



INSTITUTO POLITÉCNICO NACIONAL

UNIDAD PROFESIONAL INTERDISCIPLINARIA EN
INGENIERÍA Y TECNOLOGÍAS AVANZADAS

Proyecto Integrador

**“Sistema de proyección de los principales indicadores
automotrices para vehículos con sistema OBD II”**

Que para obtener el título de:

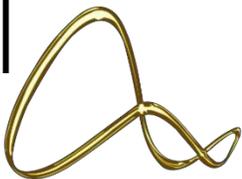
Ingeniero en Sistemas Automotrices

Presentan:

**Cedillo Cedillo Oswaldo Azael
Ochoa Martínez Guillermo**

Asesores:

**Dra. Briseño Tepepa Blanca Rosa
Ing. Vázquez Cianca Israel**



upiita-ipn

Ciudad de México, a 3 de junio del 2022

Autorización de uso de obra

Instituto Politécnico Nacional

Presente

Bajo protesta de decir verdad el que suscribe Cedillo Cedillo Osvaldo Azael (se anexa copia simple de identificación oficial), manifiesto ser autor (a) y titular de los derechos morales y patrimoniales de la obra titulada Sistema de proyección de los principales indicadores automáticos para vehículos con Sistema OBD II,

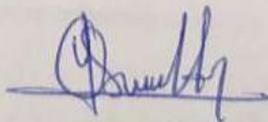
en adelante "La Tesis" y de la cual se adjunta copia, por lo que por medio del presente y con fundamento en el artículo 27 fracción II, inciso b) de la Ley Federal del Derecho de Autor, otorgo a el Instituto Politécnico Nacional, en adelante El IPN, autorización no exclusiva para comunicar y exhibir públicamente total o parcialmente en medios digitales, Plataforma de la Dirección de Bibliotecas del IPN y/o consulta directa en la Coordinación de Biblioteca de la UPIITA "La Tesis" por un periodo de 5 años contado a partir de la fecha de la presente autorización, dicho periodo se renovará automáticamente en caso de no dar aviso expreso a "El IPN" de su terminación.

En virtud de lo anterior, "El IPN" deberá reconocer en todo momento mi calidad de autor de "La Tesis".

Adicionalmente, y en mi calidad de autor y titular de los derechos morales y patrimoniales de "La Tesis", manifiesto que la misma es original y que la presente autorización no contraviene ninguna otorgada por el suscrito respecto de "La Tesis", por lo que deslindo de toda responsabilidad a El IPN en caso de que el contenido de "La Tesis" o la autorización concedida afecte o viole derechos autorales, industriales, secretos industriales, convenios o contratos de confidencialidad o en general cualquier derecho de propiedad intelectual de terceros y asumo las consecuencias legales y económicas de cualquier demanda o reclamación que puedan derivarse del caso.

Ciudad de México., 3 de Junio de 2022

Atentamente





Autorización de uso de obra

Instituto Politécnico Nacional

P r e s e n t e

Bajo protesta de decir verdad el que suscribe Ochoa Martínez Guillermo
(se anexa copia simple de identificación oficial), manifiesto ser autor (a) y titular de los derechos morales y patrimoniales de la obra titulada Sistema de proyección de los principales indicadores automotrices para vehículos con sistema OBD II

_____ ,
en adelante “La Tesis” y de la cual se adjunta copia, por lo que por medio del presente y con fundamento en el artículo 27 fracción II, inciso b) de la Ley Federal del Derecho de Autor, otorgo a el Instituto Politécnico Nacional, en adelante El IPN, autorización no exclusiva para comunicar y exhibir públicamente total o parcialmente en medios digitales, Plataforma de la Dirección de Bibliotecas del IPN y/o consulta directa en la Coordinación de Biblioteca de la UPIITA “La Tesis” por un periodo de 5 años contado a partir de la fecha de la presente autorización, dicho periodo se renovará automáticamente en caso de no dar aviso expreso a “El IPN” de su terminación.

En virtud de lo anterior, “El IPN” deberá reconocer en todo momento mi calidad de autor de “La Tesis”.

Adicionalmente, y en mi calidad de autor y titular de los derechos morales y patrimoniales de “La Tesis”, manifiesto que la misma es original y que la presente autorización no contraviene ninguna otorgada por el suscrito respecto de “La Tesis”, por lo que deslindo de toda responsabilidad a El IPN en caso de que el contenido de “La Tesis” o la autorización concedida afecte o viole derechos autorales, industriales, secretos industriales, convenios o contratos de confidencialidad o en general cualquier derecho de propiedad intelectual de terceros y asumo las consecuencias legales y económicas de cualquier demanda o reclamación que puedan derivarse del caso.

Ciudad de México, a 3 de Junio de 2022

Atentamente





INSTITUTO POLITÉCNICO NACIONAL

UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y TECNOLOGÍAS AVANZADAS

Proyecto Integrador

“Sistema de proyección de los principales indicadores automotrices para vehículos con sistema OBD II”

Que para obtener el título de:

Ingeniero en Sistemas Automotrices

Presentan:

Cedillo Cedillo Oswaldo Azael

Ochoa Martínez Guillermo

Jurado evaluador:

PRESIDENTE

Dra. Bella Citlali Martínez Seis

SECRETARIO

M. en C Arodí Rafael Carvallo Domínguez

ASESOR

Dra. Briseño Tepepa Blanca Rosa

ASESOR

Ing. Vázquez Cianca Israel

CONTENIDO

Palabras clave	1
Keywords	1
Resumen	1
Abstract	2
Objetivos	3
Objetivo general	3
Objetivos específicos	3
INTRODUCCIÓN	4
CAPÍTULO 1. ANTECEDENTES	8
1.1. Consideraciones teóricas	9
1.2. Estado del arte	37
CAPÍTULO 2. PLANTEAMIENTO DEL PROBLEMA	43
2.1. Propuesta de solución	45
2.2. Alcance del proyecto	47
CAPÍTULO 3. DESARROLLO DEL SISTEMA	48
3.1. Selección y costos de componentes	49
3.2. Diagrama de montaje del sistema	50
3.3. Configuración del Software	52
CAPÍTULO 4. VALIDACIÓN DEL SISTEMA	73
CAPÍTULO 5. ANÁLISIS DE RESULTADOS.....	78
5.1. Conclusiones	79
5.2. Trabajos futuros	81
Referencias bibliográficas	83
ANEXO A. LISTA SAE DE PIDs	85
ANEXO B. VEHÍCULOS COMPATIBLES CON EL ESCÁNER ELM327.....	89
ANEXO C. HOJA DE DATOS DEL FILM DE RETROPROYECCIÓN	100

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Panel de instrumentos mostrando el indicador de Malfunction Indicator Lamp	9
Ilustración 2. Conectores OBD I de diferentes fabricantes	10
Ilustración 3. Año de implementación del sistema OBD II en diferentes países	11
Ilustración 4. Parámetros que abarca el sistema OBD II	12
Ilustración 5. Distribución de los pines del conector OBD II	14
Ilustración 6. Distribución de pines de los protocolos PWM y VPW, respectivamente	15
Ilustración 7. Distribución de pines del protocolo ISO 9141-2	16
Ilustración 8. Distribución de pines del protocolo KWP2000	16
Ilustración 9. Distribución de pines del protocolo ISO 15765 CAN	17
Ilustración 10. Comparación del cableado de las unidades de control sin CAN y con CAN, respectivamente	17
Ilustración 11. Niveles de tensión del bus CAN	18
Ilustración 12. Trama de datos del bus CAN	19
Ilustración 13. Pila de protocolos Bluetooth	23
Ilustración 14. Modelo físico del escáner ELM327 USB.....	25
Ilustración 15. Modelo físico del escáner ELM327 Bluetooth	26
Ilustración 16. Modelo físico del escáner ELM327 WIFI.....	26
Ilustración 17. Vista superior de los pines del microcontrolador ELM327	28
Ilustración 18. Head-Up Display montado en un avión de guerra	31
Ilustración 19. Sistema Head-Up Display de BMW	31
Ilustración 20. Sistema Head-Up Display convencional	32
Ilustración 21. Sistema Head-Medium Display convencional	33
Ilustración 22. Sistema Head-Up Display desarrollado por Continental	34
Ilustración 23. Campo visual del ser humano	35
Ilustración 24. Campo visual con respecto a la velocidad de conducción	36
Ilustración 25. Sistema Head-Up Display del Oldmobile Cutlass Supreme y del Pontiac Grand Prix, respectivamente	37
Ilustración 26. Sistema de proyección NAVDY	39
Ilustración 27. Sistema de proyección HUDWAY Glass.....	40
Ilustración 28. Demostración de la obtención de la dirección Bluetooth del ELM327	53
Ilustración 29. Demostración del envío y respuesta de los comandos AT	55
Ilustración 30. Modo de envío de comandos para la obtención de parámetros provenientes de la ECU del vehículo	56
Ilustración 31. Ejemplo de comando completo para la obtención de la velocidad del vehículo	56
Ilustración 32. Respuesta recibida una vez realizada la petición de la velocidad	57
Ilustración 33. Interpretación a valor decimal, del valor de las revoluciones por minuto ...	58
Ilustración 34. Ejemplo del despliegue de la velocidad en la interfaz gráfica	64
Ilustración 35. Ejemplo de las diferentes barras para la temperatura del refrigerante	65
Ilustración 36. Ejemplo de los medidores de mezcla aire-combustible	65
Ilustración 37. Ejemplo de las diferentes barras para la temperatura del medio ambiente.	65
Ilustración 38. Ejemplo del despliegue de las revoluciones por minuto del motor en la interfaz gráfica	66

Ilustración 39. Indicador del estado de la conexión Bluetooth	66
Ilustración 40. Diseño final de la interfaz gráfica	67
Ilustración 41. Petición y respuesta de diferentes parámetros para el vehículo Seat Ibiza.	74

ÍNDICE DE FOTOGRAFÍAS

Fotografía 1. Vista frontal del sistema para el alojamiento del escáner ELM327	67
Fotografía 2. Vista lateral izquierda para la entrada de alimentación de la Raspberry PI 3B y la salida del puerto HDMI	68
Fotografía 3. Vista trasera para la salida de los puertos USB y ethernet	68
Fotografía 4. 1) Muestra la distribución interna de la Raspberry y el escaner ELM327 dentro del gabinete de plástico. 2) Vista frontal del sistema. 3) Vista trasera del sistema. 4) Vista lateral izquierda	68
Fotografía 5. Sistema Raspberry PI 3B - ELM327 conectado al puerto OBD II de un Seat Ibiza 2015	69
Fotografía 6. Inicio de la etapa de alimentación	69
Fotografía 7. Distribución la etapa de alimentación dentro del vehículo	70
Fotografía 8. Instalación del tripié y el proyector	71
Fotografía 9. 1) Se realizó un molde de hieso con la forma de la base deseada. 2) Se obtuvo el molde de la base. 3) Se rellenó el molde de hieso con resina epoxica para que finalmente se obtuviera la base final	71
Fotografía 10. Cristal colocado sobre la base, y esta a su vez está pegada sobre el tablero	72
Fotografía 11. Sistema de proyección funcionando	72
Fotografía 12. Vista del conductor con el sistema de proyección montado	76
Fotografía 13. Simulación de la conducción con el sistema de proyección encendido	77

ÍNDICE DE TABLAS

Tabla 1. Modos de operación de diagnóstico del estándar OBD II	19
Tabla 2. Algunos PIDs del modo de operación 01	21
Tabla 3. Comparativa entre los diferentes circuitos integrados ELM y los protocolos de comunicación soportados	24
Tabla 4. Características físicas y eléctricas del microcontrolador.....	27
Tabla 5. Componentes y costo total del sistema	49
Tabla 6. Comparación de la obtención de los diferentes parámetros en diferentes vehículos	75

ÍNDICE DE DIAGRAMAS

Diagrama 1. Diagrama de bloques interno del microcontrolador ELM327	28
Diagrama 2. Diagrama de bloques del funcionamiento general del sistema	45
Diagrama 3. Diagrama de bloques del montaje del sistema	50
Diagrama 4. Diagrama de flujo de los procesos realizados por parte del servidor local	59
Diagrama 5. Diagrama de flujo de los procesos realizados por parte de la interfaz gráfica	61

Sistema de proyección de los principales indicadores automotrices para vehículos con sistema OBD II

Palabras clave

Sistema OBD II, Escáner automotriz, Tarjeta de desarrollo, Interfaz gráfica, Indicadores automotrices, Sistema de proyección.

Keywords

OBD II system, Automotive scanner, Development card, Graphical interface, Automotive indicators, Projection system.

Resumen

El presente documento describe el trabajo terminal a desarrollar, en el cual se muestran las diferentes secciones que comprenden el proyecto, tales como: introducción, antecedentes, planteamiento del problema, alcance del proyecto, desarrollo, entre otros.

En el proyecto a desarrollar, se diseñará e implementará un sistema de proyección que permita desplegar los principales indicadores automotrices (velocímetro, tacómetro, temperatura del refrigerante, relación de la mezcla aire-combustible y temperatura del medio ambiente) sobre un cristal transparente montado en el tablero, para casi cualquier vehículo con sistema OBD II.

El funcionamiento principal de este sistema se logra gracias a la tarjeta programable Raspberry pi 3 B/B+ y al escáner automotriz ELM327. Los cuales trabajan en conjunto vía Bluetooth para obtener de la ECU del vehículo los parámetros a desplegar sobre el cristal. Adicional a ello, se tiene una interfaz gráfica programada en Java y un servidor local programado en Python, que permiten hacer tanto el envío (servidor local) como la recepción (interfaz gráfica) automática de los parámetros obtenidos mediante la Raspberry y el escáner.

En la actualidad, los sistemas de seguridad y confort en la industria automotriz son prioridad en la producción de cualquier vehículo. A pesar de ello, en dichos sistemas existe una

exclusión entre las diferentes gamas de vehículos, siendo los de gama alta los que gozan de una mayor seguridad y un mayor confort, esto se ve reflejado en sistemas tales como: espejos laterales automáticos, sensores y cámara de reversa, sistema de climatización automático, sistema Head-Up Display, entre otros.

Debido a esta exclusión lo que este proyecto busca, es brindar al conductor de cualquier vehículo con sistema OBD II, un sistema que permita visualizar de una manera más accesible y sin distracciones los indicadores del panel de instrumentos. Puesto que dichos indicadores se tendrían desplegados sobre el cristal, brindando así al conductor, una conducción más eficiente, cómoda y segura, permitiéndole monitorear los parámetros del vehículo sin la necesidad de una distracción que pudiera provocar un accidente, como lo es bajar la mirada al panel de instrumentos.

Abstract

This document describes the terminal work to be developed, which shows the different sections that comprise the project, such as: introduction, background, problem statement, scope of the project, development, among others.

In the project to be developed, a projection system will be designed and implemented to display the main automotive indicators (speedometer, tachometer, coolant temperature, air-fuel mixture ratio and environmental temperature) on the windshield of any vehicle with OBD II system.

The main operation of this system is achieved thanks to the Raspberry pi 3 B/B+ programmable card and the ELM327 automotive scanner. They work together via Bluetooth to obtain from the ECU of the vehicle the parameters to display on the windscreen.

In addition, there is a graphical interface programmed in Java and a local server programmed in Python, which allows both sending (local server) and receiving (graphical interface) automatic parameters obtained through the Raspberry and scanner.

Nowadays, safety and comfort systems in the automotive industry are a priority in the production of any vehicle. In spite of this, in these systems there is an exclusion between the different ranges of vehicles, being those of high range those that enjoy a greater security and a greater comfort, this is reflected in systems such as: automatic lateral mirrors, sensors and

reverse chamber, automatic air conditioning system, Head-Up display system, among others. Due to this exclusion what this project seeks is to provide the driver of any vehicle with OBD II system, a system that allows to visualize in a more accessible way and without distractions the indicators of the instrument panel. Since these indicators would be deployed on the windshield, giving the driver a more efficient, comfortable, and safe driving, allowing him to monitor the parameters of the vehicle without the need for a distraction that could cause an accident, such as looking down at the instrument panel.

Objetivos

Objetivo general

Diseñar e implementar un sistema de proyección sobre un cristal transparente montado en el tablero, para vehículos con sistema OBD II en el cual se muestren los indicadores, tales como: velocímetro, tacómetro, temperatura del refrigerante, relación de la mezcla aire-combustible y temperatura del medio ambiente. Esto con el propósito de brindar al conductor una forma de conducción más eficiente cómoda y segura teniendo a la vista los indicadores mencionados y por tanto previniendo averías en el vehículo o en casos extremos, accidentes.

Objetivos específicos

- Entablar un sistema de comunicación vía Bluetooth para la transmisión y recepción de datos provenientes de la ECU del vehículo utilizando la Raspberry y el escáner ELM327.
- Seleccionar e interpretar los parámetros a utilizar provenientes del escáner ELM327.
- Programar un servidor local que esté sensando constantemente los parámetros a utilizar y poder ser enviados cuando se requieran.
- Diseñar y programar una interfaz gráfica que permita la recepción de los parámetros provenientes del servidor local.
- Montar el sistema de proyección dentro del vehículo.
- Proyectar sobre el cristal transparente la interfaz gráfica desarrollada.

INTRODUCCIÓN

INTRODUCCIÓN

En los últimos años con el avance tecnológico, la industria automotriz ha tenido un auge importante en la implementación e innovación de los sistemas que conforman el vehículo.

Dicho avance se ha visto reflejado claramente en dos sistemas que trabajan en conjunto para brindar al conductor una conducción cómoda y segura, los cuales son el sistema de seguridad y el sistema de confort, sistemas en los cuales se va a enfocar el presente proyecto. Las empresas automotrices han tomado cartas en el asunto y han creado e implementado diversos sistemas de seguridad y confort en el vehículo con el propósito de resguardar la integridad física del conductor, derivado del alto índice de accidentes automovilísticos que se han suscitado en los últimos años, estos sistemas además brindan una conducción cómoda y eficiente. Ejemplos de estos sistemas son, los frenos ABS, las bolsas de aire, el detector de somnolencia, los faros automáticos, el aire acondicionado automático, los espejos laterales eléctricos, los Head-Up Display, por mencionar algunos. Este último sistema, a pesar de ser útil para evitar distracciones durante la conducción al evitar bajar la vista para ver los indicadores del panel de instrumentos, es un sistema que no se encuentra implementado en la mayoría de los vehículos; solamente es implementado en vehículos selectos de gama alta, dejando de lado a los vehículos de gamas inferiores.

El proyecto que se desarrolló se relaciona con el sistema Head-Up Display, debido a que proyecta la medición de algunos indicadores del panel de instrumentos del vehículo, similar a lo que hacen compañías como BMW, KIA, AUDI, Mercedes-Benz, por mencionar algunas. La diferencia entre dichos sistemas y el sistema desarrollado se basa en el enfoque. El enfoque de nuestro proyecto está dirigido para disminuir las distracciones y brindar una mejor conducción al mostrar información del estado del vehículo; mientras que para las compañías antes mencionadas los Head-Up Display son utilizados para reducir accidentes durante la conducción.

El presente sistema proyecta 5 parámetros, los cuales son: velocidad, revoluciones por minuto (rpm), temperatura del refrigerante, relación de la mezcla aire-combustible y temperatura del medio ambiente. El sistema propuesto se puede implementar en casi cualquier vehículo que cuente con un sistema OBD II.

Por lo tanto, el propósito del proyecto es permitir una conducción eficiente y segura brindando al conductor una reducción de las distracciones al tener proyectados los parámetros antes mencionados. Y con ello monitorear el estado del vehículo sobre el campo de visión de conducción y así evitar una avería o un accidente durante el camino al estar alerta de las indicaciones de estos parámetros.

A continuación, se muestran cada uno de los capítulos que comprende el presente escrito y una breve descripción de estos.

Capítulo 1. Antecedentes

Se mencionan y describen los elementos teóricos básicos que debe tener el lector para facilitarle el entendimiento y la comprensión del funcionamiento y el desarrollo del sistema. Adicional a ello, se describe el estado del arte, el cual muestra los proyectos y sistemas con alguna similitud al nuestro y la manera en que se desarrollaron.

Capítulo 2. Planteamiento del problema

Se describe la problemática actual de la industria automotriz al excluir a los vehículos de gama media-baja de diversos sistemas que pueden brindar una mayor seguridad y un mayor confort durante la conducción.

Adicional a ello, se enuncia la propuesta de solución y el alcance, los cuales describen los bloques de la estructurada en la que se llevó a cabo el presente proyecto y la delimitación de este, respectivamente.

Capítulo 3. Desarrollo del sistema

Se muestra la selección y costos de los componentes utilizados en el presente proyecto. Además, se muestran diferentes diagramas que facilitan comprender el funcionamiento del proyecto.

Adicionalmente se describen dos partes clave del proyecto, las cuales son la configuración del software y del hardware; la primera parte muestra todo lo relacionado con el desarrollo de la programación de los diferentes componentes utilizados para realizar la conexión con la ECU del vehículo, los cuales son interfaz gráfica y servidor local. Y la segunda parte muestra todo lo relacionado con la instalación física del sistema (cableado, sistema de proyección, etc.).

Capítulo 4. Validación del sistema

Se describen las diferentes pruebas realizadas del sistema, mencionando si realiza el funcionamiento esperado, y además, se mencionan algunos puntos a destacar que generaron problemas en cuanto a programación, instalación, efectividad, etc., del sistema.

Capítulo 5. Análisis de resultados

Se evalúan los resultados obtenidos comparándolos con los objetivos iniciales, además de que se analizan ventajas y desventajas del sistema y se establecen líneas futuras de trabajo en cuanto a diseño y eficiencia.

CAPÍTULO 1.

ANTECEDENTES

1.1. Consideraciones teóricas

1.1.1. OBD (On Board Diagnostic)

El denominado OBD (On Board Diagnostics) es un sistema de diagnóstico vehicular, como sus siglas en inglés lo indican, incorporado al vehículo y que tiene la función de controlar y monitorear tanto al motor como algunos otros sistemas; dentro de estos, se puede controlar el nivel de emisiones que genera la unidad y determinar si contamina. [4]

1.1.1.1. Historia del OBD

En los años 50 la mayor parte del descubrimiento de problemas en vehículos se hacía con sólo unos pocos medidores y el resto usando los sentidos. A medida que los vehículos se hicieron más complejos, entrando en los años 60, donde había más dependencia de la instrumentación, dicha detección comenzó a tornarse más complicada dejando pasar por alto problemas en el vehículo que pudieran ocasionar averías o contaminación proveniente del mismo.

Para reducir dicha contaminación, la "California Air Resources Board" (CARB) determinó en 1988 que todos los automóviles a gasolina contaran con OBD (On Board Diagnostics), que controlara los límites máximos de emisiones y además un autocontrol. Adicional a ello, para que el conductor detecte un mal funcionamiento del OBD se impuso la obligación de tener una lámpara que indique fallos (MIL - Malfunction Indicator Lamp). [21]

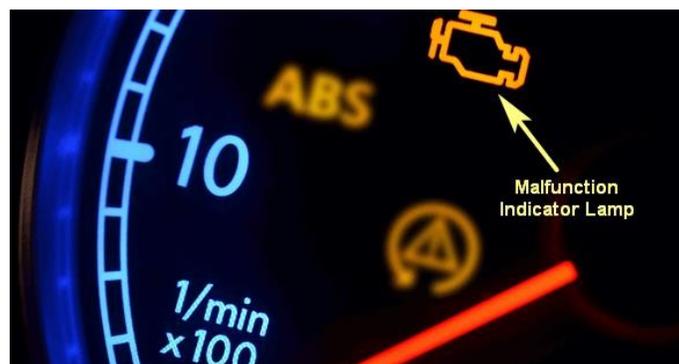


Ilustración 1. Panel de instrumentos mostrando el indicador de Malfunction Indicator Lamp [Ilustración], por asTech, 2017, asTech (<https://astech.com/news/there-wasnt-a-light-on-the-dash-malfunction-indicator-lamp-myths/>).

1.1.1.2. Tipos de estándares

- Estándar OBD I

OBD I fue la primera regulación de OBD que obligaba a los productores a instalar un sistema de monitorización de algunos de los componentes controladores de emisiones en automóviles. Obligatorios en todos los vehículos a partir de 1991, sin embargo, esta tecnología fue creada en 1983 así como implementada en algunos vehículos americanos en 1987 y 1988, los sistemas de OBD I no eran tan efectivos porque solamente monitorizaban algunos de los componentes relacionados con las emisiones, y no eran calibrados para un nivel específico de emisiones. [21]

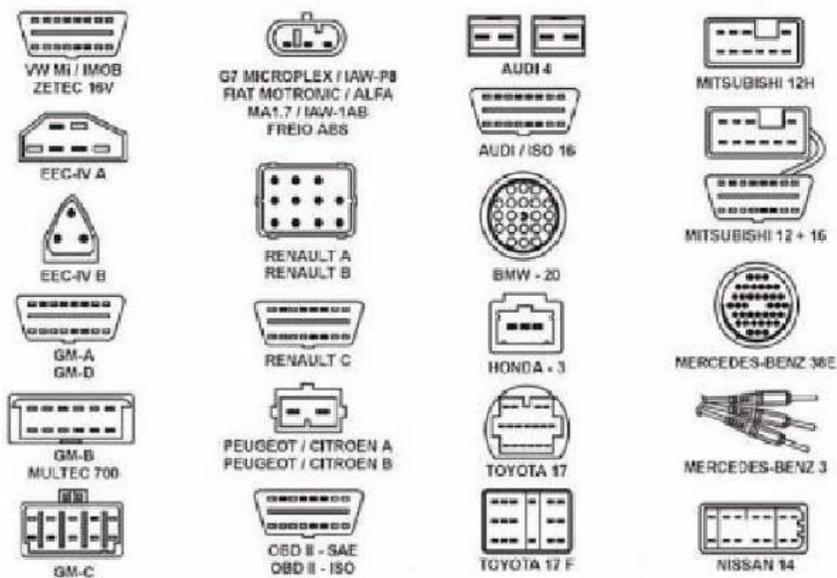


Ilustración 2. Conectores OBD I de diferentes fabricantes [Ilustración], por Montoya Jaime, 2015, Slideshare (<https://es.slideshare.net/jaimikow/conectores-y-cap-diag-a-bordo>).

- Estándar OBD II

OBD II es la segunda generación del sistema de diagnóstico a bordo, sucesor de OBD I. Alerta al conductor cuando el nivel de las emisiones es 1.5 mayor a las diseñadas. A diferencia del OBD I, el OBD II detecta fallos eléctricos, químicos y mecánicos que pueden afectar al nivel de emisiones del vehículo.



Ilustración 3. Año de implementación del sistema OBD II en diferentes países [Ilustración], por Solo electrónicos, 2018, Solo electrónicos (<https://soloelectronicos.com/2018/11/30/usos-y-fundamentos-del-odb2/>).

El sistema verifica el estado de todos los sensores involucrados en las emisiones. Cuando algo falla, el sistema se encarga automáticamente de informar al conductor encendiendo una luz indicadora de fallo (Malfunction Indication Lamp (MIL), también conocida como Check Engine o Service Engine Soon).

Para ofrecer la máxima información posible para la reparación o diagnóstico del vehículo, el sistema OBD II guarda un registro del fallo y las condiciones en que ocurrió. Cada fallo tiene un código asignado que se puede leer con un dispositivo que envía comandos al sistema OBD II llamados PID (Parameter ID).

Generalmente el conector OBD II suele encontrarse en la zona de los pies del conductor, consola central o debajo del asiento del copiloto.



Ilustración 4. Parámetros que abarca el sistema OBD II [Ilustración], por Alfaro Daniel, 2017, gob (https://www.gob.mx/cms/uploads/attachment/file/187221/diagnosticoabordo_1_260117.pdf).

Actualmente se puede conectar con el escáner automotriz de diferentes maneras, mediante Bluetooth, WIFI, USB. Este enlace, unido a un software ejecutándose desde un ordenador o un terminal móvil permite la monitorización en tiempo real de códigos de error y diversos parámetros directamente de la centralita del motor tales como las revoluciones del motor, el consumo de combustible en tiempo real (sin que el automóvil lleve equipado ordenador de a bordo) o la temperatura del refrigerante, entre muchos otros parámetros dependiendo del modelo. [21]

- Estándar EOBD

EOBD es la abreviatura de European On Board Diagnostics (Diagnóstico a Bordo Europeo), es una variación del sistema OBD II que se implementó en Europa a partir del año 2000.

Una de las diferencias es que no se monitorizan las evaporaciones del depósito de combustible. Sin embargo, EOBD es un sistema mucho más sofisticado que OBD II ya que usa "mapas" de las entradas a los sensores basados en las condiciones de operación del motor, y los componentes se adaptan al sistema calibrándose empíricamente. Esto significa que los repuestos necesitan ser de alta calidad y específicos para el vehículo y modelo. [21]

- Estándar JOBD

JOBD es la abreviatura de Japanese On Board Diagnostics (Diagnostico a Bordo Japonés), es una variación del sistema OBD II para los vehículos vendidos en Japón.

No existe mucha información ya que se encuentra protegido por la regularización del país (Japón) sin embargo, existen algunas filtraciones sobre este protocolo, las cuales indican que este es uno de los más avanzados en cuanto a regularizaciones del medio ambiente, gases contaminantes, entre otros.

1.1.1.3. Normas OBD

- ***Norma SAE J1979***

Esta norma especifica los servicios de diagnósticos básicos en un automóvil, así como, el formato de las peticiones y respuestas que deben darse para que con el apoyo de un escáner automotriz se puedan obtener los datos relacionados a la emisión de gases de los vehículos con motores de combustión interna. Estos mensajes están destinados a ser utilizados por cualquier escaner que cumpla con las especificaciones dadas por la norma SAE J1978 para la recuperación de información OBD de un vehículo. [8]

- ***Norma ISO 15031-5***

ISO 15031-5 se basa en la norma SAE J1979 y fue desarrollada originalmente para combinar las exigencias de los EE.UU con la regulación OBD Europea en el año 2000.

1.1.1.4. Conector OBD II

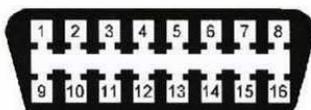
Es un componente que permite el diagnóstico del vehículo mediante la comunicación con el escáner automotriz. Para llevar a cabo esta comunicación cada vehículo implementa un protocolo de comunicación comercial o implementado por el fabricante.

El conector del sistema OBDII tiene que cumplir una serie de especificaciones según la normativa ISO 15031-3:2016. La normativa estipula que el conector para diagnóstico de OBDII o EOBD, debe de estar situado en el habitáculo, cerca del asiento del conductor. Esto es lo contrario a los sistemas anteriores donde el conector estaba en el compartimento motor.

Al comienzo de la etapa automotriz cada fabricante usaba su propio sistema de diagnóstico a bordo (OBD). En dicha etapa cada fabricante estableció su protocolo de comunicación y un conector único para el sistema de diagnóstico, por lo tanto, esto hizo que los técnicos y mecánicos tuvieran que adquirir diferentes equipos que cubrieran los diferentes protocolos y contar con los conectores para dichas marcas.

Debido a este problema, la EPA (Environmental Protection Agency) estableció una norma que dicta de que todos los vehículos que fueran vendidos en USA a partir de 1996 deberían contar con un conector trapezoidal de 16 pines para el sistema de diagnóstico a bordo conocido hoy en día como OBD II y que la mayoría de los vehículos actuales disponen. [17]

Terminales del Conector OBDII



1 – Sin uso	9 – Sin uso
2 - J1850 Bus positivo	10 - J1850 Bus negativo
3 – Sin uso	11 – Sin uso
4 - Tierra del Vehículo	12 – Sin uso
5 – Tierra de la Señal	13 – Tierra de la señal
6 - CAN High	14 - CAN Low
7 - ISO 9141-2 - Línea K	15 - ISO 9141-2 - Línea L
8 – Sin uso	16 - Batería - positivo

Ilustración 5. Distribución de los pines del conector OBD II [Ilustración], por Alfaro Daniel, 2017, gob (https://www.gob.mx/cms/uploads/attachment/file/187221/diagnosticoabordo_1_260117.pdf).

1.1.1.5. ECU

La Unidad de Control de Motor o ECU (Engine Control Unit) es una unidad de control electrónico que controla varios aspectos de la operación de combustión interna del motor.

Las ECUs más simples sólo controlan la cantidad de combustible que es inyectado en cada cilindro en cada ciclo de motor. Las ECUs más avanzadas controlan el punto de ignición, el tiempo de apertura/cierre de las válvulas, el nivel de impulso mantenido por el turbo-compresor (en vehículos con turbo-compresor), y control de otros periféricos.

Las ECUs determinan la cantidad de combustible, el punto de ignición y otros parámetros monitorizando el motor a través de sensores. Estos incluyen: sensor MAP, sensor de posición del acelerador, sensor de temperatura del aire, sensor de oxígeno, ente otros. [18]

1.1.2. Protocolos de comunicación OBD II

1.1.2.1. SAE J1850 (PWM y VPW)

Los protocolos J1850 PWM y VPW se pueden describir simultáneamente ya que en términos generales ambos son lo mismo, con la diferencia fundamental que el PWM (Pulse Width Modulation) se utiliza para velocidades de transmisión alta y media, y el VPW (Variable Pulse Width) se usa para las velocidades de transmisión bajas.

Si se habla del protocolo J1850 PWM, estamos refiriéndonos de velocidad más rápida que alcanza hasta 41,6 kbit/s, y además su transmisión la realiza en modo diferencial mediante dos cables. Por otro lado, cuando se trata del protocolo J1850 VPW, estamos ante la versión de velocidad más lenta, debido a que este tipo de protocolo trabaja a velocidades de 10,4 kbit/s.

La principal aportación de este protocolo fue la inclusión de las respuestas de los nodos destinatarios dentro de la propia trama emitida desde el nodo origen. En concreto permite: respuesta de un byte desde un simple destinatario, respuestas concatenadas de un byte desde múltiples destinatarios y respuesta de múltiples bytes desde un simple destinatario. [12]

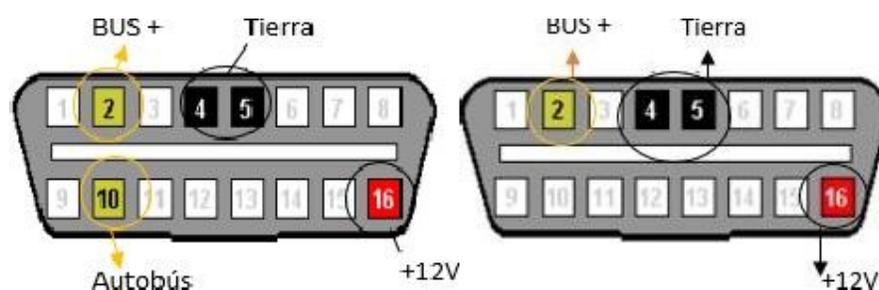


Ilustración 6. Distribución de pines de los protocolos PWM y VPW, respectivamente [Ilustración], por Global Tech LA, 2017, Global Tech LA (https://www.globaltechla.com/eBusiness/fichas_tecnicas/TEORIA-CONECTIVIDAD-OBD.pdf)

1.1.2.2. ISO 9141-2

El protocolo ISO 9141 (y su posterior revisión 9141-2) es muy parecido al protocolo RS-232 ya que es orientado a byte. Los mensajes de diagnóstico se forman en la capa de aplicación (SAE J1979) y son enviados byte a byte por la capa de enlace. En ISO 9141 sólo existen dos símbolos: el “0” y el “1” y la codificación es NRZ (Not Return Zero), por lo que el voltaje no vuelve a cero entre dos valores consecutivos a uno. El “1” lógico se representa con un nivel igual al voltaje positivo de la batería y el “0” lógico con el nivel de tierra. [17]

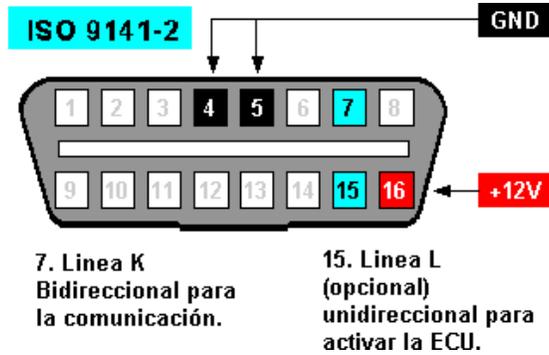


Ilustración 7. Distribución de pines de los pines del protocolo ISO 9141-2 [Ilustración], por Global Tech LA, 2017, Global Tech LA (https://www.globaltechla.com/eBusiness/fichas_tecnicas/TEORIA-CONECTIVIDAD-OBD.pdf)

1.1.2.3. ISO 14230-4 /KWP 2000

El estándar ISO 14230, más conocido como KWP2000 (Keyword Protocol 2000), es un protocolo de comunicación serie muy similar al ISO 9141. Al igual que el protocolo ISO 9141 basado en un bus bidireccional sobre una única línea (K-line), opcionalmente puede haber una segunda línea (L-line) para las señales de llamada o wakeup. La tasa de datos puede oscilar entre los 1,2 kbit/s y los 10,4 kbit/s.

La codificación en KWP2000 es NRZ (Not Return Zero) y la velocidad de 10.4 kbit/s, al igual que en ISO9141-2. El envío se hace byte a byte, pero en KWP2000 existe una estructura ya predeterminada para los datos, a diferencia de ISO 9141-2 que se adapta al protocolo de aplicación. [17]

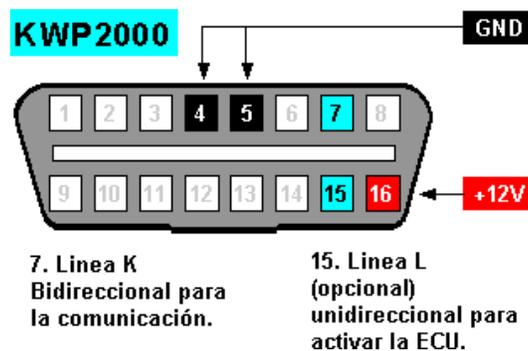


Ilustración 8. Distribución de pines de los pines del protocolo KWP 2000 [Ilustración], por Global Tech LA, 2017, Global Tech LA (https://www.globaltechla.com/eBusiness/fichas_tecnicas/TEORIA-CONECTIVIDAD-OBD.pdf)

1.1.2.4. ISO 15765-4 CAN

CAN (Controller Area Network) es un bus de comunicación serial con una velocidad de comunicación de hasta 1 Mbit/s y tiene una excelente capacidad de detección y aislamiento de errores, está diseñado para aplicaciones de control en tiempo real y fue desarrollado por Bosch en 1985 para redes en vehículos. [19]

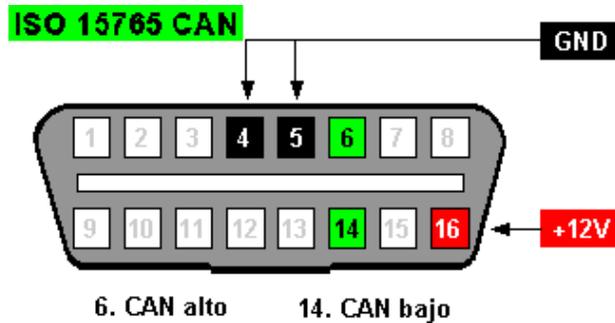


Ilustración 9. Distribución de pines de los pines del protocolo ISO 15765 CAN [Ilustración], por Global Tech LA, 2017, Global Tech LA (https://www.globaltechla.com/eBusiness/fichas_tecnicas/TEORIA-CONECTIVIDAD-OBd.pdf)

Anteriormente la industria automotriz conectaba las unidades de control a los diferentes sensores de forma directa, pero conforme se fueron agregando cada vez más, fue creciendo la necesidad de disminuir el cableado ya que aumentaba tanto el costo de producción como el peso del vehículo.

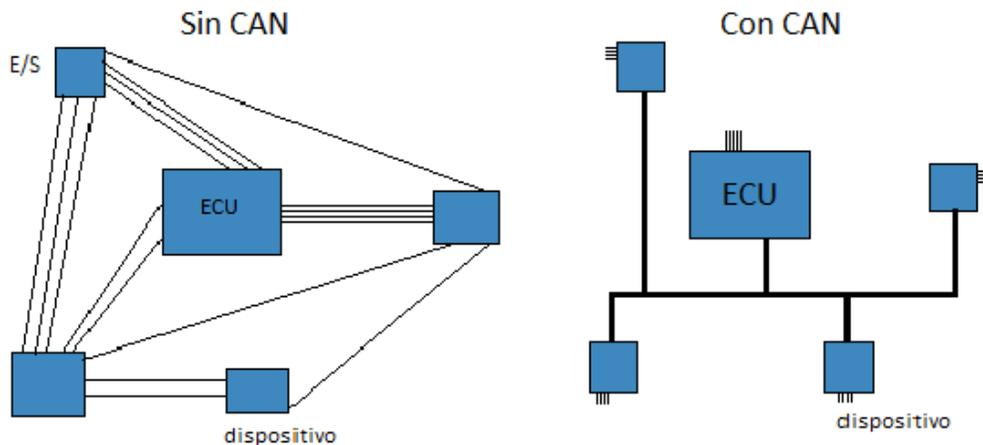


Ilustración 10. Comparación del cableado de las unidades de control sin CAN y con CAN, respectivamente [Ilustración], por Loctite Teroson, 2018, blog reparación vehículos (<https://blog.reparacion-vehiculos.es/can-bus-ventajas-fallas-en-el-sistema>)

El protocolo posee una topología en forma de bus serial, utilizando solamente 2 cables trenzados con una impedancia de 120 Ohms, para conectar todos los dispositivos en una misma red. Hay 2 señales en estos cables que se conocen como CAN high (2.5V-3.5) y CAN low (1.5V-2.5V) haciendo que el bus se encuentre en modo recesivo que es cuando ambos cables se encuentran a la misma tensión y en modo dominante cuando la diferencia de tensión entre ambos es de al menos 1.5V.

Esto tiene como objetivo proporcionar una mayor protección frente a interferencias electromagnéticas

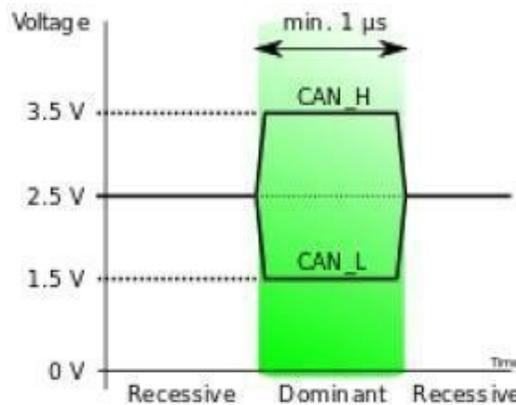


Ilustración 11. Niveles de tensión del bus CAN [Ilustración] por Wikipedia, 2018, Wikipedia (https://es.wikipedia.org/wiki/Bus_CAN)

Este protocolo está orientado a un modelo de funcionamiento de tipo productor/consumidor, si bien pueden emplearse otros como el tipo maestro/esclavo múltiple. En el primer modo, un nodo puede enviar un mensaje al bus cuando unilateralmente lo decida. El mensaje es finalmente transmitido si gana la contienda entre los nodos candidatos a transmitir en el instante de acceso. Esta contienda se realiza en base al valor del campo identificador de la trama y requiere el uso de dos niveles de señalización asimétricos de bus, denominados nivel recesivo (repose) y nivel dominante. El campo identificador (11 bits) de la trama es funcional indicando el contenido del campo de datos de la misma. Una característica importante de este protocolo es el reconocimiento de recepción (bit ACK) incluido en la propia trama (in frame response) útil cuando existe un único destinatario de la trama. También permite, incluida en la propia trama de envío, la petición de una respuesta de la estación destino (mediante el bit RTR). Finalmente, merece una cierta consideración el confinamiento local de errores en un nodo ante reiteradas detecciones de error, es decir, cuando un nodo detecta 255 veces error en su transmisión, se desconecta del bus.

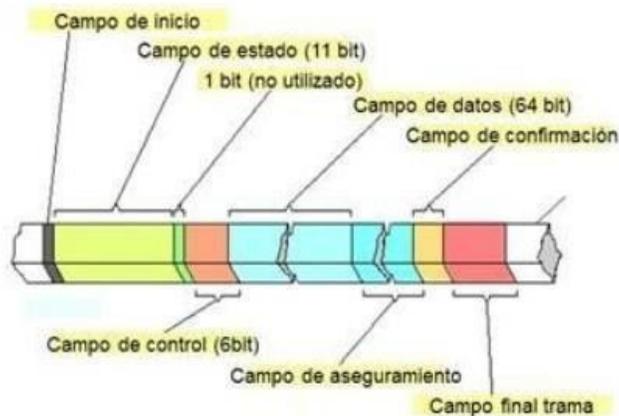


Ilustración 12. Trama de datos del bus CAN [Ilustración], por Barbadillo Francisco, 2017, Tecnomovil (<http://www.fbelectronica.com/Formacion-1/Curso-CAN-BUS.htm>)

1.1.3. OBD II – PID

OBD-II PID, (On Board Diagnostics, OBD, Parameter ID, PID), son códigos para comunicarse con el sistema de diagnóstico a bordo de un vehículo.

1.1.3.1. Modos de operación

Los modos de operación de diagnóstico OBD II han sido creados de forma que sean comunes a todos los vehículos de distintos fabricantes. Esto quiere decir que no importa qué vehículo se está probando o qué equipo de diagnóstico OBD II se esté utilizando, todas las pruebas que necesita llevar a cabo serán hechas de la misma manera para cualquier vehículo de cualquier fabricante. [10]

El estándar OBD II SAE J1979 define diez modos de operación:

Tabla 1. Modos de operación de diagnóstico del estándar OBD II [Tabla], por Wikipedia, 2019, Wikipedia (https://en.wikipedia.org/wiki/OBD-II_PIDs)

Modo (Hex)	Descripción
01	Identificación de parámetro (PID), es el acceso a datos en tiempo real de valores analógicos o digitales de salidas y entradas a la ECU(Unidad de Control Electrónico).
02	Acceso a cuadro de datos Congelados. Esta es una función muy útil del OBD II porque la ECU toma una muestra de todos los valores relacionados con las emisiones en el momento exacto de ocurrir un fallo.

03	Muestra los códigos de fallas de diagnóstico (Diagnostic Trouble Codes, DTC)
04	Borra los datos almacenados, incluyendo códigos de fallas (DTC)
05	Resultados de la prueba de monitoreo de sensores de oxígeno (solo aplica a vehículos sin comunicación Controller Area Network, CAN)
06	Resultados de la prueba de monitoreo de componentes/sistema (resultados de la prueba de monitoreo de sensores de oxígeno en vehículos con comunicación CAN)
07	Muestra los códigos de fallas (DTC) detectados durante el último ciclo de manejo o el actual
08	Este modo permite realizar la prueba de actuadores. Con esta función se pueden activar y desactivar los actuadores
09	Solicitud de información del vehículo
0A	Códigos de fallas (DTC) permanentes (borrados)

Para el desarrollo del presente proyecto, el modo de operación a utilizar es el modo 01, puesto que dicho modo nos permite el acceso en tiempo real de los valores analógicos o digitales de salidas y entradas a la ECU. Brindándonos parámetros tales como, velocidad, rpm, temperatura del refrigerante, etc.

1.1.3.2. Parámetros de identificación (PID)

La Sociedad de Ingenieros Automotrices (SAE) definió códigos PIDs para cada modo de operación bajo el estándar J1979. El fabricante del vehículo no está obligado a implementar todos los modos de operación o códigos, y tiene la libertad de añadir sus propios.

Los parámetros de identificación permiten el acceso a la información del módulo de control del tren motriz (PCM) del automóvil. Dicha información incluye entradas y salidas tanto digitales como analógicas para obtener valores calculados y estados del sistema. [20]

Existen dos tipos de normas que disponen los PIDs, la primera es una lista genérica OBD II bajo el estándar SAE J1979, en donde se muestran todos los PID a los cuales debe ser capaz de acceder el escáner automotriz de cualquier fabricante.

La segunda lista bajo el estándar ISO 1531-5, es una actualización de la primera para autos modernos y busca reducir los niveles de emisiones contaminantes.

A continuación, se muestra una tabla de algunos PIDs de las listas anteriormente mencionadas:

Tabla 2. Algunos PIDs del modo de operación 01 [Tabla], por Wikipedia, 2019, Wikipedia (https://en.wikipedia.org/wiki/OBD-II_PIDs)

PID (HEX)	Descripción	Valor mínimo	Valor máximo	Unidad
04	Carga calculada del motor	0	100	%
05	Temperatura del líquido de enfriamiento del motor	-40	215	°C
0A	Presión del combustible	0	765	KPa
0B	Presión absoluta del colector de admisión	0	255	KPa
0C	RPM del motor	0	16 383.75	rpm
0D	Velocidad del vehículo	0	255	km/h
0E	Avance del tiempo	-64	63.5	°
0F	Temperatura del aire del colector de admisión	-40	215	°C
10	Velocidad del flujo del aire MAF	0	655.35	gr/sec
11	Posición del acelerador	0	100	%
1F	Tiempo desde que se puso en marcha el motor	0	65 535	seg
2C	EGR comando	0	100	%
2D	Falla EGR	-100	99.2	%
2E	Purga evaporativa comandada	0	100	%
2F	Nivel de entrada del tanque de combustible	0	100	%

La lista completa de los PIDs se encuentra en el Anexo A

1.1.4. Bluetooth

La tecnología inalámbrica Bluetooth es una tecnología de ondas de radiofrecuencia de corto alcance (2.4 GHz) cuyo objetivo es el simplificar las comunicaciones entre dispositivos informáticos, como celulares, ordenadores, escáneres teclados, audífonos, etc. [9]

Algunos objetivos principales que se busca con la tecnología Bluetooth, son:

- Facilitar las comunicaciones entre equipos móviles.
- Eliminar los cables y conectores entre estos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre equipos personales.

1.1.4.1. Protocolos

Uno de los principales objetivos de la tecnología Bluetooth es conseguir que aplicaciones de diferentes fabricantes mantengan una comunicación fluida. Para conseguirlo, receptor y transmisor deben ejecutarse sobre la misma pila de protocolos.

La pila está constituida por dos clases de protocolos. Una primera clase llamada de protocolos específicos que implementa los protocolos propios de Bluetooth. Y una segunda clase formada por el conjunto de protocolos adoptados de otras especificaciones. Esta división en clases en el diseño de la pila de protocolos de Bluetooth permite aprovechar un conjunto muy amplio de ventajas de ambas. Por un lado, al implementar protocolos específicos de Bluetooth permite utilizar los beneficios que aporta la adopción de la tecnología Bluetooth. Por otro lado, la utilización de protocolos no específicos ofrece la ventaja de la interacción de esta tecnología con protocolos comerciales ya existentes. Así como la posibilidad de que Bluetooth este abierto a implementaciones libres o nuevos protocolos de aplicación de uso común. [5]

La pila de protocolos se puede dividir en cuatro capas lógicas:

- Núcleo de Bluetooth: Radio, Banda Base, LMP, L2CAP, SDP
- Sustitución de cable: RFCOMM
- Protocolos adoptados: PPP, UDP, TCP, IP, OBEX, WAP, IRMC, WAE
- Control de telefonía: TCS-binary, AT-Commands



Ilustración 13. Pila de protocolos Bluetooth [Ilustración], por Electrónica Fácil, 2018, Electrónica Fácil (<https://www.electronicafacil.net/tutoriales/Protocolos-Bluetooth.php>)

- RFCOMM

RFCOMM es la abreviatura del término inglés Radio Frequency Communication (Comunicación por radio frecuencia). El protocolo RFCOMM es un conjunto simple de protocolos de transporte, construido sobre el protocolo L2CAP; y que proporciona sesenta conexiones simultáneas para dispositivos Bluetooth emulando puertos serie RS-232. El protocolo está basado en el estándar ETSI TS 07.10.

RFCOMM es a menudo denominado emulación de puertos serie. El puerto serie de Bluetooth está basado en este protocolo.

1.1.5. ELM327

La mayoría de los automóviles producidos hoy en día están obligados, por ley, a disponer de una interfaz para la conexión de equipos de prueba de diagnóstico. La transferencia de datos en estas interfaces sigue varios estándares, pero ninguno de ellos es directamente utilizable por PCs o dispositivos inteligentes. El circuito integrado ELM327, basado en un PIC de Microchip, está diseñado para actuar como puente entre los puertos de diagnóstico a bordo (OBD) y un puerto serie RS232.

Además de ser capaz de detectar e interpretar automáticamente nueve protocolos OBD. El ELM327 también proporciona soporte para comunicaciones de alta velocidad, modo de reposo de baja potencia y el estándar de camiones y autobuses J1939.

1.1.5.1. Descripción del dispositivo

El dispositivo ELM327 es una herramienta para el diagnóstico 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 (Diagnostic Link Connector - DLC).

Algunas características del dispositivo ELM327, son:

- Brinda información de los sensores en tiempo real
- Lectura y borrado de fallas en el vehículo (DTC)
- Conexión directa a interface OBD II de 16 pines (DLC)
- Configurable con comandos AT
- Soporta los siguientes protocolos de comunicación OBD II:

Tabla 3. Comparativa entre los diferentes circuitos integrados ELM y los protocolos de comunicación soportados [Tabla], por ELM Electronics, 2017, ELM Electronics (<https://www.elmelectronics.com/products/ics/obd/>).

	ELM320	ELM322	ELM323	ELM325	ELM327	ELM327L	ELM328	ELM329	ELM329L
SAE J1850-PWM	✓				✓	✓			
SAE J1850-VPW		✓			✓	✓			
ISO 9141-2			✓		✓	✓			
ISO 14230-4 (slow)			✓		✓	✓			
ISO 14230-4 (fast)			✓		✓	✓			
ISO 15765-4 (CAN)					✓	✓		✓	✓
SAE J2411 (SWCAN)					✓	✓		✓	✓
KW1281 (SAE J2818)									
SAE J1939 (250kbps)					✓	✓		✓	✓
SAE J1939 (500kbps)					✓	✓		✓	✓
SAE J1708 (J1587)				✓					
SAE J1708 (J1922)				✓					

El dispositivo ELM327 se puede encontrar en 3 diferentes modelos, el USB, el Bluetooth y el WIFI.

- ELM327 USB

Este modelo es utilizado cuando se prefiere la transferencia de datos por cable, puesto que es más segura y generalmente no presenta fallas de comunicación entre el dispositivo ELM327 y el equipo receptor, el cual tiene que contar con el conector USB para la comunicación.

El equipo receptor, comúnmente una computadora o laptop, debe tener un programa compatible para ver los códigos de error del vehículo y todos los sensores soportados en dicho protocolo.



Ilustración 14. Modelo físico del escáner ELM327 USB [Ilustración], por GISSOLJ, 2019, GISSOLJ (https://www.gissolj.com/index.php?main_page=product_info&products_id=566983)

- ELM327 Bluetooth

Este modelo es comúnmente el más utilizado, ya que funciona con la mayoría de los equipos electrónicos que están en el mercado, ya sea en una PC, una Laptop, un dispositivo Android, entre otros. Además de que la instalación es muy sencilla y por ser un dispositivo Bluetooth puede transferir los datos de los sensores y los errores del vehículo a una distancia incluso de 10m del equipo receptor con el programa que se utilizará para ver la información. Lo cual se hace más práctico ya que no utiliza cable de conexión.



Ilustración 15. Modelo físico del escáner ELM327 Bluetooth [Ilustración], por Aliexpress, 2018, Aliexpress (https://www.gissolj.com/index.php?main_page=product_info&products_id=566983)

- ELM327 WIFI

Este modelo es comúnmente utilizado por dispositivos iPhone, aunque de igual manera puede trabajar con todos los equipos electrónicos que soporten la conexión WIFI, y por tal motivo es más sencilla la conexión de los equipos. A su vez, tiene mayor alcance tanto en distancia como en velocidad al momento de la transferencia de datos entre el dispositivo EML327 WIFI y el equipo receptor, que también tendrá instalado un programa compatible para poder ver los códigos de error del vehículo y todos sus sensores. [14]



Ilustración 16. Modelo físico del escáner ELM327 WIFI [Ilustración], por Aliexpress, 2018, Aliexpress (<https://es.aliexpress.com/item/32815205923.html>)

1.1.5.2. Características físicas y eléctricas del microcontrolador ELM327

A continuación, se muestra una tabla que describe las características tanto físicas como eléctricas del microcontrolador ELM327, para un uso eficiente y correcto del mismo.

Tabla 4. Características físicas y eléctricas del microcontrolador [Tabla], por ELM Electronics, 2017, ELM Electronics (<https://www.elmelectronics.com/products/ics/obd/>).

Características	Mínimo	Promedio	Máximo	Unidad
Voltaje de alimentación	4.2	5.0	5.5	V
Corriente promedio nominal		12		mA
Corriente promedio mínima		0.15		mA
Temperatura de almacenamiento	-65		+150	°C
Temperatura ambiente con alimentación	-40		+85	°C
Bytes de búfer de transmisión RS232		512		Bytes
Comandos AT		128		Comandos

Finalmente, algunas características adicionales que indica la hoja de datos del microcontrolador son las siguientes:

- Control de potencia con modo de espera
- Interfaz serial universal (RS232)
- Búsqueda automática de protocolos de comunicación OBD II
- Completamente configurable con comandos AT
- Diseño CMOS de baja potencia

Connection Diagram
PDIP and SOIC
(top view)

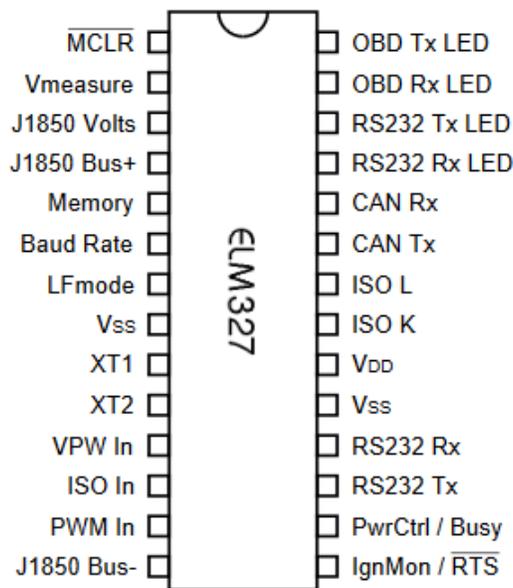


Ilustración 17. Vista superior de los pines del microcontrolador ELM327 [Ilustración], por Caiztoa Maria, 2017, Repositorio de Educación (<http://repositorio.espe.edu.ec/xmlui/bitstream/handle/21000/8209/AC-EAC-ESPE-047842.pdf?sequence=1&isAllowed=y>)

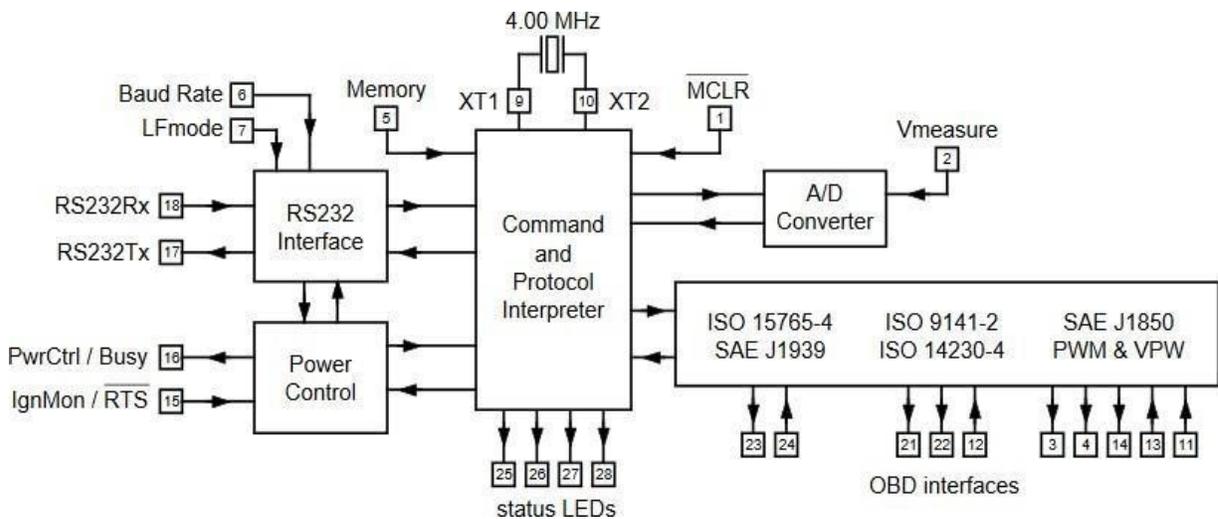


Diagrama 1. Diagrama de bloques interno del microcontrolador ELM327 [Diagrama 1], por MOTORDATA OBD, 2018, MOTORDATA OBD (<https://motordata-obd.com/adaptadores-de-agnostico-elm327/>)

1.1.5.3. *Identificación de comandos AT para la comunicación*

De acuerdo con la hoja de datos del microcontrolador ELM327, existen diversos comandos AT para configurar el microcontrolador, y de esta manera entablar la comunicación entre el microcontrolador y el puerto OBD II del vehículo.

A continuación, se enlistan los comandos AT utilizados para entablar la comunicación entre el ELM327, el puerto OBD del vehículo y la Raspberry:

- ATZ

Este comando hace que el microcontrolador realice un reset completo como si la alimentación se apagara y volviera a encenderse. Todos los ajustes vuelven a sus valores por defecto, y el microcontrolador se pone en estado de reposo, a la espera de caracteres en el bus RS232.

- ATL1

Esta opción controla el envío de caracteres de salto de línea después de cada carácter de retorno de carro. Para AT L1, los saltos de línea se generarán después de cada carácter de giro de carro, y para AT L0, estarán desactivados.

- ATH1

Estos comandos controlan si se muestran o no los bytes de información adicionales (cabeceras) en las respuestas del vehículo. Al encender las cabeceras (con AT H1) se muestra algo más que los bytes de cabecera: verá el mensaje completo tal y como se transmite, incluyendo los dígitos de comprobación.

- ATSPH (h = protocolo del vehículo a utilizar)

Este comando se utiliza para configurar la operación del ELM327 utilizando el protocolo especificado por 'h' y también para guardarlo como el nuevo valor por defecto.

El ELM327 soporta 9 protocolos diferentes, los cuales son:

- 0 - Automático
- 1 - SAE J1850 PWM (41.6 kbaud)

- 2 - SAE J1850 VPW (10.4 kbaud)
- 3 - ISO 9141-2 (5 baud init, 10.4 kbaud)
- 4 - ISO 14230-4 KWP (5 baud init, 10.4 kbaud)
- 5 - ISO 14230-4 KWP (fast init, 10.4 kbaud)
- 6 - ISO 15765-4 CAN (11 bit ID, 500 kbaud)
- 7 - ISO 15765-4 CAN (29 bit ID, 500 kbaud)
- 8 - ISO 15765-4 CAN (11 bit ID, 250 kbaud)
- 9 - ISO 15765-4 CAN (29 bit ID, 250 kbaud)

Si no se sabe con exactitud el protocolo de comunicación que se encuentra en el vehículo, la opción más adecuada es la detección automática.

1.1.6. Despliegue de información

1.1.6.1. Head-Up display

- ¿Qué es?

El Head-Up Display es un dispositivo que proyecta información en el campo visual del conductor de un vehículo, ya sea en el parabrisas o en una pequeña pantalla traslúcida sobre el tablero, en el lugar donde los ojos del conductor deberían estar mirando todo el tiempo.

Como su nombre lo indica en Head-Up Display, es un dispositivo que muestra información de tal forma que la cabeza del conductor permanezca levantada, de ahí el Head-Up.

Sus orígenes datan de la Segunda Guerra Mundial, cuando era utilizado para el desarrollo de aplicaciones militares con el fin de aligerar la carga de trabajo de los pilotos en los aviones de combate.

Estos dispositivos eran fabricados de manera rudimentaria y los equipos con el alcance tecnológico de la época contaban con una pantalla entre el piloto y el parabrisas de la aeronave, donde se podía observar una mira, para realizar los disparos más certeros aire-aire.



Ilustración 18. Head-Up Display montado en un avión de guerra [Ilustración], por Howard Bill, 2012, Extremetech (<https://www.extremetech.com/extreme/124847-bmws-full-color-hud-distraction-minimizer-of-distractions-or-costly-tech-gadget>)

En el caso de los vehículos, generalmente el Head-Up Display de la mayoría de los fabricantes proyecta la velocidad a la que viajas y el GPS, así como información trascendental de los sistemas de seguridad; en algunos casos más avanzados pueden proyectar hasta la canción que se está reproduciendo, así como alertas de seguridad, para prevenir posibles accidentes. [2]



Ilustración 19. Sistema Head-Up Display de BMW [Ilustración], por Motori Redazione, 2019, Avvenire (<https://www.avvenire.it/economia/pagine/controllo-vocale-e-display-sul-parabrezza-l-auto-cambia>)

- Tipos de Head-Up Display
- Head-Up Display

Es la variante más sofisticada, dado que la información se proyecta directamente sobre el

parabrisas, el cual tiene un tratamiento polarizado que permite capturar las imágenes y conseguir mostrarlas con nitidez, de manera que parecen flotar por delante del vehículo, a una distancia de 2 a 10 metros. La altura e intensidad de la proyección se puede regular. Pueden ofrecer hasta miles de colores, aunque los fabricantes suelen simplificar la iconografía al máximo.

Es también la opción más cara, lo que le agrega al vehículo un valor aproximado de 1300 € (\$ 28210.00 MXN) de manera que de momento sólo es frecuente encontrarla en vehículos de gama alta.

El valor del sistema se debe a 3 aspectos

- Requiere un proyector muy potente
- Requiere un parabrisas especial
- Requiere un mecanismo de proyección



Ilustración 20. Sistema Head-Up Display convencional [Ilustración], por Audi media center, 2016, AUDI (<https://www.audi-mediacycenter.com/en/photos/detail/head-up-display-42306>).

- Head-Medium Display

En esta variante, la información se proyecta sobre una lámina de cristal o de policarbonato transparente abatible que se eleva sobre el tablero.

Al igual que en el Head-Up Display, la lámina de cristal o de policarbonato debe de tener un tratamiento polarizado para mostrar las imágenes nítidamente, de no contar con dicho tratamiento, la composición laminar del parabrisas haría que la información se viera desenfocada al reflejarse en cada una de las capas que la forman.

Es una alternativa más accesible en comparación con el Head-Up Display, ya que este sistema solo le agrega al vehículo un valor aproximado de 450 € (\$ 9758.00 MXN).

Debido a que el espacio de proyección es bastante más pequeño, la cantidad de información mostrada es inferior a la de un Head-Up Display. Normalmente se limita a la velocidad, el ajuste del control de crucero e indicaciones simplificadas del navegador. [13]



Ilustración 21. Sistema Head-Medium Display convencional [Ilustración], por Computer Hoy, 2017, Computer Hoy (<https://computerhoy.com/reportajes/motor/como-funciona-head-display-coches-374329>).

- Modo de proyección

Hay dos modos de proyección, TFT y láser, dos sistemas que están presentes en el mercado a través de distintas marcas de coches. Por ejemplo, el sistema TFT utiliza diodos LED y un chip TFT, que bloquea la luz para crear los números y los datos y proyectarlos en el parabrisas. Mercedes, Audi y BMW son alguna de las marcas que utilizan este sistema.

El sistema por láser es más moderno y consta de un haz de luz que «dibuja» cada línea en tiempo real y crea los datos de manera continua. Este sistema necesita menos energía y utiliza un proyector de menor potencia, ya que el láser requiere menos potencia para emitir el haz de luz. Jaguar, Land Rover y Volvo son los fabricantes que utilizan este sistema para sus modelos.

En ambos modos, un sensor permite adaptar la intensidad de la proyección a la luminosidad ambiental de forma continua. [1]

- ¿Cómo funciona el Head-Up Display?

El Head-Up Display tiene varios componentes, que permiten crear el haz de luz, dar forma a la luz y proyectarla sobre la superficie del parabrisas o la lámina transparente.

El sistema es similar al de un retroproyector. La luz es emitida por un emisor y pasa por el colimador, el cual es un espejo con una forma especial que modifica el paralelismo de los rayos de luz, y de esta manera la imagen reflejada en el colimador parece proceder de un punto más alejado del que realmente la originó, y es más fácil de enfocar por el ojo humano.

Un segundo espejo recoge esa luz y la dirige hacia el combinador. Este espejo es móvil, de manera que en la mayoría de los Head-Up Display se puede modificar su posición, altura, distorsión y otros parámetros para ajustarlos al gusto y necesidades del conductor. Finalmente, está el combinador, que es la superficie donde se proyecta la imagen, o sea, el parabrisas o la lámina transparente. [1]

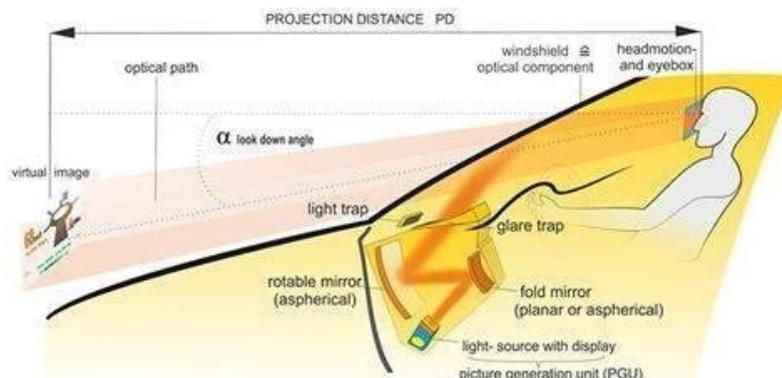


Ilustración 22. Sistema Head-Up Display desarrollado por Continental [Ilustración], por Continental, 2017, Aurburg (https://www.arburg.com/fileadmin/redaktion/sonstiges/fachvortrag_technologie-tage_2017_starke_partner-continental_handout_en.pdf)

1.1.6.2. *Campo visual del ser humano y conducción*

El área de visión perfecta se produce en un cono con un ángulo de apertura de 3° , en el que todos los objetos que se encuentran dentro de esta área se perciben con absoluta nitidez.

La visión sigue siendo clara con un ángulo de 5° o 6° . A medida que dicho ángulo de apertura se va ensanchando, la imagen de los objetos va dejándose de percibir con claridad, hasta perderse aproximadamente en el ángulo de 20° . [6]

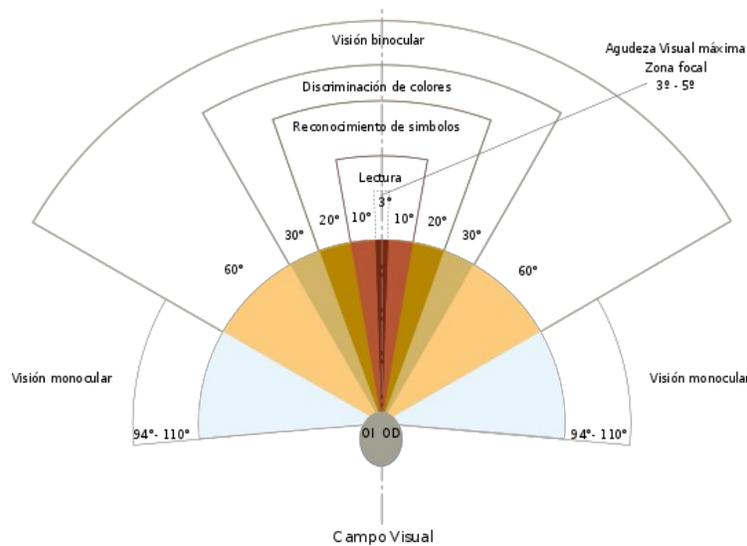


Ilustración 23. Campo visual del ser humano [Ilustración], por Wikipedia, 2016, Wikipedia (https://es.wikipedia.org/wiki/Archivo:Campo_de_vision.svg)

Cada vez que se desvía la vista para visualizar los indicadores del vehículo, se emplea de media al menos medio segundo, lo que significa que un conductor circulando a 120 km/h, recorrerá aproximadamente 33 metros cada segundo “ciego”, según el barómetro de las distracciones realizado por el RACE en colaboración con BP y Castrol.

Respecto al análisis de los factores que inciden en la seguridad, cabe destacar que, según datos de la revista Tráfico y Seguridad Vial en 2014, la velocidad inadecuada estuvo presente en el 10% de los accidentes con víctimas y que este porcentaje se eleva al 17% cuando sucede en vía interurbana. En los casos de accidentes con víctimas mortales, este factor estuvo presente en el 21% de ellos.

El primer efecto que se produce al aumentar la velocidad es el incremento en la distancia de detención, que es la suma de la distancia que recorreremos durante el tiempo de reacción y la distancia de frenado. Si a esto se le añade que el conductor posee alguna deficiencia visual indebidamente compensada o alguna patología ocular, la probabilidad de sufrir un siniestro aumenta considerablemente. Durante la conducción, las imágenes de los objetos se proyectan en la retina de forma difusa, pues se captan y suceden a gran velocidad. Ahora bien, a medida que se incrementa la velocidad, las imágenes se tornan más confusas, por lo que, a efectos prácticos, se reduce la visibilidad a lo que los expertos denominan campo de visión cinético. Y si la velocidad es muy alta, la vista del conductor sufre algo parecido al efecto túnel.

Es por esto por lo que el campo visual depende directamente de la velocidad del vehículo. Por ello, cuanto menor sea esta, mayor grado de la vía es posible abarcar con la visión.

- A 35 km/h se cubriría aproximadamente un ángulo de 104°. Gracias a esta velocidad moderada, sería fácil detectar objetos situados fuera de los márgenes de la carretera.

- A 65 km/h se reduciría la visión periférica, pudiendo observar solo un 70° de la vía, que disminuye a medida que aumenta la velocidad del vehículo.

- A 100 km/h el ángulo de visión periférica baja hasta los 42°. Esto limita la percepción a elementos muy cercanos a la calzada o en el carril anexo.

- A 130 km/h aparece el denominado “efecto túnel”, limitando la visión a solo un ángulo de 30°. Debido a este fenómeno se dejan de percibir objetos periféricos a ambos lados de la vía, como señales, semáforos, animales que puedan irrumpir en la carretera, obstáculos o peatones. A esta velocidad, el conductor puede no ser capaz de advertir una situación de riesgo que suceda en el exterior de la vía, ya que estaría fuera de su campo visual y no reaccionar adecuadamente.

- A 150 km/h se produce el ‘cataclismo perceptivo’. Este efecto consiste en la pérdida de la visión periférica e imposibilidad para evaluar correctamente distancias y velocidades. [16]

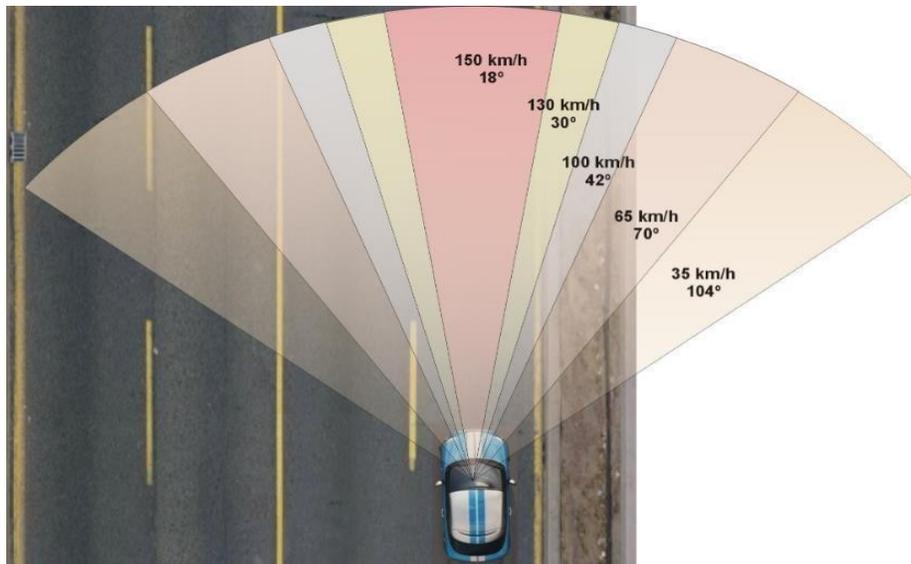


Ilustración 24. Campo visual con respecto a la velocidad de conducción [Ilustración], por Tu Optometrista, 2017, Tu Optometrista (<https://www.tuoptometrista.com/vision-y-conduccion/conduccion-y-velocidad/>)

1.2. Estado del arte

Los sistemas de proyección aplicados a la industria automotriz (mejor conocidos como Head-Up Display) se han implementado desde hace ya varios años, siendo el Oldsmobile Cutlass Supreme de General Motors en 1988 el primer vehículo en utilizar dicho sistema de proyección para proyectar la velocidad a la que se circulaba. En 1997, el vehículo se dejaría de producir y solo quedaría el Pontiac Grand Prix como el único modelo con el sistema HUD.



Ilustración 25. Sistema Head-Up Display del Oldsmobile Cutlass Supreme y del Pontiac Grand Prix, respectivamente [Ilustración], por Soler Alex, 2018, SPORT (<https://www.sport.es/es/noticias/automocion/head-up-display-que-es-como-funciona-6585510>)

Con el paso del tiempo diferentes marcas automotrices han vuelto a trabajar en la implementación de los sistemas de proyección Head-Up Display, siendo BMW la firma pionera de la implementación de estos sistemas en la actualidad. [15]

A continuación, se muestran diferentes trabajos y sistemas que muestran una similitud con el proyecto a desarrollar; ya sea una similitud en cuanto al sistema de proyección, o una similitud en cuanto a la adquisición de datos provenientes del puerto OBD II. Por tanto, para tener un orden de estos, se dividieron en dos secciones como se muestra a continuación:

- **Para el sistema de proyección**

- *Sistema de diagnóstico OBD II SAE con proyección de parámetros en el parabrisas del automóvil.*

Los vehículos actuales disponen de una unidad de control central ECU por sus siglas en inglés

“Electronic Control Unit” así como varios módulos de control encargados de gestionar los diferentes parámetros del vehículo, los cuales deben mantenerse siempre acotados para asegurar un buen funcionamiento de este.

En la actualidad los vehículos incorporan una gran cantidad de sensores que permite a la ECU determinar la condiciones máximas y mínimas de funcionamiento del vehículo y de esta manera mantener el vehículo en condiciones óptimas. En caso de que alguno de los parámetros no se encuentre dentro de los rangos marcados, el

sistema OBD II es el encargado de almacenar dicha información, y enviar una alerta al conductor mediante un indicador luminoso visualizado en el panel de instrumentos. Cabe mencionar que en el panel de instrumentos de vehículos convencionales solo se visualizan los indicadores luminosos para el consumo de combustible, temperatura del refrigerante y en ocasiones para la presión del aceite, dejando de lado otros indicadores que pudieran ser de ayuda al conductor para determinar fácilmente un problema surgido en el vehículo y la manera en cómo solucionarlo.

El presente proyecto, por lo tanto, abordó dicha problemática y el propósito esencial de este sistema fue aumentar el control del usuario con su vehículo, es decir que el conductor este enterado en todo momento de los problemas posibles que podrían solicitarse en la marcha normal del vehículo. Para esto se desarrolló un sistema de diagnóstico OBD II SAE con proyección de los parámetros como temperatura del refrigerante, velocidad, revoluciones por minutos del motor, posición del acelerador, estado de la mezcla aire-combustible y presión del múltiple de admisión, en el parabrisas del vehículo para permitir al conductor acceder a esta información sin descuidar la trayectoria de conducción.

Al final se cumplió con el objetivo general y se logró crear un sistema que tiene la capacidad de comunicarse con todos los sistemas legislados OBD II para que de esta manera se permitiera la proyección de los diferentes parámetros de los vehículos sobre sus respectivos parabrisas. [7]

- NAVDY

Navdy es un sistema Head-Medium Display, el cual mediante un conector OBD II extrae información sobre el manejo directamente de la computadora del vehículo y gracias a un

proyector, dicha información se proyecta hacia el campo visual del conductor sobre una pantalla translúcida de cristal.

Adicionalmente, Navdy te permite visualizar GPS, llamadas, mensajes y música. Además, cuenta con una cámara que sirve para detectar movimientos de las manos y de esta manera abrir mensajes o contestar llamadas.

El valor de este sistema oscila en \$ 23,500.00 MXN.



Ilustración 26. Sistema de proyección NAVDY [Ilustración], por Ortega J., 2019, Auto México (<https://automexico.com/conduccion/head-up-display-hud-mejores-aid2212>)

- HUDWAY Glass

El HUDWAY Glass, es un sistema que, mediante una aplicación, utiliza el teléfono celular para proyectar mediante reflexión información útil de la conducción, desde velocidad, hasta la ruta de navegación que se está siguiendo, sobre un panel de cristal o policarbonato transparente que se ubica justo en el campo visual de conducción.

La ventaja de estos sistemas es que son económicos, ya que no son más que un dispositivo reflectante. Algunas variantes más avanzadas también incluyen un transmisor que se conecta al puerto OBD II del vehículo para enviar al teléfono celular toda la información posible del coche.

El valor de este sistema oscila en \$ 650.00 MXN



Ilustración 27. Sistema de proyección HUDWAY Glass [Ilustración], por Ortega J., 2019, Auto México (<https://automexico.com/conduccion/head-up-display-hud-mejores-aid2212>)

- **Para el sistema OBD II**

- *Diseño de aplicación para uso de OBD II con interfaz vía Bluetooth a dispositivo móvil.*

Actualmente los teléfonos celulares se han convertido en parte importante de la vida cotidiana de millones de personas alrededor del mundo, a tal grado que el dispositivo móvil pasó de ser un instrumento que facilitaba las comunicaciones, a ser un asistente con el cual se puede interactuar como una herramienta de trabajo, como entretenimiento y multimedia. Debido a la importancia que los teléfonos celulares tienen actualmente, relacionar su uso con el ámbito automotriz, es una manera muy sofisticada y eficiente de poder entablar una comunicación. Tal es el caso del presente proyecto, en donde se creó una aplicación para dispositivos móviles que permite un mejor conocimiento de los parámetros que influyen en el comportamiento del vehículo y así evitar los mantenimientos correctivos y las altas emisiones de gases que contaminan el ambiente.

El proyecto realizado consistió básicamente en diseñar una aplicación para dispositivos móviles del tipo tableta y con sistema operativo Android en sus versiones

3.0 (HoneyComb) en adelante debido que estas versiones del S.O están diseñadas en su mayoría para dispositivos de pantallas más grandes, con el fin de enviar y recibir datos de la ECU del vehículo que cuente con el sistema OBD II, usando la comunicación entre el dispositivo móvil y el conector ELM327 en su versión con conexión Bluetooth.

El proyecto resulto un tanto complicado debido a la falta de conocimiento de diferentes áreas, tanto de la programación como del funcionamiento del sistema OBD II, sin embargo, al final la aplicación funciono y cumplió con el propósito general, además de que se adquirieron nuevos conocimientos que pudieron ser utilizados en el ámbito profesional. [11]

- Prototipo de pre verificación ambiental por medio del sistema de diagnóstico a bordo del automóvil.

Desde los años 80's la junta de Recursos del Aire de California (CARB California Air Resources Board) estipulo que era necesaria la creación de una herramienta de diagnóstico capaz de detectar un mal funcionamiento del motor con la finalidad de reducir las emisiones contaminantes de los automóviles. Las nuevas normas de verificación vehicular estipulan que de los vehículos en circulación del año 2006 y posteriores realizaran la verificación por medio del puerto OBD II, dicha tecnología es relativamente nueva en el contexto de la verificación mexicana; por lo que ha aumentado el índice de rechazados por incumplimiento de lo establecido en dicha norma. Es común que cuando conectan la interfaz de los ordenadores al puerto OBD II no puedan establecer una conexión con la ECU del automóvil siendo así rechazados. Dicho lo anterior, esto genera una incógnita en el propietario del vehículo, ya que este desconoce el funcionamiento de los monitores utilizados para la verificación del automóvil, así como el estado de los mismos, pues en aquel entonces no existía una pre verificación que le permitiera tener un parámetro del estado de su automóvil. Por lo tanto, lo que se buscó con este proyecto, fue desarrollar un sistema de pre verificación que permita conocer el estado de los puntos a cumplir en la verificación ambiental por medio del sistema de diagnóstico a bordo, indicándole al usuario si su automóvil es apto para poder acreditar la verificación, y de esta manera disminuir el índice de rechazo por incumplimiento de códigos de preparación.

Para el correcto funcionamiento del sistema se utilizó una interfaz o traductor OBD II Bluetooth para poder acceder a la información del ODB de manera remota mediante el Bluetooth del dispositivo móvil, además de que se establecieron internamente en una aplicación, una comunicación en serie entre el sistema de diagnóstico y el dispositivo móvil enviando y recibiendo información de los parámetros a desplegar, de manera simultánea. El proyecto se desarrolló al margen de los dispositivos móviles

con sistemas operativos Android ya que de acuerdo un estudio realizado por la revista FORBES, estos ocupan el 68.3 % del mercado.

Finalmente, el prototipo que se desarrolló cumplió todos los objetivos planteados desde el inicio de este, permitiendo el desarrollo de una comunicación con el sistema de diagnóstico a bordo del automóvil mediante una aplicación móvil para dispositivos Android. [3]

Los 5 proyectos y sistemas antes mencionados, nos permiten darnos cuenta de que el proyecto que deseamos abordar es viable puesto que ninguno de esos sistemas despliega sobre un cristal transparente montado en el tablero del vehículo, la velocidad, revoluciones por minuto, temperatura del refrigerante, relación de la mezcla aire-combustible y temperatura del medio ambiente al mismo tiempo con el propósito de realizar una conducción eficiente, cómoda y segura, y de esta manera prevenir averías en el vehículo.

CAPÍTULO 2.
PLANTEAMIENTO DEL
PROBLEMA

Hoy en día, debido al avance de la tecnología, cualquier industria requiere una constante innovación en los procesos y productos que realiza, y la industria automotriz no es la excepción de ello. En la industria automotriz, el crear e innovar diferentes sistemas para su implementación en el vehículo y así solucionar los problemas que demanda la sociedad actual, conlleva un trabajo de constante investigación y pruebas para tratar de solucionar dichos problemas. Desafortunadamente en ocasiones dichos sistemas solo brindan una solución para vehículos selectos y por lo tanto no se llega a cubrir del todo la problemática a abordar, dejando de lado a la mayoría de los vehículos.

Es por ello, que con el presente proyecto se pretende abordar una problemática que consideramos no se ha cubierto del todo, la cual se puede definir con la siguiente pregunta,

¿Cómo brindar al conductor la posibilidad de visualizar fácilmente los indicadores esenciales del panel de instrumentos para generar una conducción eficiente, cómoda y segura?

Los sistemas de proyección Head-Up Display solo se encuentran implementados en vehículos selectos de gama alta, dejando de lado a los vehículos de gamas inferiores sin la oportunidad de poder hacer uso de este sistema. Cabe señalar que la mayoría del parque vehicular en México al no contar con el sistema Head-Up Display, tienen una probabilidad más alta de sufrir un desperfecto o una avería al descuidar los parámetros de funcionamiento del vehículo y de esta manera no lograr una conducción eficiente y en situaciones extremas llegar a provocar un accidente.

2.1. Propuesta de solución

El presente proyecto consta de 6 bloques principales, cada uno de ellos cumple con una función específica dentro del funcionamiento general del sistema.

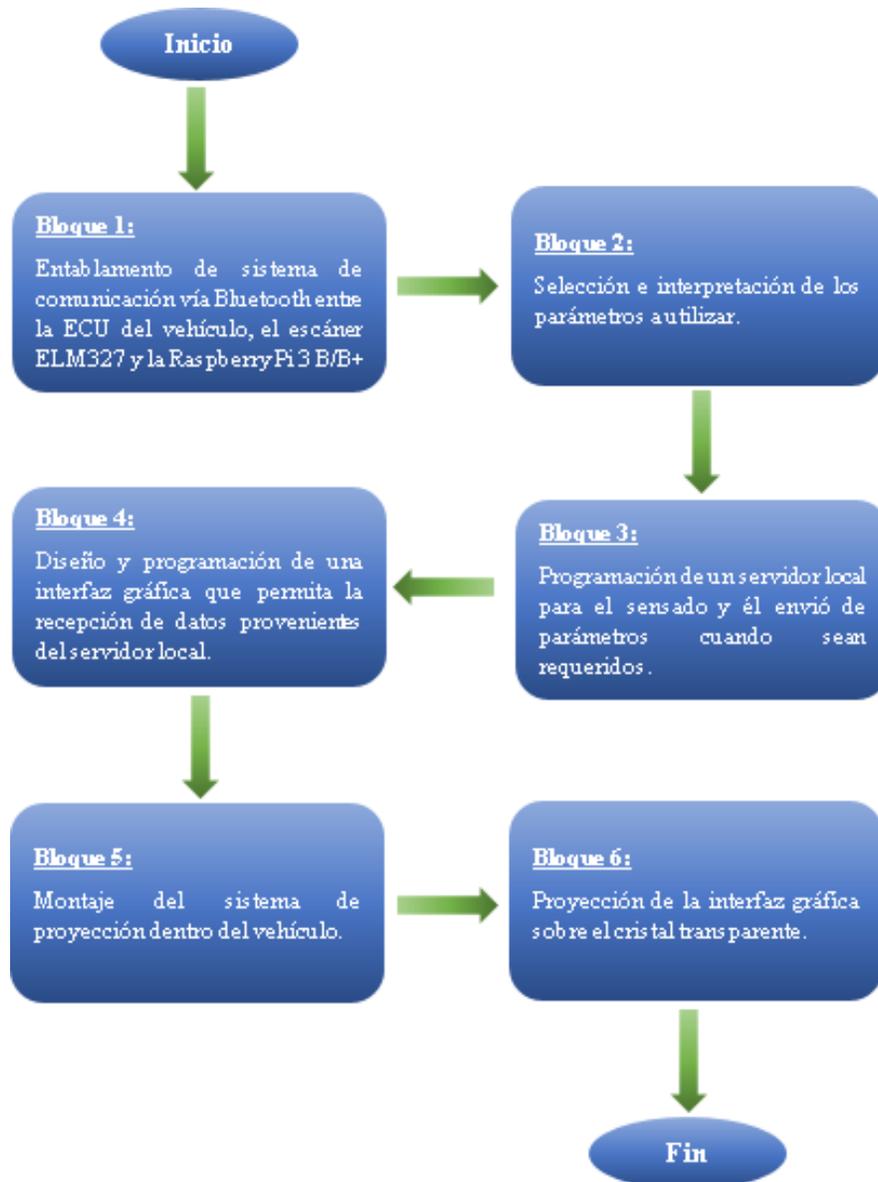


Diagrama 2. Diagrama de bloques del funcionamiento general del sistema

En el bloque 1, se entabló un sistema de comunicación vía Bluetooth entre el escáner ELM327 y la Raspberry. Esto se logra conectando el escáner ELM327 al puerto OBD II del vehículo, y enseguida en la consola de la Raspberry se emula un puerto serie RS232 para poder hacer el envío de comandos AT (ATZ, ATL1, ATH1, ATSP0), y de esta manera configurar el microcontrolador para habilitar el envío de los parámetros hacia la Raspberry.

Para el bloque 2, existe una lista de códigos PID definida por la SAE, en la cual cada código está asociado con un parámetro del vehículo, ya sea para la velocidad (0D), para las rpm (0C), para la temperatura del refrigerante (05), para la relación de la mezcla aire-combustible (14) y para la temperatura del medio ambiente (46), entre otros parámetros. Para la selección de los parámetros a utilizar en el proyecto, se tomó en cuenta dicha lista de códigos PID.

El envío de dichos códigos se realizó manualmente a través del sistema de comunicación entablado en el bloque 1, recibiendo como respuesta un valor hexadecimal variable en cada instante que se realizaba la petición de cada parámetro.

Posterior a ello mediante una librería descargada de internet y programada en Python (Python-OBD) se realizó la interpretación de los valores hexadecimales de los parámetros a utilizar. Además de que dicha librería nos permitió realizar una comunicación asíncrona para la actualización automática de los parámetros sin la necesidad de pedírselos manualmente al sistema.

En el bloque 3, se programó un servidor local en Python en el cual se estarán actualizando constantemente los parámetros seleccionados, esperando una petición de la interfaz gráfica para realizar el envío de estos.

Para el bloque 4, se realizó el diseño y la programación de una interfaz gráfica en Java que permita hacer la petición y la recepción de los parámetros provenientes del servidor local, además de realizar el despliegue distribuido de dichos parámetros.

En el bloque 5, se realizó un estudio detallado del habitáculo del vehículo para determinar el lugar más adecuado para realizar el montaje del sistema de proyección.

Una vez determinado el lugar más adecuado para la instalación, finalmente para el bloque 6, se realizó la proyección de la interfaz gráfica sobre el cristal transparente.

2.2. Alcance del proyecto

Con este sistema se pretende proyectar sobre un cristal transparente 5 parámetros esenciales del vehículo, los cuales son velocímetro, tacómetro, temperatura del refrigerante, relación de la mezcla aire-combustible y temperatura del medio ambiente para casi cualquier vehículo que cuente con sistema OBD II.

Para la instalación del proyecto, se tomaron en cuenta diversos estándares de seguridad y eficiencia para implementación del sistema, además del estudio del reglamento de tránsito actual de la Ciudad de México, y se llegó a la conclusión de que la implementación del sistema a desarrollar puede realizarse sin problema en cualquier vehículo de la Ciudad de México, siempre y cuando el sistema se ajuste para no interferir entre la visualización del camino y el conductor.

Se recomienda utilizar el presente proyecto aproximadamente después de las 5 pm, ya que es un horario en donde el sol comienza a esconderse y por tanto permite visualizar con mayor claridad los parámetros de la interfaz.

CAPÍTULO 3.

DESARROLLO DEL SISTEMA

3.1. Selección y costos de componentes

A continuación, se muestran los componentes utilizados para el desarrollo y montaje del sistema, y el costo aproximado de los mismos. Además, se muestra el costo total aproximado del sistema en general.

Tabla 5. Componentes y costo total del sistema

Componentes	Costo aproximado
Proyector Acer K11 200 lúmenes	\$ 3500.00 MXN
Tarjeta de desarrollo Raspberry PI 3 B	\$ 1100.00 MXN
Film de retroproyección (1.20m x 0.50m)	\$ 1000.00 MXN
Inversor de corriente automotriz CA-CD 400W	\$ 800.00 MXN
Conector ELM327	\$ 195.00 MXN
Gabinete de plástico de 10.2 x 4.1 x 7.7cm	\$ 78.00 MXN
Cristal de 19.5cm x 15cm x 3mm de espesor	\$ 45.00 MXN
2 metros de tubo corrugado	\$ 30.00 MXN
4 metros de cable calibre 12	\$ 28.00 MXN
Porta fusible automotriz de clavija	\$ 20.00 MXN
2 terminales ojillo 1/2	\$ 8.00 MXN
2 terminales ojillo 1/4	\$ 6.00 MXN
Fusible 30 A	\$ 3.00 MXN
TOTAL	\$ 6813.00 MXN

3.2. Diagrama de montaje del sistema

En el diagrama 3, se muestran las 3 etapas principales del sistema y los componentes que conforman cada una de estas etapas, las cuales hacen que el sistema pueda funcionar de manera correcta.

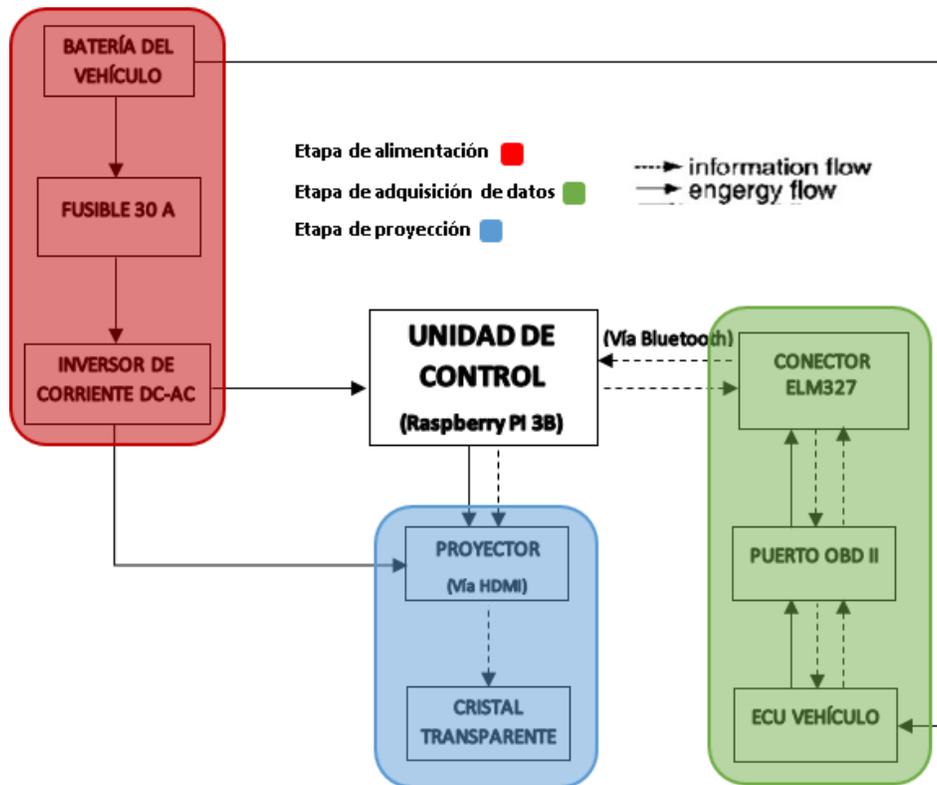


Diagrama 3. Diagrama de bloques del montaje del sistema

Como unidad de control del sistema, se tiene una Raspberry Pi 3B, la cual se eligió debido a tres factores, el primero de ellos fue la velocidad del procesador que es de 1.2 GHz, suficiente para llevar a cabo todos los procesos simultáneos del sistema, el segundo factor fue debido a su conectividad Bluetooth y el tercer factor debido a su salida HDMI.

Esta unidad de control es alimentada gracias a la etapa de alimentación, mediante un inversor de corriente automotriz DC-AC de 400W, el cual a su vez es alimentado directamente de la batería del vehículo. Además, se colocó un fusible de 30A entre la batería y el inversor, para evitar daños en el inversor de corriente debido a los picos de voltaje que genera la batería al encender el vehículo. Puesto que cuando el vehículo está apagado, el voltaje de la batería oscila entre 12V a 12.5V, y cuando el vehículo se enciende, el voltaje aumenta aproximadamente hasta 14V. Esto es debido a que una vez encendido el vehículo, los

componentes internos demandan más energía y por lo tanto la batería se ve forzada a suministrar más voltaje a los diferentes componentes que conforman el vehículo.

La etapa de adquisición de datos comienza con la computadora del vehículo, la cual es alimentada directamente de la batería del mismo. Como se vio en el capítulo 2.1.1.5., la computadora recibe y procesa información de todos los sensores del vehículo, dicha información es enviada cuando se requiere a través del puerto OBD II, el cual tiene una configuración diferente en sus pines de acuerdo al protocolo de comunicación de cada vehículo. Además, el puerto OBD II recibe alimentación proveniente de la computadora para que una vez que se conecte un escáner automotriz, este se energice sin necesidad de una fuente externa.

En este caso el escáner automotriz es el ELM327, el cual está conectado y por tanto energizado gracias al puerto OBD II del vehículo. El ELM327 está conectado vía Bluetooth a la unidad de control, por tanto, una vez que la unidad de control hace la petición de los datos mediante PIDs, el ELM327 recibe la petición y obtiene la información de los diferentes parámetros sensados por la computadora del vehículo.

Una vez que la unidad de control recibe los 5 parámetros a desplegar provenientes de la computadora del vehículo, dichos parámetros son enviados hacia una interfaz gráfica, y posteriormente en la etapa de proyección, dicha interfaz gráfica es proyectada vía HDMI sobre un cristal transparente alojado en el tablero del vehículo, gracias a un proyector alimentado por inversor de corriente.

3.3. Configuración del Software

3.3.1. Adquisición de los parámetros a desplegar del puerto OBD II

3.3.1.1. Desarrollo del entablamiento de sistema de comunicación vía Bluetooth entre la ECU del vehículo - escáner ELM327 y la Raspberry PI 3 B/B+

Para esta parte primera parte del proyecto, los elementos a utilizar son:

- Raspberry PI 3 B/B+
- Escáner ELM327
- Vehículo con sistema OBD II

Enseguida se debe de identificar si el vehículo a utilizar es compatible con el escáner ELM327, dicha información se encuentra en el Anexo B, o en el siguiente enlace:

<https://obd2-elm327.es/elm327-vehiculos-compatibles-obd2>

Una vez comprobada la compatibilidad exitosa, se procede a realizar los siguientes pasos:

1. Conectar el escáner ELM327 en el puerto OBD II del vehículo. Enseguida se observará el encendido del ELM327 debido a que el puerto OBD II siempre está energizado a 12V+ desde el pin 16.
2. Encender la Raspberry previamente cargada con el SO Raspbian y abrir la Terminal.
3. Una vez abierta la Terminal, se deben ejecutar los siguientes comandos: (Solo los comandos en rojo son los que se deben ejecutar en la Terminal)

Se inicia el modo Bluetooth

```
$ bluetoothctl
```

Nos aseguramos que esta encendido el Bluetooth de la Raspberry

```
# power on
```

Se habilita la opción para realizar la paridad con otro dispositivo mediante Bluetooth

```
# pairable on
```

Se asegura de que se esté ejecutando un agente para el dispositivo Bluetooth.

```
# agent on
```

Se asigna un agente por default para la conexión Bluetooth.

default-agent

Se comienza a realizar el escaneo de los dispositivos Bluetooth cercanos.

scan on

Una vez ejecutado el comando “scan on”, en la Terminal se verán líneas de texto similares a las que se muestran en la siguiente imagen.

```
[bluetooth]# scan on
Discovery started
[CHG] Controller B8:27:EB:B5:9E:6D Discovering: yes
[NEW] Device 19:CC:02:DA:7B:50 19-CC-02-DA-7B-50
[NEW] Device 61:B3:D1:79:BF:BA 61-B3-D1-79-BF-BA
[NEW] Device 40:CB:C0:CB:2E:6E 40-CB-C0-CB-2E-6E
[bluetooth]#
[NEW] Device 00:1D:AE:40:84:2D 00-1D-AE-40-84-2D
[CHG] Device 00:1D:AE:40:84:2D LegacyPairing: no
[CHG] Device 00:1D:AE:40:84:2D Name: OBDII
[CHG] Device 00:1D:AE:40:84:2D Alias: OBDII
[CHG] Device 40:CB:C0:CB:2E:6E RSSI: -96
[CHG] Device 61:B3:D1:79:BF:BA RSSI: -101
[CHG] Device 00:1D:AE:40:84:2D LegacyPairing: yes
```

Ilustración 28. Demostración de la obtención de la dirección Bluetooth del ELM327

Como se puede observar, la parte sombreada de blanco es la dirección del escáner ELM327, el cual tiene por nombre OBDII.

En la mayoría de los dispositivos ELM327 el nombre es el mismo, pero la dirección de cada uno es diferente.

4. Una vez identificada la dirección del ELM327, se debe copiar dicha dirección y enseguida ejecutar los siguientes comandos:

Se realiza la paridad de la Raspberry con el escáner ELM327.

pair (pegar la dirección del escáner ELM327, sin paréntesis)

Debemos esperar unos segundos a que se realice la paridad y enseguida aparecerá una respuesta por parte de la Terminal para ingresar la contraseña del ELM327. En la mayoría de los casos la contraseña es: 1234 ó 6789 ó 0000.

Una vez aceptada la contraseña, se procede a ejecutar el siguiente comando

Se hace una comunicación Bluetooth segura, entre el ELM327 y la Raspberry para que ningún otro dispositivo pueda intervenir en la misma.

trust (pegar la dirección del escáner ELM327, sin paréntesis)

Una vez realizados los pasos correctamente, salimos del modo Bluetooth.

Salimos del modo Bluetooth

quit

De acuerdo con la hoja de datos del microcontrolador ELM327, este dispositivo actúa como un puente entre el puerto OBD II del vehículo y un puerto serie RS232. Debido a que el ELM327 que se está utilizando es Bluetooth y no USB, debemos emular un puerto serie RS232, esto con el propósito de completar dicho puente y poder enviar y recibir información.

Como se vio en el capítulo 2.1.4., existen diferentes protocolos Bluetooth, pero en dicho capítulo, se enfocó solo en uno llamado RFCOMM, el cual nos permite emular sesenta conexiones simultáneas de puertos serie RS232, ideal para completar el puente anteriormente mencionado.

Para emular dicho puerto RS232 se debe ejecutar el siguiente comando en la Terminal

Se emula el puerto serie RS232

\$ sudo rfcomm bind rfcomm0 (pegar la dirección del escáner ELM327, sin paréntesis)

Una vez ejecutado dicho comando, la emulación del puerto serie RS232 se ha realizado, enseguida se debe instalar un programa llamado “screen”, el cual nos permite desplegar una ventana para realizar el envío y recepción de datos a través del puerto serie RS232 emulado anteriormente.

Para instalar dicho programa, basta con ejecutar el siguiente comando en la Terminal

Se instala el programa screen

\$ sudo apt-get install screen

Posteriormente se ejecuta dicho programa con el siguiente comando, para desplegar la ventana de envío y recepción de datos. Y de esta manera finalizar el entablamiento del sistema de comunicación entre la ECU del vehículo - escáner ELM327 y la Raspberry PI 3B.

Se ejecuta el programa screen para abrir el puerto serie RS232

\$ screen /dev/rfcomm0

3.3.1.2. Configuración del ELM327 para la detección automática de protocolos de comunicación

Por default, el escáner ELM327 no detecta ningún protocolo una vez que se conecta al puerto OBD II de cualquier vehículo, este debe configurarse de acuerdo al protocolo de comunicación del vehículo, para realizar dicha configuración se debe de seguir el siguiente procedimiento.

Una vez abierta la ventana del puerto serie RS232 que se emulo en el procedimiento anterior, se debe de enviar una serie de comandos AT para poder configurar el microcontrolador del escáner de manera que pueda detectar automáticamente el protocolo de comunicación de cualquier vehículo compatible y de esta manera poder obtener los parámetros a utilizar en el proyecto.

Como se mencionó en la sección 2.1.5.3., los comandos AT para poder configurar el microcontrolador son los siguientes: ATZ, ATL1, ATH1 Y ATSP0, los cuales deben de ser enviados en ese orden para garantizar la correcta configuración del microcontrolador y por tanto detectar automáticamente el protocolo de comunicación de cualquier vehículo compatible. Una vez que se envía cada uno de los comandos AT, la respuesta que se debe recibir es un OK, el cual nos indica que el comando AT se envió con éxito y por tanto el microcontrolador se configuro correctamente.

```
>atz  
  
ELM327 v2.1  
  
>atl1  
OK  
  
>ath1  
OK  
  
>atsp0  
OK
```

Ilustración 29. Demostración del envío y respuesta de los comandos AT

3.3.1.3. Obtención de los parámetros provenientes de la ECU del vehículo

Como se mencionó en la sección 2.1.3.1, la norma OBD II SAE J1979 define 10 modos de operación, en este caso, para la obtención de los parámetros provenientes de la ECU del vehículo, se utilizó el modo de operación 01, el cual de acuerdo a la tabla 1, dicho modo de operación nos brinda acceso a los datos en tiempo real de los valores analógicos y digitales de las salidas y entradas a la ECU del vehículo.

Además, se utilizó la lista de comandos PIDs del Anexo A, para que de esta manera teniendo el modo de operación y el PID del parámetro que se desea conocer, se pueda enviar en forma hexadecimal la petición del parámetro a través del puerto serie RS232 emulado anteriormente.

Como se muestra en la ilustración 30, él envió de comandos para la petición de los parámetros debe de seguir dicho orden.

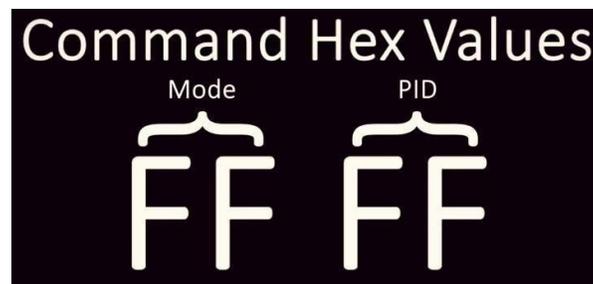


Ilustración 30. Modo de envío de comandos para la obtención de parámetros provenientes de la ECU del vehículo [Ilustración], por Tinkernut, 2018, Youtube (<https://www.youtube.com/watch?v=DABytIdutKk&t=26s>)

Primero debe de escribirse el modo de operación, en este caso 01, y enseguida el comando PID del parámetro a conocer. Tomando como ejemplo las revoluciones por minuto del vehículo, el PID sería 0C, por tanto, el comando completo para la obtención de la velocidad quedaría como se muestra en la ilustración 31.



Ilustración 31. Ejemplo de comando completo para la obtención de la velocidad del vehículo [Ilustración], por Tinkernut, 2018, Youtube (<https://www.youtube.com/watch?v=DABytIdutKk&t=26s>)

Una vez formado el comando completo, se procede a enviar dicho comando a través del puerto serie RS232, y como respuesta se recibe una línea de texto similar a la que se muestra en la ilustración 32.

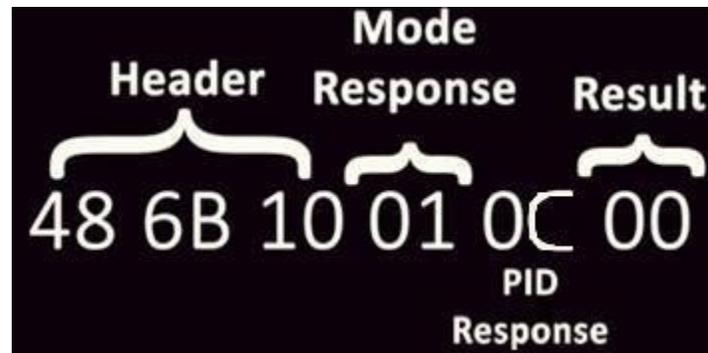


Ilustración 32. Respuesta recibida una vez realizada la petición de la velocidad [Ilustración], por Tinkernut, 2018, Youtube (<https://www.youtube.com/watch?v=DABytIdutKk&t=26s>)

Como se puede observar en la ilustración 32, la respuesta de la ECU viene dada en hexadecimales, en donde los 3 primeros valores hexadecimales son la cabecera, la cual indica los dígitos de comprobación de que la respuesta se recibió correctamente (dichos dígitos de comprobación varían dependiendo el protocolo de comunicación).

El 4to valor hexadecimal, es el modo de operación en el cual se realizó la petición de los parámetros.

El 5to valor hexadecimal, es el PID del parámetro del cual se realizó la petición.

Y finalmente el 6to valor hexadecimal, es el resultado de las revoluciones por minuto (en la ilustración 32, el resultado marca 00, simulando que el vehículo no está en encendido, pero si con el swtich abierto).

Si se desea realizar la petición de otro parámetro, al momento de realizar el envío del comando completo, basta con cambiar el valor del PID por el valor del parámetro que se desee adquirir.

3.3.1.4. Interpretación de los parámetros provenientes de la ECU del vehículo

Como se vio hace un momento, las respuestas de los diferentes valores de los parámetros sensados por la ECU, vienen dadas en valor hexadecimal, lo cual complica su lectura. Por tanto a continuación se mostrará la forma en la que se pueden interpretar dichos valores en forma decimal, para facilitar su lectura y entendimiento.

Como primer paso, se deben instalar la librería Python-OBD, de la siguiente manera

Se instala la librería Python-OBD

```
$ pip install obd
```

Una vez instalada dicha librería, se procede a crear un script con el siguiente código:

```
import obd # Se importa La Librería obd

connection = obd.OBD() # Se realiza La conexión automática en un puerto
disponible

c = obd.commands.RPM # Se selecciona un comando OBD, en este caso son Las RPM y
se asigna a una variable

response = connection.query(c) # Se realiza La consulta y se interpreta La
respuesta

print(response.value) # Se imprime el valor decimal obtenido

connection.close() # Se cierra La conexión
```

Una vez creado el script con el código anterior, se procede a ejecutar dicho script.

Como se puede observar, el código es sencillo e intuitivo puesto que solo se interpreta el parámetro de las RPM. La respuesta de dicho código se muestra en la ilustración 33.

A screenshot of a terminal window showing the output of a Python script. The text '755.25 revolutions_per_minute' is displayed in a monospaced font against a dark background.

Ilustración 33. Interpretación a valor decimal, del valor de las revoluciones por minuto

Cabe mencionar que, para obtener el valor de las revoluciones por minuto, es necesario ejecutar el script cada vez que se desee obtener dicho valor.

Más adelante se explicará cómo se estableció una comunicación asíncrona, en la cual no es necesario ejecutar el script varias veces, puesto que con dicha comunicación asíncrona, basta con ejecutar el script una vez para obtener una constante actualización de los valores en determinados intervalos de tiempo.

3.3.2. Desarrollo del servidor local

A continuación, se explica el proceso de desarrollo del servidor local. Como previamente se ha mencionado, el servidor esperará la acción del cliente para hacer su función, este cuando reciba algún carácter o mensaje por parte del cliente, mandará la cadena previamente formada que cuenta con los 5 parámetros obtenidos del automóvil.

Para la elaboración del servidor local, se desarrolló el siguiente diagrama de flujo.

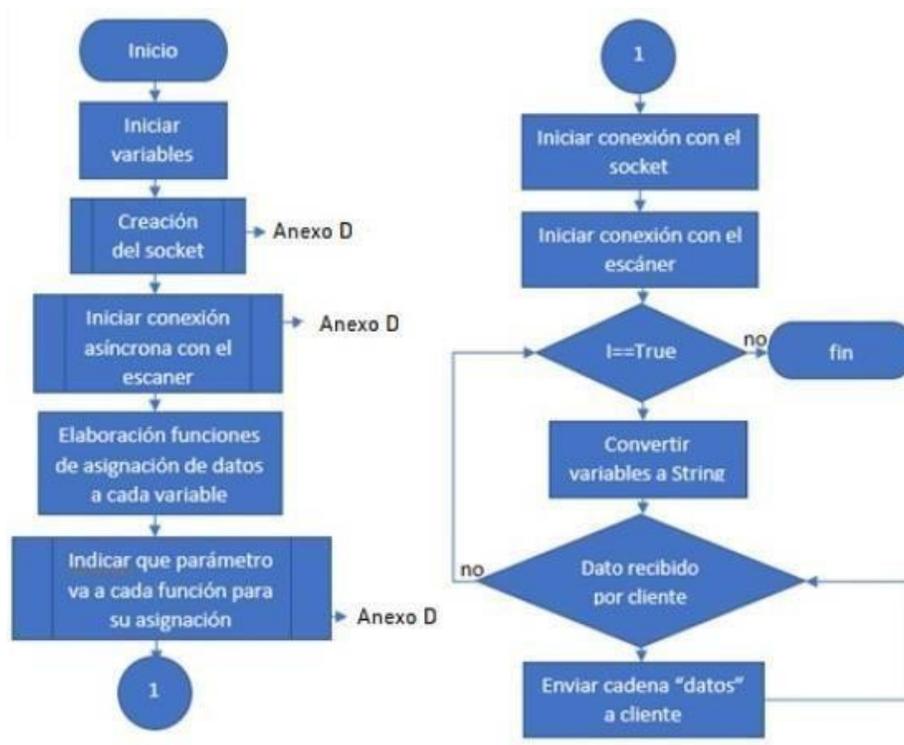


Diagrama 4. Diagrama de flujo de los procesos realizados por parte del servidor local

Para el desarrollo del servidor local se utilizó el lenguaje de programación Python, el cual nos brinda una gran facilidad a la hora de elaborar servidores locales. Además, se utilizó dicho lenguaje para poder hacer uso de la librería Python OBD para la interpretación de los parámetros obtenidos de la computadora del vehículo.

El código se inició con la importación de las siguientes librerías.

1. **Time:** Nos permite manejar los tiempos de envío de datos.
2. **Socket:** Nos brinda clases específicas para manejar el transporte de datos que en este caso sería entre un cliente y un servidor local.

Para la creación del servidor primero iniciamos nuestro objeto socket e indicamos que se va a conectar a una dirección IPv4 (familia AF_INET) y va a ser de tipo Stream.

Para no tener que cambiar el puerto en el código cada vez se desee ejecutar el servidor lo que se hizo fue utilizar la función “setsockopt()”. Para eso, establecimos el reusó de dirección especificando el nivel (SO_SOCKET) en el que se encuentra lo que quisimos establecer, después indicamos la acción que queríamos establecer (SO_REUSEADDR) y al final especificamos si lo queremos hacer (1=TRUE).

Ya una vez indicado que se reutilizaría el puerto, se le indicó al servidor que se conectara a la dirección “127.0.0.1” (localhost) en el puerto 1999 usando la función “bind()”. También se le indico que solamente se conectara a un cliente usando la función “listen()”. Para que nuestro servidor aceptara una conexión cuando el cliente se conectara, se hizo uso de la función “accept()”, que regresa un par de valores. El primer valor es un objeto de clase socket que tiene el uso de enviar y recibir datos y el otro valor es la dirección vinculada al zócalo en el otro extremo de la conexión.

Una vez desarrollado el servidor se realizó la importación de la librería OBD para la conexión con el escáner ELM 327.

Para la implementación de esta librería, se inicia declarando el objeto de la clase de conexión asíncrona para poder tener los valores de los parámetros a proyectar, actualizados cada cierto intervalo de tiempo

El primer paso que se realizó para poder mantener nuestras variables actualizadas fue la elaboración de funciones para asignarle su valor a cada una. Para eso se estableció que mientras la respuesta recibida por la librería no sea nula esta guardara el ultimo valor en la variable establecida previamente convertida a entero, puesto que la librería maneja valores de tipo cadena.

El segundo paso fue la utilización de la función watch() que cuenta con 2 objetos, el primero es el comando del valor que nosotros queremos que nos mantenga actualizados y el segundo es una llamada a una de las funciones previamente hechas para guardar dicho valor del comando.

Antes de iniciar la conexión asíncrona, se asignó a una variable un valor booleano gracias a la función `is_connected()` para verificar que se mantuviera conectado el escáner, puesto que también estábamos interesados en mantener al conductor alerta en caso de que el escáner dejase de comunicar. Y por último, se inició la conexión asíncrona con el escáner ELM327.

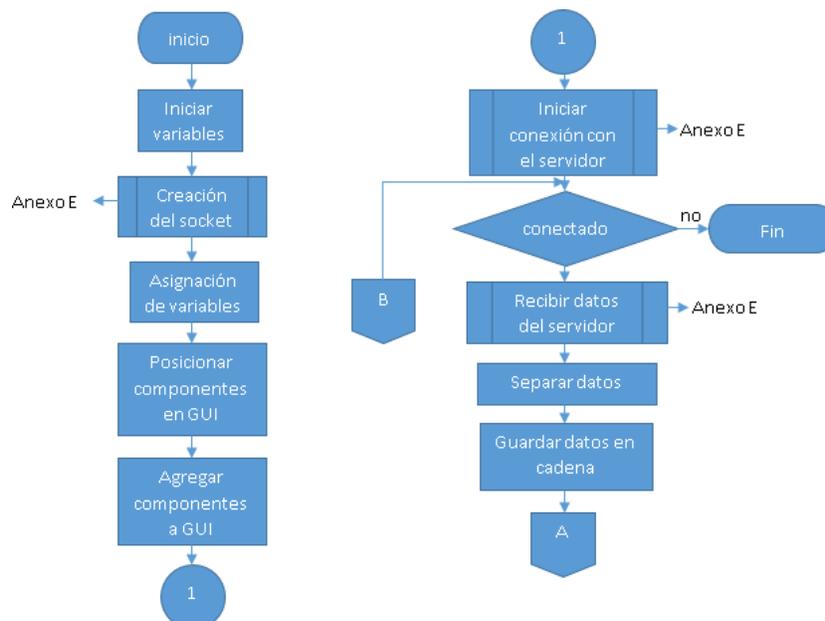
Una vez asignados los valores de los parámetros a mostrar en nuestras variables, lo último que se hizo fue sumar estos valores en una cadena llamada “datos”. Los valores se fueron sumando uno a uno, cada uno separado por una coma (“,”) para que así el cliente pudiera identificarlos y distribuirlos en la interfaz gráfica.

Para enviar la cadena se hizo uso de dos funciones, una de ellas es “`send()`” la cual nos permite enviar información en forma de bytes al cliente y es por ello que también hicimos uso de la función “`bytes()`” la cual transforma nuestra cadena para así poder ser enviada al cliente.

3.3.3. Desarrollo de la interfaz grafica

A continuación, se explica el desarrollo del cliente, o para este proyecto también llamada interfaz gráfica. El cliente en nuestro diseño se encarga de recibir y distribuir los datos obtenidos por el servidor local. A diferencia del servidor, el cliente se desarrolló en Java debido a que este lenguaje nos brinda un gran número de componentes y funciones para el desarrollo de interfaces gráficas.

Para la elaboración del cliente se desarrolló el siguiente diagrama de flujo.



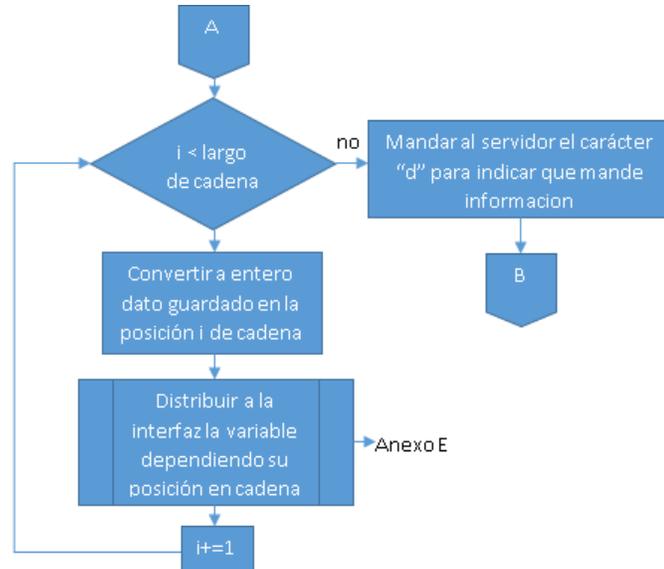


Diagrama 5. Diagrama de flujo de los procesos realizados por parte de la interfaz gráfica

El desarrollo del cliente inicia con la importación de las siguientes librerías:

1. java.net.socket : Nos proporciona las clases para la implementación de aplicaciones de red y entablar comunicación entre maquinas a través de la red (en este caso entre la red local).
2. java.net.InetSocketAddress: Esta clase implementa una dirección de socket IP (dirección IP, numero de puerto), que en nuestro caso es (localhost, numero de puerto).
3. Java.io.DataInputStream: Es una secuencia de datos de entrada que permite que la aplicación lea datos de una secuencia de entrada subyacente de la forma independiente de la máquina.
4. java.io.DataOutputStream: Es un flujo de salida de datos que permite que una aplicación escriba tipos de datos en un flujo de salida de forma portátil.
5. java.io.IOException: Nos da una señal de que se ha producido una excepción de entrada y salida de algún tipo.
6. java.awt: Contiene todas las clases para crear interfaces de usuario y para colocar gráficos e imágenes, hacemos uso de esta librería junto con una variedad de sus componentes como Color, FlowLayout, GridLayout, Frame.
7. Javax.swing: Proporciona una serie de componentes que funciona de la misma manera en todas las plataformas para que las herramientas de una interfaz gráfica no dependan de plataforma y se pueda ejecutar de la misma manera en diferentes plataformas.
8. java.util: Esta librería la utilizamos para el funcionamiento del delay colocado en la parte final del código.

Una vez importadas las librerías, se comenzó con la elaboración de las clases, para el cliente se elaboraron 3 clases y una interfaz pública. Primero se inició con la clase motor de usuario (UserEngine) el cual se encarga del envío y recepción de datos recibidos o enviados al servidor.

El motor de usuario con la importación de las librerías 1,3,4,5, hace que cuando queremos enviar un dato al servidor este lo convierte en bytes y lo manda, y cuando se recibe, este los pasa de bytes a String y con ayuda de nuestra interfaz publica UserRecibeListener guarda el String en la función UserRecibeEvent.

Habiendo desarrollado las vías de comunicación del cliente con el servidor empezamos con la implementación de la interfaz gráfica.

Para la interfaz gráfica, utilizamos la clase ventana extendiendo JFrame e implementando ActionListener y UserRecibeListener, como un programa común iniciamos con la inicialización de las variables, de estas variables las más destacables son la variable “User” del motor de usuario (UserEngine), la variable “conexión” de Socket y la variable “IP” de InetAddress ya que estas son las que nos permitirán implementar las vías de comunicación con nuestro servidor anteriormente desarrolladas.

Para la conexión se creó la función conectar con un objeto tipo entero llamado puerto. Esta función establece que dirección IP y que puertos se van a utilizar, para nuestro caso utilizamos la IP local “127.0.0.1” y el puerto 1999. Al inicio se pensaba usar el puerto 80 pero nos percatamos que no funcionaba porque la Raspberry tiene este puerto para los servidores HTTP por lo tanto se decidió utilizar un puerto menos común como lo es el 1999.

En nuestra función conectar se manda a llamar a la función “init” que tiene como función inicializar los componentes de la ventana y distribuirlos dependiendo de los valores recibidos por el servidor

3.3.3.1. Diseño de la interfaz gráfica

Para el diseño de la interfaz gráfica, se determinó y acordó un diseño sencillo y entendible para aquellas personas que tuvieran nulos conocimientos acerca de los parámetros a desplegar en la interfaz, por lo que no fue necesario indicar números exactos de temperatura del refrigerante, temperatura del medio ambiente y del medidor de mezcla aire-combustible, simplemente bastó con indicar mediante barras, el cambio de colores dependiendo si la temperatura es fría, intermedia o caliente, o en caso de la mezcla aire combustible, si dicha mezcla es rica o pobre.

Para la elaboración de la interfaz gráfica, se tomaron en cuenta 3 aspectos, los cuales son colores a ocupar, tamaño de la interfaz y distribución de la interfaz. Esto con el propósito de evitar sobrecargar

la interfaz con bastantes número e información que lejos de ayudar al usuario a realizar una mejor conducción, esto podría resultar en un factor distractor para el mismo.

Velocidad

Para el despliegue de la velocidad del automóvil se decidió hacer uso de 30 imágenes 10 para centenas, decenas y unidades, cada una mostrando un valor del 0 al 9. Se decidió utilizar imágenes para poder tener libertad sobre el diseño de los números a mostrar. Para nuestro primer parámetro decidimos ocupar el área que cubre de los pixeles 0 x 0 al 958 x 276 ya que el conductor tiene que estar en constante monitoreo de la velocidad para no incumplir con las normas vehiculares de límite de velocidad, por eso fue importante para nosotros que este valor fuera el primero que llamara la atención del usuario.



Ilustración 34. Ejemplo del despliegue de la velocidad en la interfaz grafica

Temperatura del refrigerante

Cada automóvil tiene características de funcionamiento diferentes y una de ellas es el rango de la temperatura de trabajo del motor. Esta temperatura puede variar en torno al tamaño del motor, al sistema de enfriamiento, numero de cilindros, entre otros aspectos. Pero un promedio de temperatura que se investigó para vehículos convencionales de 4, 6 y 8 cilindros, es el siguiente:

De 0°C a 70°C, la temperatura se considera baja, ya que el motor aún no alcanza su temperatura promedio de trabajo. En la interfaz, esta condición se representa con unabarra azul.

De 71°C a 104°C, la temperatura se considera promedio, dentro de estos valores el motor alcanza su temperatura promedio de trabajo, y se mantiene dentro de dicho rango siempre y cuando el vehículo se encuentre en perfectas condiciones. En la interfaz, esta condición se representa con una barra verde mayor a 105°C, se considera un sobrecalentamiento en el motor, y si dicha temperatura se mantiene en ese rango por bastante tiempo, se debe realizar un chequeo del motor o del sensor de temperatura del refrigerante.

Para este parámetro se utilizó el área de 958 x 0 px a 1254 x 276 px.

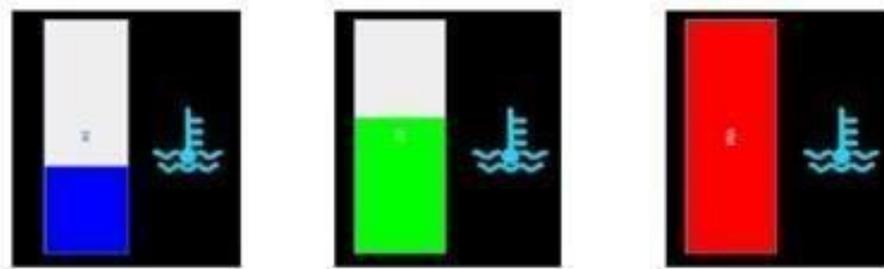


Ilustración 35. Ejemplo de las diferentes barras para la temperatura del refrigerante

Medidor de mezcla aire-combustible

Cuando se le pide al escáner ELM327 el voltaje del sensor de oxígeno 1/1, este nos regresa un valor en un rango entre 0.1V a 0.9V, por lo que se elaboraron 2 imágenes para determinar si la mezcla es rica o pobre, en caso de que la mezcla sea pobre ($\leq 0.5V$), la imagen de la barra se torna de un color rojo y cuando es una mezcla rica (> 0.5) la barra se torna de un color verde. Si el conductor se percata que la barra se queda en un solo color por demasiado tiempo significa que el sensor de oxígeno 1/1 está fallando, y por tanto la computadora no está regulando bien la mezcla aire-combustible.



Ilustración 36. Ejemplo de los medidores de mezcla aire-combustible

Temperatura del medio ambiente

Este valor al igual que la temperatura del refrigerante, se despliega en una barra de progreso. Con base en la media de temperatura en la ciudad de México se estableció un rango, para temperaturas menores a 12°C la barra se tornará de color azul, para temperaturas entre 12°C y 24°C la barra se tornará de color verde y para temperaturas superiores a 24°C la barra se tornará de color rojo.

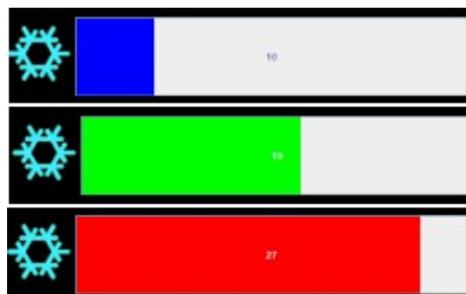


Ilustración 37. Ejemplo de las diferentes barras para la temperatura del medio ambiente

Revoluciones por minuto

Como último parámetro se desplegaron las revoluciones por minuto del motor. El despliegue es de la misma manera que la velocidad, la única diferencia es que a las revoluciones se le agregó otro dígito puesto que alcanza un valor numérico mucho mayor en comparación a la velocidad. Este parámetro es importante debido a que nos permite determinar el momento en donde debemos de realizar los cambios de velocidades, para no forzar el motor y correr el riesgo de ocasionar un desperfecto en el mismo.

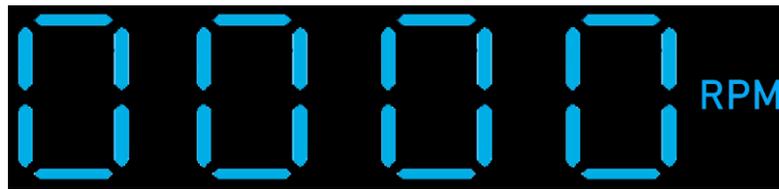


Ilustración 38. Ejemplo del despliegue de las revoluciones por minuto del motor en la interfaz gráfica

Conexión Bluetooth

Además de los 5 parámetros obtenidos de la ECU del vehículo, se decidió agregar un indicador que marca cuando el escáner se encuentra conectado con nuestro sistema. La librería utilizada para la obtención de datos nos brinda una función que regresa un valor booleano indicando el estado de la conexión con el escáner, en caso de que la función regrese un TRUE significa que hay conexión con el escáner de lo contrario la función nos avisara con un FALSE. Por lo tanto, el cliente recibe un 1 o un 0 dependiendo del estado de la conexión alternando así de una imagen del logo Bluetooth de verde (conectado) a roja (desconectado).



Ilustración 39. Indicador del estado de la conexión Bluetooth

Finalmente, en la ilustración 40 se muestra el diseño final de la interfaz gráfica.

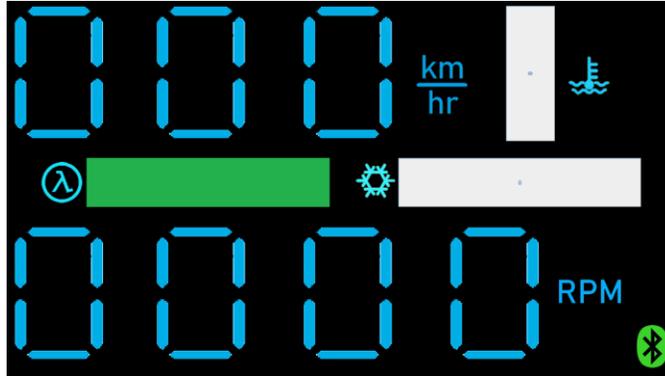


Ilustración 40. Diseño final de la interfaz grafica

3.3.4. Armado del sistema Raspberry PI 3B - ELM327

En esta parte se explica el proceso de armado del sistema en donde se aloja la Raspberry PI 3B y el escáner automotriz ELM327.

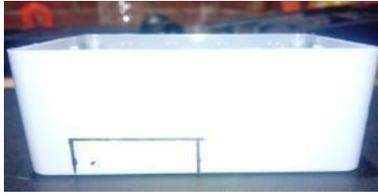
El propósito de alojar juntos estos elementos, fue para hacer un sistema más compacto que pudiera adaptarse en cualquier localización donde se encuentre el puerto OBD II en los diferentes vehículos compatibles, además, otro motivo fue debido a que la señal Bluetooth del escáner automotriz ELM327 tiene un alcance máximo de 1 metro, por tanto para evitar pérdidas de conexión se decidió alojar ambos elementos en un gabinete de plástico resistente al impacto.

A continuación se muestran diferentes fotografías que permiten ver el proceso de armado del sistema.

Primero se marcaron con un plumón los orificios que debía tener el gabinete para permitir tener a la vista los puertos de salida y de entrada de la Raspberry PI 3B, tales como puerto de alimentación, puerto HDMI, puertos USB, y puerto ethernet. Además, se hizo un orificio adicional en la parte frontal del gabinete para poder colocar el escáner ELM327.



Fotografía 1. Vista frontal del sistema para el alojamiento del escáner ELM327



Fotografía 2. Vista lateral izquierda para la entrada de alimentación de la Raspberry PI 3B y la salida del puerto HDMI



Fotografía 3. Vista trasera para la salida de los puertos USB y ethernet.

Una vez marcados y cortados los orificios del gabinete de plástico, se procedió a pintarlo y a alojar la Raspberry y el escáner dentro del mismo.



Fotografía 4. 1) Muestra la distribución interna de la Raspberry y el escaner ELM327 dentro del gabinete de plástico. 2) Vista frontal del sistema. 3) Vista trasera del sistema. 4) Vista lateral izquierda.

Finalmente, se muestra el sistema conectado al puerto OBD II de un Seat Ibiza 2015, y tal como se puede apreciar, el sistema se acopla sin problemas y sin intervenir en la conducción.



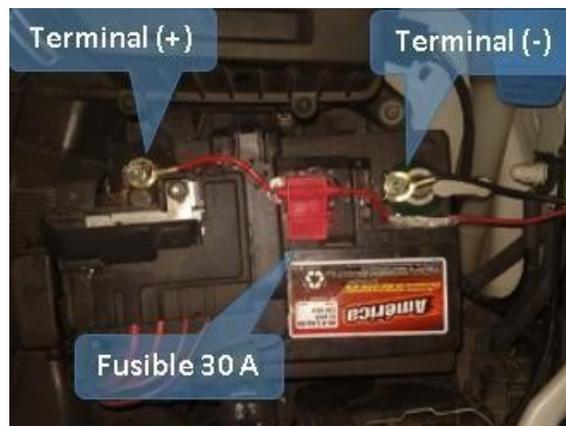
Fotografía 5. Sistema Raspberry PI 3B - ELM327 conectado al puerto OBD II de un Seat Ibiza 2015

3.3.5. Instalación de la etapa de alimentación

Como se mencionó en la sección 4.2., existe una etapa de alimentación que permite suministrar energía al proyector y a la Raspberry.

Dicha etapa de alimentación se realiza gracias al inversor de corriente automotriz de 400 W. El cual obtiene su alimentación directamente de la batería del vehículo.

Como se puede observar en la fotografía 6, en la terminal positiva del inversor se colocó un fusible de 30 A para evitar algún daño en el inversor ante cualquier pico de voltaje proveniente de la batería.



Fotografía 6. Inicio de la etapa de alimentación

El inversor de corriente se colocó debajo del asiento del conductor, para que tanto la Raspberry como el proyector, pudieran tener un punto intermedio de conexión. Cabe mencionar que el inversor cuenta con un ventilador interno, el cual ayuda a regular la temperatura interna evitando sobrecalentamientos en el mismo.



Fotografía 7. Distribución la etapa de alimentación dentro del vehículo

3.3.6. Instalación de la etapa de proyección

Debido a que el vehículo en el cual se implementó el presente proyecto era prestado, esto nos limitó demasiado, ya que no se podía perforar, cortar, quitar o alterar ninguna parte del habitáculo para poder instalar la etapa de proyección de una mejor manera.

Debido a esta limitante, la instalación se realizó utilizando un trípode para poder soportar al proyector y que este se pudiera mover libremente en cualquier dirección.

Como se puede observar en la fotografía 8, el trípode está encima de la palanca de velocidades. Cabe mencionar que se determinó una altura ideal en la cual el trípode no estorbara al momento de realizar los cambios de velocidades.



Fotografía 8. Instalación del tripié y el proyector

Una vez colocado el proyector y el tripié, lo siguiente que se realizó, fue moldear una base de resina epoxica en la cual se colocaría el cristal. Se decidió utilizar resina epoxica para poder moldear la base a una medida adecuada que se pudiera colocar en el tablero del vehículo en el cual se montaría el sistema.



Fotografía 9. 1) Se realizó un molde de yeso con la forma de la base deseada. 2) Se obtuvo el molde de la base. 3) Se rellenó el molde de yeso con resina epoxica para que finalmente se obtuviera la base final.

Posteriormente, a la base se le realizó una apertura de 2 cm de profundidad para poder colocar el cristal sobre dicha apertura. Cabe mencionar que sobre un lado del cristal se colocó una mica de retroproyección para poder visualizar la proyección de una manera más clara, y por el lado contrario del cristal, se colocó una película anti asalto para evitar que la proyección atravesara hacia el parabrisas y se viera distorsionada. Finalmente, la base se pegó con silicón sobre el tablero. Y de esta manera la instalación de la etapa de proyección quedaría terminada.



Fotografía 10. Cristal colocado sobre la base, y esta a su vez está pegada sobre el tablero

Al estar funcionando los 3 sistemas descritos anteriormente, se tiene como resultado la proyección de los 5 parámetros estipulados en el objetivo general, y por tanto, la culminación del proyecto.



Fotografía 11. Sistema de proyección funcionando

CAPÍTULO 4.

VALIDACIÓN DEL SISTEMA

Antes de comenzar con el desarrollo del servidor local, interfaz gráfica, conexiones, instalación de los componentes, entre otros aspectos del proyecto, primero se tuvo que encontrar un vehículo que nos brindara todos los valores a desplegar en la interfaz gráfica.

Para esto, se tuvo que crear un script con un código similar al de la sección 4.3.1.4., en el cual se realizó la petición de 12 parámetros, y de esta manera ver que vehículo nos brinda la mayor cantidad de parámetros y por ende el que nos conviene más.

Los 12 parámetros de los cuales se realizó la petición son los siguientes:

- RPM
- Velocidad
- Voltaje del sensor de oxígeno 1/1
- Temperatura del refrigerante
- Presión MAP
- Nivel de combustible
- Carga del motor
- Tiempo de avance
- Tiempo de encendido
- Presión barométrica
- Carga absoluta
- Temperatura ambiente

Esta prueba se realizó en 6 diferentes vehículos, y gracias a dichas pruebas, se pudo realizar una tabla comparativa, mediante la cual nos basamos para elegir al Seat Ibiza como el vehículo en el cual implementaríamos el sistema de proyección, esto debido a que como se observa en la tabla 6, dicho vehículo nos brindó todos los parámetros necesarios, además de que fue el vehículo en el cual tuvimos más accesibilidad al momento de trabajar con él.

```
import obd
connection = obd.OBD()
a = obd.commands.RPM
b = obd.commands.SPEED
c = obd.commands.COOLANT_TEMP
d = obd.commands.INTAKE_PRESSURE
h = obd.commands.ENGINE_LOAD
j = obd.commands.TIMING_ADVANCE
l = obd.commands.RUN_TIME
n = obd.commands.BAROMETRIC_PRESSURE
p = obd.commands.ABSOLUTE_LOAD
q = obd.commands.AMBIANT_AIR_TEMP
```

```
pi@raspberrypi:~ $ nano obd_hud_test.py
pi@raspberrypi:~ $ python obd_hud_test.py
951.0 revolutions_per_minute
0.0 kph
95 degC
24 kilopascal
4.3137254902 percent
2.0 degree
1258.0 second
78 kilopascal
19.2156862745 percent
18 degC
pi@raspberrypi:~ $
```

Ilustración 41. Petición y respuesta de diferentes parámetros para el vehículo Seat Ibiza

Tabla 6. Comparación de la obtención de los diferentes parámetros en diferentes vehículos

Modelo	Seat Ibiza 2015 CAN 11bit 500kb	Aveo 2009 KWP FAST	Spark 2010 KWP FAST	Touran 2011 CAN 11bit 500kb	Jetta 2015 CAN 11bit 500kb	Prius C 2019 CAN 11bit 500kb
RPM	✓	✓	✓	✓	✓	✓
Velocidad	✓	✗	✗	✓	✓	✓
Sensor de oxígeno 1/1 (v)	✓	✓	✓	✓	✗	✓
Temperatura de refrigerante	✓	✗	✗	✓	✓	✓
Presión MAP	✓	✗	✗	✓	✓	✓
Nivel de combustible	✗	✗	✗	✓	✗	✗
Carga de motor	✓	✗	✗	✓	✓	✓
Tiempo de avance	✓	✗	✗	✓	✓	✓
Tiempo de encendido	✓	✗	✗	✓	✓	✓
Presión barométrica	✓	✗	✗	✓	✓	✓
Carga absoluta	✓	✗	✗	✓	✓	✓
Temperatura ambiente	✓	✗	✗	✓	✓	✗

El hecho de que haya vehículos en los que, si se obtuvieron la mayoría de los parámetros, tales como el Seat Ibiza, el Volkswagen Jetta, la Volkswagen Touran, Toyota Prius C, es debido al protocolo de comunicación que estos vehículos manejan y a la librería Python- OBD. Ya que nos dimos cuenta que son dos aspectos que van de la mano al momento de la obtención de los parámetros, esto es debido a que la librería fue diseñada principalmente para vehículos con las diferentes variantes de protocolos CAN, en cambio los demás vehículos tales como el Chevrolet Aveo o el Chevrolet Spark, los cuales tienen un protocolo de comunicación KWP FAST, y por tanto se ven limitados ante esta librería por lo que solo responden a 2 parámetros de los 12 pedidos (RPM y voltaje de sensor de oxígeno 1/1).

Por ende, este sistema en general se ve limitado para funcionar con vehículos con un protocolo diferente al CAN, y por otro lado, el sistema brinda un funcionamiento correcto con vehículos con protocolo de comunicación CAN.

Otro dato interesante que se puede observar en la tabla 6, es que la Volkswagen Touran, es el único vehículo que nos dio el nivel de combustible, esto se debe a que existen dos formas de transmisión de señales en el vehículo, las señales directas y las señales indirectas.

Las señales directas son las que se envían de algún sensor hacia un medidor, sin pasar por la computadora del vehículo.

Las señales indirectas son las que se envían de algún sensor hacia la computadora del vehículo y después de ser procesada dicha señal, es enviada hacia un medidor.

En este caso la señal obtenida por el flotador de combustible de la Volkswagen Touran, es indirecta, ya que dicha señal es enviada hacia la computadora del vehículo y una vez procesada, se envía hacia el medidor. Es por ello que dicho parámetro si se pudo obtener utilizando el escáner ELM327.

Por otro lado, para los vehículos CAN restantes de la tabla, la transmisión de dicha señal es directa hacia el medidor de combustible sin pasar por la computadora del vehículo.

Una vez analizados estos aspectos, se procedió a montar los 3 sistemas mencionados en la sección 4.4, dentro del vehículo a utilizar, en este caso el Seat Ibiza. Dando como resultado el montaje mostrado en la fotografía 12.



Fotografía 12. Vista del conductor con el sistema de proyección montado

Como se puede observar en la fotografía 12, el sistema se alcanza a visualizar con gran claridad, siempre y cuando la proyección se realice en un horario nocturno, aunque de igual manera podría utilizarse en un horario soleado, siempre y cuando se cambie el proyector por uno que proyecte con una mayor intensidad (al menos 2500 lúmenes).

Una vez montado el sistema, se midió el ángulo de visión entre el conductor y el cristal, dando como resultado un valor de 20°, y de acuerdo a la ilustración 23 de la sección 2.1.6.2, esta apertura aún está en el campo de reconocimiento de símbolos, y solo bastaría con mover la

vista unos cuantos milímetros para llegar al campo de lectura (10°) y poder visualizar sin problema todos los parámetros de la interfaz.



Fotografía 13. Simulación de la conducción con el sistema de proyección encendido

El sistema resulto ser efectivo, y de gran ayuda para monitorear los diferentes parámetros del vehículo y por tanto prevenir averías o desperfectos en el vehículo, y en casos extremos, prevenir un accidente.

El único problema presentado durante las pruebas del sistema fue la perdida envió de datos entre la Raspberry y el escáner ELM327, esto se debe a que al realizar una comunicación Bluetooth con la Raspberry, este elemento resulta ser un poco inestable durante la conexión.

CAPÍTULO 5.

ANÁLISIS DE RESULTADOS

5.1. Conclusiones

Al inicio, como propósito principal de este sistema, se pretendía proyectar 5 parámetros principales del vehículo, los cuales eran, velocidad, revoluciones por minuto, temperatura del anticongelante, nivel de combustible y presión del múltiple de admisión, debido a que consideramos que son parámetros esenciales para determinar el correcto funcionamiento del vehículo y a su vez brindarle al conductor una tranquilidad de poder visualizar dichos parámetros sin necesidad de distraerse al voltear a ver el tablero de instrumentos. Por tanto, brindándole al usuario una conducción más eficiente, cómoda y segura.

A medida que transcurría el avance del proyecto, nos fuimos percatando de diferentes aspectos que dificultarían obtener y condicionar dos parámetros de los antes mencionados, los cuales son el nivel de combustible y la presión del múltiple de admisión, respectivamente.

El primer parámetro, como se mencionó en la validación del sistema, en la mayoría de los vehículos la señal del flotador de combustible es directa hacia el indicador de nivel de combustible, sin pasar por la ECU del vehículo, por tanto, si se realizaba la interfaz con dicho parámetro, se corría el riesgo de que hubiera vehículos donde sí se desplegara el nivel de combustible y otros donde no. Debido a esto, dicho parámetro se cambió por un medidor de mezcla aire-combustible. Como tal el nuevo parámetro no brinda información acerca del nivel de combustible, pero si nos permite darnos una idea respecto a si nuestro vehículo está inyectando más gasolina de la ideal, o si hay más partículas de aire en la mezcla. Y en caso de que nos percatemos de que dicho valor no varía y se queda fijo durante mucho tiempo, revisar el estado del sensor de oxígeno o a su vez del sistema de inyección.

El segundo parámetro que se decidió cambiar es la presión del múltiple de admisión, esto debido a que dicha presión no brindaba lecturas constantes que pudieran condicionarse para desplegarlas sobre la interfaz. Después descubrimos el motivo de ello, y esto se debió a que dicha presión es diferente en cada vehículo, ya que depende mucho de dos factores, tales como la velocidad a la que se pise el pedal de acelerador y el sistema de inyección de cada vehículo.

Por tanto, se decidió cambiar dicho parámetro, por uno que fuera más entendible para el usuario, el cual fue, la temperatura del medio ambiente. Decidimos este parámetro ya que

como se mencionó en la sección 4.3.3.1., este proyecto se enfoca más, para usuarios con conocimientos básicos de los sistemas del vehículo, y por tanto si se desplegaran otros parámetros tales como carga del motor, tiempo de avance, presión barométrica, entre otros, la interfaz podría tornarse confusa e ineficiente para el usuario.

A pesar de estos cambios, los objetivos del proyecto no tuvieron ninguna repercusión, ya que nuestro proyecto seguía cumpliendo con el propósito principal, el cual era, realizar un sistema de proyección de los indicadores del vehículo.

Conforme se seguía avanzando y realizando pruebas a la par, nos dimos cuenta que nuestro sistema tenía un fallo al momento de funcionar, el cual era la pérdida de conexión Bluetooth entre la Raspberry y el ELM327, y por tanto esto nos generaría el cierre automático de la interfaz gráfica. Para tratar de hacer más ameno el sistema, se determinó implementar un modelo cliente-servidor para el envío y recepción de los parámetros, este modelo se implementó con el propósito de que al momento en que se perdiera la conexión entre ambos dispositivos, la interfaz no se cerrara y además de que se guardara el último dato leído, quedándose a la espera de la conexión automática y nuevamente comenzar a realizar el envío y recepción de los parámetros.

Este proyecto fue un tanto difícil para nosotros, ya que utilizamos un elemento del vehículo con el cual no estábamos familiarizado a lo largo de la carrera, dicho elemento es el puerto OBD II. Que, a pesar de ser un elemento importante el diagnóstico del vehículo, desconocíamos todas sus funciones y su gran utilidad, pero con el paso del tiempo, fuimos adquiriendo los conocimientos mínimos básicos para entender el funcionamiento y el propósito de dicho puerto.

Otro motivo a destacar, fueron los lenguajes de programación utilizados, Java y Python. Con respecto a Java, nosotros tenemos un conocimiento básico/ intermedio de dicho lenguaje de programación, pero no fue suficiente para poder realizar la interfaz gráfica a desplegar, por tanto, aquí fue necesario la ayuda de nuestro asesor, con el cual analizamos y aprendimos un poco más de las diferentes funciones que nos brindaba este lenguaje de programación.

Con respecto a Python, este lenguaje de programación nunca había sido utilizado por ninguno de nosotros, pero a pesar de ello, mediante el uso de la librería Python OBD, conocimientos básicos de programación y comandos básicos de Python pudimos lograr la adquisición de los parámetros a desplegar. Y de igual manera, adquirimos nuevos conocimientos acerca de este lenguaje de programación.

A lo largo de la realización de este proyecto se generaron diversos problemas que no se habían previsto al inicio del mismo, esto fue debido al desconocimiento de los diferentes subsistemas que lo conforman. Problemas tales como pérdida de conexión, adquisición incompleta de parámetros o la instalación del sistema dentro del habitáculo, fueron los que se generaron. A la vez que se encontraba una solución para cada uno de estos problemas, nosotros íbamos generando nuevo conocimiento acerca de lenguajes de programación, ergonomía del vehículo, sistemas del vehículo, y por tanto siendo más conscientes del estudio que se debe de hacer al momento de realizar un proyecto de tal magnitud.

5.2. Trabajos futuros

El sistema actual puede ser mejorado de varias formas, Una de ellas en la implementación del proyector en una base tipo brazo manejado por servomotores el cual por medio de una interfaz de botones pueda tanto guardarse, como colocarse para iniciar con la proyección y así sea de forma más discreta ocupando menos espacio.

También se puede mejorar utilizando un proyector de menor tamaño y de mayor potencia dado que por las características de la película de retroproyección se necesitan al menos 2500 lúmenes en el proyector para una proyección óptima y definida. También se puede considerar un proyector que se conecte de forma inalámbrica para aminorar el cableado del sistema.

El vidrio de proyección mientras no está en uso puede llegar a ser algo incómodo y molesto para el conductor, por lo cual se propone integrar un sistema dentro del tablero del auto para poder desplegar el vidrio solamente cuando sea necesario.

El ahorro en el consumo de energía es de vital importancia por lo tanto también se propone un controlador en la intensidad de proyección utilizando un sistema de foto resistencias que, dependiendo de la intensidad de luz del medio ambiente, este controle un PWM y este regule la intensidad de la proyección.

Por último, se puede mejorar el diseño de distribución de los componentes para que el conductor no tenga la necesidad de interactuar con estos, también se puede mejorar el controlador para que desde el inicio de arranque se inicie la conexión asíncrona y el envío y recepción de datos, sin tener que ejecutarlos manualmente.

Referencias bibliográficas

- [1] Alós, V. (17 de Julio de 2019). *¿Qué es head up display?*. HIGHMOTOR. Recuperado el 3 de Diciembre de 2019, de <https://www.highmotor.com/que-head-up-display.html>
- [2] AutoMéxico. (13 de Marzo de 2019). *Head-up Display: Qué es y cuáles son los mejores*. AutoMéxico. Recuperado el 01 de Diciembre de 2019, de <https://automexico.com/conduccion/head-up-display-hud-mejores-aid2212>
- [3] Calvillo, M., Raúl Axel. (2017). *Prototipo de pre verificación ambiental por medio del sistema de diagnóstico a bordo del automóvil*. México DF: IPN-UPIITA.
- [4] CONUEE. (26 de Enero de 2017). *Diagnóstico a bordo (OBD).GOB*. Recuperado el 18 de Noviembre de 2019, de https://www.gob.mx/cms/uploads/attachment/file/187221/diagnosticoabordo_1_260117.pdf
- [5] Electronica facil. (Agosto de 2016). *Protocolos Bluetooth*. [electronicafacil](https://www.electronicafacil.net/tutoriales/Protocolos-Bluetooth.php). Recuperado el 29 de Noviembre de 2019, de <https://www.electronicafacil.net/tutoriales/Protocolos-Bluetooth.php>
- [6] Fundación MAPFREE. (2017). *seguridad vial*. [fundacionmapfre](https://www.fundacionmapfre.org/fundacion/es_es/programas/seguridad-vial/movil). Recuperado el 5 de Diciembre de 2019, de https://www.fundacionmapfre.org/fundacion/es_es/programas/seguridad-vial/movil
- [7] González, G., Eduardo (2012). *Sistema de diagnóstico OBD II SAE con proyección de parámetros en el parabrisas del automóvil*. México DF: IPN-UPIITA.
- [8] INTERNATIONAL, S. (2002). *SAE J1979: DIAGNOSTIC TEST MODES*. [resources](https://law.resource.org/pub/us/cfr/ibr/005/sae.j1979.2002.pdf). Recuperado el 29 de Noviembre de 2019, de <https://law.resource.org/pub/us/cfr/ibr/005/sae.j1979.2002.pdf>
- [9] mas adelante. (Febrero de 2016). *Que es bluetooth*. [masadelante](https://www.masadelante.com/faqs/que-es-bluetooth). Recuperado el 29 de Noviembre de 2019, de <https://www.masadelante.com/faqs/que-es-bluetooth>
- [10] OBD II - ELM327. (2017). *Modos prueba obd2 elm327*. [obd2-elm327](https://obd2-elm327.es/modos-prueba-obd2-elm327). Recuperado el 20 de Noviembre de 2019, de <https://obd2-elm327.es/modos-prueba-obd2-elm327>
- [11] Rios, F., Moisés, & Torres, S., Alan Iván. (2014). *Diseño y aplicación para uso de OBD II con interfaz vía bluetooth a dispositivo móvil*. México DF: IPN-UPIITA.
- [12] Sánchez, J. (Agosto de 2017). *Simulador de una ECU y diagnostico mediante CAN y OBD-II*. [ruidera](https://ruidera.uclm.es/xmlui/bitstream/handle/10578/15618/TFG%20Jorge%20Sanchez%20Carrizo.pdf?sequence=1&isAllowed=y). Recuperado el 19 de Noviembre de 2019, de <https://ruidera.uclm.es/xmlui/bitstream/handle/10578/15618/TFG%20Jorge%20Sanchez%20Carrizo.pdf?sequence=1&isAllowed=y>

- [13] Sauras, Á. (18 de Junio de 2018). *Que es y como funciona el head up display.* autofacil. Recuperado el 3 de Diciembre de 2019, de <https://www.autofacil.es/tecnologia/2018/06/18/funciona-head-up-display/44799.html>
- [14] Scanner Automotriz. (2019). *Analisis del ELM 327.* scannerautomotriz. Recuperado el 29 de Noviembre de 2019, de <https://scannerautomotriz.info/elm327/>
- [15] Soler, Á. (2018). *Todo lo que debes saber sobre el head up display.* sport. Recuperado el 5 de Diciembre de 2019, de <https://www.sport.es/es/motor/destacados/noticias/trending/head-up-display-que-es-como-funciona-6585510>
- [16] Tu optometrista. (2017). *visión y velocidad en conducción.* tuoptometrista. Recuperado el 5 de Diciembre de 2019, de <https://www.tuoptometrista.com/vision-y-conduccion/conduccion-y-velocidad/>
- [17] Villén, J. (2016). *Simulador de la ECU de un vehiculo con protocolo ISO 9141-2.* bibing. Recuperado el 19 de Noviembre de 2019, de <http://bibing.us.es/proyectos/abreproy/90766/fichero/Simulador+de+la+ECU+de+un+veh%EDculo+sobre+OBDII+con+protocolo+ISO+9141-2.pdf>
- [18] Wikipedia. (6 de Septiembre de 2019). *Unidad de control de motor.* Wikipedia. Recuperado el 19 de Noviembre de 2019, de https://es.wikipedia.org/wiki/Unidad_de_control_de_motor
- [19] Wikipedia. (29 de Julio de 2019). *Bus CAN.* Wikipedia. Recuperado el 19 de Noviembre de 2019, de https://es.wikipedia.org/wiki/Bus_CAN
- [20] Wikipedia. (10 de Septiembre de 2019). *OBD II PID.* Wikipedia. Recuperado el 29 de Noviembre de 2019, de https://es.wikipedia.org/wiki/OBD-II_PID
- [21] Wikipedia. (13 de Noviembre de 2019). *OBD.* Wikipedia. Recuperado el 18 de Noviembre de 2019, de <https://es.wikipedia.org/wiki/OBD>

ANEXO A. LISTA SAE DE PIDs

PID (HEX)	Descripción	Valor mínimo	Valor máximo	Unidad
00	PIDs implementados			
01	Estado de los monitores de diagnóstico desde que se borraron los códigos de fallas DTC; incluye el estado de la luz indicadora de fallas, MIL, y la cantidad de códigos de fallas DTC			
02	Almacena los códigos de fallas de diagnóstico DTC de un evento			
03	Estado del sistema de combustible			
04	Carga calculada del motor	0	100	%
05	Temperatura del líquido de enfriamiento del motor	-40	215	°C
06	Ajuste de combustible a corto plazo -Banco 1	-100 (Reducción de combustible: muy rico)	99.2 (Aumento de combustible: muy magro)	%
07	Ajuste de combustible a largo plazo -Banco 1			
08	Ajuste de combustible a corto plazo -Banco 2			
09	Ajuste de combustible a largo plazo -Banco 2			
0A	Presión del combustible	0	765	kPa
0B	Presión absoluta del colector de admisión	0	255	kPa
0C	RPM del motor	0	16,383.75	rpm
0D	Velocidad del vehículo	0	255	Km/h
0E	Avance del tiempo	-64	63.5	° ante TDC
0F	Temperatura del aire del colector de admisión	-40	215	°C
10	Velocidad del flujo del aire MAF	0	655.35	gr/sec
11	Posición del acelerador	0	100	%
12	Estado del aire secundario controlado			
13	Presencia de sensores de oxígeno (en 2 bancos)			
14	Sensor de oxígeno 1 A: Voltaje B: Ajuste de combustible a corto plazo	0 -100	1.275 99.2	voltios %
15	Sensor de oxígeno 2 A: Voltaje B: Ajuste de combustible a corto plazo			
16	Sensor de oxígeno 3 A: Voltaje B: Ajuste de combustible a corto plazo			
17	Sensor de oxígeno A: Voltaje B: Ajuste de combustible a corto plazo			
18	Sensor de oxígeno 5 A: Voltaje B: Ajuste de combustible a corto plazo			
19	Sensor de oxígeno 6 A: Voltaje B: Ajuste de combustible a corto plazo			
1A	Sensor de oxígeno 7 A: Voltaje B: Ajuste de combustible a corto plazo			
1B	Sensor de oxígeno 8 A: Voltaje B: Ajuste de combustible a corto plazo			
1C	Estándar OBD implementado en este vehículo			
1D	Sensores de oxígenos presentes en el banco 4			
1E	Estado de las entradas auxiliares			
1F	Tiempo desde que se puso en marcha el motor	0	65,535	sec
20	PID implementados [21 - 40]			
21	Distancia recorrida con la luz indicadora de falla (Malfunction Indicator Lamp, MIL) encendida	0	65,535	km
22	Presión del tren de combustible, relativa al colector de vacío	0	5177.265	kPa
23	Presión del medidor del tren de combustible (Diesel o inyección)	0	655,350	kPa

PID (HEX)	Descripción	Valor mínimo	Valor máximo	Unidad
	directa de gasolina)			
24	Sensor de oxígeno 1 AB: Relación equivalente de combustible - aire CD: Voltaje	0 0	< 2 < 8	prop. V
25	Sensor de oxígeno 2 AB: Relación equivalente de combustible - aire CD: Voltaje			
26	Sensor de oxígeno 3 AB: Relación equivalente de combustible - aire CD: Voltaje			
27	Sensor de oxígeno 4 AB: Relación equivalente de combustible - aire CD: Voltaje			
28	Sensor de oxígeno 5 AB: Relación equivalente de combustible - aire CD: Voltaje			
29	Sensor de oxígeno 6 AB: Relación equivalente de combustible - aire CD: Voltaje			
2A	Sensor de oxígeno 7 AB: Relación equivalente de combustible - aire CD: Voltaje			
2B	Sensor de oxígeno 8 AB: Relación equivalente de combustible - aire CD: Voltaje			
2C	EGR comandado			
2D	falla EGR	-100	99.2	%
2E	Purga evaporativa comandada	0	100	%
2F	Nivel de entrada del tanque de combustible	0	100	%
30	Cantidad de calentamientos desde que se borraron los fallas	0	255	cuenta
31	Distancia recorrida desde que se borraron los fallas	0	65,535	km
32	Presión de vapor del sistema evaporativo	-8,192	8191.75	Pa
33	Presión barométrica absoluta	0	255	kPa
34	Sensor de oxígeno 1 AB: Relación equivalente de combustible - aire CD: Actual	0 -128	< 2 <128	prop. mA
35	Sensor de oxígeno 2 AB: Relación equivalente de combustible - aire CD: Actual			
36	Sensor de oxígeno 3 AB: Relación equivalente de combustible - aire CD: Actual			
37	Sensor de oxígeno 4 AB: Relación equivalente de combustible - aire CD: Actual			
38	Sensor de oxígeno 5 AB: Relación equivalente de combustible - aire CD: Actual			
39	Sensor de oxígeno 6 AB: Relación equivalente de combustible - aire CD: Actual			
3A	Sensor de oxígeno 7 AB: Relación equivalente de combustible - aire CD: Actual			
3B	Sensor de oxígeno 8 AB: Relación equivalente de combustible - aire CD: Actual			
3C	Temperatura del catalizador: Banco 1, Sensor 1	-40	6,513.5	°C
3D	Temperatura del catalizador: Banco 1, Sensor 1			
3E	Temperatura del catalizador: Banco 1, Sensor 2			
3F	Temperatura del catalizador: Banco 2, Sensor 2			
40	PID implementados [41 - 60]			

PID (HEX)	Descripción	Valor mínimo	Valor máximo	Unidad
41	Estado de los monitores en este ciclo de manejo			
42	Voltaje del módulo de control	0	65.535	V
43	Valor absoluto de carga	0	25,700	%
44	Relación equivalente comandada de combustible - aire	0	< 2	prop.
45	Posición relativa del acelerador	0	100	%
46	Temperatura del aire ambiental	-40	215	°C
47	Posición absoluta del acelerador B	0	100	%
48	Posición absoluta del acelerador C			
49	Posición del pedal acelerador D			
4A	Posición del pedal acelerador E			
4B	Posición del pedal acelerador F			
4C	Actuador comandando del acelerador			
4D	Tiempo transcurrido con MIL encendido	0	65,535	min
4E	Tiempo transcurrido desde que se borraron los códigos de fallas			
4F	Valor máximo de la relación de equivalencia de combustible - aire, voltaje del sensor de oxígenos, corriente del sensor de oxígenos y presión absoluta del colector de entrada	0, 0, 0, 0	255, 255, 255, 2550	prop., V, mA, kPa
50	Valor máximo de la velocidad de flujo de aire del sensor de flujo de aire masivo	0	2550	g/s
51	Tipo de combustible			
52	Porcentaje de combustible etanol	0	100	%
53	Presión absoluta del vapor del sistema de evaporación	0	327.675	kPa
54	Presión del vapor del sistema de evaporación	-32,767	32,768	Pa
55	Ajuste del sensor de oxígeno secundario de plazo corto. A: banco 1. B: banco 3	-100	99.2	%
56	Ajuste del sensor de oxígeno secundario de plazo largo. A: banco 1. B: banco 3			
57	Ajuste del sensor de oxígeno secundario de plazo corto. A: banco 2. B: banco 4			
58	Ajuste del sensor de oxígeno secundario de plazo largo. A: banco 2. B: banco 4			
59	Presión absoluta del tren de combustible	0	655,350	kPa
5A	Posición relativa del pedal del acelerador	0	100	%
5B	Tiempo de vida del banco de baterías híbridas	0	100	%
5C	Temperatura del aceite del motor	-40	210	°C
5D	Sincronización de la inyección de combustible	-210.00	301.992	°
5E	Velocidad del combustible del motor	0	3276.75	L/h
5F	Requisitos de emisiones para los que el vehículo fue diseñado			
60	PID implementados [61 - 80]			
61	Porcentaje de torque solicitado por el conductor	-125	125	%
62	Porcentaje de torque actual del motor	-125	125	%
63	Torque de referencia del motor	0	65,535	Nm
64	Datos del porcentaje de torque del motor	-125	125	%
65	Entrada / salida auxiliar implementada			
66	Sensor de flujo de aire masivo			
67	Temperatura del enfriador del motor			
68	Sensor de temperatura de aire de entrada			
69	EGR comandado y falla de EGR			
6A	Control comandado del flujo de aire de entrada de Diesel y posición relativa de la entrada del flujo de aire			
6B	Temperatura de recirculación del gas del escape			
6C	Control comandado del actuador del acelerador y posición relativa del acelerador			
6D	Sistema de control de presión del combustible			

PID (HEX)	Descripción	Valor mínimo	Valor máximo	Unidad
6E	Sistema de control de presión de inyección			
6F	Presión de entrada del compresor del turbocargador			
70	Control de presión de aumento			
71	Control del turbo de geometría variable (Variable Geometry Turbo, VGT)			
72	Control de la compuerta de desperdicio			
73	Presión del escape			
74	RPM del turbocargador			
75	Temperatura del turbocargador			
76	Temperatura del turbocargador			
77	Temperatura del enfriador del aire de carga (Charge Air Cooler Temperature, CACT)			
78	Temperatura del gas del escape (Exhaust Gas Temperature, EGT) Banco 1			
79	Temperatura del gas del escape (Exhaust Gas Temperature, EGT) Banco 2			
7A	Filtro de partículas Diesel (Diesel Particulate Filter, DPF)			
7B	Filtro de partículas Diesel (Diesel Particulate Filter, DPF)			
7C	Temperatura del filtro de partículas Diesel (Diesel Particulate Filter, DPF)			
7D	Estado del área de control NOx NTE			
7E	Estado del área de control PM NTE			
7F	Tiempo que el motor ha estado en marcha			
80	PID implementados [81 - A0]			
81	Tiempo de marcha del motor para el dispositivo auxiliar de control de emisiones (Auxiliary Emissions Control Device, AECD)			
82	Tiempo de marcha del motor para el dispositivo auxiliar de control de emisiones (Auxiliary Emissions Control Device, AECD)			
83	Sensor de NOx			
84	Temperatura de superficie del colector			
85	Sistema reactivo NOx			
86	Sensor de partículas (Particular Matter, PM)			
87	Presión absoluta del colector de admisión			
A0	PID implementados [A1 - C0]			
C0	PID implementados [C1 - E0]			
C3	?	?	?	?
C4	?	?	?	?

ANEXO B. VEHÍCULOS COMPATIBLES CON EL ESCÁNER ELM327

Marca	Modelo	Año	Protocolo
	146	2006	KWP FAST
	147	2007	KWP FAST
	156	2003	ISO 9141
	159	2009	KWP FAST
	166	2004	KWP FAST
	Brera	2006	KWP FAST
	Giulietta	2010	CAN 29bit 500kb
	Gt	2004	KWP FAST
	Mito	2008	CAN 29bit 500kb
	Spider	2007	KWP FAST
	A1	2012	CAN 11bit 500kb
	A2	2003	ISO 9141
	A3	2006	KWP SLOW
	A4	2005	CAN 11bit 500kb
	A5	2012	CAN 11bit 500kb
	A6	2000	ISO 9141
	A8	2001	KWP SLOW
	Q5	2010	CAN 11bit 500kb
	Q7	2007	CAN 11bit 500kb
	S3	2007	CAN 11bit 500kb
	S4	2003	KWP SLOW
	S6	2001	ISO 9141
S8	1998	ISO 9141	
	M3	2007	CAN 11bit 500kb
	M5	2000	ISO 9141
	M6	2005	KWP FAST
	Serie 1 116d	2009	CAN 11bit 500kb
	Serie 3 318i	2008	KWP FAST
	Serie 5 520i	2004	KWP FAST
	Serie 6 645i	2004	KWP FAST
	Serie 7 730	2003	KWP FAST
	X1	2011	CAN 11bit 500kb
	X3	2009	CAN 11bit 500kb

Marca	Modelo	Año	Protocolo
	X5	2002	KWP FAST
	Z3	1999	ISO 9141
	Z4	2006	KWP FAST
	Aveo	2005	KWP FAST
	Blazer	2000	VPN
	Camaro	2012	CAN 11bit 500Kb
	Cavalier	2003	VPN
	Corsa	2012	KWP FAST
	Malibu	2000	VPN
	Matiz	2009	KWP FAST
	Spark	2012	KWP FAST
	Suburban	2003	VPN
	Volt	2012	CAN 11bit 500Kb
	300	2003	CAN 11bit 500Kb
	Crossfire	2004	KWP FAST
	Neon Le	2002	VPN
	Pacifica	2008	VPN
	Pt Cruiser	2002	ISO 9141
	Pt Touring	2001	VPN
	Sebring	2007	CAN 11bit 500Kb
	Stratus	2000	ISO 9141
	Town & Country	2002	VPN
	Voyager	2010	CAN 11bit 500Kb
	Avenger	2010	CAN 11bit 500kb
	Caliber	2006	CAN 11bit 500kb
	Caravan	2007	VPN
	Charger	2008	CAN 11bit 500kb
	Dakota	1999	ISO 9141
	Durango	2004	CAN 11bit 500Kb
	Grand Caravan	2008	CAN 11bit 500Kb
	Journey	2008	CAN 11bit 500Kb
	Nitro	2007	CAN 11bit 500Kb
	Ram 1500	2002	VPN
	Ram Slt	2001	ISO 9141
Ram 2500	2005	VPN	
	Albea	2005	KWP FAST

Marca	Modelo	Año	Protocolo
	Brava	1998	ISO 9141
	Croma	2006	KWP FAST
	Doblo	2010	CAN 29bit 500kb
	Ducato	2007	KWP FAST
	Grande Punto	2006	ISO CAN 29bit 500K
	Idea	2004	KWP FAST
	Marea	2000	KWP FAST
	Palio	2003	KWP FAST
	Punto	2011	CAN 29bit 500kb
	Scudo	2012	CAN 11bit 500kb
	Stilo	2003	KWP FAST
	Ulysse	2007	CAN 11bit 500kb
	Ba Falcon	2007	CAN 11bit 500kb
	Contour	1996	PWM
	Cougar	1999	PWM
	Escort	2001	PWM
	Explorer	1998	PWM
	F150	2006	CAN 11bit 500kb
	F250	2001	CAN 11bit 500kb
	Fiesta	2013	CAN 11bit 500kb
	Focus	2011	CAN 11bit 500kb
	Fusion	2009	CAN 11bit 500kb
	Galaxy	2007	CAN 11bit 500kb
	GT	2006	CAN 11bit 500kb
	Mustang	2006	CAN 11bit 500kb
	Ranger	2012	CAN 11bit 500kb
 HONDA	Accord	2006	ISO 9141
	Civic	2012	CAN 29bit 500kb
	Cr-v	2007	CAN 29bit 500kb
	Cr-z	2010	CAN 29bit 500kb
	Fit	2010	CAN 29bit 500kb
	Fr-v	2006	ISO 9141
	Hr-v	2001	ISO 9141
	Jazz	2005	ISO 9141
	Legend	2006	ISO 9141
	S2000	2000	ISO 9141

Marca	Modelo	Año	Protocolo
 HYUNDAI	Accent	2007	CAN 11bit 500kb
	Atos	2004	ISO 9141
	Coupe	2005	KWP FAST
	Elanta	2004	ISO 9141
	Getz	2006	ISO 9141
	H11	2011	CAN 11bit 500kb
	I10	2011	CAN 11bit 500kb
	I20	2011	CAN 11bit 500kb
	I30	2010	CAN 11bit 500kb
	Ix20	2011	CAN 11bit 500kb
	Ix35	2013	CAN 11bit 500kb
	Matrix	2007	ISO 9141
	Santa Fe	2011	CAN 11bit 500kb
Tucson	2009	ISO 9141	
 ISUZU	D-max	2009	CAN 11bit 500kb
	Kb250	2011	CAN 11bit 500kb
	N-serie	2008	CAN 11bit 500kb
	Rodeo	1996	VPN
	Trooper	2000	VPN
 JAGUAR	S-type	2006	CAN 11bit 500kb
	Xf	2008	CAN 11bit 500kb
	Xj6 X300	1996	ISO 9141
	Xj8	2001	ISO 9141
	Xkr	2007	CAN 11bit 500kb
 Jeep	Comander	2007	CAN 11bit 500kb
	Compass	2006	CAN 11bit 500kb
	Cherokee	2006	CAN 11 bit 500kb
	Laredo	2003	VPN
	Liberty	2008	CAN 11bit 500kb
	Patriot	2009	CAN 11bit 500kb
	Wrangler	2011	CAN 11bit 500kb
 KIA	Carens	2007	CAN 11bit 500kb
	Carnival	2006	CAN 11bit 500kb
	Ceed	2009	CAN 11 bit 500kb
	Cerato	2012	CAN 11bit 500kb
	Magentis	2002	KWP FAST

Marca	Modelo	Año	Protocolo
	Mohave	2008	CAN 11bit 500kb
	Picanto	2007	ISO 9141
	Sedona	2006	KWP FAST
	Sorento	2008	CAN 11bit 500kb
	Soul	2010	CAN 11bit 500kb
	Sportage	2012	CAN 11bit 500kb
	Venga	2013	CAN 11bit 500kb
	Defender 110	2007	CAN 11bit 500kb
	Discovery	2011	CAN 11bit 500kb
	Freelander	2003	ISO 9141
	Range Rover	2005	CAN 29bit 500kb
 MASERATI	3200 Gt	2001	KWP FAST
	Gransport	2006	ISO 9141
	Quattroporte	2007	CAN 11bit 500kb
	Spyder	2002	ISO 9141
 MAZDA	Mazda 2	2008	CAN 11bit 500kb
	Mazda 3	2008	CAN 11bit 500kb
	Mazda 323	2001	ISO 9141
	Mazda 5	2011	CAN 11bit 500kb
	Mazda 6	2012	CAN 11bit 500kb
	Mazda Bt50	2007	CAN 11bit 500kb
	Mazda Mpv	2000	ISO 9141
	Mazda Mx-5	2008	CAN 11bit 500kb
	Mazda Protege	2000	ISO 9141
	Mazda Rx8	2003	CAN 11bit 500kb
	Mazda Tribute	2008	CAN 11bit 500kb
 Mercedes-Benz	Clase A	2010	CAN 11bit 500kb
	Clase B	2011	CAN 11bit 500kb
	Clase C	2008	CAN 11bit 500kb
	Clase E	2007	CAN 11bit 500kb
	Clase S	2008	CAN 11bit 500kb
	Clk	2008	KWP FAST
	Cls	2006	CAN 11bit 500kb
	Glk	2010	CAN 11bit 500kb
	MI	2006	CAN 11bit 500kb
	Slk	2004	CAN 11bit 500kb

Marca	Modelo	Año	Protocolo
	Sprinter	2010	CAN 11bit 500kb
	Vaneo	2004	ISO 9141
	Viano	2007	KWP FAST
	Vito	2012	CAN 11bit 500kb
	Clubman	2007	CAN 11bit 500kb
	Cooper	2008	CAN 11bit 500kb
	One	2007	CAN 11bit 500kb
	Asx	2010	CAN 11bit 500kb
	Canter	2003	CAN 29bit 250kb
	Carisma	2001	KWP SLOW
	Colt	2005	CAN 11bit 500kb
	Eclipse	1997	ISO 9141
	Grandis	2006	CAN 11bit 500kb
	L200	2006	CAN 11bit 500kb
	Lancer	2009	CAN 11bit 500kb
	Montero	2001	ISO 9141
	Outlander	2011	KWP FAST
	Pajero	2006	CAN 11bit 500kb
	Space Star	2013	CAN 11bit 500kb
	350z	2006	ISO 9141
	Almera	2003	KWP FAST
	Altima	2006	ISO 9141
	Infiniti I30	2001	ISO 9141
	Interstar	2003	KWP FAST
	Juke	2013	CAN 11bit 500kb
	Kubistar	2007	KWP FAST
	Micra	2008	ISO 9141
	Murano	2006	ISO 9141
	Navara	2011	CAN 11bit 500kb
	Note	2008	CAN 11bit 500kb
	Nv200	2010	CAN 11bit 500kb
	Pathfinder	2006	ISO 9141
	Patrol	2008	CAN 11bit 500kb
	Platina	2005	KWP SLOW
	Primastar	2006	KWP FAST
	Primera	2004	KWP FAST
Qashqai	2012	CAN 11bit 500kb	

Marca	Modelo	Año	Protocolo
	Sentra	2001	ISO 9141
	Terrano	2002	ISO 9141
	Tida	2004	ISO 9141
	Tidan	2010	CAN 11bit 500kb
	Versa	2010	CAN 11bit 500kb
	X-trail	2008	CAN 11bit 500kb
	Agila	2008	CAN 11bit 500kb
	Ampera	2012	CAN 11bit 500kb
	Antara	2007	CAN 11bit 500kb
	Astra	2012	CAN 11bit 500kb
	Combo	2009	KWP FAST
	Corsa	2012	CAN 11bit 500kb
	Frontera	2003	ISO 9141
	Gt	2007	CAN 11bit 500kb
	Insignia	2011	CAN 11bit 500kb
	Meriva	2011	CAN 11bit 500kb
	Movano	2012	CAN 11bit 500kb
	Omega	2003	ISO 9141
	Signum	2003	CAN 11bit 500kb
	Suntra	1999	ISO 9141
	Speedster	2003	KWP FAST
	Trigra	2005	KWP FAST
	Vectra	2010	KWP FAST
	Vivaro	2011	KWP FAST
Zafira	2009	CAN 11bit 500kb	
 PEUGEOT	1007	2008	CAN 11bit 500kb
	106	2003	KWP SLOW
	107	2010	CAN 11bit 500kb
	206	2008	CAN 11bit 500kb
	207	2011	CAN 11bit 500kb
	208	2013	CAN 11bit 500kb
	3008	2013	CAN 11bit 500kb
	301	2013	CAN 11bit 500kb
	306	2001	KWP FAST
	307	2010	CAN 11bit 500kb
	308	2013	CAN 11bit 500kb

Marca	Modelo	Año	Protocolo
	4007	2007	CAN 11bit 500kb
	406	2004	KWP FAST
	407	2007	KWP FAST
	5008	2011	CAN 11bit 500kb
	508	2012	CAN 11bit 500kb
	607	2007	KWP FAST
	806	2001	KWP SLOW
	807	2007	CAN 11bit 500kb
	Bipper	2009	CAN 11bit 500kb
	Boxer	2007	KWP FAST
	Expert	2008	CAN 11bit 500kb
	Partner	2011	CAN 11bit 500kb
	Rcz	2012	CAN 11bit 500kb
	Tepee	2008	CAN 11bit 500kb
	 PONTIAC	Firebird	1996
G5		2007	CAN 11bit 500kb
Grand Prix		2005	VPN
Montana		2008	CAN 11bit 500kb
Sunfire		2002	VPN
 RENAULT	Logan	2005	KWP2000 SLOW
	Avantime	2002	KWP FAST
	Captur	2013	CAN 11bit 500kb
	Clio	2014	CAN 11bit 500kb
	Espace	2008	CAN 11bit 500kb
	Fluence	2010	CAN 11bit 500kb
	Kangoo	2012	CAN 11bit 500kb
	Koleos	2009	CAN 11bit 500kb
	Laguna	2010	CAN 11bit 500kb
	Latitude	2011	CAN 11bit 500kb
	Mascott	2004	KWP FAST
	Master	2012	CAN 11bit 500kb
	Megane	2011	CAN 11bit 500kb
	Modus	2009	CAN 11bit 500kb
	Safrane	1998	ISO 9141
	Scenic	2013	CAN 11bit 500kb
	Symbol	2010	KWP SLOW
Thalia	2004	KWP SLOW	

Marca	Modelo	Año	Protocolo
	Trafic	2012	CAN 11bit 250kb
	Twingo	2013	CAN 11bit 500kb
	Vel Satis	2006	CAN 11bit 250kb
	Alhambra	2010	CAN 11bit 500kb
	Altea	2011	CAN 11bit 500kb
	Arosa	2002	ISO 9141
	Cordoba	2009	ISO 9141
	Exeo	2012	CAN 11bit 500kb
	Ibiza	2011	CAN 11bit 500kb
	Leon	2012	CAN 11bit 500kb
	Toledo	2005	KWP SLOW
	Forester	2009	CAN 11bit 500kb
	Impreza	2009	CAN 11bit 500kb
	Legacy	2009	CAN 11bit 500kb
	Outback	2007	CAN 11bit 500kb
	Alto	2004	KWP FAST
	Grand Vitara	2007	KWP FAST
	Ignis	2004	KWP FAST
	Jimny	2006	KWP FAST
	Liana	2004	KWP FAST
	Samurai	2003	KWP FAST
	Splash	2009	CAN 11bit 500kb
	Swift	2012	CAN 11bit 500kb
	Vitara	2005	KWP FAST
	Wagon R	2004	KWP FAST
	XI-7	2005	KWP FAST
Sx4	2011	KWP FAST	
	Auris	2010	CAN 11bit 500kb
	Avanza	2012	CAN 11bit 500kb
	Avensis	2006	KWP SLOW
	Aygo	2008	ISO 9141
	Camry	2012	CAN 11bit 500kb
	Celica	1999	ISO 9141
	Corolla	2009	CAN 11bit 500kb
	Echo	2003	ISO 9141
	Fj Cruiser	2007	CAN 11bit 500kb

Marca	Modelo	Año	Protocolo
	Gt86	2012	CAN 11bit 500kb
	Hilux	2009	KWP SLOW
	Iq	2010	CAN 11bit 500kb
	Land Cruiser	2012	CAN 11bit 500kb
	Matrix	2010	CAN 11bit 500kb
	Mr-s	2000	ISO 9141
	Mr2	2000	ISO 9141
	Previa	2000	ISO 9141
	Prius	2009	CAN 11bit 500kb
	Rav4	2009	CAN 11bit 500kb
	Sienna	2010	CAN 11bit 500kb
	Tacoma	2006	KWP FAST
	Tercel	1999	ISO 9141
	Tundra	2003	ISO 9141
	Yaris	2012	CAN 11bit 500kb
	Amarok	2010	CAN 11bit 500kb
	Bora	2002	ISO 9141
	Caddy	2011	CAN 11bit 500kb
	California	2007	KWP SLOW
	Crafter	2011	CAN 11bit 500kb
	Eos	2007	CAN 11bit 500kb
	Fox	2007	ISO 9141
	Golf	2011	CAN 11bit 500kb
	Jetta	2007	KWP SLOW
	Lupo	2002	ISO 9141
	New Beetle	2003	ISO 9141
	Passat	2012	CAN 11bit 500kb
	Phaeton	2004	ISO 9141
	Pointer	2001	ISO 9141
	Polo	2011	KWP SLOW
	Scirocco	2009	CAN 11bit 500kb
	Sharan	2007	KWP SLOW
	T4	2001	ISO 9141
	T5	2007	KWP SLOW
	Tiguan	2012	CAN 11bit 500kb
Touareg	2008	CAN 11bit 500kb	

Marca	Modelo	Año	Protocolo
	Touran	2012	CAN 11bit 500kb
	Transporter	2011	CAN 11bit 500kb
	Up	2012	CAN 11bit 500kb
	850	1996	ISO 9141
	C30	2008	CAN 29bit 500kb
	C70	2007	CAN 11bit 500kb
	S40	2004	CAN 29bit 500kb
	S60	2007	CAN 29bit 500kb
	S80	2008	CAN 11bit 500kb
	V40	1999	ISO 9141
	V50	2009	CAN 29bit 500kb
	V60	2010	KWP FAST
	V70	2007	CAN 29bit 500kb
	Xc60	2011	CAN 11bit 500kb
	Xc70	2008	CAN 11bit 500kb
	Xc90	2007	CAN 29bit 500kb

ANEXO C. HOJA DE DATOS DEL FILM DE RETROPROYECCIÓN

RearProjection transparent **ASLAN RP 36**

Self-adhesive transparent rear projection film for glass surfaces

RearProjection transparent ASLAN RP 36 is a self-adhesive film that creates rear projection surfaces that can be operated with a projector. The film is so transparent that backgrounds, for instance products exhibited behind a shop window, remain visible at all times, while projections are of brilliant reproduction quality. ASLAN RP 36 is designed for application on any glass surface that is supposed to also serve as a projection screen, such as shop windows or glass panels in restaurants, malls, exhibitions, offices etc. Give your transparent surfaces an additional function and use them as a display screen for advertisement and/or broadcasts of sport events. The rear projection film is also suitable for interactive projections.

For additional information, please contact our technical advisory service:
+49 22 04. 7 08-80

Construction

face-film:	polyester (with special coating)	
thickness:	~ 50 µm	
adhesive:	pressure sensitive polyacrylate adhesive	square quantity: ~ 30 g/m ²
release liner:	siliconised polyester	square weight: ~ 75 µm

Characteristics

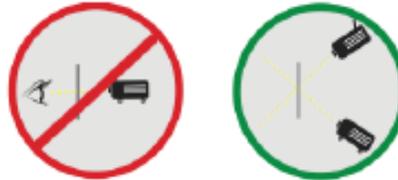
adhesive strength: (ASTMD-903)	immediately after 1 week	~ 6.0 N/25 mm ~ 14.0 N/25 mm
dimensional stability:	applied onto aluminum after 48 hours stored at 70 °C (25 x 25 cm)	max. -0.05 %
light transmission:	DIN 53 147	approx. 88 %
temperature:	application temperature: service temperature range:	min. 15 °C (dry application) min. 20 °C (wet application) -30 °C up to +80 °C
durability:	up to 5 years outdoors, with vertical exposure, in central European standard climatic conditions	
viewing angle:	+/- 80 °	

projection:

The RearProjection transparent ASLAN RP 36 film can be used with projectors, such as mobile projectors, installation projectors, short distance projectors, full HD projectors and interactive projectors.

To achieve optimum rear projection results, the following points are to be considered:

- The projector needs to have an image reversal option.
- The projection result, as with all rear projections, relates to the lighting conditions of the surroundings. In general, a high lumen projector is recommended (minimum 5,000 ANSI lumens in daylight). Direct sunlight onto the projection surface is to be avoided.



digital cutting:

The self-adhesive film is ideally suited for cutting. The vertical height of capital letters should not be smaller than 5 mm. For the application of lettering etc. we recommend using the application tape LowTackTape ASLAN TP 110.

application:

When installing the film with a flatbed applicator, the film is to be applied dry. In case of a manual installation, we recommend applying the film wet. However, a wet application is only possible to a limited extent, in consideration and compliance of our separate processing instructions for wet applications. The tape should be taken off from the middle of the material to the edge.

RearProjection transparent ASLAN RP 36 is exclusively suited for smooth, even, non-elastic and non-curved surfaces. The film must not be applied on outgassing surfaces like non-tempered acrylic glass, since bubbles might occur, particularly under short-term exposure to heat.

storage life:

Before application, the films can be stored for minimum 3 years from date of production. The film must be stored at room temperature (15-25 °C) and a relative humidity of the air of 50-60 %. To avoid pressure points appearing on the roll surface, we recommend the rolls be stored either vertical standing or for this purpose designed 'hanging' racks.

State:

All technical data and advice is based on our experience and measured testing that we believe to be reliable. It remains the customer's responsibility to test the suitability of our products for the intended purpose.

The quality of our products is regularly examined, upgraded and developed. We take the right, without prior notice, to adjust, upgrade and improve the chemical structures or physical characteristics of our products in accordance with our latest knowledge.