

UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE POSTGRADO

MAESTRIA EN INGENIERIA AUTOMOTRIZ CON MENCIÓN EN NEGOCIOS AUTOMOTRICES



TEMA:

ELABORACIÓN DE UN DISPOSITIVO DE DIAGNÓSTICO OBD II CAPAZ DE INFILTRARSE EN TIEMPO REAL CON EL ALMACENAMIENTO DTC DE LA ECU MEDIANTE CONEXIÓN REMOTA.

Trabajo de Grado previo a la obtención del Título de Magíster en Ingeniería Automotriz con
Mención en Negocios Automotrices

AUTOR:

Ing. Byron Marcelo Puma Morales

DIRECTOR:

Ing. Andrés Felipe Cevallos González, MSc.

Ibarra, 2022

CERTIFICADO

ACEPTACIÓN DEL DIRECTOR

En mi calidad de director del plan de trabajo de grado, previo a la obtención del título de Magíster en Ingeniería Automotriz con Mención en Negocios Automotrices, nombrado por el Honorable Consejo Directivo de la Facultad de Posgrado.

CERTIFICO:

Que una vez analizado el plan de grado cuyo título es "ELABORACIÓN DE UN DISPOSITIVO DE DIAGNÓSTICO OBD II CAPAZ DE INFILTRARSE EN TIEMPO REAL CON EL ALMACENAMIENTO DE LA ECU MEDIANTE CONEXIÓN REMOTA", presentado por el Ingeniero Byron Marcelo Puma Morales con número de cédula 100377046-6, doy fe que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a presentación pública y evaluación por parte de los señores integrantes del jurado examinador que se designe.

En la ciudad de Ibarra, a los 02 días del mes mayo del 2022.

Atentamente,



Firmado digitalmente
por Andrés Cevallos
González
Fecha: 2022-07-07
16:50:05:00

Ing. Andrés Felipe Cevallos González, MSc.

DIRECTOR DEL TRABAJO DE GRADO



UNIVERSIDAD TÉCNICA DEL NORTE BIBLIOTECA UNIVERSITARIA

AUTORIZACION DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACION DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO	
CEDULA DE IDENTIDAD:	1003770466
APELLIDOS Y NOMBRES:	PUMA MORALES BYRON MARCELO
DIRECCIÓN:	CALLE SUCRE 12-72 Y OLMEDO (ATUNTAQUI)
EMAIL:	bpumamorales@gmail.com

DATOS DE LA OBRA	
TÍTULO:	“ELABORACIÓN DE UN DISPOSITIVO DE DIAGNÓSTICO OBD II CAPAZ DE INFILTRARSE EN TIEMPO REAL CON EL ALMACENAMIENTO DE LA ECU MEDIANTE CONEXIÓN REMOTA”
AUTORES:	Ing. Byron Marcelo Puma Morales
FECHA:	02 de mayo del 2022
PROGRAMA:	POSGRADO
TITULO POR EL QUE OPTA	MAGÍSTER EN INGENIERÍA AUTOMOTRIZ CON MENCIÓN EN NEGOCIOS AUTOMOTRICES
ASESOR/DIRECTOR	ING. ANDRÉS FELIPE CEVALLOS GONZALEZ, MSc.

2. CONSTANCIA

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros

En la ciudad de Ibarra, a los 07 días del mes julio del 2022.

EL AUTOR



Firmado electrónicamente por:

**BYRON MARCELO
PUMA MORALES**

Nombre: Ing. Byron Marcelo Puma Morales

DEDICATORIA

El presente trabajo es el resultado de mi esfuerzo y constancia y se lo dedico a mi madre porque sembró en mi la semilla del amor, la responsabilidad y honestidad, el deseo de superación siempre inculcando valores morales y espirituales.

A mi padre, hermanas y hermano, por brindarme su apoyo y consejos, convirtiéndose pilar fundamental de mi vida personal y profesional.

A mi esposa por su amor y cariño permanente en los momentos más difíciles, a mi hija quien es el motor de mi vida, mi principal motivación para no rendirme y llegar a ser el mejor ejemplo para ella.

A mi familia y amigos por su ayuda desinteresada, por brindarme su mano siempre que la necesité para el desarrollo de este proyecto, por permitirme aprender más de la vida a su lado.

AGRADECIMIENTO

Deseo expresar mis más sinceros agradecimientos a la Universidad Técnica del Norte, institución que me ha brindado sólidos conocimientos para desempeñarme en el campo profesional, así como, a todos los docentes del programa de Maestría en Ingeniería Automotriz con Mención en Negocios Automotrices, que han impartido su sabiduría y experiencias en el transcurso del camino, a toda mi familia por su apoyo incondicional.

Mis sinceros agradecimientos al Sr. Jorge Terán quien de manera desinteresada ha proporcionado su vehículo para la realización del presente trabajo de investigación.

Esto es posible gracias a ustedes.

Contenido

CERTIFICADO.....	2
AUTORIA DE RESPONSABILIDAD.....	3
DEDICATORIA	4
AGRADECIMIENTO	5
RESUMEN.....	10
ABSTRACT.....	11
1. CAPÍTULO I.....	12
1.1 El Problema.....	12
1.1.1 Planteamiento Del Problema	12
1.1.2 Antecedentes.....	12
1.1.3 Objetivos de la Investigación	13
1.1.3.1 Objetivo General.....	13
1.1.3.2 Objetivos Específicos.....	14
1.1.4 Justificación	14
1.1.5 Alcance.....	14
2. CAPÍTULO II.....	15
2.1 Marco Referencial	15
2.1.1 Marco Teórico.....	15
2.1.1.1 Unidad de Control Electrónico ECU.....	16
2.1.1.2 OBD II.....	18
2.1.1.3 Redes de Comunicación Multiplexadas.....	19
2.1.1.4 Red CAN.....	19
2.1.1.5 Protocolo de Comunicación CAN-Bus.....	19
2.1.1.6 Microcontroladores Programables.....	20
2.1.1.7 IFTTT, Internet de las Cosas (IOT)	21
2.1.1.8 Webhook	22
2.1.1.9 Almacenamiento Digital.....	22
3. CAPÍTULO III.....	24
3.1 Materiales y Métodos	24
3.1.1 Materiales.....	24
3.1.2 Marco Metodológico.....	26
3.1.2.1 Enfoque y Tipo de Investigación.....	26
3.1.2.2 Procedimientos.....	26
3.1.2.3 Flujo de Proceso.....	26

3.1.3	Evaluación de Criterios de Vehículo de Estudio.....	28
3.1.3.1	Procedimiento Visual de Verificación de la Red CAN.....	28
3.1.3.2	Procedimiento de Verificación de la Red CAN con Escáner Automotriz.....	29
3.1.3.3	Procedimiento con Osciloscopio Automotriz Para la Verificación de la Red CAN	30
3.1.4	Conexión en Paralelo de CAN Bus Analyzer y Hércules SETUP	30
3.1.5	Obtención de información	33
3.1.6	Diseño de prototipo de extracción de la información	33
3.1.6.1	Diseño de cable de conexión OBD II a DB9.....	33
3.1.6.2	Diseño de programación Arduino MEGA 2560.....	34
3.1.6.3	Diseño de conexión Sun-Flower CAN Shield V2.....	39
3.1.6.4	Diseño de conexión y programación modulo WIFI NODEMCU (ESP8266-12E).	40
3.1.6.5	Diseño del almacenamiento de datos.	45
4.	CAPITULO IV.	46
4.1	Resultados y Discusión	46
4.1.1	Evaluación de Vehículo	46
4.1.1.1	Resultado del procedimiento visual de verificación la Red CAN.....	46
4.1.1.2	Resultado del procedimiento de verificación de la Red CAN con escáner automotriz. 47	
4.1.1.3	Resultado del procedimiento con Osciloscopio Automotriz para la verificación de la Red CAN.....	48
4.1.2	Resultado de la Conexión en Paralelo de Software´s	49
4.1.2.1	Software Hércules SETUP.....	50
4.1.2.2	CAN Bus Analyzer.	51
4.1.3	Resultado de Obtención de la Información	51
4.1.4	Resultado de Prototipo para la Extracción de la Información	54
4.1.5	Resultado de Almacenamiento de la Información	56
5.	CONCLUSIONES Y RECOMENDACIONES	58
5.1	Conclusiones.....	58
5.2	Recomendaciones.....	58
6.	REFERENCIAS.	59

INDICE DE FIGURAS

Figura 1 <i>Esquema de elementos de un sistema de inyección electrónico</i>	15
Figura 2 <i>Unidad de Control Electrónico (ECU)</i>	16
Figura 3 <i>Gestión del Motor de Combustión Interna</i>	18
Figura 4 <i>Esquema de funcionamiento Webhooks - IFTTT</i>	22
Figura 5 <i>Diagrama para la Obtención de la Información.</i>	24
Figura 6 <i>Diagrama Referencial del Dispositivo de Diagnostico Remoto</i>	25
Figura 7 <i>Flujo de Procedimientos</i>	27
Figura 8 <i>Ubicación de Conector de Diagnostico</i>	28
Figura 9 <i>Conector de Diagnostico OBD II</i>	29
Figura 10 <i>Búsqueda de Protocolo de Comunicación</i>	29
Figura 11 <i>Conexión entre Osciloscopio y Conector de Diagnostico</i>	30
Figura 12 <i>Conexión en Paralelo de los Softwares Hércules SETUP y CAN Bus Analyzer</i>	31
Figura 13 <i>Solicitud de Información DTC</i>	32
Figura 14 <i>Parámetros en Tiempo Real</i>	33
Figura 15 <i>Diagrama de Conexión Cable OBD II – DB9</i>	34
Figura 16 <i>Distribución de Arduino Mega 2560</i>	34
Figura 17 <i>Funciones de Programación de Arduino</i>	35
Figura 18 <i>Programación Función de Configuración</i>	36
Figura 19 <i>Programación Función Bucle</i>	37
Figura 20 <i>Condiciones de Lectura de DTC</i>	38
Figura 21 <i>Condiciones de Lectura de DTC</i>	39
Figura 22 <i>Distribución Sunflower CAN Shield V2</i>	40
Figura 23 <i>Pines de Conexión del Módulo ESP8266</i>	40
Figura 24 <i>Bibliotecas de Conexión Wifi y Solicitud Web</i>	41
Figura 25 <i>Bibliotecas de Conexión Wifi y Solicitud Web</i>	42
Figura 26 <i>Comunicación Serial y Modo de Conexión STATION</i>	43
Figura 27 <i>Lectura de información y almacenamiento de datos</i>	44
Figura 28 <i>Proceso de Creación de Webhooks</i>	45
Figura 29 <i>Pines Energizados Red CAN</i>	47
Figura 30 <i>Protocolo de Comunicación CAN</i>	48
Figura 31 <i>Mediciones CAN H– CAN L</i>	49
Figura 32 <i>Conexión en paralelo de ordenadores</i>	49
Figura 33 <i>Interpretación de Mensaje de Respuesta</i>	50
Figura 34 <i>Respuesta a la Solicitud de Información de códigos DTC</i>	50
Figura 35 <i>Envío y Recepción de Información</i>	51
Figura 36 <i>Transformación de Valores Hexadecimales a Números Binarios</i>	53
Figura 37 <i>Norma ISO 15031 Aplicada a Data3 y Data 4</i>	53
Figura 38 <i>Cable de conexión OBD II – DB9</i>	54
Figura 39 <i>Acople de Tarjetas Electrónicas Arduino – CAN Bus Shield</i>	55
Figura 40 <i>Monitor Serie de Arduino – Código de Falla Detectado</i>	56
Figura 41 <i>Adaptación del Módulo Node MCU a Arduino</i>	56
Figura 42 <i>Correo enviado con Información DTC</i>	57
Figura 43 <i>Diagrama de Conexión Eléctrica</i>	61

INDICE DE TABLAS

Tabla 1 <i>Materiales Utilizados para el Desarrollo del Proyecto</i>	25
Tabla 2 <i>Pines Energizados Conector Kia Rio 2019</i>	46
Tabla 3 <i>Valores de Voltaje y Frecuencia Red CAN</i>	48
Tabla 4 <i>Información Requerida</i>	52

RESUMEN

Debido a la intranquilidad de los conductores al parpadear o encenderse la luz de alerta "Check Engine" en el tablero de control del vehículo, se desarrolló un dispositivo de diagnóstico OBD-II (On Board Diagnostics) con el propósito de acceder a los códigos de falla en tiempo real de manera remota. Se identificó la información de la Red CAN (Controller Area Network), esta información permitió diseñar el dispositivo para la extracción, envío y almacenamiento de datos, finalmente para el diagnóstico en tiempo real de manera remota se construyó una plataforma en la web. Se utilizó una tarjeta electrónica Arduino MEGA 250 donde se acopló la tarjeta electrónica CAN SHIELD V2 y el módulo NodeMCU quien fue relacionador directo de la plataforma IFTTT (IF This, Then That), los resultados evidencian la posibilidad de compartir mensajes de escritura y lectura con la ECU (Engine Control Unit) y mediante un correo electrónico se reflejó la capacidad de repuesta del dispositivo para procesar y enviar la información a la web de manera inmediata, por lo tanto, se concluyó que los datos determinantes para infiltrarse en los códigos DTC (Diagnostic Trouble Code) generados en la ECU es el ID de escritura (solicitud) y el ID de lectura (respuesta), por otro lado, se concluyó que la plataforma IFTTT cumple con las condiciones del dispositivo y del IOT (Internet Of Things) que ayudo a la interconexión de dispositivos a través de una red.

ABSTRACT

Due to the concern of drivers when the "Check Engine" warning light flashes or lights up on the vehicle's control panel, an OBD-II diagnostic device (On Board Diagnostics) was developed with the purpose of accessing the codes of the vehicle. fault in real time remotely. The information of the CAN Network (Controller Area Network) was identified, this information allowed to design the device for the extraction, sending and storage of data, finally for real-time diagnosis remotely, a platform was built on the web. An Arduino MEGA 250 electronic card was used where the CAN SHIELD V2 electronic card and the NodeMCU module were attached, which was a direct linker of the IFTTT platform (IF This, Then That), the results show the possibility of sharing writing and reading messages with the ECU (Engine Control Unit) and through an email, the response capacity of the device to process and send the information to the web immediately was reflected, therefore, it was concluded that the determining data to infiltrate the DTC codes (Diagnostic Trouble Code) generated in the ECU is the writing ID (request) and the reading ID (response), on the other hand, it was concluded that the IFTTT platform complies with the conditions of the device and the IOT (Internet Of Things) that I help the interconnection of devices through a network.

1. CAPÍTULO I.

1.1 El Problema

1.1.1 *Planteamiento Del Problema*

En la actualidad los vehículos son equipados con componentes de control electrónico para lograr un desempeño óptimo del vehículo controlando las emisiones contaminantes, las señales analógicas – digitales generadas por sensores y actuadores que conforma el sistema de control, son procesados por la Unidad de Control Electrónico (ECU) (Juan Trujillo-Tello et al., 2020).

En los talleres de servicio de mantenimiento automotriz el escáner automotriz se ha convertido en una herramienta importante para acceder a los códigos de falla que se generan cuando la ECU registra un mal funcionamiento de cualquier componente que esté vinculado a ella para su solución posterior (Abramowicz, 2017).

La lectura de códigos de falla generados por la ECU debido al funcionamiento erróneo o daño de cualquiera de sus componentes, se lo realiza por medio de un scanner automotriz y en presencia del vehículo, para el caso de los vehículos varados en carretera el acceso a los códigos de falla y parámetros de operación resulta complicado al no tener la presencia del vehículo; resulta favorable para el establecimiento y el usuario observar en tiempo real los códigos de falla y parámetros de operación sin importar la distancia brindándole un primer diagnóstico y asesoramiento para el auxilio del usuario.

1.1.2 *Antecedentes*

En 1970 Estados Unidos de América implemento un la ley de “AIRE LIMPIO”, que prohibió la importación y venta de vehículos que no cumplan los estrictos parámetros y estándares permitidos para las emisiones de gases contaminantes generados por motores de combustión interna (Agency United States Environmental, 2007).

Por consecuencia la mayoría de los fabricantes de automóviles en su afán de ingresar al mercado americano a lo largo del tiempo, han desarrollado una serie de dispositivos tecnológicos que permiten controlar los parámetros de funcionamiento del motor de combustión interna (MCI) favoreciendo su rendimiento, economía y sobre todo cumpliendo con las normativas gubernamentales para la preservación del medio ambiente.

(Agency United States Environmental, 2007) menciona que, “Los vehículos de motor son responsables de casi la mitad de los compuestos orgánicos volátiles (COV) que forman smog, más de la mitad de las emisiones de óxido de nitrógeno (NOx) y aproximadamente la mitad de las emisiones tóxicas de contaminantes atmosféricos en los Estados Unidos. Los vehículos a motor, incluidos los vehículos para uso fuera de carretera, ahora representan el 75 % de las emisiones de monóxido de carbono en todo el país”.

Las exigentes regulaciones dan origen al sistema de inyección electrónica de combustible que tiene por objetivo brindar un mejor rendimiento con más ahorro en todos sus parámetros de funcionamiento y generalmente se constituyen por sensores, actuadores y la unidad de control electrónico (ECU); en la actualidad y gracias a los desarrollos tecnológicos las ECU's de los vehículos disponen de diagnóstico a bordo que nos permite comunicarnos y visualizar varios parámetros de funcionamiento, mismos que al estar fuera de sus rangos pre establecidos, generan códigos de falla en su memoria y a su vez son alertados por medio del testigo o lampara de luz CHECK ENGINE o también conocida como LUZ MIL (On & Diagnostics, 2005).

En 1996, con el afán de reducir la contaminación, EEUU exige a los fabricantes de automóviles disponer un sistema de diagnóstico a bordo estableciendo el conocido OBD II, con lo que empresas dedicadas al desarrollo y fabricación de equipos de diagnósticos automotriz han conseguido sacar al mercado una amplia gama de herramientas de diagnóstico que ayudan a interactuar con la ECU, obteniendo códigos de falla, programaciones, parámetros de funcionamiento de sensores y actuadores, pruebas de sistemas, entre otras acciones que dependen de la marca, permitiendo que los usuarios asistan con sus vehículos a talleres y/o concesionarios para la solución de una falla alertada por la ECU, bajo este contexto es preciso contar con un dispositivo que permita al taller de servicio acceder a la información generada por la ECU de manera remota con la finalidad de salvaguardar la vida útil del motor (On & Diagnostics, 2005).

1.1.3 *Objetivos de la Investigación*

1.1.3.1 Objetivo General.

Desarrollar un dispositivo de diagnóstico OBDII que permita acceder en tiempo real al almacenamiento de los códigos de falla de la ECU mediante conexión remota.

1.1.3.2 Objetivos Específicos.

- Identificar los códigos de la Red CAN del vehículo referentes a los DTC.
- Diseñar un dispositivo de extracción y almacenamiento de códigos de falla.
- Construir una plataforma de acceso remoto online para el diagnóstico del vehículo.

1.1.4 Justificación

La ejecución del presente proyecto es desarrollar un dispositivo capaz de identificar los códigos DTC conocidos por sus siglas en español como códigos de problemas de diagnóstico los cuales se almacenan en la memoria de la ECU producto de la identificación de una serie de problemas que afectan al correcto funcionamiento de parámetros establecidos del motor, este dispositivo nos permitirá extraer y almacenar los códigos de falla generados en la ECU.

Los vehículos en su diario caminar están propensos a sufrir cualquier desperfecto mecánico en carretera y muchos de los usuarios desconocen de los procedimientos al encenderse presentarse un inconveniente; al encenderse la luz de alerta Check Engine, este desarrollo tecnológico permitirá vincular digitalmente el almacenamiento de datos en una nube de datos a través de internet, de tal manera que la información se pueda visualizar sin necesidad de la presencia del vehículo, lo cual ayudará al establecimiento de servicio a brindar un primer diagnóstico de lo acontecido en el vehículo, brindando así, una asistencia y asesoría remota de los daños originados para su posterior solución.

Por otro lado el presente desarrollo tecnológico se enfoca en la política 5.3 del Plan Nacional para el Buen Vivir que menciona “Promover la investigación, la formación, la capacitación, el desarrollo y la transferencia tecnológica, la innovación y el emprendimiento, en articulación con las necesidades sociales, para impulsar el cambio de la matriz productiva (Plan Nacional Del Buen Vivir 2017-2021, 2017).

1.1.5 Alcance

En el presente desarrollo tecnológico se elaborará un dispositivo capaz de detectar y leer códigos de falla a través de la red CAN en un vehículo con sistema de diagnóstico OBD II genérico, este dispositivo tendrá la capacidad de enviar la información a una nube de datos que nos permitirá visualizarla en tiempo real; los datos de comunicación y códigos de falla serán analizados para su transmisión posterior en la Web específicamente mediante correo electrónico.

2. CAPÍTULO II.

2.1 Marco Referencial

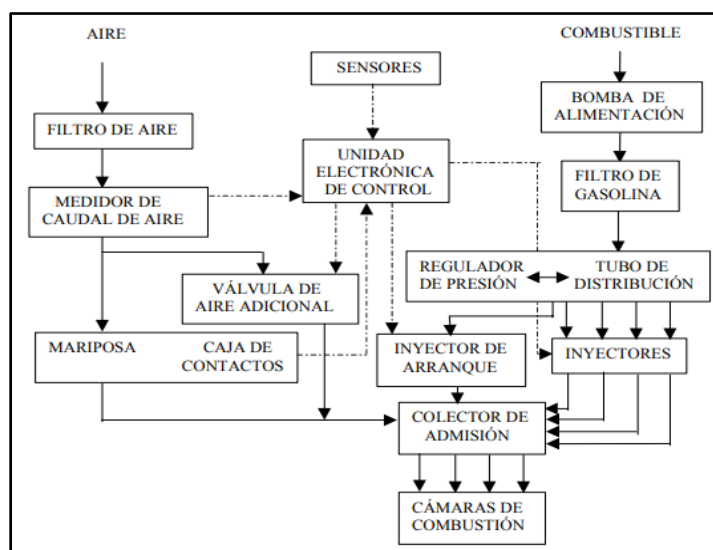
2.1.1 Marco Teórico

Las restricciones y regulaciones a la contaminación ambiental producto de las emisiones de gases de automóviles con motores de combustión interna dio origen a que los fabricantes del sector automotriz busquen alternativas para reducir el impacto ambiental dando un impulso a la innovación tecnología para el control electrónico de la combustión del motor (Pérez Darquea, 2018).

Los sistemas de inyección electrónica figura 1, son exactos gracias a la innovación de la industria automotriz y sus avances tecnológicos quienes buscan una dosificación perfecta, controlando parámetros de la inyección como la densidad del aire, la temperatura del motor, régimen de giro del motor, carga de motor, tensión de la red del vehículo, oxígeno de la mezcla, condiciones de funcionamiento, entre otros, la información de estos parámetros son transmitidos a través de señales análogas y digitales a la ECU, procesando la información y transmitiendo las ordenes de control al sistema (Oliveres, 2003).

Figura 1.

Esquema de elementos de un sistema de inyección electrónico



Nota. El Grafico simboliza los procesos del el aire y la gasolina hasta ser mezclados. Tomado de *Inyección Secuencial No Cartográfica para Motores de Ciclo Otto* (p,9), (Oliveres, 2003).

2.1.1.1 Unidad de Control Electrónico ECU.

(Agüero et al., 2015) Indican que la Unidad de Control Electrónico (ECU) son unidades de procesamiento de los sistemas de inyección electrónica que gestionan y controlan el funcionamiento del motor, en forma universal parte de una red de unidades electrónicas, compartiendo información constante por medio de sus redes de comunicación, otra de las características de la ECU es controlar el funcionamiento del motor interviniendo en la variación de apertura de válvulas, accionamiento de bobinas o apertura de inyectores con la finalidad de mantener la estequiometría de un motor para reducir las emisiones de gases contaminantes al ambiente, considerando su rendimiento y consumo de combustible.

Los avances tecnológicos para el diagnóstico de fallas fueron creciendo a pasos agigantados de tal manera que se estandarizó el acceso a esa información de las ECU's, la industria automotriz y sus fabricantes acordaron desarrollar el protocolo OBD (On Board Diagnostic), brindando un sistema de diagnóstico a bordo a través de una interfaz estandarizada, permitiendo verificar el trabajo de los componentes del sistema emitiendo alertas de falla (Agüero et al., 2015).

Las unidades de control electrónico figura 2, son capaces de tomar decisiones gracias a las entradas y salidas, siendo estas analógicas y digitales, manejando interfaces de comunicación con diferentes protocolos, la ECU está conformada por hardware y software quienes integran componentes electrónicos en una placa de circuito impreso PCB, microcontroladores, memoria EPROM, memoria Flash, entre otros (Sánchez et al., 2016).

Figura 2

Unidad de Control Electrónico (ECU)

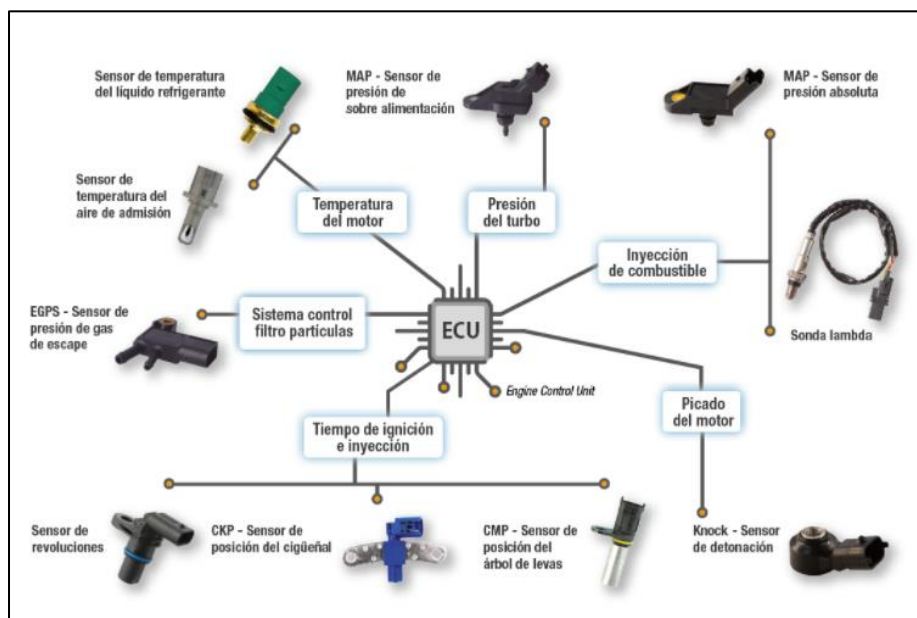


Nota. La ECU se integra por microcircuitos que procesan la información de los sensores (p.10), extraido de *Revisión Documental del Protocolo CAN como Herramientas de Comunicación y Aplicación de Vehículos* (Sánchez et al., 2016).

Sensores. Los sensores se utilizan en vehículos proporcionar información a los diferentes competentes del motor, a la Unidad de Control Electrónica (ECU) y por medio de esta al conductor, proporcionando la información por una pantalla o medidor; los sensores convierten las mediciones que son fenómenos físicos en señales eléctricas con la finalidad monitorear las condiciones óptimas de funcionamiento de sistema transmitiendo su información a la ECU (Denton, 2011).

Actuadores. Los actuadores son dispositivos mecánicos accionados electrónicamente proporcionando una fuerza como una presión neumática, presión hidráulica y/o fuerza motriz eléctrica con la finalidad de desplazar o “hacer actuar” otro dispositivo mecánico, los actuadores son gobernados por una unidad de control (ECU) y en función de su necesidades, los actuadores receptan las señales eléctricas transformándolas en energía para un determinado trabajo programado (Salazar & Ruiz del Solar, 2015).

En la figura 3 se puede observar un diagrama básico de como la unidad de control (ECU) comanda los sensores que conforman la gestión del motor.

Figura 3*Gestión del Motor de Combustión Interna*

Nota. Adaptado de *Reportaje Técnico Sobre Sensores FAE para la Gestión del Motor* (FAE, 2019)

2.1.1.2 OBD II.

Es la segunda versión del OBD, que se modificó con la finalidad de monitorear el catalizador elemento que interviene afecta en el control de emisiones de gases de la combustión, implementando los sensores de oxígeno o también llamadas sonda lambda, el sistema OBD II también tiene la función enviar señales de alerta cuando algún elemento del sistema trabaja de manera errónea, estas señales son almacenadas y registradas en la memoria del sistema permitiendo la ubicación del problema de manera ágil y con ayuda de las herramientas de diagnóstico (On & Diagnostics, 2005).

Cuando estos parámetros antes salen de los rangos establecidos por el fabricante, el sistema OBD II, tiene la función de almacenar esta información, y avisar al conductor de que ocurrió un fallo y el sistema está operando de manera errónea, que mediante un indicador luminoso informa de manera preliminar que se ha producido un error, la información almacenada por el OBD II (Kia Ecuador, 2021).

2.1.1.3 Redes de Comunicación Multiplexadas.

Los dispositivos electrónicos y computadoras independientes por cada sistema que los fabricantes han implementado en los automóviles modernos han dado origen a la arquitectura multiplexada. Una red multiplexada es un circuito único que interconecta las computadoras, disminuyendo así notablemente el cableado y ofreciendo a la vez más seguridad y confort (AutoAvance, 2020).

En la actualidad cada fabricante propone su diseño de interconexión conservando los protocolos de comunicación estándar, la red multiplexada automotriz facilita la emisión de una o más señales que son transmitidas por un solo canal permitiendo la posibilidad de tener varias conexiones al mismo tiempo, en vehículos de media gama podemos encontrar entre 4 a 10 computadoras, mientras que en vehículos de alta gama encontramos hasta 40 computadoras, estas computadoras disponen de gestión electrónica independiente con la capacidad de compartir datos entre sí (AutoAvance, 2020).

2.1.1.4 Red CAN.

Según (Sánchez et al., 2016) el CAN (Controller Area Network) es una red de alta velocidad de dos cables efectiva en sus costos por su funcionalidad y confiabilidad, CAN está diseñado para requerimientos en tiempo real con comunicación abierta para dispositivos inteligentes teniendo como características:

- Opera en tasas de hasta 1 Megabit por segundo
- Soporta máximo 8 bytes por mensaje de trama
- Soporta múltiples ID de mensajes
- Soporta prioridad de mensajes, el ID menor es el ID de mayor prioridad
- Soporta dos tamaños de ID, 11 bits (estándar) y 29 bits (extendido)
- Necesita dos cables, CAN High y CAN Low

2.1.1.5 Protocolo de Comunicación CAN-Bus.

En el protocolo de comunicación CAN-Bus la transmisión de datos se realiza por impulsos eléctricos a través del bus de datos, cada unidad de control dispone de transceptor quien es el encargado de transformar los impulsos eléctricos en un mensaje binario, lenguaje amigable para el controlador y microprocesador (Sánchez et al., 2016).

2.1.1.6 Microcontroladores Programables.

Un microcontrolador es un microcomputador el cual tiene por unidad central de procesos (CPU) un microprocesador. Los microcontroladores son desarrollados para cubrir una gran variedad de aplicaciones en campos como la automoción, instrumentos electrónicos, equipos médicos e industriales, electrodomésticos, juguetes, entre otros (Pérez & Areny, 2007).

Los microcontroladores son empleados para realizar tareas puntuales para lo cual ejecuta un programa que es almacenado permanente en su memoria. El microcontrolador es capaz de interactuar con el exterior adquiriendo datos temporales que van a ser tratados por el mismo e interactuando con el exterior por medio de entradas y salidas que dispone, en aplicaciones a gran escala se recomienda emplear varios microcontroladores quienes serán los encargados de realizar un pequeño grupo de tareas (Pérez & Areny, 2007).

Un microcontrolador asemeja los componentes esenciales de un microcomputador, que son la unidad central de procesos (CPU), memoria y elementos de entrada y salida en una sola pastilla o circuito integrado (Pérez & Areny, 2007).

Los microcontroladores requieren de un oscilador de cristal externo encargado de generar pulsos para la sincronización en su interior con la finalidad de su correcto funcionamiento, todos los procesos realizados en el interior del microcontrolador obligadamente deben ser secuenciales y la velocidad de ejecución de instrucciones depende directamente de la frecuencia que tenga el oscilador (Valdés & Pállas, 2007).

Entrada y salida. Por medio de la entrada y salida, el microcontrolador interactúa con el mundo exterior ya que recibe y envía información dependiendo del tipo de aplicación, las señales de entrada y salida pueden ser tanto digitales como analógicas, para estas últimas se requiere de conversores análogos digitales y viceversa (A/D y D/A), una de las exigencias para los microcontroladores es que posean terminales de entrada y salida numerosos y un tamaño pequeño, empleando sus los puertos en serie o en paralelo (Pérez & Areny, 2007).

Bus. El bus del sistema tiene la función de enlazar todos los elementos internos del microcontrolador, por una parte tenemos el bus de direcciones que determina dirección de almacenamiento datos, y el origen para la obtención de las instrucciones del programa, también encontramos el bus donde los datos viajan a todos los componentes y por último el bus de control que permite al CPU controlar a los componentes (Pérez & Areny, 2007)

Microcontrolador Arduino. Es un conjunto de elementos de software y hardware que tiene el objetivo de facilitar la construcción de circuitos electrónicos con gran variedad de aplicaciones, en la parte de hardware, Arduino es una placa la cual puede ser acoplada junto con otras. Cada placa Arduino posee un microcontrolador de programación en el cual se carga un programa que ha sido programado previamente por el diseñador el cual contiene las acciones que va a realizar el circuito electrónico. Además del microcontrolador de programación, una placa Arduino posee otros elementos que funcionan en conjunto (Moreno & Córcole, 2017):

Pines Digitales. Los pines digitales tienen dos estados High (5V) y Low (0V) y estos pines pueden ser utilizados como entradas o salidas.

Pines Analógicos. Los pines analógicos en cambio son capaces de medir un rango de valores de voltaje, la placa Arduino tiene la capacidad de representar los valores en un rango de 0 a 1023 para lectura, se establece este rango de posibilidades porque la placa Arduino representa la información utilizando 10 bits ($2^{10}=1024$), y para escritura utiliza el rango de 0 a 255 ya que la información es representada utilizando 8 bits ($2^8=256$) (Moreno & Córcole, 2017).

Pines de alimentación de sensores. Por medio de estos pines la placa Arduino tiene la capacidad de alimentar dispositivos externos utilizando 5V o 3.3V.

Microcontrolador de comunicaciones. El microcontrolador de comunicaciones tiene la función de establecer la comunicación entre los diferentes dispositivos presentes en la placa.

Software. El software utilizado para la creación de programas para la placa Arduino lleva el mismo nombre y es la herramienta oficial para desarrollar la programación que utilizara la placa para realizar las acciones de manera automatizada. (Moreno & Córcole, 2017).

La interfaz es muy amigable para el usuario tanto que con tan solo iniciar el programa en el ordenador ya puedes empezar a programar, una vez que el código del programa este completo desde el programa puedes seleccionar el botón para cargar el programa en la placa o guardar el programa en tu computador junto con otras funciones básicas (Moreno & Córcole, 2017)

2.1.1.7 IFTTT, Internet de las Cosas (IOT)

EL IFTTT “IF THIS THEN THAT o en español “Si Esto, Entonces Aquello”, permite construir, publicar, aprender y crecer entre aplicaciones que dispongan de una conexión a internet, favoreciendo y facilitando la difusión de contenido (Gil-Solés, 2015)

2.1.1.8 Webhook

Un webhook permite que una aplicación proporcione aplicaciones externas con información en tiempo real, así como también proporciona datos para estas aplicaciones a medida que suceden, obteniendo datos de instantáneos, el webhook crea una solicitud de HTTP o URL con la información deseada, esta información es transmitida a la aplicación de interés por medio de la plataforma IFTTT. (Bloemen, 2021)

Figura 4

Esquema de funcionamiento Webhooks - IFTTT



Nota. Los Webhooks son solicitudes de http o URL que se originan cuando existe un evento transmitiendo la información en tiempo real.(Bloemen, 2021).

2.1.1.9 Almacenamiento Digital.

Desde finales de la década de 1980, la revolución digital ha cambiado la economía y la sociedad iniciando por una economía conectada y desarrollada, caracterizada por el uso generalizado y la difusión de Internet red de banda ancha, seguidamente la economía digital ha crecido debido a la expansión del uso desarrollando una plataforma digital como modelo de negocio para la provisión de bienes y servicios, ahora tiene el objetivo de conseguir economía digital basada en modelos de producción y consumo mediante una combinación de tecnologías en todos los aspectos económicos, sociales, industriales y medioambientales. (Rovira Sebastian, 2022)

La adopción de la tecnología a los sistemas digitales ha revolucionado nuestra sociedad, un sistema digital puede generar, transmitir, procesar, o almacenar señales digitales como libros, fotografías o códigos binarios; el sistema digital se define también por la oposición al sistema analógico, empleado tradicionalmente por la tecnología, el sistema analógico convierte los datos a señales analógicas, es decir, aquellas que expresan los datos a través de impulsos eléctricos. (Delgado Planás, 2016)

Nube de Datos. Existen diversos tipos de nubes (cloud) atendiendo a las necesidades de una empresa o persona que dependerán de su ubicación, aplicaciones y qué clientes pueden usarlas. (Kezherashvili, 2013)

Nube Pública. Tienen la capacidad de procesamiento y almacenamiento sin instalar máquinas localmente, por lo que no tiene una inversión inicial o gasto de mantenimiento y el servicio tiene un valor.

Nube Privada. Dispone de una plataforma ubicada dentro de las instalaciones del usuario y no suele ofrecer servicios a terceros, una nube privada es una plataforma que incluye máquinas, almacenamiento e infraestructura de red (IaaS), pero también puede desplegar aplicaciones (PaaS) e incluso aplicaciones (SaaS).

Nubes Híbridas. Las nubes híbridas consisten en combinar las aplicaciones locales con las de la nube pública, permitiendo a una empresa mantener el control de sus principales aplicaciones sin importar el tiempo y el lugar.

Nubes Combinadas. Consiste en la combinación de dos o más nubes privadas o públicas, administradas por diferentes usuarios y proveedores facilitando el cambio de sus servicios a sus clientes.

Nubes Comunitarias. Este tipo de nubes sirven para que varias organizaciones participen de sus recursos informáticos y tecnológicos al colaborar en negocios, servicios u objetivos.

3. CAPÍTULO III.

3.1 Materiales y Métodos

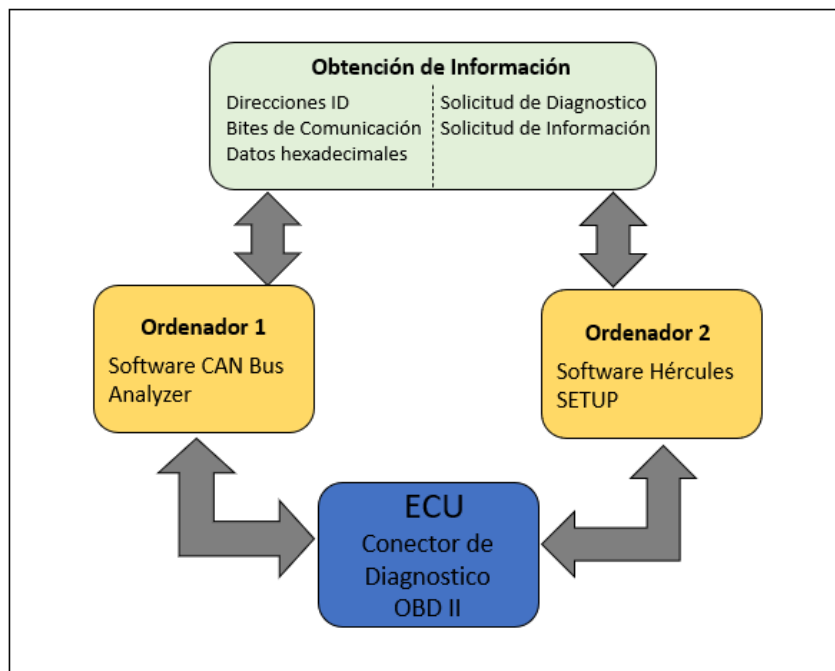
3.1.1 Materiales

En la elaboración del dispositivo de diagnóstico OBD II con acceso remoto y en tiempo real al almacenamiento de la ECU, se utilizó un Scanner Launch THINKDIAG, un cable dual “splitter” de diagnóstico OBD II para la conexión en paralelo de los softwares CAN Bus Analyzer y Hércules, la información extraída se utilizó en el software de programación para la interpretación de datos del microprocesador de Arduino MEGA 2560 y el micro procesador CAN Bus Shield, por último la información procesada y traducida en una nube digital para la interpretación en línea y en tiempo real.

En la figura 5 se muestra el diagrama de referencia para la extracción de la información originada por un código de falla en la ECU.

Figura 5

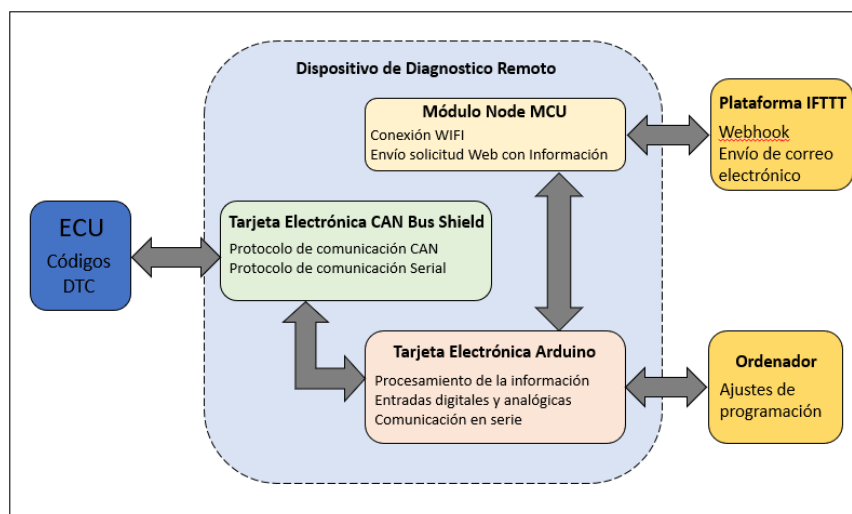
Diagrama para la Obtención de la Información.



En la figura 6 se muestra el diagrama de referencia y funciones para la construcción del dispositivo de diagnóstico remoto y en tiempo real.

Figura 6

Diagrama Referencial del Dispositivo de Diagnostico Remoto



En la Tabla 1, se detalla los materiales utilizados durante el proceso de elaboración del dispositivo de diagnóstico OBD II con acceso remoto y en tiempo real al almacenamiento de la ECU.

Tabla 1

Equipos Utilizados para el Desarrollo del Proyecto

Ítem	Cantidad	Descripción	Características	Observaciones
1	1	Scanner Automotriz	Launch ThinkDiag	Año 2019
2	1	Osciloscopio Automotriz	MicSig	Año 2019
3	1	Cable Dual Splitter	Diagnostico OBD II	-
4	2	Computadores portátiles	LG - Asus	Ordenadores
5	1	CAN Bus Analyzer	Software	Escáner Automotriz
6	1	Hércules SETUP	Software	Escáner Automotriz
7	1	Arduino Mega 2560	Microprocesador	Año 2020
8	1	SunLower CAN Shield V2	Microprocesador	Año 2019
9	1	Arduino EP8226	Modulo Wifi	Año 2021
10	1	IDE de Arduino	Software	Programación
11	1	Sitio Web ITFF	WEB	Receptor de URL
12	1	Correo Electrónico	Gmail	Uso Personal
Total	13	-	-	-

3.1.2 Marco Metodológico

El método experimental se basa en la identificación, extracción, almacenamiento y visualización de las variables de investigación con la finalidad de establecer los procedimientos deseados.

3.1.2.1 Enfoque y Tipo de Investigación.

El propósito del dispositivo de diagnóstico OBD II con acceso remoto y en tiempo real al almacenamiento de la ECU, permitió extraer y almacenar los códigos DTC que fueron visualizados de forma remota con la ayuda de una nube digital de almacenamiento de datos.

La recopilación de información a través de libros, revistas, blog, artículos de investigación científica, foros entre otros documentos que respaldaron el desarrollo tecnológico.

3.1.2.2 Procedimientos

Fase 1. En esta primera fase se determinó el vehículo que permitió el acceso a la red CAN con OBD II genérico, obteniendo datos en tiempo real y lecturas de códigos de fallas.

Fase 2. Partiendo de los resultados de la fase 1, se analizó el sistema de diagnóstico en el vehículo, identificando los mensajes hexadecimales que se generan al existir un código de falla.

Fase 3. Se diseñó un dispositivo utilizando una placa Arduino y una placa CAN Bus Shield, quienes facilitaron la comunicación con la red CAN del Vehículo mediante el conector de diagnóstico OBD II.

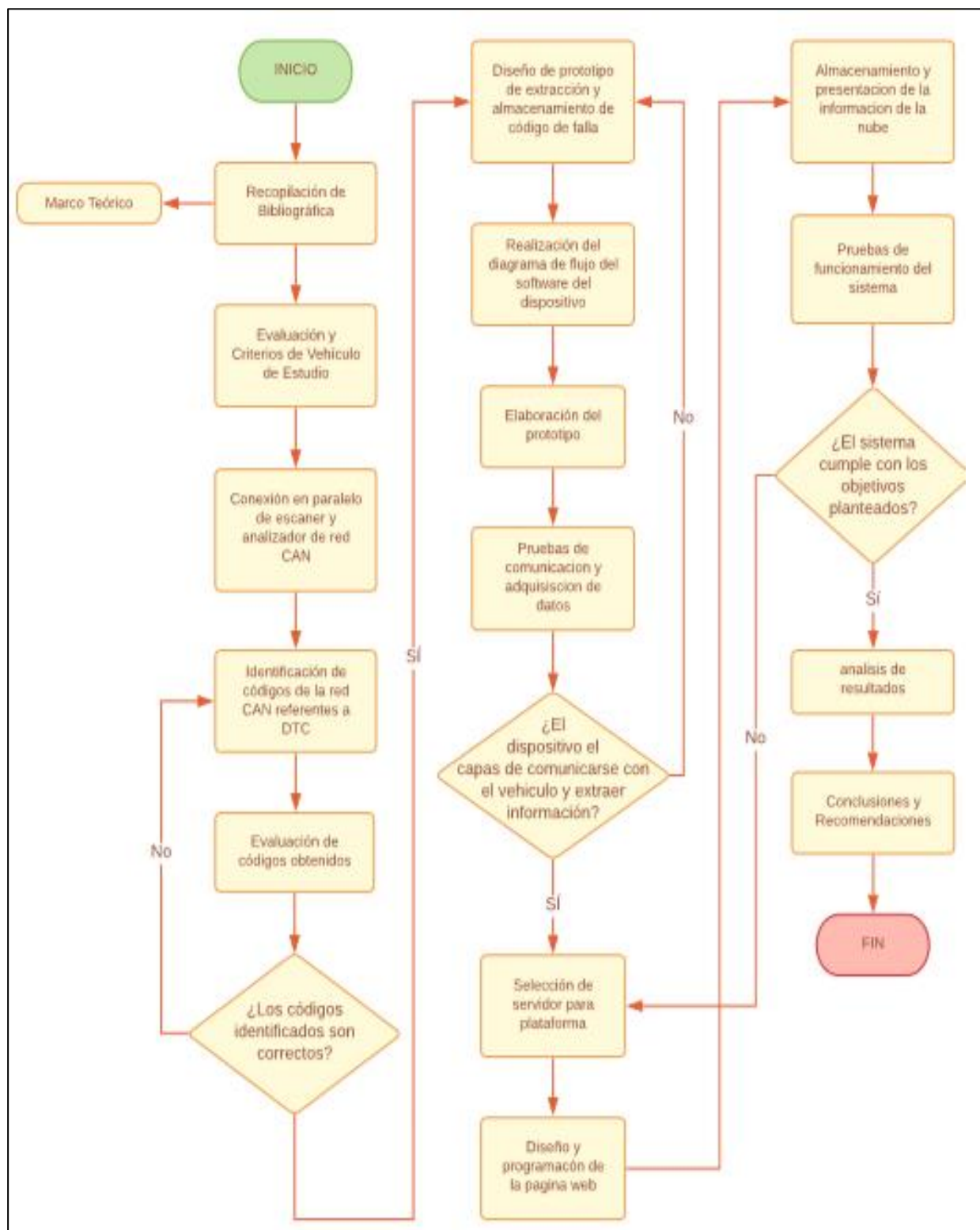
Fase 4. según los resultados y la información de las fases 1, 2, 3, con ayuda de un módulo wifi compatible con Arduino (ESP8266), mismo que ayudo la conectividad a internet y de esta forma se transfirió los datos a la plataforma ARDUINO CLOUD permitiéndonos el acceso a los códigos de falla en línea generados por la ECU y alertados por el testigo Check Engine.

3.1.2.3 Flujo de Proceso

En la figura 7 se plantea el procedimiento para la elaboración del dispositivo de diagnóstico remoto.

Figura 7

Flujo de Procedimientos



3.1.3 Evaluación de Criterios de Vehículo de Estudio

Para el desarrollo del presente trabajo, se utilizó un vehículo de servicio público marca Kia, modelo Rio LX 1.4 4P 4x2 TM, Año 2019, este vehículo dispone de su conector de diagnóstico en la parte inferior izquierda del volante del conductor, específicamente en el compartimiento de los fusibles, dentro de este compartimiento se encuentra el conector de diagnóstico OBD II como lo indica la figura 8.

Figura 8

Ubicación de Conector de Diagnostico



3.1.3.1 Procedimiento Visual de Verificación de la Red CAN

Desde el asiento del conductor, se ubica la zona izquierda del volante, se retira la cubierta del compartimiento de fusibles y conectores, seguidamente se localiza el conector de diagnóstico OBDII, en la figura 9 se muestra el conector de diagnóstico OBDII, finalmente se verifica todos los pines que dispone el conector y cuál son los energizados, para determinar el protocolo de comunicación de la ECU.

Figura 9

Conector de Diagnóstico OBD II



3.1.3.2 Procedimiento de Verificación de la Red CAN con Escáner Automotriz

Luego de ubicar el conector de diagnóstico, se conecta el escáner automotriz de la marca LAUNCH, modelo ThinkDiag con actualización 2021, mismo que busca todos los posibles protocolos de comunicación disponibles, proceso que se aprecia en la figura 10.

Figura 10

Búsqueda de Protocolo de Comunicación

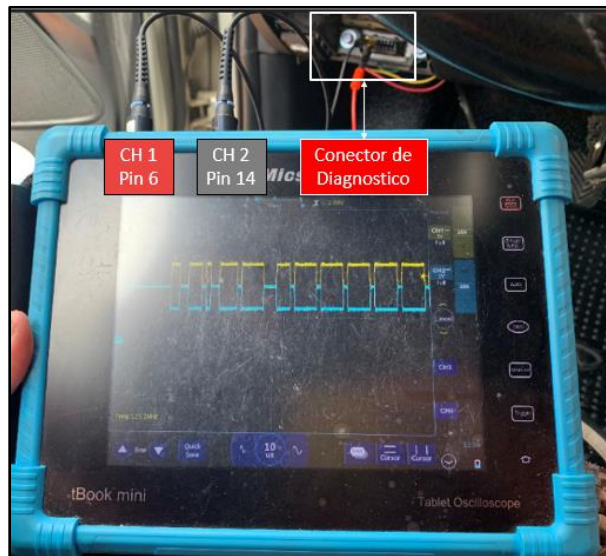


3.1.3.3 Procedimiento con Osciloscopio Automotriz Para la Verificación de la Red CAN

Para este procedimiento se utiliza un Osciloscopio Automotriz de la marca MIGSIG modelo de 4 canales, se conecta los canales en los pines energizados de acuerdo con el protocolo de comunicación y sus conexiones a tierra, una vez establecida la conexión, se visualiza y analiza los valores de velocidad, frecuencia y voltaje como lo indica la figura 11.

Figura 11

Conexión entre Osciloscopio y Conector de Diagnostico

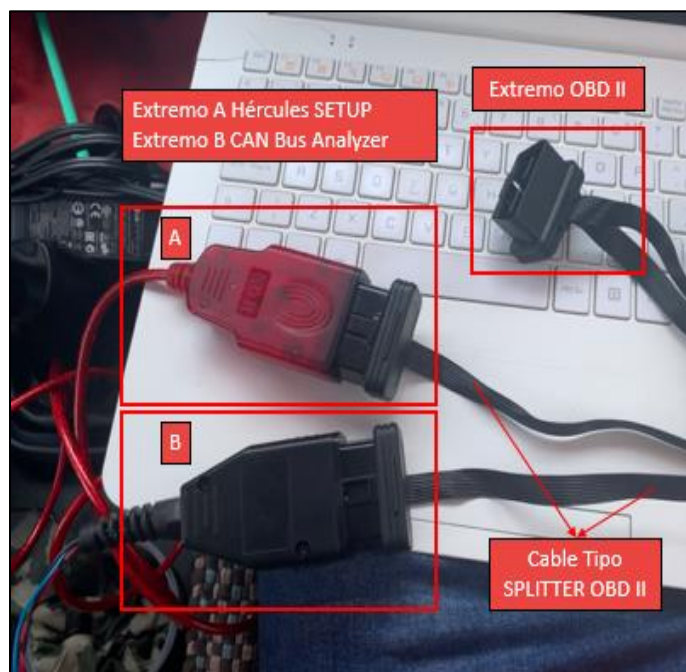


3.1.4 Conexión en Paralelo de CAN Bus Analyzer y Hércules SETUP

En la figura 12 se observa la conexión en paralelo de dos softwares con sus respectivos ordenadores al cable Splitter y este al conector de diagnóstico del vehículo, esto permitirá el rastreo, registro y transmisión de mensajes entre ordenadores.

Figura 12

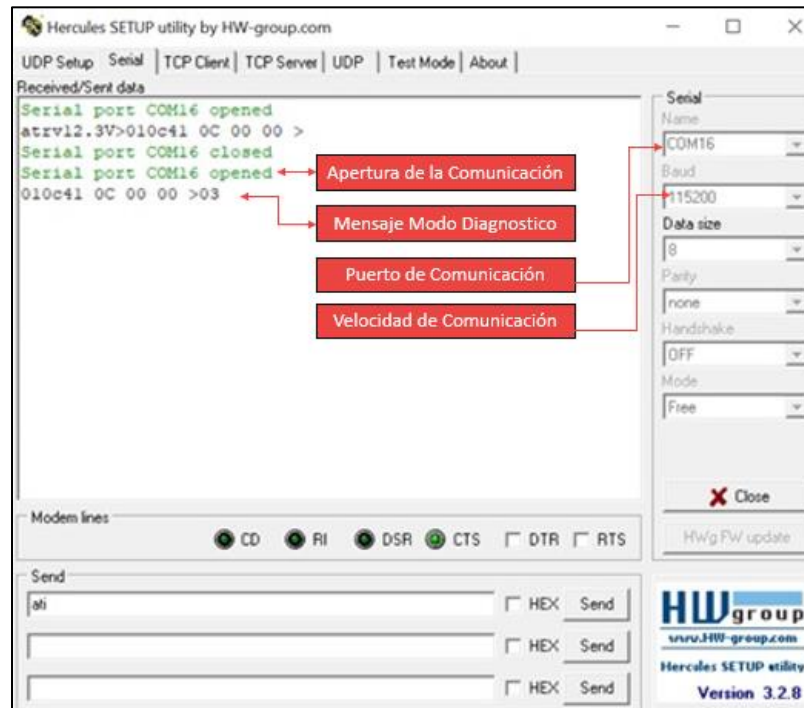
Conexión en Paralelo de los Softwares Hércules SETUP y CAN Bus Analyzer



En las opciones "SERIAL" del software **Hércules SETUP**, se configura el puerto de conexión correspondiente para establecer la comunicación con la Unidad de Control (ECU), lograda la comunicación, se permite enviar mensajes desde el ordenador a la Unidad de Control (ECU) a través del software solicitando la información deseada, la Unidad de Control (ECU) responde de manera inmediata el mensaje de respuesta a la solicitud; en la figura 13 se observa el puerto de comunicación serial, la apertura de comunicación entre la Unidad de Control (ECU) y el envío de la solicitud de información DTC.

Figura 13

Solicitud de Información DTC



En el software paralelo **CAN Bus Analyzer**, permite rastrear y analizar los mensajes que se están transmitiendo en tiempo real en la Unidad de Control (ECU), para nuestro caso se selecciona todos los parámetros de la pantalla, en el momento que se envía el mensaje de solicitud de información desde Hércules SETUP, se origina y visualiza la información nueva y apartada de lo ya señalado que pertenece a la respuesta de la solicitud.

En la figura 14 se puede observar los parámetros en tiempo real que la Unidad de Control (ECU) se encuentra transmitiendo en fracciones de segundos antes de solicitar la información.

Figura 14*Parámetros en Tiempo Real*

TRACE	ID	DLC	DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5	DATA 6	DATA 7	TIME STAMP (sec)	TIME DELTA (sec)
RX	0x316	8	0x31	0x00	0x00	0x00	0x00	0x0A	0x00	0x7F	279.0882	0.001
RX	0x387	8	0x20	0xC7	0x86	0x15	0x00	0x89	0x07	0x00	279.0872	0.000
RX	0x381	8	0x80	0x00	0x40	0x00	0x00	0x9D	0x5D	0x00	279.0872	0.000
RX	0x2B0	5	0xA9	0xF2	0x00	0x07	0x3A				279.0872	0.002
RX	0x50C	8	0x88	0x10	0x00	0xA0	0x06	0x00	0x00	0x20	279.0852	0.000
RX	0x4F1	4	0x00	0x00	0x5C	0x60					279.0852	0.006
RX	0x329	8	0x0F	0xA2	0x63	0x0E	0x11	0x2D	0x00	0x0E	279.0793	0.000
RX	0x260	8	0x00	0x00	0x00	0x00	0x16	0x8C	0x5F	0x10	279.0792	0.000
RX	0x200	6	0x00	0x00	0x00	0x00	0x00	0x00			279.0792	0.000
RX	0x1BF	8	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	279.0792	0.000
RX	0x18F	8	0x00	0x00	0x00	0x00	0x00	0x5A	0x00	0x00	279.0792	0.001
RX	0x316	8	0x31	0x00	0x00	0x00	0x00	0x0A	0x00	0x7F	279.0782	0.000
RX	0x386	8	0x00	0x80	0x00	0x00	0x00	0x40	0x00	0x80	279.0782	0.001
RX	0x38A	8	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	279.0772	0.000
RX	0x2B0	5	0xA9	0xF2	0x00	0x07	0x09				279.0772	0.003

3.1.5 Obtención de información

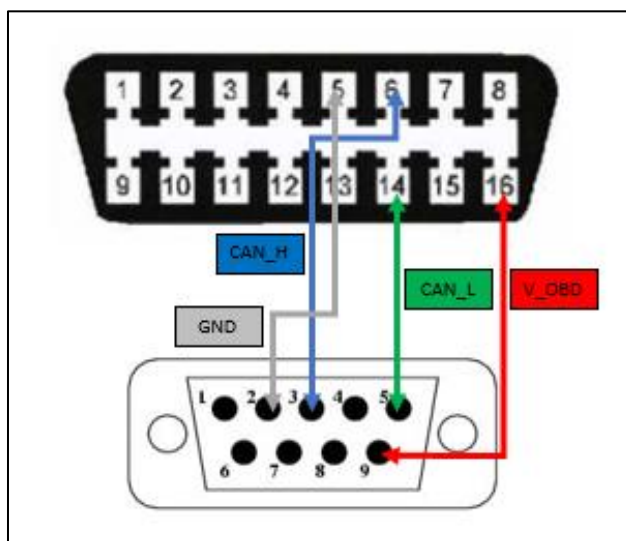
El vehículo de prueba posee un código de falla presente, mismo que lo visualizamos por la presencia de la luz MIL encendida en el tablero, por parte del software Hércules SETUP se logra de manera inmediata la respuesta a la solicitud enviada “código de falla”, lo mismo ocurre con el Software CAN Bus Analyzer al ser la conexión en paralelo y enviar la solicitud se adquiere la información solicitada.

3.1.6 Diseño de prototipo de extracción de la información**3.1.6.1 Diseño de cable de conexión OBD II a DB9.**

Para el diseño del cable de conexión se necesita de un extremo un conector OBD II (macho) mismo que permitirá la conexión con el conector de diagnóstico (hembra) del vehículo, en el otro extremo un conector DB9 (hembra) que permitirá la conexión con la placa SunFlower CAN Shield V2, adicionalmente para la conexión de los conectores se utiliza un cable UTP 5 tomando en cuenta el diagrama ubicado en la figura 15.

Figura 15

Diagrama de Conexión Cable OBD II – DB9

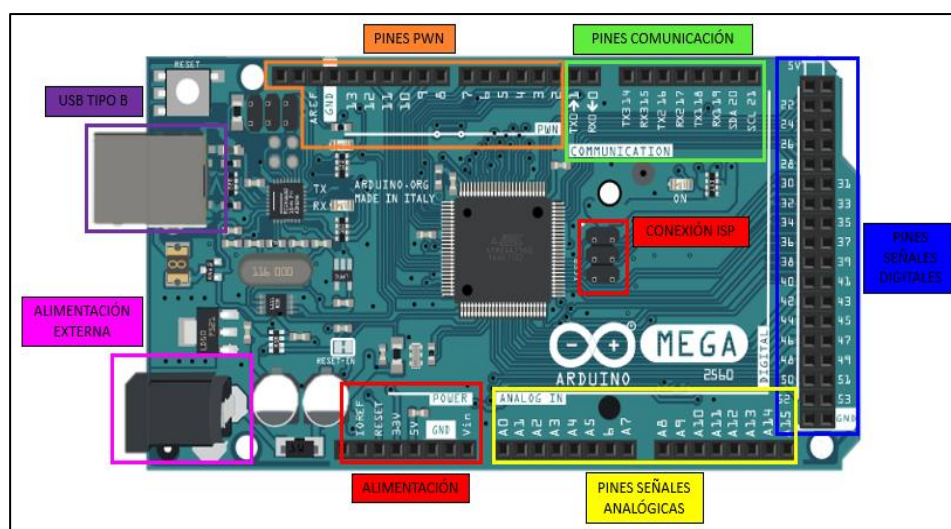


3.1.6.2 Diseño de programación Arduino MEGA 2560.

El dispositivo Arduino MEGA 2560 permite la comunicación entre dispositivos electrónicos, este dispositivo dispone de pines de alimentación, entradas/salidas analógicas y digitales, entradas/salidas PWM, comunicación en serie, ICSP e I2C como lo indica la figura 16.

Figura 16

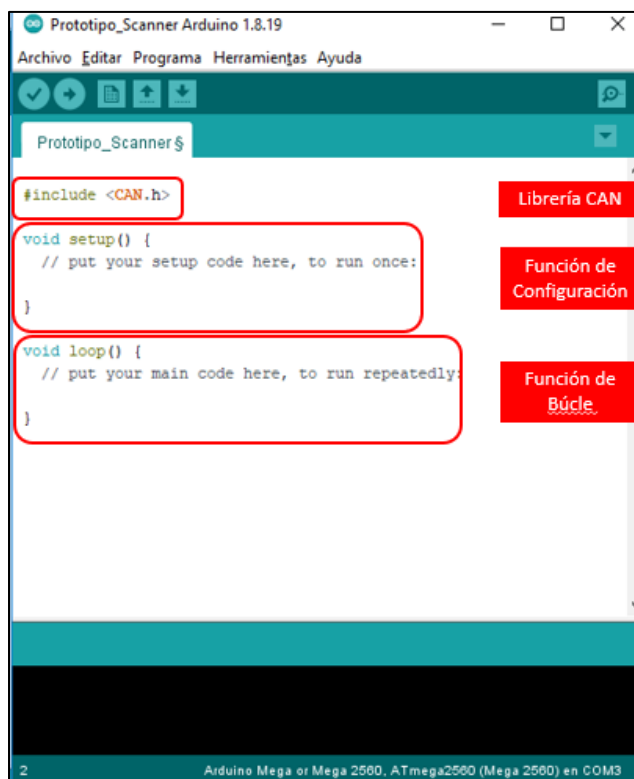
Distribución de Arduino Mega 2560



La página web oficial de Arduino (www.arduino.cc) ofrece el software gratuito para descargarlo y ejecutarlo en el ordenador, este software contiene la librería de comunicación por protocolo CAN de fabrica en sus últimas versiones (Arduino IDE), se establece las librerías para el desarrollo del prototipo de comunicación entre los dispositivos (Arduino Mega 2560 – SunFlower CAN Shield V2), la Figura 17 indica las tres funcione de programación de la tarjeta electrónica Arduino.

Figura 17

Funciones de Programación de Arduino



En la figura 18 se observa la función de configuración, donde inicializa los puertos de comunicación para el módulo SunFlower CAN Shield V2 y el módulo Node MCU (ESP8266). además de la velocidad de transmisión de datos, la cantidad de bits, la dirección de comunicación y la condición de inicio fallido en caso de que no exista la comunicación entre estos dos dispositivos.

Figura 18

Programación Función de Configuración

```

Prototipo_Scanner $
#include <CAN.h>
void setup() {
  Serial.begin(9600);
  Serial1.begin(9600);
  if (!CAN.begin(500E3)) {
    Serial.println("INICIO FALLIDO");
    while (1);
  }
  if (useStandardAddressing) {
    CAN.filter(0x7e8);
  } else {
    CAN.filterExtended(0x18daf110);
  }
}
void loop() {
}

```

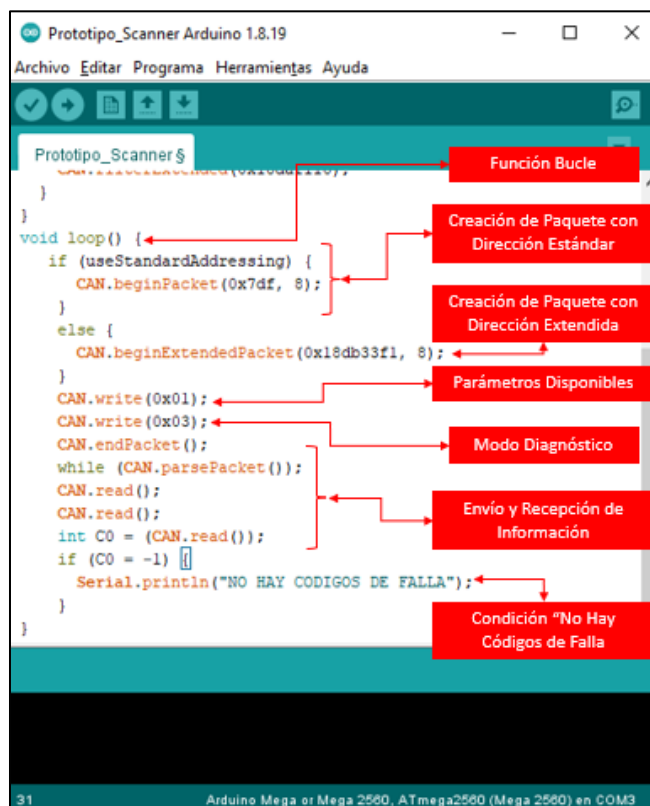
Diagram labels and their corresponding code elements:

- Inicio de puertos de comunicación con dispositivos**: Points to the `Serial.begin(9600);` and `Serial1.begin(9600);` lines.
- Velocidad de Transmisión**: Points to the `500E3` value in the `CAN.begin(500E3)` call.
- Condición "INICIO FALLIDO"**: Points to the `"INICIO FALLIDO"` string in the `Serial.println` call.
- Dirección de Comunicación**: Points to the `0x7e8` value in the `CAN.filter(0x7e8)` call.
- Cantidad de Bits**: Points to the `0x18daf110` value in the `CAN.filterExtended(0x18daf110)` call.

En la función de bucle se ubican todas las instrucciones que se repiten de manera constante en el programa, de este modo, en esta función se crea un paquete de información que es enviado a la ECU desde Arduino por medio del protocolo CAN, en la información a enviar se establece los parámetros disponibles y se solicita los códigos de fallas de diagnóstico, para posteriormente recibir la respuesta de la ECU arribando su información en el puerto serial 0 de Arduino, en este paquete también se establece la traducción de la información a términos amigables para su diagnóstico (código DTC), esta lectura se realiza con un dígito a la vez como se observa en la figura 19.

Figura 19

Programación Función Bucle



Para la lectura de datos se establece las siguientes condiciones:

1. El tamaño del paquete debe ser mayor a "0".
2. El valor de la lectura de información por protocolo CAN, debe ser mayor a "3".
3. El valor de la lectura de información por protocolo CAN, debe ser igual a "0x43".

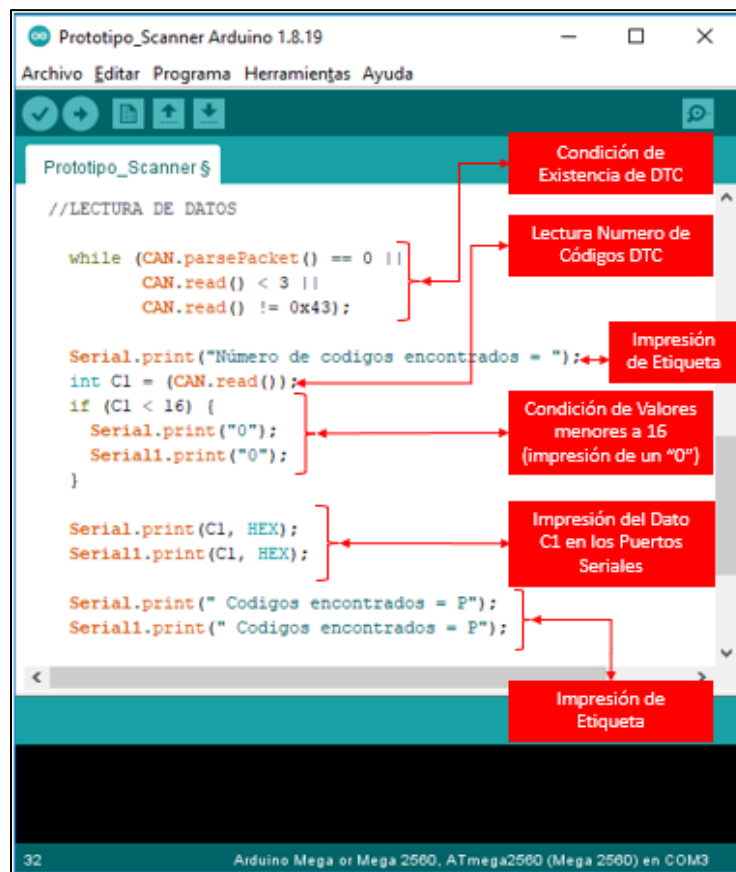
Si cualquiera de estas condiciones se cumple, debe existir un código de falla, por lo tanto, el siguiente paso es imprimir una etiqueta "numero de códigos encontrados" en la pantalla y posteriormente los códigos DTC, realizando la lectura de cada dígito del mismo, por medio del puerto serial "0", al trabajar en el sistema hexadecimal se establece una condición de que:

Si el valor leído por el puerto serial 0 es menor a 16 en el monitor serie se va a imprimir primeramente el valor 0 antes del valor C1 (número de códigos de falla encontrados) caso

contrario únicamente se imprime el valor C1 (esta condición se aplica para todos los datos leídos por el Arduino), luego de este valor se imprime una etiqueta “códigos encontrados = P”, como establece en la figura 20.

Figura 20

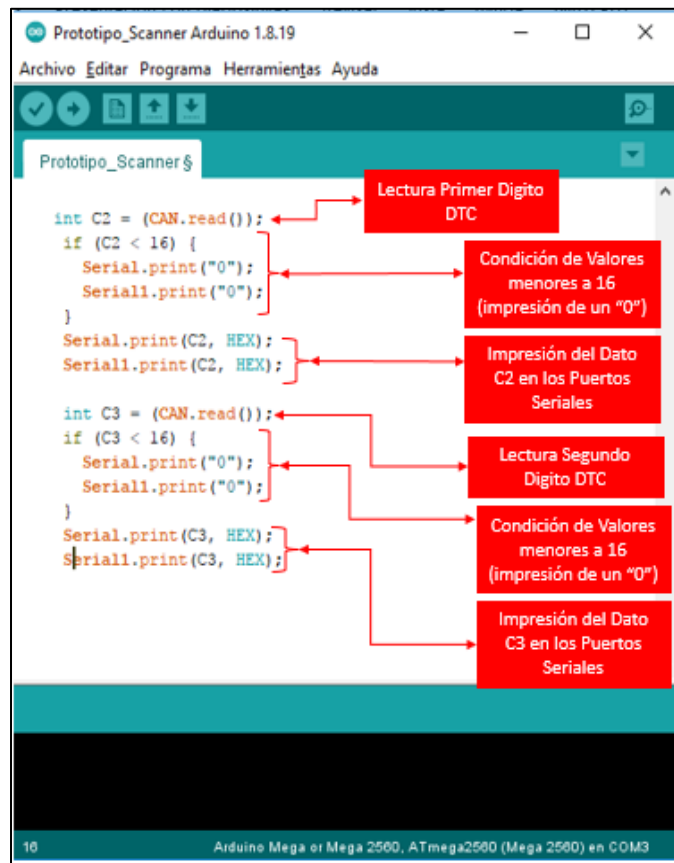
Condiciones de Lectura de DTC



De manera analógica, se realiza el mismo proceso para las variables C2 y C3 que son el primer y segundo dígito del código DTC respectivamente como lo indica la figura 21.

Figura 21

Condiciones de Lectura de DTC



Los datos podrán ser apreciados visualmente en el monitor serie del Arduino IDE tanto.

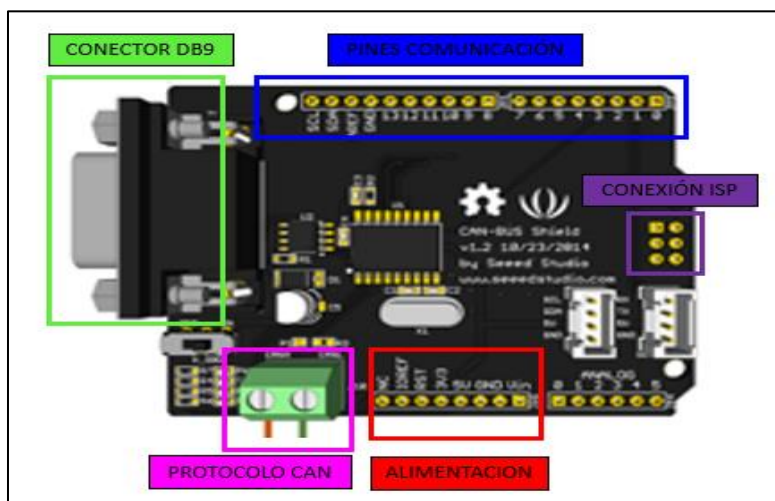
3.1.6.3 Diseño de conexión Sun-Flower CAN Shield V2.

Este dispositivo es la interfaz de comunicación que permite la transmisión de información en diferentes protocolos, por protocolo CAN con la ECU y por protocolo serial con Arduino (UART).

En la figura 22 se muestra los componentes de la tarjeta electrónica CAN Bus Shield.

Figura 22

Distribución Sun-Flower CAN Shield V2

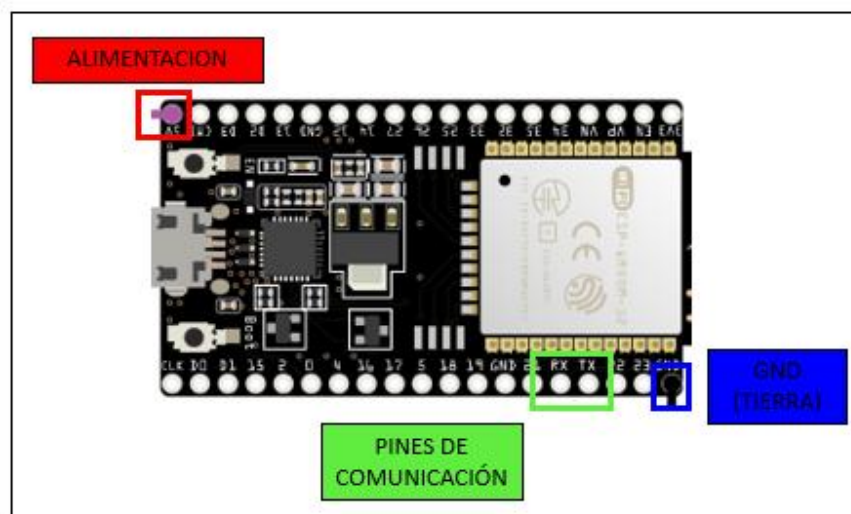


3.1.6.4 Diseño de conexión y programación módulo WIFI NODEMCU (ESP8266-12E).

La conexión a internet del dispositivo se realiza mediante el módulo WiFi NODEMCU (ESP8266-12E), este módulo contiene pines de comunicación Rx y Tx, la conexión a la tarjeta electrónica Arduino se realiza por cables en sus pines similares como lo indica la figura 23.

Figura 23

Pines de Conexión del Módulo ESP8266



En la figura 24 se observa el ingreso de la librería de conexión wifi en la programación del módulo Node MCU y la biblioteca con la solicitud web, en la figura 25 se define los datos que

permiten conectarse a una red Wifi, red Wifi, contraseña especificada por el usuario, además del nombre del evento creado en el sitio Web, la clave de acceso único para el usuario

Figura 24

Bibliotecas de Conexión Wifi y Solicitud Web

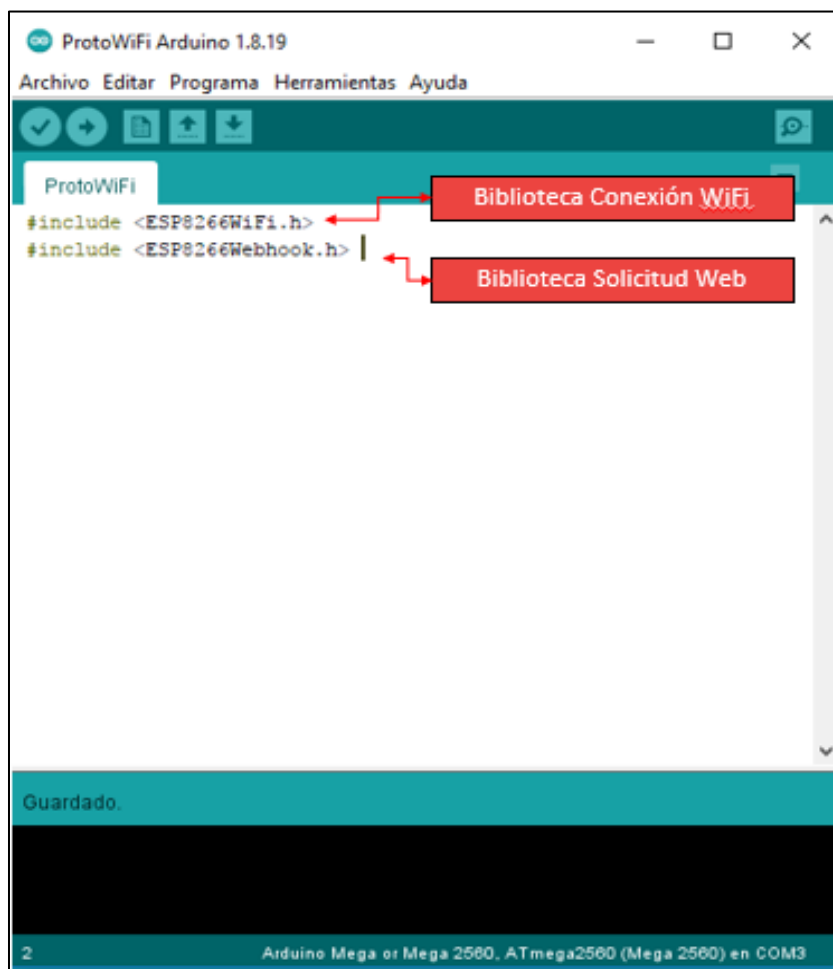


Figura 25

Bibliotecas de Conexión Wifi y Solicitud Web

```

ProtoWiFi Arduino 1.8.19
Archivo Editar Programa Herramientas Ayuda

ProtoWiFi$
#include <ESP8266WiFi.h>
#include <ESP8266Webhook.h>

#define _SSID "iPhone de Byron"
#define _PASSWORD "1234567890"
#define KEY "m-V42841baq5HJqbSuqgzOf_iKcj0FndgBPTRlsf35V"
#define EVENT "Codigo_de_Falla"

Webhook webhook(KEY, EVENT);

char valor1=0;

```

Annotations in the image:

- Nombre Red Wifi (points to `_SSID`)
- Contraseña (points to `_PASSWORD`)
- Llave de Acceso (points to `KEY`)
- Evento (points to `EVENT`)
- Creación de Objeto (URL) (points to `Webhook webhook(KEY, EVENT);`)
- Definición Valor 1 (points to `char valor1=0;`)

Guardado.

8 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) en COM3

En la función denominada "SETUP" se inicializa tanto la comunicación Serial como la conexión Wifi en modo "STATION ó STA", el estado de conexión Wifi se puede observar mediante el monitor de serie de Arduino.

En la figura 26 se observa la programación en la función de configuración donde se define el inicio del puerto serial y la información del estado de conexión y la escritura para el modo estación.

Figura 26

Comunicación Serial y Modo de Conexión STATION

```

ProtoWiFi Arduino 1.8.19
Archivo Editar Programa Herramientas Ayuda
ProtoWiFi $
void setup() {
  // Begin the Serial at 9600 Baud
  Serial.begin(9600);
  while (!Serial) {}
  // Conectamos a la red Wifi
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(1000);
  Serial.println();
  Serial.println();
  Serial.println("Conectando a la red Wifi... ");
  Serial.println(_SSID);
  WiFi.begin(_SSID,_PASSWORD);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print("-");
  }
  Serial.println("");
  Serial.println("Wifi conectada");
  Serial.println("Direccion IP: ");
  Serial.println(WiFi.localIP());
  delay(500);
}
Guardado.
Información de Estado de Conexión
13 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) en COM3

```

Por otro lado, en la función “Loop” de la figura 27, se establece que si existe información entrante por el puerto serial se realiza la lectura de la información y se la almacena en la variable “VALOR1” misma que es añadida al URL con el que se realizara la solicitud Web por medio de la función “webhook - trigger”.

Finalmente se verifica si la solicitud realizada fue exitosa que es cuando se obtiene un valor de respuesta “200” por parte de la página Web, caso contrario la conexión es fallida.

Figura 27

Lectura de información y almacenamiento de datos.

```
ProtoWiFi Arduino 1.8.19
Archivo Editar Programa Herramientas Ayuda

ProtoWiFi $
void loop() {

if (Serial.available()) {
  valor1=Serial.read();
  Serial.println(valor1);
}

int response = webhook.trigger(String(valor1));
if(response == 200)
  Serial.println("OK");
else
  Serial.println("Failed");
}
}
```

Lectura e Impresión de Datos Entrantes del Puerto Serial

Inclusión de Valor 1 en URL

Verificación Exitosa

Guardado.

14 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) en COM3

3.1.6.5 Diseño del almacenamiento de datos.

Para el desarrollo del almacenamiento de datos utilizamos la página web <https://www.ifttt.com> que facilita la integración de dispositivos, servicios y aplicaciones con los pasos descritos en la figura 28:

Figura 28

Proceso de Creación de Webhooks



4. CAPITULO IV.

4.1 Resultados y Discusión

4.1.1 Evaluación de Vehículo

4.1.1.1 Resultado del procedimiento visual de verificación la Red CAN.

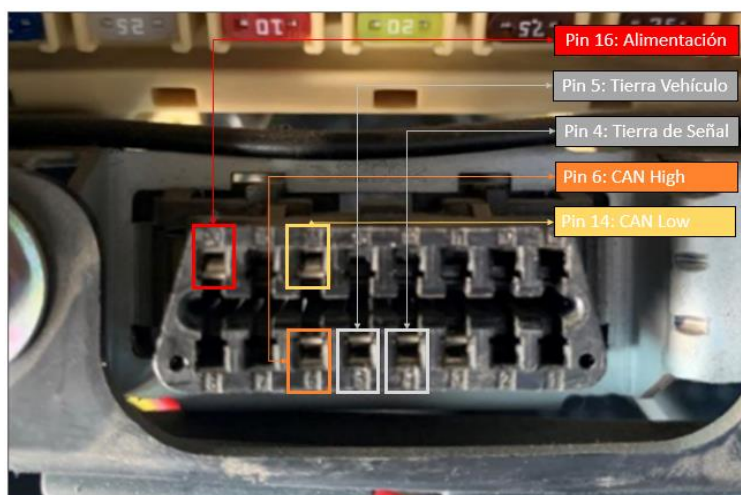
Se logró la ubicación del conector de diagnóstico OBDII en el vehículo y se realizó un análisis visual de los pines del conector energizados, en la Tabla 2 se observa los pines energizados del conector con su respectiva referencia.

Tabla 2

Pines Energizados Conector Kia Rio 2019

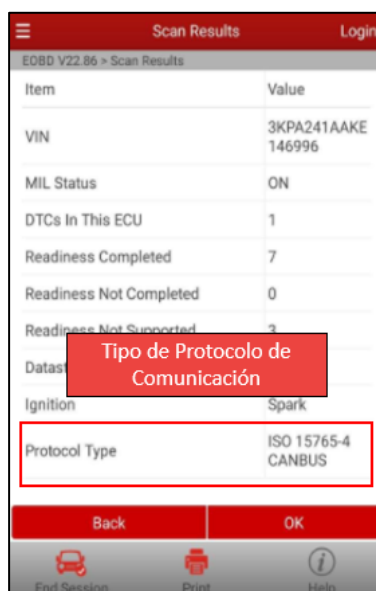
Pin	Referencia	Condición
16	Alimentación	Energizado
5	Tierra Vehículo	Energizado
4	Tierra Señal	Energizado
6	CAN High	Energizado
14	CAN Low	Energizado

Por la condición de los pines del conector, se establece el protocolo de comunicación por Red CAN (Controller Area Network), en la figura 29 se observa el conector de diagnóstico y se indica la ubicación de cada pin energizado.

Figura 29*Pines Energizados Red CAN*

4.1.1.2 Resultado del procedimiento de verificación de la Red CAN con escáner automotriz.

La función OBD II genérico del escáner de diagnóstico automotriz de la marca LAUNCH, modelo ThinkDiag, encuentra el protocolo de comunicación disponible, se establece el tipo de protocolo de comunicación "ISO 15765-6 CAN Bus", en la figura 30 se observa el estado de comunicación entre el escáner y la Unidad de Control Electrónico, así como también, el número VIN del vehículo incluso se indica la cantidad de Códigos DTC.

Figura 30*Protocolo de Comunicación CAN*

4.1.1.3 Resultado del procedimiento con Osciloscopio Automotriz para la verificación de la Red CAN.

Se utilizó un Osciloscopio Automotriz de la marca MIGSIG de 4 canales, se conecta el canal 1 en el Pin 6 de CAN High, el canal 2 se conecta en el Pin 14 de CAN Low con sus respectivas conexiones a tierra, una vez establecida la conexión, se visualizó y analizó los valores de velocidad, frecuencia y voltajes, obteniendo los resultados expresados en la Tabla 3.

Tabla 3*Valores de Voltaje y Frecuencia Red CAN*

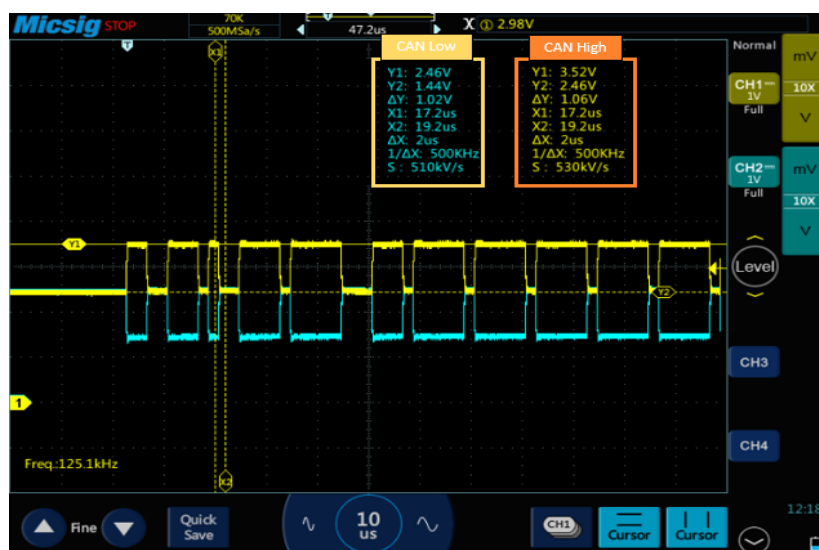
Canal	Detalle	Unidad	Valor	Protocolo Comunicación
1	CAN High	Voltaje Y1 – Y2 (V)	2,5 – 3,5	Red CAN
2	CAN Low	Voltaje Y1 – Y2 (V)	1,5 – 2,5	Red CAN
1 y 2	CAN H - CAN L	Frecuencia (KHz)	500	Red CAN

Los valores obtenidos con ayuda del osciloscopio automotriz son propios de la Red CAN, siendo el único protocolo de comunicación que dispone de una frecuencia de 500 KHz.

En la figura 31 se muestra las gráficas de CAN High y CAN Low, determinado sus valores de propios de protocolo de comunicación CAN.

Figura 31

Mediciones CAN H – CAN L

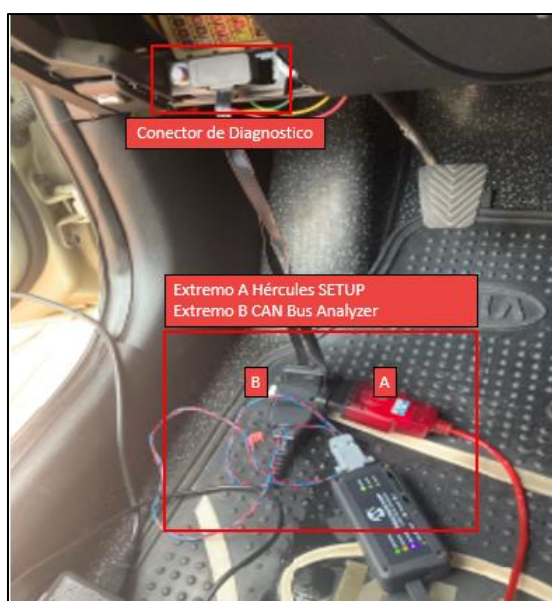


4.1.2 Resultado de la Conexión en Paralelo de Softwares

La conexión en paralelo de los softwares CAN Bus Analyzer y Hércules SETUP se realizó mediante un cable de conexión OBD II denominado “SPLITTER” 2 a 1 como lo indica la figura 32.

Figura 32

Conexión en paralelo de ordenadores.

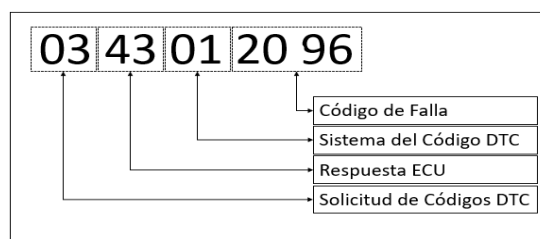


4.1.2.1 Software Hércules SETUP.

Este software se comunicó con la ECU mediante su interfaz, se envió un mensaje con el texto "03", este texto es la solicitud de información de códigos DTC enviados desde el ordenador hacia la ECU, la respuesta de la ECU es de manera inmediata, obteniendo como el mensaje "0343 01 20 96" como respuesta, en la figura 33 se muestra el análisis del texto.

Figura 33

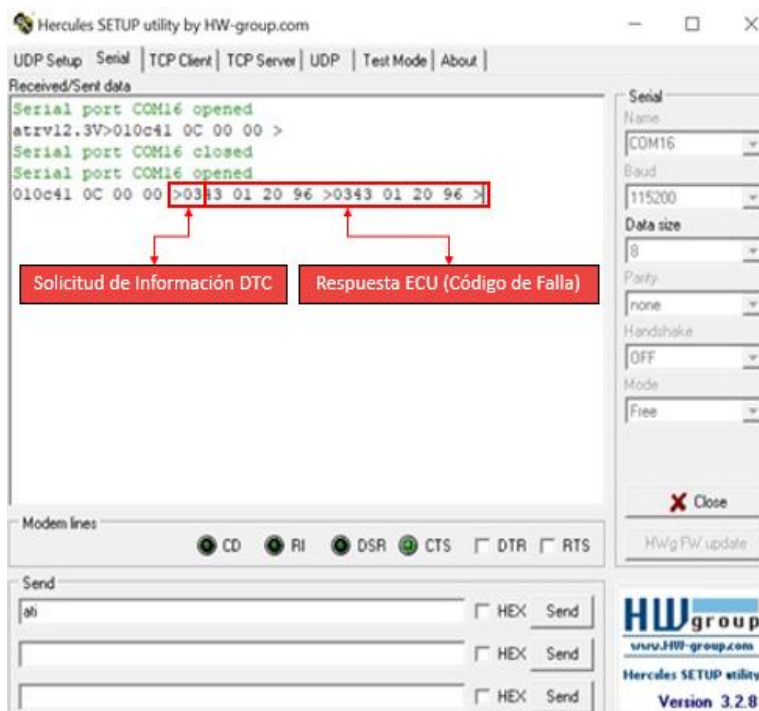
Interpretación de Mensaje de Respuesta



En la figura 34 se observa la apertura del puerto serial de comunicación entre la ECU y el Software con los mensajes de solicitud de información y respuesta a la solicitud.

Figura 34

Respuesta a la Solicitud de Información de códigos DTC

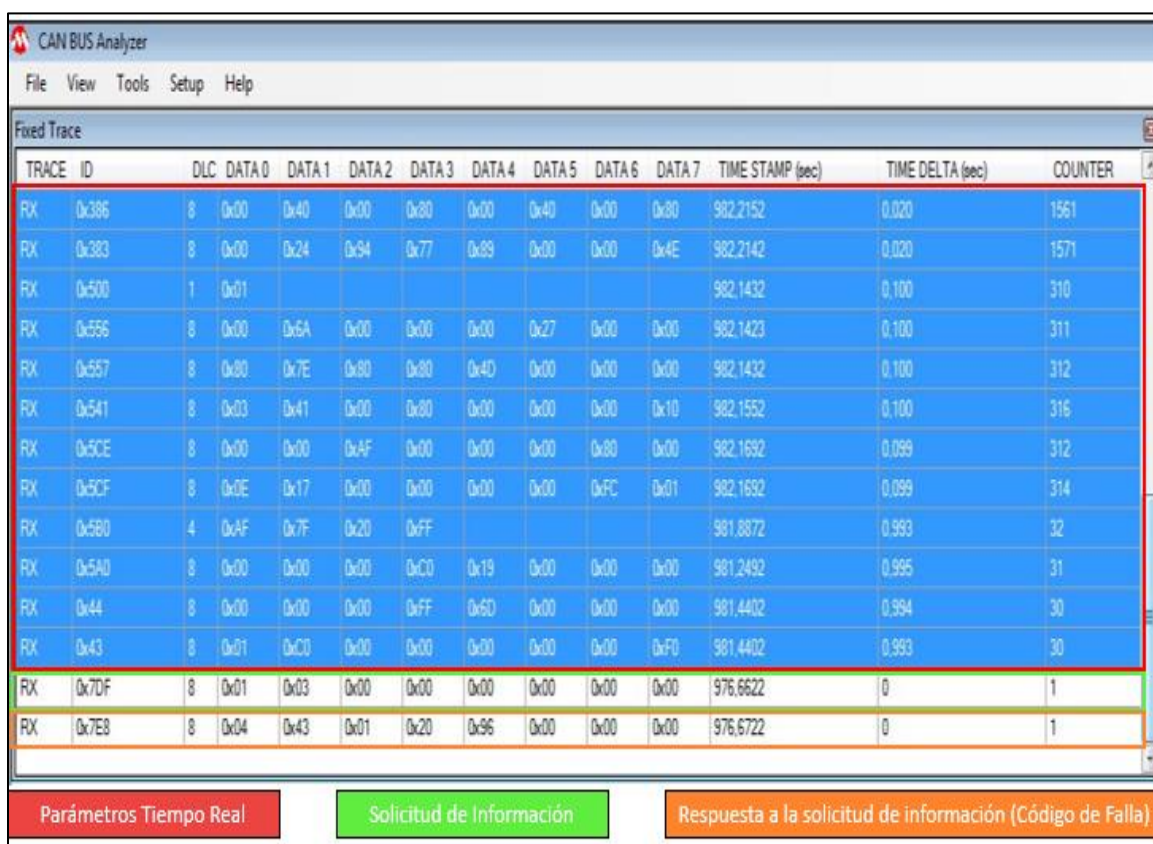


4.1.2.2 CAN Bus Analyzer.

Es un Software que dispone de una interfaz con funciones de configuración, rastreo, transmisión y registro de los parámetros, cuando la conexión del software es estable y segura se procede a la identificación del ID referente a la escritura para la solicitud de información y del ID en el que se lee la respuesta, para ello se marca los parámetros de la pantalla, se envía el mensaje "03" de solicitud de información desde Hércules SETUP, producto de la conexión en paralelo, CAN Bus Analyzer registro la solicitud enviada y la respuesta de la ECU, logrando la infiltración a la Red CAN y la utilización de los datos originados como lo indica la figura 35.

Figura 35

Envío y Recepción de Información



TRACE ID	DLC	DATA 0	DATA 1	DATA 2	DATA 3	DATA 4	DATA 5	DATA 6	DATA 7	TIME STAMP (sec)	TIME DELTA (sec)	COUNTER
RX 0x386	8	0x00	0x40	0x00	0x80	0x00	0x40	0x00	0x80	982.2152	0.020	1561
RX 0x383	8	0x00	0x24	0x94	0x77	0x89	0x00	0x00	0x4E	982.2142	0.020	1571
RX 0x500	1	0x01								982.1432	0.100	310
RX 0x556	8	0x00	0x6A	0x00	0x00	0x27	0x00	0x00	0x00	982.1423	0.100	311
RX 0x557	8	0x80	0x7E	0x80	0x80	0x40	0x00	0x00	0x00	982.1432	0.100	312
RX 0x541	8	0x03	0x41	0x00	0x80	0x00	0x00	0x00	0x10	982.1552	0.100	316
RX 0x5CE	8	0x00	0x00	0xAF	0x00	0x00	0x00	0x80	0x00	982.1692	0.099	312
RX 0x5CF	8	0x0E	0x17	0x00	0x00	0x00	0x00	0xFC	0x01	982.1692	0.099	314
RX 0x580	4	0xAF	0x7F	0x20	0xFF					981.8872	0.993	32
RX 0x540	8	0x00	0x00	0x00	0xC0	0x19	0x00	0x00	0x00	981.2492	0.995	31
RX 0x44	8	0x00	0x00	0x00	0xFF	0x60	0x00	0x00	0x00	981.4402	0.994	30
RX 0x43	8	0x01	0xC0	0x00	0x00	0x00	0x00	0x00	0xF0	981.4402	0.993	30
RX 0x7DF	8	0x01	0x03	0x00	0x00	0x00	0x00	0x00	0x00	976.6622	0	1
RX 0x7E8	8	0x04	0x43	0x01	0x20	0x96	0x00	0x00	0x00	976.6722	0	1

Parámetros Tiempo Real Solicitud de Información Respuesta a la solicitud de información (Código de Falla)

4.1.3 Resultado de Obtención de la Información

En la tabla 4 se extiende la información y los datos necesarios para la construcción del prototipo del dispositivo de extracción y almacenamiento de códigos de falla.

Tabla 4*Información Requerida*

Estado	Ruta	ID	DLC	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8
Escritura	Rx	0x7DF	8	01	03	00	00	00	00	00	00	00
Lectura	Rx	0x7E8	8	04	43	01	20	96	00	00	00	00

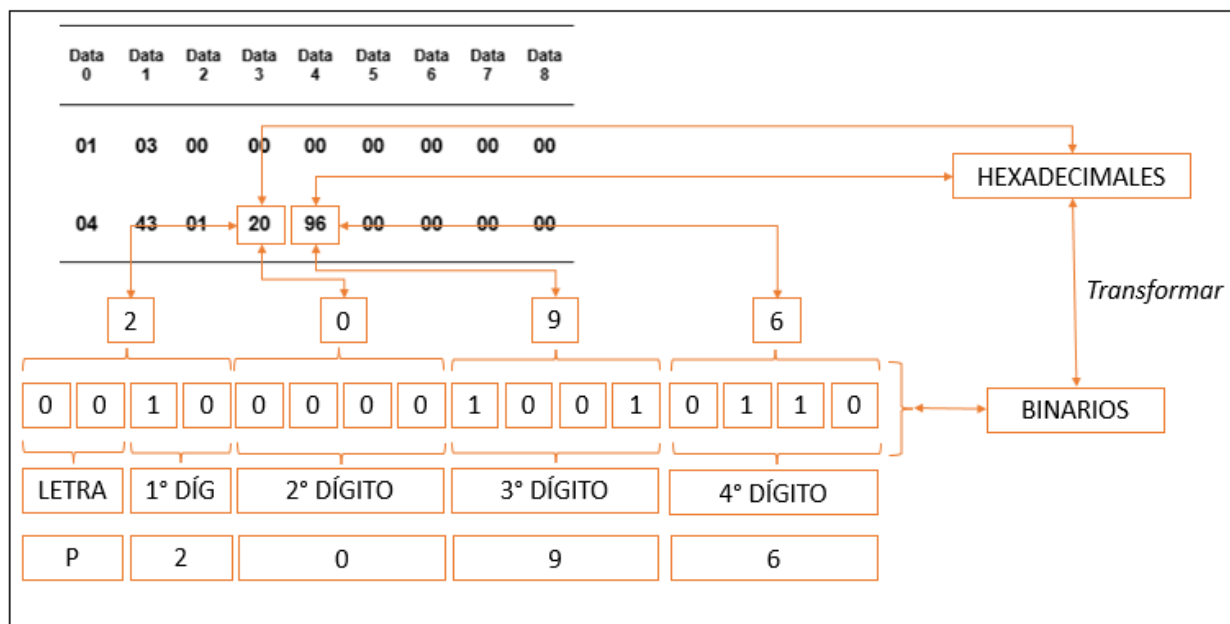
En la fila de estado escritura de la tabla 4 se analiza que, “Rx” pertenece a la ruta por el cual se realizará la comunicación y transmisión de datos entre dispositivos, “0x7DF” es el ID o header de escritura de datos, “8” DLC es la longitud total del mensaje, data 0 es la cantidad adicional de bytes utilizados “01”, data 1 es el modo de operación utilizado “03” que corresponde a los códigos PID estandarizados según el protocolo de comunicación OBD-II para obtención de códigos de falla, las datas del 2 al 8 son “00” para complementar la longitud del mensaje.

Por otro lado, en la fila de estado de lectura de la tabla 4 se analiza que, “Rx” pertenece a la ruta por el cual se realizará la comunicación y transmisión de datos entre dispositivos, “0x7E8” es el ID o header de lectura de datos, “8” DLC es la longitud total del mensaje, data 0 es la cantidad adicional de bytes utilizados “04”, data 1 es la respuesta al modo utilizado “43”, data 02 define la cantidad de códigos de falla “01”, data 3 y 4 corresponden al código de falla expresado como respuesta en mensaje hexadecimal, las datas del 5 al 8 son “00” para complementar la longitud del mensaje.

La estructura de los códigos DTC se especifican por área determinada del vehículo, para determinar el área se procede a transformar los valores hexadecimales de data 3 y data 4 a números binarios como indica la figura 36.

Figura 36

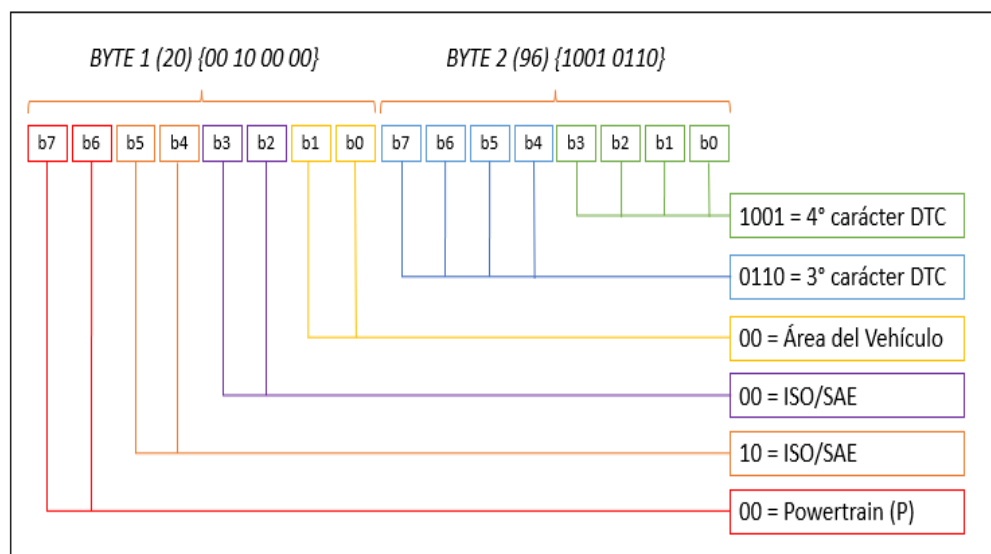
Transformación de Valores Hexadecimales a Números Binarios



La norma ISO 15031 proporciona la uniformidad estandarizada para los códigos de falla, en los valores binarios obtenidos se procede a identificar según la norma como lo indica la figura 37.

Figura 37

Norma ISO 15031 Aplicada a Data3 y Data 4



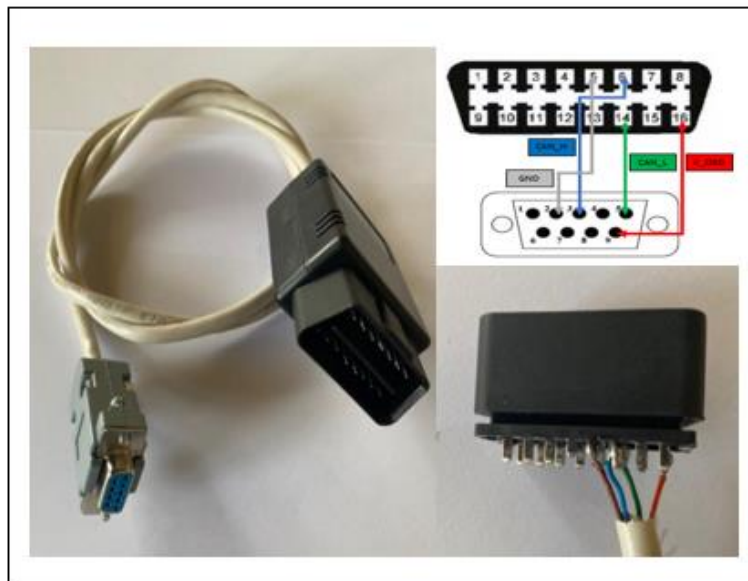
Todos los datos obtenidos son importantes para el desarrollo de la programación en el prototipo de extracción y almacenamiento de la información.

4.1.4 Resultado de Prototipo para la Extracción de la Información

En primera instancia se creó un cable de conexión que permite la comunicación CAN entre el conector de diagnóstico del vehículo y el dispositivo CAN Bus Shield V2, en la figura 38 podemos observar el cable con sus conectores OBDII y DBD9 al igual que el diagrama de conexión.

Figura 38

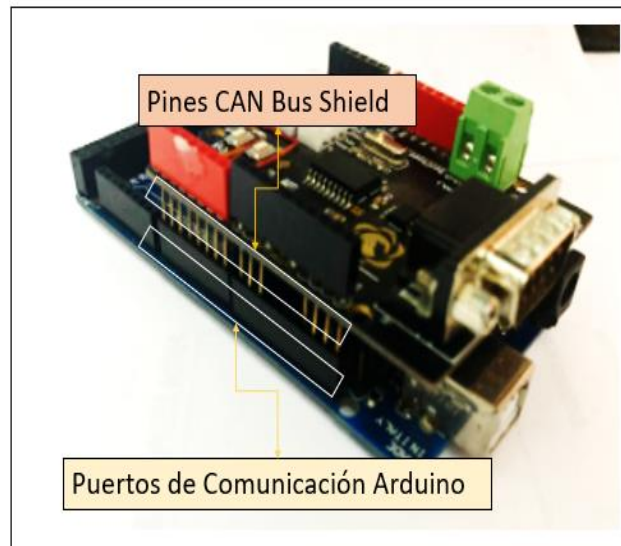
Cable de conexión OBD II – DB9



La tarjeta electrónica Arduino y la tarjeta electrónica CAN Bus Shield V2 son totalmente compatibles, permitiendo el ensamble entre ellas de forma sencilla por la arquitectura de sus puertos y pines de comunicación, en la figura 39 se puede observar el acople entre las dos tarjetas electrónicas.

Figura 39

Acople de Tarjetas Electrónicas Arduino – CAN Bus Shield



Para obtener los resultados de la figura 40, se debe disponer del testigo de alerta “Check Engine” encendido en el vehículo de prueba, con la llave en posición OFF , se ubicó el conector de diagnóstico y se procede a conectar el extremo del cable OBD II, seguidamente del extremo DB9 en la tarjeta electrónica CAN Bus Shield, al colocar la llave en la posición ACC, se encienden las luces de transmisión de datos indicando que los dispositivos están en funcionamiento, posteriormente, se conectó la tarjeta electrónica Arduino al ordenador y por medio de la opción Monitor de Serie del software, se comprueba que la información obtenida es la deseada.

Figura 40

Monitor Serie de Arduino – Código de Falla Detectado

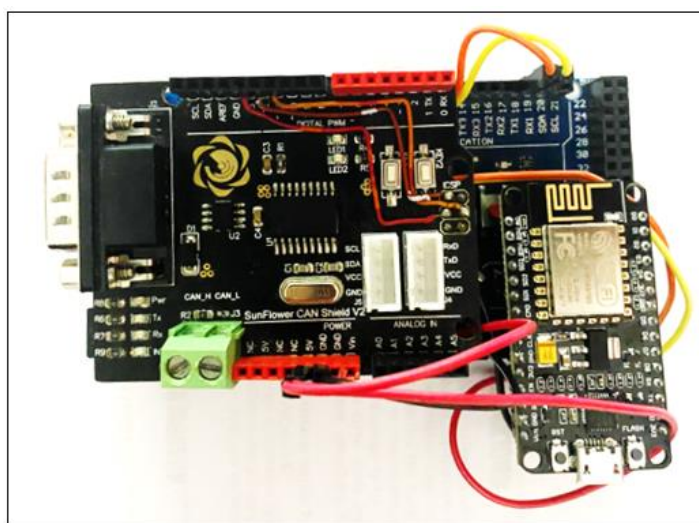


4.1.5 Resultado de Almacenamiento de la Información

Para enviar el código de falla utilizando el internet de las cosas IOT se acopla el módulo Node MCU al dispositivo como lo indica la figura 41.

Figura 41

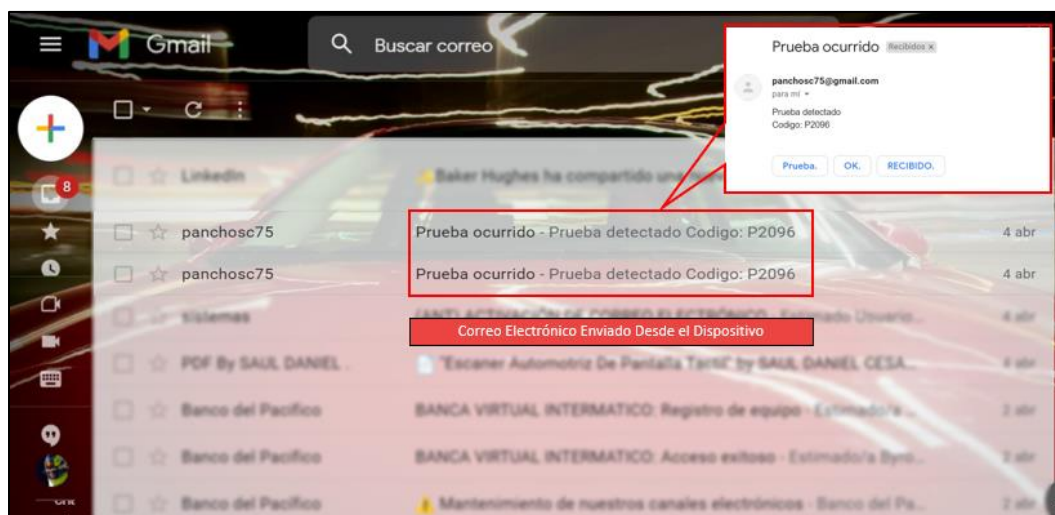
Adaptación del Módulo Node MCU a Arduino



Este módulo permitió enviar una solicitud URL con el código de falla, por medio de la plataforma IFTTT “IF THIS, THEN THAT” o en español SIEEE “SI ESTO, ENTONCES AQUELLO”, permitiendo crear y programar acciones y tareas en internet, dicho esto, uno de los receptores del URL enviado en la aplicación WEBHOOK, esta se encargó de recibir el URL con la información y para ser enviado inmediatamente por correo electrónico, en la figura 42 se observa el correo electrónico y dentro del mismo el código de falla.

Figura 42

Correo enviado con Información DTC.



5. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

La información primordial que se generó en la ECU por un código DTC pertenece al ID de escritura identificado como 0x7DF donde se escribe el mensaje hexadecimal para el modo diagnóstico y el ID de lectura identificado como 0x7E8 donde se recibe la respuesta hexadecimal, se transformó esta respuesta a números binarios y se determinó los valores para la verificación del área del código de falla según el protocolo CAN.

La programación en Arduino se realizó con funciones de envío, recepción y transmisión de paquetes de datos entre el dispositivo y la ECU, se visualizó el código de falla en el monitor serial del Arduino que finalmente mediante un correo electrónico se reflejó la capacidad de repuesta del dispositivo para procesar y enviar la información a la web de manera inmediata.

La IoT “Internet de la Cosas” promete una sociedad interconectada para monitorear y controlar acciones desde cualquier dispositivo que disponga de conexión a internet, la plataforma que se utilizó es IFTTT, la cual cumple con las condiciones de conexión del dispositivo que aplicó el IoT para la interconexión entre dispositivos a través de una red, donde se permitió que el código de falla generado en la ECU del vehículo viaje a través del internet de manera inmediata a un correo electrónico.

5.2 Recomendaciones

Para futuros estudios se recomienda añadir un dispositivo con red wifi de buena cobertura en el automóvil con la finalidad de obtener la información deseada en el mismo instante que el dispositivo entra en funcionamiento.

Diseñar una aplicación móvil que permita recibir notificaciones de códigos de falla, así como también, diferentes parámetros de operación en tiempo real, con el propósito realizar un mejor diagnóstico preliminar y remoto del vehículo.

Al ser un dispositivo de diagnóstico preliminar, se recomienda a autoridades y docentes fortalecer el presente trabajo, implementando al dispositivo los protocolos de comunicación existentes en la industria, con el fin de convertirse en una herramienta indispensable en los talleres de servicio automotriz.

6. REFERENCIAS.

- Abramowicz, H. (2017). *Estudio Para La Implementación De Equipos De Diagnóstico Electrónico En Un Taller Automotriz*.
<https://repositorio.uide.edu.ec/bitstream/37000/2371/1/T-UIDE-165.pdf>
- Agency United States Environmental, 2007. (2007). *Un resumen de la Ley de Aire Limpio*.
https://espanol.epa.gov/sites/default/files/2020-05/documents/050720_cleanairact_un_resumen_de_la_ley_de_aire_limpio.pdf?VersionId=5MvcYrbAuHTIXXBeAHDiP0ZENq7HUuji
- Agüero, M. J., Colman, M., Dubois, F., Rabadán, M., & Amarilla, S. (2015). *Sistema de interacción con la ECU de un automóvil empleando procesamiento de lenguaje natural*. October. <http://sedici.unlp.edu.ar/handle/10915/50538>
- AutoAvance. (2020). *Sistema de Red y Multiplexado Automotriz*. Blog Técnico Automotriz.
<https://www.autoavance.co/blog-tecnico-automotriz/22-sistema-de-redes-y-multiplexado/>
- Bloemen, J. (2021). *No Title*. Amplíe Sus Servicios Con Webhooks.
<https://www.ixon.cloud/es/base-de-conocimientos/amplie-sus-servicios-con-webhooks>
- Plan Nacional del Buen vivir 2017-2021, 1 Educational Research 150 (2017).
<http://www.planificacion.gob.ec/wp-content/uploads/downloads/2017/07/Plan-Nacional-para-el-Buen-Vivir-2017-2021.pdf>
- Delgado Planás, A. (2016). Sistema de protección de los intereses del autor y el empresario en la propiedad intelectual (régimen del empresario en el ámbito de la propiedad intelectual). *TDX (Tesis Doctorals En Xarxa)*, 1–555.
http://www.tdx.cat/handle/10803/400934%0Ahttps://www.tdx.cat/bitstream/handle/10803/400934/Tesis_Antonio_Delgado.pdf?sequence=1
- Denton, T. (2011). Automobile Mechanical and Electrical Systems - Automotive Technology: VEHICLE Maintenance and Repair. In *Statewide Agricultural Land Use Baseline 2015* (Vol. 1).
- FAE. (2019). *Reportaje técnico sobre sensores FAE para la gestión del motor*. FAE.
<http://www.fae.es/es/reportaje-tecnico-sobre-sensores-fae-para-la-gestion-del-motor-11-212>

- Gil-Solés, D. (2015). IFTTT : 10 usos en Unidades de Información. *ELis*.
<http://eprints.rclis.org/32774/>
- INECC. (2006). Fuentes De Emisión. *Inecc*, 25–34.
<http://www2.inecc.gob.mx/publicaciones2/libros/618/vehiculos.pdf>
- Juan Trujillo-Tello, J. T.-T., Padilla-Padilla, C., Buenaño-Moyano, L., & Cuaical-Angulo, B. (2020, September). Evolución y Tendencia de los Sistemas de Control de Motores de Combustión Interna Alternativos, Una Revisión Bibliográfica. *Ingenio*, 3(2), 5–14.
<https://doi.org/10.29166/ingenio.v3i2.2718>
- Kezherashvili, B. (2013). *Computación en la nube*. 1, 223–245.
- Kia Ecuador. (2021). *Especificaciones Equipamiento interior Equipamiento exterior*. Ficha Técnica. <https://www.kia.com/ec/shopping-tools/download-a-brochure.html>
- Oliveres, C. (2003). *Líneas actuales de investigación y desarrollo de la gestión electrónica de inyección de gasolina en el colector*. 9–64.
- On, O. B. D., & Diagnostics, B. (2005). *Diagnóstico a Bordo*.
https://www.gob.mx/cms/uploads/attachment/file/187221/diagnosticoabordo_1_260117.pdf
- Pérez, F. E. V., & Areny, R. P. (2007). *MICROCONTROLADORES : FUNDAMENTOS Y APLICACIONES CON PIC Autores : Fernando E . Valdés Pérez Ramon Pallàs Areny Composición y preimpresión : Índice general*. 12–13.
- Rovira Sebastian, P. W. (2022). Tecnologías digitales para un nuevo futuro. *Naciones Unidas*, 3(XXXX), 99.
- Salazar, R., & Ruiz del Solar, J. (2015). *Sensores, Actuadores y Efectos*. 1–41.
- Sánchez, L., Molano, M., Fabela, M., & Hernández, J. (2016). Revisión documental del protocolo CAN como herramienta de comunicación y aplicación en vehículos. *Instituto Mexicano Del Transporte*, 474(474), 1–69.
<https://imt.mx/archivos/Publicaciones/PublicacionTecnica/pt474.pdf>

7. ANEXOS

Figura 43

Diagrama de Conexión Eléctrica

