos ya filtrados de los PID's y mediante la programación pertinente en el entorno de programación gratuito "MIT App Inventor" el cual nos permite crear aplicaciones Android de manera bastante práctica.

Figura 52

Fragmento de la programación de la aplicación

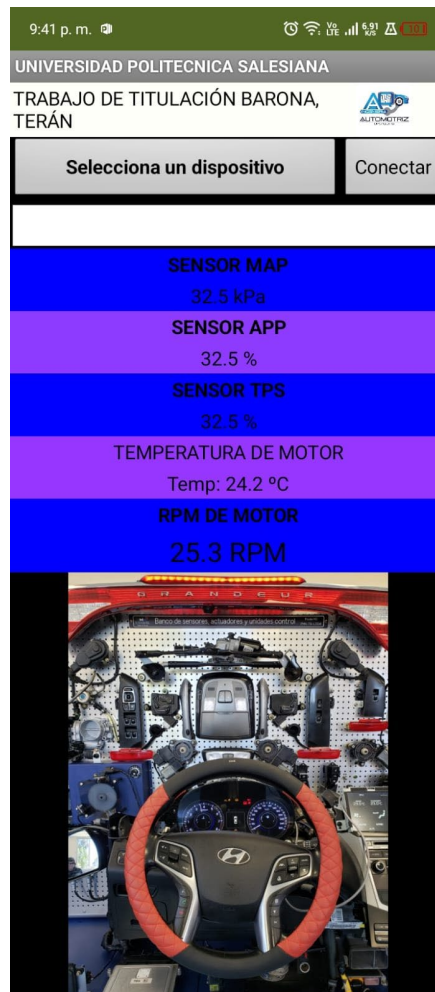


Fuente: (Autores)

- En el primer apartado tenemos la conexión bluetooth, la cual se encarga de establecer la conexión bluetooth entre el celular y la interfaz, nos establece los dispositivos bluetooth disponibles y nos permite conectarnos a uno de ellos.
- El segundo apartado es el más importante, ya que nos permite realizar la programación de la aplicación, y mostrar los datos en vivo del vehículo, ya que si el bluetooth se encuentra conectado genera listas donde guardar los datos, filtrarlos y luego mostrarlos.

Figura 53

Visualización de datos recibidos en tiempo real por Red CAN BUS del Banco de Sensores



Fuente: (Autores)

CAPÍTULO VI

Análisis e interpretación de resultados

En el último capítulo a revisar se demuestra la obtención de variables conseguidas a tiempo real haciendo uso de las fórmulas algebraicas de cada variable a obtener, además de realizar una comprobación mediante la aplicación CAN – Bus Monitor, y un Scanner Automotriz que nos asistirá en la presentación de los datos exactos del Banco de Sensores, Actuadores y Unidades de Control Electrónico.

Una vez comparados y comprobados los datos a tiempo real recibidos al Monitor Serial del microcontrolador Arduino Uno y en la Aplicación CAN – Bus Monitor se puede empezar a descartar cada PID que no resulte útil para el presente proyecto mediante la manipulación de las piezas mecánicas como (la mariposa TPS, Perilla del sensor CKP simulando las Revoluciones por minuto, el Pedal del sensor APP y la presión en el sensor MAP mediante una pistola de vacío y la asistencia de la aplicación CAN- Bus Monitor al lograr obtener visualmente la activación de la variación de los datos hexadecimales.

Al hallar el PID y el valor en el data byte en vivo como valor hexadecimal de las variables requeridas, se puede empezar a utilizar las fórmulas algebraicas analizadas en el Capítulo III para cada variable, de este modo logramos obtener el valor final a tiempo real de las variables. Para obtener la cifra exacta será necesario proceder a 3 pasos que a continuación se describirá brevemente:

1. Valor hexadecimal de la variable a tratar (tener en claro su ubicación en el data byte).
2. Transformación de del valor hexadecimal a decimal.
3. Implementar en la fórmula algebraica propia de cada variable para obtener la cifra exacta.

Analizaremos un monitoreo del Banco de Sensores en donde visualizaremos a tiempo real todas las variables planteadas y el valor a presentarse debe ser fiel o con un desfase leve de la cifra exacta para comprobar estos datos haremos uso de la App CAN-Bus Monitor en donde observaremos el PID exacto, el valor en el Data Byte y finalmente lo comprobaremos en el Scanner Automotriz.

6.1 Temperatura (Sensor ECT) PID (A0)

Figura 54

Análisis PID A0 (Temperatura ECT) desde App CAN-Bus Monitor a tiempo real

Identifizier	Länge	Datenbytes	Periode	Anzahl
000h	0		960	6
018h	8	00 00 01 60 01 80 00 00	200	1884
034h	8	00 00 00 00 00 00 00 00	999	379
040h	8	0F 00 00 00 00 00 00 00	200	1880
042h	8	00 FF 00 FF 00 FF FF 00	1000	374
043h	8	10 00 30 F4 C3 FF F7 00	999	371
044h	8	01 FF FF 78 FF FF 00 00	1001	369
050h	4	03 03 08 01	199	1895
0A0h	8	00 76 A8 10 00 44 01 00	100	3801
0A1h	8	80 80 00 00 46 00 00 00	100	3802
110h	8	A8 2C 28 05 62 19 00 00	200	3746
120h	4	00 00 00 00	200	1889
18Fh	8	FA FF 00 00 00 82 00 20	10	38064
260h	8	26 00 44 B0 00 BC 90 14	10	38034
2A0h	8	2E 00 5C 08 AB 0F 49 06	10	38013
2B0h	5	A5 F0 00 07 07	9	37897
2C0h	1	10	100	3737
316h	8	05 00 9C 10 00 27 00 5A	10	37948
329h	8	0F FF 59 15 11 20 00 18	10	37829
350h	8	00 24 24 76 77 00 00 01	20	18875

Fuente: (Autores)

Como se puede apreciar el valor hexadecimal obtenido es “76” ubicado en la 2da columna del “datenbytes” del PID A0 y mediante el uso de la fórmula algebraica para esta variable da como resultado 78 °C.

Figura 55

Análisis PID A0 (Temperatura ECT) desde Scanner Automotriz a tiempo real.

[001] Tensión Lógica de la Batería 11.8 V	[002] Voltaje de batería después de encender el... 11.8 V
[003] Velocidad real del motor 1066 RPM	[004] Meta de RPM de ralentí 1300 RPM
[005] Voltaje de señal del sensor de presión 2.8 V	[006] Presión del colector de admisión 705.6 hPa
[007] Voltaje de temperatura de agua 0.6 V	[008] Temperatura del agua 78 °C
[009] Temperatura de aire de ambiente	[010] Voltaje de temperatura de admisión de aire

Fuente: (Autores)

Se puede apreciar el dato correspondiente a 78 °C.

Figura 56

Reflejo de la temperatura del ECT a una temperatura exacta de 78 °C en la aplicación



Fuente: (Autores)

6.2 Revoluciones por minuto (Sensor CKP) PID (A0)

Figura 57

Análisis PID A0 (Temperatura -ECT) desde App CAN-Bus Monitor a tiempo real

canMonitor

Datei Optionen Hilfe

Übertragung

Start Receive

Stop Receive

Anzeige

Tabelle

Liste

Grafik

Clear

Status

Bus Off

Error

Overrun

Receive

500 Kbit/s

Identifer	Länge	Datenbytes	Periode	Anzahl
000h	0		960	6
018h	8	00 00 00 60 00 80 00 00	200	2814
034h	8	00 00 00 00 00 00 00 00	1000	566
040h	8	0F 00 00 00 00 00 00 00	200	2801
042h	8	0E FF 0E FF 00 FF FF 00	999	558
043h	8	14 40 70 F2 D3 FF F7 40	1000	558
044h	8	01 FF FF 78 FF FF 00 00	3999	533
050h	4	03 03 08 01	200	2823
0A0h	8	00 3E B0 13 00 2F 01 00	100	5685
0A1h	8	80 80 00 00 46 00 00 00	100	5686
110h	8	A8 2C 28 05 62 19 00 00	100	5586
120h	4	00 00 00 00	200	2806
18Fh	8	FA FF 00 00 00 82 00 20	11	56891
260h	8	21 00 3A B0 00 B5 93 36	11	56851
2A0h	8	6E 00 5C 08 AB 0F 49 06	11	56782
2B0h	5	01 F0 00 07 4D	10	56418
2C0h	1	10	100	5584
316h	8	05 00 B0 13 00 25 00 5A	11	56607
329h	8	8E FF 59 15 11 20 00 18	10	56295
350h	8	00 20 04 76 78 20 00 07	20	28113

Identifer Länge Datenbytes Periode Anzahl

Fuente: (Autores)

Se puede observar el valor hexadecimal obtenido es “B0 13” ubicado en la 3er y 4rta columna del “datenbytes” del PID A0 en este caso será necesario ubicar el valor 4rto en primer lugar y el 3ero en segundo lugar al unir los dos datos hexadecimales formarán uno solo “13B0” y mediante una conversión de número hexadecimal a decimal obtendremos el valor de 5040 posterior a esto se aplicará el uso de la fórmula algebraica específica para esta variable y como resultado se obtendrá 1260 rpm.

Figura 58

Análisis PID A0 (Revoluciones por minuto - CKP) desde Scanner Automotriz a tiempo real



Fuente: (Autores)

Se puede apreciar el dato correspondiente a 1260 RPM.

Figura 59

Reflejo de la temperatura de las revoluciones exactas de 1260 rpm en la aplicación



Fuente: (Autores)

6.3 Presión absoluta del múltiple (MAP) PID (A1)

Figura 60

Análisis PID A1 (Presión [kPa] - MAP) desde App CAN-Bus Monitor a tiempo real

canMonitor

Datei Optionen Hilfe

Übertragung

Start Receive

Stop Receive

Anzeige

Tabelle

Liste

Grafik

Clear

Status

Bus Off

Error

Overrun

Receive

Identifer	Länge	Datenbytes	Periode	Anzahl
000h	0		960	6
018h	8	00 00 00 60 00 80 00 00	200	4915
034h	8	00 00 00 00 00 00 00 00	999	996
040h	8	0F 00 00 00 00 00 00 00	200	4883
042h	8	00 FF 00 FF 00 FF FF 00	1000	968
043h	8	14 00 30 F4 C3 FF F7 00	1000	975
044h	8	01 FF FF 78 FF FF 00 00	1000	911
050h	4	03 03 08 01	201	4967
0A0h	8	00 3E 00 00 00 25 00 00	100	10017
0A1h	8	80 80 00 00 1E 00 00 00	100	10018
110h	8	A8 2C 28 05 62 19 00 00	101	9769
120h	4	00 00 00 00	199	4899
18Fh	8	FA FF 00 00 00 82 00 00	10	100188
260h	8	22 00 FF 20 02 A6 5F 06	9	100070
2A0h	8	6E 00 5F 08 AB 0F 49 06	11	99813
2B0h	5	00 EF 00 07 17	10	98992
2C0h	1	10	100	9721
316h	8	05 00 00 00 00 23 00 5A	10	99123
329h	8	0F FF 59 17 11 20 00 18	10	98297
350h	8	00 24 D4 76 78 00 00 FF	20	49275

Fuente: (Autores)

En esta variable el valor hexadecimal obtenido es “1E” ubicado en la 5ta columna del “datenbytes” del PID A1 y mediante el uso de la fórmula algebraica para esta variable da como resultado 30 kPa.

Figura 61

Análisis PID A1 (Presión absoluta del múltiple - MAP) desde Scanner Automotriz a tiempo real



Fuente: (Autores)

Se puede apreciar la variable obtenida es 301 Hectopascasles [hPa] que transformada a kPa es igual a un valor de 30 kPa.

Figura 62

Reflejo de la presión absoluta del múltiple exacta de 30 kPa en la aplicación



Fuente: (Autores)

6.4 Sensor de Posición del Pedal de Acelerador (APP) PID (329)

Figura 63

Análisis PID 329 (Posición del pedal de aceleración [%] - APP) desde App CAN-Bus Monitor a tiempo real

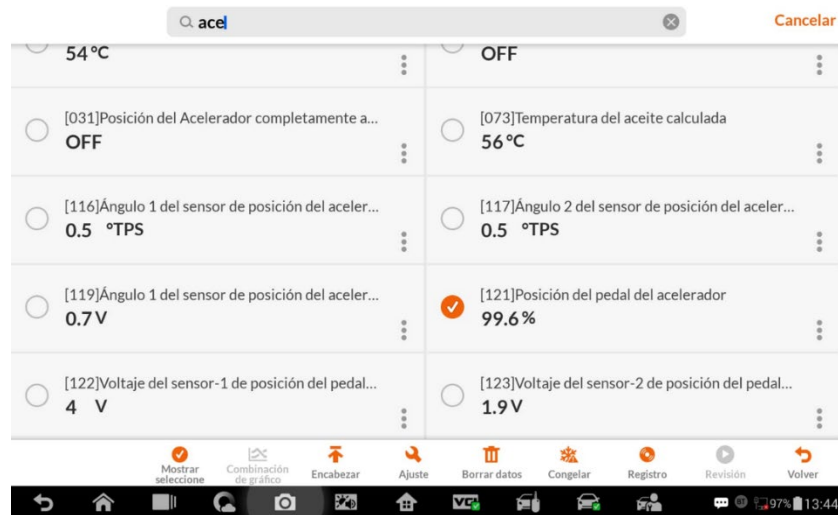
Identifler	Länge	Datenbytes	Periode	Anzahl
043h	8	14 00 30 F4 C3 FF F7 00	999	1192
044h	8	01 FF FF 78 FF FF 00 00	1999	1119
050h	4	03 03 08 01	201	6085
0A0h	8	00 3E 00 00 00 25 00 00	100	12306
0A1h	8	80 80 00 00 46 00 00 00	100	12307
110h	8	A8 2C 28 05 62 19 00 00	101	11973
120h	4	00 00 00 00	200	6006
18Fh	8	FA FF 00 00 00 82 00 00	10	123079
260h	8	22 00 B6 20 06 A6 5F 4B	10	122957
2A0h	8	0E 00 5E 08 AB 0F 49 06	10	122687
280h	5	7B E9 00 07 E2	10	121166
2C0h	1	10	100	11887
316h	8	05 00 00 00 00 23 00 5A	10	121959
329h	8	FF 59 17 11 20 FE 18	10	121073
350h	8	00 24 74 76 78 00 00 5E	20	60445
370h	8	00 20 00 80 00 00 00 00	10	120975
382h	8	00 00 00 00 00 BF 0E E8	19	60320
390h	8	03 7B E9 00 0E 00 00 75	19	58669
392h	8	03 C2 07 7B E9 05 00 35	20	59704
420h	8	06 30 FF F3 7F 1F 40 00	20	60268

Fuente: (Autores)

En el caso actual la variable del valor hexadecimal obtenido es “1E” ubicado en la 7ma columna del “datenbytes” del PID 329 y mediante el uso de la fórmula algebraica para esta variable da como resultado 99. 61 % lo que nos indica que el Pedal se encuentra accionado hasta el final de su carrera.

Figura 64

Análisis PID 329 (Posición del pedal de aceleración [%] - APP) desde Scanner Automotriz a tiempo real



Fuente: (Autores)

Se logra comprobar el porcentaje de 99,6% valor exacto obtenido.

Figura 65

Reflejo de la posición del pedal de aceleración exacto de 99,61% en la aplicación



Fuente: (Autores)

6.5 Sensor de apertura del cuerpo de aceleración (TPS) PID (329)

Figura 66

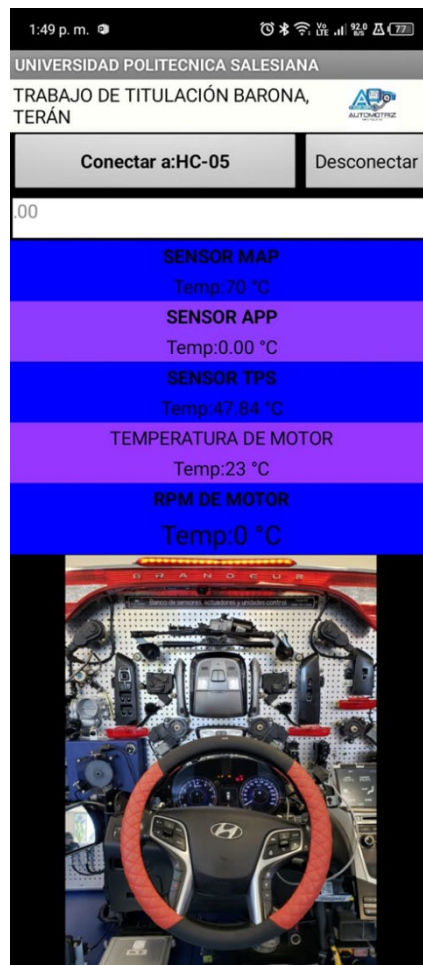
Análisis PID 329 (Sensor de apertura del cuerpo de aceleración [%] - TPS) desde App CAN-Bus Monitor a tiempo real

Fuente: (Autores)

En la última variable a analizar el valor hexadecimal obtenido es “7A” ubicado en la 6ta columna del “datenbytes” del PID 329 y mediante el uso de la fórmula algebraica para esta variable da como resultado un valor de 47.84 % lo que se puede traducir como la abertura de la mariposa a casi la mitad de su totalidad.

Figura 67

Reflejo de la posición del pedal de aceleración exacto de 47,84% en la aplicación



Fuente: (Autores)

CONCLUSIONES

- El objetivo del proyecto se alcanzó de manera exitosa pues los datos que se obtuvieron reflejan la información precisa del comportamiento de las variables propuestas del Banco de Sensores, Actuadores y Unidades de Control. Conforme a los hallazgos expuestos en el trabajo de titulación y de acuerdo a los datos conseguidos en tiempo real demostrando veracidad de los valores con respecto a información precisa de instrumentación automotriz.
- Los resultados de este estudio, logró demostrar que la tarjeta de desarrollo Arduino es una herramienta de usos innumerables, con esta se pueden crear una gran cantidad de proyectos, en la presente investigación se llevó a cabo el uso de este dispositivo y conforme a un uso adecuado con fines académicos de la misma, obteniendo una ejecución de importantes aplicativos, con ayuda de agregados como el módulo CAN MCP2515 es posible dar cumplimiento a una lectura y enlace entre módulos mediante CAN BUS, esto tiene varios propósitos, como a monitorear datos, permitir implementar módulos en el interior y obtener un control más óptimo del vehículo.
- La evidencia presentada lleva a la conclusión que la obtención de los datos de la Red Can Bus por medio del diseño del Dashboard propuesto mediante hardware accesible y la aplicación desarrollada para monitoreo de la información cumplió con los resultados esperados conforme al actual proyecto, además exhibe una solución práctica y de acceso cómodo para visualización de datos en vivo importantes del vehículo. Así mismo es de gran importancia para la implementación en estudios posteriores que se requiera o se desee desarrollar aún más las herramientas compartidas en la presente investigación.
- Luego del análisis de las fórmulas algebraicas revisadas anteriormente se puede comprobar que en el aplicativo desarrollado del proyecto, fueron bases fundamentales para la obtención de resultados efectivos pues en la comprobación de estos datos se puede poner en evidencia el equilibrio de nivel de precisión en relación a un scanner automotriz.
- Para finalizar se concluye que la comunicación de la red CAN BUS conlleva información amplia del estado del vehículo en cuestión de nanosegundos, lo que hace que sea una de las redes automotrices más seguras creadas hasta la fecha y la más precisa, ya que todos los valores que son enviados por la red no se pierden ni se distorsionan a menos que exista algún fallo dentro del sistema, se demostró que la distribución de la red Can Bus ayuda a los técnicos a reconocer los dígitos hexadecimales que corresponden cada valor, esto con ayuda de los PID, los cuales son enlistados según el fabricante lo prefiera.

RECOMENDACIONES

- Es importante reconocer desde un inicio los pines del DLC OBD II del Banco de Sensores, Actuadores y Unidades de Control y en especial el PIN 6 Can High y PIN 4 Can Low del DLC OBDII para que no exista una confusión al momento de realizar la conexión pertinente hacia el dispositivo elaborado, de igual manera se debe revisar el buen estado de los cables Dupont y acoplar debidamente hacia los distintos pines de los módulos agregados.
- El presente proyecto queda como una base para estudios posteriores ya que se puede ampliar las variables mediante los pasos que se han llevado a cabo, por medio de un análisis comprensivo de este documento se obtiene fundamentos claves para quien desee descubrir una fuerte herramienta de diagnóstico automotriz a bajo costo y alta efectividad.
- El diagnóstico dentro de un vehículo moderno, equipado con red de comunicación CAN debe realizarse usando los equipos adecuados, ya que un diagnóstico erróneo puede generar grandes gastos indebidos para el cliente y una mala reputación para el taller.
- El conocimiento del sistema de comunicación de un vehículo antes de comprarlo es indispensable, no solamente para buscar talleres autorizados para realizar reparaciones en el mismo, sino para conocer bajo que estándares de calidad está fabricado el vehículo y según eso tomar una decisión sobre cual vehículo adquirir.
- La interfaz de la aplicación desarrollada será importante al momento de visualizar todos los datos o variables que se desee, es por lo cual que la constante actualización de la misma se logrará mediante el desarrollo de su programación interna obteniendo una aplicación más pulida y agradable para el usuario.
- Realizar aplicaciones que permitan guardar datos en un formato el cual se pueda leerlo e interpretarlo en una hoja de cálculo o software que permita realizar esos análisis.

BIBLIOGRAFÍA

- Alvarado, S., & Resendiz, O. (2018). “*Sistema de Medición de Consumo de Energía Eléctrica Residencial a distancia.*”
- Arduino.cc. (2023). *Arduino* (1.8.19). <https://www.arduino.cc/>
- Asociación Española de Normalización y Certificación. (2013). *Vehículos de carretera; Sistemas de diagnosis; Parte 2: Requisitos CARB para el intercambio de información digital.* www.aenor.es
- Carranco, C. (2022). *Autotrónica Sensor MAF-IAT.*
- Control, T. B., & Committee, C. N. (2023). *J1939 Digital Annex.* https://doi.org/https://doi.org/10.4271/J1939DA_202301
- CSS Electronics. (2023a). *OBD2 Explained - A Simple Intro.* <https://www.csselectronics.com/pages/obd2-explained-simple-intro>
- CSS Electronics. (2023b). *OBD2 PID Overview [Lookup/Converter Tool, Table, CSV, DBC].* <https://www.csselectronics.com/pages/obd2-pid-table-on-board-diagnostics-j1979>
- DevCom. (2022). *CAN BUS Analysis.* <https://www.devcom.cz/en/automotive-systems/can-bus-analysis/>
- Espinosa, F. S. (2021). *El Microcontrolador ATmega328 P de Microchip: Programación en ensamblador, lenguaje c y un enlace con arduino.* (1era ed.).
- García, Á., & Tejero, J. (2015). *Comunicación multimaestro a través de par Trenzado rs-485 (half-duplex).* Universidad de Málaga.
- Ingeniería Mecánica Automotriz. (2021). *¿Qué es el sensor de Posición de Pedal de Acelerador APP y cómo funciona?* <https://www.ingenieriaymecanicaautomotriz.com/1959-2/>
- International Standard. (2005). *Road vehicles — Diagnostics on Controller Area Networks (CAN) — Part 4: Requirements for emissions-related systems (ISO 15765-4).*
- Jiménez, M. (2020). *Sistemas Numéricos.*
- KMTecSoft. (2017). *CAN Bus - Monitor* (1.3.6).

- Mallamaci, M., Gordon, P., Krepi, D., Mónaco, J., González, E., & D'Alessio, H. (2021). *¿Qué puede un bit? Datos y Algoritmos como relación social fundamental de la Economía Digital*. Sociológica. http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S0187-01732020000300215&lng=es&tlng=es.
- Martínez Requena, A., & García Martín, J. (2017). *Introducción a CAN bus: Descripción, ejemplos y aplicaciones de tiempo real* [Proyecto Fin de Máster]. Universidad Politécnica de Madrid.
- Martínez Requena, A. (2017). *Introducción a CAN bus: Descripción, ejemplos y aplicaciones de tiempo real* [Universidad Politécnica de Madrid]. https://msde.etsisi.upm.es/wp-content/uploads/2017/09/Memoria_TFM_Adrian_Martinez_Requena.pdf
- Molina, A., & Oliva, R. (2014). *Análisis del Comportamiento Dinámico de un Vehículo con controlador PID*.
- Picoauto. (2022). *Picoscope 7 Automotive* (7.0.116 Stable).
- PremierAuto Trade. (2023). *Accelerator Pedal Position Sensors (APPS)*. <https://premierautotrade.com.au/news/accelerator-pedal-position-sensors.php>
- Recalde, M., & Stefany, J. (2022). *Desarrollo de un dispositivo basado en Microcontrolador Arduino para la adquisición de datos a través de la Red Can para conocer las condiciones que suscitan accidentes vehiculares*. [Universidad Técnica del Norte]. <chrome-extension://efaidnbnmnibpcajpcglclefindmkaj/http://repositorio.utn.edu.ec/bitstream/123456789/12304/2/04%20MAUT%20179%20TRABAJO%20GRADO.pdf>
- Sánchez Carrizo, J. (2017). *Simulador de una ECU y diagnóstico mediante CAN y OBD II*. Escuela Politécnica Cuenca.
- UT05. *Redes multiplexadas: CAN bus de datos*. (n.d.).
- Wilson, S., Julio, C., Danilo, C., & Gabriel, L. (2016). Design and Implementation of a Remote Monitoring System of a vehicle's engine based on Obd-II and the Arduino platform. In *Revista Politécnica-Marzo* (Vol. 37, Issue 1).

Zambrano Beltrán, J., García Barrero, F., & Sánchez García, J. (2015). *Desarrollo de un simulador electrónico de una ECU y su diagnóstico sobre CAN y OBD-II* [Dep. Ingeniería Electrónica]. Universidad de Sevilla.

Zambrano, J. B., García, F. B., & Sánchez García, J. (2015). *Desarrollo de un simulador electrónico de una ECU y su diagnóstico sobre CAN y OBD-II*.

ANEXOS

ANEXO 1

Código final utilizado para el proyecto parte 1

```
#include <mcp_can.h>
#include <SoftwareSerial.h>

const int spiCS = 10;
MCP_CAN can(spiCS);

SoftwareSerial bluetooth(4, 5); // Configurar los pines para el módulo Bluetooth

void setup() {
  Serial.begin(9600);
  bluetooth.begin(9600); // Inicializar la comunicación con el módulo Bluetooth

  if (can.begin(MCP_ANY, CAN_500KBPS, MCP_8MHZ) == CAN_OK) {
    Serial.println("MCP2515 iniciado correctamente!");
    can.setMode(MCP_NORMAL);
  } else {
    Serial.println("Error al iniciar el MCP2515...");
  }
}

void loop() {
  unsigned long canId;
  unsigned char len;
  unsigned char buf[8];

  bool pid329Received = false; // Variable para controlar si se ha recibido el PID 329

  if (can.checkReceive()) {
    if (can.readMsgBuf(&canId, &len, buf) == CAN_OK) {
      // Verificar si el PID es "A0"
      if ((canId & 0x7FF) == 0xA0) {
        Serial.print("#S1,");

        // Unir los dígitos del tercer y cuarto dígito
        int value = (buf[3] << 8) | buf[2];

        // Multiplicar por 0.25
        int result = value * 0.25;
        Serial.print("01,");
        Serial.println(result);
      }
    }
  }
}
```

```
Serial.println(result);
// Mostrar el segundo dígito del PID repetido
Serial.print("#S2,");
int secondDigit = buf[1] - 40;
Serial.print("01,");
Serial.println(secondDigit);

// Enviar los datos al módulo Bluetooth
bluetooth.print("RPM: ");
bluetooth.println(result);
bluetooth.print("#S2,");
bluetooth.println(secondDigit);
}

// Verificar si el PID es "A1"
if ((canId & 0x7FF) == 0xA1) {
    Serial.print("#S3,");

    // Mostrar solo el quinto valor del PID A1
    Serial.print("01,");
    Serial.println(buf[4], DEC);
    // Después de mostrar el PID A1, mostrar el valor dividido por 2.55 del PID 329 una vez
    if (!pid329Received) {
        // Buscar y mostrar el PID 329
        while (can.checkReceive()) {
            if (can.readMsgBuf(&canId, &len, buf) == CAN_OK && (canId & 0x7FF) == 0x329) {
                Serial.print("#S4,");

                // Dividir el valor del PID 329 entre 2.55
                int value = buf[6];
                float result = value / 2.55;
                Serial.print("01,");
                Serial.println(result);

                pid329Received = true; // Marcar que se ha recibido el PID 329

                // Enviar los datos al módulo Bluetooth

                bluetooth.println(result);
```

```

        break;
    }
}
}

// Repetir la lectura del PID 329 bajo el mismo escenario
if (pid329Received) {
    Serial.print("#S5,");

    // Dividir el valor del PID 329 entre 2.55
    int value = buf[5];
    float result = value / 2.55;
    Serial.print("01,");
    Serial.println(result);

    // Enviar los datos al módulo Bluetooth
    bluetooth.print("TPS ");
    bluetooth.println(result);

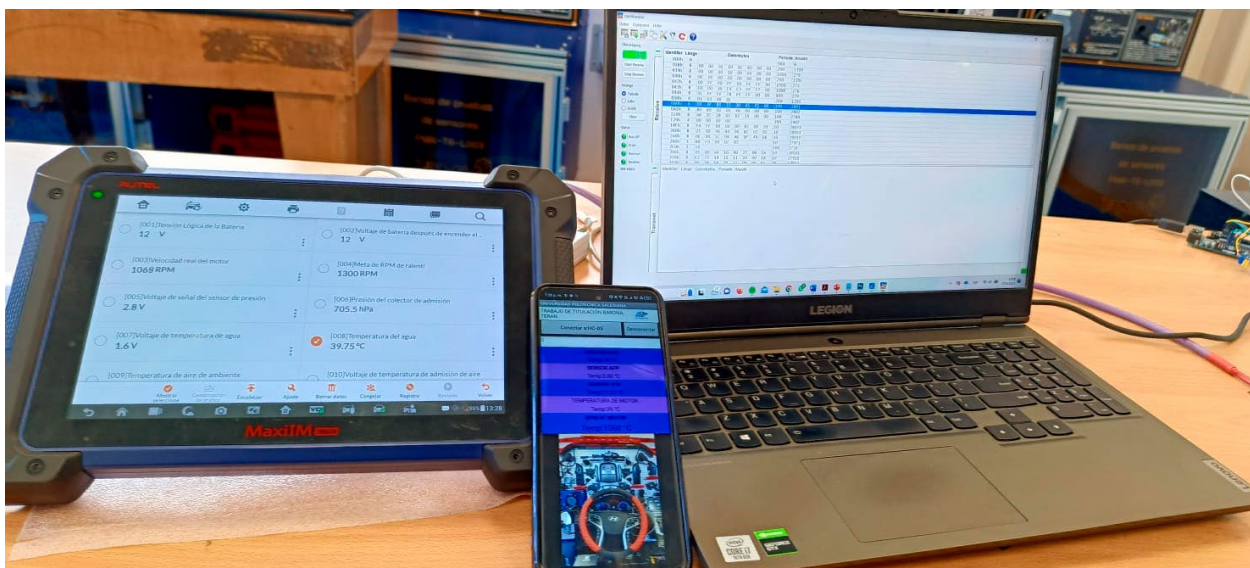
}
}
}
}
}
}

```

Fuente: (Autores)

ANEXO 2

Verificación datos (Can Bus- CanBus-Monitor, Scanner, Aplicación)



Fuente: (Autores)

ANEXO 3

Verificación datos Sensor ECT (CSSElectronics – CanBusMonitor)

The screenshot displays the CSSElectronics CanBusMonitor software interface. The main window shows the 'OBD2 PID Overview [Lookup/Converter Tool, Table, CSV, DBC]' page. A table lists various OBD2 PIDs, including 'Engine coolant temperature (°C)' (PID 07). The table columns include PID, Name, BIT start, BIT Length, Scale, Offset, Min, Max, and Unit. Below the table, there is a section for 'Request' and 'Response (example)' with corresponding CAN IDs and byte values. A 'Physical value (DEC)' is also shown as 21 degC.

PID	Name	BIT start	BIT Length	Scale	Offset	Min	Max	Unit
07	Engine coolant temperature (°C)	39	8	1	-40	-40	125	degC

Request: 70F 02 01 07 AA AA AA AA
Response (example): 70F 03 41 07 30 AA AA AA
Physical value (DEC) = -40 * 1 * 01 = 21 degC

On the right side, there is a 'Receive' data stream table with columns: Identifier, Länge, Datenbytes, Periode, and Anzahl. The table shows various received frames, with the most recent one being 0A0h 8 00 3D A4 37 00 25 08 00 100 4219.

Fuente: (Autores)

ANEXO 4

Verificación datos Sensor ECT (Can Bus- CanBus-Monitor, Scanner)

The screenshot shows a laptop screen displaying the CSSElectronics CanBusMonitor software interface, similar to the one in Anexo 3. The 'OBD2 PID Overview' page is visible, showing the 'Engine coolant temperature (°C)' PID. In the foreground, a blue OBD2 scanner is connected to the laptop. The scanner's screen displays a list of engine parameters with their values and units:

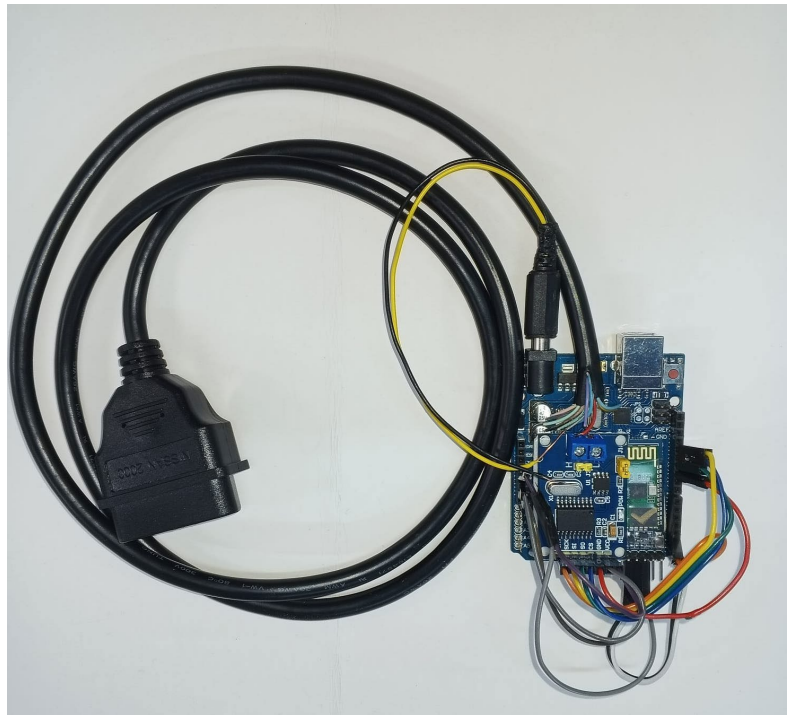
Valor	Unidad
11.4 V	V
11.4 V	V
3574 RPM	RPM
1460 RPM	RPM
2.8 V	V
0.7 bar	bar
2.4 V	V
21.8 °C	°C
19.5 °C	°C
2.4 V	V

The scanner also has buttons for 'Stop' and 'Function' at the bottom.

Fuente: (Autores)

ANEXO 5

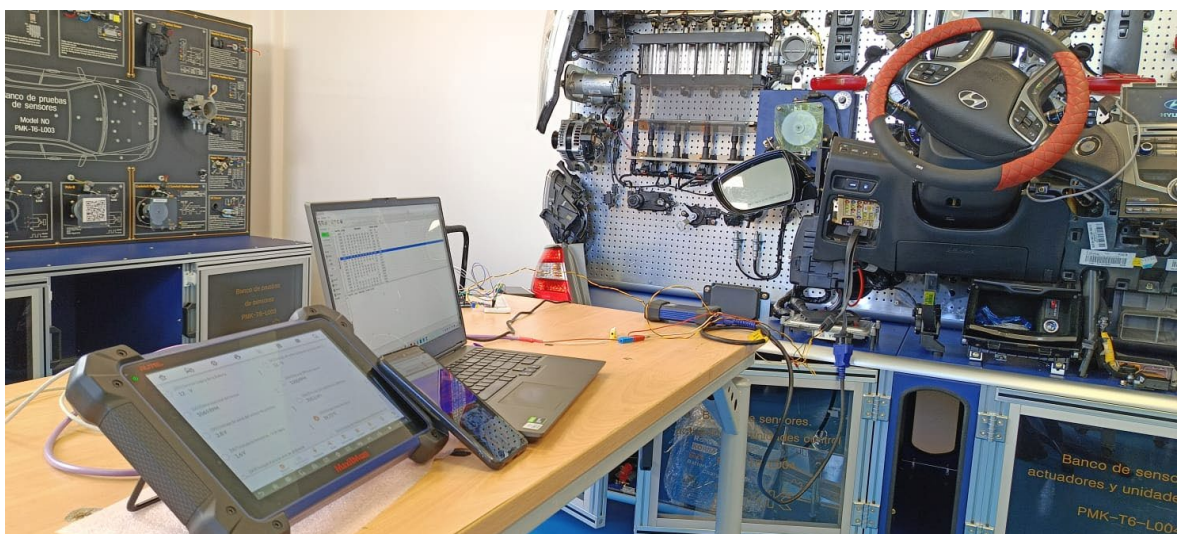
Diseño Final del Dispositivo



Fuente: (Autores)

ANEXO 6

Ensayo de recopilación de datos desde el Banco de Sensores, Actuadores y Unidades de Control



Fuente: (Autores)