UNIVERSIDAD PÚBLICA DE EL ALTO
VICERRECTORADO
DIRECCIÓN DE INVESTIGACIÓN CIENCIA Y TECNOLOGÍA
INSTITUTO DE INVESTIGACIÓN
INGENIERÍA AUTOTRÓNICA - INGENIERÍA DE SISTEMAS

“DESARROLLO E IMPLEMENTACIÓN DE SCANNER AUTOMOTRIZ PARA EL DIAGNÓSTICO DE POTENCIAS Y TORQUE EN VEHÍCULOS LIVIANOS”

PROYECTO FINANCIADO CON RECURSOS PROPIOS
Resolución HCC N° 057/2021 y Resolución HCC N° 083/2019

EQUIPO DE INVESTIGADORES:

- Ing. Julio Cesar Valdez Arzabia
- Univ. Joel Colque Humerez
- Ing. Grover Wilson Quisbert Ibañez
- Univ. Pamela Mamani Vallejos
- Univ. Ricardo Pilcomayo Quenallata

EL ALTO – BOLIVIA
2021

UNIVERSIDAD PÚBLICA DE EL ALTO

AUTORIDADES

Dr. Carlos Condori Titirico
RECTOR

Dr. Efrain Chambi Vargas
VICERRECTOR

Dr. Antonio López Andrade Ph. D.
DIRECTOR DE INVESTIGACIÓN CIENCIA Y TECNOLOGÍA

Ing. David Carlos Mamani Quispe
DIRECTOR DE CARRERA DE INGENIERÍA DE SISTEMAS

Ing. Yolanda Escobar Mancilla
COORDINADOR INSTITUTO DE INVESTIGACIÓN INGENIERÍA DE SISTEMAS

Ing. Roger Omar Llanque Villavicencio
DECANO ÁREA DE INGENIERÍA

Ing. Felix Orlando Mendoza Calani
DIRECTOR DE CARRERA DE INGENIERÍA AUTOTRÓNICA

Ing. Ángel Challapa Huarachi
COORDINADOR INSTITUTO DE INVESTIGACIÓN AUTOTRÓNICA

Derechos Reservados: Universidad Pública de El Alto

Dirección UPEA: Av. Sucre S/N Zona Villa Esperanza

Dirección Ingeniería Autotrónica: Zona villa tejada rectangular, Calle George Squiert No 1080

Octubre 2021
El Alto – Bolivia

PRESENTACIÓN

La Universidad Pública de El Alto es una institución que se proyecta al desarrollo de sus actividades académico-productivas, científicas, tecnológicas de interacción social contemporánea, para priorizar la investigación científica en todos los campos del conocimiento relacionando la teoría con la práctica para transformar la estructura económica, social, cultural y política vigente en favor de las naciones originarias y clases populares.

La investigación se basa fundamentalmente en una estructura de trabajo realizados y de acuerdo a las necesidades en nuestro entorno social, de acuerdo a estructuras geográficas en el departamento de La Paz.

La presente investigación se basa en tres aspectos fundamentales que son: la conectividad del OBDII, que es parte del automovil, a través del ELM327 hacemos una conectividad de diagnóstico del motor y en su análisis del por qué la perdida de potencia y torque (fuerza) que a través de un desarrollo de una aplicación Android se realiza el reconocimiento de conectividad, con los códigos de errores, así generando un flujo de información a través de los tres componentes.

Los protocolos de comunicación son muy primordiales en el trabajo de investigación por lo que el mecanismo de funcionamiento de diferentes tipos de motores hace la necesidad de investigar en cálculos de manejo de potencia y fuerza, donde ingresamos a la Física que es la Ley de Palancas.

Ing. Yolanda Escobar Mancilla
COORDINADORA
IIIS INGENIERÍA DE SISTEMAS

Ing. Angel Challapa Huarachi
COORDINADOR
II DE INGENIERÍA AUTOTRÓNICA

AGRADECIMIENTOS INSTITUCIONALES

Somos una institución académica que formamos profesionales técnicos y competitivos, buscando el desarrollo científico y tecnológico a través de la investigación e innovación.

Agradecemos al Dr. Antonio López Andrade Ph.D director DICyT; Al Ing. David Carlos Mamani Quispe, al Ing. Felix Orlando Mendoza Calani director de la Carrera de Ingeniería Autotrónica, a la Ing. Yolanda Escobar Mancilla Coordinadora del Instituto de Investigación de Ingeniería de Sistemas, al Ing. Angel Challapa Huarachi Coordinador del Instituto de Investigación de Ingeniería Autotrónica por el apoyo otorgado para conclusión de este proyecto de investigación. Asegurando la calidad de la investigación **"DESARROLLO E IMPLEMENTACIÓN DE SCANNER AUTOMOTRIZ PARA EL DIAGNOSTICO DE POTENCIA Y TORQUE EN VEHÍCULOS LIVIANOS".**

RESUMEN

En la potencia de un motor se identifica que al pasar de los años va sufriendo cambios que convergen en la pérdida de fuerza (torque) por lo que el automotor ya no tiene su rendimiento al máximo.

La manera de identificar esta fallas es trabajando en un mecanismo de verificación rápida que a traves de un ELM327 que tienen los protocolos de comunicación con el OBDII, se crea un prototipo en Android para que se pueda identificar las fallas a través de códigos de errores del automovil.

El prototipo de desarrollo de basa en identificar las fallas en el momento sin conocimientos en mecánica, que por lo cual se ahorraría mucho el tiempo y dinero en donde sería una herramienta que estaría al alcance en todos los parque automotores.

ABSTRACT

In the power of an engine it is identified that over the years it undergoes changes that converge in the loss of force (torque) so that the automotive no longer has its maximum performance.

The way to identify these faults is by working on a quick verification mechanism that through an ELM327 that has the communication protocols with the OBDII, a prototype is created in Android so that the faults can be identified through the error codes of the automotive.

The development prototype is based on identifying faults at the moment without mechanical knowledge, which would therefore save a lot of time and money where it would be a tool that would be available in all automotive parks.

INDICE

CAPÍTULO I: INTRODUCCIÓN 1
INTRODUCCIÓN 1
1. PLANTEAMIENTO DEL PROBLEMA 2
1.1. Problema de Investigación 2
1.2. Formulación del Problema 5
1.3. Preguntas de Investigación 5
2. OBJETIVOS DE LA INVESTIGACIÓN 6
2.1. Objetivo General 6
2.2. Objetivo Específico 6
3. HIPÓTESIS DE LA INVESTIGACIÓN 7
4. JUSTIFICACIÓN 7
CAPITULO II: MARCO TEÓRICO 9
1. MENCIÓN DE OTROS ESTUDIOS RELATIVOS AL TEMA 9
1.1. Antecedentes Internacionales 9
1.2. Antecedentes Nacionales 10
2. MENCIÓN DE LOS PUNTOS DE VISTA DE OTROS INVESTIGADORES 11
3. CORRIENTE O ENFOQUE ELEGIDO POR EL INVESTIGADOR 13
3.1. Técnica e Instrumentos 14
4. PROCEDIMIENTO DE LA INVESTIGACIÓN 16
5. BASES TEÓRICAS 17
5.1. La Unidad Electrónica Control (ecu) 17
5.2. Entrada de Señales 18
5.2.1. El núcleo de la CPU 19
5.2.2. Las etapas de salida 20
5.2.3. Componentes 22
5.2.4. Modulo de Control del Motor 22
5.2.5. Luz indicadora de falla (MIL) 23
5.2.6. Conector de Diagnostico (DCL) 24
5.2.7. Protocolos 25
5.2.8. Estándares 26
5.2.9. Modos de Operación 26

5.2.10. Funcionamiento del Sistema 28
5.2.11. Ciclo de Manejo 32
5.2.12. Viaje del Sistema 33
5.3. Protocolo de Comunicación Serial 33
5.3.1. Interfaz RS-232 34
5.4. Herramientas Tecnológicas 37
5.4.1. Bus Can MCP 2515 37
5.4.2. Pantalla LCD monocromática 16x2 38
5.4.3. Arduino 39
5.4.4. CAN-BUS Shield V2 40
5.4.5. Conector OBD II 41
5.4.6. Software Arduino 42
5.4.7. Java 43
5.4.8. Apache NetBeans 43
5.4.9. Maven 44
5.4.10. GitHub 44
CAPITULO III: MARCO METODOLÓGICO 46
1. TIPOS DE INVESTIGACIÓN 46
1.1. Características de la Investigación Experimental 46
1.2. Etapas de la Investigación Experimental 46
2. DISEÑO DE LA INVESTIGACIÓN 47
2.1. Propósito de la Investigación 47
3. ENFOQUE CUANTITATIVO 48
4. VARIABLES DE INVESTIGACIÓN 49
4.1. Par Motor (Variable independiente) 49
4.2. Potencia (Variable dependiente) 50
5. PRUEBA ESTADÍSTICA PARAMÉTRICAS 51
6. NIVEL DE INVESTIGACIÓN 52
7. POBLACIÓN Y MUESTRA 52
8. AMBIENTES DE INVESTIGACIÓN 54
9. PROCEDIMIENTO DE LA INVESTIGACIÓN 55
10. VALIDACIÓN DE INSTRUMENTO 56
CAPITULO IV: MARCO APLICATIVO 56

1. VARIABLES DE INVESTIGACIÓN 57
2. POBLACIÓN DE MUESTRA 59
3. AMBIENTES DE INVESTIGACIÓN 61
4. TÉCNICAS E INSTRUMENTOS UTILIZADOS 62
5. PROCEDIMIENTO DE LA INVESTIGACIÓN 62
6. DESARROLLO DE SIMULADOR OBD 63
7. DESARROLLO DEL SCANNER OBDII 65
8. MINI ELM 327 CAN BUS BLUETOOH V2.1 69
8.1. Características 69
8.1.1. Lectura de Datos 69
8.2. Sistema de Identificación OBDII y su Funcionamiento en el Area Automotriz. 73
8.3. Proceso de comunicación 76
8.4. Comunicación e integrando las fases de cambios 77
8.5. Simulador de ECU 78
CAPITULO V: RESULTADOS 80
1. INTERPRETACIÓN DE LOS RESULTADOS 80
2. DETERMINACIÓN DE POTENCIA INDICADA Y EFECTIVA DE UN MOTOR 1SZ-FE TOYOTA 88
2.1. Determinación en la presión indicada por el diagrama redondeado 88
2.2. Potencia indicada de un motor = Ni. 88
2.3. Perdidas mecánicas en el motor (Nm) 89
2.4. Potencia efectiva (Ne). 90
2.5. Posibles causas de una pérdida de potencia y torque 91
CAPITULO VI: CONCLUSIONES 95
CAPITULO VII: RECOMENDACIONES 96
BIBLIOGRAFÍA 97
ANEXOS 100

LISTADO DE FIGURAS

Figura 1. Diseño Técnicas e Instrumentos 14
Figura 2. Gestión electrónica del sistema de control de un motor. 17
Figura 3 Sistema Motronic 1.3 de un BMW 320i 18
Figura 4. Esquema electrónico de la ECU 18
Figura 5. Señal del captador del cigüeñal 19
Figura 6. Señal del sensor del árbol de levas 19
Figura 7. SAB 80C515/80C535 20
Figura 8. La etapa de salida o potencia 20
Figura 9. Control de la activación de actuadores 21
Figura 10. Fuente de alimentación 21
Figura 11. Módulo de Control del Motor 23
Figura 12. Estructura de un código de error. 29
Figura 13. Especificaciones eléctricas de la interfaz RS23 35
Figura 14. Interfaz RS232 35
Figura 15. Bus CAN MCP2515 38
Figura 16. LCD 16x2 I2C 39
Figura 17. ARDUINO UNO 40
Figura 18. CAN BUS SHIELD V2 41
Figura 19. Cable de conexión OBDII 41
Figura 20. Arduino IDE 42
Figura 21.JAVA 43
Figura 22. APACHE NETBENS 44
Figura 23. MAVEN 44
Figura 24. Github 45
Figura 25. Esquema para la obtención del par motor 50
Figura 26. QR repositorio simuladorObd2 52
Figura 27. Prototipo de SimuladorObd2Java 53
Figura 28. Aplicación simuladorObd2Java 53
Figura 29. Tecnología Empleada en el SIMU-OBDII1.0 63
Figura 30. Diseño CAD Base Simulador 63
Figura 31. Diseño CAD Tapa Simulador 64
Figura 32. Aplicacion de Escritorio Simulador OBD2 64
Figura 33. Simulador OBD SIMU. vista frontal y lateral 65
Figura 34. Diseño CAD de tapa de Scanner 66
Figura 35. Tecnología Empleada en el Sccaner OBDII 67
Figura 36. Dispositivo Scanner OBDII 1.0 (Desarmado) 67
Figura 37. Dispositivo SCANNER-OBDII (Vista Frontal y Lateral) 68
Figura 38. Repositorio GitHub proyecto 68
Figura 39. Diagrama de conexión PDIP y SOIC 71
Figura 40. Diagrama de Bloques 72
Figura 41. Monitoreo de una herramienta de Escaneo 72

Figura. 42 Sistema de diagnóstico de un Vehículo por Escáner 73
Figura 43. Funcionamiento del OBDII 73
Figura 44. ECU programable 75
Figura 45. Comunicacion entre el motor, ELM327 y dispositivo android 78
Figura 46. API (OBDII, ELM327) 79
Figura 47 instrumentos más utilizados para el diagnóstico de fallas 80
Figura 48. Grafica de escáner mas utilizados en diagnóstico 81
Figura 49. Necesidad de diagnóstico de potencia y torque 82
Figura 50. Instrumentos utilizados en diagnóstico del motor 83
Figura 51. Marcas comunes en talleres 85
Figura 52. ECU de prueba 85
Figura 53. Simulador de Fallos 86
Figura 54. Pruebas de ECU y escaner ..... **¡Error! Marcador no definido.**
Figura 55. Escaner de Prueba ..... **¡Error! Marcador no definido.**
Figura 56. Potencia y rpm diagnosticado con escáner 90

LISTADO DE TABLAS

Tabla 1 Parque automotor, según departamento y tipo de servicio, 2003 - 2019 2
Tabla 2 La Paz: parque automotor, según tipo de servicio y clase de vehículo, 2003 - 2019 3
Tabla 3 Dispositivos para medir potencia y torque más usados 4
Tabla 4. Actividades y Ambientes de Investigación 13
Tabla 5. Diseño, Técnicas e Instrumentos 15
Tabla 6. Procesos y Detalles 16
Tabla 7. Conector de diagnostico (DLC) 24
Tabla 8. Monitores Continuos 30
Tabla 9. Monitores No Continuos 31
Tabla 10. Especificaciones de voltaje DC a RS-232 36
Tabla 11. Designaciones de Terminales en EIA RS-232 36
Tabla 12. Características De Bus CAN MCP2515 38
Tabla 13. Dispositivos de visualización 39
Tabla 14. Características técnicas de Arduino Uno 40
Tabla 15. Pines del Conector 41
Tabla 16. Variable Toque y variables fuerza, radio 50
Tabla 17 Variable Potencia, Torque, Velocidad angular 51
Tabla 18. Etapa, Actividades y ambientes 54
Tabla 19. Procedimiento de Investigación 55
Tabla 20.Variables del trabajo de Investigación 58
Tabla 21. Serie historica de vehiculos automotores de tipo unipersonal 59
Tabla 22. Serie histórica según todos los tipos de sociedad dedicados a la reparación de vehículos automotores 60
Tabla 23. Cantidad de talleres dedicados a la reparación de vehículos automotores en la ciudad de El Alto a Enero de 2021 60
Tabla 24. Técnicas Utilizadas 62
Tabla 25. Velocidad de datos 76
Tabla 26. Instrumentos más utilizados en Diagnostico 80
Tabla. 27 marcas de escáner más utilizados 81
Tabla 28. Necesidad de diagnosticar potencia y torque 82
Tabla 29. Instrumentos para el diagnostico de potencia y torque 83
Tabla 30. Herramientas utilizadas para diagnóstico de desgaste el motor 83
Tabla 32. Tipo de puerto mas comunes en talleres 84
Tabla 33. Marcas de vehículos comunes en la ciudad de El Alto 84
Tabla 34. Herramientas utilizados en diagnostico de potencia y torque 85

CAPÍTULO I: INTRODUCCIÓN

INTRODUCCIÓN

Durante los años 70´ y 80´ ya existían los componentes electrónicos, es decir que se tenía el control en los automóviles, ya por los 90´ surgió el sistema de OBD I continuando con suversión mejorada OBDII[1] (On Board Diagnostic) donde el principal objetivo era el hallar las fallas en el sistema de inyección provocadas en la combustión.

El problema se centra en que en la actualidad la tecnología va avanzando y los sistemas mecánicos están dependiendo de la electrónica, por ende el vehículo como tal, ya no depende solo de la mecánica sino que trabaja en conjunto con la electrónica para así aumentar la eficiencia dela conducción, por este motivo es necesario realizar mantenimientos o revisiones periódicasa estos sistemas en especial a la electrónica, donde intervienen sensores y actuadores delos diferentes sistemas del automóvil.

El acceso a un lector OBDII con un sistema (aplicación móvil, aplicación de escritorio, servicio Web, página Web) que interprete la información ofrecida al interior del mercado nacional, representa un costo elevado de inversión, esto hace que la población en general no pueda acceder a estos instrumentos y menos poder trabajar con los mismos, esta realidad no se aleja de los estudiantes de la Universidad Pública de El Alto que dentro de los laboratorios de investigación no se cuenta con un dispositivo de estas características y menos uno elaborado dentro de nuestro país.
El diseño de un escáner bajo el protocolo OBDII (Diagnóstico a bordo) permitirá a un técnicoautomotriz y/o público en general contar con una herramienta de bajo costo, que permita diagnosticar automóviles. El diseño propuesto ofrece una herramienta simple, externa, de reducido tamaño y fácil de usar. Las opciones de diagnóstico implementadas con el escáner son: Lectura de DTCs almacenando, Borrado de DTCs,

1 OBD II, es un sistema de diagnóstico a bordo en vehículos (coches y camiones). Actualmente se emplean los estándares OBD-II (Estados Unidos), EOBD (Europa) y JOBD (Japón) que aportan un monitoreo y control completo del motor y otros dispositivos del vehículo. ya que es un sistema de mucha ayuda.

Lectura en vivo de sensores, detección de protocolos de comunicación, cálculo en vivo de la potencia y torque del vehículo. El presente proyecto será desarrollado en base a Software Libre. El software libre es aquel que les da a sus usuarios la libertad de ejecutar, copiar, estudiar, modificar y distribuir el software. En otras palabras, da la posibilidad de controlar el programa y lo que hace (Free Software Foundation Staff, 2012)

1. PLANTEAMIENTO DEL PROBLEMA

1.1. Problema de Investigación

El parque automotor del país no para de incrementarse, de acuerdo a los datos del RegistroÚnico para la Administración Tributaria Municipal (RUAT, 2021) en Bolivia fueron reportados 2.013.400 vehículos en 2019, cantidad superior en 5.4% a la de 2018, cuando se contabilizaban 1.910.127 unidades.

Según los datos ofrecidos por el Instituto Nacional de Estadística (INE, 2021) el departamento de La Paz hasta la gestión 2019 cuenta con la siguiente información:

Tabla 1 Parque automotor, según departamento y tipo de servicio, 2003 - 2019

La Paz	**471,873**
Particular	445,835
Público	11,612
Oficial	14,426

Fuente: INE, Datos estadísticos (INE, 2021)

En 2019 el departamento de La Paz contaba con alrededor de 471.873 vehículos, los cuales se encuentran agrupados en diferentes tipos de acuerdo a las características propiasde cada uno, partiendo desde el punto de observación de vehículo liviano “Son vehículos livianos, los automóviles, camionetas, vagonetas, jeeps” (POLICIA BOLIVIANA, 8 de junio 1978), se puede resumir la siguiente información:

Tabla 2 La Paz: parque automotor, según tipo de servicio y clase de vehículo, *2003 - 2019*

TIPO DE SERVICIO (2)/CLASE DE VEHÍCULO	2018	2019
TOTAL	**449,945**	**471,873**
Particular	**425,032**	**445,835**
Automóvil	82,242	85,129
Camioneta	38,736	40,562
Jeep	14,515	14,702
Vagoneta	131,765	138,032
Público	**10,853**	**11,612**
Automóvil	1,299	1,305
Camioneta	459	498
Jeep	13	12
Vagoneta	740	819

Fuente: INE, Datos estadísticos (INE, 2021)

En un porcentaje menor de los casos, el parque automotor cuenta con avances tecnológicosque incluyen sistemas inteligentes a bordo, sistemas de diagnóstico de fallas y análisis de funcionamiento que incluye eficiencia energética y efectos medioambientales.

Estos avances han permitido simplificar la reparación de los vehículos, detectando fallos potenciales, simplificando y centralizando la visualización de variables en tiempo real, queahora pueden ser leídas directamente desde un computador o ECU (Electronic Control Unit,Unidad de Control Electrónico).

En el parque automotor el constante movimiento y circulación y debido a la geografía de nuestras ciudades va en decremento y reduciendo la potencia y torque, esto provoca fallas en las movilidades. Es estudio y control de estos indicadores conllevara

a la mejora a bajo costo y esperando asimilar los valores de fábrica, para así recuperar su fuerza en su totalidad.

Tabla 3 Dispositivos para medir potencia y torque más usados

Dispositivo	Funciones
Banco dinamométrico de rodillos	✓ Pruebas de bajo carga ✓ Inclinación y pendiente ✓ Resistencia aerodinámica ✓ Resistencia a la rodadura
Método mediante la aceleración libre	✓ Ser de bajo costo ✓ Son versátiles en cuanto a manejo ✓ Determinan el torque y potencia con cierta imprecisión o precisión
Scanner automotriz	✓ Existen en el mercado escáner automotrices capaz de determinar potencia y Torque del motor en tiempo real, sin embargo, no todos los escáneres cuentan con el diagnostico de esas variables.

Fuente: Elaboración propia

Ante el costo elevado de los bancos dinamométricos de rodillos surge la necesidad de utilizar un scanner automotriz para el uso de estos propósitos, existen muchos scanner para el uso de estos propósitos, sin embargo, estos scanner son de uso restringido y limitado para ciertas marcas, por esta razón surge la necesidad de determinar potencias y torques en un scanner que pueda ser utilizado para muchas marcas de vehículos livianos.

Determinar el torque y potencia permite diagnosticar una pérdida de potencia o torque para una futura corrección o reparación de un automóvil, además esto permitirá comparar potencia y torque después de la reparación de un automóvil, para observar las mejoras del valor de torque y potencia.

La problemática del parque automotor, por el constante movimiento y circulación de movilidades, tanto en la potencia y torque se observa una disminución de estas variables por desgaste, condiciones de la ciudad de El Alto provocando fallas en las movilidades y emisiones de contaminantes.

Lo que se quiere es conservar su potencia y torque de diseño de las movilidades a bajo costo y volviéndolo a su estado de fábrica para así pueda recuperar sus características iniciales de funcionamiento normal.

1.2. Formulación del Problema

¿Cuál será el impacto económico, social y medio ambiental de la implementación de prototipos de scanner automotrices para la medición de potencia y torque en los talleres automotrices y la sociedad en la ciudad de El Alto del departamento de la Paz?

¿Qué tan necesario resulta contar con un escáner que Diagnostique Potencia y Torque en un motor en tiempo real, además que económicamente tenga un costo menor a los utilizados en talleres de la ciudad de El Alto?

1.3. Preguntas de Investigación

- ¿Es factible la implementación del scanner desarrollado para los talleres automotrices como también para la sociedad?
- ¿Qué controlador se utilizará para el desarrollo del scanner automotriz?
- ¿Cuáles serán los protocolos e ISOs a implementar?
- ¿Cuánto se ha avanzado y alcanzado en el uso de escáner automotriz y dinamómetros en talleres especializados en la ciudad de El Alto?
- ¿Es factible desarrollar un escáner que se comunique efectivamente con OBD II para la aplicación desarrollada?
- ¿Es posible diagnosticar potencia y torque en vehículos livianos en las condiciones de la ciudad de El alto?
- ¿Qué tan necesario resulta implementar un escáner para diagnóstico de potencia y

torque en taller de la carrera de ingeniería Autotrónica?

- ¿Es posible desarrollar e implementar un escáner de diagnóstico de potencia y torque en tiempo real?
- ¿Es posible desarrollar un escáner con posibilidad de diagnóstico de potencia y torque de menor costo a los utilizados en talleres de la ciudad de El Alto?

2. OBJETIVOS DE LA INVESTIGACIÓN

2.1. Objetivo General

El propósito del presente proyecto de investigación es desarrollar e implementar un prototipo de scanner automotriz a un costo económico, el cual colaborará a la medición y cálculo de potencia y torque con lectura de posibles errores en OBDII, identificando fallas en tiempo real, mejorando el medio ambiente para la sociedad en la ciudad de El Alto del departamento de La Paz.

2.2. Objetivo Específico

- Identificar la mejor alternativa para el proceso de comunicación para el desarrollo del prototipo de scanner automotriz.
- Desarrollar un prototipo de simulador OBDII para poder realizar pruebas.
- Desarrollar un prototipo de scanner automotriz para diagnosticar las pérdidas de potencia por desgaste.
- Implementar comunicación entre scanner y vehículo para la lectura de datos de potencia y torque.
- Determinar medidas de torque y potencia a tiempo real.
- Comparar el costo y funcionalidad con los diferentes Scanner existentes en el mercado con el prototipo a desarrollar.
- Socializar el prototipo de scanner automotriz desarrollado.
- Realizar una encuesta sobre uso de Escáner automotrices, dinamómetros en talleres de la ciudad de El Alto.
- Implementar comunicación entre scanner y vehículo para la lectura de datos de potencia y torque.

- Proveer con el conocimiento y datos sobre la necesidad de implementación de escáner para diagnóstico de potencia y Torque en la carrera de ingeniería Autotrónica.

3. HIPÓTESIS DE LA INVESTIGACIÓN

El prototipo de escáner automotriz desarrollado para la ciudad de El Alto del departamento de La Paz, reducirá el costo de implementación de instrumentos de medición y evaluación en talleres, institutos y otros, además permitirá un diagnóstico de potencia y torque en tiempo real de vehículos livianos bajo el protocolo OBDII.

4. JUSTIFICACIÓN

En la aplicación de la ingeniería se busca entre otras cosas la optimización de los procesos; en el caso de los vehículos es indispensable contar con herramientas que permitan el diagnostico a favor del funcionamiento óptimo de los sistemas que los componen además de poder ofrecer el cálculo de potencia y torque. El prototipo de escáner OBD2 permite tener acceso a información en tiempo real del vehículo facilitando el diagnostico de fallas en su funcionamiento. El funcionamiento óptimo trae importantes ventajas, como el uso eficiente del combustible y mantener bajos los niveles de contaminación, temas que han adquirido una mayor relevancia en los últimos años.

Este proyecto de investigación consiste en que, a través del diseño y la programación, se pueda crear una interfaz gráfica de fácil interpretación para mecánicos o estudiantes, que muestre información del sistema OBD2 a través de la comunicación entre el prototipo y el automóvil. Fundamental para el diagnóstico de los vehículos modernos. Utilizando un prototipo desarrollado que cumple la función de adquisición de datos entre el vehículo y un computador, a través del uso de Software y hardware Libre.

Socioeconómico, El prototipo de escáner desarrollado, tendrá un costo menor a los encontrados en el mercado nacional, además contará con una interfaz amigable para el usuario, incluso que se pueda escanear, sin la necesidad de contar con conocimientos especializados en el área.

Área académica, el presente documento permitirá contar con conocimiento e información sobre diagnóstico, contar con diferentes alternativas, difundir los conocimientos del desarrollo del scanner automotriz en específico en la medición de potencia y torque de

motores para el beneficio de los estudiantes de la carrera de Ingeniería Autotrónica y/o Ingeniería de Sistemas.

Tecnológica, contar con un nuevo dispositivo, resulta de gran ayuda y permite contar con una mayor gama de tecnología en la universidad, así como también en los talleres técnicos especializados.

CAPITULO II: MARCO TEÓRICO

1. MENCIÓN DE OTROS ESTUDIOS RELATIVOS AL TEMA

Como antecedentes al proyecto de investigación se desarrolló la búsqueda de similitudes en temas de estudios relacionados al desarrollo e implementación de un prototipo de scanner automotriz para el cálculo de potencia y torque en vehículos livianos, los cuales se resumen en el siguiente punto.

1.1. Antecedentes Internacionales

En la ciudad de Quito, Ecuador en el año 2006 en la Escuela Politécnica Nacional, la Srta. Estévez Freire Lila Grimanesa desarrolla la tesis con el tema “Diseño y Construcción de un Sistema de Información de Parámetros de un Automóvil” El presente proyecto se diseña y construye un sistema de información de parámetros de un automóvil que es capaz de detectar los diferentes parámetros eléctricos y mecánicos del mismo. Para el monitoreo de los diferentes parámetros se utiliza un microcontrolador PIC16F877A, toda la información que este deba dar al usuario se presenta en un display LCD alfanumérico 20X4, adicionalmente se tiene un led bicolor que indica el estado del sistema, si es normal está en verde en caso contrario se pone en rojo. Para hacer más fácil el uso y para tener un adecuado mantenimiento de los vehículos en la actualidad la mayor parte de fabricantes incluyen en ellos un sistema de información de parámetros al cual le denominan Computador de a Bordo. Con el presente trabajo se pretende dar una alternativa económica de sistema de información de parámetros de un automóvil para ser montado en cualquier tipo de vehículo motorizado. (Estevez Freire Lila Grimanesa, 2006)

En la Ciudad de Ambato, Ecuador en el año 2015 en la Universidad Técnica de Ambato, el Sr. Torres Edwin Ismael desarrolla la tesis con el tema "Sistema Electrónico por comando de voz para la seguridad física, iluminación y accionamiento automático de los accesorios primarios de un automóvil" En este proyecto se ha desarrollado un Prototipo Electrónico de asistencia de conducción, para el control del sistema de iluminación, ejecución de dispositivos y un sistema de seguridad; todo controlado por reconocimiento de voz. La implementación del Prototipo permitió al usuario volver a conducir un vehículo de forma segura disminuyendo las distracciones por accionamiento manual, resolviendo los problemas ocasionados por su discapacidad física. (Torres Valverde Edwin Ismael, 2015).

En la ciudad de Pinar del Rio, Cuba, en el año 2018 en la Universidad de Pinar del Rio, se presenta un proyecto implementa un sistema de diagnóstico abordo basado en el protocolo de diagnosis OBD-II sobre CAN-BUS que permitirá, tanto visualizar variables

en tiempo real, como realizar un diagnóstico del estado del automóvil que muestre los códigos de funcionamiento, falla y rendimiento energético. Los sistemas de diagnóstico de a bordo nos permiten no solo conocer los códigos de fallos almacenados, también nos permiten conocer en tiempo real un gran número de variables de especial relevancia como la velocidad, el nivel de combustible, el nivel de emisión de CO2, etc. El trabajo desarrollado se centra en un sistema de diagnóstico específico basado en el protocolo OBD-II sobre CAN-BUS, como una de las posibles aplicaciones que posee dicho bus de comunicación. En el proyecto se implementó un sistema de diagnosis OBD-II centrándose en el extremo del BUS que corresponde al escáner o unidad de diagnóstico. El sistema propuesto consiste en el desarrollo de un equipo capaz de recuperar los datos relacionados al estado del vehículo bajo prueba. Estos equipos revisten gran utilidad a la hora de detectar o, al menos, conducir al personal de mantenimiento automotor, a la solución de averías en los automóviles. En el sistema se desarrolló el estándar OBD-II en una placa Arduino Mega 2560, conectada un módulo compuesto transceiver-controller CAN. El scanner posee una conexión USB la cual permite la visualización de los datos recuperados de forma versátil en una PC a través de una interfaz gráfica creada en LabVIEW™. (Rodriguez Rodriguez, Armando; Vento Alvarez, Jose Raul, 2018)

1.2. Antecedentes Nacionales

En la ciudad de La Paz, en el año 2013 en la Universidad Mayor de San Andrés, el Señor Jorge Luis Soria Mamani, presente la Tesis de Grado “Sistema Experto Para El Diagnostico De Fallas En Motores A Inyección Electrónica De Vehículos”, Los sistemas basados en conocimiento constituyen una rama de la inteligencia artificial que hace un uso amplio de conocimiento especializado para resolver problemas como un especialista humano. Los sistemas expertos son programas que logran reproducir el proceso intelectual de un experto humano en un campo en particular, con alta fiabilidad, pudiendo así mejorar su productividad, la construcción de un sistema experto es una tarea muy complicada, se necesita mecanismos para la adquisición de conocimiento, considerada la tarea más complicada en la ingeniería del conocimiento, además involucra a muchos participantes, cada uno de los cuales colabora para que el sistema experto a desarrollar sea robusto, sencillo de manejar y simple para su mantenimiento. El trabajo se desarrolla un prototipo de sistema experto probabilístico para el diagnóstico de fallas en motores a inyección electrónica de vehículos, ofreciendo así al usuario un método más para el diagnóstico correcto de su motor y este no sea engañado. Además, que pueda aprender sobre algunas de las partes que tiene el motor de su vehículo. (Jorge Luis Soria Mamani, 2013)

Cabe aclarar que, dentro de la búsqueda en repositorios nacionales de las diferentes Universidad e Institutos Superior, no se encontró trabajos específicos relacionados con el tema de investigación.

En el desarrollo e implementación del escáner automotriz para el diagnóstico de potencia y torque para vehículos livianos, se encuentran muy pocos trabajos documentados, algunas investigaciones realizadas respecto a este tema se encuentra:

(Castillo y Rojas, 2016) desarrollan un modelo matemático para determinar potencia y torque para vehículos de combustión interna a gasolina en cualquier locación, se realiza la implementación de un sensor de presión adaptado en la bujía de uno de los cilindros, este sensor toma de dato la presión en el cilindro, para después realizar un cálculo de la fuerza y así luego determinar la potencia y torque de un motor para vehículos de cilindraje entre 1400 y 2400 centímetros cúbicos.

(Andrade y Pereira, 2019) diseñaron e implementaron un prototipo de escáner con visualización de datos en tiempo real con cálculo de parámetros indicados y efectivos para motores de combustión interna. Este prototipo satisface las necesidades más relevantes consultadas a especialistas automotrices y se establece que debe realizar:

- Medir torque, potencia y consumo especifico
- Leer códigos de falla
- Leer flujo de datos en tiempo real
- Eliminar códigos de falla
- Interfaz de usuario amigable
- Presentar reporte histórico de datos de potencia y torque
- Es compatible con los protocolos ISO 15765-4 (CAN), ISO 14230-4 (KP2000), ISO 9141-2, SAE J1850 VPW, SAE J1850 PWM, ISO 15765, ISO 11898, SAE 1939

2. MENCIÓN DE LOS PUNTOS DE VISTA DE OTROS INVESTIGADORES

Según la investigación “Sistema Electrónica de Alerta y Monitoreo Para el Mantenimiento de los Vehículos de la Cruz Roja Junta Provincial de Napo”, de Danny Paul Morocho Medina menciona que “Es importante conocer los avances tecnológicos en el campo automotriz, ya que al estar actualizado es posible entender su funcionamiento y aplicación, siendo más notorio el desarrollo debido a que se utilizan

los mismos dispositivos con diferentes objetivos. Los conocimientos adquiridos en las aulas permiten entender de manera sencilla y ágil estos avances, ya que son de donde parten todas las innovaciones".

El **objetivo** del trabajo Implementar un Sistema Electrónico de alerta y monitoreo para el mantenimiento de los vehículos de la Cruz Roja Junta Provincial Napo.

El **resultado** del trabajo resulta ser capaz de cumplir los requerimientos de alerta y monitoreo, demostrándose su capacidad de adquisición de información, procesamiento y aviso mediante los dispositivos utilizados, determinándose como principal ventaja un monitoreo en tiempo real, prologando así la vida útil de los vehículos de la empresa.

Según el proyecto de grado "Programa para sistema de Diagnostico de Vehiculos (OBDII) y comprobación de actuadores que funcionen con señales PWM" de Juan Baptista Lopez Perez menciona "Aunque me ha decepcionado un poco la baja cantidad de sensores soportados por el fabricante en las unidades de control. Los vehículos americanos ofrecían gran cantidad de sensores soportados, mientras que los europeos y asiáticos casi la mitad. Pero a pesar de esto, los sensores soportados eran de gran importancia para el seguimiento de una posible avería"

El **objetivo** elaboración de un prototipo de sistema de diagnóstico de vehículos ob2.

El **resultado** del trabajo Los casi 8000 códigos de avería, genéricos y específicos del fabricante que incorpora en la base de datos del programa, ayudaron en muchas ocasiones a resolver averías que "sorprendentemente" en talleres con aparatos de diagnosis de alto coste no interpretaban.

Según la investigación "Desarrollo de una interfaz para la visualización y adquisición de datos provenientes de la ECU a travez de OBD2, mediante un dispositivo de comunicación serial y del analizador de gases QRotech 6000" menciona "En el análisis estacionario (Analizador de Gases) se puede observar un enfoque de información al adquirir estos datos, mientras que en el análisis transitorio (OBD), se puede observar un error de 9% al adquirir los datos por instrumentación externa"

El **objetivo** del trabajo es Desarrollar una interfaz, para visualizar los datos provenientes de la ECU a travez del OBDII, con la ayuda de un dispositivo de comunicación serial y analizador de gases.

Los **resultados** del trabajo, se conecto la interfaz y el analizador de gases en un vehículo Chevrolet AVEO 1.6L del año 2012 y asi realizar el muestreo de los gases de escape y los sensores del automóvil, para posteriormente poderlos validar con un programa estadístico llamado Minitab, el cual nos muestra el comportamiento del programa al momento de tomar los datos, y así poder ver las diferencias entre el almacenamiento automático de la interfaz.

Castillo y Rojas, en 2016 implementan un dispositivo exclusivamente para leer datos en tiempo real de potencia y torque de un motor de combustión interna a gasolina, esto realiza una función similar a la realizada por dinamómetros de rodillos, el dinamómetro de rodillos tiene un costo muy elevado comparado con la implementación de sensor de presión, sin embargo, esta implementación no se utiliza para el diagnóstico o lectura de códigos de errores.

Andrade y Pereira Desarrollan un escáner capaz de determinar potencia y Torque en tiempo real capaz de visualizar estas variables en forma gráfica, además de determinar código de fallas.

3. CORRIENTE O ENFOQUE ELEGIDO POR EL INVESTIGADOR

El presente proyecto de investigación al ser experimental se debe tener en cuenta para el desarrollo del scanner automotriz en una primera etapa se lo realizará en las instalaciones del instituto de investigaciones de la carrera de Ingeniería de Sistemas donde se desarrollará la parte de comunicación entre el Ordenador y el prototipo a desarrollar.

En una segunda etapa se realiza pruebas de campo PC – Prototipo a Desarrollar – Vehículo, revisando y evaluando la comunicación, esta se llevará a cabo en los ambientes del taller de la carrera de Ingeniería Autotrónica para ir controlando, verificando y la funcionalidad del scanner. Para esta etapa se utilizara el enfoque cuantitativo.

“Para Cerda (2005) el método cuantitativo se refiere a los proyectos de investigación que se van a trabajar con aspectos medibles en cantidades, como cifras, datos y muy posiblemente con estadísticas” (Perico-Granados, Galarza, Díaz, Arévalo-Galarza y Perico-Martínez, 2020, p.58).

Este enfoque permite determinar, medir valores observados en el prototipo a desarrollarse.

Tabla 4. Actividades y Ambientes de Investigación

Etapa	Actividades	Ambientes
Primera	Investigación de parámetros de diseño Desarrollo del Simulador de OBDII Desarrollo de scanner automotriz	Instalaciones de Instituto de Investigaciones de la Carrera de Ingeniería de Sistemas

Segunda	Verificación de circuito para la comunicación entre PC-Automóvil Pruebas de campo Funcionalidad de scanner automotriz	Taller de la carrera de Ingeniería Autotrónica

Fuente: Elaboración propia

3.1. Técnica e Instrumentos

Se conoce al procedimiento o la forma particular de obtener datos a las técnicas de investigación. Las técnicas de recolección de datos son las distintas formas o maneras de obtener la información. Son ejemplos de técnicas; la observación directa, la encuesta en sus modalidades: oral o escrita (cuestionario), la entrevista, el análisis documental, análisis de contenido, etc. (Fidias G. Arias, 2006)

Para registrar o almacenar la información se utiliza los respectivos instrumentos de recolección de datos, estos pueden ser en papel o formato digital, los medios materiales los cuales se emplean para recoger y almacenar la información son los respectivos instrumentos. (Fidias G. Arias, 2006)

En la siguiente tabla se muestra la clasificación entre diseño técnicas e instrumentos:

Figura 1. Diseño Técnicas e Instrumentos

Diseño	Técnicas	Instrumentos	
Diseño de Investigación Documental	Análisis documental	Fichas Computadora y sus unidades de almacenaje	
	Análisis de contenido	Cuadro de registro y clasificación de las categorías	
	Observación	Estructurada	Lista de cotejo Escala de estimación
		No Estructurada	Diario de campo Cámaras: fotográfica y de video

Diseño de Investigación de Campo	Encuesta	Oral	Guía de encuesta (Tarjeta) Grabador Cámara de video
		Escrita	Cuestionario
	Entrevista	Estructurada	Guía de entrevista Grabador / Cámara de video
		No estructurada	Libreta de notas Grabador / Cámara de video

Fuente: (Fidias G. Arias, 2006)

Para el presente proyecto se muestra en la siguiente tabla el resumen de aplicación de técnicas e instrumentos:

Tabla 5. Diseño, Técnicas e Instrumentos

Proyecto	Diseño	Técnica	Instrumentos
Desarrollo de scanner automotriz	Diseño de investigación Documental	Análisis documental de las pruebas realizadas de la construcción del scanner y el simulador automotriz	Fichas de pruebas de simulador automotriz. Fichas de pruebas de scanner automotriz. Computadora y Software en cuales se observará el control de avance del proyecto.
	Diseño de la investigación de campo	Observación del proceso de construcción y desarrollo del proyecto.	No estructurada, Manejo de informes con los resultados obtenidos. Fotografías y videos de avance y pruebas realizadas en el proyecto.

Fuente: Elaboración Propia

4. PROCEDIMIENTO DE LA INVESTIGACIÓN

En el procedimiento de la investigación para el presente proyecto se realizará mediante el proceso de los siguientes pasos, los cuales están descritos desde el inicio del proyecto así también como en el proceso de construcción y desarrollo del scanner automotriz.

Tabla 6. Procesos y Detalles

Proceso	Detalle	
0	Tener las bibliografías correspondientes para el desarrollo del escáner automotriz.	Datos de Scaner
		Códigos de errores
		Comunicación
		Tipos de conectores
		Parámetro iniciales
1	Desarrollar un Simulador de OBD	Arduino y Java
2	Determinar el tipo de controlador que se utilizara para la comunicación de entre PC – Vehículo.	
3	Determinar el tipo de comunicación que se tendrá entre PC-Vehículo.	
4	Realizar la investigación correspondiente para tener en cuenta los materiales a ser utilizado para el scanner automotriz.	
5	Realizar las respectivas pruebas de funcionamiento, comunicación y verificación de PC-Vehículo.	Pruebas de comunicación
		Pruebas de funcionamiento de scanner automotriz
6	Realizar las respectivas conclusiones con los datos obtenidos como también las mejoras que se pueden realizar tanto en funcionalidad como en costo.	

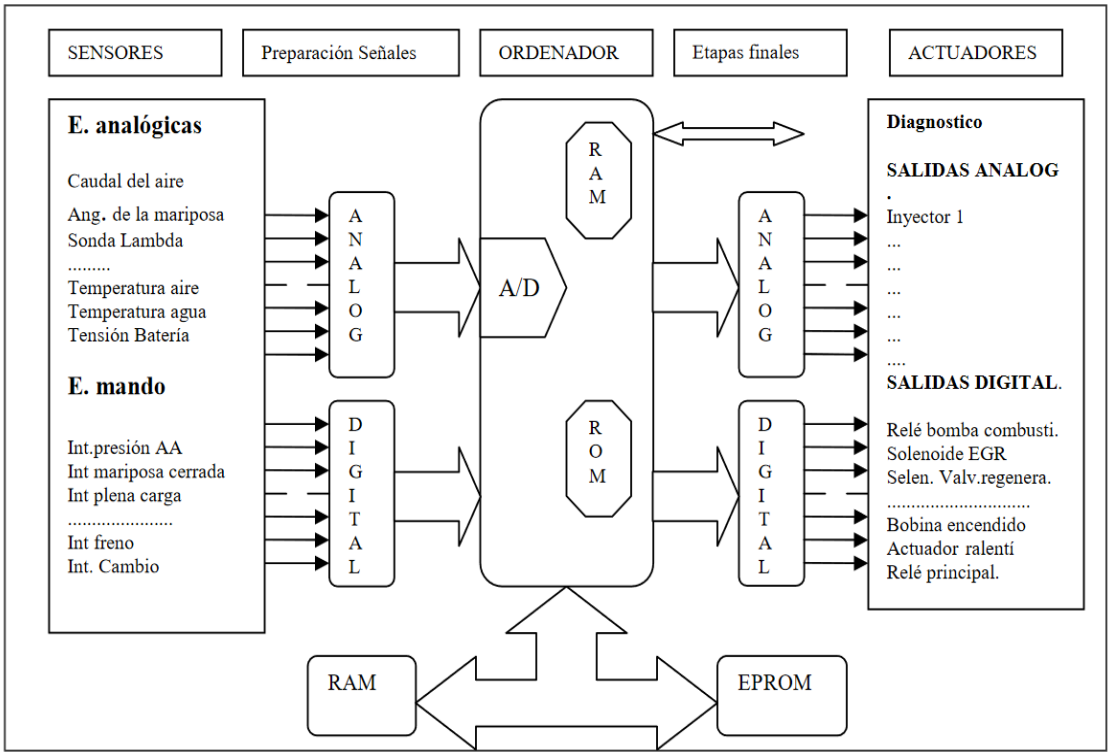
Fuente: Elaboración Propia

5. BASES TEÓRICAS

5.1. La Unidad Electrónica Control (ecu)

La unidad electrónica de control recibe las señales de los diferentes sensores, gestiona estos datos y mediante una cartografía previamente grabada en memoria, aplica a los actuadores las señales registradas para que el motor funcione dentro de los márgenes de potencia y consumo que el fabricante ha estipulado. En el siguiente diagrama (Figura 2) se analizan los diferentes bloques de que consta la gestión electrónica del sistema de control de un motor.

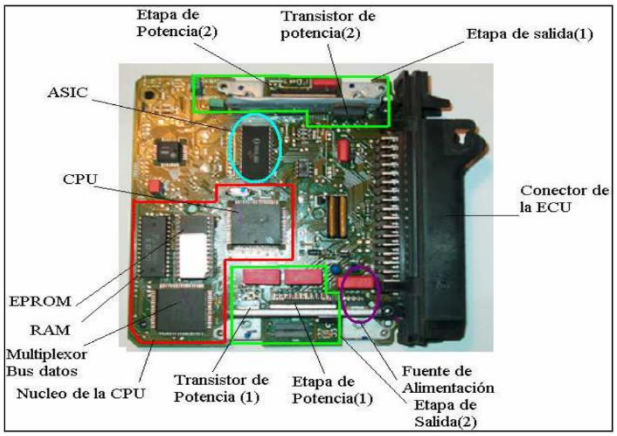
Figura 2. Gestión electrónica del sistema de control de un motor.



Fuente: (CHOLL, 1978)

En la figura 3 se muestra una ECU del sistema Motronic 1.3 de un BMW 320i, en el que se puede apreciar las diferentes partes de que consta la unidad de control. Seguidamente haremos un estudio de cada una de sus partes de forma generalizada.

Figura 3 Sistema Motronic 1.3 de un BMW 320i

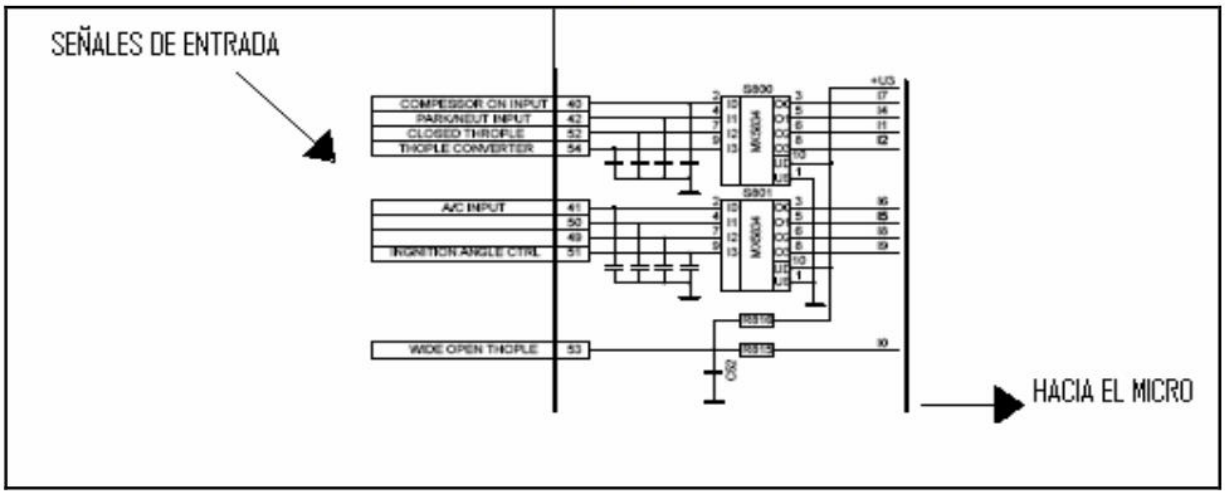


Fuente: Extraído del libro (BOSCH, 2002)

5.2. Entrada de Señales

Las señales de entrada son preparadas por circuitos electrónicos que adaptan la señal para poder ser leídas por el microcontrolador. Las señales de entrada de mando son normalmente interruptores con tensiones de 12v ó 5v que antes de ser leído por el microcontrolador pasan por un buffer para convertir a señales de tensión (5v), con la intención de protegerlo. En figura 4. muestra una parte del esquema electrónico de la ECU de la figura anterior, en la que una serie de señales de mando son previamente intercaladas por un buffer antes de ser leídas por el micro.

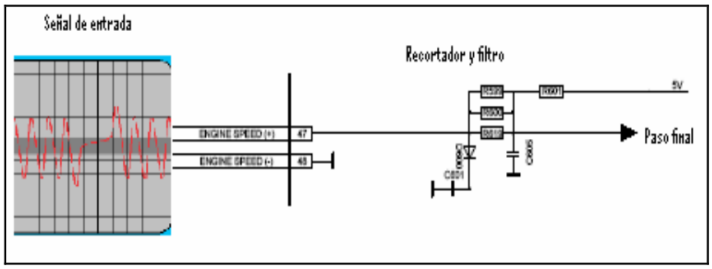
Figura 4. Esquema electrónico de la ECU



Fuente: Extraído del libro (BOSCH, 2002)

Las señales de entrada analógicas son preparadas a través de convertidores de señal, amplificadores y filtros antes de llegar al micro. En siguiente figura se muestra un ejemplo de cómo se trata la señal del captador del cigüeñal.

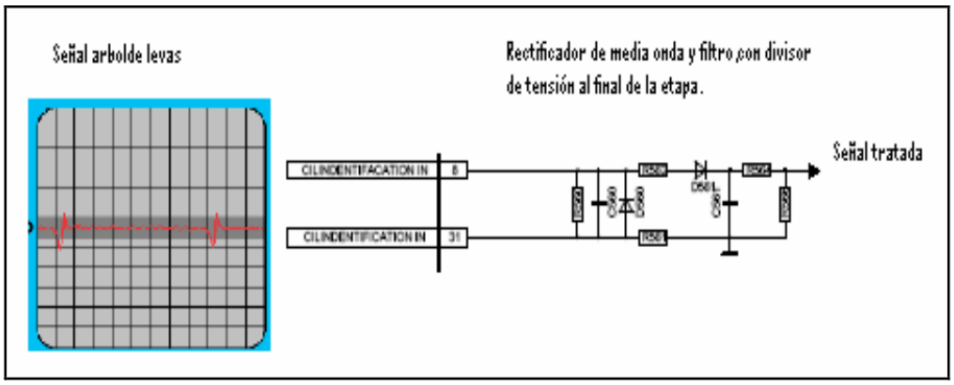
Figura 5. Señal del captador del cigüeñal



Fuente: Extraído del libro (BOSCH, 2002)

Para señal del sensor del árbol de levas, se rectifica la señal produciendo a la salida pulsos de tensión que se interpretarán como que el cilindro 1 está en compresión, puesto que el sistema Motronic 1.3 es semisecuencial

Figura 6. Señal del sensor del árbol de levas



Fuente: Extraído del libro (BOSCH, 2002)

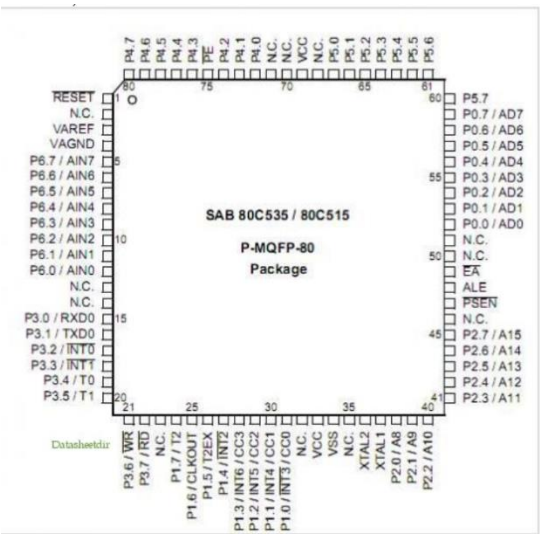
5.2.1. El núcleo de la CPU

El procesamiento de las señales está gestionado por el núcleo de la CPU, formado por un microcontrolador (80C515), una memoria RAM adicional de escritura-lectura de 8K (necesaria para almacenar valores de cálculo, valores de adaptación y posibles averías que surjan en el sistema global (diagnóstico)), una memoria EPROM adicional de 16K (en la cual están almacenadas las curvas características del motor y los campos característicos para el control del motor). En CPU'S más modernas, los datos para el bloqueo electrónico del arranque, datos de adaptación y fabricación, memoria de averías y datos

congelados de esta avería, se guardan en una memoria EEPROM (memoria de escritura y lectura no volátil, se borra eléctricamente).

El microcontrolador 80C515 contiene una memoria ROM de 8K de 8 bits, una memoria RAM de 256 bytes, 6 puertos de 8 bit de entrada-salida digitales, un canal serie full-duplex, 12 vectores de interrupción, 8 A/D de 8 bits, 3 timers/contadores de 16 bits y perro guardián con timer de 16bits. La frecuencia de trabajo es de 12MHz, realizando tiempos de instrucción de 1microsegundo.

Figura 7. SAB 80C515/80C535

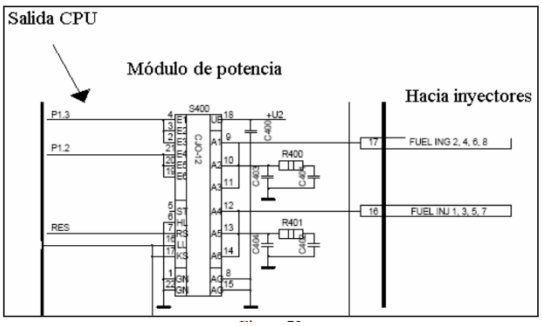


Fuente: Extraído del libro (datasheetdir, 2021)

5.2.2. Las etapas de salida

En la (figura 8) se aprecian dos etapas de salida diferenciadas. La etapa de salida o potencia (1) se encarga de proporcionar las señales de activación a los inyectores.

Figura 8. La etapa de salida o potencia

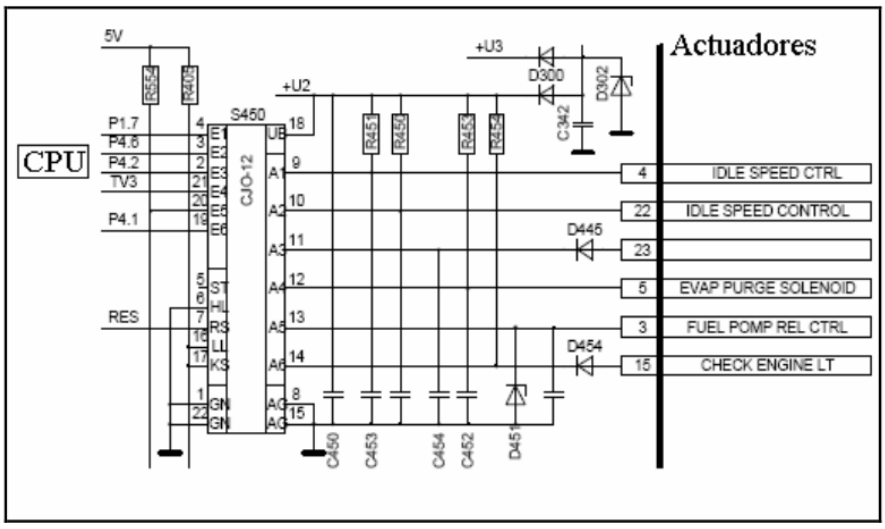


Fuente: Extraído del libro (BOSCH, 2002)

En esta etapa se encuentra adosada a la chapa refrigeradora el transistor de potencia (1) que se encarga de la activación del relé del control del motor

La etapa de potencia (2) (figura 8) se encarga del control de la activación de actuadores como: la electroválvula regeneradora de emisiones de evaporación de combustible, relé de la bomba de combustible, relé del calentador del sensor de oxígeno y la electroválvula de aire adicional. En esta etapa se encuentra adosada a la chapa refrigeradora el transistor de potencia (2) que se encarga de conmutar a masa la salida del primario de la bobina de encendido.

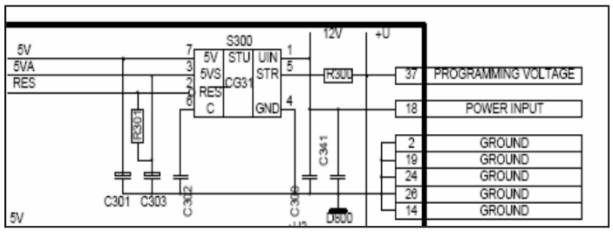
Figura 9. Control de la activación de actuadores



Fuente: Extraído del libro (BOSCH, 2002)

La fuente de alimentación (figura 10) tiene la misión de suministrar el voltaje de 5v necesario para la alimentación de todos los componentes IC digitales, la CPU, memorias, etc. Un voltaje de programación necesario para las memorias externas y los 12 v filtrados para la circuitería de preparación de las señales de entrada y etapas de potencia de salida.

Figura 10. Fuente de alimentación



Fuente: Extraído del libro (BOSCH, 2002)

El componente ASIC que se puede ver en la foto de la (figura 2) es un circuito integrado específico para esta ECU que cumple funciones de control y preparación de las señales de entrada.

- **Sistema de diagnostico a bordo (obd2)**

OBD II es la segunda generación de sistemas de Diagnóstico a Bordo (OBD). El sistema se incorpora dentro de las computadoras de todos los vehículos nuevos desde 1996. Se incluye en vehículos de pasajeros, camionetas pick up y vehículos deportivos.

El sistema OBD II monitorea cualquier componente que pueda afectar el desempeño del sistema de control de emisiones; esto asegura que el vehículo se mantenga amigable con el ambiente durante su vida útil. Si un problema es detectado, el sistema OBDII encenderá una luz de advertencia en el tablero para dar a entender al conductor que existe una falla y que se requiere un servicio para corregir el problema. “El sistema OBDII almacena la información con lo cual el técnico podrá con precisión encontrar y corregir el problema” [2].

5.2.3. Componentes

Fundamentalmente el sistema de diagnostico a bordo de segunda generación está compuesto por el Modulo de Control del Motor (PCM) conocida popularmente como computadora del vehículo, la Luz Indicadora de Fallas (MIL), la cual se encuentra en el tablero, y el Conector de Diagnostico (DLC), el cual nos permite la comunicación con la PCM

5.2.4. Modulo de Control del Motor

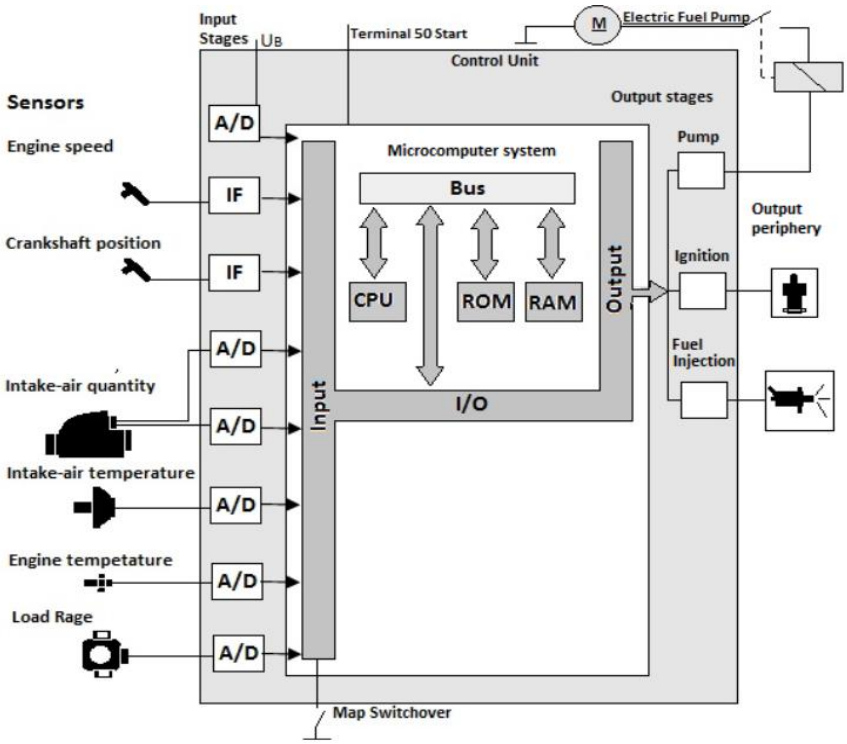
El modulo PCM (Power trainControl Module) es el cerebro que controla el motor. Antes estos computadores eran llamados ECU (Unidad de Control del Motor) y ECM (Modulo Electrónico de Control), pero con la llegada del sistema OBD2 y la asignación de nuevas tareas se le llama Control del Tren de Potencia o PCM. Sin este computador ningún vehículo moderno podría cumplir con las regulaciones modernas de emisiones y eficiencia, manteniendo un bajo consumo de combustible sin afectar el rendimiento. Este computador constantemente ajusta el tiempo de encendido y el ancho de pulso de los inyectores mientras el vehículo se encuentre en operación, estas decisiones la toma basada en una comparación entre un flujo de datos proveniente de varios sensores y un mapa tridimensional incluido en el programa de operación del módulo. Este mapa contiene dos tipos de información: fija y variable. La información fija incluye el sistema operativo del computador y constantes del vehículo como el número de cilindros, el tipo de sistema de emisiones que posee el vehículo y rangos de operación. La información variable es la que se obtiene durante la operación del vehículo como la velocidad, flujo de entada

[2] Que es obd2 [En línea].-11 de Diciembre de 2009.-http://www.mzcomer.com/QUE%20ES%20OBDII.pdf.

de aire, posición del acelerador, tiempo de encendido, temperatura del motor y toda la información obtenida a través de los diferentes sensores.

La función del PCM se divide en tres partes de acuerdo a la figura 11: etapa de entrada, procesamiento y etapa de salida. La etapa de entrada convierte las señales de entrada de todos los sensores a un formato digital que el PCM requiere para su almacenamiento y su posterior procesamiento. En la etapa de salida se encuentran las señales de control de los actuadores, las cuales pueden ser digitales como la chispa de ignición, el pulso de control de los inyectores, el encendido del ventilador, el control de la válvula del sistema de recirculación de gases y muchos otros sistemas cuya operación es de encendido y apagado. Existen también las salidas de señales analógicas, las cuales posen un convertidor digital a análogo que le permite controlar ciertos actuadores que lo requieren. Este controlador también requiere de una serie de circuitos integrados que le permita comunicarse a través de una gran variedad de estándares de comunicación usados actualmente en los vehículos modernos como elprotocolo CAN (Controller Area Network) creado por la proliferación de vehículos con varias unidades de control ubicadas en diferentes áreas, y la cuales necesitan comunicarse constante metea través de un bus de datos que opera baja el protocolo de comunicación CAN. (Jim Moore, 2008)

Figura 11. Módulo de Control del Motor



Fuente: Extraído del libro (Jim Moore, 2008)

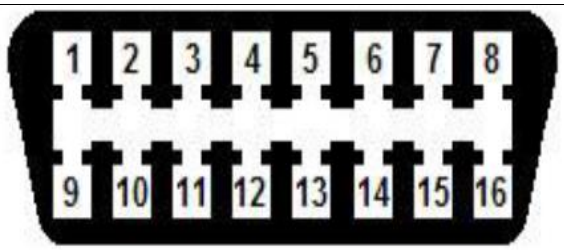
5.2.5. Luz indicadora de falla (MIL)

La luz indicadora de falla, mejor conocida como MIL(Malfunction Indicator Light) por sus siglas en ingles cumple un propósito muy importante a la hora del control de emisiones, alertando al conductor de la necesidad de una revisión del vehículo. La luz MIL se enciende a causa de un código de falla permanente que se ha almacenado y que requiere de la atención inmediata del conductor. La luz MIL posee dos tipos de alerta: una es cuando la luz se enciende continuamente, lo cual significa que se requiere una revisión inmediata del vehículo y cuando la luz se enciende intermitentemente lo cual significa que se requiere de una revisión en el corto plazo. Estos estados de la luz MIL son directamente proporcionales al código de falla que lo genera, si el código es de gran importancia como el relacionado con el control de emisiones se requerirá de servicio inmediato y si es un código del chasis o de un sistema secundario, generará un estado intermitente o no se dará esta alerta. (Henderson Bob Haynes John, 2006).

5.2.6. Conector de Diagnostico (DCL)

El DLC (Diagnostic Link Connector) se encuentra en el compartimento de los pasajeros, generalmente en la parte baja del tablero de instrumentos, al lado del conductor y es el medio por el cual los scanners se conectan físicamente y establecen comunicación con el PCM. El DLC es también conocido como el conector J1962. Una designación tomada del número asignado al estándar dado por la Asociación de Ingenieros Automotrices (SAE), que también establece la configuración y función de cada uno de los pines y que se muestran en la siguiente tabla.

Tabla 7. Conector de diagnostico (DLC)

	1. A discreción del fabricante 2. SAE J1850 line (BUS +) Línea positiva 3. A discreción del fabricante 4. Tierra (chasis) 5. Señal de tierra 6. Bus de datos CAN, Alto -ISO 15765-4 7. Línea-K -ISO 9141-2/ISO 14230-4 8. A discreción del fabricante 9. A discreción del fabricante 10. SAE J1850 (BUS +) Línea negativa 11. A discreción del fabricante 12. A discreción del fabricante 13. A discreción del fabricante 14. Bus de datos CAN, Baja-ISO 15765-4 15. Línea-L 16. Positivo de la batería

Fuente: Extraído del libro (John, Henderson Bob Haynes, 2006)

5.2.7. Protocolos

Básicamente existen 3 protocolos de comunicación del sistema OBDII. Y los fabricantes han escogido un único protocolo que será utilizado en todos los vehículos que fabrican.

- ISO 9141-2:Se encuentra en vehículos europeos, asiáticos y Chrysler con variantes
- SAE J1850 VPW: Que significa Ancho de Pulso Variable (Variable Pulse Width) y lo utiliza GM USA (General Motors).
- SAE J1850 PWM: Que indica Modulación Ancho de Pulso (Pulse Width Modulatión) utilizado por Ford USA.
- KWP 1281 y KWP 2000: Utilizado por el grupo VAG.
- ISO 14230: Utilizado por Renault.
- ISO 15765-4: Protocolo desarrollado por Bosch para la industria automotriz, aeroespacial, industrial y de equipo médico. Conocido ampliamente como Controller Area Network (CAN). Permite la comunicación entre dispositivos sin una computadora host. La velocidad máxima de transferencia es de hasta 1Mbit/s o 1 000 000 baudios en redes con distancia menor a los 40 metros de distancia. Al disminuir la velocidad de transmisión aumenta la distancia entre dispositivos. CAN sigue el modelo OSI de redes computacionales; con la capa de datos (LLC y MAC) y unos aspectos de la capa física. CAN fue estandarizado en 1986, y a partir del 2008, todos los automóviles livianos utilizarán este protocolo sustituyendo gradualmente a los demás. (Ortiz J, 2014, p.9)
- SAE J1939: Protocolo basado en CAN, originalmente utilizado en la industria de camiones pesados. Actualmente este se utiliza en gran variedad de aplicaciones propulsadas por motores diesel, entre ellos vehículos de carretera, todo terreno, propulsión marina, bombas industriales y generación de energía. SAE J1939 ha sido adoptado por muchos fabricantes de motores diesel por la creciente demanda de control de emisión de emisiones. (Ortiz J, 2014, p.9)

Cada uno de estos protocolos, requiere de un tratamiento de la información diferente, antes de conectar el OBD2con el computador personal, y por tanto, se requieren interfaces de conexión diferentes. Existe la posibilidad de utilizar una interfaz multiprotocolo, capaz de utilizar todos los protocolos e incluso seleccionar automáticamente cual es el protocolo utilizado por el vehículo a conectar.

5.2.8. Estándares

Los fabricantes de vehículos y de herramientas de escaneo se rigen por los siguientes estándares propuestos por la SAE (Asociación de Ingenieros Automotrices) y la ISO (International Organization for Standarizartion) para la interfaz de diagnóstico OBD2 (Henderson Bob Haynes John, 2006) :

- J1962 SAE, Este estándar define la conexión física usada en la interfaz de OBD2.
- J1850 SAE, Estándar para la interfazde red de comunicación clase B.
- J1939SAE, ISO 11898 Estándar para el envío de datos por un bus CAN en vehículos pesados.
- J1978 SAE, Estándar para las herramientas de diagnóstico en OBD2.
- J1979 SAE, Estándar para los test de diagnóstico del sistema OBD2.
- J2012 SAE, estándar para los reportes de los test de emisiones.
- J2178-1 SAE, Estándar de mensajes para redes de comunicaciones clase B: dirección y formato.
- J2178-2 SAE, Estándar de mensajes para redes de comunicaciones clase B: parámetros de datos.
- J2178-3 SAE,Estándar de mensajes para redes de comunicaciones clase B: Identificación de cuadros de datos de un byte.
- J2178-4 SAE,Estándar de mensajes para redes de comunicaciones clase B: Definiciones de mensajes. (OBDII Bus, s.f.)

5.2.9. Modos de Operación

El sistema OBD2 permite el acceso a información en tiempo real y almacenado en la PCM agrupada en 9 modos:

Modo 1: Datos de Diagnostico Actuales del Tren de Fuerza

El modo 1 es utilizado para monitorear constantemente los datos de un sensor o actuador. La herramienta de exploración o scanner envía un requerimiento de datos de diagnóstico al PCM (Power Control Module), el cual responde con los datos actualizados identificando el parámetro o de forma de PID's.

Modo 2: Datos del Cuadro Congelado del Tren de Fuerza

Un cuadro congelado es simplemente un flujo de datos instantáneo relacionado con las emisiones que se genera después de que se almacena un DTC. Esta información se almacena en forma de PIDS. Estos datos son muy útiles para duplicar una falla intermitente, ya que se muestran las condiciones de operación y el técnico puede reproducir las mismas condiciones de manejo.

Modo 3: Códigos de Falla de Diagnostico Relacionados con la Emisiones

Es el modo más útil ya que muestra DTCS ya confirmados, los cuales identifican una falla especifica y repetitiva, Para confirmar un DTC es necesario que la falla se presente durante una serie de ciclos de manejo especifica y será almacenado por un periodo de 40 ciclos de calentamiento después de que la luz indicadora de falla haya sido apagada. Esta información se obtiene a través de un requerimiento de la herramienta de exploración o scanner.

Modo 4: Borrado de la Información de Diagnostico Relacionada con las Emisiones

Este modo se utiliza para borrar los DTCS pero también borra los ciclos de calentamiento, los datos del cuadro congelado y los datos del modo 6, lo cual puede impedir la aprobación enuna eventual verificación de emisiones.

Modo 5: Resultados de las Pruebas de Monitoreo del Sensor de Oxigeno

Este modo muestra los resultados de las pruebas del sensor de oxigeno, que tambiénestán disponibles en el modo 6. Algunas herramientas de exploración muestran los datos en tiempo real que dependiendo del vehículo y fabricante puede estar o no disponible.

Modo 6: Resultados de las Pruebas de Monitoreo Abordo Para Sistemas Específicos Monitoreados

El modo 6 son datos codificados que se utilizainternamente por el PCM para determinar el estado de la luz indicadora de fallas. No son valores en tiempo real, pero muestran una serie de resultados de pruebas realizadas.En algunos vehículos de última generación con protocolo CAN puede variar el contenido del modo 6.

Modo 7: Códigos de Falla de Diagnostico Relacionados con las Emisiones Detectados Durante el Último o Actual Ciclo de Manejo Completado

Estos son los resultados de las pruebas en un vehículo que está siendo conducido durante un ciclo de manejo. Cualquier DTC pendiente relacionado con el sistema de emisiones, es reportado en este modo.Para pasar al modo 3 de DTCS confirmados, se pueden requerir hasta 3 ciclos de manejo continuo con la misma falla.

Modo 8: Control de la Prueba, Del Sistema de Monitoreo Abordo

En este modo se requiere que la herramienta de exploración realice una prueba especialque active un dispositivo o realice una verificación dinámica y dependiendo de esa pruebael PCM puede posteriormente reportar los DTCS.

Modo 9: Requerimiento de la Información del Vehículo

Este es un requerimiento de mostrar el número de identificación del vehículo o VIN, la identificación del PCM y las calibraciones del software almacenados en el PCM. (Jim Moore, 2008)

5.2.10. Funcionamiento del Sistema

El sistema de diagnóstico a bordo OBD2 en su programación tiene un comportamiento especifico que permite el monitoreo y la ejecución de rutinas de control que se encargan de vigilar el funcionamiento de todos los componentes del vehículo y generan códigos de fallas alertando de esta forma al usuario que existen averías de distintos grados de severidad.

A) Códigos De Falla DTC (Data Trouble Codes)

Un código de falla es una herramienta que permite la identificación de un problema específico. Los códigos de fallase pueden generar a través de diferentes pruebas llamadas monitores, los cuales verifican el funcionamiento de cada uno de los sistemas en cada viaje del sistema y se almacenan en el modo 3 del sistema OBD2.

Cada código de falla posee una identificación del sistema al cual pertenece, un número que indica si es un código genérico o especifico de un fabricante y un número de identificación de la falla.

El estándar SAE J2Q12 define un código de 5 dígitos como se observa en la figura siguiente, en el cual cada dígito representa un valor predeterminado. Todos los códigos son presentados de igual forma para facilidad del mecánico. Algunos de estos se definen por este estándar y otros son reservados para uso de los fabricantes.

El código tiene el siguiente formato YXXXX (ej, P0308)

Donde Y, el primer dígito, representa la función del vehículo:

- P -Electrónica de Motor y Transmisión (Powertrain)
- B -Carrocería (Body)
- C -Chasis (Chassis)
- U -No definido (Undefíned)

El segundo dígito índica la organización responsable de definir el código,

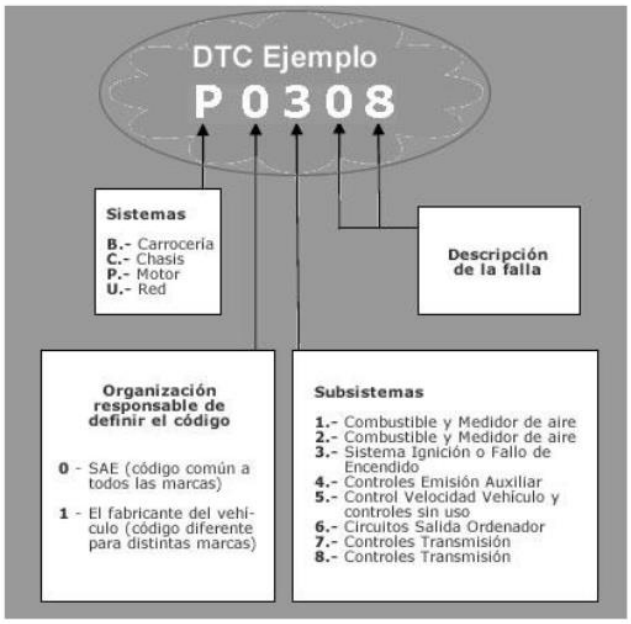
- 0 -SAE (código común a todos las marcas)
- 1 -El fabricante del vehículo (código diferente para distintas marcas)

El tercer dígito representa una función específica del vehículo:

- 0 -El sistema electrónico completo
- 1 y 2 -Control deaire y combustible
- 3 -Sistema de encendido
- 4 -Control de emisión auxiliar
- 5 -Control de velocidad y ralentí
- 6-ECU y entradas y salidas
- 7 -Transmisión

El cuarto y quinto dígito están relacionados específicamente con la falla. Entonces el código P0308 indica un problema en la electrónica de motor (P), definido por SAE (0) y común a cualquier vehículo, relacionado con el sistema de encendido (3), y falla en el cilindro número 8 (08).

Figura 12. Estructura de un código de error.



Fuente: Extraído del libro (TOOL, 2021)

B) Monitores Continuos y no Continuos

Los monitores son rutinas de control y verificación que se encuentran programadas en la unidad de control electrónico y se encargan de vigilar el funcionamiento de todos los componentes del sistema. Estos se dividen en monitores continuos y nocontinuos:

Tabla 8. Monitores Continuos

MONITORES CONTINUOS	
Se les llama monitores continuos porque se están ejecutando constantemente, bajo toda clase de condiciones y son los siguientes	
Monitor de Falta de Chispa	Este monitor realiza pruebas verificando la velocidad del cigüeñal para detectar fallas en cada una de las levas del mismo. Este monitor es tan preciso que detecta el cilindro que está fallando y define la gravedad de la falla
Monitor del Sistema de Combustible	El monitor del sistema de combustible es una prueba de dos viajes y es uno de los monitores más importantes y de alta prioridad. Este monitor controla la entrega de combustible necesario a corto y largo plazo. Si la cantidad de combustible entregada es poca o demasiada, se generará un código de falla con su respectivo cuadro de datos según su prioridad
Monitor de los componentes compresivo	El monitor de los componentes compresivos agrupa varios monitores, los cuales se ejecutan constantemente. Algunos de estos monitores son el de la posición del cigüeñal, posición del árbol de levas, Temperatura del refrigerante del motor y sensor de la masa de aire. Este monitor verifica los rangos de las señales de salida y busca corto circuitos y circuitos abiertos en cada uno de los sensores

Fuente: Extraído del libro (Jim Moore, 2008)

Tabla 9. Monitores No Continuos

MONITORES NO CONTINUOS	
Los monitores no continuos son aquellos que se ejecutan durante un viaje del sistema, bajo condiciones específicas de velocidad, aceleración, tiempo de manejo y varían de acuerdo al monitor de cada sistema. Los monitores más comunes en toda clase de vehículos, son los siguiente:	
Monitor de Eficiencia del catalizador:	Esta estrategia monitorea los dos sensores de oxigeno, comparando la concentración de oxigeno presente en los gases de escape antes y después del convertidor catalítico. La cantidad de oxigeno presente en los gases de escape después de haber pasado por el catalizador, es un indicador del funcionamiento deeste ya que el catalizador utiliza el oxigeno presente en la reacción química.
Monitor del Sensor de Oxigeno:	Este monitor consta de dos pruebas, que involucran a los dos sensores de oxigeno. Las pruebas consisten en el seguimiento a la señal de cada uno de los sensores y la segunda es la verificación del funcionamiento de los elementos calefactores cuya función es mantener cada uno de los sensores a una temperatura de 675 grados centígrados. Una prueba que hace el sistema es fijar un ancho de pulso predeterminado para el control de los inyectores, verificando el tiempo y voltaje de respuesta de cada sensor de oxigeno
Monitor del Sistema EGR	El monitor del sistema de recirculación de los gases de escape tiene una prioridad alta, ya que este sistema controla la temperatura de la cámara de combustión, evitando que alcance temperaturas por encima de los 1371 grados centígrados ya que a temperaturas superiores el nitrógeno se combina con el oxigeno para formar Oxido Nitroso (NOX) y Dióxido de Nitrógeno (NO2), los cuales son dos gases nocivos para el medio ambiente ya que combinados con los hidrocarburos forman smog fotoquímico
Monitor del Sistema EVAP	El monitor del sistema de emisiones evaporativas, verifica el funcionamiento

	del sistema evaporativo del tanque de combustible. Los vapores de combustible del tanque son almacenados en un deposito que contiene carbón activo y el monitor lo que hace es verificar el sello de todo el sistema con una bomba detectora o con valores estrictos de fuga, los cuales en el sistema de diagnostico abordo no pueden superar las 0.020 pulgadas de diámetro y más de 1.35 gramos de hidrocarburos por milla
Monitor AIR	El sistema de inyección de aire secundario controla la cantidad de combustible no quemado en el escape, ya que durante la fase de encendido en frio se incrementa la cantidad de hidrocarburos en el convertidor catalítico. El sistema de aire secundario ayuda a la post oxidación, creando un efecto de post combustión e inyectando oxigeno adicional. El monitor verifica la respuesta de los sensores de oxigeno, la cual consiste en inyectar aire antes del primer sensor, produciendoun cambio de la concentración de gases en el catalizador y un cambio en la señal de respuesta de los sensores

(Jim Moore, 2008)

5.2.11. Ciclo de Manejo

El Ciclo de Manejo del sistema OBD2 es un nombre dado a una serie de pruebas que el sistema de diagnóstico abordo realiza para verificar el funcionamiento de los diferentes dispositivos que conformas los sistemas del vehículo. Estas pruebas comprenden diferentes modos de conducción. Durante el Ciclo de Manejo se requiere operar el vehículo hasta un punto en el cual entre en lazo cerrado, es decir hasta que el vehículo complete su ciclo de calentamiento, alcanzando su temperatura estándar de operación además de otras condiciones de manejo requeridas para verificar una o más fallas. La secuencia de las pruebas puede variar ligeramente con cada fabricante. Sin embargo, los resultados deben ser similares para asegurar que un problema pueda identificar sea pesar del fabricante. (Jim Moore, 2008).

En un ciclo de manejo se debe realizar un diagnóstico de todos los sistemas. Normalmente toma alrededor de 15 minutos y requiere de los siguientes pasos:

1. Arranque en frío. El motor debe estar a menos de 50°C y con una diferencia máxima de 6°C con respecto a la temperatura ambiente.

2. Ralentí. El motor debe andar por 2 minutos y medio con al aire acondicionado y el desempañado trasero encendidos. A mayor carga eléctrica mejor. Esto prueba el calentador de los sensores de oxígeno, purga del canister, falla de encendido y el ajuste de combustible (Fuel trim).
3. Aceleración. Acelerar hasta alcanzar los 85 km/h con el aire acondicionado y demás cargas eléctricas apagadas.
4. Velocidad constante. Mantener la velocidad constante a 85Km/h durante 3 minutos. Durante este periodo se prueba la respuesta de los sensores de oxigeno, el sistema de recirculación de los gases de escape (EGR)y el ajuste de combustible.
5. Desaceleración. Se debe soltar el acelerador, y sin pisar el freno se debe dejar que el vehículo disminuya la velocidad gradualmente hasta alcanzar los 30 Km/h.
6. Aceleración. Acelerar a 3/4 del pedal hasta alcanzar de 85 a 95 Km/h.
7. Velocidad constante. Mantener la velocidad constante a 85 Km/h durante 5 minutos. Esto probara lo mismo del paso 4 y el catalizador.
8. Desaceleración. Soltar el pedal del acelerador, sin pisar el freno o embrague. Con este paso se deben completar todas las pruebas a los diferentes sistemas

5.2.12. Viaje del Sistema

Un viaje del sistema OBD2 consiste de la operación del vehículo, seguido de un periodo de tiempo con el motor apagado, teniendo en cuenta que se necesita el tiempo suficiente para que se complete al menos uno o más monitores. Un viaje se completa cuando se cumplen una serie de criterios de habilitación para un procedimiento de diagnóstico dado. Cada monitor requiere de diferentes criterios para poder realizarse, que dependen del modo en que se conduzca el vehículo en aspectos como el calentamiento del motor, velocidad crucero, aceleración, desaceleración y un cierto requerimiento de tiempo en cada modo.

El propósito es verificar que todos los componentes relacionados con las emisiones estén funcionando adecuadamente, como una forma de inspección y mantenimiento, con el fin de generar códigos de falla que permitan evitar incrementos de emisiones por un tiempo prolongado. (Jim Moore, 2008).

5.3. Protocolo de Comunicación Serial

La comunicación serial de datos es el protocolo más comúnmente utilizado para la comunicación entre dispositivos. (Comunicación Serial, 2021) Todos los computadores poseen puertos seriales RS-232 que permiten la comunicación con otros computadores o dispositivos como impresoras scanner y otros dispositivos de adquisición de datos. (Tutorial del Protocolo RS-232, 2010).

La comunicación serial envía y recibe información de a un bit, hasta completar todos los bits de un dato, a diferencia de la comunicación en paralelo que permite la transmisión de un byte completo pero tiene la desventaja de utilizar una gran cantidad de líneas por lo que es más costoso y tiene la desventaja de que a largas distancias se atenúa la

señal, debido a la capacitancia entre los conductores y a los parámetros distribuidos. Por ejemplo la especificación IEEE 488 para la comunicación en paralelo especifica que la distancia máxima de trasmisión en una comunicación en paralelo es de 20 metros, y no más de 2 metros entre equipos. A diferencia de la comunicación serial que puede llegar a ser a una distancia de 1200 metros. (Comunicación Serial, 2021)

Normalmente la comunicación serial se realiza a través de tres líneas: transmisión de datos, recepción de datos y tierra. Existen dos tipos de transmisión serial, síncrona y asíncrona, en la comunicación asíncrona es posible recibir y transmitir simultáneamente y la duración de cada bit está dada por la velocidad de transmisión. Normalmente cuando no se está transmitiendo información, la línea permanece en estado alto y cuando se quiere iniciar la transferencia de datos se coloca en bajo por un tiempo determinado, esto se conoce como bit de arranque. Después de transmitir cada uno de los bits del menos significativo al más significativo se coloca la línea de nuevo en alto un tiempo determinado antes de transmitir otro byte de información. Para realizar una comunicación síncrona, existen otras líneas de intercambio de pulsos de sincronización, los cuales indican cuanto un byte es válido. Los parámetros más importantes en la comunicación serial son:

Velocidad de transmisión (Baud rate): La velocidad de transmisión se define como el númerode bits por segundo que se envían. Por ejemplo una comunicación a 300 baudios significan 300 bits por segundo.

Bits de datos: Este parámetro se refiere al número de bits que se enviaran por paquete. El numero bits por paquete no siempre es de la longitud de un byte, pueden ser menos o más bits. Los bits de todos los determina el tipo de información que se envía. Un paquete se refiere a la transferencia de los bits de datos, bits de inicio/parada y paridad. Bits de parada: Se refiere al el bit que indica el fin de la comunicación de un paquete. Debido a que cada equipo posee su propio reloj, es posible que no estén sincronizados. Por lo que los bits de parada también proporcionan un margen de tolerancia.

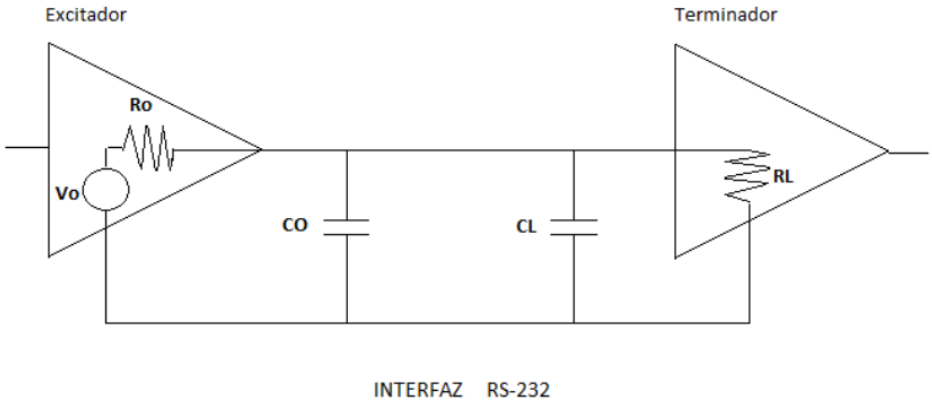
Paridad: El bit de paridad sirve para verificar si hay errores en la comunicación serial. Existen tres tipos de paridad: par, impar y espaciada. En la paridad par e impar, la comunicación serial fijara un bit en un valor específico después de los bits de datos, para asegurarse que el dato tenga un número par o impar de bits en alto. (Comunicación Serial, 2021).

5.3.1. Interfaz RS-232

La norma RS-232 especifica un conector DB25P/DB25S. Las características eléctricas de la conexión que se observan en la figura siguiente, especifican una capacitancia de 2500pF, que incluye la capacitancia de la línea de transmisión. La

impedancia en el extremo de la línea de transmisión debe ser de 3000Ohms a 7000 Ohms y la impedancia de salida mayor a 300 Ohms. Teniendo en cuenta estas especificaciones eléctricas y con un límite de la velocidad de transmisión de 20.000 bits por segundo o 20.000 baudios, la longitud máxima de la línea de transmisión en la interfaz RS-232 es de unos 15 metros o 50 pies. (Tomasi Wayne, 2003).

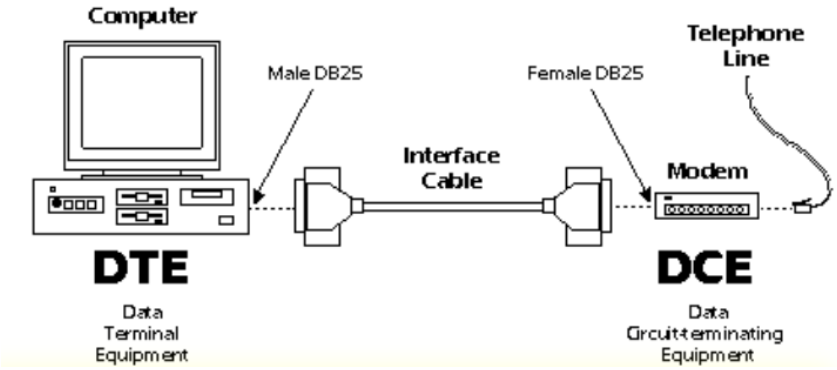
Figura 13. Especificaciones eléctricas de la interfaz RS23



Fuente: Extraído del libro (Tomasi Wayne, 2003)

Como está definido en la norma, el quipo ubicado al final de la conexión tiene el nombre de DTE (Data terminal equipment), y posee un puerto macho. El equipo más cercano a la línea es llamado DCE (Data circuit terminal equipment) y posee un puerto hembra como se muestra a continuación en la figura:

Figura 14. Interfaz RS232



(Tutorial del Protocolo RS-232, 2010)

La norma de la interfaz RS-232 también especifica los valores de voltaje que se envían por la línea de transmisión por el DTE y el DCE. Ambos equipos el DTE Y DCE poseen circuitos que garantizan los valores de voltaje especificados por la norma, convirtiéndolos a valores RS-232. Por ejemplo, si un DTE usa lógica TTL y se comunica con un DCE que usa lógica ECL, sus valores lógicos internos no son compatibles. Los circuitos de nivelación de voltaje internos en cada equipo convierten esos valores internos a los especificados por la norma. Un nivelador se llama excitador si produce un voltaje de señal al cable, y un terminador si acepta un voltaje de señal del cable.

Tabla 10. Especificaciones de voltaje DC a RS-232

	Terminales de Datos		**Terminales de Control**	
	-1 Logico	0 Logico	Activar ON	Desactivar Off
Excitador	-5 a +15	+5 a +15	+5 a +15	-5 a -15
Terminador	-3 a -25	+3 a +25	+3 a +25	-3 a -25

Fuente: Extraído del libro (Tomasi Wayne, 2003)

Se puede ver en la tabla 10 que los límites para un excitador son más incluyentes que los de un terminador. El excitador puede producir cualquier voltaje entre +5 y +15, o -5V y -15Vcd, y in terminador acepta cualquier voltaje entre +3V y +25V, y -3V y -25Vcd. Esta diferencia entre los valores de voltaje entre el excitador y terminador es llamada margen de ruido. Este margen de voltaje reduce la susceptibilidad de la interfaz al ruido en la línea de transmisión. Los voltajes normales que se usan en las señales de datos y de control son +-7Vcd y +-10Vcd

Las terminales del cable de interfaz RS-232 se clasifican funcionalmente como tierra, datos, control o de sincronización. Todas las terminales que se observan en la tabla 11, de la interfaz RS232 son unidireccionales, esto quiere decir que las señales solo se propagan del DTE al DCE o viceversa.

Tabla 11. Designaciones de Terminales en EIA RS-232

Terminal	**Nomenclatura EIA**	**Acrónicos Comunes**	**Dirección**
1	Tierra de protección (AA)	GWG	Ninguna
2	Datos transmitidos (BA)	TD, SD	DTE a DCE
3	Datos recibidos (BB)	RD	DCE a DTE
4	Petición de transmitir (CA)	RS, RTS	DTE a DCE
5	Borrar para transmitir (CB)	CS, CTS	DCE a DTE
6	Conjunto de datos listos (CC)	DSM, MR	DCE a DTE
7	Tierra de señal (AB)	GND	Ninguna

8	Detección de señal recibida en línea (CF)	RLNS, CD	DCE a DTE
9	Sin asignar		
10	Sin asignar		
11	Sin asignar		
12	Detección de señal recibida en línea (SCF)	SRLSD	DCE a DTE
13	Borrar secundaria para transmitir (SCB)	SCS	DCE a DTE
14	Datos secundarios transmitidos (SBA)	STD	DTE a DCE
15	Sincronización de elemento de señal de transmisión (DB)	SCT	DCE a DTE
16	Datos secundarios recibidos (SBB)	SRD	DCE a DTE
17	Sincronización de elementos de señal en receptor (DD)	SCR	DCE a DTE
18	Retroalimentación local	LL	DTE a DCE
19	Petición secundaria para transmitir (SCA)	SRS	DTE a DCE
20	Terminal de datos lista (CD)	DTR	DTE a DCE
21	Detector de calidad de señal (CG)	SQD	DCE a DTE
22	Indicador de timbrado (CE)	RL	DCE a DTE
23	Selector de rapidez de señal de datos (CH)	RI	DTE a DCE
24	Sincronización de elemento de señal de transmisión (DA)	DSRS	DTE a DCE
25	Modo de prueba	SCTE	DCE a DTE

Fuente: Extraído del libro (Tomasi Wayne, 2003)

5.4. Herramientas Tecnológicas

5.4.1. Bus Can MCP 2515

El MCP2515 cuenta con dos máscaras de aceptación y seis filtros de aceptación que se utilizan para descartar mensajes no deseados, lo que reduce la sobrecarga del micro controlador anfitrión. Este es el responsable de la recepción y transmisión de tramas de datos en el sistema. (ALEXANDER, 2020).

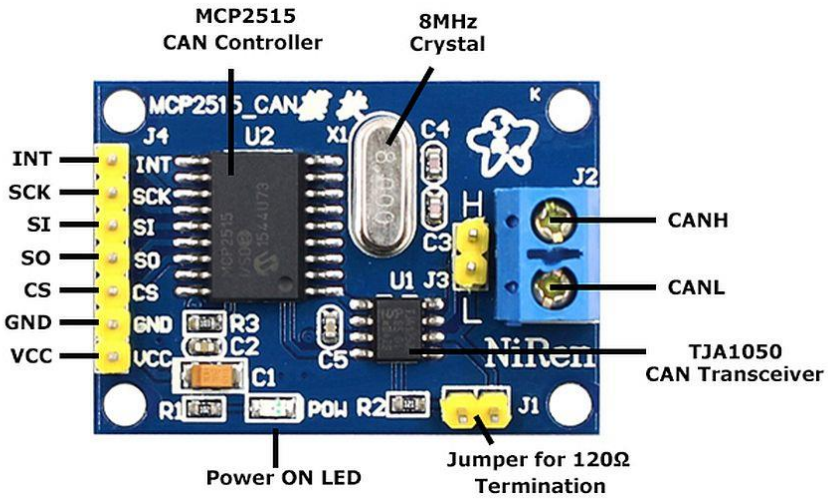
Este módulo permite usar CAN BUS (Controller Area Network BUS) a través del SPI (Serial Periferal Interface) de Arduino. CAN BUS es un estándar diseñado para comunicar un microcontrolador y sus periféricos. (Liu, JM,Wang,JH y Sun, 2014)

Tabla 12. Características De Bus CAN MCP2515

Soporta CAN V2.0B
Velocidad de comunicación: hasta 1Mb/seg
Soporta SPI
Alimentación: 5V
Consumo: 5mA
Tamaño: 4 x 2,28cm

Fuente: Extraído del proyecto (Liu, JM,Wang,JH y Sun, 2014)

Figura 15. Bus CAN MCP2515



Fuente: Extraído del sitio web (htt2)

5.4.2. Pantalla LCD monocromática 16x2

La pantalla de visualización se utiliza para presentar los datos procesados por el Arduino Uno, existen diferentes dispositivos que permiten la visualización como contadores, matrices de leds y pantallas. En la tabla 13 se presentan los más utilizados.

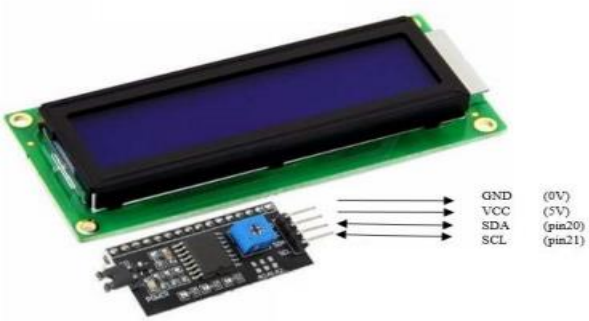
Tabla 13. Dispositivos de visualización

Dispositivo	Resolución	BUS	Panel Táctil	Color	Memoria	Costo
LCD Oled	128x64 pixeles	4 bits	NO	Monocromático	NO	
LCD 16x2 12C	32 caracteres	4 bits	NO	Monocromático	RAM 80bits	
TFT3.2’	320x 240 pixeles	16 bits	SI	65536	SD hasta 1Gb	

Fuente: Extraído del proyecto (ALEXANDER, 2020)

Para la selección del dispositivo de visualización se analizó la tabla 12 y se tomó en cuenta que la función principal de esta pantalla es presentar el consumo de energía eléctrica acumulado, por ello se utiliza una LCD 16x2 con comunicación I2C que facilita el intercambio de información con Arduino. En la figura 16 se muestra la conexión con Arduino y sus terminales señalizadas.

Figura 16. LCD 16x2 I2C



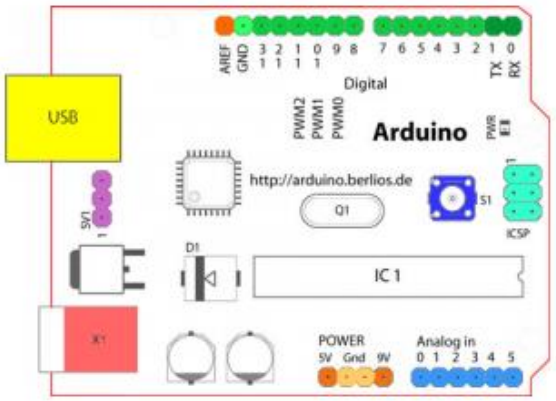
Fuente: Extraído del proyecto (ALEXANDER, 2020)

5.4.3. Arduino

Arduino Uno es una placa electrónica basada en el microcontrolador ATmega328. Cuenta con 14 entradas/salidas digitales, de las cuales 6 se pueden utilizar como salidas PWM (Modulación por ancho de pulsos) y otras 6 son entradas analógicas. Además, incluye un resonador cerámico de 16 MHz, un conector USB, un conector de alimentación, una cabecera ICSP y un botón de reseteado. La placa incluye todo lo necesario para que el microcontrolador haga su trabajo, basta conectarla a un ordenador

con un cable USB o a la corriente eléctrica a través de un transformador. (ARDUINO, 2018).

Figura 17. ARDUINO UNO



Fuente: Extraído del sitio web (ARDUINO, 2018)

Tabla 14. Características técnicas de Arduino Uno

Microcontrolador: ATmega328
Voltaje: 5V
Voltaje entrada (recomendado): 7-12V
Voltaje entrada (limites): 6-20V
Digital I/O Pins: 14 (de los cuales 6 son salida PWM)
Entradas Analogicas: 6
DC Current per I/O Pin: 40 mA
DC Current parar 3.3V Pin: 50 mA
Flash Memory: 32 KB (ATmega328) de los cuales 0.5 KB son utilizados para el arranque
SRAM: 2 KB (ATmega328)
EEPROM: 1 KB (ATmega328)
Clock Speed: 16 MHz

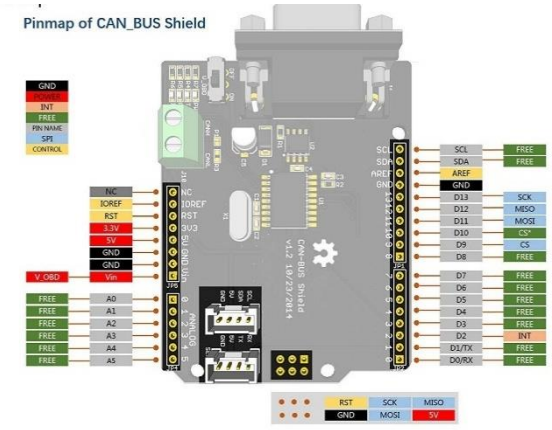
Fuente: Extraído del sitio web (ARDUINO, 2018)

5.4.4. CAN-BUS Shield V2

Para poder realizar una lectura de redes CAN con un programa de Arduino, fue necesario el uso de la tarjeta CAN-BUS Shield que nos permitiese la adquisición y lectura de dichas redes para su posterior procesamiento. La tarjeta CAN-BUS Shield es común encontrarla en buses de diagnóstico de autos. Dicha tarjeta adopta el controlador MCP2515 CAN Bus con interfaz SPI y MCP2551 CAN transmisor-receptor para ser utilizado en Arduino. Implementa CAN V2.0B arriba de 1 Mb/s, interfaz SPI arriba de 10 MHz, dos buffers de

recepción con almacenaje prioritario de mensajes y conector sub-D estándar de 9 pines (FLORES, 2014)

Figura 18. CAN BUS SHIELD V2

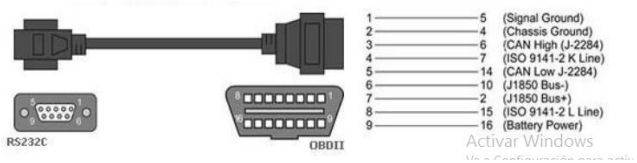


Fuente: Extraído del informe (FLORES, 2014)

5.4.5. Conector OBD II

Este dispositivo se conecta al vehículo a través de un conector RS232C por lo que nos ayuda a obtener datos reales de las fallas que existe en la E.C.U.

Figura 19. Cable de conexión OBDII



Fuente: Extraído del informe (FLORES, 2014)

Tabla 15. Pines del Conector

Pin Number	Signal Name	Signal Description
1	No connected	N/A
2	CAN L	Dominant Low
3	GND	Ground
4	No connected	N/A
5	No connected	N/A

6	GND	Ground
7	CAN H	Dominant High
8	No connected	N/A
9	No connected	N/A

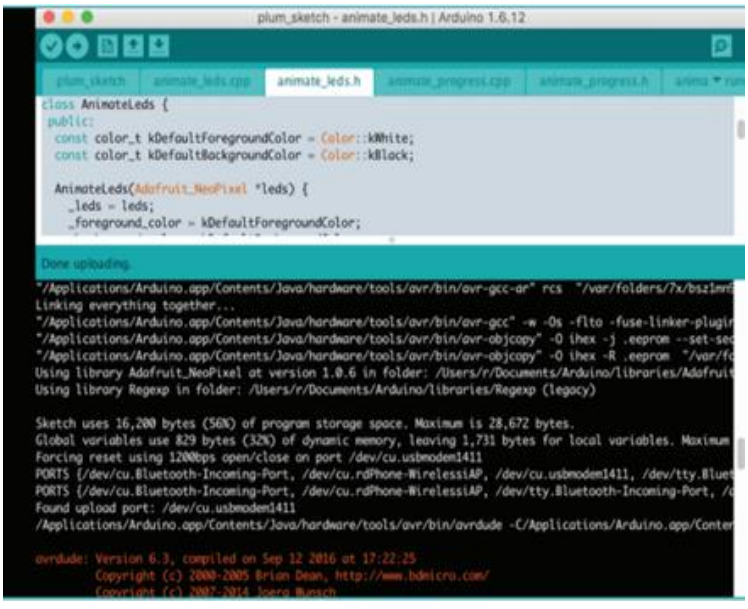
Fuente: Extraído de (can bus, 2011)

5.4.6. Software Arduino

Arduino es una placa de desarrollo basado en microcontrolador atmel es importante precisar que los microcontroladores son circuitos integrados en lo que es posible grabar instrucciones las que deben inscribirse en un lenguaje de programación utilizando un entorno de desarrollo compatible.

Arduino IDE se distribuye en forma gratuita por lo que sólo necesitas Acceder al sitio web oficial de la aplicación para descargar una copia instalable. También se trata de un software que distribuye una licencia libre de modo que es posible Acceder al código Fuente del aire y construir un el instalador desde el organizar las modificaciones que consideres necesarias. (peña, 2020).

Figura 20. Arduino IDE



Fuente: Extraído de (peña, 2020)

5.4.7. Java

Java fue concebido por James Gosling, Patrick Naughton, Chriss Warth, en 1991 el desarrollo de la primera versión duró dieciocho meses y se llamó ¨oak¨ el motivo principal era la necesidad de un lenguaje independiente de la plataforma que se pudiese utilizar para crear software para diversos dispositivos electrónicos, como hornos, microondas y control controles remotos. Sin embargo con la aparición de World wide web Java sido impulsado al frente del diseño de lenguaje de programación ya que la red también exigía programas portables java a está relacionado con c++y que es un descendiente directo de se la mayor parte del carácter de java es heredado por los estudios del lenguaje Java. (Anaya Multimedia/Wrox, 2011).

Figura 21.JAVA

Fuente: Extraído de (Anaya Multimedia/Wrox, 2011)

5.4.8. Apache NetBeans

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos. Actualmente Sun Microsystems es administrado por Oracle Corporation.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las API de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software. (Wikipedia, 2021).

Figura 22. APACHE NETBENS

Fuente: Extraído de (Wikipedia, 2021)

5.4.9. Maven

Es una herramienta open-source, que se creó en 2001 con el objetivo de simplificar los procesos de build (compilar y generar ejecutables a partir del código fuente).

Antes de que Maven proporcionara una interfaz común para hacer builds del software, cada proyecto solía tener a alguna persona dedicada exclusivamente a configurar el proceso de build.

Además, los desarrolladores tenían que perder tiempo en aprender las peculiaridades de cada nuevo proyecto en el que participaban.
Si queríamos compilar y generar ejecutables de un proyecto, teníamos que analizar qué partes de código se debían compilar, qué librerías utilizaba el código, dónde incluirlas, qué dependencias de compilación había en el proyecto. (GARZAS, 2012).

Figura 23. MAVEN

Fuente: Extraído de (GARZAS, 2012)

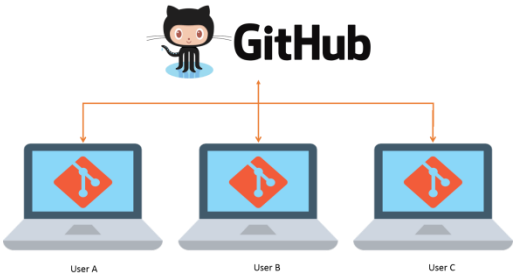
5.4.10. GitHub

GitHub es una compañía sin fines de lucro que ofrece un servicio de hosting de repositorios almacenados en la nube. Esencialmente, hace que sea más fácil para individuos y equipos usar Git como la versión de control y colaboración.

La interfaz de GitHub es bastante fácil de usar para el desarrollador novato que quiera aprovechar las ventajas del Git. Sin GitHub, usar un Git generalmente requiere de un poco más de conocimientos de tecnología y uso de una línea de comando. (kinsta, 2021)

esto nos ayuda esencialmente en la coordinación y verificación de versiones utilizadas y para compartir código.

Figura 24. Github



Fuente: Extraído de (kinsta, 2021)

CAPITULO III: MARCO METODOLÓGICO

1. TIPOS DE INVESTIGACIÓN

El tipo de investigación que se aplicara en el procedente proyecto de investigación es la investigación experimental, que es aquella que se obtiene a través de la experimentación y los compara con variables constantes, a fin de determinar las causas y efectos de los fenómenos en estudio. Además de ser una investigación cuantitativa. Se basa en un protocolo de control, la pretendencia de variables, la manipulación de dichas variables y la observación de resultados cuantificables.

En resumen, la investigación experimental está integrada por un conjunto de actividades metódicas y técnicas que se realizan para recabar la información y datos necesarios sobre el tema a investigar o resolver.

1.1. Características de la Investigación Experimental

La investigación experimental se presenta mediante la manipulación de una variable experimental no comprobada, en condiciones rigurosamente controladas, con el fin de describir de qué modo o por qué causa se produce una situación o acontecimiento particular. Su diferencia con los otros tipos de investigación es que el objetivo de estudio y su tratamiento dependen completamente del investigador, de las decisiones que tome para manejar su experimento. El experimento es una situación provocada por el investigador para introducir determinadas variables de estudio manipuladas por él para controlar el aumento o disminución de esas variables y su efecto en las conductas observadas. En el experimento, el investigador maneja de manera deliberada la variable experimental y luego observa lo que ocurre en condiciones controladas. La experimentación es la repetición voluntaria de los fenómenos para verificar su hipótesis

1.2. Etapas de la Investigación Experimental

Etapas que el investigador debe realizar para llevar a cabo una investigación experimental.

- Presencia de un problema. Para el cual se ha realizado una revisión bibliográfica
- Identificación y definición del problema.
- Definición de hipótesis y variables y la operación de las mismas.
- Diseño del plan experimental. - Diseño de investigación. -Determinación de la población y muestra. -Selección de instrumentos de medición. -Elaboración de instrumentos. -Procedimientos para obtención de datos.

- Prueba de confiabilidad de datos.
- Realización del experimento.
- Tratamiento de datos. Aquí en este punto hay que tener en cuenta que una cosa es el dato bruto, otro el dato procesado y otro, el dato que hay que dar como definitivo.

2. DISEÑO DE LA INVESTIGACIÓN

2.1. Propósito de la Investigación

Como se describo en el primer capítulo del presente trabajo se determinó el tema de investigación:

"DESARROLLO E IMPLEMENTACIÓN DE SCANNER AUTOMOTRIZ PARA EL DIAGNÓSTICO DE POTENCIAS Y TORQUE EN VEHÍCULOS LIVIANOS"

Del mismo se identificó el problema:

¿Cuál será el impacto económico, social y medio ambiental de la implementación de prototipos de scanners automotrices para la medición de potencia y torque en los talleres automotrices y la sociedad en la ciudad de El Alto del departamento de la Paz?

¿Qué tan necesario resulta contar con un escáner que Diagnostique Potencia y Torque en un motor en tiempo real, además que económicamente tenga un costo menor a los utilizados en talleres de la ciudad de El Alto?

Y enmarcando los objetivos de la investigación:

El propósito del presente proyecto de investigación es desarrollar e implementar un prototipo de scanner automotriz a un costo económico, el cual colaborará a la medición y cálculo de potencia y torque con lectura de posibles errores en OBDII, identificando fallas en tiempo real, mejorando el medio ambiente para la sociedad en la ciudad de El Alto del departamento de La Paz.

- Identificar la mejor alternativa para el proceso de comunicación para el desarrollo del prototipo de scanner automotriz.
- Desarrollar un prototipo de simulador OBDII para poder realizar pruebas.
- Desarrollar un prototipo de scanner automotriz para corregir las pérdidas de potencia por desgaste.
- Implementar comunicación entre scanner y vehículo para la lectura de datos de potencia y torque.

- Determinar medidas de torque y potencia a tiempo real.
- Comparar el costo y funcionalidad con los diferentes Scanner existentes en el mercado con el prototipo a desarrollar.
- Evaluar los diferentes dinamómetros utilizados para determinar potencia y torque.
- Socializar el prototipo de scanner automotriz desarrollado.
- Realizar una encuesta sobre uso de Escáner automotrices, dinamómetros en talleres de la ciudad de El Alto.
- Implementar comunicación entre scanner y vehículo para la lectura de datos de potencia y torque.
- Proveer con el conocimiento y datos sobre la necesidad de implementación de escáner para diagnóstico de potencia y Torque en la carrera de ingeniería Autotrónica.

Convergiendo en la formulación de la Hipótesis de investigación:

El prototipo de escáner automotriz desarrollado para la ciudad de El Alto del departamento de La Paz, reducirá el costo de implementación de instrumentos de medición y evaluación en talleres, institutos y otros, además permitirá un diagnóstico de potencia y torque en tiempo real de vehículos livianos bajo el protocolo OBDII.

De acuerdo a las características del proyecto de investigación se elige la estrategia concebida para responder a las preguntas de investigación se elige Experimentos Puros[3].

3. ENFOQUE CUANTITATIVO

[3] Experimentos Puros: Estudio de investigación en que se manipulan los datos. Existen tres requisitos para hacer un buen trabajo experimental. Primer requisito. Es la manipulación intencional de una o más variables dependientes. El grado para medir las variables dependientes son dos: El primer grado-Presencia -Ausencia Segundo grado-Más de dos formas Segundo requisito Es medir el efecto que la variable independiente tiene en la variable dependiente. Tercer requisito Debe cumplir con un control o validez interno de la situación experimental

Señala que, bajo la perspectiva cuantitativa, la recolección de datos es equivalente a medir. (Marcelo M. Gomez, 2006); por otro lado se puede identificar que los estudios de corte cuantitativo pretenden la explicación de una variable desde el punto de vista externa y objetiva.

Su intención es buscar la exactitud de mediciones o indicadores sociales con el fin de generalizar sus resultados a poblaciones o situaciones amplias, Trabajan fundamentalmente con números, el dato cuantificable. (Galeano, 2004).

Con los antecedentes antes mencionados y en vista de que durante las pruebas a los prototipos, se trabajara con información exacta de acuerdo a los protocolos y estándares internacionales, el enfoque de investigación será cuantitativo.

4. VARIABLES DE INVESTIGACIÓN

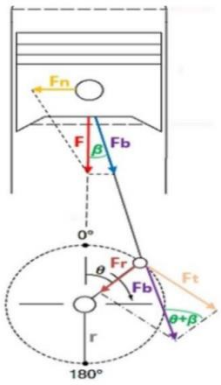
Las variables son características, atributos, cualidades o propiedades que se dan en individuos, grupos u objetos. La validez de una variable depende sistemáticamente del marco teórico que fundamente el problema y del cual va a ser desprendido, y su relación directa con hipótesis que la respalda. Para poder identificar las variables del presente proyecto se tomo en cuenta los siguientes factores:

- La definición nominal de la variable a medir
- La definición real o dimensión de la variable
- La definición operacional o indicador de la variable

4.1. Par Motor (Variable independiente)

La combustión de la mezcla aire-combustible, por acción de la chispa eléctrica, genera un aumento de la presión y temperatura en el interior de los cilindros del motor. Esta presión interior produce a su vez una fuerza de empuje F sobre el pistón que lo desplaza generando el clásico mecanismo de biela-manivela de los motores de combustión interna alternativo, donde el movimiento lineal del pistón en el interior del cilindro se transforma en un movimiento rotatorio del cigüeñal.La fuerza que actúa sobre el pistón es proporcional a la presión media efectiva durante la carrera de expansión (Genta, 2014) ,la misma que depende del grado de llenado de los cilindros, de la relación de compresión, de la cilindrada y de la eficacia con que se desarrolla la combustión. En la siguiente Figura 25, la fuerza F ejercida sobre el pistón se descompone en la fuerza Fb que actúa sobre la cabeza de biela y en la Fn que produce un empuje lateral sobre las paredes del cilindro.

Figura 25. Esquema para la obtención del par motor



Fuente: Extraído de (Genta, 2014)

Por lo tanto el producto de la fuerza F_b por el radio *r* de la manivela determina el par motor, reflejando aritméticamente en la siguiente educación:

$$T = F_b * r \quad \text{Ecuación (1)}$$

Donde se identifica las siguientes variables:

Tabla 16. Variable Toque y variables fuerza, radio

Nro	Tipo de Variable	Variable
1	Dependiente	T: torque o par motor (N.m)
2	Independiente	F_b : Fuerza sobre la biela (N)
3	Independiente	*r:* Radio de la manivela de cigüeñal (m)

Fuente: Elaboracion propia

4.2. Potencia (Variable dependiente)

El concepto de potencia expresa cuantas veces está disponible el par motor en el tiempo, es decir, con qué velocidad se puede disponer del par. La potencia desarrollada por un motor depende de la relación de compresión y de la cilindrada, ya que a mayores valores de estas le corresponde mayor explosión y más fuerza aplicada al pistón; también depende íntimamente de las revoluciones por minuto a las que gira el motor. En consecuencia, la potencia puede ser determinada mediante la Ecuación:

$$P = T * w = \frac{T*n}{\frac{60}{2\pi}} \quad \text{Ecuación (2)}$$

Donde:

P: potencia

T: Torque

w: Velocidad angular

n: revoluciones por minuto

Se identifica las siguientes variables:

Tabla 17 Variable Potencia, Torque, Velocidad angular

Nro	Tipo de Variable	Variable
1	Dependiente	P: Potencia del motor (kW)
2	Independiente	T: Torque o par motor (N.m)
3	Independiente	w: Velocidad angular del eje del cigüeñal (rad /s) o Revoluciones por minuto (n)

Fuente: Elaboracion propia

5. PRUEBA ESTADÍSTICA PARAMÉTRICAS

Las pruebas estadísticas paramétricas, como la de la "t" de Student o el análisis de la varianza (ANOVA[4]), se basan en que se supone una forma determinada de la distribución de valores, generalmente la distribución normal, en la población de la que se obtiene la muestra experimental.

En contraposición de las técnicas no paramétricas, las técnicas paramétricas si presuponen una distribución teórica de probabilidad subyacente para la distribución de los datos. Son más potentes que las no paramétricas.

Dentro de las pruebas paramétricas, las más habituales se basan en la distribución de probabilidad normal, y al estimar los parámetros del modelo se supone que los datos constituyen una muestra aleatoria de esa distribución, por lo que la elección del estimador y el cálculo de la precisión de la estimación, elementos básicos para construir intervalos de confianza y contrastar hipótesis, dependen del modelo probabilístico supuesto.

[4] La técnica de análisis de varianza (ANOVA) también conocida como análisis factorial y desarrollada por Fisher en 1930, constituye la herramienta básica para el estudio del efecto de uno o más factores (cada uno con dos o más niveles) sobre la media de una variable continua. Es por lo tanto el test estadístico a emplear cuando se desea comparar las medias de dos o más grupos. Esta técnica puede generalizarse también para estudiar los posibles efectos de los factores sobre la varianza de una variable

Cuando un procedimiento estadístico es poco sensible a alteraciones en el modelo probabilístico supuesto, es decir que los resultados obtenidos son aproximadamente válidos cuando éste varía, se dice que es un procedimiento robusto.

6. NIVEL DE INVESTIGACIÓN

“La investigación de tipo descriptiva trabaja sobre realidades de hechos, y su característica fundamental es la de presentar una interpretación correcta. Para la investigación descriptiva, su preocupación primordial radica en descubrir algunas características fundamentales de conjuntos homogéneos de fenómenos.

Utilizando criterios sistemáticos que permitan poner de manifiesto su estructura o comportamiento. De esta forma se pueden obtener las notas que caracterizan a la realidad estudiada” (Sabino, 2014).

Por lo descrito en los párrafos anteriores y de acuerdo a la naturaleza de la presente investigación se utilizará el nivel de **investigación descriptivo**, por el trabajo de experimentación sobre hechos y datos reales, de acuerdo a los estándares establecidos en el Protocolo OBDII.

7. POBLACIÓN Y MUESTRA

De acuerdo a las características del presente proyecto de investigación se puede observar que el contar con una población y muestra, representa una inversión, ya que la comunicación de datos de los vehículos hacia lectores externos requiere de la compra de diferentes dispositivos, mismos que pueden afectar el funcionamiento del vehículo si no son operados de manera correcta.

Ante esta eventualidad y precautelando los activos a los que se pueden acceder a través de convenios interinstitucionales, se decidió elaborar un prototipo de emulador de obd2, que permita trabajar con información de igual manera que en un caso real. De esta manera se cumpliría con uno de los objetivos del presente trabajo de investigación, nuestro EmuladorObd2Java, fue desarrollado sobre Arduino Uno, y el lenguaje de programación java, mismo se encuentra en el siguiente enlace:

https://github.com/wilwolf/simuladorobd2javaarduino

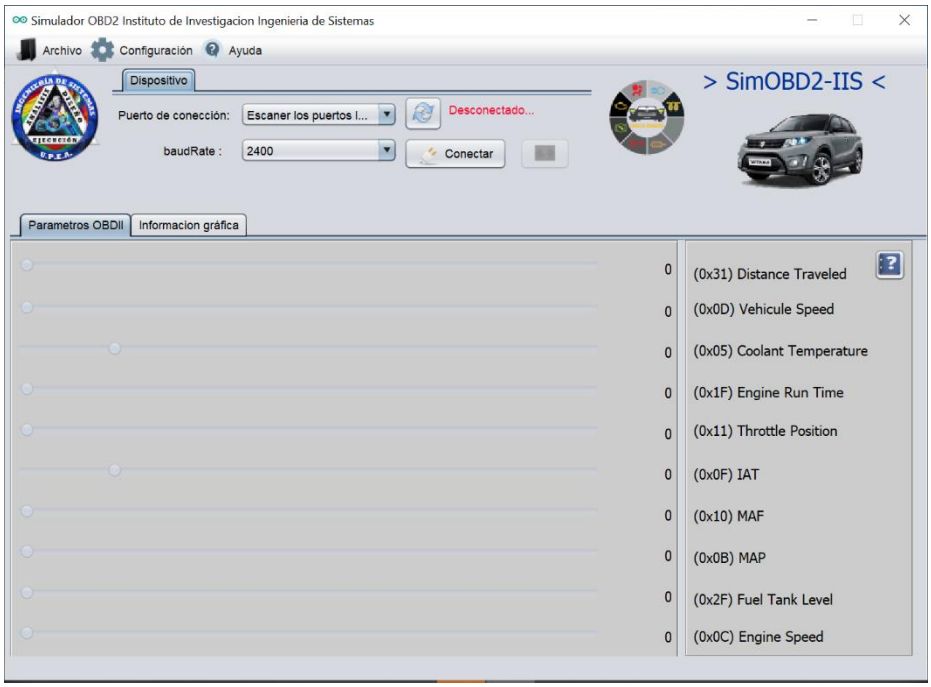
Figura 26. QR repositorio simuladorObd2

Fuente: Repositorio Github – Grover Wilson Quisbert Ibañez

Figura 27. Prototipo de SimuladorObd2Java

Fuente: Desarrollo Propio, Equipo de Investigación

Figura 28. Aplicación simuladorObd2Java



Fuente: Elaboración propia Equipo de investigación

8. AMBIENTES DE INVESTIGACIÓN

El presente proyecto de investigación al ser experimental se debe tener en cuenta para el desarrollo del scanner automotriz en una primera etapa se lo realizará en las instalaciones del instituto de investigaciones donde se desarrollará la parte de comunicación entre la PC, simulador obd2, y pruebas teórico practicas.

Vehículo para luego pasar a una segunda etapa la cual es las pruebas de comunicación el cual se desarrollará en el taller de la carrera de Ingeniería Autotrónica para ir controlando, verificando y la funcionalidad del scanner.

Tabla 18. Etapa, Actividades y ambientes

Etapa	Actividades	Ambientes
Primera	Investigación de parámetros de diseño Desarrollo de scanner automotriz	Instalaciones de Instituto de Investigaciones
Segunda	Verificación de circuito para la comunicación entre PC-Automóvil	Taller de la carrera de Ingeniería Autotrónica

	Funcionalidad de scanner automotriz	

Fuente: Elaboración propia

9. PROCEDIMIENTO DE LA INVESTIGACIÓN

Este procedimiento se tendrá que tener su respectiva planeación los cuales se deben cuidar para obtener resultados que sean objetivos como posibles. (Nohammad Naghi, 2005, pág. 55).

En el procedimiento de la investigación para el presente proyecto se realizará mediante el proceso de los siguientes pasos, los cuales están descritos desde el inicio del proyecto así también como en el proceso de construcción y desarrollo del scaner automotriz.

Tabla 19. Procedimiento de Investigación

Proceso	Detalle	
1	Tener las bibliografías correspondientes para el desarrollo del escáner automotriz.	Datos de Scaner
		Códigos de errores
		Comunicación
		Tipos de conectores
		Parámetros iniciales
2	Determinar el tipo de controlador que se utilizara para la comunicación de entre PC – Vehículo.	
3	Determinar el tipo de comunicación que se tendrá entre PC-Vehículo.	
4	Realizar la investigación correspondiente para tener en cuenta los materiales a ser utilizado para el scaner automotriz.	
5	Elaborar un prototipo de simulador OBD2 que permita realizar pruebas	

6	Realizar las respectivas pruebas de funcionamiento, comunicación y verificación de PC-Vehiculo.	Pruebas de comunicación
		Pruebas de funcionamiento de scaner automotriz
7	Realizar las respectivas conclusiones con los datos obtenidos como también las mejoras que se pueden realizar tanto en funcionalidad como en costo.	

Fuente: Elaboración propia

10. VALIDACIÓN DE INSTRUMENTO

La validación de un instrumento de investigación, se refiere al proceso de recolección de datos. Según (Supo, 2012), establece 6 pasos para poder validar un instrumento:

1) Revisar la Literatura
2) Explorar el Concepto
3) Enlistar los temas
4) Formular Ítems
5) Seleccionar los jueces
6) Aplicar la prueba piloto
7) Evaluar la consistencia
8) Reduce los ítems
9) Reduce las dimensiones
10) Identifica el criterio

De donde se puede resumir en los siguientes pasos de evaluación de instrumentos de investigación:

1) Paso 1: Realizar una prueba del instrumento, Consiste en ofrecer una encuesta a un grupo familiarizado con el tema de investigación, en este caso se ofrecerá una encuesta para la prueba del prototipo.
2) Paso 2: Ejecutar la prueba piloto, se seleccionará un 10% de la población para poder ofrecer las pruebas al prototipo.
3) Paso 3: Limpia los datos recopilados, después de recolectado los formularios se procederá a evaluar los resultados y corregir errores si se encontraron.
4) Paso 4: Validación del prototipo, después de validar el prototipo y corregir los errores, se procede a la validación del mismo.

CAPITULO IV: MARCO APLICATIVO

1. VARIABLES DE INVESTIGACIÓN

La variable dependiente Potencia, Torque será leído en el prototipo de escáner sin ninguna manipulación, en cambio la variable independiente escáner automotriz será manipulado para condiciones de presión atmosféricas de la ciudad de El Alto, revoluciones del motor (velocidad Angular), considerando cierto desgaste de motor de un automóvil.

“La variable dependiente no se manipula, sino que se mide para ver el efecto que la manipulación de la variable independiente tiene en ella” (Fernández y Baptista, 2014, p.131).

Las variables dependientes, Potencia y Torque se desglosarán en dos ecuaciones independientes, también se observará la relación entre estas.

¿Cuántas variables independientes y dependientes deben incluirse en un experimento?

No hay reglas para ello; depende de cómo se haya planteado el problema de investigación y las limitaciones existentes. Si un investigador se encuentra interesado en contrastar efectos de apelaciones emotivas frente a racionales en comerciales televisivos sobre la predisposición de compra de un producto, y sólo le interesa este problema, tendrá una variable independiente única y una sola dependiente. Pero si también pretende analizar el efecto de utilizar comerciales en blanco y negro frente a los que son a color, agregaría esta variable independiente y la manipularía. Tendría dos variables independientes (apelación y colorido) y una dependiente (predisposición de compra), son cuatro grupos (sin contar el de control): a) Grupo expuesto a apelación emotiva y comercial en blanco y negro b) Grupo expuesto a apelación emotiva y comercial en color c) Grupo expuesto a apelación racional y comercial en blanco y negro d) Grupo expuesto a apelación racional y comercial en color O también se podría agregar una tercera variable independiente: duración de los comerciales, y una cuarta: realidad de los modelos del comercial (personas vivas en contraposición a dibujos animados) y así sucesivamente. Claro está que conforme se aumenta el número de variables independientes se incrementan las manipulaciones que deben hacerse y el número de grupos requerido. Entonces, entraría en juego el segundo factor: limitantes, tal vez no pueda reclutar las suficientes personas para varios grupos o contar con el presupuesto para producir tal variedad de comerciales. Por otro lado, en cada caso podría optar por medir más de una variable dependiente y evaluar múltiples efectos de las independientes (en distintas variables). Por ejemplo, además de la predisposición de compra, medir la recordación del comercial y la evaluación estética de éste. Resulta obvio que al aumentar las variables dependientes, no tienen que incrementarse los grupos, porque estas variables no se manipulan. Lo que aumenta es el tamaño de la medición (cuestionarios con más preguntas, mayor número de observaciones, entrevistas más largas, etc.) porque hay más variables que medir. (Fernández y Baptista, 2014, p.135)

En primer lugar, observaremos la variable dependiente potencia, su dependencia de los factores del mismo escáner, de la presión atmosférica de la ciudad de El Alto, de las revoluciones por minuto en tiempo real.

Tabla 20.Variables del trabajo de Investigación

Variables	Enunciado
Independientes (X)	Escáner Automotriz, Presión atmosférica, Numero de revoluciones del motor
Dependiente (Y)	Diagnóstico de potencia y torque

Fuente: Elaboración propia

La manipulación o variación de una variable independiente puede realizarse en dos o más grados. El nivel mínimo de manipulación es de presencia o ausencia de la variable independiente. Cada nivel o grado de manipulación comprende un grupo en el experimento. (Fernández y Baptista, 2014, p.131)

Este nivel o grado implica que un grupo se expone a la presencia de la variable independiente y el otro no. Posteriormente, los dos grupos se comparan para saber si el grupo expuesto a la variable independiente difiere del grupo que no fue expuesto. (Fernández y Baptista, 2014, p.131)

Tras la encuesta realizada se observa que la mayoría de los especialistas en diagnostico con escáner afirmaron que es muy importante contar con un escáner para diagnóstico de potencias y torque debido a que este ayuda a realizar un correcto mantenimiento y conservar o tener un mantenimiento adecuado del motor, además determinar la influencia de las condiciones de presión atmosférica de la ciudad de el alto, por esta razón contar con un escáner que diagnostique potencia y Torque influye y es muy determinante.

“En otras ocasiones, es posible hacer variar o manipular la variable independiente en cantidades o grados” (Fernández y Baptista, 2014, p.132).

“Hay otra forma de manipular una variable independiente que consiste en exponer a los grupos experimentales a diferentes modalidades de la variable, pero sin que esto implique cantidad” (Fernández y Baptista, 2014, p.132).

En el presente trabajo se hará variar las revoluciones por minuto (en cantidades o grados). También se expone a las condiciones de la otra variable que es presión atmosférica, además de la influencia del escáner en sí mismo.

2. POBLACIÓN DE MUESTRA

La población está delimitada por el investigador, la muestra se extrae de esta población con el fin de ser una representación de la población.

Por tanto, para seleccionar una muestra, lo primero que hay que hacer es definir la unidad de muestreo/análisis (si se trata de individuos, organizaciones, periodos, comunidades, situaciones, piezas producidas, eventos, etc.). Una vez definida la unidad de muestreo/análisis se delimita la población. Para el proceso cuantitativo, la muestra es un subgrupo de la población de interés sobre el cual se recolectarán datos, y que tiene que definirse y delimitarse de antemano con precisión, además de que debe ser representativo de la población. El investigador pretende que los resultados encontrados en la muestra se generalicen o extrapolen a la población (en el sentido de la validez externa que se comentó al hablar de experimentos). El interés es que la muestra sea estadísticamente representativa. (Fernández y Baptista, 2014, p.173)

Una vez que se ha definido cuál será la unidad de muestreo/análisis, se procede a delimitar la población que va a ser estudiada y sobre la cual se pretende generalizar los resultados. Así, una población es el conjunto de todos los casos que concuerdan con una serie de especificaciones (Lepkowski, 2008). Una deficiencia que se presenta en algunos trabajos de investigación es que no describen lo suficiente las características de la población o consideran que la muestra la representa de manera automática. Suele ocurrir que algunos estudios que sólo se basan en muestras de estudiantes universitarios (porque es fácil aplicar en ellos el instrumento de medición, pues están a la mano) hagan generalizaciones temerarias sobre jóvenes que tal vez posean otras características sociales. Es preferible, entonces, establecer con claridad las características de la población, con la finalidad de delimitar cuáles serán los parámetros muestrales. (Fernández y Baptista, 2014, p.174).

Tabla 21. Serie historica de vehiculos automotores de tipo unipersonal

Tipo de estadística: SERIES HISTÓRICAS
Tipo societario: EMPRESA UNIPERSONAL
Departamento: LA PAZ
Actividad económica: REPARACIÓN DE VEHÍCULOS AUTOMOTORES
Serie histórica: ANUAL (2010 - 2021) Elaborar Estadísticas

AÑO	NUMERO DE EMPRESAS
2011	297

2012	374
2013	600
2014	830
2015	870
2016	914
2017	937
2018	1.027
2019	1.065
2020	1.107

Fuente: FundaEmpresa 2020

Tabla 22. Serie histórica según todos los tipos de sociedad dedicados a la reparación de vehículos automotores

Tipo de estadística: SERIES HISTÓRICAS
Tipo societario: Todos
Departamento: LA PAZ
Actividad económica: REPARACION DE VEHICULOS AUTOMOTORES
Serie histórica: ANUAL (2010 - 2020)Elaborar Estadísticas

AÑO	NUMERO DE EMPRESAS
2010	331
2011	365
2012	452
2013	679
2014	913
2015	956
2016	1.003
2017	1.029
2018	1.127
2019	1.167
2020	1.216

Fuente: FundaEmpresa 2020

Tabla 23. Cantidad de talleres dedicados a la reparación de vehículos automotores en la ciudad de El Alto a Enero de 2021

SECCIÓN	ACTIVIDAD	2020 Enero	2021 Enero	CRECIMIENTO En cantidad	CRECIMIENTO En porcentaje
G	Venta por mayor y por menor: reparación de vehículos automotores y motocicletas	10.168	10.350	182	1,8%
G_1	Venta por mayor y por menor	9.621	9.787	166	1,7%
G_2	Reparación de vehículos automotores	495	510	15	3,0%
G_3	Venta y reparacion de motocicletas	52	53	1	1,9%
F	Construcción	5.615	5.822	207	3,7%
C	Industria manufacturera	3.385	3.458	73	2,2%
H	Transporte y almacenamiento	2.701	2.831	130	4,8%
M	Servicios profesionales y técnicos	1.227	1.263	36	2,9%
J	Información y comunicaciones	1.209	1.217	8	0,7%
I	Actividades de alojamiento y servicio de comidas	959	978	19	2,0%
N	Actividades de servicios administrativos y de apoyo	500	523	23	4,6%
B	Explotación de minas y canteras	401	451	50	12,5%
S	Otras actividades de servicios	373	364	-9	-2,4%
P	Servicios de educación	322	334	12	3,7%
Q	Servicios de salud y de asistencia social	284	298	14	4,9%
L	Actividades inmobiliarias	256	247	-9	-3,5%
K	Intermediación financiera y seguros	175	171	-4	-2,3%
R	Actividades artísticas de entretenimiento y recreativas	165	165	0	0,0%
A	Agricultura, ganadería, caza, pesca y silvicultura	78	80	2	2,6%
E	Suministro de agua; evacuación de aguas residuales, gestión de desechos y descontaminación	46	52	6	13,0%
D	Suministro de electricidad, gas, vapor y aire acondicionado	4	4	0	0,0%
	Actividad no declarada	3	3	0	0,0%
TOTAL		27.871	28.611	740	2,7%

Fuente: FundaEmpresa 2021

En las tablas anteriores se muestran datos de reparación de vehículos automotores, dentro de estos están los servicios de chapa y pintura, mantenimiento y reparación eléctrico, llanterías, entre ellos servicio de diagnóstico con escáner, etc.

La población considerado es talleres que prestan servicio de diagnóstico con escáner en la ciudad de el alto, sin embrago debido a que no se cuenta con una institución que contabilice el dato exacto de la cantidades de estos talleres, además de la existencia de talleres no registrados o no censados se toma una muestra de 10 talleres especializados en diagnostico con escáner automotriz en vehículos a combustión interna a gasolina, además se observó una pequeña población de talleres que dan este servicio.

3. AMBIENTES DE INVESTIGACIÓN

El presente trabajo de investigación se realizará por etapas y en los siguientes ambientes:

- Simulación del prototipo mediante programas computacionales desarrollado en el instituto de investigación de la carrera de Ingeniería Autotrónica e Ingeniería de Sistemas.
- Implementación y construcción del escáner en ambientes del instituto de investigaciones de la carrera de Ingeniería Autotrónica e Ingeniería de Sistemas.

- Prueba desarrollada en taller de la carrera de ingeniería Autotrónica.
- Pruebas desarrolladas en calles y avenidas de la ciudad de la Paz y el Alto

4. TÉCNICAS E INSTRUMENTOS UTILIZADOS

Tabla 24. Técnicas Utilizadas

Diseño	**Técnicas**		**Instrumentos**
Diseño de investigación documental	Análisis documental		Fichas, computadora y sus unidades de almacenaje, Celulares inteligentes, tabletas.
	Análisis de contenido		Cuadro de registros y su clasificación de sus categorías.
Diseño de investigación de campo	Observación	No estructurada	Diario de Campo Cámara fotográfica
	Encuesta	Oral	Guía de Encuesta (tarjeta)
		Escrita	Cuestionario
	Entrevista	No estructurado	Libreta de notas

Fuente: Elaboración Propia

5. PROCEDIMIENTO DE LA INVESTIGACIÓN

El procedimiento a desarrollar se detalla a continuación:

- Recolección de bibliografía sobre temas de desarrollo de escáner y determinación de potencia y torque
- Determinar la problemática de la ciudad de El Alto en cuanto a potencia y torque de vehículos para el diseño del prototipo.
- Identificar las limitaciones de comunicación entre los diferentes protocolos de comunicación.
- Identificar los instrumentos equipos, dispositivos electrónicos a utilizar.
- Diseño y Simulación del prototipo.
- Implementación y montaje del escáner automotriz.
- Lectura, recolección de datos, pruebas y resultados.
- Interpretación y análisis de los resultados.
- Conclusiones y recomendaciones.

6. DESARROLLO DE SIMULADOR OBD

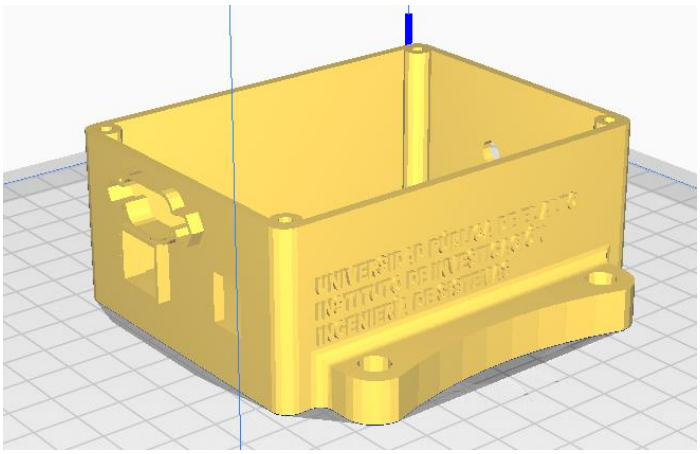
De acuerdo a lo planteado dentro de los objetivos del presente proyecto, se debe desarrollar un dispositivo que nos permita realizar el trabajo de simulación y comportamiento de un ECU (Unidad de Control de Motor), mismo dispositivo fue realizado, de acuerdo a al siguiente esquema:

Figura 29. Tecnología Empleada en el SIMU-OBDII1.0

Funte: Elaboracion Propia

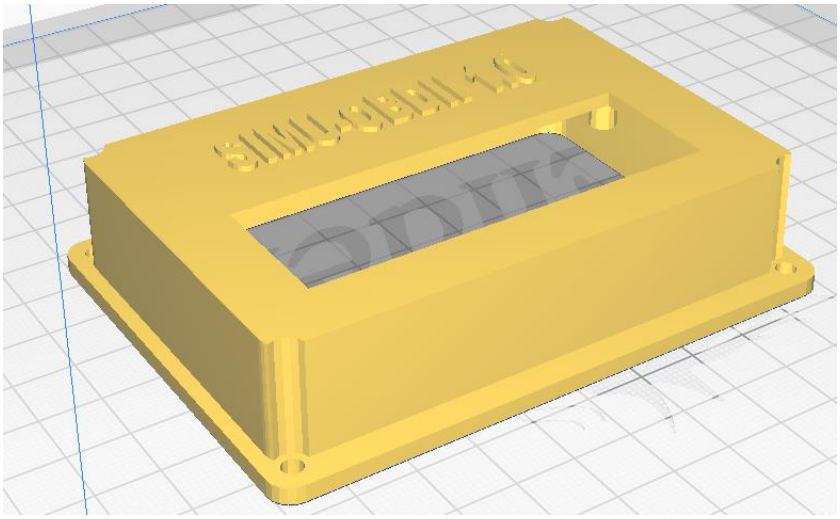
Para poder contar con un dispositivo más profesional se desarrollo el modelo e impresión 3d, de acuerdo a las siguientes figuras:

Figura 30. Diseño CAD Base Simulador



Fuente: Elaboracion Propia

Figura 31. Diseño CAD Tapa Simulador

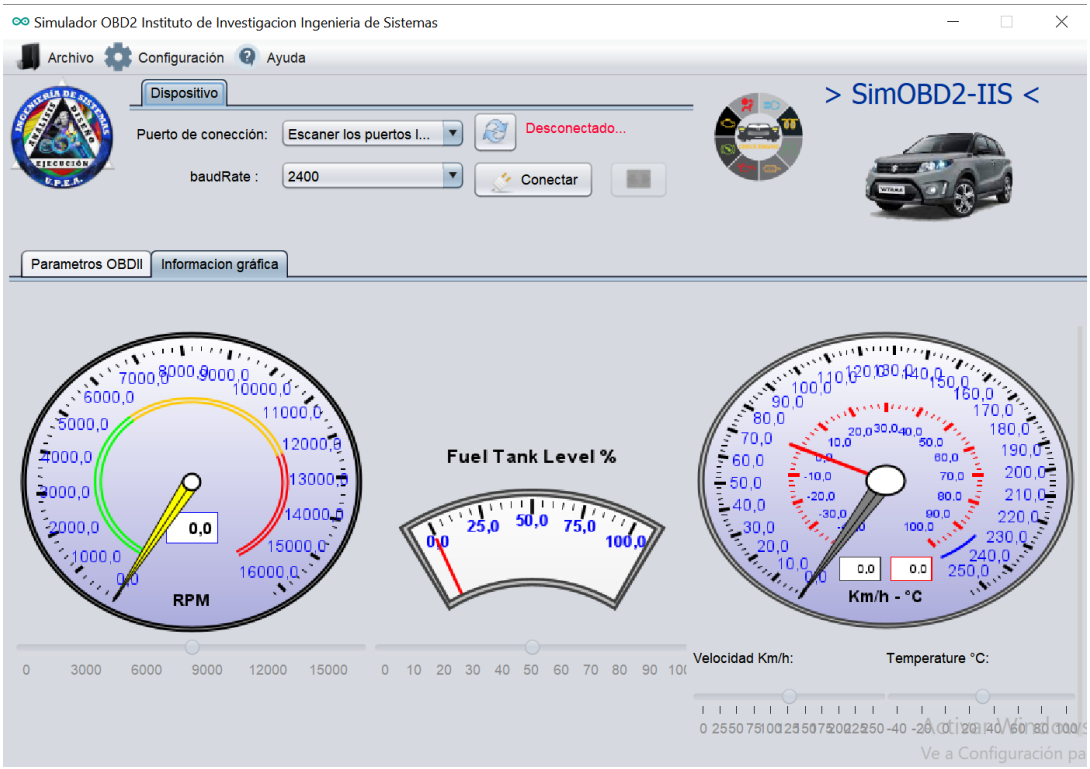


Fuente: Elaboracion Propia

A la misma vez se desarrolló una aplicación de escritorio que permita simular el funcionamiento de los sensores y actuadores de un automóvil, mismos que se desarrollo en JAVA y esta disponible en el repositorio del proyecto.

Después de termino el diseño y desarrollo del prototipo de simulador OBDII, se puede apreciar los resultados en la Figura 33.

Figura 32. Aplicacion de Escritorio Simulador OBD2



Fuente. Elaboración Propia

Figura 33. Simulador OBD SIMU. vista frontal y lateral

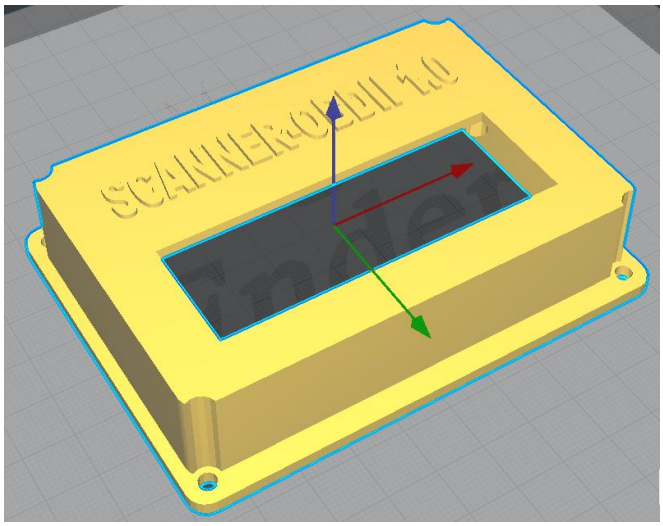
Fuente. Elaboracion Propia

7. DESARROLLO DEL SCANNER OBDII

Después de finalizado el desarrollo del prototipo simulador, se procede a desarrollar la interfaz de comunicación (Scanner) entre la aplicación móvil y ECU, para tal cometido se empleó la misma arquitectura que se utilizó en el simulador OBD2, con la excepción que se incluye un módulo de comunicación Bluetooth para poder tener transferencia de información entre el prototipo de scanner y el dispositivo móvil (celular).

Bajo el mismo cometido de contar con dispositivo profesional se realizo el diseño CAD del prototipo. Como se puede apreciar en la siguiente figura.

Figura 34. Diseño CAD de tapa de Scanner

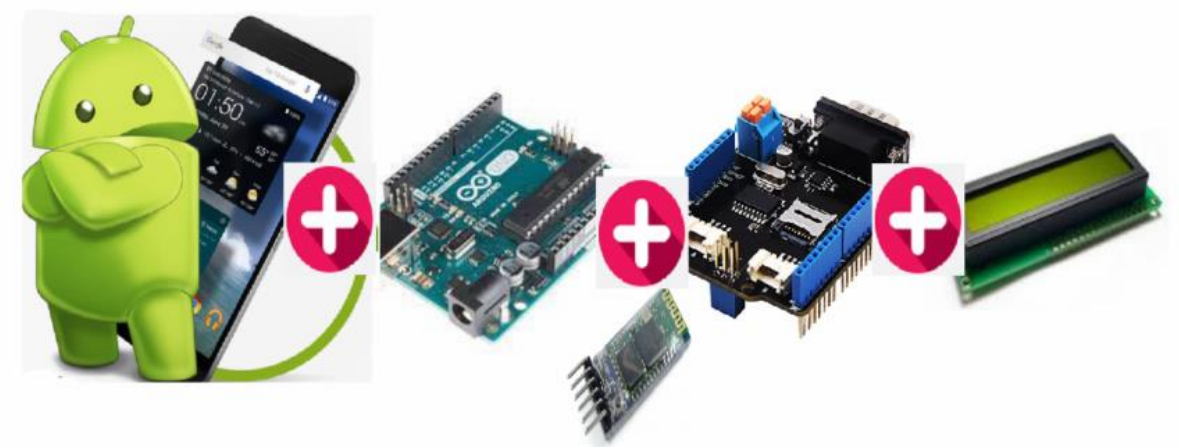


Fuente: Elaboracion Propia

El diseño de la base es el mismo que se utilizo para el simulador, como se puede apreciar en la figura figura 29.

Para el desarrollo del scaner se utilizo tecnologia libre tanto en hardware como software de acuerdo al siguiente grafico, cabe aclarar que la arquitectura es parecida al simulador por que se emplea la misma capa de comunicación.

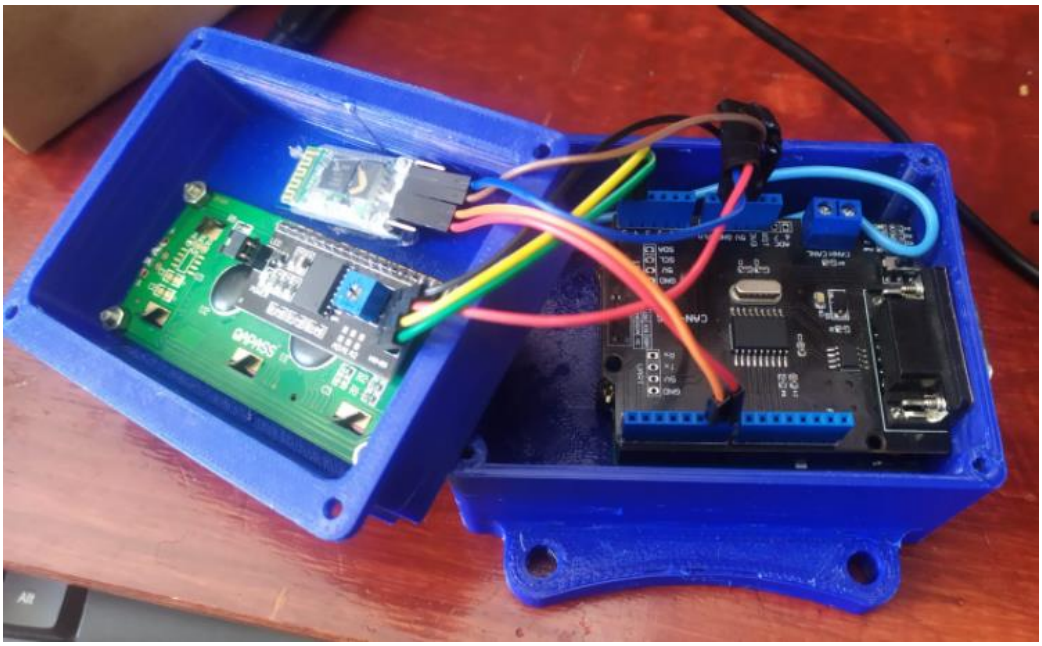
Figura 35. Tecnología Empleada en el Sccaner OBDII



Fuente: Elaboracion Propia

Una vez coluido el desarrollo se procedio al ensamblaje del scanner obd2 de acuerdo a las sigueintes figuras:

Figura 36. Dispositivo Scanner OBDII 1.0 (Desarmado)



Fuente: Elaboración Propia

Figura 37. Dispositivo SCANNER-OBDII (Vista Frontal y Lateral)

Fuente: Elaboracion Propia

El codigo fuente (C++ y Java) empleado para el desarrollo de los prototipos se enceuntra disponible al interior del repositortio del proyecto, con la intencion de que si algun estudiante, docente o persona externa a la Universidad pueda distribuir o mejorar el mismo.

Figura 38. Repositorio GitHub proyecto

Fuente: Elaboración Propia

8. MINI ELM 327 CAN BUS BLUETOOH V2.1

Intérprete de protocolos OBDII a UART. Este está basado en un PIC de Microchip con software desarrollado por ELM Electronics y tiene la cualidad de interpretar todos los protocolos OBDII con pocos circuitos externos. Tiene las siguientes funciones:

Encargada de convertir los protocolos OBDII a UART con niveles de voltaje TTL, para ser enviados e interpretados por el Arduino Uno.

Realiza automáticamente la búsqueda del protocolo OBDII que tiene el automóvil e inicializa la comunicación con la ECU.

Uso de comandos AT para lograr la comunicación con la ECU para poder realizar peticiones a La ECU de configuración del intérprete, ELM 327 utiliza comandos con números hexadecimales. Los comandos AT se emplean en módems y son un conjunto de comandos compuestos por cadenas de texto cortos que se combinan entre sí para producir órdenes completas para realizar operaciones.

Casi todos los automóviles producidos hoy están obligados, por ley, a proporcionar una interfaz para el Conexión de equipos de prueba de diagnóstico. Los datos la transferencia en estas interfaces sigue varios estándares, pero ninguno de ellos es utilizable directamente por PC o dispositivos inteligentes dispositivos. El ELM 327 está diseñado para actuar como puente entre estos puertos de diagnóstico a bordo (OBD) y una interfaz serial RS232 estándar.

Además de poder detectar automáticamente e interpretar nueve protocolos OBD, el ELM 327 también proporciona soporte para comunicaciones de alta velocidad, un modo de suspensión de baja potencia, y el camión y autobús J1939 estándar. También es completamente personalizable, debería desea modificarlo para que se adapte más a sus necesidades.

8.1. Características

• Control de energía con modo de espera

• Interfaz serie universal (RS232)

• Busca protocolos automáticamente

• Totalmente configurable con comandos AT

• Diseño CMOS de baja potencia

8.1.1. Lectura de Datos

Comunicación ECU Mini ELM 327 Can Bus Bluetooh V2.1

La comunicación entre Vehículo y Escáner se realiza de la siguiente manera:

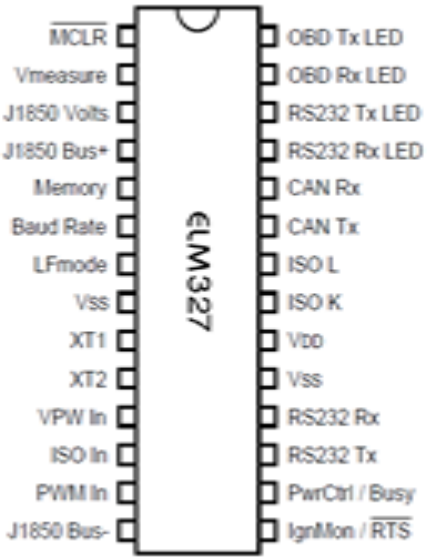
- Conexión de datos con el ECM a través del DLC
- interprete de datos de la ECM o ECU
- interpretación de las marcas más comerciales en la ciudad de el alto: Nissan, Toyota, Mitsubishi, Zuzuki, King Long.
- Protocolos de comunicación utilizados para las marcas mencionadas con puerto OBD2 (ISO 9141-2).

El escáner tiene capacidad de determinar:

- RPM del motor
- Cálculo del valor de carga
- Temperatura del refrigeración
- Estado del sistema de combustible
- Velocidad del vehículo
- Ajuste de combustible a corto plazo
- Ajuste de combustible a largo plazo
- Avance de tiempo
- Temperatura del aire
- Flujo de aire
- Posición de acelerador absoluta
- Estado del sistema de carburador
- Presión de carburante y testeo.
- Códigos de Error (DTC)
- Presión Absoluta Múltiple (MAP)

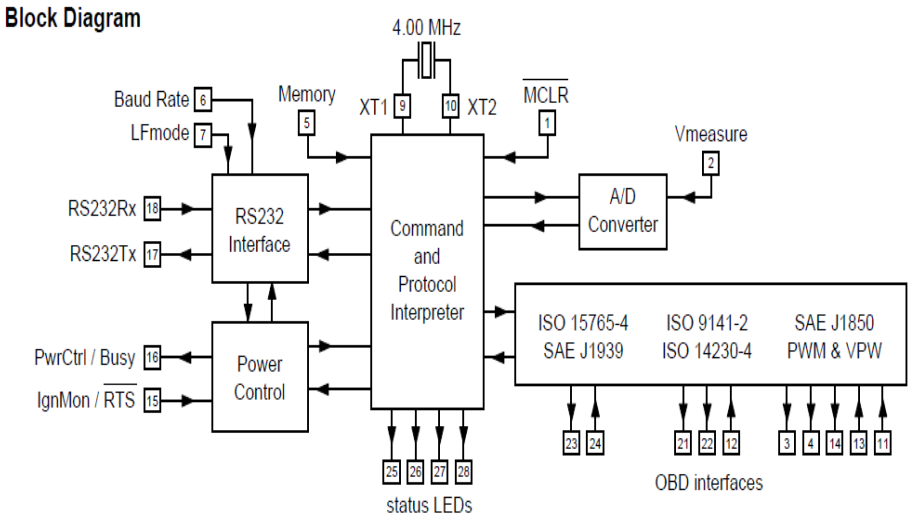
- Sensores de Oxígeno (Voltajes)
- Tanque de Combustible
- Carga del Motor
- Avance
- Potencia y torque del motor

Figura 39. Diagrama de conexión PDIP y SOIC



Fuente. www.elmelectronics.com

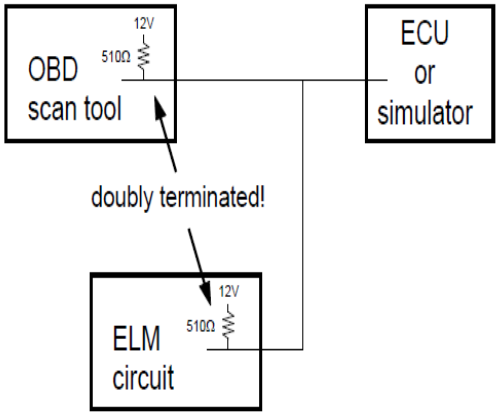
Figura 40. Diagrama de Bloques



Fuente. www.elmelectronics.com

Para el Escaneo de un Vehículo con el ELM327, se realiza la conexión de la ECU (unidad de control electrónico) con OBD2 que tiene un puerto incorporado en el vehículo, para luego conectarse con el escáner ELM 327 como se muestra en la figura 41.

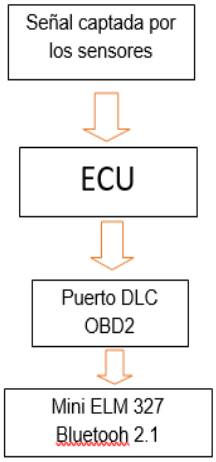
Figura 41. Monitoreo de una herramienta de Escaneo



Fuente. www.elmelectronics.com

Los Sensores toman datos para enviar estas Señales ECU este procesa la información para después ser enviado a los actuadores, además el automóvil posee el puerto DLC OBD2 para el diagnóstico del estado del vehículo para ser diagnosticado con un Escáner, en la figura 42 se utiliza el escáner mini ELM327 Bluetooh 2.1.

Figura. 42 Sistema de diagnóstico de un Vehículo por Escáner

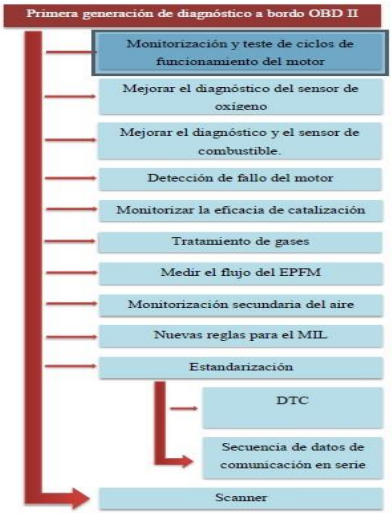


Fuente. Elaboración propia

8.2. Sistema de Identificación OBDII y su Funcionamiento en el Area Automotriz.

OBD-II a diferencia del sistema OBD-I aporta un control casi completo del motor, también monitoriza partes del chasis y otros dispositivos del vehículo. Los grupos reguladores fueron los principales impulsores de esta nueva mejora. Dentro de estas mejoras cabe destacar las siguientes:

Figura 43. Funcionamiento del OBDII



Fuente: Elementronic.

Un sistema OBDII automotriz tiene básicamente los siguientes componentes:
Al escanear el Módulo de Control de Motor (ECM, por sus siglas en inglés: Engine Control Module), si el vehículo presenta una falla, arrojará un código.

Estos códigos se denominan códigos DTC (siglas de Diagnostic Trouble Codes, lo que es Códigos de Problemas de Diagnóstico, traducido al inglés). ¿Cómo se clasifican los DTC en el sistema OBD2?

POR DESCRIPCION: Describe los 5 dígitos del código que empieza con un digito alfabético (“P”,”C”,”B” o “U”) y seguido de los 4 dígitos numéricos (ya lo conocemos porque se toco el tema antes, pero por si acaso dejamos la imagen abajo).

POR ACTIVACION: Hay tres tipos.

TIPO A

1.- Relacionado con emisiones.

2.- Requiere iluminación de la luz “check engine” la primera vez que ocurre la falla en un ciclo de ignición.

¿Qué es un ciclo de ignición?

Es el Periodo de funcionamiento del motor con llave contacto, off – on - off, en el cual se dan las condiciones para el PCM pruebe un elemento o sistema.

3.- Guarda el DTC en la memoria PCM y almacena un FREEZE FRAME

TIPO B

1.- Relacionado con emisiones.

2.- Fija un DTC pendiente después de detectar una falla en un ciclo de ignición.

3.- Enciende la luz “check engine” luego de encontrar una falla en dos ciclos de ignición consecutivos.

4.- Guarda el DTC en la memoria PCM y almacena un FREEZE FRAME.

5.- Borra el DTC pendiente si en el siguiente ciclo no detecta el fallo.

TIPO C

1.- No relacionado con emisiones.

2.- Almacena un DTC solo en la memoria la primera vez que detecta un fallo en un ciclo de ignición.

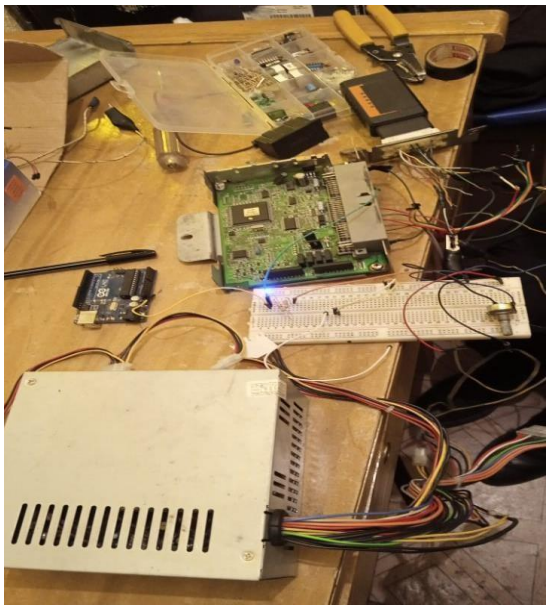
3.- No enciende la luz “check engine”.

4.- En algunos modelos enciende la luz denominado “Service Engine”.

Nota: FREEZE FRAME significa Cuadro congelado o capturas almacenadas tipo "instantáneas" de estados de los sensores en momentos determinados que almacena la PCM por decir un ejemplo.

- **Computadora del motor (ECU Motor, ECM),** está encargada de controlar el encendido y la gestión del motor de combustión interna.

Figura 44. ECU programable



Fuente: Elaboración propia

A. ELM327 connectividad con Adroid: Fabricado por Elm Electronics, el ELM 327 es un microcontrolador programado y diseñado específicamente para traducir la interfaz OBD, que se encuentra en la memoria de los vehículos. En pocas palabras: es un obd2 a base del chip ELM327 (herramientaautomotriz).

El Elm 327 es un microcontrolador que esta ´programado y diseñado para traducir la interfaz OBD que se encuentra en la memoria del vehículo. Estos datos nos ayudaran para comunicarnos con la ECU de la computadora para luego procesar los respectivos datos.

La Interfaz ELM327 es una herramienta para la diagnosis de vehículos equipados con sistemas OBD II – EOBD – CAN normalmente encontrados en vehículos americanos, europeos y asiáticos fabricados partir de 1996 y que tengan el conector de diagnóstico de 16 pines. Esta interfaz permite la conexión a la computadora del vehículo para leer y borrar códigos de error DTC, la famosa Luz “CHECK ENGINE” y mejor llamada MIL

(Malfunction Indicator Lamp). Así mismo nos permite visualizar los valores de sensores y actuadores del motor en pleno funcionamiento (KitoSDev-CMS, 2020).

Este equipo es muy útil gracias a que es muy simple de usar puede ser usado por profesionales y aficionados, este nos ayuda acceder a los módulos de control del vehículo que nos permite ver parámetros en tiempo real del vehículos, fallos e información que se puede visualizar en pc o en dispositivos con sistema Android, este equipo es muy útil para la detección rápida de problemas en vehículo al mismo tiempo es casi indispensable para cualquier profesional contar con este tipo de equipo que ayudan mucho a ahorrar tiempo y recursos en la búsqueda de fallas.

Tabla 25. Velocidad de datos

standar	Velocidad de datos
ISO 9141-2	(5 baudios init, 10,4 kbit / s)
ISO 14230-4 KWP	(5 baudios init, 10,4 kbit / s)
ISO 14230-4 KWP	(init rápido, 10,4 kbit / s)
ISO 15765-4 CAN	(11 ID bit, 500 kbit / s).
ISO 15765-4 CAN	(29 bit ID, 500 kbit / s)
ISO 15765-4 CAN	(29 bit ID, 500 kbit / s)
ISO 15765-4 CAN	(29 bit ID, 250 kbit / s)

Nota.ELM327 versión 2.1, por KitoSDev-CMS (https://obd2elm327.es/elm327-version-2.1).

8.3. Proceso de comunicación

Para el proceso de comunicación se tendrá las siguientes fases de comunicación:

Primera fase

La primera fase consta de tener el vehículo el cual tiene que cumplir las características que nos propone el ELM327 que es el que realizara el interfaz entre vehículo y App (Android). Por lo tanto, este vehículo está limitado para su respectiva comunicación, además llevar a

un lugar donde se tenga el espacio suficiente para las respectivas prácticas de comunicación.

Segunda fase

Para la segunda fase tendremos que ya tener a disposición del interfaz en este caso realizaremos la comunicación con un Elm 237 el cual se comunicará con la memoria del vehículo además de ir convirtiendo los diferentes protocolos de comunicación que se tendrá para tener los respectivos datos y posterior análisis.

Tercera fase

En esta fase es la comunicación entre el vehículo, interfaz Elm 327 y la app(Android) en el cual se mostrarán los respectivos datos para posterior análisis de los mismos.

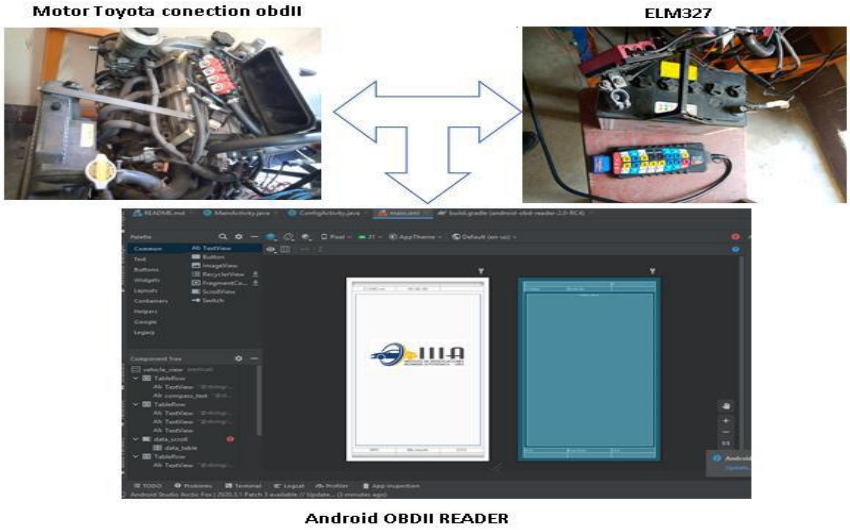
De esa forma será el proceso de comunicación para la obtención de datos del automóvil el cual ya se tendrá en la app(android) para su respectiva lectura de códigos de erros lo cual llevará al mantenimiento del vehículo o sustitución del elemento que se encuentra en falla.

8.4. Comunicación e integrando las fases de cambios

Para las respectivas pruebas se realizaran los cambios en la primera fase la cual está dada por el Simulador de la ECU el cual estará encargado de generar las respectivas señales que un vehículo nos estaría enviando, para luego enviar esos datos al ELM 327 que es el interfaz que tendrá las respectivas señales y luego serán enviadas al respectivo programa app(android) para su respectivo tratamiento de los datos y visualización en la computadora personal.

La obtención de estos datos se los nombrara en los siguientes subtítulos donde se da más detalle de los respectivos fases para la comunicación con el scanner y sus respectivos componentes así también los respectivos programas para el análisis de los mismos datos que se tienen a partir de la respectiva interfaz.

Figura 45. Comunicacion entre el motor, ELM327 y dispositivo android



Fuente: Elaboración Propia

8.5. Simulador de ECU

La comunicación y los datos que tenemos que tomar en cuenta primeramente debemos pensar en un generador de señales que va a entregarnos las respectivas señales para el funcionamiento del Scanner, entonces se tuvo que realizar un simulador que en este caso simulara las señales de la ECU.

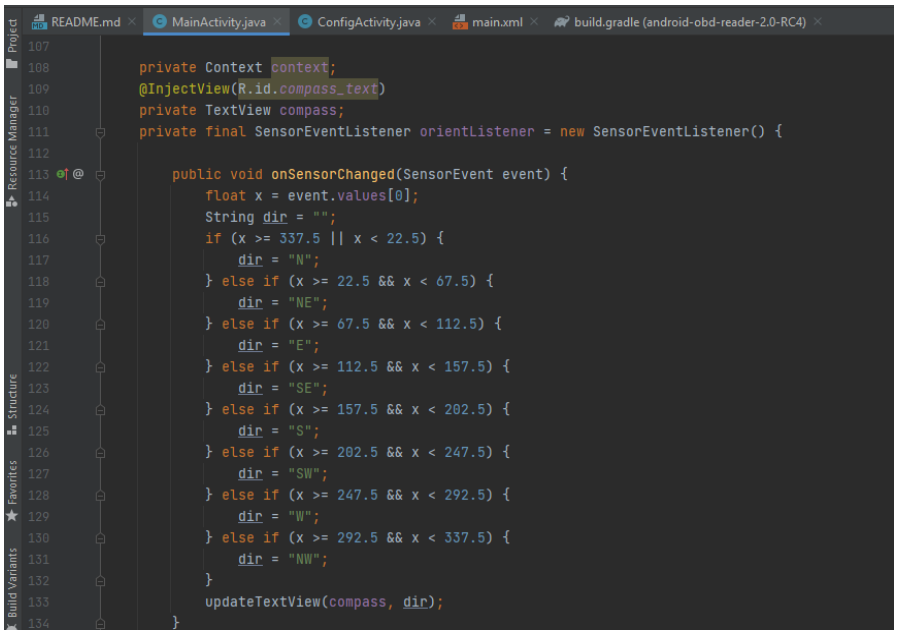
Para las pertinentes pruebas a realizar en el Scanner automotriz se investigó y construyó el siguiente simulador en el cual se simula estas señales con el protocolo ISO 9141-2, ya que sin este simulador no podríamos realizar las respectivas pruebas del scanner automotriz.

El simulador está compuesto de una tarjeta de Arduino Mega como también de componentes electrónicos los cuales van controlando las respectivas señales para su funcionamiento.

Apis de Conectividad OBDII-ELM327ADROID, identificación a través de código de errores.

El sistema de identificación de conectividad de obd2, con el elm327 transmitir y recibir las señales bluetooth, que a atreves de los Apis creados hará la conexión de comunicación así realizara la lectura de códigos de errores.

Figura 46. API (OBDII, ELM327)



Fuente: Elaboración propia.

En el área automotriz esta Api (Android) es una elaboración exclusiva para su comunicación y flujo de información.

CAPITULO V: RESULTADOS

1. INTERPRETACIÓN DE LOS RESULTADOS

Se realizó una encuesta con el fin de tomar parámetros de diseño de acuerdo a los requerimientos de talleres, esto con el fin de responder a las necesidades que se quiere abarcar en cuanto al diagnóstico de potencia y torque.

Los resultados de la encuesta a cada pregunta realizada fueron:

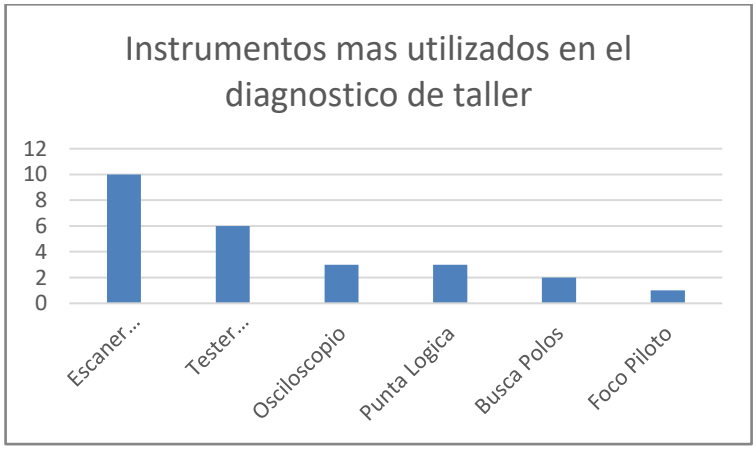
¿Qué instrumentos y/o equipos utiliza generalmente para diagnosticar fallas en un automóvil?

Tabla 26. Instrumentos más utilizados en Diagnostico

Instrumento Utilizado	Número de talleres
Escáner Automotriz	10
Multímetro analógico y/o digital	6
Osciloscopio	3
Punta Lógica	3
Busca Polos	2
Foco Piloto	1

Fuente: Elaboración propia

Figura 47 instrumentos más utilizados para el diagnóstico de fallas



Fuente. Elaboración propia

Como se puede Observar El escáner es el instrumento más utilizado en estos talleres, razón por que el escáner es un instrumento fundamental para el diagnóstico del estado del motor

de un vehículo.

¿Cuenta con al menos un escáner automotriz en su taller?

Todos los encuestados respondieron contar con un escáner en el taller, esto también muestra lo fundamental de este instrumento, debiéndose remplazar rápidamente el mas estado de un escáner.

¿Mencione que marcas de escáner conoce de modo que diagnostique a la mayor cantidad de vehículos de diferentes marcas?

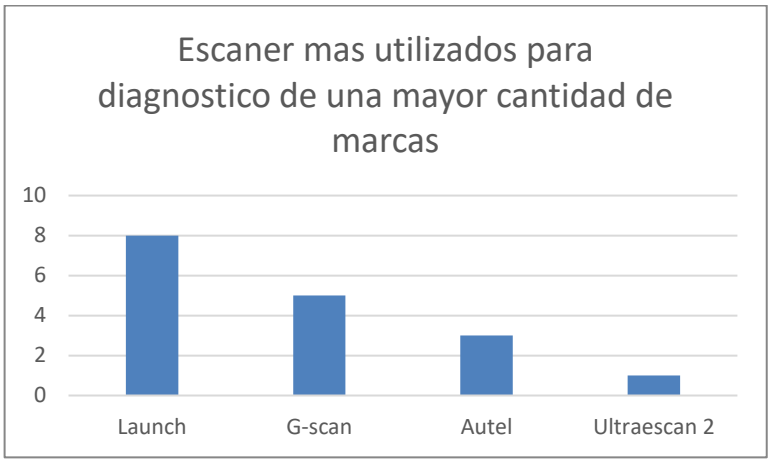
El resultado fue el siguiente:

Tabla. 27 marcas de escáner más utilizados

Escáner Utilizado	Numero de talleres
Launch	8
G-Scan	5
Autel	3
Ultraescan 2	1

Fuente: Elaboración propia

Figura 48. Grafica de escáner mas utilizados en diagnóstico



Fuente: Elaboración propia

Como se puede observar una variedad de marcas de escáner, el más preferido y utilizado en los talleres es el Launch, este escáner abarca a una mayor cantidad de marcas, pudiéndose realizar el diagnóstico de una mayor variedad de marcas, siendo las otras

marcas especificas solo para ciertas marcas, es decir leen una menor cantidad de protocolos de comunicación.

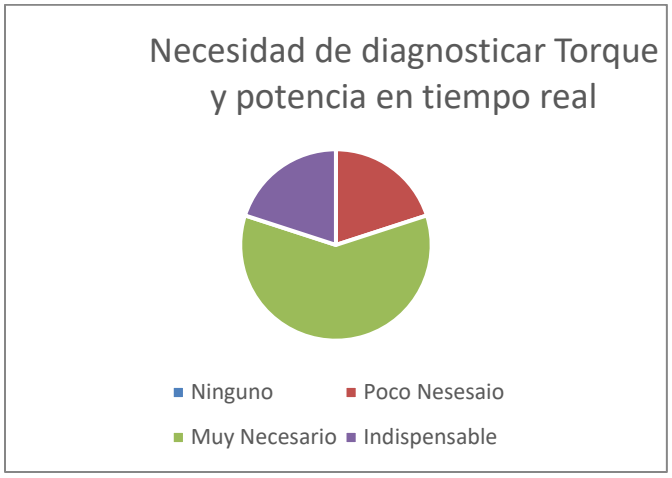
¿Considera necesario determinar o verificar potencia y torque (par motor) en un motor en tiempo real?

Tabla 28. Necesidad de diagnosticar potencia y torque

Ninguno	0
Poco Necesario	2
Muy Necesario	6
Indispensable	2

Fuente: Elaboración propia

Figura 49. Necesidad de diagnóstico de potencia y torque



Fuente: Elaboración propia

Para la mayoría de los encuestados determinar potencia y torque del motor resulta muy Necesario, razón por el que se considera determinar estas variables en el prototipo

¿Qué instrumentos y/o equipos conoce para determinar potencia y torque (par motor) en tiempo real en un motor? (mencione)

Tabla 29. Instrumentos para el diagnostico de potencia y torque

Escáner Profesional	2
Dinamómetro	2
Ninguno	3

Fuente: Elaboración propia

La mayoría dijo desconocer algún instrumento para medir estas variables, sin embargo, dos talleres dijeron conocer el escáner profesional y el dinamómetro para estos fines.

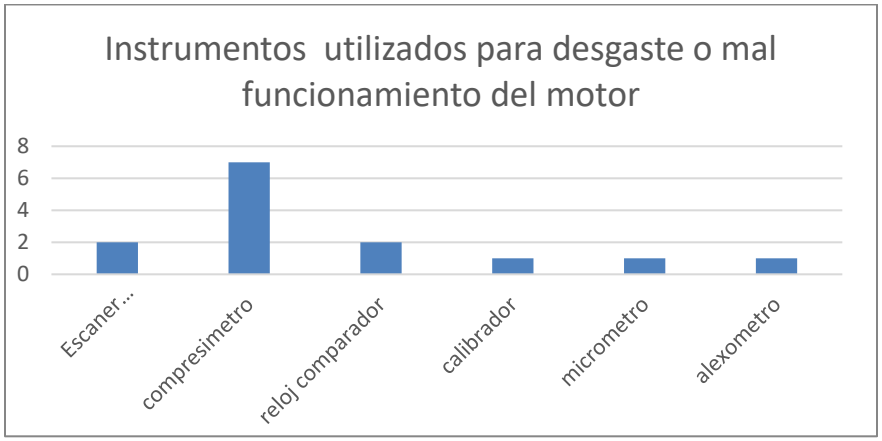
¿Con que instrumento o equipo verifica el desgaste o mal funcionamiento del motor?

Tabla 30. Herramientas utilizadas para diagnóstico de desgaste el motor

Escáner Profesional	2
Compresímetro	7
reloj comparador	2
calibrador	1
micrómetro	1
alexómetro	1

Fuente: Elaboración propia

Figura 50. Instrumentos utilizados en diagnóstico del motor



Fuente: Elaboración propia

Se observa que el compresímetro es el instrumento más utilizado para estos fines, también es importante el escáner automotriz.

¿Qué tipo de escáner para vehículos livianos es el más utilizado en su taller?

Tabla. Escáner comúnmente utilizados en diagnóstico

Autel Maxidas	4
Launch	6
Scanner Delphi	1

Fuente: Elaboración propia

Se observa que el escáner más utilizado es el Launch, seguido de Autel Maxidas

¿Qué tipo de puerto utilizan generalmente los automóviles diagnosticados en su taller?

Tabla 31. Tipo de puerto más comunes en talleres

OBDI	0
OBDII	10

Fuente: Elaboración propia

Se observa que la mayoría de los vehículos de diagnóstico para escáner en la ciudad de El Alto usan OBDII, razón por lo que se trabaja con este protocolo en el prototipo de investigación.

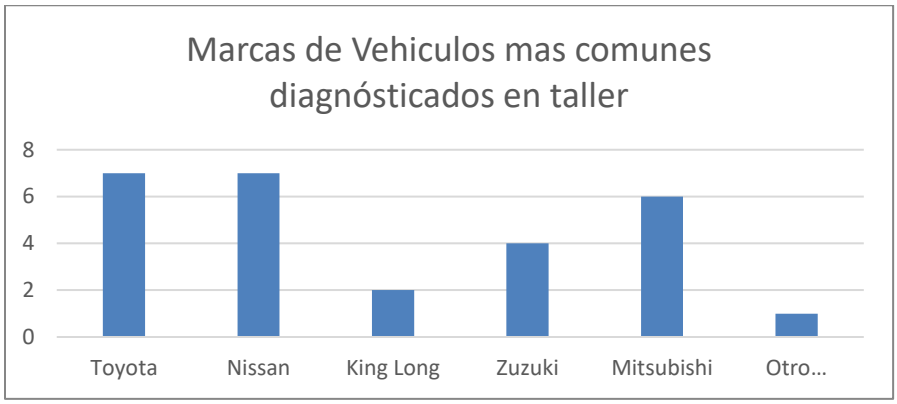
¿Cuál es la marca de vehículo comúnmente diagnosticado con un escáner en el taller?

Tabla 32. Marcas de vehículos comunes en la ciudad de El Alto

Toyota	7
Nissan	7
King Long	2
Suzuki	4
Mitsubishi	6
Otro…	1

Fuente: Elaboración propia

Figura 51. Marcas comunes en talleres



Fuente: Elaboración propia

Como se observa las marcas más comunes en la ciudad de El Alto Es Toyota, Nissan, Mitsubishi, además en mayor cantidad Suzuki y King Long. Para el diseño de prototipo de escáner se considerará estas cinco marcas.

¿Con que instrumento o equipo diagnostica una pérdida de potencia o torque en un motor?

Tabla 33. Herramientas utilizados en diagnostico de potencia y torque

Manómetro de presión	2
Compresímetro	4
Escáner	1
Reloj comparador	1
Alesómetro en embrague	1

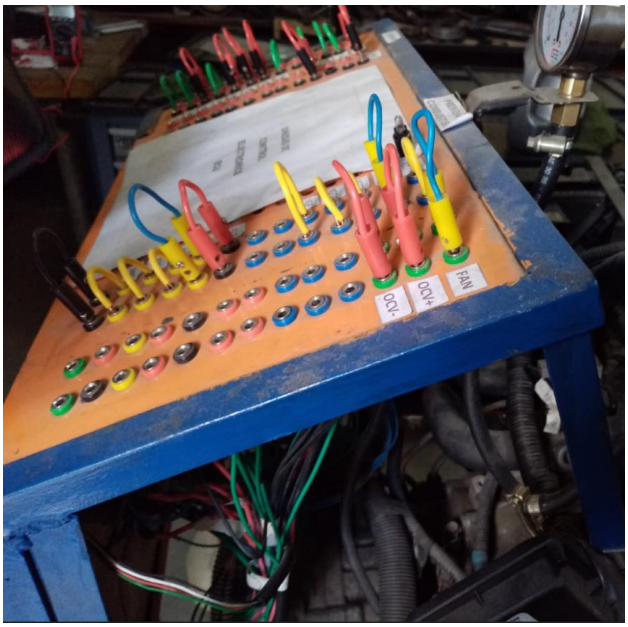
Fuente: Elaboración propia

Entre los instrumentos más utilizados en talleres se tiene al compresímetro, manómetro de presión, entre los menos utilizados el escáner, reloj comparador y acelerómetro en embrague. De acuerdo al trabajo realizado en los laboratorios de la carrera de Autotronica, se realizo pruebas en dispositivos reales. Como se describe en las siguientes figuras:

Figura 52.ECU de Prueba

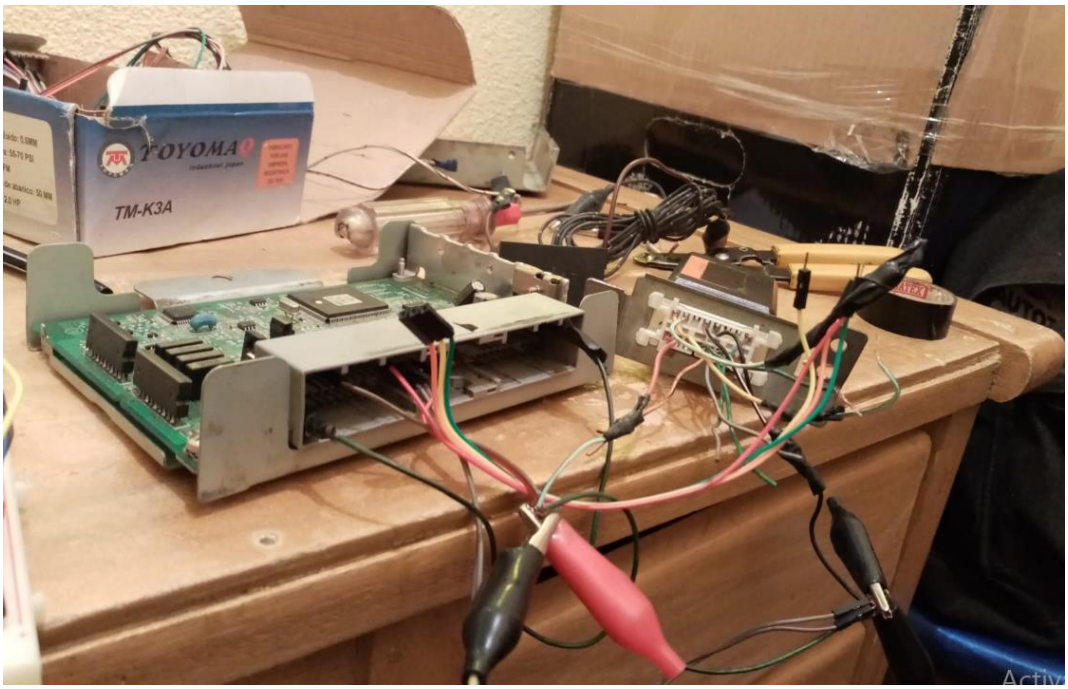
Fuente: Elaboracion propia

Figura 53 Simulador de fallos



Fuente: Elaboracion propia

Figura 54. Pruebas de ECU y escaner



Fuente: Elaboracion propia

Figura 55. Escaner de Prueba

Fuente: Elaboracion propia

2. DETERMINACIÓN DE POTENCIA INDICADA Y EFECTIVA DE UN MOTOR 1SZ-FE TOYOTA

2.1. Determinación en la presión indicada por el diagrama redondeado

La desviación de los valores de Pi, con respecto a los cálculos se evalúa a través del coeficiente de redondeamiento (plenitud). La presión media indicada del ciclo es (Jovaj, 1982, pág. 170)

$$Pi = \varphi_1 * Pi_{(an)} n_r$$

Algunos datos experimentales muestran que φ_1 = (0.92-0.97) (Jovaj, 1982). La desviación del ciclo real respecto al teórico en los motores a gasolina donde sus valores de $\varphi_1$se aproximan al límite superior.

$$Pi = 0{,}97 * 1{,}0 = 0{,}97\ MPa$$

2.2. Potencia indicada de un motor = Ni.

La potencia indicada de un motor, está determinada por la presión media del ciclo por la cilindrada del motor (Jovaj, 1982, pág. 171)

$$Ni = \frac{Pi * Vh * n}{30}$$

Dónde:

Ni: expresado en Watt

n: revoluciones por minuto

Pi= Presión media indicada del ciclo (MPa)

Vh= Cilindrada del cilindro (cc o litros)

Vh= cilindrada del cilindro (Cm3)

La cilindrada del cilindro se calcula por la siguiente ecuación

$$Vh = \frac{D^2 * \pi * S}{4}$$

Dónde:

D: diámetro del pistón (69.00 mm=6,9 cm)

S: Carrera del pistón (66.70 mm=6,67 cm)

$$Vh = \frac{D^2 * \pi * S}{4}$$

$$Vh = \frac{(6{,}9\ Cm)^2 * \pi * 6{,}67 cm}{4}$$

$$Vh = 249{,}41\ cm^3$$

Para el cálculo de Potencia indicada del motor:

$$Ni = \frac{(0{,}97\ MPa) * (249{,}41\ Cm^3) * (6000 rpm)}{30}$$

$$Ni = 48\,385{,}54\ W = 48{,}38\ KW$$

$$Ni = 64{,}86\ HP$$

2.3. Perdidas mecánicas en el motor (Nm)

Parte del trabajo indicado se gasta en vencer el rozamiento en las piezas móviles, en el intercambio de gases y en el accionamiento de los mecanismos auxiliares. (Jovaj, 1982, pág. 209)

$$Nm = \frac{Pm * Vh * n}{30}$$

Dónde:

Nm: esta expresado en Watt

Pm= Potencia media del ciclo para el rango de trabajo de (0.145-0.150) (Jovaj, 1982, pág. 210)

Considerando Pm=0,145

$$Nm = \frac{0{,}145(249{,}41 Cm^3) * (6000 rpm)}{30}$$

$$Nm = 7\,232{,}89\ W$$

$$Nm = 9.69\ HP$$

2.4. Potencia efectiva (Ne).

La potencia que puede obtenerse en el cigüeñal del motor y se aprovecha para accionar cualquier máquina, se llama potencia efectiva y se designa (Ne) (Jovaj, 1982, p. 176)

$$\boldsymbol{Ne = Ni - Nm}$$

$$\boldsymbol{Ne} = 48\,385{,}54\,W - 7\,232{,}89\,W$$

$$Ne = 41\,152{,}65W$$

$$Ne = 55{,}16\,HP$$

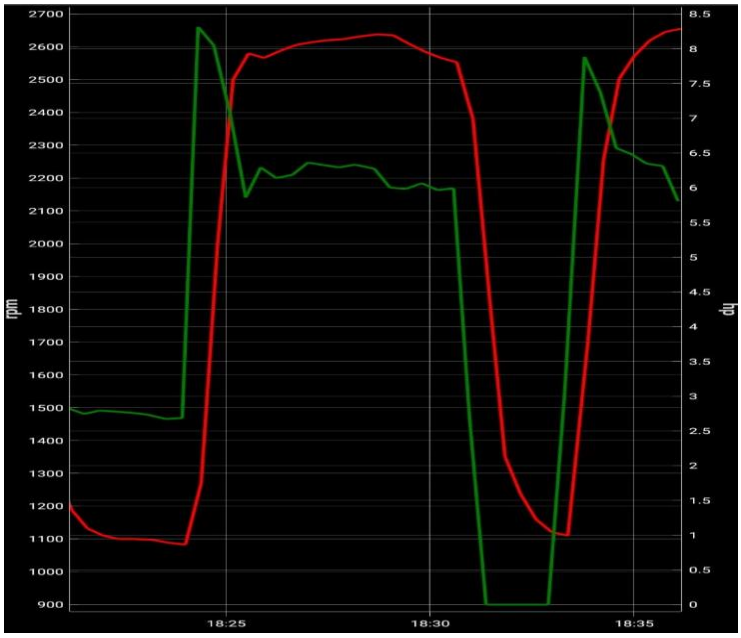
En un escáner se puede observar las revoluciones por minuto, asi como la potencia en tiempo real. A continuacion se muestra esas variables en un diagnóstico.

Se procede a conectar el ELM327 al puerto DLC OBDII.

Se realiza el encendico del vehiculo, para luego realizar la comunicación con el dispositivo Smartphone.

Se procede al diagnóstico de potencia y rpm del motor.

Figura 56. Potencia y rpm diagnosticado con escáner



Revoluciones por minuto máximo (2655 rpm)

Potencia

Fuente. Elaboración propia

Para estas condiciones de funcionamiento se realiza los cálculos de potencia indicada, perdidas mecánicas del motor, potencia efectiva.

Potencia indicada del motor:

$$Ni = \frac{(0{,}97\ MPa) * (249{,}41\ Cm^3) * (2655\ rpm)}{30}$$

$$Ni = 21410{,}60\ W = 21{,}41\ KW$$

$$Ni = 28{,}70 HP$$

Perdida mecánica en el motor:

$$Nm = \frac{0{,}145(249{,}41 Cm^3) * (2655 rpm)}{30}$$

$$Nm = 3200{,}55 W$$

$$Nm = 4{,}29 HP$$

Potencia Efectiva:

$$\boldsymbol{Ne = Ni - Nm}$$

$$\boldsymbol{Ne} = 21410{,}60\ W - 3200{,}55 W$$

$$Ne = 18210{,}05 W$$

$$Ne = 24{,}41\ HP$$

Realizando la comparación de potencia efectiva de fábrica y la Potencia efectiva diagnosticado se tiene:

$$\frac{24{,}41\ HP}{55{,}16\ HP} * 100\%$$

Solo conserva el 44,25 % de la potencia de fábrica el motor en esas condiciones diagnosticado, por lo que se tiene un motor en un mal estado.

2.5. Posibles causas de una pérdida de potencia y torque

En el cálculo de de potencia se utiliza las siguientes ecuaciones:

$$P = \frac{W}{t} = Fv$$

$$T = \boldsymbol{F} \times \boldsymbol{d} = Pw$$

En vista de que trabajaremos con vehículos livianos de las marcas Toyota, Nissan, Mitsubishi, Suzuki, King Long, se observan los siguientes códigos de falla relacionados con pérdida de potencia. La potencia está relacionada con el torque por la siguiente ecuación:

$$T = P/w$$

Donde:

T: es el torque

P: es la potencia

W: es la velocidad angular

La potencia es una variable importante en vehículos livianos, el torque en vehículos pesados, en condiciones de velocidad angular constante rpm (revoluciones por minuto) el torque es directamente proporcional a la potencia de un vehículos, es decir que una disminución de potencia también causará una disminución de Torque, por esta razón se considerará las posibles fallas que causan una disminución de potencia.

A) Fallos de encendido

Que la chispa de las bujías se genere en el momento exacto es fundamental para evitar la pérdida de potencia de un automóvil de gasolina. Una puesta a punto deficiente hace que se tenga una potencia menor.

Acontinuacion se muestran los códigos de falla genérica mostrada en un escáner:

P0010 Actuador posición árbol levas (bloque 1) - circuito defectuoso.

El código P0010 producirá una luz Check Engine Light en el panel de instrumentos

P0011 Posición árbol levas (bloque 1) - encendido avanzado, rendimiento

P0012 Posición árbol levas (bloque 1) - encendido atrasado.

P0014 Actuador posición árbol levas (bloque 1) - encendido avanzado

P0015 Actuador posición árbol levas (bloque 1) - encendido atrasado

P0016 Posición del cigueñal incorrecta. Este código OBD2 nos indica que existe un fallo en el sensor del cigueñal.

P0017 Posición cigüeñal-árbol levas (bloque 1 sensor B) – correlación

P0018 Posición cigüeñal-árbol levas (bloque 2 sensor A) - correlación

P0019 Posición cigüeñal-árbol levas (bloque 2 sensor B) – correlación

P0020 0 Actuador posición árbol levas (bloque 2) - circuito defectuoso

P0021 Posición árbol levas (bloque 2) - encendido avanzado, rendimiento

P0022 Posición árbol levas (bloque 2) - encendido atrasado

P0023 Actuador posición árbol levas (bloque 2) - circuito defectuoso

P0024 Actuador posición árbol levas (bloque 2) - encendido avanzado, rendimiento

P0025 5 Actuador posición árbol levas (bloque 2) - encendido atrasado

P0026 6 Circuito solenoide control válvula admisión (bloque 1) – rango

P0027 7 Circuito solenoide control válvula escape (bloque 1) – rango

P0028 8 Circuito solenoide control válvula admisión (bloque 2) - rango

P0029 9 Circuito solenoide control válvula escape (bloque 2) - rango

B) Bujías en mal estado.

P0301 falla de encendido en el cilindro No 1

P0302 falla de encendido en el cilindro No 2

P0303 falla de encendido en el cilindro No 3

P0304 falla de encendido en el cilindro No 4

Válvulas que no cierran por completo: Que las válvulas no cierren el cilindro de forma completa y estanca condiciona que la explosión no sea la correcta y que se produzcan pérdidas de potencia. En el caso de tener un coche de gasolina, al motor le costará trasladarse por encima de las 4.500 rpm.

Árbol de levas gastado: el árbol de levas es el que empuja a las válvulas, si el coche tiene muchos kilómetros de recorrido, es posible que esté gastado y por lo tanto no se cierren a la perfección.

Mala preparación de la mezcla: Una mala preparación de la mezcla de combustible, tanto en un coche alimentado por gasolina como en un diésel puede ser la causa de una pérdida de potencia. Una mezcla de combustible y aire excesivamente rica (con mayor combustible en la relación combustible- aire) incrementará el consumo de combustible y reducirá la potencia. Mientras que una mezcla pobre (con menor combustible en la relación de combustible - aire), hará que el automóvil alcance menores velocidades máximas.

P0172 Poco oxígeno en el banco No 1

P0173 Poco oxígeno en el banco No 2

CAPITULO VI: CONCLUSIONES

El diseño de este prototipo automotriz que se desarrolló con las pruebas en el presente proyecto, se puede concluir que se logró colocar en marcha tanto la comunicación con los datos del simulador de la ECU con la App(Android) así también con el vehículo que se realizaron las pruebas de comunicación.

Se eligió la alternativa de utilizar ELM 327 que es la alternativa de interfaz para su respectiva comunicación mediante sus protocolos de comunicación el cual es el encargado de transmitir y transferir los respectivos datos desde el vehículo o simulador hacia el computador mediante una conexión Bluetooth.

Se logró identificar a través de los códigos de errores una lectura rápida de las fallas de la perdida de potencia y torque del motor con el respectivo interfaz de comunicación primeramente con el simulador que se construyó para la simulación de estas señales que son transmitidas por el puerto OBDII del respectivo automóvil, se tuvo que realizar la respectiva investigación para realizar el simulador de ECU el cual cuenta con el protocolo de comunicación ISO 9141-2 este con el fin de tener las lecturas correspondientes de los datos que simulan la conexión con el automóvil. Para luego realizar las respectivas pruebas con el vehículo.

El cuestionario con preguntas con Escala Likert y para su validación utilizado el alfa de Cron Bach presentado en los resultados una confiabilidad del instrumento del 60 %, lo que conlleva al resultado que se encuentra correlacionados de manera confiable y muy aceptable conforme a la escala que presenta.

De acuerdo a los cálculos observados se vio una reducción de la perdida de potencia efectiva del motor, en condiciones de la ciudad de El Alto en un 55.75. Entre las posibles fallas se enumeran a continuación:

- Puesta a punto deficiente (chispa de encendido)
- Bujías en mal estado
- Falla en el cierre de válvulas
- Árbol de levas desgastado
- Mezcla de combustible deficiente
- Presión deficiente de combustible
- Caudal de combustible deficiente
- Inyectores en mal estado
- Desgaste en los cilindros y/o anillas

En el presente trabajo se diagnosticó la potencia en tiempo real, asi como se muestra en la figura 53. teniendo encuenta que el motor diagnosticado es del año 1999 la perdida de potencia es significativa.

CAPITULO VII: RECOMENDACIONES

Se recomienda identificar el protocolo de comunicación del vehículo a utilizar. Realizar medicines del vehiculo a diagnosticar con las herramientas adecuadas, comparar los diferentes diagnosticos realizados para reducir los errores de instrumentación, asi como evitar cometer otro tipo de errores, para mostrar datos precisos y confiables.

Se recomienda dar continuidad a las actividades de investigación que son referentes a entender los protocolos de comunicación, lo que permitirá la creación de dispositivos y software realizado por los estudiantes y docentes de las carreas de Ingenieria Autotronica e Ingenieria de Sistemas.

Poder gestionar a travez de los institutos de investigación la producción y distribución de prototipos funcionales para poder ofertar soluciones a bajo costo desde la Universidad hacia la población en general.

Poder enfocar la producción de dispositivos minimizando los costos, con la compra de dispositivos de menor coste, pero que tinen las mismas funcionalidades, esto reducirá significativamente la producción, pero se deberá realizar mas adecuaciones para su funcionamiento.

BIBLIOGRAFÍA

(s.f.). Obtenido de http://domoticx.com/raspberry-pi-can-bus-communicatie-gpio/

ALEXANDER, A. P. (2020). *Desarrollo de una interfaz electrónica para la.* IBARRA.

Anaya Multimedia/Wrox, Y. F. (2011). *Programación Java.* Anaya Multimedia.

ARDUINO. (05 de 02 de 2018). Obtenido de https://www.arduino.cc/

Arias, F. G. (2012). *El Proyecto de Investigacion - Introduccion a la metodologia cientifica.* Caracas Venezuela: Editorial Episteme, C.A.

BOSCH. (2002). *Los Sensores en el Automovil.*

can bus. (2011). Microchip-tecnology.

Cantos Calderon, M., & Ulco Ulco, S. (2016). *Estudio cuantitativo de los sistemas de comunicación en vehículos de la región andina mediante norma ISO 9141.* Quito.

CHOLL, H. a. (1978). *ELECTRONIC APPLICATIONS TO THE AUTOMOBILE BY ROBERT BOSCH GMBH.* SAE Technical Paper 780834, 1978.

Comunicación Serial. (16 de 05 de 2021). *Comunicación Serial*. Obtenido de http://digital.ni.com/public.nsf/allkb/039001258CEF8FB686256E0F005888D1

datasheetdir. (17 de 05 de 2021). *datasheetdir*. Obtenido de http://www.datasheetdir.com/SAB80515+8051-Microcontrollers

DIAGNOSTIC, O. B. (17 de 05 de 2021). *OBD* . Obtenido de http://www.mecanicavirtual.org/obd2.htm

Elm. (s.f.). Elmelectronics. Recuperado el 16 de Marzo de 2011, de http://www.elmelectronics.com/ DSheets/ELM327DSF.pdf

Estevez Freire Lila Grimanesa. (2006). *DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE INFORMACIÓN DE PARÁMETROS DE UN AUTOMÓVIL.* Quito Ecuador: REPOSITORIO ESCUELA POLITECNICA NACIONAL.

Fidias G. Arias. (2006). *El Proyecto de Investigación.* Caracas, Venezuela: Suplidora Van C.A.

FLORES, M. B. (2014). *DESARROLLO DE INTERFAZ VIRTUAL PARA MONITOREO DE REDES CAN EN APLICACIONES AUTOMOTRICES.* 10: 07.

Free Software Foundation Staff. (2012). *What is freee software and why is it so important for society?*

Galeano, M. E. (2004). Diseño de proyec-tos en la investigación cualitativa. *EAFIT*, 1.

GARZAS, J. (2012). *MAVEN*. Obtenido de https://www.javiergarzas.com/2014/06/maven-en-10-min.html

Genta, G. M. (2014). *Engine. En The Motor Car: Past, Present and Future.* New York: Springer.

GmbH, R. B. (2005). *Manual de la Tecnica del Automovil.* Alemania: Postfach.

Henderson Bob Haynes John. (2006). *OBD2 & Electronic Engine Management System Manual.* Newburdy: Haynes.

INE. (16 de 4 de 2021). *Sitio Oficial INE*. Obtenido de https://www.ine.gob.bo/index.php/estadisticas-economicas/transporte/parque-automotor-cuadros-estadisticos

Jim Moore. (2008). *Estrategias de Diagnostico del Sistema OBD-2 .* Delph: iVol. Primera Edición.

Jim Moore. (2008). *Estrategias de Diagnostico del Sistema OBD-2.* Delphi.

John, Henderson Bob Haynes. (2006). *OBD2 & Electronic Engine Management System Manual.* NewBurdy Park: haynes.

Jorge Luis Soria Mamani. (2013). *Sistema Experto Para El Diagnostico De Fallas En Motores A Inyección Electrónica De Vehículos.* La Paz: Repositorio UMSA.

Jovaj, M. S. (1982). *Motores del automóvil*. Editorial MIR Moscu

kinsta. (2021). Obtenido de https://kinsta.com/es/base-de-conocimiento/que-es-github/

Liu, JM,Wang,JH y Sun. (2014). *DH*. Obtenido de El diseño del sistema de comunicación CAN Bus basado en MCP2515 y S3C2440.: https://doi.org/10.4028/www.scientific.net/amr.933.516

Marcelo M. Gomez. (2006). *INtroduccion a la Metodologia de la Investigacion cientifica .* Cordoba: Editorial Brujas.

OBDII Bus. (s.f.). Obtenido de http://www.interfacebus.com/OBDII-pinout-signal-assingment.html

peña, c. (2020). *Arduino IDE :Configuracion y uso.* buenos aires: claudio peña.

POLICIA BOLIVIANA. (8 de junio 1978). *Reglamento del Código del Tránsito.*

Rodriguez Rodriguez, Armando; Vento Alvarez, Jose Raul. (2018). *IMPLEMENTACIÓN DE UNA HERRAMIENTA PARA DIAGNÓSTICO OBD-II SOBRE BUS CAN CON ARDUINO.* Cuba.

RUAT. (16 de 4 de 2021). *SITIO OFICIAL RUAT*. Obtenido de https://www.ruat.gob.bo/Principal.jsf

Sabino, C. (2014). *El proceso de investigacion.* Guatemala: Editorial Episteme.

Supo, D. J. (2012). *Como validar un instrumento.* Peru, : Biblioteca Nacional del Peru.

Tomasi Wayne. (2003). *Sistemas de Comunicaciones Electrónicas .* México: Pearson Educativa.

TOOL, A. D. (17 de 05 de 2021). *AUTOMOTIVE DIAGNOSTIC TOOL*. Obtenido de http://lasolucion.freeoda.com/docs/obd2.pdf

Torres Valverde Edwin Ismael. (2015). *Sistema Electrónico por comando de voz para la seguridad física, iluminación y accionamiento automático de los accesorios primarios de un automóvil.* Ambato, Ecuador: FISEL Electronica.

Tutorial del Protocolo RS-232. (13 de 06 de 2010). Obtenido de Tutorial del Protocolo RS-232: http://www.rootshell.be/~wcruzy/cd/tutorialserialrs232.pdf.

Wikipedi. (13 de 03 de 2021). *Wikipedia*. Obtenido de https://es.wikipedia.org/wiki/NetBeans

ANEXOS

ANEXO A.

Especificaciones técnicas del motor 1SZ-FE

La denominación de la compañía, fabricante del automóvil.	Toyota
Serie La serie a la que pertenece este modelo.	Yaris
Modelo El modelo de automóvil.	Yaris
Inicio de la fabricación El año en el cual empieza la fabricación de este modelo.	1999
Carrocería / Latonería Información sobre el tipo de carrocería/latonería de este automóvil.	hatchback
Tipo de sistema de tracción El tipo de sistema de tracción, que es utilizado en el vehículo.	FWD (tracción delantera)
Asientos Número de asientos con las cuales dispone el automóvil.	5
Puertas Número de puertas con las cuales dispone el automóvil.	3
Longitud	3614.00 mm (milímetros)
Longitud del automóvil - la distancia entre el punto más posterior del automóvil hasta el punto más delantero.	142.2835 in (pulgadas) 11.8570 ft (pies)
Anchura	1660.00 mm (milímetros)

Anchura del automóvil - espejos, manijas de las puertas, luces y demás dispositivos no están incluidos en las mediciónes. El ancho se mide a puertas cerradas y las ruedas de posición hacia adelante.	65.3543 in (pulgadas) 5.4462 ft (pies)
Altura Altura del automóvil - la distancia entre el punto más bajo, junto a la tierra, y el punto más alto del automóvil.	1499.00 mm (milímetros) 59.0157 in (pulgadas) 4.9180 ft (pies)
Distancia entre ejes La distancia entre los centros de las ruedas delanteras y traseras; la distancia entre el eje delantero y trasero.	2369.00 mm (milímetros) 93.2677 in (pulgadas) 7.7723 ft (pies)
Vía delantera La distancia entre las ruedas del eje delantero.	1450.00 mm (milímetros) 57.0866 in (pulgadas) 4.7572 ft (pies)
Vía trasera La distancia entre las ruedas del eje trasero.	1431.00 mm (milímetros) 56.3386 in (pulgadas) 4.6949 ft (pies)
Peso	914 kg (kilos)

El peso propio del automóvil se mide por medio de equipo estándar y en presencia de todos los materiales necesarios, sin la presencia de pasajeros y de equipaje adicional.	2015.03 lb (libras)
Fabricante del motor Denominación de la compañía–fabricante del motor.	Toyota
Código del motor Información sobre el código del motor.	1SZ-FE
Cilindrada / Desplazamiento / Capacidad del motor La capacidad máxima de la mezcla de combustible que el motor puede manejar en un ciclo completo. El desplazamiento del motor es la suma de los desplazamiento de los pistónes (la parte del cilindro entre el punto muerte superior y el punto muerte inferior).	~ 1.0 l (litros) 997 cc (centímetros cúbicos)
Número de cilindros El número total de los cilindros en el motor. El cilindro es el espacio en el cual el pistón se mueve entre el PMS (punto muerte superior) y el PMI (punto muerte inferior).	4
Disposición de cilindros Información sobre el orden/la disposición de los cilindros en el motor. Las más frecuentes disposiciónes de los cilindros son: en linea, en V y en oposición (boxer).	en línea
Número de válvulas por cilindro	4

La mayoría de los motores de combustión interna contemporáneos tienen dos o más válvulas por cilindro que controlan los procesos en el cilindro. Las válvulas de admisión sirven para controlar el aire y el combustible entrando en el cilindro. Las válvulas de escape garantizan la limpieza del cilindro de los gases de escape.	
Diámetro del cilindro Información sobre el diámetro de los cilindros en el motor. En la mayoría de los motores contemporáneos el diámetro del cilindro varia en los marcos de 70 mm - 105 mm.	69.00 mm (milímetros) 2.7165 in (pulgadas) 0.2264 ft (pies)
Carrera del piston La distancia de movimiento del pistón entre el punto muerto superior y el punto muerto inferior dentre el cilindro.	66.70 mm (milímetros) 2.6260 in (pulgadas) 0.2188 ft (pies)
Relación de compression La relación de compresión indica cuántas veces la mezcla de combustión reduce su volumen durante el movimiento del pistón del punto muerte inferior al punto muerte superior. La relación de compresión es un parámetro firmemente integrado en el diseño del motor y no cambia con el tiempo. .	10.00:1
PME / BMEP	164.51 psi (libras por pulgada cuadrada)

"Presión media efectiva" o "Break mean efficiency pressure" es la presión sobre el pistón de los motores de combustión interna. En el caso de los motores con turbo esta presión es siempre mayor que la presión en los motores atmosféricos.	1134.26 kPa (kilopascales) 11.34 bar (bares)
Llenado Según el tipo de llenado el motor puede ser atmosférico (naturalmente aspirado) y sobrealimentado (turbo/supercargado).	atmosférico (con aspiración natural/compresor de aire)
Tipo de motor El tipo del motor según el número y la disposición de los árboles de levas, las válvulas de admisión, las válvulas de escape, etc.	DOHC (doble árbol de levas a la cabeza)
Refrigeración Principalmente se utilizan dos tipos de refrigeración del motor – refrigeración por aire y refrigeración por agua. En el caso de la refrigeración por aire el calor se libera a través del contacto con el aire, mientras que en el caso de refrigeración por agua se utilizan fluidos líquidos con compuestos anticongelantes.	refrigeración por agua
Intercooler	no está disponible

El intercooler se instala entre el turbocompresor y el colector de aspiración y sirve para la refrigeración de aire entrando de la turbina. A través de disminución de la temperatura aumenta la densidad del aire y de ese modo en el motor entran más moléculas de oxígeno.	
Posición del motor Información sobre si el motor se posicióne al frente (motor delantero), detrás (motor trasero) o en la mitad del automóvil (motor central).	en la parte delantera
Orientación del motor Información sobre si el motor es orientado de manera longitudinal o transversal del vehículo.	transversal
Sistema de combustion El sistema de combustión sirve para formar, transportar y alimentar los cilindros con mezcla de combustible.	EFI (inyección electrónica de combustible)
Convertidor catalítico / Catalizador El convertidor catalítico (catalizador) maneja los gases de escape del motor antes de dejar el automóvil. Que elimina una parte de las emisiones nocivas de gases a través de una reacción química.	disponible
Potencia máxima La potencia máxima que puede desarrollar el motor.	50 kW (kilovatios) 68 hp (caballos de fuerza-métricos) 67 bhp (caballos de fuerza-británicos)
Revoluciones (potencia máxima)	

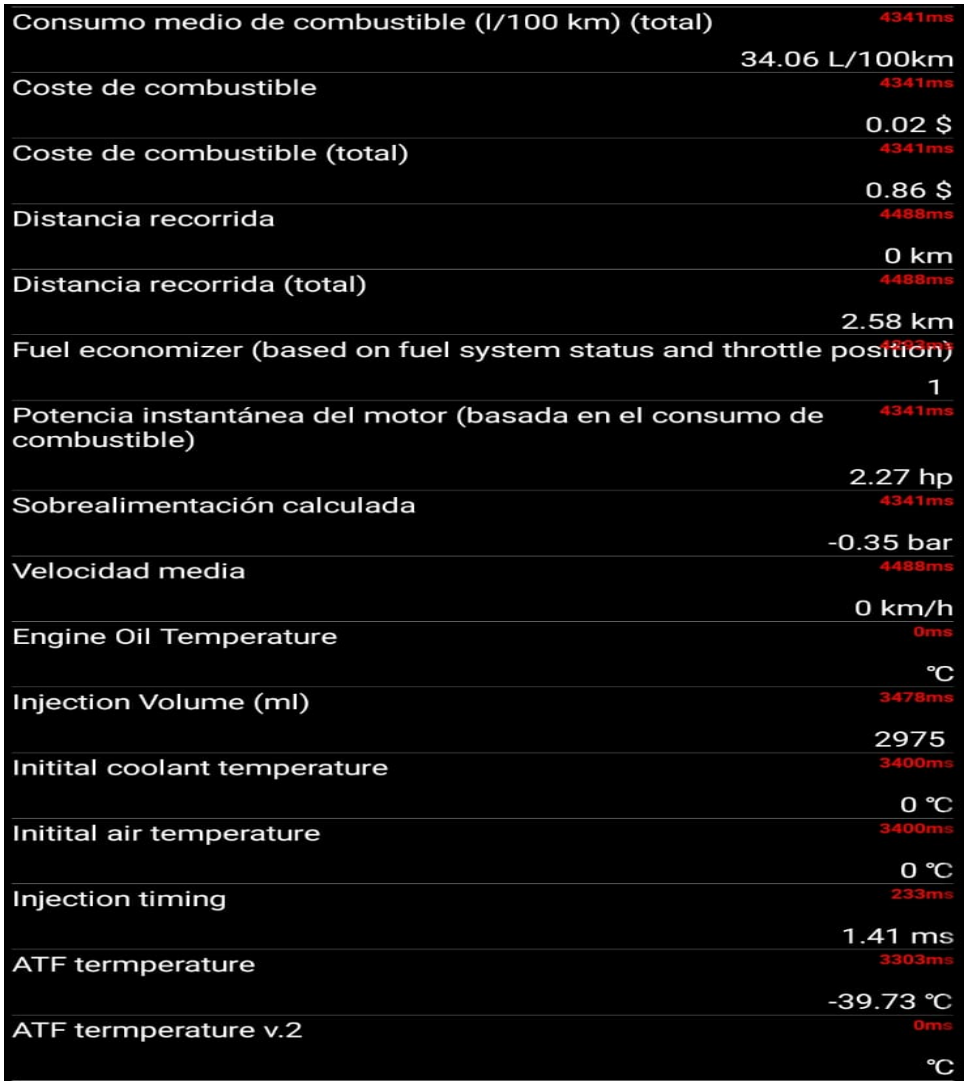
Las revoluciones por minuto necesarias para que el motor desarrole su potencia máxima.	6000 rpm (rotación por minuto)
Par motor máximo El par máximo que puede ser generado por el motor.	89 Nm (newton-metros) 65 ft-lb (pie-libras) 9 kgm (kilogram-metros)
Revoluciones (par máximo) Las revoluciones por minuto necesarias para que el motor desarrole su par máximo.	4100 rpm (rotación por minuto)
Velocidad máxima La velocidad máxima que puede ser desarrollada por el automóvil.	156 km/h (kilómetros por hora) 96.93 m/h (millas por hora)
0 - 100 km/h El timpo en segundos, necesario al vehículo para acelerar de 0 a 100 kilómetros por hora.	13.60 s (segundos)
Volumen / capacidad de depósito /tanque La cantidad máxma de combustible/carburante que el depósito/tanque del automóvil puede contener.	45.00 l (litros) 11.89 US gal (galones estadounidenses) 9.90 UK gal (galónes británicos)
Suspensión delantera	sistema independiente

Información sobre la suspensión delantera. La suspensión es el sistema que relaciona las ruedas y los ejes al cuadro/el cupé del automóvil. La suspensión aisla el cupé de las influencias externas tales como vibraciones en resultado de rugosidades, etc.	sistema MacPherson (de pierna)
Suspensión trasera Información sobre los elementos, utilizados en la suspensión trasera del automóvil.	sistema independiente barra de torsión
Transmisión La transmisión/caja de velocidades/caja de cambios transmite el par de rotación del motor a las ruedas del automóvil.	manual
Cambios / Engranajes El número de los cambios/engranajes de la caja de velocidades del automóvil.	5
Sistema de frenado - ruedas delanteras Sistema de frenado, utilizada para las ruedas delanteras. El sistema de frenado sirve para tardar el movimiento del automóvil, así como para su parada completa.	frenos de discos
Sistema de frenado - ruedas traseras Sistema de frenado, utilizada para las ruedas traseras.	frenos de tambor
Diámetro de giro El mínimo diámetro posible del círculo que se describe por las ruedas externas del automóvil al efectuar una vuelta con volante encontrándose en posición final.	9.80 m (metros) 385.8268 in (pulgadas)

	32.1522 ft (pies)
Sistema de dirección El sistema de dirección es el sistema a través del cual se puede cambiar la dirección de movimiento del automóvil.	de piñón y cremallera

ANEXO B.

Diagnóstico de otras variables con el ELM327



CÓDIGO FUENTE SIMULADOR

```
/********* VARIABLES Y SET *****************/

LiquidCrystal_I2C lcd(0x27, 16, 2);// Creamos la clase display de 16x2

#define CAN_2515

const int SPI_CS_PIN = 10;

const int CAN_INT_PIN = 2;

MCP_CAN CAN(SPI_CS_PIN);

//mcp2515_can CAN(SPI_CS_PIN);   // Set CS pin

uint32_t  canId = 0x000;       // Set ID CAN

unsigned char len = 0;         // Size array

unsigned char buf[8];          // Array Buffer

//unsigned long rxID;

uint8_t DegreeBitmap[]= { 0x6, 0x9, 0x9, 0x6, 0x0, 0, 0, 0 };      // create a dot character pattern for degree symbol

String versionSoftware= "1.0";


char str[20];


String BuildMessage="";

String BuildMessage2="";

int MSGIdentifier=0;

/********** VARIABLES PARA EL TRATAMIENTO DE CADENAS *********/

String valor = "";

String opcion = "";

String texto = "";

/********** VARIABLES PID SOPORTADAS ******************/

char vDistanceTraveled =  0;
```

```
char vVehiculeSpeed = 0;
char vCoolTemperatura = 0;
char vEngineRunTime = 0;
char vThrottlePosition = 0;
char vIAT = 0;
char vMAF = 0;
char vMAP = 0;
char vFuelTankLevel = 0;
char vEngineSpeed = 0;
/********** DECLARACION DE VARIABLES PIDS GLOBALES *******/
const char PID_NAME_0x00[] PROGMEM = "PIDs supported [01 - 20]";
const char PID_NAME_0x01[] PROGMEM = "Monitor status since DTCs cleared";
const char PID_NAME_0x02[] PROGMEM = "Freeze DTC";
const char PID_NAME_0x03[] PROGMEM = "Fuel system status";
const char PID_NAME_0x04[] PROGMEM = "Calculated engine load";
const char PID_NAME_0x05[] PROGMEM = "Engine coolant temperature";
const char PID_NAME_0x06[] PROGMEM = "Short term fuel trim — Bank 1";
const char PID_NAME_0x07[] PROGMEM = "Long term fuel trim — Bank 1";
const char PID_NAME_0x08[] PROGMEM = "Short term fuel trim — Bank 2";
const char PID_NAME_0x09[] PROGMEM = "Long term fuel trim — Bank 2";
const char PID_NAME_0x0a[] PROGMEM = "Fuel pressure";
const char PID_NAME_0x0b[] PROGMEM = "Intake manifold absolute pressure";
const char PID_NAME_0x0c[] PROGMEM = "Engine RPM";
const char PID_NAME_0x0d[] PROGMEM = "Vehicle speed";
const char PID_NAME_0x0e[] PROGMEM = "Timing advance";
const char PID_NAME_0x0f[] PROGMEM = "Intake air temperature";
const char PID_NAME_0x10[] PROGMEM = "MAF air flow rate";
```

```
const char PID_NAME_0x11[] PROGMEM = "Throttle position";

const char PID_NAME_0x12[] PROGMEM = "Commanded secondary air status";

const char PID_NAME_0x13[] PROGMEM = "Oxygen sensors present (in 2 banks)";

const char PID_NAME_0x14[] PROGMEM = "Oxygen Sensor 1 - Short term fuel trim";

const char PID_NAME_0x15[] PROGMEM = "Oxygen Sensor 2 - Short term fuel trim";

const char PID_NAME_0x16[] PROGMEM = "Oxygen Sensor 3 - Short term fuel trim";

const char PID_NAME_0x17[] PROGMEM = "Oxygen Sensor 4 - Short term fuel trim";

const char PID_NAME_0x18[] PROGMEM = "Oxygen Sensor 5 - Short term fuel trim";

const char PID_NAME_0x19[] PROGMEM = "Oxygen Sensor 6 - Short term fuel trim";

const char PID_NAME_0x1a[] PROGMEM = "Oxygen Sensor 7 - Short term fuel trim";

const char PID_NAME_0x1b[] PROGMEM = "Oxygen Sensor 8 - Short term fuel trim";

const char PID_NAME_0x1c[] PROGMEM = "OBD standards this vehicle conforms to";

const char PID_NAME_0x1d[] PROGMEM = "Oxygen sensors present (in 4 banks)";

const char PID_NAME_0x1e[] PROGMEM = "Auxiliary input status";

const char PID_NAME_0x1f[] PROGMEM = "Run time since engine start";

const char PID_NAME_0x20[] PROGMEM = "PIDs supported [21 - 40]";

const char PID_NAME_0x21[] PROGMEM = "Distance traveled with malfunction indicator lamp (MIL) on";

const char PID_NAME_0x22[] PROGMEM = "Fuel Rail Pressure (relative to manifold vacuum)";

const char PID_NAME_0x23[] PROGMEM = "Fuel Rail Gauge Pressure (diesel, or gasoline direct injection)";

const char PID_NAME_0x24[] PROGMEM = "Oxygen Sensor 1 - Fuel–Air Equivalence Ratio";

const char PID_NAME_0x25[] PROGMEM = "Oxygen Sensor 2 - Fuel–Air Equivalence Ratio";

const char PID_NAME_0x26[] PROGMEM = "Oxygen Sensor 3 - Fuel–Air Equivalence Ratio";

const char PID_NAME_0x27[] PROGMEM = "Oxygen Sensor 4 - Fuel–Air Equivalence Ratio";
```

```
const char PID_NAME_0x28[] PROGMEM = "Oxygen Sensor 5 - Fuel–Air Equivalence Ratio";

const char PID_NAME_0x29[] PROGMEM = "Oxygen Sensor 6 - Fuel–Air Equivalence Ratio";

const char PID_NAME_0x2a[] PROGMEM = "Oxygen Sensor 7 - Fuel–Air Equivalence Ratio";

const char PID_NAME_0x2b[] PROGMEM = "Oxygen Sensor 8 - Fuel–Air Equivalence Ratio";

const char PID_NAME_0x2c[] PROGMEM = "Commanded EGR";

const char PID_NAME_0x2d[] PROGMEM = "EGR Error";

const char PID_NAME_0x2e[] PROGMEM = "Commanded evaporative purge";

const char PID_NAME_0x2f[] PROGMEM = "Fuel Tank Level Input";

const char PID_NAME_0x30[] PROGMEM = "Warm-ups since codes cleared";

const char PID_NAME_0x31[] PROGMEM = "Distance traveled since codes cleared";

const char PID_NAME_0x32[] PROGMEM = "Evap. System Vapor Pressure";

const char PID_NAME_0x33[] PROGMEM = "Absolute Barometric Pressure";

const char PID_NAME_0x34[] PROGMEM = "Oxygen Sensor 1 - Fuel–Air Equivalence Ratio";

const char PID_NAME_0x35[] PROGMEM = "Oxygen Sensor 2 - Fuel–Air Equivalence Ratio";

const char PID_NAME_0x36[] PROGMEM = "Oxygen Sensor 3 - Fuel–Air Equivalence Ratio";

const char PID_NAME_0x37[] PROGMEM = "Oxygen Sensor 4 - Fuel–Air Equivalence Ratio";

const char PID_NAME_0x38[] PROGMEM = "Oxygen Sensor 5 - Fuel–Air Equivalence Ratio";

const char PID_NAME_0x39[] PROGMEM = "Oxygen Sensor 6 - Fuel–Air Equivalence Ratio";

const char PID_NAME_0x3a[] PROGMEM = "Oxygen Sensor 7 - Fuel–Air Equivalence Ratio";

const char PID_NAME_0x3b[] PROGMEM = "Oxygen Sensor 8 - Fuel–Air Equivalence Ratio";
```

```
const char PID_NAME_0x3c[] PROGMEM = "Catalyst Temperature: Bank 1, Sensor 1";

const char PID_NAME_0x3d[] PROGMEM = "Catalyst Temperature: Bank 2, Sensor 1";

const char PID_NAME_0x3e[] PROGMEM = "Catalyst Temperature: Bank 1, Sensor 2";

const char PID_NAME_0x3f[] PROGMEM = "Catalyst Temperature: Bank 2, Sensor 2";

const char PID_NAME_0x40[] PROGMEM = "PIDs supported [41 - 60]";

const char PID_NAME_0x41[] PROGMEM = "Monitor status this drive cycle";

const char PID_NAME_0x42[] PROGMEM = "Control module voltage";

const char PID_NAME_0x43[] PROGMEM = "Absolute load value";

const char PID_NAME_0x44[] PROGMEM = "Fuel–Air commanded equivalence ratio";

const char PID_NAME_0x45[] PROGMEM = "Relative throttle position";

const char PID_NAME_0x46[] PROGMEM = "Ambient air temperature";

const char PID_NAME_0x47[] PROGMEM = "Absolute throttle position B";

const char PID_NAME_0x48[] PROGMEM = "Absolute throttle position C";

const char PID_NAME_0x49[] PROGMEM = "Absolute throttle position D";

const char PID_NAME_0x4a[] PROGMEM = "Absolute throttle position E";

const char PID_NAME_0x4b[] PROGMEM = "Absolute throttle position F";

const char PID_NAME_0x4c[] PROGMEM = "Commanded throttle actuator";

const char PID_NAME_0x4d[] PROGMEM = "Time run with MIL on";

const char PID_NAME_0x4e[] PROGMEM = "Time since trouble codes cleared";

const char PID_NAME_0x4f[] PROGMEM = "Maximum value for Fuel–Air equivalence ratio,
oxygen sensor voltage, oxygen sensor current, and intake manifold absolute pressure";

const char PID_NAME_0x50[] PROGMEM = "Maximum value for air flow rate from mass air
flow sensor";

const char PID_NAME_0x51[] PROGMEM = "Fuel Type";

const char PID_NAME_0x52[] PROGMEM = "Ethanol fuel percentage";

const char PID_NAME_0x53[] PROGMEM = "Absolute Evap system Vapor Pressure";

const char PID_NAME_0x54[] PROGMEM = "Evap system vapor pressure";

const char PID_NAME_0x55[] PROGMEM = "Short term secondary oxygen sensor trim";
```

```
const char PID_NAME_0x56[] PROGMEM = "Long term secondary oxygen sensor trim";

const char PID_NAME_0x57[] PROGMEM = "Short term secondary oxygen sensor trim";

const char PID_NAME_0x58[] PROGMEM = "Long term secondary oxygen sensor trim";

const char PID_NAME_0x59[] PROGMEM = "Fuel rail absolute pressure";

const char PID_NAME_0x5a[] PROGMEM = "Relative accelerator pedal position";

const char PID_NAME_0x5b[] PROGMEM = "Hybrid battery pack remaining life";

const char PID_NAME_0x5c[] PROGMEM = "Engine oil temperature";

const char PID_NAME_0x5d[] PROGMEM = "Fuel injection timing";

const char PID_NAME_0x5e[] PROGMEM = "Engine fuel rate";

const char PID_NAME_0x5f[] PROGMEM = "Emission requirements to which vehicle is designed";


const char* const PID_NAME_MAPPER[] PROGMEM = {

  PID_NAME_0x00,

  PID_NAME_0x01,

  PID_NAME_0x02,

  PID_NAME_0x03,

  PID_NAME_0x04,

  PID_NAME_0x05,

  PID_NAME_0x06,

  PID_NAME_0x07,

  PID_NAME_0x08,

  PID_NAME_0x09,

  PID_NAME_0x0a,

  PID_NAME_0x0b,

  PID_NAME_0x0c,

  PID_NAME_0x0d,
```

PID_NAME_0x0e,

PID_NAME_0x0f,

PID_NAME_0x10,

PID_NAME_0x11,

PID_NAME_0x12,

PID_NAME_0x13,

PID_NAME_0x14,

PID_NAME_0x15,

PID_NAME_0x16,

PID_NAME_0x17,

PID_NAME_0x18,

PID_NAME_0x19,

PID_NAME_0x1a,

PID_NAME_0x1b,

PID_NAME_0x1c,

PID_NAME_0x1d,

PID_NAME_0x1e,

PID_NAME_0x1f,

PID_NAME_0x20,

PID_NAME_0x21,

PID_NAME_0x22,

PID_NAME_0x23,

PID_NAME_0x24,

PID_NAME_0x25,

PID_NAME_0x26,

PID_NAME_0x27,

PID_NAME_0x28,

PID_NAME_0x29,

PID_NAME_0x2a,

PID_NAME_0x2b,

PID_NAME_0x2c,

PID_NAME_0x2d,

PID_NAME_0x2e,

PID_NAME_0x2f,

PID_NAME_0x30,

PID_NAME_0x31,

PID_NAME_0x32,

PID_NAME_0x33,

PID_NAME_0x34,

PID_NAME_0x35,

PID_NAME_0x36,

PID_NAME_0x37,

PID_NAME_0x38,

PID_NAME_0x39,

PID_NAME_0x3a,

PID_NAME_0x3b,

PID_NAME_0x3c,

PID_NAME_0x3d,

PID_NAME_0x3e,

PID_NAME_0x3f,

PID_NAME_0x40,

PID_NAME_0x41,

PID_NAME_0x42,

PID_NAME_0x43,

PID_NAME_0x44,

PID_NAME_0x45,

PID_NAME_0x46,

PID_NAME_0x47,

PID_NAME_0x48,

PID_NAME_0x49,

PID_NAME_0x4a,

PID_NAME_0x4b,

PID_NAME_0x4c,

PID_NAME_0x4d,

PID_NAME_0x4e,

PID_NAME_0x4f,

PID_NAME_0x50,

PID_NAME_0x51,

PID_NAME_0x52,

PID_NAME_0x53,

PID_NAME_0x54,

PID_NAME_0x55,

PID_NAME_0x56,

PID_NAME_0x57,

PID_NAME_0x58,

PID_NAME_0x59,

PID_NAME_0x5a,

PID_NAME_0x5b,

PID_NAME_0x5c,

PID_NAME_0x5d,

PID_NAME_0x5e,

```
  PID_NAME_0x5f,
};

const char PERCENTAGE[] PROGMEM = "%";
const char KPA[] PROGMEM = "kPa";
const char PA[] PROGMEM = "Pa";
const char RPM[] PROGMEM = "rpm";
const char KPH[] PROGMEM = "km/h";
const char DEGREES_BEFORE_TDC[] PROGMEM = "° before TDC";
const char GRAMS_PER_SECOND[] PROGMEM = "grams/sec";
const char SECONDS[] PROGMEM = "seconds";
const char RATIO[] PROGMEM = "ratio";
const char COUNT[] PROGMEM = "count";
const char KM[] PROGMEM = "km";
const char VOLTS[] PROGMEM = "V";
const char MINUTES[] PROGMEM = "minutes";
const char GPS[] PROGMEM = "g/s";
const char DEGREES[] PROGMEM = "°";
const char DEGREES_CELCIUS[] PROGMEM = "\001C";
const char LPH[] PROGMEM = "L/h";

const char* const PID_UNIT_MAPPER[] PROGMEM = {
  NULL,
  NULL,
  NULL,
  NULL,
  PERCENTAGE,
```

DEGREES_CELCIUS,

PERCENTAGE,

PERCENTAGE,

PERCENTAGE,

PERCENTAGE,

KPA,

KPA,

RPM,

KPH,

DEGREES_BEFORE_TDC,

DEGREES_CELCIUS,

GRAMS_PER_SECOND,

PERCENTAGE,

NULL,

NULL,

PERCENTAGE,

PERCENTAGE,

PERCENTAGE,

PERCENTAGE,

PERCENTAGE,

PERCENTAGE,

PERCENTAGE,

PERCENTAGE,

NULL,

NULL,

NULL,

SECONDS,

NULL,

KM,

KPA,

KPA,

RATIO,

RATIO,

RATIO,

RATIO,

RATIO,

RATIO,

RATIO,

RATIO,

PERCENTAGE,

PERCENTAGE,

PERCENTAGE,

PERCENTAGE,

COUNT,

KM,

PA,

KPA,

RATIO,

RATIO,

RATIO,

RATIO,

RATIO,

RATIO,

RATIO,

RATIO,

DEGREES_CELCIUS,

DEGREES_CELCIUS,

DEGREES_CELCIUS,

DEGREES_CELCIUS,

NULL,

NULL,

VOLTS,

PERCENTAGE,

RATIO,

PERCENTAGE,

DEGREES_CELCIUS,

PERCENTAGE,

PERCENTAGE,

PERCENTAGE,

PERCENTAGE,

PERCENTAGE,

PERCENTAGE,

MINUTES,

MINUTES,

NULL,

GPS,

NULL,

PERCENTAGE,

KPA,

PA,

PERCENTAGE,

```
  PERCENTAGE,
  PERCENTAGE,
  PERCENTAGE,
  KPA,
  PERCENTAGE,
  PERCENTAGE,
  DEGREES_CELCIUS,
  DEGREES,
  LPH,
  NULL,
};

//Mode 1 PID

unsigned char SupportedPID1[8] = {0x02, 0x41, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};
unsigned char SupportedPID2[8] = {0x02, 0x41, 0x20, 0x00, 0x00, 0x00, 0x00, 0x00};
unsigned char SupportedPID3[8] = {0x02, 0x41, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00};
unsigned char SupportedPID4[8] = {0x02, 0x41, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00};
unsigned char SupportedPID5[8] = {0x02, 0x41, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00};
unsigned char SupportedPID6[8] = {0x02, 0x41, 0xA0, 0x00, 0x00, 0x00, 0x00, 0x00};
unsigned char SupportedPID7[8] = {0x02, 0x41, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00};

//Mode 2
unsigned char DistTravelled[8] = {0x03, 0x41, 0x31, 0x50, 0x00, 0x00, 0x00, 0x00};
unsigned char Throttle[8] = {0x03, 0x41, 0x11, 0x02, 0x00, 0x00, 0x00, 0x00};

unsigned char EngineSpeed[8] =  {0x04, 0x41, 0x0C, 0x4e, 0x20, 0x00, 0x00, 0x00};
```

```
unsigned char VehiculeSpeed[8] = {0x03, 0x41, 0x0D, 0x50, 0x00, 0x00, 0x00, 0x00};
unsigned char CoolantTemp[8] = {0x03, 0x41, 0x05, 0x46, 0x00, 0x00, 0x00, 0x00};

unsigned char EngRunTime[8] = {0x04, 0x41, 0x1F, 0x00, 0x78, 0x00, 0x00, 0x00};
unsigned char IAT[8] = {0x03, 0x41, 0x0F, 0x60, 0x00, 0x00, 0x00, 0x00};
unsigned char MAF[8] = {0x04, 0x41, 0x10, 0x00, 0xE5, 0x00, 0x00, 0x00};
unsigned char MAP[8] = {0x03, 0x41, 0x0B, 0x22, 0x00, 0x00, 0x00, 0x00};

unsigned char FuelTankLevel[8] = {0x03, 0x41, 0x2f, 0x22, 0x00, 0x00, 0x00, 0x00};
unsigned char DistTraveledWithMIL[8] = {0x03, 0x41, 0x21, 0x22, 0x00, 0x00, 0x00, 0x00};

//mode 3 Frame
unsigned char DTCFrame[8] = {0x04, 0x43, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00};

//mode 4 Frame
unsigned char DTCClearFrame[8] = {0x01, 0x44, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};

//mode 9 PID
unsigned char CalibID[8] = {0x03, 0x49, 0x04, 0x22, 0x00, 0x00, 0x00, 0x00};
unsigned char CVN[8] = {0x03, 0x49, 0x06, 0x22, 0x00, 0x00, 0x00, 0x00};
unsigned char ECU_NAME[8] = {0x03, 0x49, 0x0A, 0x22, 0x00, 0x00, 0x00, 0x00};
/********** SETUP ************************************/
void setup() {
   lcd.begin();   // iniciamos el display
   lcd.createChar ( 1, DegreeBitmap );
   lcd.backlight(); //encendemos el display
   imprimirMensaje("Iniciando...");
```

```
delay(2000);

SERIAL_PORT_MONITOR.begin(115200); //Iniciamos y esperamos a la apertura del puerto


while(!Serial){

 imprimirMensaje("Conecte USB..."); //imprimimos mensaje mientras no se conecte a la computadora

}


imprimirMensaje(" USB conectado!");  //Si se conecta a una computadora

SERIAL_PORT_MONITOR.println("USB iniciado..");

delay(1000);


START_INIT:

 if(CAN_OK == CAN.begin(MCP_STDEXT, CAN_250KBPS, MCP_16MHZ)){ //bit de frecuencia y seteamos crystal frecuencia

   SERIAL_PORT_MONITOR.println("CAN BUS Shield iniciado ok!");

   imprimirMensaje("CAN iniciado!");

  }else{

    imprimirMensaje("CAN fallo!");

    SERIAL_PORT_MONITOR.println("CAN BUS Shield fallo al iniciar");

    SERIAL_PORT_MONITOR.println("inicie de nuevo CAN BUS Shield");

    delay(100);

    goto START_INIT;

  }

 delay(1000);

 imprimirMensaje("...");

 SERIAL_PORT_MONITOR.println("**********************");
```

```
        SERIAL_PORT_MONITOR.print("Simulador de OBDII
");SERIAL_PORT_MONITOR.print(versionSoftware);SERIAL_PORT_MONITOR.println("
Iniciado...");

        SERIAL_PORT_MONITOR.println("**********************");

    }


    /********** LOOP ********************/

    void loop() {

      //GENERAL ROUTINE

      unsigned char SupportedPID[10] = {1,2,3,4,5,6,7,8,9,10};

      unsigned char MilCleared[7] = {4,65,63,34,224,185,147};


     while(1){

      if(SERIAL_PORT_MONITOR.available()){


          char _valor;

          valor = SERIAL_PORT_MONITOR.readString();

          int str_len = valor.length()+1;

          char array_str[str_len];

          valor.toCharArray(array_str,str_len);

          const char s[2]="|";

          char *token;


          token = strtok(array_str,s);

          while(token!=NULL){

            valor = token;

            opcion = valor.substring(0,4);

            texto = valor.substring(4);
```

```
                if(opcion=="disp"){

                if(texto=="apagarDisplay"){apagarDisplay();imprimirMensaje("Apagando...");}
//Apagar Display

                else
if(texto=="encederDisplay"){encederDisplay();imprimirMensaje("Encendiendo...");} //Encendemos
Display

                else if(texto=="limpiarDisplay"){limpiarDisplay();}//Limpiamos el Display

                else { imprimirMensaje(texto);}

                }else if(opcion=="0x31"){

                    vDistanceTraveled = texto.toDouble();

                    imprimirPid(pidName(0x31),pidUnits(0x31), texto);

                    SERIAL_PORT_MONITOR.println(pidName(0x31)+"-
"+pidUnits(0x31)+":"+texto);

                    DistTravelled[4] = texto.toDouble();

                    DistTravelled[3] = texto.toDouble();

                    //sendPid(canId, DistTravelled);

                    delay(100);

                }else if(opcion=="0x0D"){

                    vVehiculeSpeed = texto.toDouble();

                    imprimirPid(pidName(0x0D),pidUnits(0x0D), texto);

                    SERIAL_PORT_MONITOR.println(pidName(0x0D)+"-
"+pidUnits(0x0D)+":"+texto);

                    VehiculeSpeed[3]  = texto.toDouble();

                    //sendPid(canId, VehiculeSpeed);

                    delay(100);

                }else if(opcion=="0x05"){

                    vCoolTemperatura = texto.toDouble();

                    imprimirPid(pidName(0x05),pidUnits(0x05), texto);
```

```
                    SERIAL_PORT_MONITOR.println(pidName(0x05)+"-
"+pidUnits(0x05)+":"+texto);

                    CoolantTemp[3] = texto.toDouble();

                    //sendPid(canId, CoolantTemp);

                    delay(100);

                }else if(opcion=="0x1F"){

                    vEngineRunTime = texto.toDouble();

                    imprimirPid(pidName(0x1F),pidUnits(0x1F), texto);

                    SERIAL_PORT_MONITOR.println(pidName(0x1F)+"-
"+pidUnits(0x1F)+":"+texto);

                    EngRunTime[4] = texto.toDouble();

                    EngRunTime[3] = texto.toDouble();

                    //sendPid(canId, EngRunTime);

                    delay(100);

                }else if(opcion=="0x11"){

                    vThrottlePosition = texto.toDouble();

                    imprimirPid(pidName(0x11),pidUnits(0x11), texto);

                    SERIAL_PORT_MONITOR.println(pidName(0x11)+"-
"+pidUnits(0x11)+":"+texto);

                    Throttle[3] = texto.toDouble();

                    //sendPid(canId, Throttle);

                    delay(100);

                }else if(opcion=="0x0F"){

                    vIAT = texto.toDouble();

                    imprimirPid(pidName(0x0F),pidUnits(0x0F), texto);

                    SERIAL_PORT_MONITOR.println(pidName(0x0F)+"-
"+pidUnits(0x0F)+":"+texto);

                    IAT[3] = texto.toDouble();

                    //sendPid(canId, IAT);
```

```
                delay(100);
            }else if(opcion=="0x10"){
                vMAF = texto.toDouble();
                imprimirPid(pidName(0x10),pidUnits(0x10), texto);
                SERIAL_PORT_MONITOR.println(pidName(0x10)+"-
"+pidUnits(0x10)+":"+texto);
                MAF[4] = texto.toDouble();
                MAF[3] = texto.toDouble();
                //sendPid(canId, MAF);
                delay(100);
            }else if(opcion=="0x0B"){
                vMAP = texto.toDouble();
                imprimirPid(pidName(0x0B),pidUnits(0x0B), texto);
                SERIAL_PORT_MONITOR.println(pidName(0x0B)+"-
"+pidUnits(0x0B)+":"+texto);
                MAP[3] = texto.toDouble();
                //sendPid(canId, MAP);
                delay(100);
            }else if(opcion=="0x2F"){
                vFuelTankLevel = texto.toDouble();
                imprimirPid(pidName(0x2F),pidUnits(0x2F), texto);
                SERIAL_PORT_MONITOR.println(pidName(0x2F)+"-
"+pidUnits(0x2F)+":"+texto);
                FuelTankLevel[3] = texto.toDouble();
                //sendPid(canId, FuelTankLevel);
                delay(100);
            }else if(opcion=="0x0C"){
                vEngineSpeed = texto.toDouble();
```

```
                    imprimirPid(pidName(0x0C),pidUnits(0x0C), texto);
                    SERIAL_PORT_MONITOR.println(pidName(0x0C)+"-
"+pidUnits(0x0C)+":"+texto);
                    EngineSpeed[4] = texto.toDouble();
                    EngineSpeed[3] = texto.toDouble();
                    //sendPid(canId, EngineSpeed);
                    /*Borrar*/
                    CAN.sendMsgBuf(0x7E8, 0, 8, vEngineSpeed);
                    SERIAL_PORT_MONITOR.println(" "+pidName(0x0C));
                    delay(100);
                  }else{
                    imprimirMensaje("Pid desconocido");
                    SERIAL_PORT_MONITOR.println(" Sin datos");
                  }

                 token = strtok(NULL,s);
               }

              }
             /*SERIAL_PORT_MONITOR.println("CAN_MSGAVAIL");
             SERIAL_PORT_MONITOR.println(CAN_MSGAVAIL);
             SERIAL_PORT_MONITOR.println("CAN.checkReceive");
             SERIAL_PORT_MONITOR.println(CAN.checkReceive());
             SERIAL_PORT_MONITOR.println("==================");*/
             if(CAN_MSGAVAIL == CAN.checkReceive()){
                /*   SERIAL_PORT_MONITOR.println("CAN_MSGAVAIL");
                  SERIAL_PORT_MONITOR.println(CAN_MSGAVAIL);
```

```
    SERIAL_PORT_MONITOR.println("CAN.checkReceive");
    SERIAL_PORT_MONITOR.println(CAN.checkReceive());
    SERIAL_PORT_MONITOR.println("==================");*/
  CAN.readMsgBuf( &canId, &len, buf);
  //canId = CAN.getCanId();
  SERIAL_PORT_MONITOR.println();
  SERIAL_PORT_MONITOR.print("<");
  SERIAL_PORT_MONITOR.print(canId,HEX);
  SERIAL_PORT_MONITOR.print(", PID ");
  SERIAL_PORT_MONITOR.print(buf[2], HEX);
  SERIAL_PORT_MONITOR.print(", ");
  for(int i=0; i< len; i++){
    SERIAL_PORT_MONITOR.print(buf[i], HEX);
    SERIAL_PORT_MONITOR.print(" ,");
  }
  SERIAL_PORT_MONITOR.print("--");

  /*Other*/
  BuildMessage2="";
  SERIAL_PORT_MONITOR.print("<");Serial.print(canId);Serial.print(",");

   for(int i = 0; i<len; i++)
   {
     BuildMessage2 = BuildMessage2 + buf[i] + ",";
   }
   SERIAL_PORT_MONITOR.println(BuildMessage2);
  /*****/
```

```
BuildMessage="";
 //Si estamos en estado 1 enviamos los pids solicitados
 if (buf[2] == 0x05 && buf[1] == 0x0) {
   CAN.sendMsgBuf(0x7E8, 0, 8, CoolantTemp);
   SERIAL_PORT_MONITOR.print(" "+pidName(0x05));
 }
 if (buf[2] == 0x0B && buf[1] == 0x01) {
   CAN.sendMsgBuf(0x7E8, 0, 8, MAP);
   SERIAL_PORT_MONITOR.print(" "+pidName(0x0B));
 }
 if (buf[2] == 0x0C && buf[1] == 0x01) {
   CAN.sendMsgBuf(0x7E8, 0, 8, EngineSpeed);
   SERIAL_PORT_MONITOR.print(" "+pidName(0x0C));
 }
 if (buf[2] == 0x0D && buf[1] == 0x01) {
   CAN.sendMsgBuf(0x7E8, 0, 8, VehiculeSpeed);
   SERIAL_PORT_MONITOR.print(" "+pidName(0x0D));
 }
 if (buf[2] == 0x0F && buf[1] == 0x01) {
   CAN.sendMsgBuf(0x7E8, 0, 8, IAT);
   SERIAL_PORT_MONITOR.print(" "+pidName(0x0F));
 }
 if (buf[2] == 0x10 && buf[1] == 0x01) {
   CAN.sendMsgBuf(0x7E8, 0, 8, MAF);
   SERIAL_PORT_MONITOR.print(" "+pidName(0x10));
 }
 if (buf[2] == 0x1F && buf[1] == 0x01) {
```

```
  CAN.sendMsgBuf(0x7E8, 0, 8, EngRunTime);
  SERIAL_PORT_MONITOR.print(" "+pidName(0x1F));
}
if (buf[2] == 0x31 && buf[1] == 0x01) {
  CAN.sendMsgBuf(0x7E8, 0, 8, DistTravelled);
  SERIAL_PORT_MONITOR.print(" "+pidName(0x31));
}
if (buf[2] == 0x11 && buf[1] == 0x01) {
  CAN.sendMsgBuf(0x7E8, 0, 8, Throttle);
  SERIAL_PORT_MONITOR.print(" "+pidName(0x11));
}
if (buf[2] == 0x2F && buf[1] == 0x01) {
  CAN.sendMsgBuf(0x7E8, 0, 8, FuelTankLevel);
  SERIAL_PORT_MONITOR.print(" "+pidName(0x2F));
}
if (buf[2] == 0x21 && buf[1] == 0x01) {
  CAN.sendMsgBuf(0x7E8, 0, 8, DistTraveledWithMIL);
  SERIAL_PORT_MONITOR.print(" "+pidName(0x21));
}

//Enviamos los PIds para el modo de pids soportados
if (buf[2] == 0x00 && buf[1] == 0x01) {
  CAN.sendMsgBuf(0x7E8, 0, 8, SupportedPID1);
  SERIAL_PORT_MONITOR.println("SupportedPID1");
}
if (buf[2] == 0x20 && buf[1] == 0x01) {
  CAN.sendMsgBuf(0x7E8, 0, 8, SupportedPID2);
```

```
  SERIAL_PORT_MONITOR.println("SupportedPID2");
}
if (buf[2] == 0x40 && buf[1] == 0x01) {
  CAN.sendMsgBuf(0x7E8, 0, 8, SupportedPID3);
  SERIAL_PORT_MONITOR.println("SupportedPID3");
}
if (buf[2] == 0x60 && buf[1] == 0x01) {
  CAN.sendMsgBuf(0x7E8, 0, 8, SupportedPID4);
  SERIAL_PORT_MONITOR.println("SupportedPID4");
}
if (buf[2] == 0x80 && buf[1] == 0x01) {
  CAN.sendMsgBuf(0x7E8, 0, 8, SupportedPID5);
  SERIAL_PORT_MONITOR.println("SupportedPID5");
}
if (buf[2] == 0xA0 && buf[1] == 0x01) {
  CAN.sendMsgBuf(0x7E8, 0, 8, SupportedPID6);
  SERIAL_PORT_MONITOR.println("SupportedPID6");
}
if (buf[2] == 0xC0 && buf[1] == 0x01) {
  CAN.sendMsgBuf(0x7E8, 0, 8, SupportedPID7);
  SERIAL_PORT_MONITOR.println("SupportedPID7");
}

//Mode 3 PID
  if (buf[1] == 0x03) {
    CAN.sendMsgBuf(0x7E8, 0, 8, DTCFrame);
    SERIAL_PORT_MONITOR.print(" DTCFrame");
```

```
  }

//Mode 4 PID
  if (buf[1] == 0x04) {
    CAN.sendMsgBuf(0x7E8, 0, 8, DTCClearFrame);
    SERIAL_PORT_MONITOR.print(" DTCClearFrame");
    DTCFrame[2] = 0;
    DTCFrame[3] = 0;
    DTCFrame[4] = 0;
  }

//Mode 3 PID
if (buf[1] == 0x04) {
  CAN.sendMsgBuf(0x7E8, 0, 8, DTCFrame);
  SERIAL_PORT_MONITOR.print(" DTCFrame");
}

//Mode 9 PID
if (buf[2] == 0x04 && buf[1] == 0x09) {
  CAN.sendMsgBuf(0x7E8, 0, 8, CalibID);
  SERIAL_PORT_MONITOR.print("CalibID");
}
if (buf[2] == 0x06 && buf[1] == 0x09) {
  CAN.sendMsgBuf(0x7E8, 0, 8, CVN);
  SERIAL_PORT_MONITOR.print("CVN");
}
if (buf[2] == 0x0A && buf[1] == 0x09) {
```

```
            CAN.sendMsgBuf(0x7E8, 0, 8, ECU_NAME);
            SERIAL_PORT_MONITOR.print("ECU_NAME");
          }
          SERIAL_PORT_MONITOR.println();

      }
    }
  }

/********** FUNCTIONES ***************/
/* ======================================
 * Function imprimirMensaje(String)
 * line2
   ====================================== */
void imprimirMensaje(String texto){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(">SimOBD2-IIIS<");
    lcd.setCursor(0,1);
    lcd.print(texto);
}
/* ======================================
 * Function imprimirPid(String,String, String)
 * line2
   ====================================== */
void imprimirPid(String NamePid, String UnidadPid, String valorPid){
    lcd.clear();
```

```
    lcd.setCursor(0,0);
    lcd.print(NamePid);
    lcd.setCursor(0,1);
    lcd.print(UnidadPid+" "+valorPid);
}

/* ======================================
 * Function apagarDisplay()
 * Apaga el LCD / luz de fondo off
   ====================================== */
void apagarDisplay(){
  lcd.noBacklight();
  SERIAL_PORT_MONITOR.println("Display apagado");
}

/* ======================================
 * Function encederDisplay()
 * enciende el LCD / luz de fondo on
   ====================================== */
void encenderDisplay(){
  lcd.backlight();
  SERIAL_PORT_MONITOR.println("Display encendido");
}

/* ======================================
 * Function limpiarDisplay()
 * Clear el LCD / Display none
```

```
   ========================================== */
void limpiarDisplay(){
  lcd.clear();
  SERIAL_PORT_MONITOR.println("Display clear");
}

/* ========================================
 * Function pidName(pid HEX)
 * retorna el nombre del PID
   ========================================== */
String pidName(uint8_t pid)
{
  if (pid > 0x5f) {
    return "Unknown";
  }

  #ifdef __AVR__
    const char* pgmName = pgm_read_ptr(&PID_NAME_MAPPER[pid]);
    String name;

    if (pgmName != NULL) {
      while (char c = pgm_read_byte(pgmName++)) {
        name += c;
      }
    }

    return name;
```

```
  #else
    return PID_NAME_MAPPER[pid];
  #endif
}
/* ========================================
 * Function pidUnits(pid HEX)
 * retorna la unidad del PID
   ======================================== */
String pidUnits(uint8_t pid)
{
  if (pid > 0x5f) {
    return "";
  }
#ifdef __AVR__
  const char* pgmUnits = pgm_read_ptr(&PID_UNIT_MAPPER[pid]);
  String units;

  if (pgmUnits != NULL) {
    while (char c = pgm_read_byte(pgmUnits++)) {
      units += c;
    }
  }

  return units;
#else
  return PID_UNIT_MAPPER[pid];
#endif
```

```
}


/* ========================================
 * Function sendPid(0x16, __pid[X], texto)
 * Envia el Pid por el puerto CAN
   ======================================== */
void sendPid(char canId,  unsigned char __pid[], String texto) {
    CAN.sendMsgBuf(canId, 0, 8, __pid);
    SERIAL_PORT_MONITOR.print(texto);
}
```