

Computer-to-vehicle linking system, Alexa auto

Sistema de enlace para el funcionamiento entre computador y vehículo, Alexa auto

Alexis Javier Villacis Ninasunta¹, Mariana Pinargote Basurto¹

¹Universidad Técnica Estatal de Quevedo, Facultad de Ingeniería, Departamento de Ingeniería Mecánica. Ecuador.

Cite as

Villacis Ninasunta AJ, Pinargote Basurto M. Computer-to-vehicle linking system, Alexa auto. SAP Transport, Mobility & Society. 2025; 4:202. <https://doi.org/10.56294/tms2025202>

Publication History

Submitted: 28-03-2025

Revised: 26-06-2025

Accepted: 09-10-2025

Published: 10-10-2025

ABSTRACT

The automotive industry has seen the need to improve the comfort of the user by integrating intelligent voice-controlled systems. This aims to prevent potential distractions while driving and enhance the user experience on their journeys. Nowadays, voice assistants that interact with specific vehicle information are exclusive to high-end models or specific brands. However, there are other voice assistant alternatives available on the market such as the Echo Auto, which is designed for use in vehicles allowing access to the functionalities of the Alexa voice assistant. Conversely, its functions are limited to access to entertainment and information applications. Considering this context, this work presents the development of a system that allows extracting information from sensors of the Toyota Hilux CS 2020 vehicle, such as the status of the fuel system, ignition timing, coolant temperature and the air-fuel ratio. It also includes control of the activation of the door locks of a Mazda CX-3 vehicle. To do this, an MCP2515 CAN bus controller has been used to extract or write information from the bus and an Arduino Nano microcontroller to process it and send it to the cloud using a WIFI module. The Blynk IoT platform is responsible for receiving this data or sending it to the microcontroller and presenting it in an interface so that the user can monitor it. Through an Alexa skill programmed in Voiceflow, it is possible to access this data or modify it using voice commands given by the user. The main limitations encountered stem from the characteristics of the CAN bus in the vehicle, as not all the modules for function control are always connected to the main bus, for this reason it was necessary to use different vehicles.

Keywords: Echo Auto; Alexa; CAN bus; Vehicle Information; Function Control.

RESUMEN

La industria automotriz ha visto la necesidad de mejorar el confort del usuario integrando sistemas inteligentes controlados por voz. Esto busca prevenir distracciones mientras se conduce y mejorar la experiencia del usuario en sus travesías. Actualmente, los asistentes de voz que interactúan con información específica del vehículo son exclusivos de modelos de alta gama o marcas específicas. Existen otras alternativas de asistentes de voz disponibles en el mercado como el Echo Auto, que está diseñado para uso en vehículos permitiendo acceder a las funcionalidades del asistente de voz Alexa. Sin embargo, sus funciones se limitan al acceso a aplicaciones de entretenimiento e información. En este contexto, este trabajo presenta el desarrollo de un sistema que permite extraer información de sensores del vehículo Toyota Hilux CS 2020, como el estatus del sistema de combustible, el tiempo de avance, la temperatura del refrigerante y la relación aire-combustible. También incluye el control de la activación de los seguros de las puertas de un vehículo Mazda CX-3. Para ello, se ha utilizado un controlador de bus CAN MCP2515 para extraer o escribir información del bus y un microcontrolador Arduino Nano para procesarla y enviarla a la nube mediante un módulo WIFI. La plataforma IoT Blynk se encarga de recibir estos datos o enviarlos al microcontrolador y presentarlos en una interfaz para que el usuario pueda monitorearlos. A través de una skill para Alexa programada en Voiceflow, es posible acceder a estos datos o modificarlos mediante comandos de voz dados por el usuario. Las principales limitaciones encontradas se deben a las características del bus CAN en el vehículo, ya que no siempre los módulos para el control de funciones están conectados al bus principal, por esta razón fue necesario utilizar vehículos diferentes.

Palabras clave: Echo Auto; Alexa; Bus CAN; Información del Vehículo; Control de Funciones.

INTRODUCCIÓN

Antecedentes

Con la llegada de la Industria 4.0 y la introducción de nuevas tecnologías digitales en el diario vivir, se produjo un cambio en el modo en el que el usuario interactúa con las máquinas. Como lo explica Ortiz et al.⁽¹⁾, las necesidades de facilitar la búsqueda de información y el avance del Internet de las Cosas han favorecido al desarrollo de los asistentes virtuales de voz, y a su vez han modificado la forma en que el usuario realiza una búsqueda o solicita información. En este caso se prioriza el uso de comandos de voz para lograr una interacción más rápida en Internet, de modo que el usuario no tenga la necesidad de usar las manos o vista.

Entre las opciones más conocidas en el mercado están Alexa de Amazon, Google Assistant, Cortana de Microsoft, Siri de Apple, Bixby de Samsung, entre otros. Su uso se destaca principalmente para aplicaciones domóticas por su compatibilidad con una amplia gama de sensores y actuadores. En cuanto a las opciones destinadas a su uso dentro del vehículo se encuentran asistentes derivados de los anteriores tales como Echo Auto, Android Auto, Carplay, JBL Link Driv, Anker Roav Bolt y otras alternativas.

Las funcionalidades a las que tienen acceso estos asistentes dependerán de sus características específicas, así como del modelo del vehículo. Puesto que, varias marcas como BMW, Audi o Mercedes-Benz cuentan con sus propios asistentes virtuales, que permiten al usuario el acceso a información del vehículo e incluso a funciones de control específicas; como regular la temperatura dentro de la cabina, operación de limpiaparabrisas, cerradura de puertas y más, entre los asistentes más actuales están ¡Hey Mercedes! (Mercedes), Laura (Skoda), Ok Honda (Honda), etc.⁽²⁾ Mientras que el resto de los asistentes se limitan a funciones más comunes que no tienen relación directa con el estado del vehículo o su control, y mucho menos cuentan con un software libre que permita ampliar los comandos de acceso o manejo de información dentro del vehículo.⁽³⁾

De este modo actualmente, importantes empresas como Google, Apple, Microsoft, Amazon, e incluso reconocidas marcas de vehículos, invierten capitales importantes para integrar la tecnología de voz en sus productos, y naturalmente se incluye el área automotriz para lograr un mayor confort en los usuarios.⁽⁴⁾ Incluso cada vez son más los modelos de vehículos que ya vienen con esta tecnología incorporada de fábrica, mientras que para los que no la poseen o modelos anteriores se desarrollan dispositivos enfocados para funcionar en estos entornos, tales como Echo Auto de Amazon o Android Auto. Los cuales han mejorado sus capacidades para el reconocimiento de voz en lugares ruidosos.

En el caso de Amazon en su página oficial presenta una lista de vehículos en los cuales ya se ha integrado Alexa al vehículo. Entre las marcas y modelos más reconocidos se destacan Audi, BMW, Chevrolet, Ford, Toyota, Nissan, entre otras. Sin embargo, las listas únicamente incluyen modelos de los últimos años, en su mayoría desde 2019.⁽⁵⁾ Y sus funcionalidades de control sobre el vehículo son limitadas y dependerán de las características de hardware y software que posea tal vehículo. En adición, en 2021 Amazon anunció Alexa Custom Assistant que permite a fabricantes de dispositivos y proveedores de servicios crear sus propios asistentes sobre la tecnología de Alexa y así integrarla desde su fabricación.⁽⁶⁾

La mayoría de los dispositivos que realizan una conexión con el vehículo a través de su puerto de OBDII (OnBoardDiagnostics) se limitan a tipos de escáneres para la detección de averías. Los cuales pueden contar con sus propias pantallas para la visualización de información o requieren de una aplicación móvil que se conecta principalmente vía bluetooth. El más conocido en el mercado y de bajo costo es el ELM327, el cual es un dispositivo de diagnóstico utilizado para leer y analizar códigos del vehículo, que informan de algún problema.⁽⁷⁾

Los aplicativos móviles compatibles con el ELM327 como Infocar, Car Scanner, OBD Auto Doctor, entre otras, disponibles en Play Store, permiten visualizar varios datos como velocidad, temperatura, nivel de combustible, etc.⁽⁸⁾ Sin embargo, estos dispositivos no cuentan con opciones de control que se puedan procesar por comandos de voz, ya que su función principal es diagnosticar el estado del vehículo. Por otra parte, esto ofrece una oportunidad importante ya que se cuenta con una interfaz, donde se pueda verificar rápidamente la información que se obtenga al construir una interfaz propia.

En cuanto a la conexión del microcontrolador al asistente de voz, la función principal es la comunicación y enlace entre la lectura de variables del puerto OBDII del vehículo con el asistente de voz y su capacidad de proporcionar de manera audible esta información al usuario.⁽⁸⁾ Actualmente, existen varias plataformas que permiten realizar esta conexión tales como Blynk, Particle, Home Assistant, entre otras, de este modo el usuario puede dar comandos de voz para acceder a datos de sensores o control de actuadores que se encuentren conectados al microcontrolador. Lo que posibilita llevar a cabo la conexión desde el vehículo al microcontrolador y de este hacia el asistente virtual para lograr los fines propuestos.

¿Cómo desarrollar una solución de hardware y software que permita integrar asistentes de voz, específicamente Alexa Auto, con el puerto OBDII de un vehículo para controlar funciones y acceder a variables en tiempo real, superando las limitaciones actuales de compatibilidad y control presentes en los asistentes y dispositivos comerciales?

Objetivo

Desarrollar hardware y software que permita la interacción con Alexa auto para controlar funciones y leer variables del vehículo.

MÉTODO

Diseño del estudio

Se llevó a cabo un estudio aplicado de desarrollo tecnológico con evaluación experimental y descriptiva. La fase de ingeniería (prototipado de hardware–software) se complementó con ensayos de validación en un solo grupo con medidas repetidas para: (a) precisión de lectura de variables de sensores vía OBDII/CAN y (b) efectividad de activación de actuadores mediante comandos de voz.

Población, escenario y materiales

- Vehículos ensayados: Toyota Hilux CS 2020 (lectura de sensores estándar SAE J1979) y Mazda CX-3 2019 (activación

de actuadores en bus CAN secundario).

- Interfaz vehicular: Puerto OBDII (pines CAN_H 6, CAN_L 14; buses de 500 kbps y 125 kbps según modelo).
- Hardware: Arduino Nano (ATmega328), controlador/transceptor CAN MCP2515/TJA1050, módulo WiFi ESP01 (ESP8266), regulación DC-DC (MP1584 → 5 V; LM1117 → 3,3 V), cable OBDII y extensión dúplex.
- Asistente y plataformas: Echo Auto (Alexa), skill diseñada en Voiceflow; backend IoT Blynk para lectura/escritura de variables; app PC “CANSniffer” para inspección de tramas en Mazda.
- Entorno de prueba:
 - Vehículo en reposo (motor encendido, sin marcha).
 - Vehículo en movimiento ($\approx 30\text{--}40$ km/h, ventanas cerradas).
 - Se registraron condiciones operativas de referencia (ruido ambiental aproximado, rpm).

Variables

- Independientes/controladas: condición del vehículo (reposo vs movimiento), tipo de comando (consulta vs acción), intervalo de muestreo (500 ms para envío a Blynk), filtros CAN, calidad de enlace (WiFi/Bluetooth).
- Dependientes (sensores en Toyota):
 - Estado del sistema de combustible (PID 03).
 - Temperatura del refrigerante (PID 05, °C).
 - Tiempo de avance de encendido (PID 0E, ° antes del PMS).
 - Relación aire-combustible (AFR) derivada de λ (PID 24). Precisión cuantificada como % de error respecto de un escáner de referencia ELM327 conectado en paralelo.
- Dependiente (actuadores en Mazda): Cumplimiento de orden (éxito/fracaso) para bloqueo/desbloqueo de seguros de puertas; se reportó efectividad (%).

Procedimiento

1. Implementación HW/SW: Ensamblaje del circuito (placa PCB dedicada) y conexión OBDII–MCP2515–Arduino Nano (SPI)–ESP01 (UART). Inicialización de bus CAN (500 kbps o 125 kbps según caso) y filtros por ID.
2. Lectura de sensores (Toyota):
 - Envío de solicitudes OBDII (ID 0x7DF, Servicio 0x01, PID según variable).
 - Recepción/parseo de respuestas (ID 0x7E8), decodificación y normalización a unidades físicas.
 - Transmisión a Blynk cada 500 ms para consumo por la skill de Alexa.
 - Validación: ELM327 en paralelo mediante cable dúplex; por cada variable y condición (reposo/movimiento) se realizaron 5 mediciones; se calculó error puntual y promedio.
3. Activación de actuadores (Mazda):
 - Identificación empírica/por ingeniería inversa de tramas CAN relevantes (apoyo con CANSniffer y documentación comunitaria).
 - Diseño de intents en la skill; actualización de variables virtuales en Blynk; propagación al ESP01 y publicación de la trama CAN desde Arduino Nano.
 - Validación: 10 intentos por acción (bloquear/desbloquear) con el vehículo en modo parking; registro de cumplimiento.
4. Robustez de reconocimiento de voz: ejecución de los comandos en ambos escenarios (ruido moderado en reposo y en movimiento) para observar sensibilidad práctica del canal de voz.

Análisis de datos

- Sensores: estadística descriptiva; comparación frente a ELM327; cálculo de error (%) por intento y error promedio por variable y condición.
- Actuadores: proporción de cumplimiento y efectividad total (%) por tipo de orden.
- Criterios de aceptación: errores ≤ 5 % para variables continuas y efectividad ≥ 80 % para acciones por voz se consideraron adecuados para la prueba de concepto.

Consideraciones éticas y de seguridad

- Seguridad operativa: pruebas de conducción a baja velocidad y rutas controladas; activación de actuadores solo en modo parking para evitar riesgos; no se alteraron sistemas críticos de seguridad.
- Integridad del vehículo: uso de mensajes estándar SAE J1979 para lectura; para escrituras en Mazda se limitaron a tramas documentadas/no críticas; sin modificaciones permanentes ni invasivas.
- Privacidad y datos: no se recolectó información personal identificable; los tokens de Blynk y credenciales se gestionaron de forma restringida; no se almacenaron datos sensibles del usuario.
- Cumplimiento normativo: respeto de normativas de tránsito vigentes y buenas prácticas de laboratorio; el sistema se evaluó como prototipo experimental, no destinado a uso comercial sin certificaciones adicionales.

RESULTADOS Y DISCUSIÓN

Diseño del sistema de comunicación Vehículo - Alexa Auto Introducción

En esta etapa se detalla el diseño tanto del hardware como de software. Como primera parte se realiza una propuesta general del hardware y los componentes a ser utilizados, y lograr la comunicación de la ECU del vehículo con un microcontrolador que se encargará del procesamiento de los datos para ser enviados por WIFI a la nube.

Para ello se parte del diagrama de la figura 1, que indica la conexión entre los diferentes componentes desde el vehículo al asistente de voz. En cuanto al software que se requiere, se amplía la información sobre la estructura y contenido de los mensajes CAN que se deben utilizar para lograr la comunicación con el bus de datos. Una vez obtenidos los datos se requiere de la configuración de las plataformas IoT Blynk y Voiceflow, las cuales se utilizan para almacenar las variables leídas desde el vehículo y para el desarrollo de la skill de Alexa respectivamente.

Diseño del hardware utilizado

En la figura 1, se puede observar el diagrama de bloques de los componentes de hardware utilizados para el sistema de enlace entre computador de vehículo y Alexa Auto, partiendo desde la conexión con el vehículo hasta la comunicación WIFI con el asistente de voz.

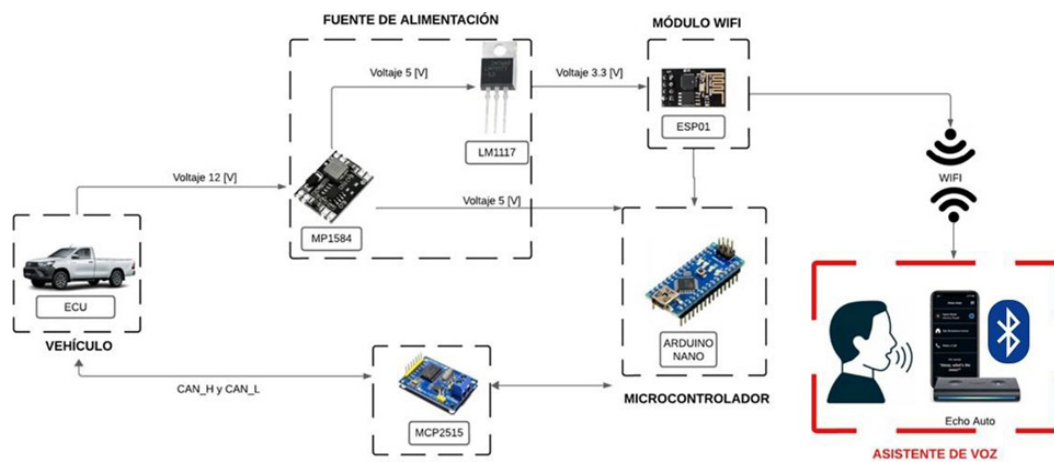


Figura 1. Diagrama de bloques del hardware para la interconexión del computador del vehículo y Alexa Auto

Posteriormente, se describen las especificaciones de cada elemento que conforma cada bloque del diagrama mencionado.



Figura 2. Designación de pines del puerto OBDII con uno o dos Bus CAN⁽⁹⁾

Especificaciones técnicas de los componentes en el Vehículo

Es importante detallar la funcionalidad de los componentes y dispositivos esenciales para la implementación del sistema de enlace entre el computador del vehículo y el asistente de voz Echo Auto. Estos componentes incluyen el cable OBDII y los vehículos con los sensores o actuadores utilizados en este trabajo.

Vehículos utilizados

Para establecer la conexión con la computadora del vehículo, es fundamental conocer el protocolo al que se rige el puerto OBDII, ya que este puede variar según el fabricante del vehículo, y esta información brindará los criterios básicos para la selección de los componentes electrónicos necesarios para el sistema general.

En ocasiones, los fabricantes incorporan uno o dos buses CAN en un mismo vehículo para distinguir los módulos que interactúan con la computadora del vehículo, tal como se muestra en la figura 2, donde los pines 3 y 11 también son designados como Bus CAN (Object Detection), posibilitando la comunicación con los módulos instrumentales del vehículo, es decir, con los actuadores. Asimismo, los pines 6 (CAN_H) y 14 (CAN_L), definidos por la norma ISO 15765-4, se encargan de conectar los módulos principales del vehículo.

En el caso de la camioneta Toyota Hilux CS 2020, se determinó que únicamente cuenta con un bus CAN, lo que implica que los módulos para el control de vidrios, luces, seguros de puertas u otros, son módulos independientes que no se conectan directamente a la ECU. Por tanto, se optó por otra alternativa que es un Mazda CX3, el cual si cuenta con dos buses CAN; el bus principal utilizado para módulos importantes en el funcionamiento del vehículo y un bus secundario para módulos como el control de clima, audio, seguros de puertas, entre otros.

Toyota Hilux CS 2020

Es una camioneta de marca Toyota, la cual es conocida por su calidad, durabilidad y capacidad todoterreno. Tiene un diseño de gran robustez al incorporar un paragolpes frontal, una renovada parrilla y lámparas con LED para mejorar la visibilidad del entorno. Se puede apreciar la imagen del vehículo descrito, así como sus características técnicas. En donde adicionalmente, se puede observar el puerto OBDII del vehículo, del cual se identificaron los pines habilitados. Estos se destacan con un recubrimiento metálico y corresponden a los pines: 4, 5, 6, 7, 13, 14 y 16. Según la información proporcionada en la figura 3, este vehículo cuenta con un bus CAN de alta velocidad (500 kbps).

En la figura 3, se puede apreciar una representación del bus CAN con varios de los módulos que se conectan a este. Este diagrama está construido en base a un diagrama del sistema de comunicación CAN de otro modelo de vehículo Toyota⁽¹⁰⁾, puesto que tal información es limitada respecto al modelo específico que se está utilizando, sin embargo, entre modelos se mantienen cierta similitud que resulta de gran utilidad.

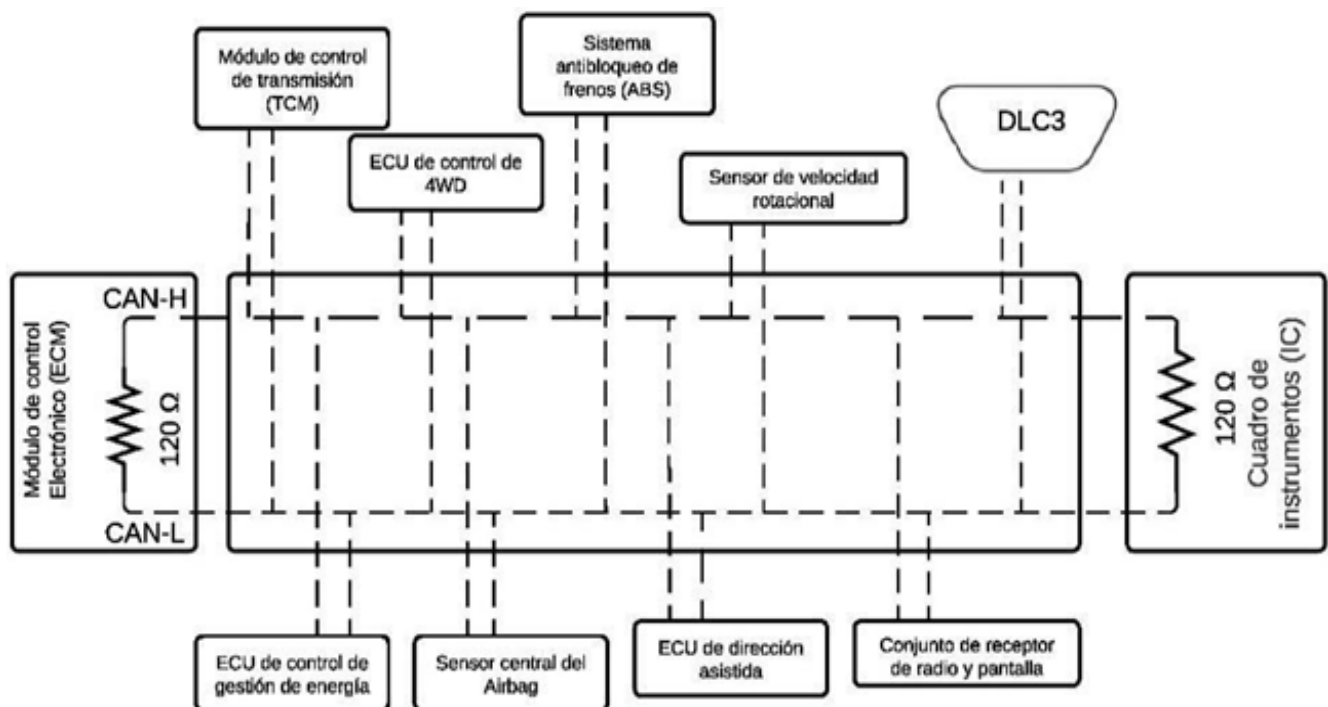


Figura 3. Bus CAN del vehículo Toyota Hilux CS 2020⁽¹⁰⁾

Como se mencionó anteriormente, en este vehículo se va a realizar la lectura de sensores, los cuales se describen a continuación:

Sensor de oxígeno

Convierte la cantidad de oxígeno de los gases que produce el automóvil en una señal eléctrica, luego la Unidad de Control de Motor (ECU) lee esa señal para comparar y ver si la mezcla aire-combustible es la ideal o adecuada. Si hay mucho oxígeno en los gases de escape, aumenta el tiempo de inyección y de esta forma se obtiene la potencia optima del motor, de manera que se obtenga una relación estequiométrica de 14,7 partes de aire, por una parte de combustible.⁽¹¹⁾ Se puede observar este sensor y varias de sus características técnicas.

Sensor de temperatura del líquido refrigerante

Este sensor informa a la computadora central la temperatura del refrigerante, el modelo físico se observa, junto con sus características técnicas. Este permite verificar si el motor está frío o caliente, y la temperatura en la que se encuentra, para ajustar la inyección y el arranque del vehículo. Su funcionamiento básico se basa en la variación de una resistencia interna dependiendo de la temperatura medida.⁽¹²⁾

Sensores de posición

Para medir el tiempo de avance o avance de encendido, se necesita la información de dos sensores de posición. El primero será el sensor de posición del cigüeñal, el cual detecta la posición del eje del cigüeñal y mide el número de revoluciones de giro del motor.⁽¹³⁾ Se puede observar este sensor junto a sus características técnicas.

El segundo es el sensor de posición del árbol de levas, que se utiliza para monitorear la posición y la velocidad de rotación del árbol de levas.⁽¹⁴⁾ Se detalla las características respectivas del sensor de árbol de levas. La combinación de datos de ambos sensores permite a la computadora del vehículo calcular y ajustar el tiempo de avance del encendido del vehículo.

Sistema de combustible

El sistema de combustible, como se observa en la figura 4, está formado por el depósito de combustible, la bomba, el filtro, los inyectores y regulador de presión, que se encargan de suministrar el combustible al motor. Una vez que se acciona, este sistema indica al computador del vehículo que se encuentra encendido, apagado, presenta alguna falla, entre otras opciones. Dentro del contexto mencionado se va a utilizar el parámetro proporcionado a la ECU por este sistema para conocer el estatus general del auto una vez que se ha encendido.

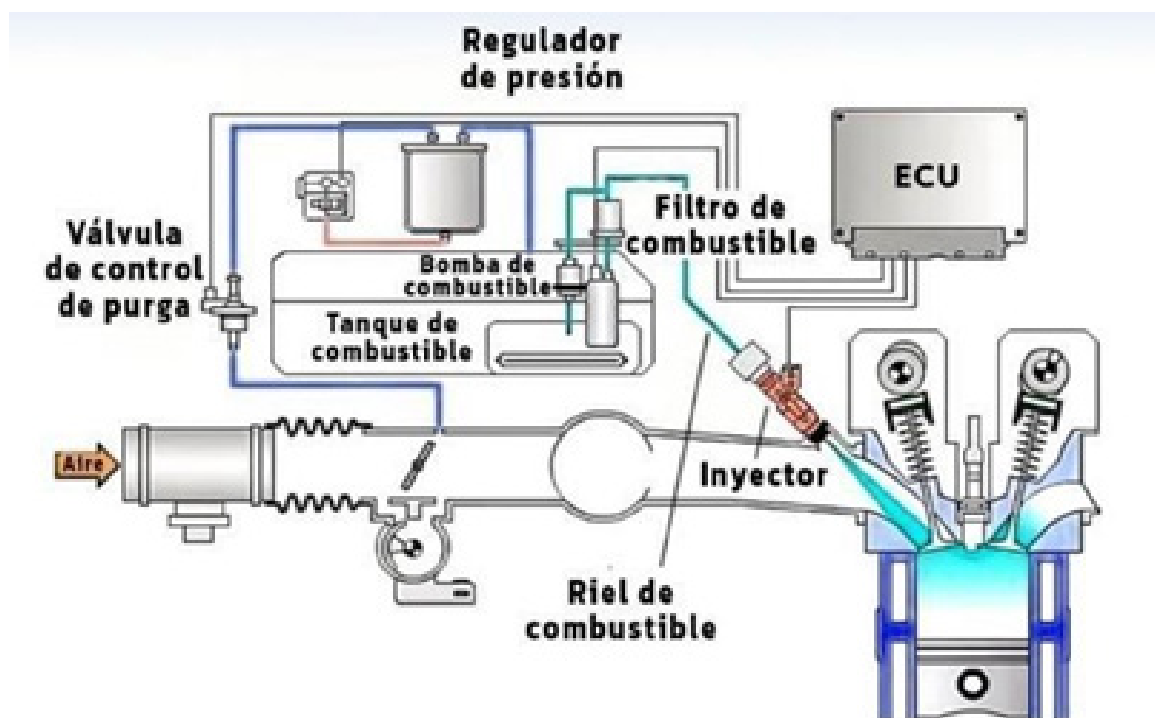


Figura 4. Representación general de un sistema de combustible

Mazda CX-3 2019

El Mazda CX-3 2019 es un vehículo compacto que ofrece una variedad de características donde destaca su diseño, calidad interior y desempeño deportivo, con buenas características de aceleración y manejo. Cuenta con un sistema de entretenimiento con pantalla táctil, alarma con sensor de intrusión, asistencia electrónica de frenado, entre otras características.⁽¹⁴⁾ Se presenta la imagen del vehículo descrito y también varias de sus características técnicas. También se puede apreciar el puerto OBDII del vehículo, del cual se identificaron los pines habilitados. Estos corresponden a los pines: 3, 4, 5, 6, 8, 11, 14, 15 y 16. Según la información proporcionada en la figura 15, este vehículo cuenta con dos buses CAN, lo que da posibilidad de acceso a los sensores y ciertos actuadores del vehículo conectados a la ECU.

En este caso el vehículo Mazda CX-3 cuenta con dos buses CAN uno de alta velocidad (500 kbps) y otro de menor velocidad (125 kbps). En la figura 5 y 6, se puede apreciar una representación de estos buses. Cabe destacar que este diagrama está construido en base a la información obtenida del software de diagnóstico ForSCAN que es capaz de leer todos los módulos del vehículo conectados a la red CAN. Y de igual manera con información adicional del diagrama de la red CAN de un MAZDA CX-5,⁽¹⁴⁾ el cual es un modelo que comparte ciertas características con el Mazda CX-3.

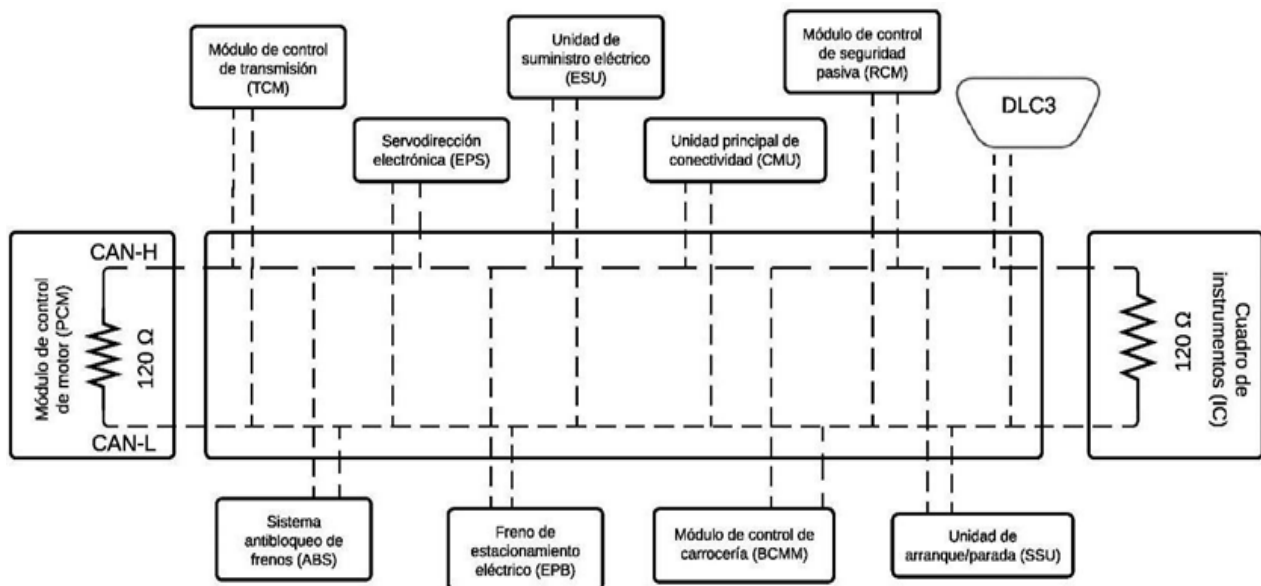


Figura 5. Bus CAN (500 kbps) del vehículo Mazda CX-3 2019

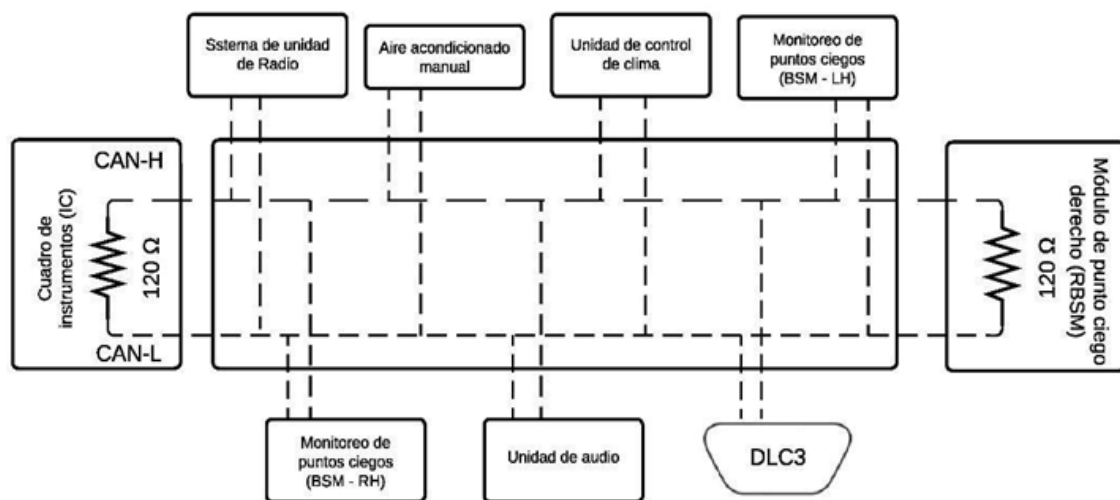


Figura 6. Bus CAN (125 kbps) del vehículo Mazda CX-3 2019

Debido a estas características en este vehículo se va a realizar la activación de un actuador, en este caso el seguro de las puertas, el cual se describe a continuación:

Actuador eléctrico para la cerradura de puertas

Este sistema permite mover la cerradura de la puerta del vehículo con el fin de acceder o no a él, como se observa. La señal de control viene en un sistema eléctrico que puede estar conectado a un botón, una llave o a la computadora del vehículo.⁽¹⁵⁾

Cable de extensión para conector OBDII

Para evitar cualquier molestia u obstrucción del sistema que se debe conectar al puerto OBDII, se hace uso de un cable de extensión conectado a este puerto. Se muestra el cable utilizado y sus características. Está diseñado según las especificaciones de la SAE J1962. Con una longitud de aproximadamente 60 cm, permite conectar el sistema a un lado del conductor puesto que normalmente el puerto OBDII se encuentra bajo el volante o cerca de este, por tanto, es necesario que no obstruya ningún lugar que el conductor utilice mientras maneja.

Especificaciones técnicas de la Fuente de alimentación

El circuito de alimentación se compone de dos partes. En primer lugar, se realiza la regulación de la fuente de voltaje proveniente del vehículo, reduciéndola de 12[V] DC a 5 [V] DC, este nivel de voltaje es esencial para alimentar el microcontrolador. En segundo lugar, se lleva a cabo la regulación de los 5 [V] resultantes a 3,3 [V], que es el voltaje requerido para suministrar energía al módulo WIFI ESP01.

Regulador de voltaje fijo DC/DC (MP1584)

El convertidor DC-DC MP1584 es un regulador de tipo conmutado reductor (Step- Down o Buck) con una alta eficiencia de conversión, excelente regulación de línea y bajo voltaje de rizado,⁽¹⁶⁾ el componente físico se puede observar con sus características técnicas. Se eligió este regulador de voltaje en función de los requerimientos de alimentación del microcontrolador Arduino Nano. Estableciendo un voltaje de salida de 5[V] fijo para prevenir posibles variaciones que podrían dañar los componentes electrónicos.

Transistor LM1117

Es un regulador de tensión positiva de tres terminales, su tensión de salida regulada fija es de gran ayuda en aplicaciones estándares de electrónica. Aunque se ha diseñado principalmente como regulador de voltaje fijo, se puede utilizar con componentes externos para obtener voltajes y corrientes ajustables.⁽¹⁷⁾ Se configuró específicamente para proporcionar una salida de 3,3 [V] con el propósito de alimentar el módulo WIFI ESP01. Se encuentran las características técnicas del transistor.

Especificaciones técnicas del Controlador de bus CAN: Módulo CAN MCP2515

El módulo CAN MCP2515 permite comunicar distintos dispositivos mediante el protocolo CAN. Este módulo tiene la capacidad de recibir y enviar paquetes de datos en formato estándar y extendido, aunque posee máscaras y filtros de acceso, que reducen la carga del microcontrolador principal.⁽¹⁷⁾ Cabe recalcar que incluye el chip controlador CAN MCP2515 con interfaz SPI y el chip transceptor CAN TJA1050 con el fin de facilitar la comunicación con microcontroladores y tarjetas de desarrollo, en este caso Arduino. Se puede apreciar el módulo físico mencionado y sus características técnicas.

Especificaciones técnicas del Microcontrolador: Arduino Nano

Arduino Nano es una placa microcontroladora basada en el ATmega328, muy característica por su tamaño reducido en comparación de otras placas Arduino, como se puede observar. Se caracteriza por ser compatible con gran variedad de componentes electrónicos, especialmente diseñada para trabajar en protoboard y facilitar el prototipado de los circuitos o proyectos que dispongan de espacio limitado.⁽¹⁸⁾

Especificaciones técnicas del Módulo WIFI ESP01

El módulo WIFI ESP01 integra el Sistema en un chip (SOC, System On a Chip) ESP8266 y compatible con Arduino. Integra un potente procesador con arquitectura de 32 bits y conectividad Wifi, permitiendo trabajar como host de aplicaciones o reducir la carga de WiFi Networking de otro procesador, siendo apto para aplicaciones IoT.⁽¹⁹⁾ Puede ser utilizado en una variedad de aplicaciones, como el control remoto de dispositivos, la recopilación de datos, el monitoreo y control de dispositivos IoT, entre otros.⁽²⁰⁾ Se puede observar el módulo físico y se encuentran las características técnicas del módulo.

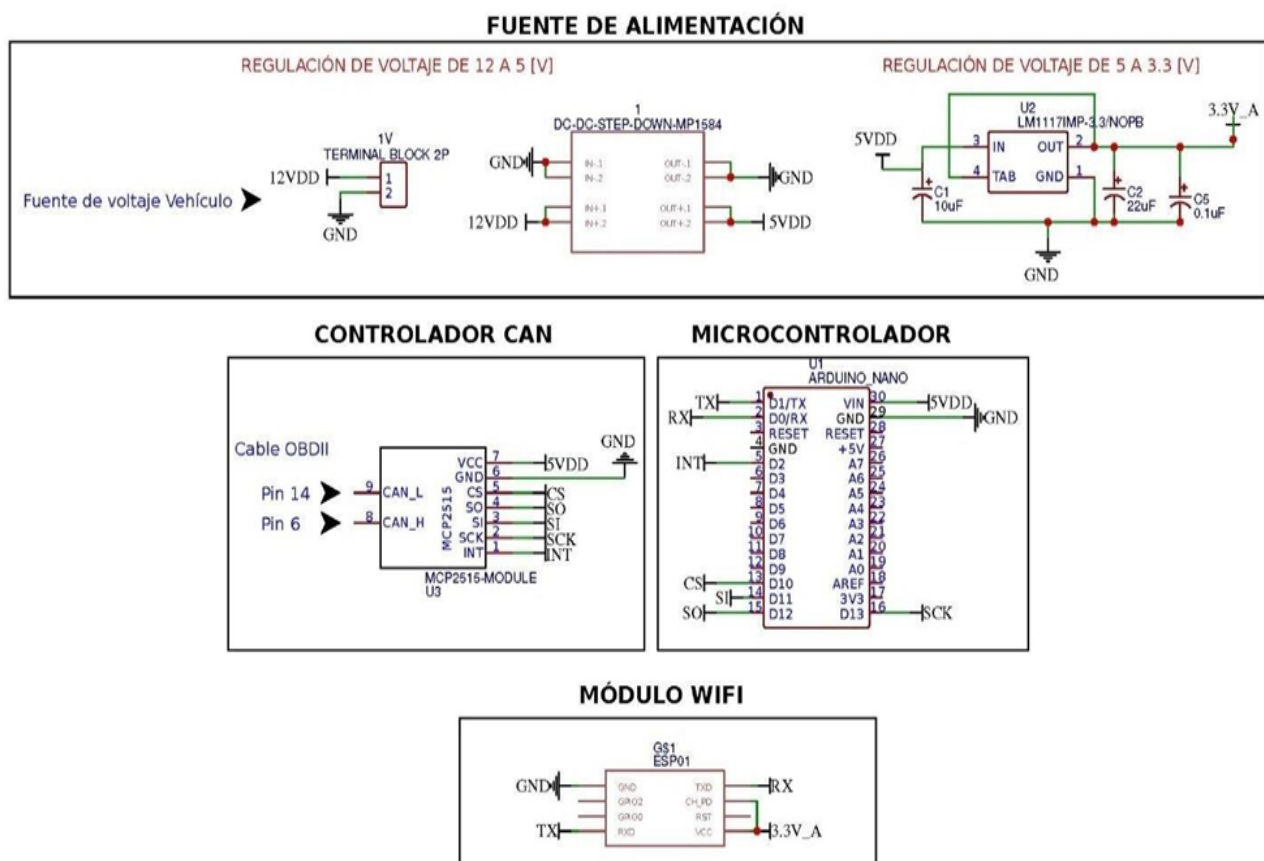


Figura 7. Circuito esquemático del sistema de enlace Alexa Auto y el computador del vehículo

Esquema general de hardware para la conexión del vehículo a un microcontrolador

Según el diagrama de bloques presentado en la figura 7, se implementó el esquemático de la figura 7, donde se observan cada uno de los componentes principales del circuito.

Especificaciones técnicas del Asistente de voz: Echo Auto

El Echo Auto de Amazon Alexa, como se observa, es un asistente de voz que se conecta por Bluetooth a la aplicación móvil Amazon Alexa. Se emplea para recibir las solicitudes del usuario y procesar la información deseada a través de la programación de la skill específica de este proyecto. De este dispositivo se puede destacar su integración de 8 micrófonos, que lo hace apto para aplicaciones con mucho ruido, lo cual es un factor importante durante el reconocimiento de comandos de voz.

En la primera sección del circuito, se aborda la regulación de la fuente de voltaje proveniente del vehículo, que suministra 12 [V], destinada a alimentar todo el sistema. Utilizando el regulador MP1584, se genera un voltaje de salida de 5 [V] usado por el microcontrolador Arduino Nano y MCP2515. Adicionalmente se emplea este voltaje para alimentar el transistor LM1117, que está configurado para ajustar su voltaje de salida a 3,3 [V], con el propósito de suministrar energía al módulo WIFI ESP01.

Para conectar el Arduino Nano con el módulo MCP2515 se emplea una comunicación SPI, conectando los pines respectivos (CS, MISO, MOSI, SCK e INT). De igual manera en el módulo MCP2515, se lleva a cabo la conexión con el bus CAN del vehículo por medio de los pines CAN_H y CAN_L respectivamente. Para finalizar, se realiza la conexión entre el microcontrolador Arduino Nano y el módulo WIFI ESP01, utilizando una comunicación serial entre los dos componentes, por medio de los pines RX y TX.

Después de verificar el correcto funcionamiento y las respectivas conexiones de todos los componentes electrónicos, se procede con el diseño de una placa de circuito impreso (PCB). Esta tiene dimensiones de 73,66 mm de ancho por 73,53 mm de largo. Muestra el diseño de la placa PCB para la interconexión de todos los componentes. Asimismo, se indica el mismo diseño en un modelo tridimensional, permitiendo visualizar las ubicaciones precisas de cada elemento.

Finalmente, se ha puesto en funcionamiento este circuito y se ha realizado la medición respectiva corriente del circuito, dando un valor promedio de:

$$I = 146,2 \text{ [mA]}$$

Identificación de mensajes CAN utilizados en el protocolo OBDII

Para establecer la comunicación con el bus CAN es necesario conocer la estructura que tienen los mensajes CAN y los servicios a los que se puede acceder en este protocolo. Como parte del campo de datos, la norma⁽²¹⁾ define 9 servicios de diagnóstico, que se deben especificar en el mensaje de consulta. Estos servicios se detallan, de los cuales el que se utilizará para la lectura de datos de los sensores es el servicio 0x01 que permite solicitar información de diagnóstico actual. Aparte de estos 9 servicios, en abril de 2021 la SAE presento una sucesión de la SAE J1979, denominada SAE J1979-2 que especifica 26 servicios unificados de diagnóstico (UDS, Unified Diagnostic Services) adicionales, empezando en la dirección 0x10 hasta 0x2F.⁽²²⁾ Para realizar una consulta o respuesta estándar en el bus CAN del vehículo, se utilizan direcciones funcionales. Para una consulta de diagnóstico se utiliza el ID 0x7DF, el cual es escuchado por la ECU y da su respuesta con el ID 0x7E8. Estas especificaciones están definidas en la norma SAE J1979, así como también la estructura que debe tener el campo de datos del mensaje y la estructura de la respuesta respectiva. Para conocer los IDs de los parámetros (PID) disponibles en la computadora de la camioneta Toyota, se realiza una solicitud del PID 00. El cual permite mostrar los PID compatibles con el vehículo, en un formato de 4 bytes de datos, donde cada uno de los bits que conforman la respuesta, representa uno de los siguientes 32 PID establecidos en la SAE J1979, especificando si ese PID es compatible o no. La solicitud realizada es la siguiente; 0x7DF020100, donde los bytes enviados al bus CAN se detallan en la tabla 1, a continuación:

Tabla 1. Solicitud de PID 00 – Mostar PID disponibles

Bytes de datos				
Identificador de 11 bits (Solicitud)	Número de bytes de datos adicionales	Servicio (01 = mostrar datos actuales)	Código PID (00 = Mostrar PIDs)	(00 = No usados (ISO 15765-2 sugiere CCh)
7DF	02	01	00	

La respuesta de esta solicitud es la siguiente; 0x7E8064100BE1FA813, donde los bytes recibidos corresponden a los parámetros especificados en la tabla 2.

Tabla 2. Respuesta de PID 00 – Mostar PID disponibles

Bytes de datos										
Identificador de 11 bits (Respuesta)	Número de bytes de datos adicionales	Servicio de consulta, mas 40h (41h = mostrar datos actuales)	Código PID (00 = Mostrar PIDs)	Dato byte 0	Dato byte 1	Dato byte 2	Dato byte 3	No usado		
7E8	06	41	00	BE	1F	A8	13			

Utilizando los bytes de la respuesta anterior, se procede a decodificarlos en un total de 32 bits. Se detalla cada uno de estos bits y

su relación con los PID respectivos. Con lo cual es posible determinar que PIDs están disponibles o no en el rango de 0x01 a 0x20.

Siguiendo esta misma metodología, se procede a realizar dos solicitudes adicionales de los PID 0x20 y PID 0x40, para conocer qué otros parámetros se encuentran disponibles en el vehículo en el rango de 0x21 a 0x40 y de 0x41 a 0x60 respectivamente. Con las respuestas obtenidas de estos PIDs, se han decodificado y se detallan todos los PID que son compatibles con el vehículo Toyota Hilux CS 2020.

Para el caso del Mazda CX3, la identificación de los ID de los actuadores conlleva un proceso diferente. Una opción es utilizar el hardware diseñado anteriormente para leer los datos del bus CAN y presentarlos en una interfaz que los agrupe y ordene para un mejor análisis. En este caso se utiliza la aplicación de código abierto “CANSniffer”, disponible en GitHub.⁽²³⁾ Esta aplicación de escritorio está diseñada específicamente para la lectura de mensajes CAN de un puerto COM del computador. Para esto se conecta el microcontrolador al computador vía USB y se imprime en el puerto serial todos los mensajes que se lean desde el bus CAN. De este modo la aplicación se conectará al puerto serial y empezará con la presentación de los mensajes en la interfaz presentada en la figura 8:



Figura 8. Interfaz gráfica de CANSniffer

En esta interfaz se pueden acceder a diversas funciones como la agrupación de paquetes, filtrado de IDs, decodificación de paquetes y entre otras. Una vez que se ha iniciado la aplicación de escritorio, el proceso para la identificación de los actuadores es empírica o por ingeniería inversa, ya que se procede a accionar algún actuador del vehículo manualmente y se verifica en la interfaz los paquetes nuevos que se han desplegado. Al repetir este proceso varias veces se pudo identificar el mensaje CAN que está relacionado con la activación de un actuador específico. Este proceso es necesario ya que no se cuenta con información o documentación oficial que especifique los paquetes CAN que están asociados con los diferentes de actuadores del vehículo.

Una alternativa no oficial para obtener más información sobre los mensajes CAN asociados a configuraciones de marcas como Mazda son los foros de usuario, los cuales proporcionan información valiosa respecto a la decodificación de los mensajes. Donde incluso se ha encontrado disponible una plantilla de Excel en la que se han identificado varios paquetes de datos. También cuenta con herramientas para la conversión de binario a hexadecimal en el formato que se debe enviar el mensaje al bus CAN, haciendo mucho más sencillo y accesible la modificación de ciertos parámetros de la ECU.⁽²⁴⁾

Selección de variables

Partiendo de la información previamente detallada, se ha realizado la selección de las siguientes variables, tanto para sensores en función de los PIDs disponibles en el vehículo Toyota, como para los actuadores según la información extraída de los foros de usuario para Mazda.

Sensores del vehículo Toyota Hilux CS 2020

En función de los datos presentados, se han seleccionado las siguientes variables detalladas en la tabla 3, relacionados a los sensores del vehículo.

Tabla 3. Solicitud de PID 00 – Mostar PID disponibles

PID	Descripción	Min	Max	Fórmula	Unidades
03	Estado del sistema de combustible	-	-	-	Tabla 5
05	Temperatura del refrigerante del motor	-40	215	A – 40	°C
0E	Tiempo de avance	-64	63,5	A() – 64 2	° Antes del PMS
24	Sensor de oxígeno 1 (Relación Aire- combustible)	0	<2	2/(256A+B) 65536	Ratio
	Sensor de oxígeno 1 (Voltaje)	0	<8	8/(256C+D) 65536	V

PID 03 – Estado del sistema de combustible

El PID 03 da como respuesta dos bytes de datos, el primero describe el sistema de combustible principal del vehículo y el segundo describe sistemas de combustible secundarios si existen. Los valores de datos que puede tomar cada byte se describen en la tabla 4:

Tabla 4. Estados del sistema de combustible

Valor	Descripción
0	El motor está apagado
1	Circuito abierto debido a temperatura insuficiente del motor
2	Circuito cerrado, utilizando retroalimentación del sensor de oxígeno para determinar la mezcla de combustible
4	Circuito abierto debido a la carga del motor o corte de combustible debido a la desaceleración
8	Circuito abierto debido a una falla del sistema
16	Circuito cerrado, usando al menos un sensor de oxígeno, pero hay una falla en el sistema de retroalimentación

En condiciones normales y cuando el auto está encendido, el valor que se debe obtener es 2, lo que indica que el sistema de combustible se encuentra en correcto estado y utiliza el sensor de oxígeno para determinar la mezcla de combustible.

PID 05 – Temperatura del refrigerante del motor

Muestra la temperatura del refrigerante del motor. La medición de esta variable se utiliza en el cálculo de la mezcla de aire y combustible. Y la relación que se mantiene es inversamente proporcional, ya que a menor temperatura se requiere de una mayor cantidad de combustible y viceversa. Como se especifica en HELLA⁽¹²⁾ la temperatura del refrigerante debe estar entre 85°C a 95°C cuando el motor está en el régimen de plena carga y entre 95°C a 110°C con carga parcial para un motor de 1,6L. Cabe destacar que estos rangos pueden variar un poco según el motor.

La respuesta de la solicitud del PID 05, tiene 1 byte de datos que al transformarse a número decimal se puede representar un rango de temperatura entre -40°C y 215°C.

PID 0E – Tiempo de avance

El tiempo de avance o avance de encendido, corresponde al momento en que se debe encender la mezcla aire-combustible para lograr una máxima presión de combustión después del punto muerto superior (PMS). Para ello la ignición se debe producir en un ángulo de giro del cigüeñal aproximado entre 5° a 10° antes del PMS. Este rango va a depender de las características físicas del motor y de la velocidad a la que se encuentre.⁽²⁵⁾ Si este valor es elevado, la chispa se producirá antes de lo necesario y ocasionará que la mezcla se encienda de manera descontrolada, provocando lo que se conoce como detonación. Y sus consecuencias pueden ser daños en los pistones, bielas, entre otros componentes, y también afecciones en la eficiencia del motor. En el caso contrario si el tiempo de avance se retrasa o se produce después del PMS, ocurre una combustión incompleta de la mezcla, afectando a la potencia del motor y su rendimiento ya que no se aprovecha la máxima cantidad de energía liberada en la combustión.⁽²⁶⁾

La solicitud del PID 0E da como respuesta 1 byte de datos con el cual se puede describir un rango entre -64° a 63,5° antes del PMS. Y para este caso se ha considerado como un rango normal entre 0° y 25° como se sugiere en CERVANTES GAS⁽²⁶⁾ en función de las pruebas realizadas para diferentes modelos de vehículos.

PID 24 – Sensor de oxígeno 1

El sensor de oxígeno permite determinar la composición de los gases de escape, midiendo la concentración de oxígeno en el tubo de escape. La señal que proporciona este sensor se utiliza para determinar la relación óptima entre aire-combustible. Según sea el valor de la relación, se puede determinar si una mezcla es rica o pobre. Si hay mayor cantidad de aire que combustible en la mezcla se considera una mezcla pobre, y caso contrario si existe mayor cantidad de combustible que aire la mezcla se considera rica.⁽²⁷⁾ La solicitud del PID 24 da como respuesta 4 bytes de datos, los dos primeros bytes A y B se utilizan para el cálculo de la relación de

equivalencia aire-combustible denominado lambda (λ). Según las especificaciones de la SAE J1979 los valores que puede tomar λ están entre 0 y menores que 2.

Para esta aplicación se utiliza el factor λ , el cual si es $\lambda > 1$ indica que se tiene una mezcla pobre, y si $\lambda < 1$ la mezcla es rica. Y cuando el valor es de $\lambda = 1$ se considera que la relación de aire-combustible (AFR, Air-Fuel Ratio) es exactamente la necesaria para una combustión eficiente, considerada como mezcla estequiométrica. En el caso de los motores a gasolina esta relación aire-combustible es de 14.7:1.⁽²⁸⁾ De este modo para determinar la relación aire-combustible (AFR) se utiliza la fórmula:

$$\lambda = \frac{AFR}{AFR_{\text{estequimetrica}}} \quad (1)$$

De donde se obtiene que:

$$AFR = \lambda \times AFR_{\text{estequimetrica}} = \lambda \times 14.7 \quad (2)$$

Valores de AFR menores a 14,7, indica que se tiene una mezcla rica y valores mayores indican una mezcla pobre. Dependiendo del funcionamiento del motor, el valor de AFR puede variar, sin embargo, como se explica Fernández F⁽²⁸⁾, cuando se tiene una carga y aceleración moderada la relación debe acercarse a la relación estequiométrica (AFR = 14.7:1), si se tiene mayor carga o potencia máxima esta relación puede llegar hasta AFR = 13:1, y caso contrario si la carga y aceleración es baja la relación puede ser hasta AFR = 16:1. Valores mayores o menores en este rango pueden indicar que existe algún problema en el funcionamiento del vehículo.

Actuadores del vehículo Mazda CX-3

Para la selección del actuador que se pretende activar se hace uso de una plantilla de Excel desarrollada por Litnevskyi⁽²⁹⁾, donde ha organizado gran parte de la información de los paquetes de datos que se manejan en el bus CAN para vehículos Mazda. En este se detallan cada uno de los módulos que puede tener el vehículo y los bloques de datos en cada uno. Adicionalmente, ha decodificado varios de los códigos asociándolos a variables del vehículo o actuadores que posee, como ajuste de luces automáticas, bloqueo automático de puertas, entre otros.

Para esta aplicación se ha seleccionado la opción que permite activar o desactivar el seguro de las puertas. Para lo cual se ha identificado los mensajes que se debe enviar al bus CAN:

- Para la desactivación o desbloqueo de puertas se utiliza el siguiente mensaje, que está asociado al módulo con ID 7B7, en el bloque 02 y la primera trama, el mensaje completo es: 0x7B70201400000000001.
- Para la activación o bloqueo de puertas se utiliza un mensaje dirigido a la misma dirección o ID, cambiando los bits que corresponden para la activación del actuador respectivo, el mensaje completo es: 0x7B7020110000000000D1

Según la información proporcionada por Litnevskyi⁽²⁹⁾ estos mensajes mantienen la siguiente estructura general, que se presenta en la tabla 5, para uno de los mensajes anteriores:

Tabla 5. Estructura de mensaje CAN para activación de seguros de puertas en Mazda								
ID	N° Bloque		N° de mensaje	Bytes de datos			Suma de verificación	
07	B7	02	01	40	00	00	00	01

En este tipo de mensajes cada bit de los bytes de datos está asignado o relacionado con una función específica del vehículo. Debido a la limitada información o datos oficiales, se desconoce el funcionamiento de la gran mayoría, por lo cual no se recomienda modificar estos datos sin tener la certeza de su función.

Diseño de software para lectura y escritura de datos en el computador del vehículo

Para la lectura y escritura de mensajes en el bus CAN del computador del vehículo; se hace uso de la librería “Arduino-CAN”. Esta es compatible con el módulo MCP2515 y varios modelos de placas Arduino. Cuenta con las funciones necesarias para escribir los mensajes en el bus, así como también para leer y filtrar los mensajes que se encuentran en este. De este modo es posible realizar solicitudes en el bus para la lectura de datos de los sensores del vehículo, y cuando el módulo respectivo devuelve una respuesta con el valor de la variable deseada es posible filtrar el mensaje correspondiente para procesarlo. Por otra parte, para la activación de actuadores, en lugar de realizar una solicitud de datos, únicamente se envía al bus el mensaje respectivo que especifica la información necesaria para la activación del actuador deseado.

Funciones generales de la librería Arduino-CAN

Para llevar a cabo la comunicación con el bus CAN, la librería “Arduino-CAN” cuenta con diversas funciones que permiten, tanto la lectura como la escritura de un mensaje. A continuación, se describen varias de las funciones necesarias para lograr este propósito:

- `CAN.begin(500E3)`: con esta función se inicializa el bus CAN a 500 kbps, que es la velocidad estándar utilizada para un bus de comunicación de alta velocidad, la cual se utiliza en el vehículo para la interconexión de sistemas importantes.
- `CAN.filter(ID)`: esta función permite añadir un filtro, para recibir únicamente mensajes que tengan un ID específico. De este modo únicamente se leerán los mensajes que tengan ese ID y no todos los mensajes que se encuentren en el bus.
- `CAN.beginPacket(ID)`: inicializa un nuevo paquete o mensaje CAN con el ID especificado. Para la lectura de datos se utiliza el ID 0x7DF que indica que se está realizando una solicitud de información.
- `CAN.write(Data)`: envía un mensaje CAN al bus, con el ID previamente especificado.

Con esta función se deben especificar cada uno de los parámetros o datos del mensaje byte por byte.

- `CAN.endPacket()`: finaliza el paquete que se escribió en el bus CAN.
- `CAN.parsePacket()`: esta función verifica si hay algún mensaje nuevo en el búfer de recepción, y si está disponible para ser leído.
- `CAN.read()`: permite realizar la lectura de los mensajes del bus CAN como una estructura definida que almacena los datos que conforman el mensaje.

Utilizando estas funciones se procede a diseñar el código que permita la lectura de varias variables del vehículo y de igual manera para la activación de alguno de los actuadores disponibles en este.

Lectura de variables de sensores del vehículo

En la figura 9 se describe el procedimiento para realizar una solicitud de información a la ECU; para la lectura de variables de los sensores conectados al vehículo. En primera instancia se realiza una inicialización del bus CAN a la velocidad de 500 kbps, y se agrega un filtro para leer únicamente los mensajes de los datos de las variables que se solicitan. Utilizando la función `CAN.filter(0x7E8)`, que se encarga de filtrar los mensajes que posean este ID, el cual corresponde a las respuestas de una solicitud de información.

A continuación, es necesario definir el PID de la variable a leer y también la longitud de datos (DLC) del mensaje que se escribirá en el bus. Con esta información se da inicio a un nuevo mensaje utilizando la función `CAN.beginPacket(0x7DF)`. Seguidamente, se escriben los datos en el bus, donde se especifican el PID de la variable que se desea, así como también otros parámetros necesarios para el mensaje y se lo finaliza con la función `CAN.endPacket()`.

Una vez escrito el mensaje en el bus, se espera a que llegue un mensaje de respuesta y que corresponda al ID de filtro previamente configurado; lo cual es realizado por la función `CAN.parsePacket()`. Y cuando se tienen un mensaje disponible, se procede a realizar una verificación para garantizar que el mensaje corresponde a la información solicitada.

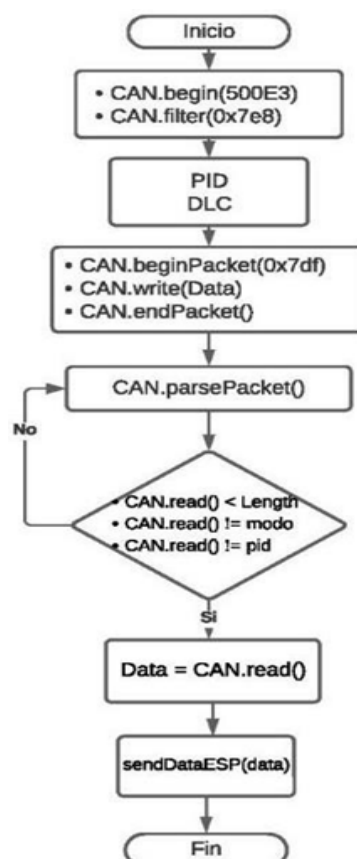


Figura 9. Diagrama de flujo para la lectura de información de sensores de la ECU utilizando Arduino Nano

En este proceso se verifica que la longitud del mensaje, el PID y el servicio utilizado sean los correctos, caso contrario se mantiene en la espera del mensaje correcto. Y cuando la verificación es exitosa, se procede con la lectura de los datos del mensaje, los cuales corresponderán a las variables de los sensores que se solicitó. Finalmente, se envían estos datos a la ESP01, por medio de una comunicación serial. Donde se los procesará y serán enviados a la nube de la plataforma Blynk.

Este procedimiento se realiza en bucle para la lectura de cada una de las variables de los sensores del vehículo. Donde los únicos parámetros que se modifican cada vez que se ejecuta; son el PID y el DLC ya que depende de cada variable deseada. El código respectivo para el Arduino Nano se encuentra disponible.

En el módulo WIFI ESP01, se realiza una lectura y procesamiento de los datos enviados por el puerto serial del Arduino Nano. Este proceso se presenta en la figura 10, en la cual se inicia con la configuración para la conexión con la plataforma Blynk y también se utiliza la función `timer.setInterval()` para establecer el intervalo de tiempo en el que se ejecutaran ciertas funciones, tales como la lectura de datos o el envío de los mismo a la plataforma Blynk.

A continuación, se realiza la lectura de los datos que se han recibido en el puerto serial; donde cada trama de datos que se recibe cuenta con un identificador que corresponde a cada una de las variables seleccionadas. Cabe señalar que los datos recibidos no cuentan con ningún procesamiento previo, es decir, corresponden a los datos hexadecimales que dio como respuesta la ECU. Por lo tanto, una vez que se leen y se identifican, se los procesa según corresponda el caso, ya que para las variables de los sensores sus datos deben ser transformados a una escala conocida. Finalmente, los valores obtenidos para cada sensor son enviados a la plataforma Blynk. Para lo cual se hace uso de la función `Blynk.virtualWrite(V0, Dato)`, la cual que permite escribir los datos de cada sensor en las variables virtuales de la plataforma.

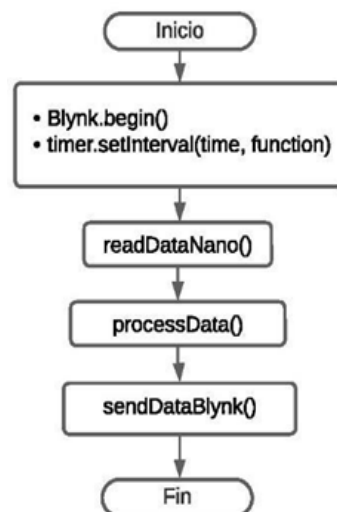


Figura 10. Diagrama de flujo para lectura de datos de puerto serial del módulo ESP01 y envío a plataforma Blynk

En este caso la función que se encarga de la lectura de los datos del puerto serial, se ejecutará cada 500 ms, lo cual permitirá actualizar los datos constantemente en la plataforma Blynk. El código respectivo en Arduino para el módulo WIFI.

Activación de actuadores del vehículo



Figura 11. Diagrama de flujo para monitoreo de cambios en variables virtuales

En el caso de los actuadores, el usuario es quien da la orden para la activación de alguno. Para esto la skill se encarga de registrar la orden y asociarla a una variable específica, ya sea para la activación o desactivación. Estas variables se crean en la plataforma

Blynk, y desde la skill se realiza una llamada a la API de Blynk para actualizar el valor de una de las variables asociada a un actuador específico.

En la figura 11, se muestra el proceso implementado en el módulo ESP01 cuando se desea activar algún actuador. Como primer paso se inicializa la comunicación con la plataforma Blynk utilizando la función `Blynk.begin()`, a continuación, el programa utiliza la función `BLYNK_WRITE(V0)` para monitorear algún cambio en una variable específica; en este caso la variable virtual V0. De este modo una vez que la skill actualiza el valor de la variable, se ejecuta esta función que se encarga de enviar por medio de comunicación serial, la trama con el ID del actuador que debe ser activado. Y nuevamente quedará en espera de algún cambio en alguna variable virtual de la plataforma Blynk. El código en Arduino para el módulo WIFI se encuentra disponible.

En el Arduino Nano se sigue el procedimiento presentado en la figura 12, donde se inicializa el bus CAN a una velocidad de 125kbps. Y se procede con la lectura de los datos enviados por la ESP01 del puerto serial, para identificar las tramas recibidas y el actuador al que corresponde. Con esta información se procede a escribir el mensaje respectivo en el bus CAN, el cual especifica el ID del actuador y los datos necesarios para activarlo.

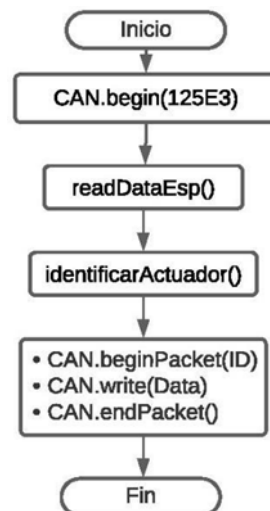


Figura 12. Diagrama de flujo para lectura e identificación de tramas para la activación de actuadores

El código respectivo para el Arduino Nano se encuentra disponible.

Configuración de la plataforma IoT Blynk

Para utilizar Blynk, es necesario realizar un registro y crear una cuenta en la plataforma. Posteriormente, se recibe un token de autenticación por correo electrónico, el cual se utiliza para acceder a datos de las variables o dispositivos configurados en la plataforma. En este caso, el dispositivo será el módulo ESP01 que actúa como módulo WIFI y que se conecta con el microcontrolador Arduino Nano.

En la programación del módulo WIFI, se deben agregar las librerías necesarias y la información requerida para establecer la comunicación con la plataforma Blynk. Como se observa en la figura 13, se requiere de la definición del ID del proyecto, el token y de las librerías para la conexión con datos de la plataforma:

```

Token de autenticación
#define BLYNK_TEMPLATE_ID "TMPL20TiFmwHU"
#define BLYNK_TEMPLATE_NAME "ProyectoAuto"
#define BLYNK_AUTH_TOKEN "STs8xHqiL5VtcsNic_6VKAq11zmJT_sw"

/* Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial

Librerías
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

Red de conexión WIFI
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Celular Kim";
char pass[] = "salj18431";
  
```

Figura 13. Configuración de librerías, token de autenticación y red de conexión WIFI

Una vez que el dispositivo se ha conectado a la plataforma, se procede a crear el correspondiente panel de control con cada una de las variables. En esta instancia, se han configurado seis variables. La primera variable es un temporizador que indica los segundos transcurridos desde la conexión del dispositivo a la plataforma IoT. La segunda variable refleja que se ha establecido la conexión del módulo WIFI con la plataforma. Las cuatro variables restantes representan los valores provenientes de los sensores del vehículo con sus rangos establecidos.

En la figura 14, se aprecia la interfaz con las variables seleccionada, donde se puede monitorear constantemente su estado.



Figura 14. Panel de visualización de las variables seleccionadas en la plataforma Blynk

Desarrollo de la Skill para Alexa

El diseño y programación de la skill se la realiza en la plataforma Voiceflow, donde se utiliza una programación por bloques, para desarrollar cada una de las funcionalidades que podrá tener la skill. Adicionalmente, esta plataforma permite transferir todas las configuraciones y datos necesarios que requiere la skill a la propia consola de desarrollo de Amazon, y así poder utilizarla en los dispositivos Alexa.

Lectura de variables de sensores

En la figura 15, se presenta el diagrama de flujo de la programación en Voiceflow, para la lectura de las variables de los sensores seleccionados. Para este caso, la frase de activación de la skill será “Alexa, abre mi Toyota”. Con lo cual se iniciará la conversación, donde el asistente le preguntará al usuario por la variable que desea conocer, y según sea su respuesta se presentan varios casos. Estos corresponden a cada una de las variables, ya que se compara la respuesta del usuario con las opciones disponibles, y con la que coincida se dará como respuesta. En caso de que el usuario no conozca el funcionamiento de la skill, se cuenta con dos alternativas, la primera se trata de la opción “Ayuda” en la que se dará a conocer el propósito de la skill y sus funcionalidades. La segunda alternativa es la de “Opciones”, que simplemente recordará al usuario que funcionalidades puede seleccionar. Una vez que Alexa haya dicho el valor de la variable, se realiza una verificación para saber si este valor se encuentra dentro de los rangos normales, caso contrario se da un mensaje de alertar al usuario. Si todo se encuentra en orden, se pregunta al usuario si desea alguna información adicional o no, si la respuesta es afirmativa el usuario podrá elegir otra variable, y si no desea más información se da fin a la conversación. Estatus del sistema de combustible: da como respuesta alguna de las alternativas presentadas en la tabla 5, que hace referencia a los diferentes estados del sistema de combustible.

- Temperatura del refrigerante: da a conocer al usuario la temperatura a la que se encuentra el líquido refrigerante.
- Tiempo de avance: da a conocer al usuario el tiempo de avance o avance de encendido, antes del PMS.
- Sensor de oxígeno: da a conocer al usuario la relación aire-combustible (AFR).
- Ayuda: da a conocer más información sobre la skill y sus funcionalidades.
- Opciones: da a conocer al usuario únicamente las funcionalidades que posee.

Activación de actuadores

Para el caso de los actuadores, el flujo del programa de la skill es similar que el de lectura de sensores, sin embargo, en lugar de que se envía a la plataforma de Blynk una solicitud de información, se realiza una actualización de datos sobre las variables respectivas. En la figura 16, se muestra el diagrama de flujo correspondiente para la activación de los seguros de puertas del vehículo.

En este caso se utiliza el vehículo Mazda CX3, ya que como se ha mencionado anteriormente este cuenta con dos buses CAN. Sin embargo, debido a la limitada información acerca de los mensajes que se deben escribir en el bus CAN para la activación de cualquiera sus actuadores. Se ha seleccionado la posibilidad de que el usuario pueda activar o desactivar el seguro de puertas del vehículo.

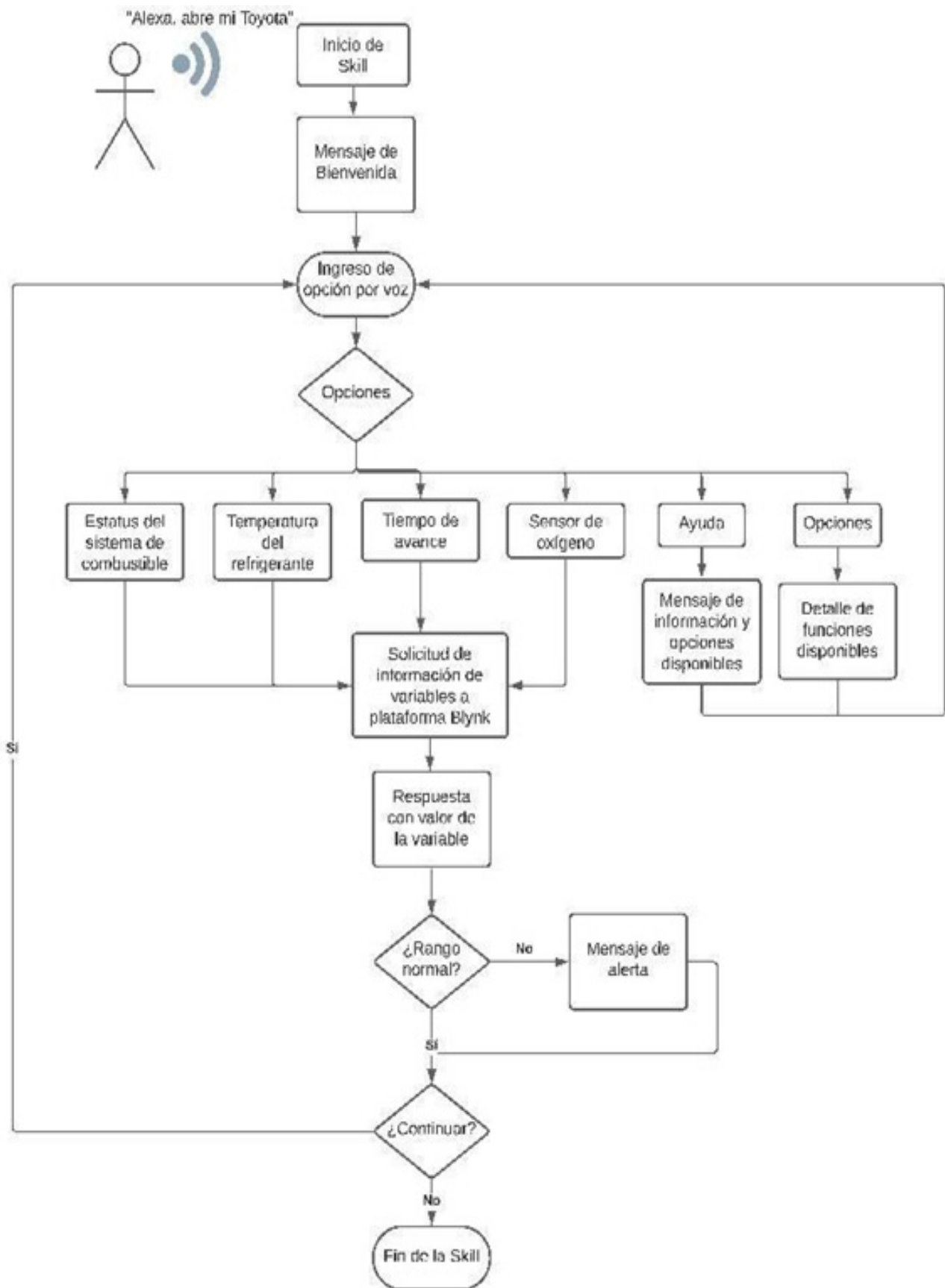


Figura 15. Diagrama de flujo de la skill diseñada para la lectura de variables de los sensores del vehículo Toyota Hilux CS 2020

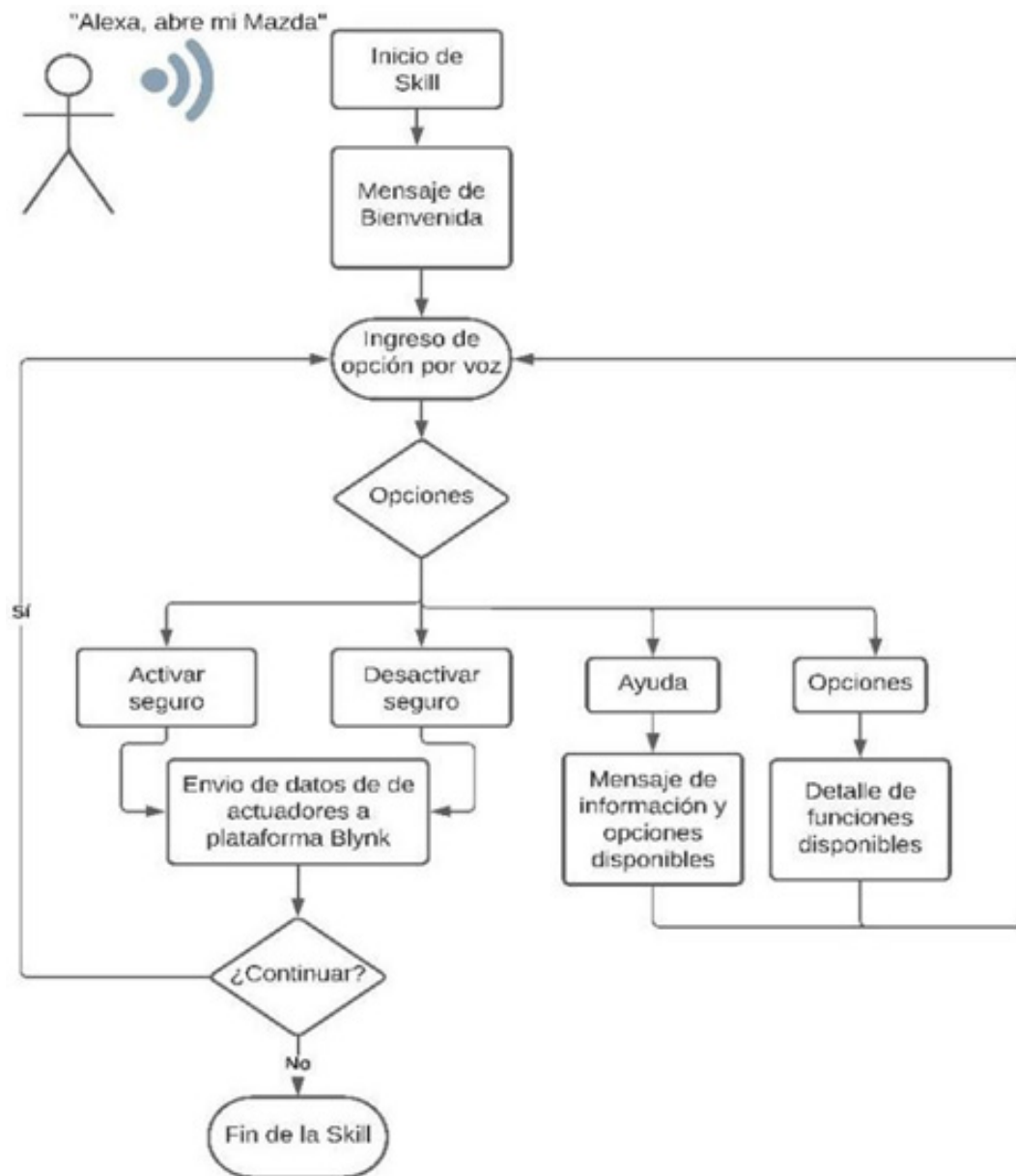


Figura 16. Diagrama de flujo de la skill diseñada para la activación de seguros del vehículo

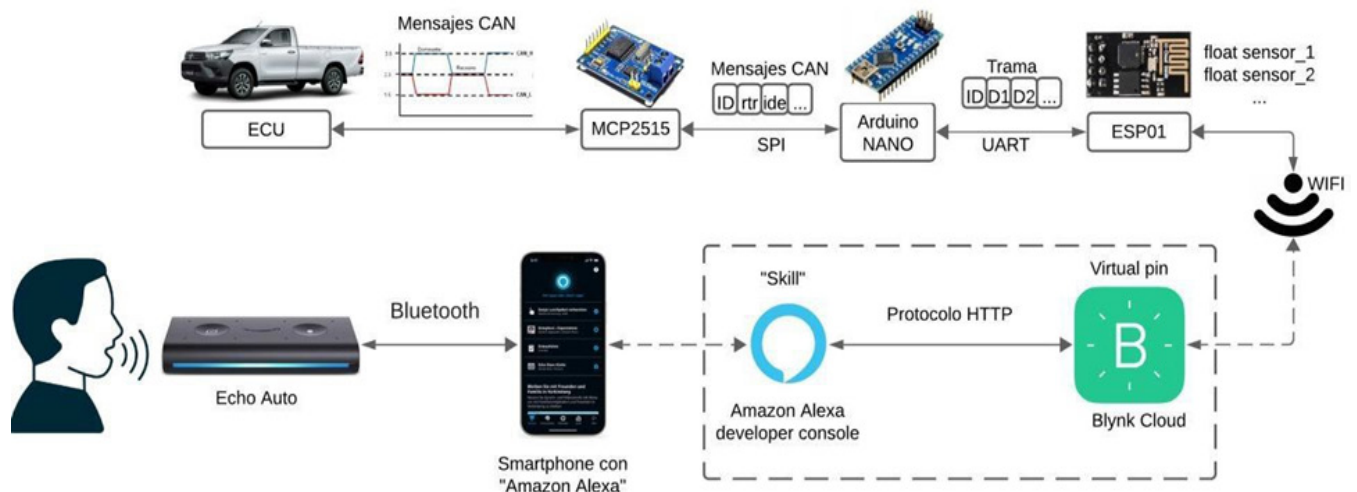


Figura 17. Esquema para interconexión de Alexa Auto y el computador del vehículo

El flujo que sigue la conversión entre Alexa y el usuario es muy similar tanto para la lectura de sensores como para la activación de actuadores. En este caso las funcionalidades que se han programado son:

- Activar seguro: se encarga de actualizar una variable en la plataforma Blynk relacionada con la activación del seguro de las puertas.
- Desactivar seguro: se encarga de actualizar una variable en la plataforma Blynk relacionada con la desactivación del seguro de las puertas.

Las alternativas de “Ayuda” y “Opciones” tienen la misma funcionalidad que en la lectura de sensores.

Esquema de conexión general del sistema

El objetivo principal del sistema es la conexión desde el vehículo hasta el asistente de voz Alexa Auto. Donde se da prioridad a la adquisición de variables en tiempo real y a su integración en un servidor en la nube para gestionar eficientemente la información. En la figura 17, se aprecia el esquema de conexión de los dispositivos y plataformas utilizadas.

Para el esquema presentado en la figura 17, se inicia con la conexión del módulo MCP2515 al computador del vehículo por los pines correspondientes al estándar (CAN_H pin 6 y CAN_L pin 14) del puerto OBDII, lo que posibilita la transmisión y recepción de paquetes de datos a través del bus CAN. Por una parte, esto permite realizar una solicitud de los datos de los sensores con los PID respectivos y también de recibir como respuesta los datos correspondientes. Por otra parte, permite el envío de mensajes en el bus CAN para la activación de actuadores específicos, siempre y cuando se conozca el ID del módulo y los bits que se deben modificar.

El controlador CAN emplea una interfaz de comunicación SPI para conectarse al microcontrolador Arduino NANO, el cual se encarga tanto de supervisar la información que será enviada al vehículo como de recibir los datos correspondientes de los sensores.

Luego, se envía o recibe por medio de comunicación serial los datos de cada uno de los sensores o actuadores al módulo WIFI ESP01. En este punto se establece la conexión con la plataforma Blynk, a la cual se transmiten constantemente los datos leídos de los sensores y en cambio esta envía información en caso de que el usuario dese activar un actuador.

En la plataforma Blynk, se crea un panel para la visualización de las variables recibidas, donde se podrá verificar constantemente los valores de los sensores y también si el módulo WIFI se encuentra conectado. Por otra parte, la plataforma Voiceflow, permite la creación de la skill, donde se implementa la lógica que guiará la interacción del usuario con el asistente de voz Alexa. Y a través de los bloques API, se recopilan los datos provenientes de Blynk. Para ello, las solicitudes que se realizan tienen la siguiente estructura:

- Para la lectura de datos de una variable: `https://{server_address}/external/api/getAll?token={token}`
- Para la actualización de valores en una variable: `https://{server_address}/external/api/update?token={token}&{pin}={value}`

Cabe señalar que cuando se realiza la lectura de una variable, es necesario almacenarla en una variable local de la skill para que esta pueda darla al usuario como respuesta audible. De este modo se puede obtener la información necesaria y responder a las preguntas del usuario.

La skill incluirá acciones que el asistente de voz ejecutará en relación con la información previamente recaudada de las variables del vehículo, es importante destacar que dentro de la lógica de programación de la skill se verifica que los datos obtenidos de los sensores se encuentren en rangos normales, lo que permite alertar al usuario sobre comportamientos anormales del vehículo.

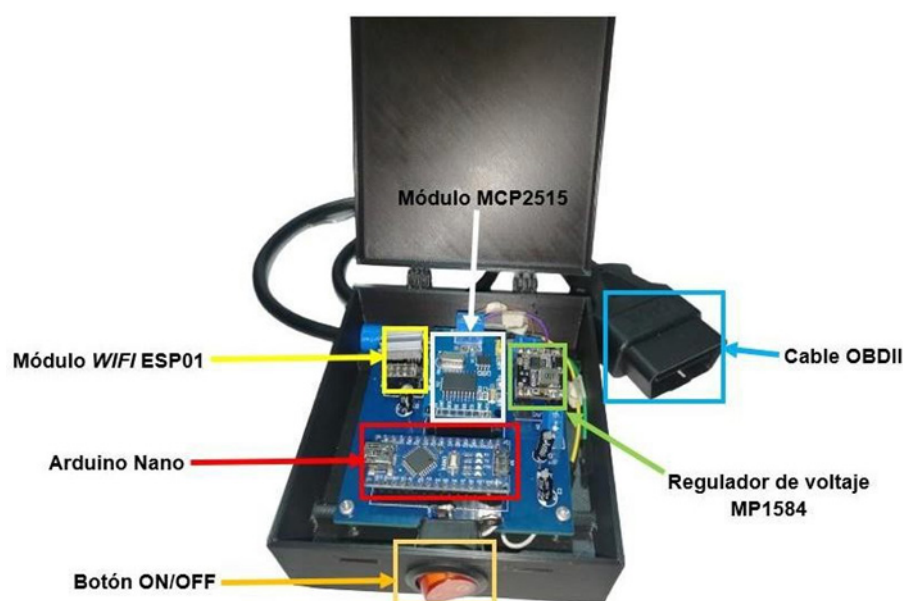


Figura 18. Implementación del dispositivo

Implementación del hardware y software del sistema de comunicación vehículo

La implementación del sistema comprende el uso de un cable OBDII conectado al puerto del vehículo, una caja fabricada mediante impresión 3D que alberga la placa PCB, y un botón para la alimentación del circuito. El dispositivo Echo Auto también forma del sistema ya que se encarga de recibir los comandos de voz del usuario y transmitirlos a las diferentes plataformas. La caja para albergar la placa PCB tiene un espesor de 3 [mm] y sus dimensiones son 10,3 [cm] de ancho, 9,3 [cm] de largo y 4 [cm] de alto, excluyendo la estructura de la tapa. Como se muestra en la figura 18, en la parte inferior izquierda de la caja, se presenta una abertura para la conexión del cable OBDII a la placa PCB, mientras que en la parte frontal se creó una hendidura para insertar un interruptor ON/OFF que posibilitará la activación del sistema.

Es importante destacar que estos dispositivos serán estratégicamente ubicados con el objetivo de no causar molestias al conductor

Pruebas y Resultados

Introducción

Las pruebas de funcionamiento de la skill desarrollada, se enfocan en la lectura de sensores y la activación del actuador en el vehículo respectivo, con el fin de evaluar que los valores que se dan al usuario son correctos y que se logra el control de los seguros de las puertas. Estas pruebas se van a llevar a cabo en diferentes escenarios o situaciones, entre las cuales se encuentran: el vehículo en reposo con el motor encendido, y el vehículo en movimiento.

Pruebas de funcionamiento de la skill para lectura de sensores

Estas pruebas se llevaron a cabo en base a las solicitudes programadas en la skill, que se refieren a cada una de las variables seleccionadas del vehículo Toyota Hilux CS 2020. En la tabla 6, se describen dichas variables y el comando de voz respectivo, mediante los cuales se realizarán las pruebas en varios entornos. Cabe recalcar que en la programación de la skill se consideraron diversas alternativas para cada comando, es decir el usuario puede usar palabras similares para seleccionar una de las alternativas.

Tabla 6. Variables y comandos de voz para las pruebas del funcionamiento de la skill	
Variable seleccionada	Comando de voz
Estado del sistema de combustible	Estatus del auto
Temperatura del anticongelante del motor	Temperatura del anticongelante
Tiempo de avance	Tiempo de avance del auto
Sensor de oxígeno	Sensor de oxígeno
Ayuda	Ayuda
Opciones	Opciones

Cada uno de los valores registrados fueron comprobados mediante el módulo ELM327, este módulo es un escáner automotriz de 16 pines que se conecta directamente a la interfaz OBDII, es utilizado para leer los códigos de errores y también permite visualizar todos los parámetros del vehículo, mediante una aplicación móvil o de escritorio a las que se conecta de forma inalámbrica por Bluetooth.⁽³⁰⁾ Este dispositivo (ELM327) se conectará al vehículo al mismo tiempo con el dispositivo desarrollado en este proyecto para recolectar la información en tiempo real, por medio de un cable OBDII dúplex. En cuanto a la activación del seguro de las puertas, las pruebas realizadas son similares ya que se verifica que se reconozcan correctamente los comandos de voz y que se ejecute la acción solicitada.^(31,32)

Pruebas de lectura de sensores con el vehículo en reposo

Para realizar pruebas con el auto en reposo se ha considerado la situación en la que el motor está encendido, pero no en marcha. Es decir, el auto no se encuentra en movimiento.^(33,34) Con lo cual se busca determinar la capacidad que tiene el Echo Auto para responder en situaciones con poco ruido en el entorno. Para cada caso se han considerado un total de 5 pruebas por cada una de las interacciones disponibles.^(35,36)

Para realizar esta prueba se tienen las siguientes consideraciones durante la prueba realizada:

- El motor está encendido, pero el vehículo no se encuentra en marcha.
- El ruido ambiente aproximado es de 61 dB.
- Revoluciones por minuto: 656 rpm.
- El auto está en neutro.
- No se está presionando el acelerador.

Siguiendo estas consideraciones, se ha realizado el registro de los valores de cada uno de los sensores seleccionados.

Estado del sistema de combustible

La tabla 7, presenta los resultados obtenidos de la lectura del estatus de combustible, con el motor encendido:

En este caso el valor obtenido de 2, según la tabla 8 indica que se ha cerrado el circuito del sistema de combustible, utilizando retroalimentación del sensor de oxígeno para determinar la mezcla de combustible. Tanto el valor obtenido de la skill como del ELM327 coinciden en cada uno de los intentos realizados.^(37,38,39)

Tabla 7. Resultado de pruebas de lectura de la variable del estado del sistema de combustible

Prueba	1	2	3	4	5
Estatus del auto	2	2	2	2	2
Valor ELM327	2	2	2	2	2

Temperatura del anticongelante

Para el registro de los valores de temperatura en la tabla 8, se ha mantenido el motor encendido por aproximadamente 10 min. Por lo cual la temperatura se ha elevado hasta aproximadamente 45°C.^(40,41,42)

Tabla 8. Resultados de las pruebas de lectura de la variable de la temperatura del anticongelante del vehículo

Prueba	1	2	3	4	5
Temperatura del anticongelante	45°C	45°C	45°C	46°C	46°C
Valor ELM327	45°C	45°C	45°C	46°C	46°C
Error	0 %	0 %	0 %	0 %	0 %

En esta prueba tanto el valor proporcionado por la skill como por el escáner ELM327, coinciden al 100 %. En este caso cabe destacar que tras un periodo de tiempo con el motor encendido la temperatura se ha regulado en aproximadamente 45°C.^(43,44,45)

Tiempo de avance

Los resultados de la lectura del tiempo de avance se presentan en la tabla 9, en este caso específico al estar el motor encendido, pero no en marcha, no existe mayores variaciones en el tiempo de avance registrado.^(46,47,48)

Tabla 9. Resultados de las pruebas de lectura de la variable de tiempo de avance del vehículo

Prueba	1	2	3	4	5
Tiempo de avance	10°	10°	10°	11°	10°
Valor ELM327	9°	10°	10°	10°	10°
Error	11,11 %	0 %	0 %	10 %	0 %

En este caso el promedio de los errores obtenidos es de 4,22 %. Dado que el auto no está en marcha el valor del tiempo de avance sea establecido en aproximadamente 10° antes del PMS.^(49,50,51)

Sensor de oxígeno

En la tabla 10, se registran los valores de la relación aire-combustible (AFR) obtenidos del sensor de oxígeno:

Tabla 10. Resultados de las pruebas de lectura de la variable del sensor de oxígeno del vehículo

Prueba	1	2	3	4	5
Sensor de oxígeno	14,72	14,75	14,73	14,71	14,75
Valor ELM327	14,72	14,76	14,73	14,72	14,76
Error	0 %	0,06 %	0 %	0,06 %	0,06 %

El error promedio encontrado es de 0,036 %, lo cual no indica una variación significativa con respecto a los valores referenciales del ELM327.^(52,53)

En esta prueba al tener una fuente adicional de ruido, se han identificado dos puntos principales que generan variaciones en el registro de los datos de los sensores.^(54,55,56) Por una parte, la pronunciación de comandos extensos puede ocasionar que el Echo Auto no registre correctamente el audio. Por otra parte, en variables como el tiempo de avance o la relación aire-combustible (AFR), los valores proporcionados por la skill con respecto a los del ELM327 presentan mayor error.^(57,58) Esto se debe principalmente a que son variables que cambian rápidamente en el tiempo, y generan pequeñas variaciones en los datos ya que ambos sistemas no realizan la lectura de las variables exactamente al mismo tiempo.^(59,60)

Pruebas de lectura de sensores con el vehículo en movimiento

Para realizar estas pruebas se han considerado una situación en la que el vehículo se encuentra en marcha. Ya que en este caso se tiene una diferencia considerable del ruido en el interior del vehículo. De igual manera para el registro de los datos de los sensores

se han considerado un total de 5 pruebas por cada una de las interacciones disponibles.^(61,62,63,64)

Para llevar a cabo esta prueba se tienen las siguientes consideraciones:

- El motor está encendido y el vehículo en marcha.
- Ventanas cerradas.
- El ruido ambiente aproximado es de 63 dB.
- El auto está en una velocidad promedio de 30-40 km/h.

En estas condiciones se procede a realizar el registro de las variables de los sensores y compararlos con los que proporciona el escáner ELM327.⁽⁶⁵⁾

Estado del sistema de combustible

La tabla 11, presenta los resultados obtenidos de la lectura del estatus de combustible, con el vehículo en movimiento:

Tabla 11. Resultados de las pruebas de lectura de la variable del estado del sistema de combustible					
Prueba	1	2	3	4	5
Estatus del auto	2	2	2	2	2
Valor ELM327	2	2	2	2	2

En este caso el valor de 2 se mantiene al igual que en las pruebas anteriores ya que el motor está encendido y el sistema de combustible no presenta ninguna falla.

Temperatura del anticongelante

Los valores de temperatura del anticongelante se presentan en la tabla 12. Al estar en marcha los valores de temperatura se han elevado considerablemente, pero se mantienen dentro de los rangos normales.^(66,67,68)

Tabla 12. Resultados de las pruebas de lectura de la variable de la temperatura del anticongelante del vehículo					
Prueba	1	2	3	4	5
Temperatura del anticongelante	81°C	81°C	80°C	81°C	81°C
Valor ELM327	81°C	80°C	80°C	80°C	81°C
Error	0 %	1,25 %	0 %	1,25 %	0 %

En esta prueba el promedio de los errores obtenidos es de 0,5 %. Dado que el sistema en general mantiene una conexión de internet intermitente, por ello se presentan errores en la recepción de datos del vehículo en movimiento.^(69,70,71)

Tiempo de avance

En la tabla 13 se presentan los resultados de la lectura de los datos del tiempo de avance.^(72,73)

Tabla 13. Resultados de las pruebas de lectura de la variable de tiempo de avance del vehículo					
Prueba	1	2	3	4	5
Tiempo de avance	29°	41°	37°	42°	42°
Valor ELM327	30°	40°	37°	42°	41°
Error	3,33 %	2,5 %	0 %	0 %	2,44 %

En este caso se presenta un error promedio de 1,65 %. Dado que el vehículo está en marcha el valor del tiempo de avance es muy variante ya que depende de factores como la carga, velocidad y aceleración a lo largo del recorrido. Sin embargo, los valores leídos no presentan mucha variación con respecto a los valores leídos por el escáner.^(74,75,76)

Sensor de oxígeno

En la tabla 14, se registran los valores de la relación aire-combustible (AFR) obtenidos del sensor de oxígeno, con el vehículo en movimiento. En este caso específico al estar el vehículo en movimiento, estos valores varían según las condiciones en las que se está conduciendo, como la velocidad o la carga del vehículo:

Tabla 14. Resultados de las pruebas de lectura de la variable del sensor de oxígeno del vehículo					
Prueba	1	2	3	4	5
Sensor de oxígeno	14,82	14,60	14,77	14,77	14,56
Valor ELM327	14,81	14,62	14,71	14,76	14,54
Error	0,06 %	0,14 %	0,41 %	0,06 %	0,14 %

En esta situación se muestra un error promedio de 0,16 %. Ya que los valores son menores al 1 %, se considera un error aceptable entre los datos proporcionados por el circuito propuesto y el valor obtenido con el dispositivo ELM327.^(77,78,79)

Los errores presentados en las pruebas de lectura de sensores con el vehículo en movimiento son menores al 2 %, lo que indica que la diferencia existente se puede considerar aceptable en esta aplicación, es importante considerar que se presentan múltiples factores externos al circuito, los cuales pueden afectar la respuesta dada.^(80,81,82,83,84) Estos factores pueden provenir del vehículo como la velocidad, carga, aceleración y entre otras características físicas durante la conducción, así como también se pueden encontrar factores externos al vehículo como lo es la conexión a internet del módulo WIFI, el enlace bluetooth entre el dispositivo móvil y el Echo Auto, lo cuales pueden afectar el funcionamiento del dispositivo. Sin embargo, los resultados registrados de cada variable no presentan errores significativos.^(85,86)

Pruebas de la skill para activación de actuadores

Para realizar las pruebas de funcionamiento de los actuadores se han considerados las siguientes condiciones respecto al estado del auto (Mazda CX3) y el entorno:

- El motor está encendido, pero el vehículo no se encuentra en marcha.
- El ruido ambiente aproximado es de 50 dB.
- Revoluciones por minuto: 750 rpms.
- El auto está en modo parking.
- No se está presionando el acelerador.

En estas condiciones se presenta en la tabla 15, los resultados respecto al cumplimiento de la orden solicitada, ya sea para activar o desactivar el seguro en las puertas del vehículo.

Tabla 15. Resultados de las pruebas de la skill para activación de actuadores del vehículo											
Prueba	1	2	3	4	5	6	7	8	9	10	Total [%]
Activar seguro	C	C	C	C	C	X	C	C	C	C	90
Desactivar seguro	C	C	X	C	C	C	C	C	C	X	80

En esta prueba se ha logrado obtener un 85 % de efectividad, ya que en las pruebas que presentan error, el sistema no reconoció correctamente el comando de voz respectivo a la activación o desactivación del seguro de la puerta del vehículo.^(87,88) Al ser los comandos muy similares, el usuario puede recurrir en errores de pronunciación y con el ruido de fondo adicional puede ocasionar esta clase de errores al registrar el comando con el asistente de voz.^(89,90) Cabe recalcar que esta activación se la realizó en modo parking, ya que en este modo el actuador se encuentra en funcionamiento automático y se puede interactuar con el actuador mediante la programación descrita anteriormente.^(91,92) Es importante mencionar que este modo de interacción con el vehículo, se lo implementó debido a las medidas de seguridad propias del vehículo para evitar que las puertas puedan abrirse accidentalmente mientras el vehículo esté en movimiento.⁽⁹³⁾

CONCLUSIONES

Se ha implementado un sistema capaz de interconectar el computador de un vehículo al asistente de voz Alexa por medio del dispositivo Echo Auto. Este sistema permite el acceso a información de sensores y la activación de actuadores específicos, para lo cual se ha seleccionado el hardware capaz de conectarse al puerto OBDII del vehículo y a su vez a la red de comunicación CAN, con la finalidad de poder enviar mensajes al bus para la lectura o escritura de alguna variable.

El hardware diseñado incluye un microcontrolador Arduino NANO, que se encarga de especificar que mensajes se deben enviar y cuál es la información que deben contener los mensajes CAN, de este modo se definen los datos necesarios para la obtención de datos de variables específicas. Además, envía estos datos al módulo WIFI que se encarga de procesarlos y enviarlos a la plataforma Blynk, donde estará toda esta información disponible para que el usuario pueda monitorearla.

Las plataformas IoT ofrecen una gran ventaja, ya sea para la administración de dispositivos IoT o para el acceso a información de variables de estos dispositivos. Lo cual ha permitido el óptimo desarrollo de una skill para Alexa, puesto que por medio de solicitudes HTTP es posible acceder a la información de la plataforma Blynk y dar al usuario una respuesta audible sobre datos de sensores disponibles.

Este sistema presenta una oportunidad importante para el desarrollo de aplicaciones que permitan incrementar el confort de las personas dentro de un vehículo y también de anticipar posibles averías en el funcionamiento del motor. Ya que por medio de

comandos de voz se puede acceder a información de diagnóstico del auto o incluso controlar ciertos actuadores, todo esto sin la necesidad que el conductor quite las manos del volante o desvíe su atención del camino.

La lectura de variables o de parámetros del vehículo en la camioneta Toyota Hilux CS 2020 ha conllevado un estudio exhaustivo sobre el protocolo OBDII, ya que para este fin es necesario conocer la estructura de los mensajes utilizados y la normativa que rige el sistema de comunicación. Esto ha permitido identificar qué parámetros estándar son o no compatibles con el vehículo y de este modo seleccionar las más adecuadas. Con esta información se ha llevado a cabo el correcto desarrollo del software que escribirá o leerá los mensajes del bus de comunicación CAN.

La escasa información oficial acerca de los parámetros o mensajes que se manejan dentro del bus de comunicación CAN de los vehículos, limitan en gran medida las posibilidades sobre el control de actuadores. A lo cual se suma, las características de la arquitectura de la red de comunicación ya que no todos los módulos de control del vehículo están conectados a la red CAN, por tanto, su acceso está restringido. A pesar de esto se ha extraído de fuentes no oficiales información de varios paquetes de mensajes decodificados, relacionados con las funciones de control de ciertos actuadores para el Mazda CX-3, de este modo se ha logrado la activación y desactivación del seguro de sus puertas utilizando comandos de voz a través del sistema desarrollado.

Las pruebas realizadas indican un gran nivel de efectividad en el reconocimiento de los comandos de voz tanto cuando el vehículo está en reposo como en movimiento. Los valores de las variables leídas no presentan más de un 4 % de error con respecto a los datos referenciales proporcionados por el escáner ELM327, cuando el auto está en reposo. Mientras que cuando está en movimiento el error es menor al 2 %. Estos errores se pueden explicar por el hecho de que el tiempo de avance y la AFR son variables que cambiar rápidamente, y el sistema diseñado no realiza la lectura de las variables al mismo tiempo que el ELM327, generando así pequeñas variaciones.

Con respecto a las pruebas realizadas para la activación del actuador en el vehículo Mazda CX-3, se obtuvo un 85 % de efectividad, lo que señala un alto nivel de reconocimiento con respecto a los comandos de voz. Sin embargo, la implementación de esta función solo se logró en condiciones específicas, debido a las características propias de la configuración del vehículo que limitan la modificación de valores de ciertos parámetros.

RECOMENDACIONES

Buscar información oficial de la red de comunicación del vehículo disponible, antes de intentar modificar cualquier parámetro del vehículo.

No exceder una longitud de 50 cm desde el puerto OBDII hasta el sistema físico desarrollado para evitar interferencias o caídas de tensión importantes.

Verificar el protocolo con el que cuenta el vehículo previo a conectar el sistema puesto que el sistema descrito es compatible únicamente para el protocolo CAN.

Previo a la lectura de PID realizar una verificación de los PID que están disponibles en el vehículo utilizado, para evitar que el sistema envíe mensajes al bus de datos solicitudes de variables no disponibles y que pueden ocasionar errores en el programa.

Proporcionar al sistema una red WIFI estable ya que los datos se envían constantemente a la nube y ante intermitencias de conexión las variables no se actualizan y el usuario puede obtener resultados que no correspondan al momento en que se solicitaron.

FINANCIACIÓN

Ninguna.

CONFLICTO DE INTERESES

Los autores declaran que no existe conflicto de intereses.

CONTRIBUCIÓN DE AUTORÍA

Conceptualización: Alexis Javier Villacis Ninasunta, Mariana Pinargote Basurto.

Curación de datos: Alexis Javier Villacis Ninasunta, Mariana Pinargote Basurto.

Análisis formal: Alexis Javier Villacis Ninasunta, Mariana Pinargote Basurto.

Redacción – borrador original: Alexis Javier Villacis Ninasunta, Mariana Pinargote Basurto.

Redacción – revisión y edición: Alexis Javier Villacis Ninasunta, Mariana Pinargote Basurto.

REFERENCIAS BIBLIOGRÁFICAS

1. Ortiz A, Dávila R. Implementación de un asistente virtual para los estudiantes de pregrado de una universidad peruana. Rev Conrado. 2023;19(92):121-8.

2. Gonzales P. Los coches que mejor te hablan. 2022. <https://hackercar.com/los-coches-que-mejor-te-hablan/>

3. Fernández R. Amazon Alexa Auto Tech: The ultimate guide to Alexa in your car. MUO. 2022. <https://www.makeuseof.com/alexa-auto-tech-ultimate-guide/>

4. Mihale-Wilson A, Zibuschka J, Hinz O. User preferences and willingness to pay for in-vehicle assistance. Electron Markets. 2019;29:37-53.

5. Amazon. ¿Qué es una Skill de Alexa? 2024. <https://developer.amazon.com/es-ES/alexa/alexa-skills-kit>
6. Amazon. Design Your Skill. 2024. <https://developer.amazon.com/en-US/docs/alexa/design/design-your-skill.html>
7. Villén J. Simulador de la ECU de un vehículo con protocolo ISO 9141-2. Sevilla: Universidad de Sevilla; 2016.
8. ELM Electronics. OBD Interpreter: Protocols Support - ELM327 (OBD Interpreter ICs). 2023. <https://www.elmelectronics.com/products/ics/obd/>
9. Weis O. Conector OBD2 explicado. Flexihub; 2023. <https://www.flexihub.com/es/oobd2-pinout>
10. Toyota Venza. Toyota Venza: system diagram. 2016. https://www.tovenza.com/system_diagram-1256.html
11. Opinautos. ¿Qué es el sensor de oxígeno (o sonda lambda) en un Hilux? 2024. <https://www.opinautos.com/ec/toyota/hilux/guias/sensor-de-oxigeno>
12. HELLA. Sensor de temperatura del refrigerante. 2024. <https://www.hella.com/techworld/mx/Informacion-Tecnica/Sensores-y-actuadores/Revision-del-sensor-de-temperatura-del-refrigerante-4277>
13. AUTODOC CLUB. Sensor del cigüeñal: fallas, función, síntomas. 2021. <https://club.autodoc.es/magazin/sensor-del-ciguenal-fallas-funcion-sintomas>
14. Mazda. Mazda CX-3. 2024. <https://www.mazda.com.ec/mazda-cx-3-360>
15. Transductor.net. Apertura eléctrica de puertas actuadores. 2023. <https://transductor.net/apertura-electrica-de-puertas-actuadores/>
16. MACTRONICA. Convertidor DC-DC 12V a 5V 1A reductor. 2024. <https://www.mactronica.com.co/convertidor-dc-dc-12v-a-5v-1a-reductor>
17. AV Electronics. Módulo WiFi ESP8266EX. 2024. <https://avelectronics.cc/producto/modulo-wifi-esp8266ex/>
18. Arduino. Arduino Nano. 2024. <https://arduino.cl/arduino-nano/>
19. AV Electronics. Arduino Nano. 2024. <https://avelectronics.cc/producto/nano-v3/>
20. Arduino Spain. Conectar módulo WiFi ESP8266 ESP-01 Arduino. 2023. <https://arduino-spain.site/modulo-esp8266/>
21. SAE. SAE J1979: Surface Vehicle Standard. 2017. https://img.antpedia.com/standard/files/pdfs_ora/20200926/SAE%20J1979-2017.pdf
22. Softing. SAE J1979-2 OBDOnUDS – diagnostic standard. Softing; 2023. <https://automotive.softing.com/standards/protocols/obdonuds-sae-j1979-2.html>
23. Varga A. CANDRIVE. GitHub; 2023. <https://github.com/adamtheone/canDrive>
24. FORScan forum. Configuration and programming. 2021. <https://forscan.org/forum/viewforum.php?f=16>
25. E-AUTO. Avance de encendido. 2012. <https://www.e-auto.com.mx/engew/index.php?view=article&id=3429>
26. CERVANTES GAS. Consecuencias de un auto con avance excesivo: ¿qué debes saber? 2022. https://cervantesgas.com.ar/que-pasa-si-el-auto-esta-muy-avanzado/?expand_article=1
27. Freudenburger B. Sensores de oxígeno 101. 2020. <https://www.walkerproducts.com/wp-content/uploads/2020/05/OXYGEN-SENSORS-101-INFORMATION-BOOKLET-Spanish.pdf>
28. Fernández F. La sonda lambda o sensor de oxígeno. 2012. https://www.ealuro.com/eb3emd/Sonda_Lambda/Sonda_lambda.htm
29. Litnevskyi S. Manual de usuario de Excel Mazda SkyActiv OBD-II calc (FORScan). 2021. <https://www.youtube.com/watch?v=N9si40XZbcg>

30. AutomotrizEscaner. Escáner Automotriz ELM327. 2024. <https://www.automotrizescaner.com/escaner-automotriz-elm327-usb>
31. Amazon. Conector macho de 16 pines J1962 OBD2 OBD-II para abrir el cable, cable de extensión de diagnóstico OBD para bricolaje. 2020. https://www.amazon.com/dp/B07F16GPMB?starsLeft=1&ref_=cm_sw_r_cso_wa_apan_dp_KM170K9HRAWAJMJKM2WX&th=1
32. Amazon. For the first time, Amazon enables companies to access Alexa’s advanced AI to build their own intelligent assistants with Alexa Custom Assistant. Fiat Chrysler Automobiles is the first Automotive OEM to implement in vehicles. 2024. <https://developer.amazon.com/en-US/blogs/alexa/alexa-auto/2021/01/Amazon-Announces-Alexa-Custom-Assistant>
33. Amazon. ¿Qué es el Alexa Skills Kit? 2024. <https://developer.amazon.com/es-ES/alexa/alexa-skills-kit>
34. Amazon. Build Your Skill. 2024. <https://developer.amazon.com/en-US/docs/alexa/build/build-your-skill-overview.html>
35. Amazon. Certify and Publish Your Skill. 2024. <https://developer.amazon.com/en-US/docs/alexa/certify/certify-your-skill.html>
36. Amazon. Develop Your First Alexa Skill With the ASK SDK for Node.js. 2024. <https://developer.amazon.com/en-US/docs/alexa/alexa-skills-kit-sdk-for-nodejs/develop-your-first-skill.html>
37. Amazon. Echo Auto - Pon Alexa en tu coche. 2024. https://www.amazon.es/dp/B078YP59TT?tag=xtk-pivot-21&asc_refurl=&asc_source=xataka&asc_campaign=everlasting
38. Amazon. Lleva a Alexa en tu vehículo: Todo lo demás. 2024. <https://www.amazon.com/-/es/generaci%C3%B3n-modelo-Lleva-Alexa-veh%C3%ADculo/dp/B09X27YPS1?th=1>
39. Amazon. Monitor Your Skill Metrics and Earnings. 2024. <https://developer.amazon.com/en-US/docs/alexa/monitor/monitor-your-skill.html>
40. Amazon. Skill development workflow. 2024. <https://developer.amazon.com/en-US/docs/alexa/ask-overviews/what-is-the-alexa-skills-kit.html>
41. Amazon. Test and Debug Your Skill. 2024. <https://developer.amazon.com/en-US/docs/alexa/test/test-your-skill-overview.html>
42. Amazon. Vehicles with Alexa. 2024. https://www.amazon.com/b?node=17744356011&ref_=ALEXA_AUTO_HP_VEHICLES%C2%A0&tag=muo-v2-519n9pv-20&ascsubtag=UUmuoUeUpU2013241&asc_refurl=https%3A%2F%2Fwww.makeuseof.com%2Falexa-auto-tech-ultimate-guide%2F&asc_campaign=Evergreen
43. Android Auto. Compatibilidad con Android Auto. 2017. https://www.android.com/intl/es_es/auto/compatibility/
44. Arévalo F, Ortega A. Desarrollo de una interfaz para la visualización y adquisición de datos provenientes de la ECU a través de OBD-II mediante un dispositivo de comunicación serial y del analizador de gases Qrotech 6000. Universidad Politécnica Salesiana; 2016. <https://dspace.ups.edu.ec/bitstream/123456789/12029/1/UPS-CT005836.pdf>
45. Blynk.io. Introduction - Blynk Documentation. 2024. <https://docs.blynk.io/en/>
46. Braun M, Mainz A, Chadowitz R, Pfleging B, Alt F. At Your Service: Designing Voice Assistant Personalities to Improve Automotive User Interfaces. Conf Hum Factors Comput Syst (CHI ‘19). 2019;1-11.
47. Bret K. Consumers will use voice assistants more while driving in 2019 and they are already influencing new car buyer decisions. Voicebot. 2019. <https://voicebot.ai/2019/01/22/consumers-will-use-voice-assistants-more-while-driving-in-2019-and-they-are-already-influencing-new-car-buyer-decisions/>
48. Caicedo K, Boderio L. Diseño e implementación de un sistema de conectividad para el monitoreo en tiempo real de los sistemas funcionales de un automóvil. Guayaquil: Universidad Politécnica Salesiana; 2021. <https://dspace.ups.edu.ec/bitstream/123456789/21754/1/UPS-GT003585.pdf>

49. Calleja J. On the use of virtual assistant for smart cities parking service deployment. Santander: Universidad de Cantabria; 2021. <https://repositorio.unican.es/xmlui/bitstream/handle/10902/22078/435274.pdf>
50. Contreras J. Usos del puerto OBD2 para diagnóstico del motor de un vehículo desde un dispositivo móvil. Huejutla: Instituto Tecnológico de Huejutla; 2020. <https://rinacional.tecnm.mx/bitstream/TecNM/1127/1/JOSE%20ANTONIO%20CONTRERAS%20RAMIREZ.pdf>
51. Domínguez J. Desarrollo de una skill educativa sobre el asistente virtual Alexa: estudia conmigo. Elche: Universidad Miguel Hernández; 2022. <http://hdl.handle.net/11000/26573>
52. Edu J, Ferrer X, Such J, Suarez G. SkillVet: Automated Traceability Analysis of Amazon Alexa Skills. IEEE Trans Depend Secure Comput. 2021;20:161-75.
53. Electronilab. Módulo CAN bus MCP2515. 2024. <https://electronilab.co/tienda/modulo-can-bus-mcp2515/>
54. ElectrOnline. MINI ELM327 V2.1 Bluetooth HH OBD OBDII avanzado calidad. 2024. <https://www.electronline.cl/mini-elm327-v21-bluetooth-hh-obd-obdii-avanzado>
55. ENDADO. Sensor temperatura del refrigerante HELLA 89422-35010. 2024. <https://www.endado.com/p/sensor-temperatura-del-refrigerante-hella-6pt009107481>
56. Falch M. OBD2 explained - a simple intro. 2023. <https://www.csselectronics.com/pages/obd2-explained-simple-intro>
57. Google Cloud. Cómo implementar un chatbot de IA con Dialogflow. 2020. <https://www.cloudskillsboost.google/focuses/634?locale=es&parent=catalog>
58. HELLA. Refrigeración del vehículo: conocimientos básicos para el taller. 2013. https://gruasytransportes.files.wordpress.com/2014/09/bhs_2013_refrigeracion_del_vehiculo.pdf
59. Huang J. La norma SAE J1962 OBD II, el conjunto de cable de diagnóstico automotriz. 2024. https://es.made-in-china.com/co_edgarwireharness/product_SAE-J1962-OBD-II-Automotive-Diagnostic-Cable-Assembly_esiineheu.html
60. IONOS. Programación visual: la entrada más sencilla al mundo digital. 2020. <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/programacion-visual/>
61. Apple. iOS - CarPlay. España: Apple; 2021. <https://www.apple.com/es/ios/carplay/>
62. International Organization for Standardization. ISO 15031-3: Road vehicles: Communication between vehicle and external equipment for emissions-related diagnostics. 2016. <https://cdn.standards.iteh.ai/samples/64636/635b127e64d54bfd90d498c272008392/ISO-15031-3-2016.pdf>
63. JBL. JBL Link Drive Optimized for the Google Assistant. 2024. <https://ca.jbl.com/LINK+DRIVE.html>
64. Katreddi S, Kasani S, Thiruvengadam A. A review of applications of artificial intelligence in heavy duty trucks. Energies. 2022;15:7457.
65. Mazda CX-5 Service & Repair Manual: Controller Area Network (CAN) System. 2016. https://www.mcx5.org/controller_area_network_can_system-1163.html
66. Motores Auto. Sistema de inyección: partes, funcionamiento, tipos. 2023. <https://www.motoresauto.com/sistema-de-inyeccion/>
67. Muñoz C. Cómo crear tus skills de Alexa, aunque no sepas de programación. Tuexperto.com; 2022. <https://www.tuexperto.com/2022/03/24/como-crear-tus-skills-de-alexa-aunque-no-sepas-de-programacion/>
68. National Instruments. ¿Cuál es la longitud máxima del cable para un bus CAN? 2023. <https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z000000P7ikSAC&l=es-EC>
69. Navarro A. Creación e implementación de una skill de Alexa para laboratorios farmacéuticos. Barcelona: Escola Tècnica Superior d'Enginyeria Industrial de Barcelona; 2020.

70. Naylamp Mechatronics. Módulo CAN MCP2515. 2023. <https://naylampmechatronics.com/alambrico/253-modulo-can-mcp2515.html>
71. PiEmbSysTech. CAN protocol. 2021. <https://piembsystech.com/can-protocol/>
72. Portilla X. Mi primera Alexa Skill sin escribir código. 2020. <https://planetachatbot.com/alexa-skill-sin-codigo/>
73. Ramírez M. Asistentes de voz se perfilan como aliados de la revolución automotriz. Forbes México. 2021. <https://www.forbes.com.mx/forbes-life/auto-asistentes-de-voz-aliados-revolucion-automotriz/>
74. Sánchez L, Molano M, Fabela M, Martínez M, Hernández J, Vázquez D, et al. Revisión documental del protocolo CAN como herramienta de comunicación y aplicación en vehículos. 2016. <https://imt.mx/archivos/Publicaciones/PublicacionTecnica/pt474.pdf>
75. SEPLN-PD. InLIFE: Tecnologías del Lenguaje aplicadas al envejecimiento activo. In: Annual Conference of the Spanish Association for Natural Language Processing 2022: Projects and Demonstrations; 2022. Coruña. p. 21-3.
76. Smith G. ¿Qué es el bus CAN (red de área del controlador) y cómo se compara con otras redes de bus de vehículos? 2021. <https://dewesoft.com/es/blog/que-es-el-bus-can>
77. Smith S. Digital voice assistants in use to triple to 8 billion by 2023. Juniper Research; 2019. <https://www.juniperresearch.com/press/press-releases/digital-voice-assistants-in-use-to-8-million-2023>
78. STP. Sistema de combustible y aditivos de combustible. STP; 2023. <https://www.stp.com/es/sistema-de-combustible-y-aditivos-de-combustible>
79. Tahsim A. Una descripción general de los conceptos básicos del diseño en Voiceflow. Voiceflow; 2023. <https://learn.voiceflow.com/hc/en-us/articles/9174803155341-What-are-Steps-Blocks>
80. Tahsim A. Paso API: ¿Cómo hago llamadas API? ¿Cómo conecto mi Voiceflow a las API? Voiceflow; 2023. <https://learn.voiceflow.com/hc/en-us/articles/9262787577613>
81. Cardozo Rueda KS. Aplicación de redes neuronales artificiales para el pronóstico de precios de café. Revista Colombiana de Tecnología De Avanzada (RCTA). 2022; 39:113–117. <https://doi.org/10.24054/rcta.v1i39.1403>
82. Tobisch V, Funk M, Emfield A. Dealing with input uncertainty in automotive voice assistants. In: Automotive user interfaces and interactive vehicular applications. 2020. p.161-8.
83. Toyota Guatemala. ¡Toyota Hilux 2020 versión full equipo! Toyota Guatemala; 2021. <https://www.youtube.com/watch?v=GPbdY-u3HYw>
84. UBUY. Dorman 937-700 motor de actuador de cerradura de puerta lateral del conductor delantero compatible con modelos seleccionados de Mazda. UBUY; 2023. <https://www.ubuy.gt/es/product/2CRNZRDI-dorman-937-700-front-driver-side-door-lock-actuator-motor-for-select-mazda-models>
85. Brunengo L, Barrón P. Digital transformation in SMEs: challenges and potential in the data age. *EthAIca*. 2023; 2:56.
86. Villamar I. Estudio y análisis de los sistemas de diagnóstico en los automóviles modernos sistemas OBD. Cuenca: Universidad de Azuay; 2018.
87. Voicebot. Voice assistant influence on car purchase decision. Voicebot; 2019. <https://voicebot.ai/google-home-google-assistant-stats/#voice-asst-infl-car-desc>
88. Voiceflow. Quick start guide. Voiceflow; 2024. <https://developer.voiceflow.com/docs/get-started>
89. Williams K, Peters JC, Breazeal C. Towards leveraging the driver's mobile device for an intelligent, sociable in-car robotic assistant. In: IEEE Intelligent vehicles symposium. 2013.
90. Yadav A. Creación de una habilidad de postre aleatoria personalizada utilizando NodeJS alojado en Alexa. C# Corner; 2020. <https://www.c-sharpcorner.com/article/creating-custom-random-dessert-skill-using-alexa-hosted-node-js/>

91. Rudrawar K, Choudhar N, Meshram A. Voice assisted bots for automobile applications. In: Proceedings of the 3rd International Conference on Advanced Technologies for Societal Applications. Springer; 2021. p. 489-97.
92. Spectra Premium. Sensores de posición de árbol de levas y de cigüeñal. Spectra Premium; 2024. <https://www.spectrapremium.com/es/aftermarket/north-america/camshaft-and-crankshaft-position-sensors>
93. Vistrónica. Regulador LM1117 3.3V. Vistrónica; 2023. <https://www.vistronica.com/componentes-activos/regulador-lm1117-3-3v-detail.html>