ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

Diagnóstico avanzado en sistemas de inyección de gasolina mediante el uso de la red LPWAN - Sigfox para determinar planes de mantenimiento y reparación

Calvopiña Osorio, Fausto Andrés y Gancino Gavilánez, John Renato

Departamento de Ciencias de la Energía y Mecánica

Carrera de Ingeniería Automotriz

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

Ing. Erazo Laverde, Washington Germán MSc.

28 de noviembre del 2022

Latacunga

Reporte de verificación de contenido

Identical Words	796
Words with Minor Changes	368
Paraphrased Words	912
Omitted Words	0

Ing. Erazo Laverde, Washington Germán.

C. C.: 050143263-7

Departamento de Ciencias de la Energía y Mecánica

Carrera de Ingeniería Automotriz

Certificación

Certifico que el trabajo de titulación: "**Diagnóstico avanzado en sistemas de inyección de gasolina mediante el uso de la red LPWAN - Sigfox para determinar planes de mantenimiento y reparación**" fue realizado por los señores **Calvopiña Osorio, Fausto Andrés y Gancino Gavilánez, John Renato** el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Latacunga, 28 de noviembre del 2022

Ing. Erazo Laverde, Washington Germán.

C. C.: 050143263-7

ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Departamento de Ciencias de la Energía y Mecánica

Carrera de Ingeniería Automotriz

Responsabilidad de Autoría

Nosotros, **Calvopiña Osorio, Fausto Andrés y Gancino Gavilánez, John Renato**, con cédulas de ciudadanía N° 050379518-9; 175180502-7,declaramos que el contenido, ideas y criterios del trabajo de titulación: **"Diagnóstico avanzado en sistemas de inyección de gasolina mediante el uso de la red LPWAN - Sigfox para determinar planes de mantenimiento y reparación"** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 28 de noviembre del 2022

Calvopiña Osorio, Fausto Andrés
C.C.: 050379518-9

Gancino Gavilánez, John Renato
C.C.: 175180502-7

Departamento de Ciencias de la Energía y Mecánica

Carrera de Ingeniería Automotriz

Autorización de Publicación

Nosotros **Calvopiña Osorio, Fausto Andrés y Gancino Gavilánez, John Renato**, con cédula/cédulas de ciudadanía N° 050379518-9; 175180502-7, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **"Diagnóstico avanzado en sistemas de inyección de gasolina mediante el uso de la red LPWAN - Sigfox para determinar planes de mantenimiento y reparación"** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi/nuestra responsabilidad.

Latacunga, 28 de noviembre del 2022

Calvopiña Osorio, Fausto Andrés
C.C.: 050379518-9

Gancino Gavilánez, John Renato
C.C.: 175180502-7

Dedicatoria

Este proyecto va de dedicado a mi familia, en especial a mis padres Santiago y Susana que me han apoyado durante este largo camino que es la Universidad, a mi hermana María Paula que me ha apoyado en los buenos y malos momentos que he pasado, a mis tías Consuelo y Eugenia que me daban el impulso para no rendirme nunca.

De igual forma va dedicado para todos los que fueron parte de mi vida universitaria tanto maestros con compañeros, con los cuales hemos compartido tantos momentos.

Fausto Calvopiña

Dedicatoria

Dedico el presente proyecto a mis padres, Verónica Gavilánez y Jorge Gancino quienes me han apoyado y han confiado en mi durante toda la vida y en especial en mi carrera universitaria y han sido mi motivación para esforzarme como ellos lo hacen.

A mi hermana Alison que han sido el ejemplo de dedicación y perseverancia y a mis hermanas Anahí y Daniela que me motivan a ser mejor cada día.

Por último, a mi amigo Rocky, que me acompañó y disfrutamos juntos la etapa universitaria desde el primer hasta el último día de clases.

John Gancino

Agradecimiento

Agradezco a mi familia por apoyarme siempre dándome ánimos para seguir adelante, mis padres Santiago y Susana que han sido un pilar fundamental en este logro ya que me apoyaron tanto en lo económico y emocional, de igual forma en la forma en que me educaron con buenos valores, guiándome con sus consejos para poder obtener este logro tan anhelado.

A los amigos que conseguí en esta etapa, con los cuales con apoyamos en todo y nos dábamos ánimos para seguir adelante en especial a John, Sebas, Andrés, Edwin y Henry.

De igual forma a los maestros que compartieron los conocimientos para formar buenos profesionales, a la Universidad de la Fuerza Armadas ESPE que se esfuerza por tener instalaciones y personal preparado para tener un futuro mejor en nuestro país.

Fausto Calvopiña

Agradecimiento

Quiero agradecer a mis padres Jorge Gancino y Verónica Gavilánez por el apoyo incondicional, emocional como económico, por el buen ejemplo, valores, perseverancia que me han inculcado durante mi vida y sus buenos consejos que me han permitido escribir estas palabras.

A mi novia María Camila por estar presente en mi vida, ser mi apoyo y su ayuda durante el desarrollo del proyecto.

A todos los amigos que forme durante esta etapa de mi vida con los que estudiamos y nos preparamos juntos y sobre todo por su sincera amistad en especial a Andrés, Fausto, Edwin y Jefferson.

A la Universidad de las Fuerzas Armadas ESPE-EL. y su personal que con vocación y dedicación han compartido sus conocimientos.

John Gancino

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

Resumen

Se desarrolló el diagnóstico avanzado en sistemas de inyección de gasolina mediante el uso de la red LPWAN - Sigfox para determinar planes de mantenimiento y reparación con la placa Arduino UNO, realizando la programación en lenguaje propio de la placa para el funcionamiento y recopilación de datos del sistema. Con el módulo Shield can bus V2.0 se recibe la información de la unidad de control del vehículo y para enviar mediante la placa Thinxtra RCZ4 hacia la nube. Se configuró el Callback de Sigfox para el envío de estos datos hacia el dashboard de Ubidots en donde se observa toda la información con indicadores visuales estableciendo el mensaje de alerta en caso de que se pierda la señal del sensor del sistema de inyección o salga de su rango de buen funcionamiento. Se efectuó la comprobación con el escáner automotriz generando el reporte con la información de los sensores de interés para realizar una comparativa y determinar si los datos obtenidos por los módulos son precisos. Se desarrollaron pruebas de ruta para comprobar el correcto funcionamiento de los sensores registrados. Además, se llevó a cabo el análisis comparativo entre los valores registrados y los rangos de funcionamiento de los dispositivos de entrada para obtener el plan de reparación y mantenimiento en caso de ser necesario.

Palabras Clave: *Diagnostico automotriz, red LPWAM, arduino, sigfox, IoT.*

Abstract

Advanced diagnosis in gasoline injection systems was developed through the use of the LPWAN - Sigfox network to determine maintenance and repair plans with the Arduino UNO board, programming in the board's own language for the operation and data collection of the system. system. With the shield can bus V2.0 module, information is received from the vehicle control unit and sent through the Thinxtra RCZ4 board to the cloud. The Sigfox callback was configured to send this data to the Ubidots dashboard where all the information can be seen with visual indicators, establishing an alert message in case the signal from an injection system sensor is lost or it goes out. of its range of good operation. The verification was carried out with the automotive scanner, generating a report with the information of the sensors of interest to make a comparison and determine if the data obtained by the modules is accurate. Route tests were developed to check the correct operation of the registered sensors. In addition, a comparative analysis was carried out between the recorded values and the operating ranges of the input devices to obtain a repair and maintenance plan if necessary.

Keywords: Diagnosis automotive, LPWAM network, arduino, sigfox, IoT.

Capítulo I

Planteamiento del problema de investigación

Antecedentes investigativos

Según (Guanoluisa & Paucar, 2020)establece el concepto del (IOT) internet de las cosas; como la evolución de la interacción de las personas con los componentes electrónicos que existen y que están por venir, con el objetivo de que exista comunicación entre los usuarios y la información que proporcionan los dispositivos. Al conectar los dispositivos a una red, permitirá avanzar el siguiente paso para el desarrollo de la automatización, precisión, domótica y entre otros aspectos que facilitarán la vida a los usuarios.

Según (Bermeo, 2019) la aplicabilidad que existe se maneja en diferentes áreas estratégicas como en corporaciones, empresas e industria automotriz, transporte, atención a la salud, telecomunicaciones, servicios públicos y fabricación usando tecnología de máquina a máquina (M2M). Uno de los grandes beneficios que brinda IoT es la eficiencia operativa, para muchas empresas ha sido un impulso a la innovación generando nuevas ideas que les inyecte mayores ingresos y productividad laboral, obteniendo clientes satisfechos.

Los servicios de conectividad permiten la implementación de IoT, a través de la Red SIGFOX, con un bajo costo, bajo consumo de energía y además un largo alcance, transmitiendo datos sin la necesidad de mantener conexiones de red, administrando todo en la nube en lugar de usar dispositivos.

Según (Torres, 2017) la aplicabilidad del IoT se puede extrapolar a la industria automotriz debido a que existen diferentes tipos de mantenimiento, bien sea predictivo o basado en la condición del vehículo. Para el mantenimiento predictivo se toma una serie de acciones y técnicas con el objetivo de encontrar fallas y defectos en la máquina en etapas

iniciales para evitar daños catastróficos que perjudiquen su operación y ocasionen paros espontáneos. Actualmente el diagnóstico y mantenimiento predictivo de un MEP (Motor de Encendido Provocado), se basa en la medición de los parámetros de funcionamiento, dicha información se recopila desde la inclusión de la electrónica en el vehículo.

Según (Denton, 2016) para el diagnóstico se implementan varios sistemas a bordo del vehículo empezando por la estandarización del OBD I. Debido a las resoluciones de CARB en 1968 se plantea que este debe incluir un sistema de diagnóstico a bordo en todos los vehículos vendidos en California que pesen menos de 14.000 libras, el SAE en 1988 y la Organización internacional de estándares (ISO) en 1989, entregó estándares que eran como OBD.

Después de que existieran varios inconvenientes debido a la poca estandarización de este sistema se introdujo OBD II como menciona (Denton, 2016) en 1994; CARB y la EPA se dieron cuenta de que había muchas fallas con los sistemas OBD de los fabricantes actuales por este motivo la EPA enmendó la Ley de Aire Limpio de 1990 para incluir el requisito de que todos los vehículos vendidos en los Estados Unidos estén equipados con algún tipo de sistema OBD.

El CARB trabajó con la EPA para establecer las reglas para un sistema OBD que incluía códigos de falla estandarizados, ubicaciones de conectores, pines de conectores, información sobre bus de datos, etc. Los principales objetivos de OBD-II son, aumentar el ahorro de combustible asegurándose condiciones óptimas de funcionamiento del motor, reducir las emisiones, de igual forma reducir el tiempo entre una falla del sistema y las notificaciones mediante el monitoreo constante y la comparación de datos aceptables del sistema.

La generación actual de OBD es un sistema muy sofisticado y capaz de detectar problemas de emisiones. Sin embargo, es necesario que el técnico o agente regulador realice este proceso de forma manual. La idea es llevar OBD a un paso más allá, agregando

transferencia de datos remota. Un vehículo equipado con OBD- II podría reportar problemas de emisiones directamente a una autoridad reguladora al igual que lo haría en un futuro él OBD3. Los OBD recolecta los PIDs, estos son parámetros de identificación de los valores escalar que interpretan los sensores, esto muestra en tiempo real las unidades en la que son medidos, desempeño y buen funcionamiento de los componentes.

Planteamiento del problema

El sistema de inyección electrónica de gasolina es usado en la mayoría de los motores (MEP) debido a sus prestaciones como una mayor facilidad de control, menor consumo de combustible y menor emisión de gases contaminantes al ambiente, su buen funcionamiento es crucial para mantener estas características vigentes.

Para mantener el correcto funcionamiento del sistema de inyección electrónico de gasolina, fiabilidad y alargar la vida útil del vehículo, uno de los métodos es llevar a cabo el control periódico preventivo comúnmente basado en el odómetro (indica el kilometraje) o por tiempo, además que puede ayudar a evitar reparaciones costosas en el futuro, brinda seguridad a los ocupantes, buenas condiciones de trabajo y confort.

En varias ocasiones el mantenimiento preventivo no es suficiente para conservar al vehículo en óptimas condiciones, además el descuido de los usuarios o factores externos como la mala calidad de gasolina que se maneja en el país pueden provocar la necesidad de un trabajo correctivo. Para conocer las reparaciones requeridas en el sistema de inyección electrónica de gasolina se necesita el conocimiento previo o personal especializado que pueda interpretar la información que dan los sensores de dicho sistema.

Con el desarrollo tecnológico de la red LPWAN - SIGFOX y la interoperabilidad del IoT (internet de las cosas) se procesan los datos que entregan los sensores en tiempo real,

mediante la interfaz gráfica podrán ser compartidos con el usuario del vehículo, así como con el técnico de servicio para facilitar el diagnóstico, mantenimiento y/o reparación de este.

Descripción resumida del proyecto

En el presente proyecto se obtiene información sobre la aplicación de los conceptos de IoT juntamente con la tecnología LPWAN - SIGFOX, para facilitar el diagnóstico, realizar planes de mantenimiento y reparación para el sistema de inyección electrónica a gasolina mediante contenidos relacionados con el tema del proyecto como: libros, artículos científicos, investigaciones científicas y páginas de internet de contenido confiable.

Fundamentar teóricamente IoT, redes de comunicación, programación, módulo Shield can bus, módulo SIGFOX RCZ4 DEV KIT, Arduino uno, sistemas de inyección de gasolina, OBD-II, planes de mantenimiento, reparación a partir de fuentes bibliográficas y fichas técnicas de los fabricantes.

Realizar el código de programación para el funcionamiento del módulo SHIELD CAN BUS, juntamente con la placa de Arduino UNO y el módulo SIGFOX RCZ4 DEV KIT previo a su instalación que será en al puerto OBDII del vehículo.

Se realiza la base de datos de los rangos de funcionamiento permitidos en los sensores y actuadores (voltaje de señal) (PIDs) según el tipo de vehículos a los que se vaya a monitorear.

Se elabora una ficha de identificación para cada vehículo de la flota donde consten datos del vehículo (marca, modelo, año, placa), datos del propietario (nombres, teléfono de contacto, email), datos técnicos del motor (modelo, tipo de sistema de inyección, cilindraje).

Se obtiene los PIDs mediante el puerto OBD II utilizando tecnologías LPWAN Sigfox para almacenar los datos en la nube.

Se clasifica la información obtenida del OBD-II en la red LPWAN según el sistema automotriz al cual corresponda para verificar sus valores de funcionamiento característicos de cada componente.

Se compara la información recopilada del OBD-II mediante la red LPWAN con los valores de la base de datos previamente investigada que contienen los parámetros correctos de funcionamiento de cada componente.

Se analiza dependiendo la diferencia de valores que exista entre los voltajes de señal de los componentes del sistema de inyección del vehículo con la base de datos para determinar el procedimiento que se deberá seguir al momento de reparar las averías.

Se realiza el plan de mantenimiento al sistema de inyección para los vehículos de la flota según su kilometraje y/o estado de los componentes.

Se analiza todos los parámetros obtenidos para enviar alertas y recordatorios al propietario del vehículo sobre el estado del sistema de inyección, al igual que mantenerlo informado sobre los mantenimientos próximos a realizar.

Justificación e importancia

Al momento de adquirir un vehículo sea nuevo o usado, se busca mantener su seguridad, fiabilidad y alargar vida útil durante el mayor tiempo posible y al menor costo, es por ello por lo que los mantenimientos periódicos y un buen diagnóstico a tiempo es fundamental.

El proceso de diagnóstico automotriz ha ido evolucionando con el tiempo, pasando desde el conocimiento empírico de los técnicos, hasta el uso de sistemas de diagnóstico a

bordo del vehículo (OBD I, OBD II), para lo cual se han ido perfeccionando herramientas que permitan desarrollar esta actividad.

Debido a que el sistema OBD II monitorea y controla el motor y otros dispositivos del vehículo, entre ellos el sistema de inyección, existen muchos más códigos de falla e información, para lo cual se necesita herramientas adecuadas que poseen un precio elevado y necesitan una capacitación previa para el manejo, por lo cual muchos talleres no cuentan con estos y el diagnóstico es más difícil de realizar.

Para lograr un diagnóstico avanzado, rápido y eficiente se pueden implementar módulos que permiten monitorear los parámetros del vehículo como es el SHIELD CAN BUS V2, que es el encargado de la comunicación con el protocolo ya que hace de interfaz entre el puerto serial del Arduino y CAN bus.

Acoplando los conceptos del IoT y el módulo SIGFOX RCZ4 DEV KIT juntamente con Arduino y la interfaz gráfica, se conoce y compara los parámetros del correcto funcionamiento de los elementos del sistema de inyección electrónico a gasolina en tiempo real, esto permite programar mantenimientos con mayor precisión evitando daños graves, además de facilitar y reducir los tiempos de diagnóstico optimizando recursos tanto en los talleres como en los usuarios de los vehículos.

Objetivos

Objetivo general.

Desarrollar el diagnóstico avanzado en sistemas de inyección de gasolina mediante el uso de la red LPWAN - SIGFOX para determinar planes de mantenimiento y reparación.

Objetivos específicos.

- Obtener información sobre la aplicación de los conceptos de IoT en conjunto con la tecnología LPWAN – SIGFOX y el módulo Shield Can Bus V2.0, para facilitar el diagnóstico, realizar planes de mantenimiento y reparación para el sistema de inyección electrónica a gasolina mediante temas relacionados con el tema del proyecto como: libros, artículos científicos, investigaciones científicas, páginas de internet de contenido confiable.

- Fundamentar teóricamente IoT, redes de comunicación, programación, módulo Shield Can Bus V2.0, placa SIGFOX RCZ4 DEV KIT, Arduino uno, sistemas de inyección de gasolina, OBD-II, planes de mantenimiento y reparación a partir de fuentes bibliográficas y fichas técnicas de los fabricantes.

- Realizar el código de programación para el funcionamiento del módulo Shield Can Bus V2.0, juntamente con la placa de Arduino UNO y el módulo SIGFOX RCZ4 DEV KIT previo a su instalación que será en al puerto OBDII del vehículo.

- Generar la base de datos de los rangos de funcionamiento permitidos en los sensores y actuadores del sistema de inyección según el tipo de vehículos a los que se vaya a monitorear.

- Procesar la información obtenida de los sensores y actuadores del sistema de inyección electrónica a gasolina extraída con el módulo Shield Can Bus V2.0 y enviada backend de Sigfox.

- Configurar el Callback para observar los datos obtenidos de la ECU mediante la utilización de widgets en el dashboard.

- Comparar la información recopilada del OBD-II mediante el uso del dashboard con los valores de la base de datos previamente investigada, que contienen los parámetros correctos de funcionamiento de cada componente.

- Analizar los valores obtenidos al de los componentes del sistema de inyección del vehículo con la base de datos para determinar el procedimiento que se deberá seguir al momento de reparar las averías.

- Crear el plan de reparación para los sensores y actuadores cuando los valores obtenidos se encuentren fuera de su rango de funcionamiento optimo.

- Diseñar el plan de mantenimiento al sistema de inyección para los vehículos de la flota según su kilometraje y/o estado de los componentes.

Metas

- Diagnosticar en tiempo real el sistema de inyección electrónica de gasolina para optimizar los recursos del taller y el usuario.

- Aplicar el código de programación que permita la comunicación de la red can del vehículo con la base de datos para su almacenamiento y posterior comparación.

- Realizar los planes de mantenimiento y reparación de los vehículos puestos a prueba según los datos obtenidos de los mismos.

Hipótesis

¿El diagnóstico avanzado del sistema de inyección mediante la red LPWAN - SIGFOX permitirá reducir los tiempos en el desarrollo de planes de mantenimiento y reparación?

Variables de investigación

Sistemas de inyección - Red LPWAN-SIGFOX.

Tabla 1

La variación de los datos obtenidos.

Concepto	Categoría	Indicadores	Ítem	Técnicas	Instrumentos
Los sensores son componentes eléctricos que informan un determinado trabajo del vehículo mediante una magnitud eléctrica	CKP (posición del cigüeñal)	Voltaje	V	Medición - experimentación	Protocolo de pruebas
	TPS (Posición del acelerador)	Voltaje	V	Medición - experimentación	Protocolo de pruebas
	WTS (Temperatura del refrigerante del motor)	Voltaje	V	Medición - experimentación	Protocolo de pruebas
	MAP (Presión de aire en la admisión)	Voltaje	V	Medición - experimentación	Protocolo de pruebas
	VSS (Sensor de velocidad del vehículo)	Voltaje	V	Medición - experimentación	Protocolo de pruebas
	O2 (Sonda lambda)	Voltaje	V	Medición - experimentación	Protocolo de pruebas

Concepto	Categoría	Indicadores	Ítem	Técnicas	Instrumentos
Un actuador es el mecanismo electromecánico que actúa según las señales que le envía la ECU, este actuador envía la retroalimentación a la ECU para informar su estado.	Bobina de alta tensión	Voltaje	V	Medición - experimentación	Protocolo de pruebas
	Bomba de gasolina	Voltaje	V	Medición - experimentación	Protocolo de pruebas
	Inyector	Voltaje	V	Medición - experimentación	Protocolo de pruebas
	ISC (Válvula de control de marcha mínima)	Voltaje	V	Medición - experimentación	Protocolo de pruebas
	EGR (válvula de recirculación de gases)	Voltaje	V	Medición - experimentación	Protocolo de pruebas
	MIL (Luz indicadora de mal funcionamiento)	Voltaje	V	Medición - experimentación	Protocolo de pruebas
PID parámetros de identificación en magnitudes físicas.	Posición del cigüeñal	Grados	°	Medición - experimentación	Protocolo de pruebas
	Posición del acelerador	Porcentaje	%	Medición - experimentación	Protocolo de pruebas

Concepto	Categoría	Indicadores	Ítem	Técnicas	Instrumentos
	Temperatura del refrigerante del motor	Temperatura	°C	Medición - experimentación	Protocolo de pruebas
	Presión de aire en la admisión	Presión	kPa	Medición - experimentación	Protocolo de pruebas
	Velocidad del vehículo	Velocidad	Km/h	Medición - experimentación	Protocolo de pruebas
Red LPWAN-SIGFOX es la red de obtención de datos con una larga cobertura y un bajo consumo de energía.	Obtención de datos	-	-	Análisis	Protocolo de pruebas
	Subida de datos	-	-	Análisis	Protocolo de pruebas
	Procesamiento de datos	-	-	Análisis	Protocolo de pruebas

Variables dependientes

Planes de mantenimiento – Reparación.

Tabla 2

Operacionalización de la variable dependiente.

Concepto	Categoría	Indicadores	Ítem	Técnicas	Instrumentos
Los planes de mantenimiento son el conjunto de intervenciones u operaciones que se realizan a los vehículos para prevenir posibles fallas.	**Datos de entrada**	**Kilometraje**	**Km**	**Análisis**	**Protocolo de pruebas**
		Códigos de falla intermitentes	**-**	**Análisis**	**Protocolo de pruebas**
	Datos de salida	**Mensaje de aviso**	**-**	**Análisis**	**Protocolo de pruebas**
Reparación de fallas, al obtener los datos y que el sistema de inyección se encuentre en mal estado se deberá realizar la reparación y cambio del elemento que está fallando.	**Datos de entrada**	**Códigos de falla continuos**	**-**	**Análisis**	**Protocolo de pruebas**
	Datos de salida	**Mensaje de aviso**	**-**	**Análisis**	**Protocolo de pruebas**

Metodología de desarrollo del proyecto

Para el desarrollo del proyecto de investigación se van a utilizar diferentes metodologías y técnicas en diversas ramas de la ciencia. En el proceso de investigación se van a aplicar procedimientos con mayor o menor incidencia, al realizar el proyecto los métodos empleados son comparativo, experimental, medición, científico, inductivo y deductivo que permiten llegar a cumplir con los objetivos trazados.

Método comparativo

Para la investigación, el método comparativo permitirá establecer similitudes y diferencias en los voltajes de señal que da la red LPWAN – SIGFOX, los datos obtenidos y tabulados permiten dar el diagnóstico del sistema de inyección.

Método experimental

En base a este método se realizarán pruebas sobre la obtención de los datos mediante la red LPWAN - SIGFOX, al igual que se obtendrá el diagnóstico mediante el análisis de los datos en la nube donde se encuentran almacenados.

Método de medición

Con este método se obtendrán valores medidos de los voltajes de señal de los sensores y actuadores del sistema de inyección, para comprobar su funcionamiento.

Método científico

Este método se utilizará para dar el diagnóstico utilizando la comparación de los datos obtenidos mediante la red LPWAN - SIGFOX y los documentos científicos consultados al inicio

de la investigación, al igual que manuales de mantenimiento según el vehículo que se está diagnosticando.

Método inductivo

Este método permitirá obtener los voltajes que se encontraban fuera de rango, con lo cual detecta las fallas en tiempo real, después de realizar el análisis de los datos almacenados en la nube.

Método deductivo

Mediante el método deductivo se analizarán los datos obtenidos de la red LPWAN - SIGFOX, así como los valores de funcionamiento (Voltaje, resistencia, intensidad, frecuencia, entre otros) de los componentes del sistema de inyección para realizar la comparación entre ellos y ejecutar el plan de mantenimiento y los procesos de reparación según varíen los datos unos de otros.

Capítulo II

Marco teórico

Sistema de inyección de gasolina

Funcionamiento del sistema de inyección.

Para el correcto funcionamiento de un motor a gasolina la ratio ideal es 14.7:1 esto quiere decir que, para quemar correctamente la gasolina, por cada renovación de la carga aire/combustible deberán entrar 14.7 partes de aire y 1 de gasolina. Este es la cantidad teórica de aire requerida para quemar el combustible. Se le da un valor 'lambda (λ)' de 1 según (Denton, 2016, págs. 168–169)

Dicha relación aire/combustible se altera en los siguientes casos:

- Arranque en frio: es cuando se necesita una mezcla para compensar la condensación del combustible.
- Carga o aceleración: la mezcla deberá ser rica para mejorar la performance del vehículo.
- Velocidad crucero o cargas livianas: para economizar combustible y evitar mayores emisiones.

En la actualidad la mayoría de los vehículos están comandados electrónicamente, contando así con una cantidad de sensores y actuadores para su funcionamiento. Esto permite que el sistema de inyección se ajuste cerca de los requisitos óptimos del motor. Este proceso se ejecuta durante el desarrollo en bancos de pruebas y en el dinamómetro, así como en el diseño del automóvil. Los datos ideales de operación para una gran cantidad de procesos del automóvil se almacenan en la ROM en la ECU. Este estrecho control de cantidad de combustible inyectado permite el ajuste óptimo cuando se tienen en cuenta con diversos factores en el uso diario.

Los fabricantes de automóviles diseñan los motores de diferentes formas con la finalidad de aprovechar el rendimiento y la eficiencia de la máquina lo mejor posible, dando como resultado varios tipos de sistemas de inyección.

Estos sistemas se encuentran divididos según su forma y tipo de inyección (Montero & Paguay, 2021):

Por el lugar donde sucede la inyección

- Por la cantidad de inyectores del sistema
- Por la cantidad de inyecciones

Para motivos del proyecto se utilizó el sistema de inyección directa multipunto debido a la compatibilidad que tiene el protocolo CAN BUS ISO 15765 CAN con el módulo SEEED CAN BUS y los pines 6 y 14.

Sensores del sistema de inyección de gasolina

Para el funcionamiento del sistema de inyección electrónico a gasolina es indispensable el buen funcionamiento de los sensores, que son los encargados de enviar la información del comportamiento del motor y las condiciones en las que este se encuentra trabajando a la unidad de control, con esto se permite que los actuadores se accionen de forma efectiva estableciendo el correcto régimen de funcionamiento como menciona (Correa, 2013) en su tesis.

Figura 1

Bahía del motor del vehículo Kia Rio 2019

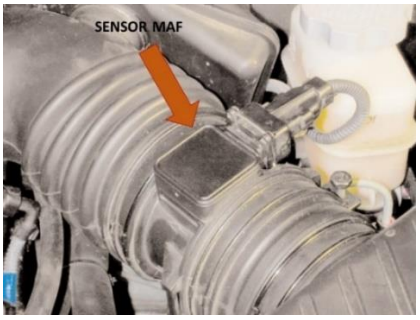
Nota. En la figura se observa señalado con una línea entre cortada roja el sensor MAF instalado correctamente por (Kia, 2019).

Sensor de flujo de aire MAF (Mass Air Flow).

La función del sensor MAF es determinar el porcentaje de aire que entra al motor, este se encuentra en una ubicación estratégica que es entre el cuerpo de aceleración y el filtro de aire. Está compuesto en su interior por la resistencia llamada hilo caliente, el cual se calienta hasta los 100 C, debido al voltaje que recibe y éste a su vez se enfría con el aire que pasa a la admisión, el cambio llega a la ECU en forma de señal, logrando que se ajusten los parámetros de inyección de gasolina (Montero & Paguay, 2021).

Figura 2

Sensor de flujo de la masa de aire (MAF)



Nota. En la figura se observa señalado con una flecha roja el sensor MAF instalado correctamente (Kia, 2019)

Sensor de presión de aire MAP (Maniflod Absolute Pressure Sensor).

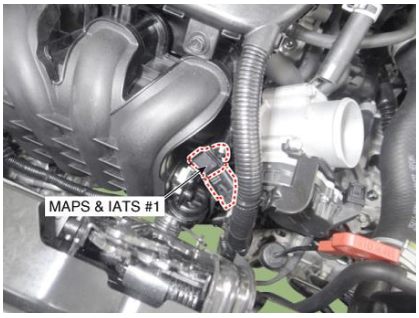
La principal función de este sensor es medir el volumen de aire que entra al colector de admisión calculando la presión en el mismo, de esta manera informa a la ECU mediante señales eléctricas y ésta realiza el ajuste para le mezcla aire-combustible que necesite el motor.

Se encuentra localizado sobre el múltiple de admisión, en donde existe el conducto de vacío para su buen funcionamiento. La gran mayoría de vehículos usan el sensor piezoeléctrico, es decir que contienen cristales de silicio, los cuales están sometidos a la presión del vacío de

donde se encuentran ubicados y esto permite que varíe su resistencia(Montero & Paguay, 2021).

Figura 3

Sensor de presión de aire (MAP)



Nota. En la figura se observa señalado con una línea entre cortada roja el sensor MAP conjuntamente con el IATS instalado correctamente en el múltiple de admisión por (Kia, 2019).

Sensor de posición del cigüeñal (CKP).

Encargado de captar la velocidad del motor, las revoluciones por minuto y la posición del cigüeñal, estas mediciones son enviadas a la ECU para calcular los pulsos y la sincronización del arco eléctrico de las bujías con la inyección del combustible en el cilindro en cualquier régimen que se encuentre el motor según (Obregón, 2016).

Figura 4

Sensor De Posición Del Cigüeñal (CKP)

Nota. Se muestra en la figura indicado con una flecha roja el sensor CKP ubicado a un costado de la rueda dentando del cigüeñal por (Kia, 2019).

El sensor de efecto hall transmite la señal eléctrica de acuerdo con la velocidad de giro de las ranuras del engranaje del árbol de levas.

Sensor de temperatura del refrigerante (ECT).

Es el sensor encargado de registrar los cambios de temperatura del motor mediante la medición del refrigerante, es básicamente un transmisor que puede ser medido con facilidad. La ubicación más común del ECT es antes del termostato, los datos que entrega ayudan a varias funciones del motor como, la inyección de combustible, sincronización de válvulas ya que conociendo la temperatura del motor la ECU calcula la entrega de combustible, así lo menciona (Obregón, 2016).

Figura 5

Sensor de temperatura del refrigerante (ECT)

Nota. Dentro de la figura se muestra el sensor de temperatura del refrigerante que se encuentra a la salida del líquido refrigerante del motor (Kia, 2019).

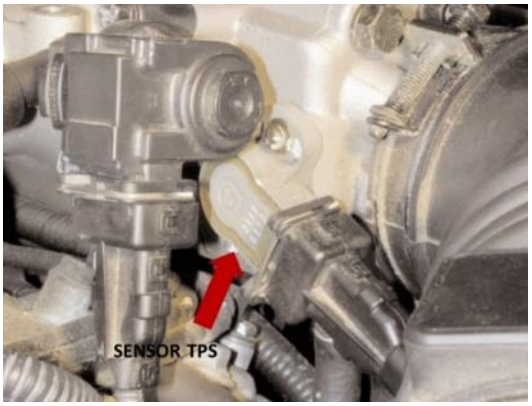
El sensor de temperatura es un termistor, que varía de resistencia a medida que la temperatura cambia, existe termistores que aumenta de resistencia acorde aumenta la temperatura, y viceversa.

Sensor de posición de la mariposa de aceleración (TPS).

Se encarga de indicar el ángulo de apertura de la mariposa lo que genera que exista un mayor flujo de aire en caso de estar abierta y una aceleración o desaceleración en caso de estar cerrada debido al cálculo que realiza la ECU para la inyección de combustible mediante dichas señales recibidas (Correa, 2013).

Figura 6

Sensor de posición de la mariposa de aceleración (TPS)



Nota. Señalado mediante una flecha de color rojo se puede observar el sensor de posición de la mariposa acoplado al costado del cuerpo de aceleración Kia OBII Training (2019).

Ubicado en el cuerpo de aceleración, este sensor es similar a un potenciómetro con resistencia variable, se alimenta de una fuente de 5 voltios, que varía según la posición del eje y la resistencia, cuenta con 3 terminales en su socket donde se puede medir la señal, voltaje y la conexión a tierra o masa (Correa, 2013).

Actuadores del sistema de inyección de gasolina

Los actuadores se conocen como elementos electromecánicos a mando de la ECU que por medio de los datos enviados por los sensores permite realizar acciones específicas a cada actuador para mantener el vehículo funcionando según el régimen y las condiciones que se

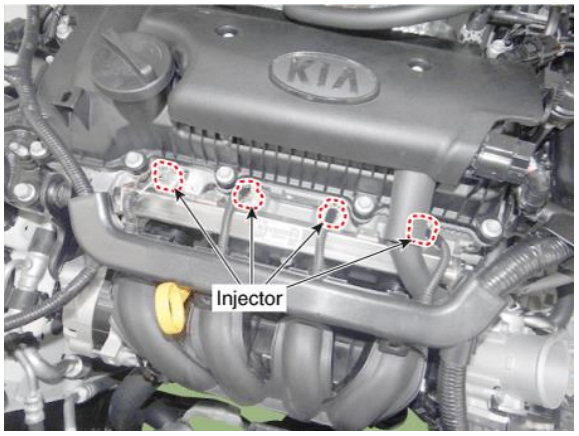
necesiten, dichos elementos actúan a través de corriente 12 voltios o pulsos negativos. (Montero & Paguay, 2021)

Inyectores.

El inyector es un elemento electromecánico alimentado con una fuente de 12 voltios, dicho voltaje estará presente cuando el motor se encuentre encendido o en marcha por que está controlado por un relé que funciona solo cuando hay una señal de velocidad disponible desde el motor. Los inyectores tendrán un suministro de voltaje constante mientras el motor esté funcionando y la ruta a tierra se conmutará a través de la ECU, cuando se quita la conexión a tierra el inyector recibe voltaje y se registrará un pico cercano a 60 v. Si el voltaje pico es aproximado es de 35 v es debido a la presencia de un diodo Zener en la ECU para mantener el voltaje (Obregón, 2016).

Figura 7

Inyectores Kia Rio.



Nota. Se observa en la figura resaltados con líneas rojas entre cortadas los 4 inyectores correspondiente a los 4 cilindros del vehículo por (Kia, 2019).

Bobinas independientes COP.

El encendido directo es, en cierto modo, la continuación del encendido sin distribución. Este sistema utiliza una bobina inductiva para cada cilindro. Estas bobinas se montan

directamente en las bujías. El uso de una bobina individual para cada bujía asegura que el tiempo de subida del devanado primario de baja inductancia sea muy rápido. Esto asegura que se produzca una chispa de muy alto voltaje y alta energía. Este voltaje, que puede superar los 400 kV, proporciona un inicio eficiente del proceso de combustión en condiciones de arranque en frío y con mezclas débiles. Algunos sistemas de encendido directo usan encendido por descarga de capacitor. (Denton, 2016).

Figura 8

Bobinas de Encendido

Nota. Marcado en una línea entre cortada se aprecian las bobinas independientes del motor del vehículo KIA Rio. Tomado de (Kia, 2019)

Voltajes de funcionamiento de los sensores del sistema de inyección electrónico de gasolina

Tabla 3

Voltajes de funcionamiento de sensores

Denominación	Voltaje De Referencia	Voltaje De Señal
Sensor De Presión De Aire MAP (Maniflod Absolute Pressure Sensor)	5Vcc	0,5 a 4,5Vcc
Sensor De Posición Del Cigüeñal (Ckp)	5V	-5 a 5Vca
Sensor De Temperatura Del Refrigerante (Ect)	5V	0,5 a 4,5Vcc
Sensor De Posición De La Mariposa De Aceleración (Tps)	5V	0,5V ralentí 4,5V Abierto

Tabla 4

Valores de operación de los sensores.

Sensor	Valor mínimo	Valor máximo	Unidad de medición
ECT (Engine coolant temperature)	-40	215	°C
TPS (Throttle position)	0	100	%
IAT (Intake Air Temperatura)	-40	215	°C
MAP (Manifold Absolute Pressure)	0	255	kPa
Velocidad de rotación del motor	0	16.384	rpm
Velocidad del vehículo	0	255	Km/h

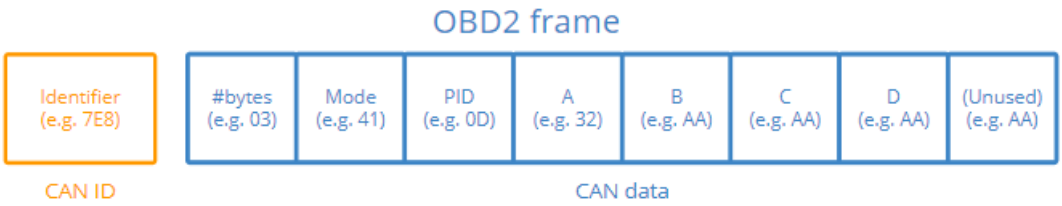
Nota. En la tabla se indican los valores que pueden tener los sensores mostrados que fueron extraídos. Tomado de (CSSELECTRONICS, 2022a).

Identificación de parámetro (PIDs)

Para comenzar a registrar datos OBD2, es útil comprender los conceptos básicos de la estructura del mensaje OBD2 sin procesar. En términos simplificados, el mensaje OBD2 se compone del identificador y datos. Además, los datos se dividen en Modo, PID y bytes de datos (A, B, C, D) como se muestra a continuación.

Figura 9

Estructura de los mensajes OBDII



Nota. En la figura se muestra la estructura que manejan los mensajes que envía la ECU, teniendo en un cuadro amarillo el identificador del PID y los siguientes siete bytes el mensaje en formato hexadecimal del valor solicitado. Tomado de (CSSELECTRONICS, 2022).

Identificador: para los mensajes OBD2, el identificador es estándar de 11 bits y se utiliza para distinguir entre "mensajes de solicitud" (ID 7DF) y "mensajes de respuesta" (ID 7E8 a 7EF). Tenga en cuenta que 7E8 normalmente será donde responde el motor principal o la ECU.

Longitud: esto simplemente refleja la longitud en número de bytes de los datos restantes (03 a 06). Para el ejemplo de Velocidad del vehículo, es 02 para la solicitud (dado que solo siguen 01 y 0D), mientras que para la respuesta es 03, ya que siguen 41, 0D y 32.

Modo: Para solicitudes, será entre 01-0A. Para las respuestas, el 0 se reemplaza por 4 (es decir, 41, 42, …, 4A). Hay 10 modos como se describe en el estándar SAE J1979 OBD2. El modo 1 muestra los datos actuales y se utiliza, por ejemplo, para observar la velocidad del vehículo en tiempo real, las RPM, etc.

PID: para cada modo, existe una lista de PID OBD2 estándar; por ejemplo, en el modo 01, PID 0D es la velocidad del vehículo. Cada PID tiene una descripción y algunos tienen un mínimo/máximo específico y su fórmula de conversión.

Protocolos de Comunicación

Los protocolos de comunicación son el conjunto de normas y estandarizaciones que se deberá cumplir entre las máquinas y programas que intervienen en la comunicación de datos mediante equipos, sin los cuales la comunicación resultaría difícil de realizar, por lo cual se puede decir que un protocolo de comunicación es el idioma en los cuales diferentes módulos intervienen y se transmite datos.

Un protocolo CAN es indispensable para la interconexión de los diversos módulos de control que posee un vehículo debido a que ofrece velocidades de comunicación 50 a 100 veces más rápidas que los protocolos típicos es por ello que varios fabricantes adoptaron diferentes tipos.

SAE J1962: Este estándar define el conector físico utilizado para la interfaz OBD2, es decir, el conector OBD2. El estándar describe tanto el conector OBD2 del vehículo como el conector utilizado por el equipo de prueba externo (por ejemplo, un escáner OBD2 o un registrador de datos OBD2). En particular, el estándar dicta la ubicación y el acceso al conector OBD2 (SAE Standard, 2001).

SAE J1979: El estándar SAE J1979 describe los métodos para solicitar información de diagnóstico a través del protocolo OBD2. También incluye dentro de la lista de ID de parámetros OBD2 públicos estandarizados (OBD2 PID) que los OEM automotrices pueden implementar en los automóviles (aunque no están obligados a hacerlo). Los OEM de vehículos también pueden decidir implementar PID OBD2 patentados adicionales más allá de los descritos por el estándar SAE J1979 (SAE Standard, 2002).

SAE J1939: El estándar J1939 describe el protocolo de datos utilizado para la comunicación de vehículos pesados. Si bien la información PID OBD2 solo está disponible a pedido del equipo de prueba OBD2, el protocolo J1939 se usa en la mayoría de los vehículos pesados como el medio básico para comunicar el tráfico CAN, lo que significa que los datos se transmiten continuamente (*Csselectronics 2.Pdf*, n.d.).

ISO 15765-2: El estándar describe la 'Capa de transporte', es decir, cómo enviar paquetes de datos que exceden los 8 bytes a través del bus CAN. Este estándar es importante ya que forma la base para la comunicación de los Servicios de diagnóstico unificados, que se basa en el envío de paquetes de datos CAN multitrama (SAE Standard, 2016).

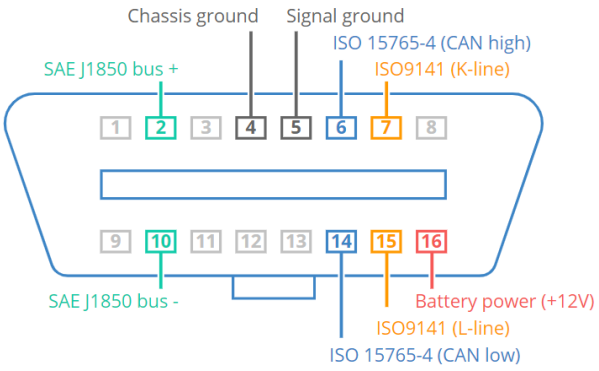
DLC

El conector de comunicación serial que se encuentran en los vehículos con tecnología OBD II o Diagnostic Link Connector conocido como DLC consta de 16 pines para establecer la comunicación entre el equipo de diagnóstico y la ECM del vehículo, aunque no todos son usados a la vez en ningún protocolo. El conector OBD2 le permite acceder fácilmente a los datos de su automóvil. El estándar SAE J1962 especifica dos tipos de conectores hembra OBD2 de 16 pines (A y B).

En la figura, se muestra un ejemplo de un conector pin OBD2 tipo A (también conocido como conector de enlace de datos, DLC). Se deberá mantener en cuenta que conector OBD2 está cerca de su volante, pero puede estar oculto detrás de cubiertas además que el pin 16 suministra energía de la batería, el pinout OBD2 depende del protocolo de comunicación y para identificar esto, aunque el protocolo más común es CAN (a través de ISO 15765), lo que significa que los pines 6 (CAN-H) y 14 (CAN-L) normalmente se conectarán (López, 2014).

Figura 10

Conector DCL 16 Pines



Nota. Se muestra en la figura la disposición del conector DCL y los pines que usa cada protocolo de comunicación, para el presente proyecto se necesita el ISO 15765. Tomado de (CSSELECTRONICS, 2022)

Mantenimiento preventivo

El mantenimiento preventivo, o mantenimiento técnico planificado, consiste en trabajar en los equipos a intervalos regulares o según criterios predefinidos. Su objetivo principal es reducir el riesgo de averías en bienes, máquinas y equipos, pero también ayuda a conseguir objetivos más generales. Por esta razón, la mayoría de las fábricas y empresas buscan aumentar la proporción de mantenimiento preventivo implementado en contraposición al mantenimiento correctivo.

Tabla 5

Mantenimiento del vehículo en función del kilometraje.

Elemento de mantenimiento	Kilometraje (Km)
Aceite y filtro de motor	**7.500**
Bujías	**75.000**
Refrigerante del motor	**30.000**
Filtro de combustible	**30.000**
Aceite de transmisión	**120.000**
Forros y tambores de freno	**300.000**
Líquido de freno y embrague	**15.000**
Rótulas de suspensión	**15.000**
Estado de la batería	**15.000**

Nota. En la tabla se observa el kilometraje en el que se debería realizar el mantenimiento de cada elemento, estos datos se tomaron del manual de mantenimiento del vehículo.

Arduino uno

Arduino es la Plataforma de código abierto que se utiliza en la electrónica, es muy utilizada en proyectos debido a su versatilidad ya que se basa en el hardware y software libre, es fácil de usar. (Torrente, n.d.)En este tipo de placas se conectan periféricos a las entradas y salidas, dentro de estos circuitos se graban instrucciones, que se escriben mediante el lenguaje de programación, la información se ingresa por los periféricos de entrada y se traslada al microcontrolador el cual se encarga de procesar los datos, para después la información ya procesada se muestre en periféricos de salida que reproducen los datos que se necesitan. (Blum, 2019)

El Arduino UNO es una de las placas más utilizadas, es la tarjeta de desarrollo open-sourse (fuente abierta), se la programa en el entorno de desarrollo que implementa el lenguaje processing/wiring (procesamiento/cableado). Se puede comunicar con la PC mediante el puerto serial utilizando lenguajes como flash, Processin, MaxMSP, etc. (Atmel, 2015)

Figura 11

Arduino uno

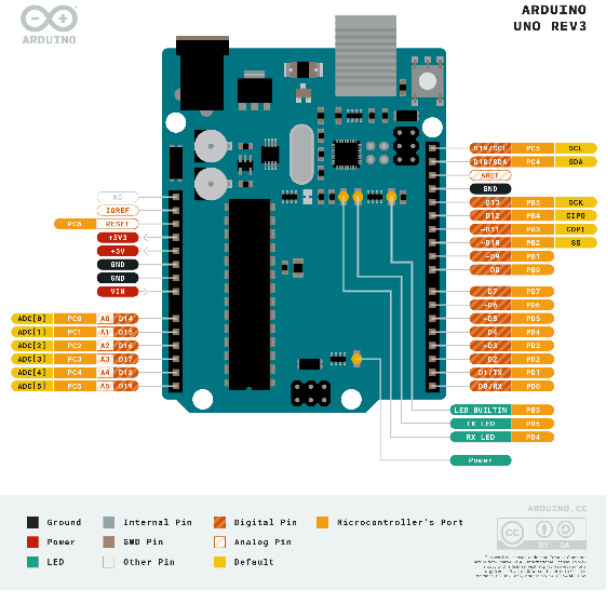
La memoria del ATmega 328T tiene 32 KB de memora Flash (se utiliza para guardar el sketch ya compilado, de aquí se ejecuta el programa) que se utilizan para almacenar el código, 0,5 KB son usados para el gestor de arranque, tiene 2 KB de SRAM (memoria volátil utilizada como banco de registros) y 1 KB de EEPROM (memoria no volátil donde se mantienen los datos después de realizar un reset).

Distribución de pines del Arduino uno.

El Arduino uno posee 14 pines de entradas/salidas digitales de los cuales 6 pueden utilizarse como salidas PWM, 6 entradas analógicas, 1 UART (puerto serial por hardware), un cristal oscilador de 16 MHz, conector USB, Jack de alimentación, conector ICSP y botón de reset, cada uno de los pines está conectado con el procesador del Arduino que realizan una función en específico.

Figura 12

Diagrama de distribución de pines del Arduino uno



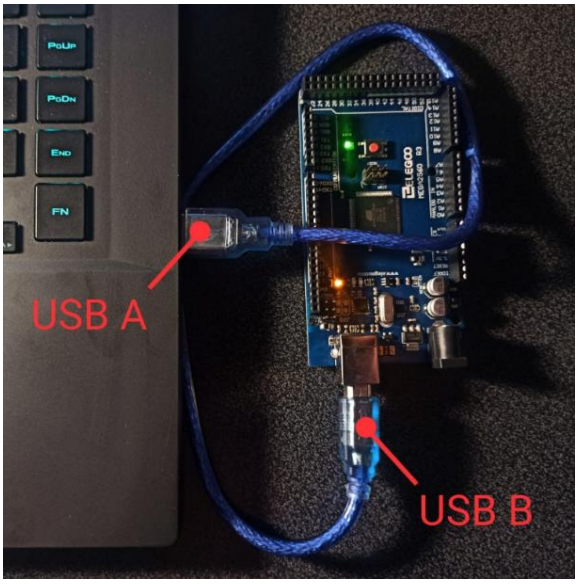
Nota. En la figura indica el Arduino uno, detallando cada uno de sus pines y la función que cumplen. Tomado de (Arduino Store, 2022)

Conexión del Arduino con la PC.

El Arduino se conecta con la PC mediante un cable, por el cual se sube la programación al microcontrolador, en un extremo tiene el conector USB tipo A que se conecta en la PC, en el otro extremo tiene el conector USB tipo B que va conectado a la placa Arduino como se muestra en la figura 13, de esta forma se energiza la placa, también se podría energizarla con baterías, esto queda al criterio del usuario o el uso que se le vaya a dar.

Figura 13

Conexión del Arduino con la PC



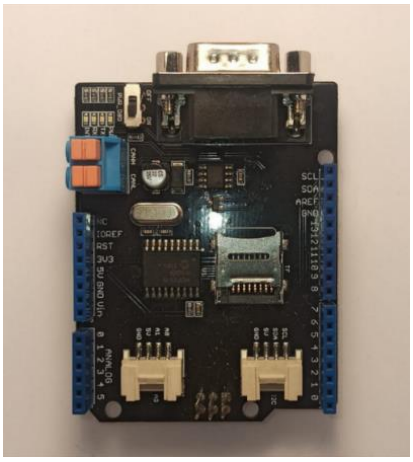
Nota. En la figura se ve como se conecta el Arduino a la PC para poder subir el programa al microcontrolador.

Seeed Shield CAN BUS V2.0

Seeed Shield Can Bus V2.0 es el módulo compatible con Arduino que integra un controlador MCP2515 CAN-BUS, con el cual se puede interactuar con la ECU de un automóvil mediante el puerto OBD II. Mediante la red CAN-BUS el módulo se comunica con la ECU permitiendo la adquisición de datos, utilizando CAN_H y CAN_L, se puede utilizar para comunicarse entre módulos que utilicen el protocolo can bus, donde se pueden enviar o recibir mensajes. (Microchip Technology Inc., 2007)

Figura 14

SEEED SHIELD CAN BUS V2.0

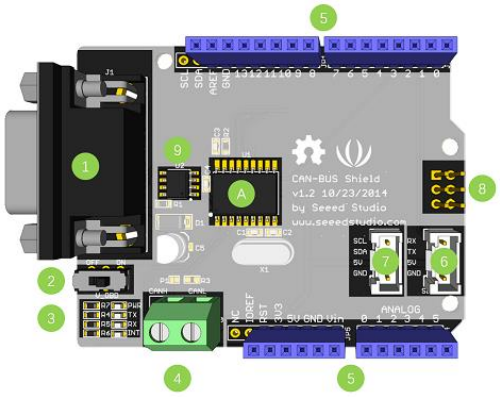


Nota. La figura indica el módulo CAN-BUS de la marca Seeed, el cual se puede conectar con el Arduino uno.

Elementos del Seeed Shield CAN BUS V2.0.

Figura 15

Descripción de los elementos del Shield CAN BUS



Nota. se muestra cada uno de los elementos que forman el Shield CAN BUS. Tomado de (Seeed Studio, 2022)

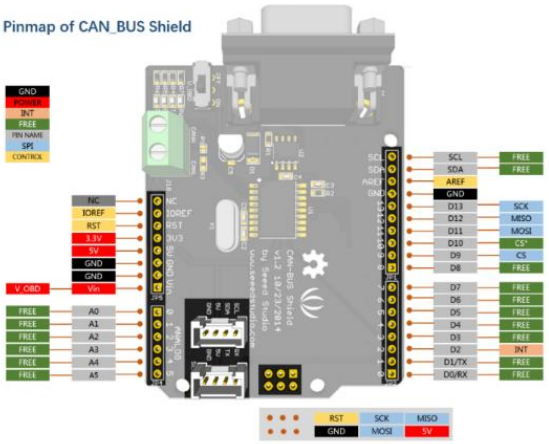
En la figura 15 se muestra los elementos que forman el Shield CAN BUS que a continuación se va a numerar:

1. **Interfaz DB9:** se utiliza para conectar el módulo al conector OBD II mediante el cable OBD-serial.
2. **V_OBD:** interruptor de encendido (on/off) del módulo.
3. **Indicador de funcionamiento:**
    - **PWR:** potencia.
    - **TX:** el indicador parpadea cuando se están enviando los datos.
    - **RX:** el indicador parpadea cuando se recibe los datos.
    - **INT:** interrupción de datos.
4. **Terminal CAN_H y CAN_L**
5. **Pines de conexión con el Arduino**
6. **Conector Grove serie**
7. **Conector I2C**
8. **Pines ICSP**
9. **IC – MCP2515: controlador CAN independiente con interfaz SPI**

Pines utilizados para la conexión entre el Shield CAN BUS y el Arduino.

Figura 16

Pines utilizados para la conexión entre el Shield CAN BUS y el Arduino.



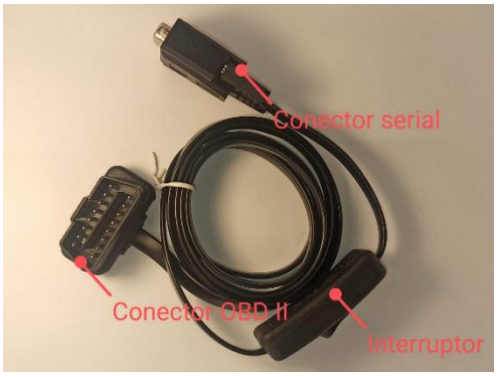
Nota. en la figura se ve que los pines que se encuentran señalados con color negro son tierra, los que se encuentran señalados con color rojo son de alimentación, los que están señalados con color verde son pines libres, los que están señalados con color celeste se utilizan para el protocolo de comunicación SPI y los pines que están señalados con color amarillo son los de control. Tomado de (Seeed Studio, 2022)

Cable OBD II – Serial.

Este cable se lo utiliza para establecer la conexión de la ECU con el Shield CAN BUS, en uno de sus extremos tiene el conector serial DB9 que posee 9 puntos de conexión, también se tiene el interruptor con el cual se reinicia la conexión de ser necesario y en el otro extremo hay el conector OBD II que posee 16 pines.

Figura 17

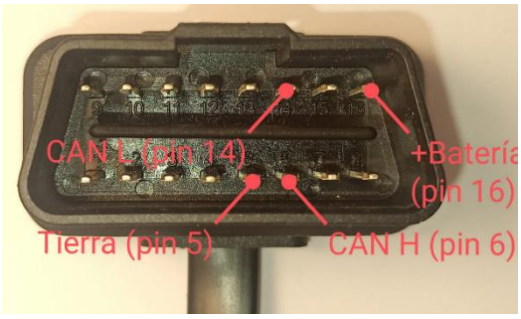
Cable OBD II serial.



En el conector OBD se utiliza los pines 5, 6, 14 y 16 como se puede observar en la figura 18 debido al protocolo de comunicación que tiene el Shield CAN BUS que es el ISO 1725.

Figura 18

Conector OBD II.

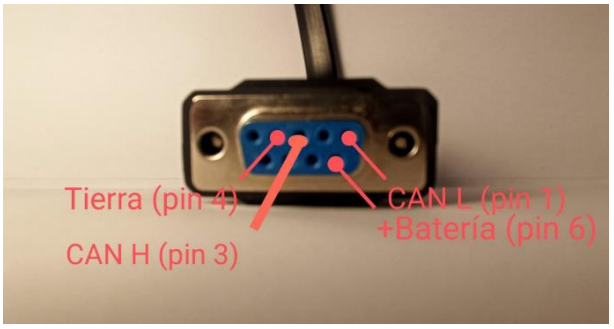


Nota. en la figura se ve los pines que son utilizados para la comunicación entre la ECU y el Shield CAN BUS mediante el protocolo ISO 1725, donde el pin 5 es tierra, el pin 6 es CAN H, el pin 14 es CAN L y el pin 16 es el del voltaje de la batería (12V).

En el conector serial DB9 se van a utilizar los pines de conexión 1, 3, 4, 6, como se observa en la figura 19.

Figura 19

Conector serial DB9.



Nota. en la figura se mira los pines que son utilizados del conector DB9, donde el pin 1 es CAN L, el pin 3 es CAN H, el pin 4 es tierra y el pin 6 es el del voltaje de la batería (12V).

Internet Of The Things (IoT)

El IoT no es un concepto nuevo dentro de la industria, este concepto es simple, pero de gran ayuda para la industria, donde los objetos que se utilizan de forma cotidiana se encuentran equipados con conexión inalámbrica, para que se puedan comunicar entre sí y al mismo tiempo puedan ser controlados mediante computadoras conectadas a la misma red. (Bonilla-Fabela et al., 2016)

La evolución que ha tenido el internet con la interconexión que tienen las personas al utilizar dispositivos inteligentes, que pueden crear un vínculo entre las cosas que se utilizan a diario, de forma que se pueda controlar el ambiente mediante información obtenida, para poder procesarla y tomar decisiones sobre qué es lo que se va a realiza y si va a beneficiar al desarrollo de nuestra vida cotidiana. (Bonilla-Fabela et al., 2016)

Internet Of The Things Industria Automotriz.

Dentro de la industria automotriz el uso del IoT (internet de las cosas) se utiliza para los adaptadores inteligentes de diagnóstico que ayudan a obtener datos en tiempo real con la finalidad de saber si existen fallas que pueda tener el vehículo a la hora de utilizarlo, también se lo utiliza para obtener un sistema de simulación vehicular que ayuda a desarrollar aplicaciones y probarlas simulándolas de modo que no se invierte en hardware, con lo que se pretende obtener un análisis de los datos y el comportamiento de estos dispositivos y poder ver si es factible su construcción. (Barelli et al., 2018)

El monitoreo del consumo de combustible se lo puede realizar a con el uso del IoT con lo que se realiza un análisis de comportamiento de manejo durante una ruta, esto indica cómo es la eficiencia del consumo de combustible según el comportamiento del conductor durante el manejo en ruta. (Barelli et al., 2018)

Otro de los grandes aportes del IoT en la industria automotriz es el seguimiento de flotas, la cual ayuda a rastrear cualquier tipo de flota de vehículos para mejorar su eficiencia operativa, con lo cual se puede monitorear aspectos como si el vehículo viaja a exceso de velocidad, paradas, frenado, aceleraciones, también se puede obtener los reportes de las trayectorias realizadas por dichos automotores. (Barelli et al., 2018)

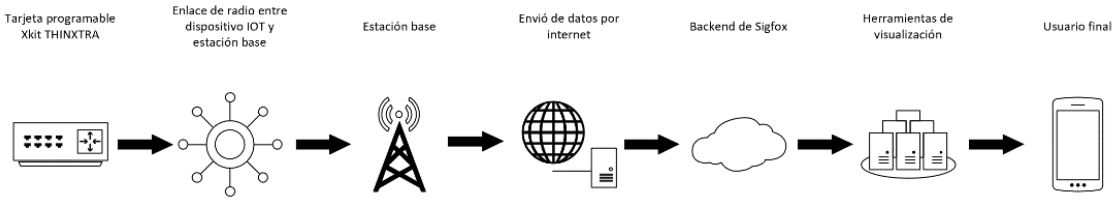
Red LPWAN

La red LPWAN es una red de bajo consumo y de extensa área, es el sistema inalámbrico de comunicación que facilitan la transmisión de datos a larga distancia, este tipo de red permite instalar numerosos nodos lo que se traduce en una mayor cobertura, este tipo de red es muy utilizada debido a que sus elementos finales (sensores) pueden ser alimentados con baterías, por lo que es una buena elección en el campo automotriz.

Lo más importante dentro de estas redes de comunicación es la aplicación en la innovación tecnológica, que ayudan a optimizar los mantenimientos, seguimientos de los vehículos y sistemas de transporte (Allauca & Bryan, 2020).

Figura 20

Configuración de la red LPWAN



Nota. La figura indica la configuración y elementos que tiene la red LPWAN.

Thinxtra Sigfox Developer XKIT

Es el kit que se utiliza dentro de la red LPWAN, el cual consta del Arduino uno, el módulo SIGFOX, la antena y la porta baterías para poder energizar el módulo. SIGFOX es una empresa enfocada en el IOT (el internet de las cosas), por lo que esta red se diseñó para comunicaciones de baja velocidad y bajo consumo de energía en los dispositivos conectados (Gómez et al., 2019). Cada dispositivo SIGFOX tiene un ID único, que es utilizado para enrutar y firmar los mensajes, de la misma manera se utiliza para autenticar el dispositivo.

Figura 21

Thinxtra Sigfox developer XKIT

Nota. La figura indica la configuración del módulo Thinxtra Sigfox.

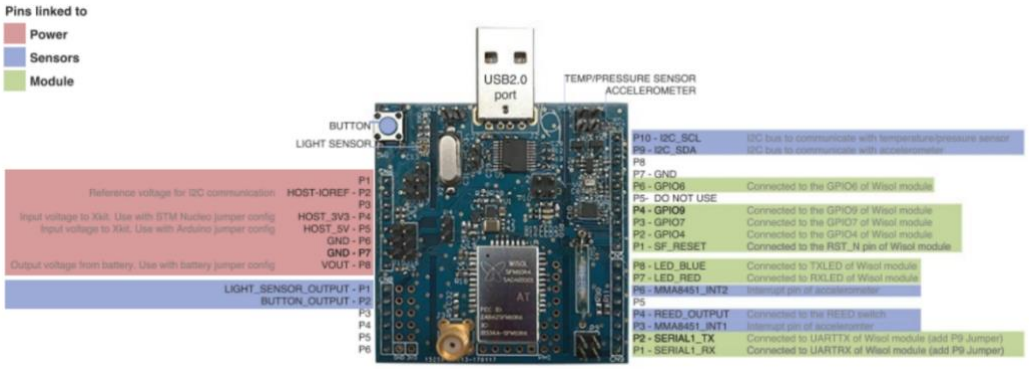
Las características que tiene el módulo Thinxtra Sigfox son:

- Acelerómetro de 3 ejes
- Sensor digital de temperatura y presión
- Reed switch
- Sensores de luz
- Leds rojo y azul
- Pulsador

Configuración de pines del módulo Thinxtra Sigfox.

Figura 22

Pines Thinxtra Sigfox developer XKIT

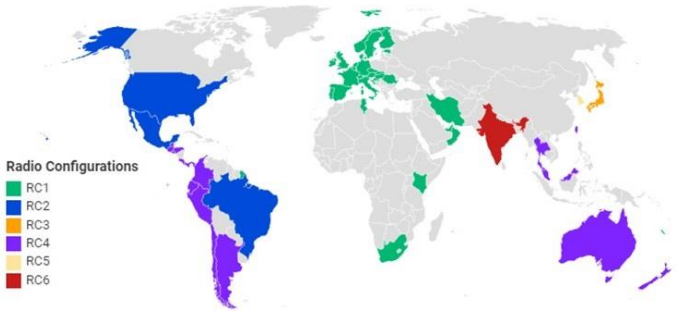


Nota. En la figura se observa los pines que utiliza el Thinxtra Sigfox al conectarse con la placa Arduino, los pines que están señalados con el color rojo están destinados a la alimentación del módulo, los pines que están señalados con color celeste son los utilizados por los sensores integrados en la placa para comunicarse con el Arduino y los pines que están señalados con color verde se utilizan para la comunicación entre los módulos *Thinxtra Sigfox* y Arduino. Tomado de (GitHub, 2022)

Se utiliza el Xkit RCZ4 debido a que la Radio configuración está destinada según los países y regiones donde se encuentra la red SIGFOX como se puede observar en la figura 23.

Figura 23

Cobertura de SIGFOX en el mundo



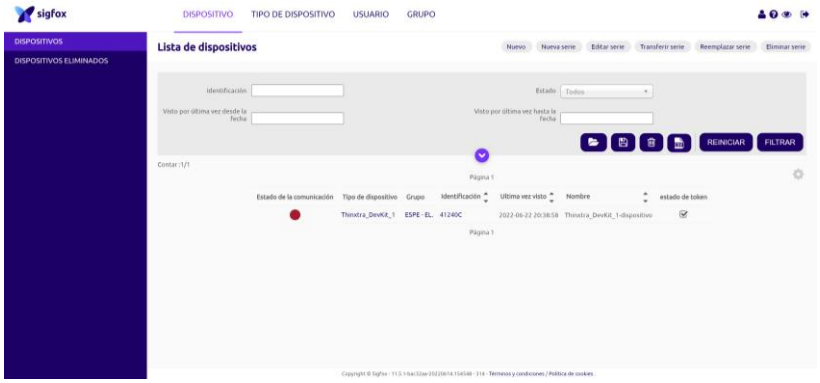
Nota. La figura indica que radio configuración utiliza SIGFOX según las regiones en donde tiene cobertura. Tomado de (AGElectrónica, 2022)

Backend Sigfox.

El backend sirve para poder acceder a la información enviada por las placas SIGFOX utilizando la programación que antes se cargó en el módulo Arduino, por lo cual se deben registrar las placas SIGFOX con su respectivo ID, estos dispositivos registrados envían mensajes en números hexadecimales, de aquí se envía la información a CALLBACKS (Ubidots) donde se puede procesar la información para ser mostrada en el destino final que es la página web o la aplicación celular.

Figura 24

Backend SIGFOX

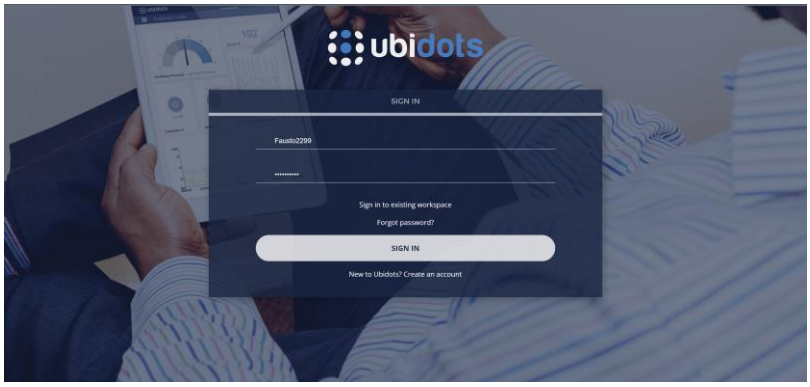


Ubidots

Ubidots es la plataforma del internet de las cosas, donde se visualiza datos de sensores o extraerlos desde el Backen de Sigfox, que puede ser considerado como un Callback, se utiliza para configurar tableros donde se muestra la información en tiempo real que se analiza después.

Figura 25

Ventana de inicio de Ubidots



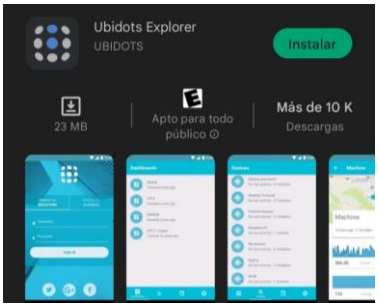
Ubidots explorer.

Ubidots tiene una aplicación móvil que se la puede encontrar en Play Store, donde se visualiza nuestros tableros desde el celular que cuente con conexión a internet, de esta forma

se ve los datos de los sensores aun estando lejos del vehículo, con esto se supervisa de una mejor manera y en tiempo real.

Figura 26

Ubidots explorer



Capítulo III

Análisis para la obtención y envió de datos a través de los módulos Shield Can Bus Y Thinxtra Kit4

Para iniciar con el proceso de obtención de datos se configura el hardware de manera adecuada teniendo en cuenta la disposición de los módulos y los cables para que exista comunicación entre el vehículo y los dispositivos. Para ello es necesario recrear los diagramas donde se detallan la disposición de cada placa y cable.

Caracterización de vehículo.

Para la presente investigación el vehículo seleccionado es un Kia Rio Sedan del año 2019, debido a que cuenta con las características necesarias para el desarrollo de ésta, como lo es el sistema de inyección electrónica multipunto, posee conector tipo OBD2 y se rige bajo la normativa SAE J1979 con los cuales se pueden identificar los OBDII PIDs, aunque algunos fabricantes asignan otros valores para realizar este trabajo.

Figura 27

Conector OBD2 Kia Rio 2019

Este conector OBD2 cuenta con protocolo ISO 15765, debido a que se puede observar el pin 6 (CAN-H) y 14 (CAN-L) que realizan la conexión. Se podrá encontrar dentro del habitáculo del lado del conductor, en la parte izquierda inferior del volante, detrás de la tapa de la fusiblera la que al retirar permite observar en la parte inferior dicho conector.

Características de los módulos utilizados

Tabla 6

Características del Arduino uno.

Parámetro	Valor
Microcontrolador	ATmega328P
Voltaje de operación	5 V
Voltaje de entrada	7 – 12 V
Pines entrada/salida digital	14
Pines de entrada analógica	6
Memoria Flash	32 KB
SRAM	2 KB
EEPROM	1 KB
Frecuencia de reloj	16MHz
Longitud	68,6 mm
Ancho	53,4 mm
Peso	25 g

Nota. Los datos de la tabla se obtuvieron del datasheet del componente.

Tabla 7

Características del Shield CAN-BUS.

Parámetro	Valor
Microcontrolador	MCP2515
Transceptor CAN	MCP2551
Voltaje de operación	4,8~5,2 V
Velocidad SPI	10 MHz
Velocidad CAN-BUS	1 Mb/s
Longitud	74,8 mm
Ancho	53,4 mm
Alto	27,1 mm
Peso	48 g

Nota. Los datos de la tabla se obtuvieron del datasheet del componente.

Tabla 8

Características del Thixtra SIGFOX.

Parámetro	Valor
Banda de frecuencia	RC4
Configuración de radio	920; 1375-922; 6625 MHz
Potencia de salida	24dBm
Cantidad de mensajes enviados	140
Tamaño de mensaje enviados	12 bytes
Voltaje de operación	3,3; 5; 9 V

Nota. Los datos de la tabla se obtuvieron del datasheet del componente.

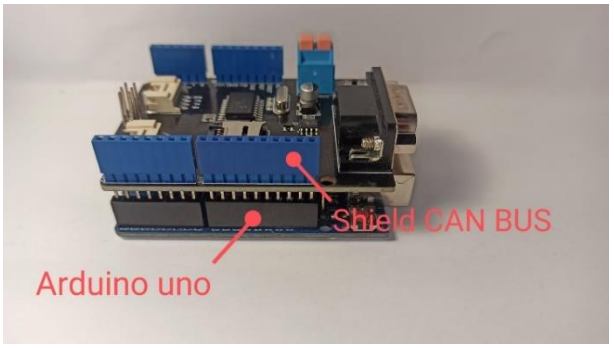
Obtención de datos

Shield can bus.

A fin de obtener los datos que entrega la ECU del automóvil, se precisa usar dos módulos; el módulo Shield Can Bus V2.0, la placa Arduino uno, en los que se realiza la conexión módulo – vehículo, la carga de la programación y recolección de datos respectivamente.

Figura 28

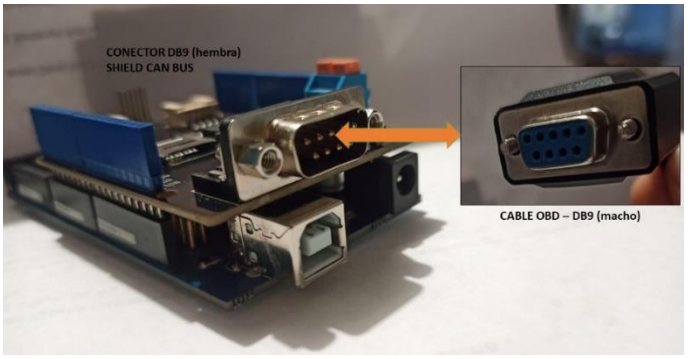
Montaje del módulo Shield Can bus sobre la placa Arduino uno



Como se puede observar en la *figura 28* para la obtención de datos, conviene montar la placa Shield can bus sobre la placa Arduino uno, empatando el Pin 0 con el pin 0-RX respectivamente. Tomar en cuenta que queden todos los pines del módulo CAN estén dentro de su correspondiente entrada.

Figura 29

Conexión del montaje Arduino – Shield Can bus con el cable conector DB9



Una vez se disponga los módulos montados y después de subir el programa que ayuda para la obtención de datos, se continúa con la conexión del cable DB9-OBD a la placa CAN BUS con lo que se debe mantener correctamente alineados el DB9 hembra de la placa con el DB9 macho de cable.

Librerías utilizadas.

Las librerías utilizadas para la programación del módulo Shield CAN BUS son las siguientes:

- SPI.h esta librería se la utiliza en la interfaz de comunicación de cuatro hilos, que ayuda a la comunicación de los microcontroladores y los periféricos que en nuestro caso es el Shield CAN BUS, una de sus características es la alta velocidad de comunicación que se puede obtener.
- mcp2518fd_can.h esta librería se utiliza para poder programar el controlador can MCP2518 que es un microchip de 14 pines que viene integrado en el Shield CAN BUS, su velocidad de comunicación es de 8 Mbps.
- mcp2515_can.h se utiliza esta librería para poder comunicarnos entre la ECU del vehículo y el Shield CAN BUS, de forma que se envía la petición del PID a la ECU con el identificador hexadecimal, se obtiene como respuesta el PID en números hexadecimales que después se los debe convertir en decimales y se lo multiplica por la escala para obtener el valor escalar que indica el sensor.

Conversión de datos.

MAP.

Conversión de hexadecimal a decimal

Para obtener la información del sensor MAP la ECU entrega el bite con dos números en formato hexadecimal por lo que se deben aplicar las siguientes fórmulas para obtener el valor escalar:

Byte entregado = ba

$$MAP_{10} = b * 16^1 + a * 16^0$$

Obtención del valor escalar

$$Valor\ escalar(kPa) = 0 + 1 * MAP_{10}$$

Velocidad del vehículo.

Conversión de hexadecimal a decimal

Para obtener la velocidad del vehículo la ECU entrega el bite con dos números en formato hexadecimal por lo que se deben aplicar las siguientes fórmulas para obtener el valor escalar:

Byte entregado: ba

$$Velo_{10} = b * 16^1 + a * 16\text{^}0$$

Obtención del valor escalar

$$Valor\ escalar\left(\frac{Km}{h}\right) = 0 + 1 * Velo_{10}$$

RPM.

Conversión de hexadecimal a decimal

Para obtener las revoluciones del motor la ECU entrega dos bites con dos números cada bite en formato hexadecimal por lo que se deben aplicar las siguientes fórmulas para obtener el valor escalar:

Bytes entregados: dc ba

$$RPM_{10} = d * 16^3 + c * 16^2 + b * 16^1 + a * 16^0$$

Obtención del valor escalar

$$Valor\ escalar(RPM) = 0 + 0{,}25 * RPM_{10}$$

IAT.

Conversión de hexadecimal a decimal

Para obtener la temperatura del aire de admisión la ECU entrega el bite con dos números en formato hexadecimal por lo que se deben aplicar las siguientes fórmulas para obtener el valor escalar:

Byte entregado: ba

$$IAT_{10} = b * 16^1 + a * 16^0$$

Obtención del valor escalar

$$Valor\ escalar(°C) = -40 + 1 * IAT_{10}$$

TPS.

Conversión de hexadecimal a decimal

Para obtener la temperatura del aire de admisión la ECU entrega el bite con dos números en formato hexadecimal por lo que se deben aplicar las siguientes fórmulas para obtener el valor escalar:

Byte entregado: ba

$$TPS_{10} = b * 16^1 + a * 16\text{^}0$$

Obtención del valor escalar

$$Valor\ escalar(\%) = 0 + \frac{1}{2{,}55} * TPS_{10}$$

ECT.

Conversión de hexadecimal a decimal

Para obtener la temperatura del refrigerante la ECU entrega el bite con dos números en formato hexadecimal por lo que se deben aplicar las siguientes fórmulas para obtener el valor escalar:

Byte entregado: ba

$$ECT_{10} = b * 16^1 + a * 16^0$$

Obtención del valor escalar

$$Valor\ escalar(°C) = -40 + 1 * ECT_{10}$$

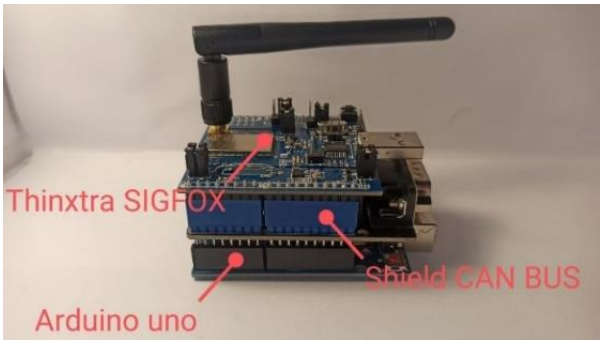
Envió de datos

Xkit Thinxtra Sigfox.

Para el envío de datos se utiliza el módulo Thinxtra Sigfox, este módulo Sigfox va colocado sobre el Shield CAN BUS, los tres módulos se comunican y cumplen funciones específicas que se encuentran definidas en la programación, se recolecta los datos con el módulo Can Bus y se envían mediante la red LPWAN con el módulo Sigfox.

Figura 30

Montaje del módulo Sigfox sobre la placa CAN BUS y el Arduino uno



Como se muestra en la figura 30, los módulos Can Bus y Sigfox utilizan todos los pines del Arduino uno que son alimentados por la conexión del puerto serial DB9, que toma energía del puerto OBD II, de esta forma se energizan los tres módulos.

Librerías utilizadas.

Las librerías utilizadas para la programación del módulo Sigfox son las siguientes:

- Wisol está librería controla el chip del módulo, dentro de su programación vienen definidos los pines con los que se comunica con el Arduino para enviar los datos al backend de Sigfox.
- SimpleTimer sirve para controlar los temporizadores internos de la placa Arduino, con la que se define los tiempos de envió de los mensajes hacia el backend de Sigfox.

- avr/wdt esta librería se utiliza para realizar un reset de la programación cada cierto periodo de tiempo, con el objetivo de que si existe algún fallo el programa no se quede bloqueado.

Disposición del hardware

El cable de obtención de datos debe ir conectado en el puerto OBD II del vehículo, para esto se ubica el conector en el vehículo que se diagnostica, tal como se indica en la figura 27, el cable tiene dos tipos de conectores el que se conecta es el OBD II en el vehículo, como se ve en la figura 31.

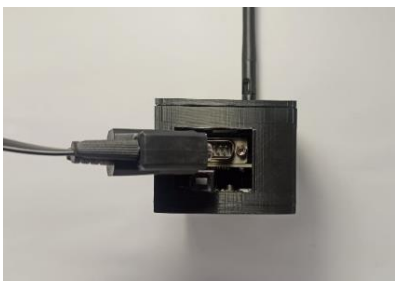
Figura 31

Conexión con el puerto OBD II

En el módulo se conecta el conector serial DB 9, este conector se encuentra en el Shield CAN BUS, se ajusta los tornillos que aseguran el conector para que no se salga al momento que existan vibraciones, como lo indica la figura 32.

Figura 32

Conexión con el puerto DB 9



El hardware se encuentra ubicado en un lugar visible, debido a que el módulo Sigfox no debe tener interferencias para poder enviar los mensajes al backend, por lo que se optó por ubicarlo sobre el tablero de instrumentos, apegado al parabrisas como se ve en la figura 33.

Figura 33

Ubicación del módulo

Programación del módulo Shield CAN BUS V2 y el módulo Xkit Thinxtra mediante Arduino para la obtención y envió de datos hacia el Backen de Sigfox

La programación que se encuentra en la aplicación de Arduino, dentro del código permite obtener los PIDs, se convierte de formato hexadecimal a decimal, se forma el payload de datos y se envía los mensajes al Backend de Sigfox.

En la primera parte de la programación se incluye las librerías que ayudan a controlar los módulos que se están utilizando, para lo cual se define que tipos de variables son, a algunas de ellas se les da valores que son necesarios para cumplir con la recolección y envío de datos, esto se observa en la figura 34.

Figura 34

Incluir librerías y definir variables

```
#include <WISOL.h>
#include <Wire.h>
#include <math.h>
#include <SimpleTimer.h>
#include <avr/wdt.h>
#include <SPI.h>
#define CAN_2515


int tl=0;
const int SPI_CS_PIN = 9;
const int CAN_INT_PIN = 2;

#include "mcp2515_can.h"
mcp2515_can CAN(SPI_CS_PIN);

#define CAN_ID_PID          0x7DF
```

Para controlar el buffer (espacio de memoria temporal), se crea la función `void set_mask_filt()` donde se tiene máscaras y filtros, que ayudan a aceptar o rechazar los mensajes.

Figura 35

Máscaras y filtros

```
void set_mask_filt()
{
  CAN.init_Mask(0, 0, 0x7FC);
  CAN.init_Mask(1, 0, 0x7FC);
  CAN.init_Filt(0, 0, 0x7E8);
  CAN.init_Filt(1, 0, 0x7E8);
  CAN.init_Filt(2, 0, 0x7E8);
  CAN.init_Filt(3, 0, 0x7E8);
  CAN.init_Filt(4, 0, 0x7E8);
  CAN.init_Filt(5, 0, 0x7E8);
}
```

En la figura 36 se ve la función para solicitar los PIDs de la ECU, se crea el vector llamado `tmp[8]` donde se define el ID del PID a solicitar, y con `CAN.sendMsgBuf` se envía la solicitud a la ECU, teniendo en cuenta el ID de PID, y el vector de solicitud.

Figura 36

Solicitud de PIDs

```
void sendPid(unsigned char __pid)
{
  unsigned char tmp[8] = {0x02, 0x01, __pid, 0, 0, 0, 0, 0};
  CAN.sendMsgBuf(CAN_ID_PID, 0, 8, tmp);
}
```

Dentro del `void setup()` se inicia la comunicación I2C utilizando `Wire.begin()`, con esto se van a comunicar los módulos Can Bus y Sigfox con el Arduino, se envían datos byte por byte utilizando 0 y 1, esta comunicación existe solo cuando se la solicita, también se define el tiempo de envío de datos de Sigfox. Véase figura 37.

Figura 37

Void setup

```
void setup()
{

  int flagInit;
  Wire.begin();
  //Wire.setClock(100000);
  Serial.begin(9600);

  //inicializo el watchdog timer
  //watchdogSetup();
  //watchdogCounter = 0;

  // WISOL test
  flagInit = -1;
  while (flagInit == -1)
  {
    Serial.println(""); //reset
    delay(1000);
    PublicModeSF = 0;
    flagInit = Isigfox->initSigfox();
    Isigfox->testComms();
    GetDeviceID();
  }
  //temporizador de envio para el Sigfox cada 3 minutos
  unsigned long tiempo_envio = 3000;//30000;
  timer.setInterval(tiempo_envio,enviar_OBDII);
  Serial.println("");
  delay(1000);
  while (CAN_OK != CAN.begin(CAN_500KBPS))
  {

    delay(100);
  }
```

Dentro del `void loop ()` se tiene la programación que se va a repetir, aquí se recibe la respuesta de la ECU a los PIDs que se envían, están definidos los ID de los PIDs a solicitar

que en este caso son las rpm, velocidad del auto, IAT, TPS, ECT y MAP, este proceso se lo realiza cada décima de segundo con `delay(100).` Véase en la figura 38.

Figura 38

Void loop

```
93 void loop()
94 {
95   timer.run();
96
97   digitalWrite(13,!digitalRead(13));
98
99   delay(100);
100
101  //1 rpm
102  taskCanRecv();
103  pid=0xC;
104 sendPid(pid);
105  delay(100);
106
107
108  //2 velocidad del auto
109  taskCanRecv();
110   pid=0xD;
111  sendPid(pid);
112  delay(100);
113
114
115  //3 IAT
116  taskCanRecv();
117  pid=0x0F;
118  sendPid(pid);
119  delay(100);
120
121  //4 TPS
122  taskCanRecv();
123  pid=0x11;
124  sendPid(pid);
125  delay(100);
126
127
128  //5 TEMPERATURA REFRIGERANTE
129  taskCanRecv();
130  pid=0x05;
131  sendPid(pid);
132  delay(100);
133
134
135  //6 MAP
136  taskCanRecv();
137  pid=0x0B;
138  sendPid(pid);
139  delay(100);
```

En la figura 39 se observa la función `void taskCanRecv()`, esta función recibe la respuesta de la ECU de los PIDs solicitados en formato hexadecimal, los convierte en números decimales y realiza las operaciones para cada sensor como se muestra en el apartado 3.3.3

para obtener la información real de los sensores del vehículo. se guarda estos datos en el arreglo `OBDII[ ]` que va a servir para formar el payload de envió de datos a través de Sigfox.

Figura 39

Recepción de datos de la ECU

```
void taskCanRecv()
{
  unsigned char len = 0;
  unsigned char buf[8];
  if (CAN_MSGAVAIL == CAN.checkReceive())
  {
    CAN.readMsgBuf(&len, buf);    ¿
    if(pid==0x0C) //rpm
    {
      numero=(buf[3]*256)+buf[4];
      rev=numero*0.25;
      word rev1=rev;
      OBDII[0]=highByte(rev1);
      OBDII[1]=lowByte (rev1);
    }
    if(pid==0x0D)//speed
    {
      numero=buf[3];
      vel=numero*1;
      OBDII[2]=vel;
    }
    if(pid==0x0F)//IAT
    {
      numero=buf[3];
      IAT=(numero-40);//restar 40
      OBDII[3]=IAT;
    }
    if(pid==0x11)//TPS
    {
      numero=buf[3];
      TPS=(numero*0.4);
      OBDII[4]=TPS;
    }
    if(pid==0x05)//Temperatura refrigerante
    {
      numero=buf[3];
      ECT=(numero);//restar 40
      OBDII[5]=ECT;
    }
    if(pid==0x0B)//MAP
    {
      numero=buf[3];
      MAP= numero;
      OBDII[6]=MAP;
    }
```

La función `void envio_dato()` envía los datos de nuestro arreglo `OBDII[ ]` donde se encuentran guardados los datos de los sensores del vehículo, se define el formato de los datos a enviar que son `uint8_t` y `uint16_t` , los cuales sirven para la decodificación en el Backend y

envío al Callback Ubidots para poderlos visualizar en la aplicación web al igual que en la aplicación móvil. Véase la figura 40.

Figura 40

Envió de datos al backend de Sigfox.

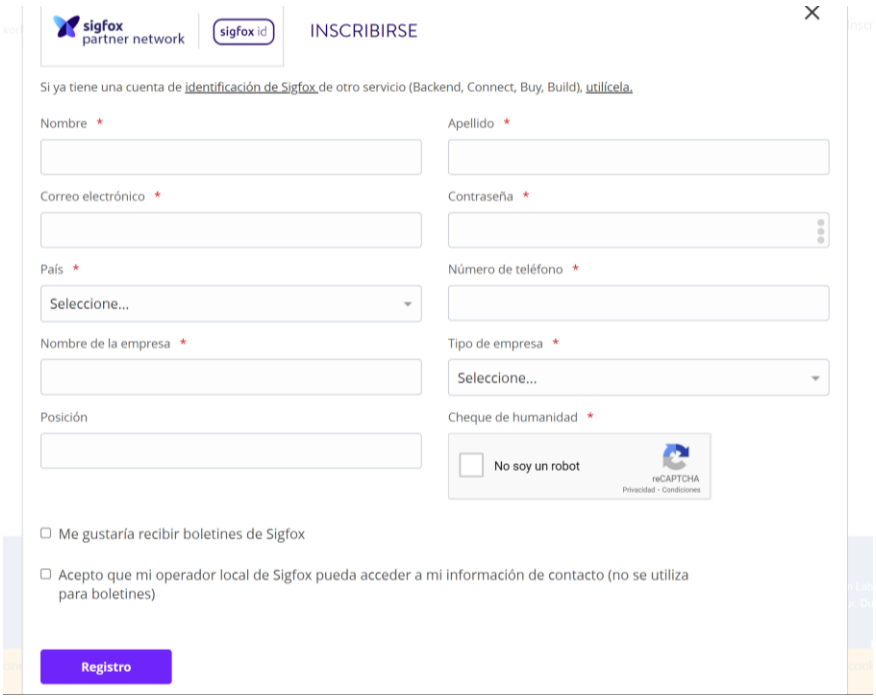
```
215 void envio_dato(uint8_t *sendData, const uint8_t len)
216 {
217   // No downlink message require
218   recvMsg *RecvMsg;
219   RecvMsg = (recvMsg *)malloc(sizeof(recvMsg));
220   Isigfox->sendPayload(sendData, len, 0, RecvMsg);
221   for(int i = 0; i < RecvMsg->len; i++)
222   {
223     Serial.print(RecvMsg->inData[i]);
224   }
225   Serial.println("");
226   free(RecvMsg);
227 }
```

Registro y validación de Xkit SIGFOX

Se ingresa a la página de SIGFOX donde se realiza la compra de los componentes electrónicos, después se realiza el registro, para lo cual se ingresa nuestros datos personales, correo electrónico, número de celular, contraseña, el país donde se utiliza el dispositivo y dar clic en registrarse.

Figura 41

Registro backend de SIGFOX



Una vez registrados se valida nuestro dispositivo, al obtener nuestro kit de desarrollo SIGFOX, dentro del paquete se observa la etiqueta que indica el SIGFOX ID y el PAC los cuales son únicos para cada dispositivo, aquí también se encuentra especificado la banda de frecuencia que es la RCZ4 que corresponde a nuestro país.

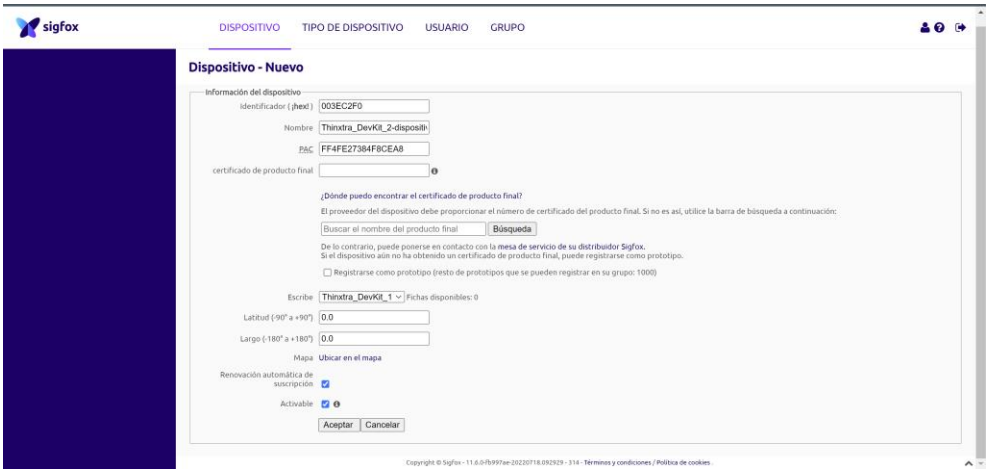
Figura 42

Etiqueta del dispositivo SIGFOX

Para continuar con el registro se ingresa al backend de SIGFOX, ir a la pestaña dispositivos y dar clic en nuevo, ahí se desplegará el formulario de registro que se llena con los datos del dispositivo, al llenar todos los datos que se solicitan, dar clic en aceptar, de esta forma nuestro dispositivo esta registrado en el backend de SIGFOX para recibir los mensajes.

Figura 43

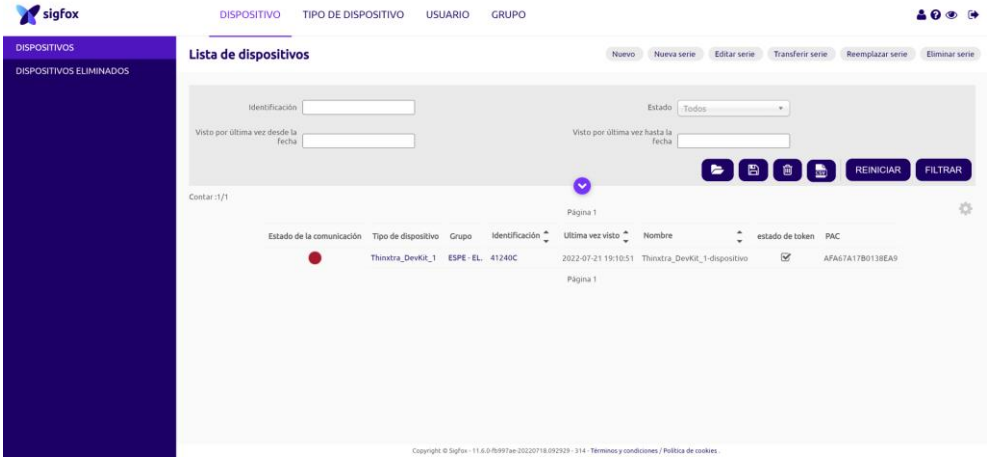
Registro y validación del dispositivo SIGFOX



Realizado estos pasos y concluido el registro, se procede a verificar en la pestaña DISPOSITIVO que nuestro módulo se encuentre listo para funcionar, se visualiza detalles del dispositivo como estado de la comunicación, tipo de dispositivo, grupo, identificación, última vez visto, nombre, estado del token y PAC (Porting Authorization Code).

Figura 44

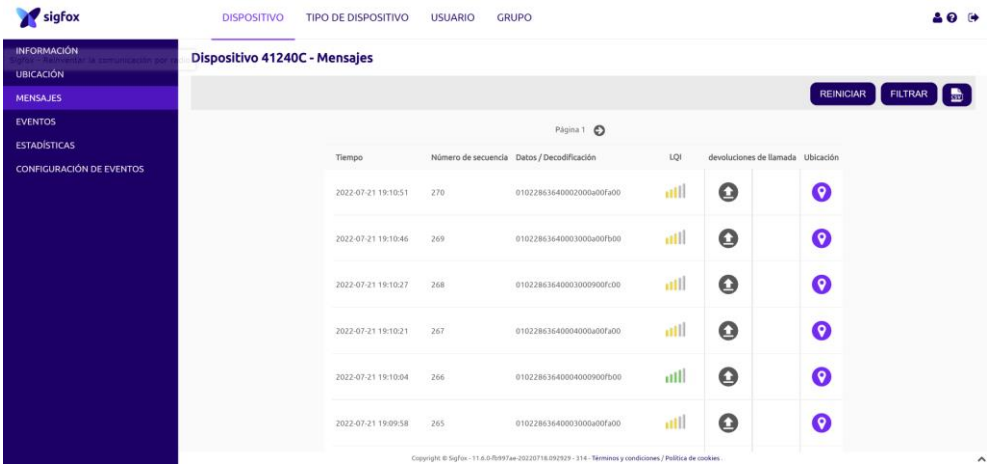
Backend SIGFOX lista de dispositivos



Con nuestro dispositivo funcionando se ve los mensajes que son enviados al backend, estos datos se encuentran codificados en números hexadecimales, se puede ver la lista donde se indica la hora en que se envió el mensaje, el número del mensaje, los datos codificados, el indicador de calidad de enlace, callbacks y la ubicación.

Figura 45

Mensajes enviados por el dispositivo SIGFOX

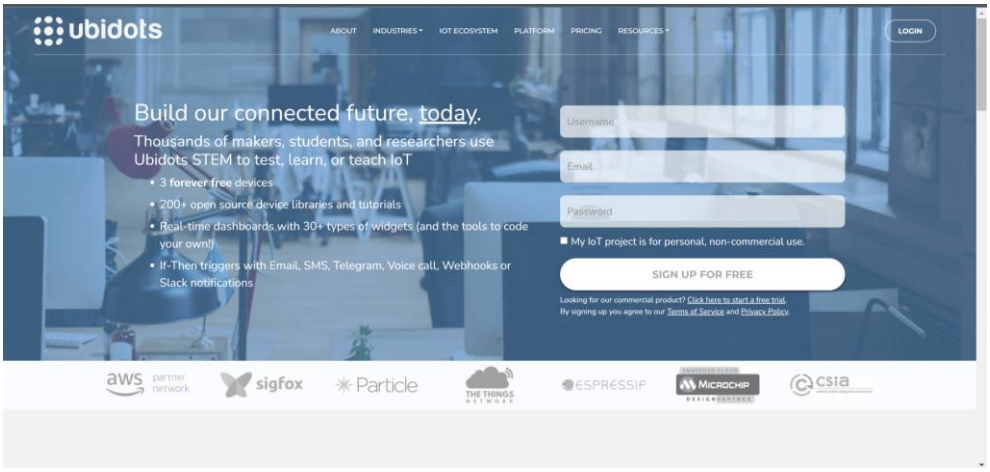


Registro callback (Ubidots)

Para crear la cuenta en Ubidots se ingresa a su página de inicio y dar clic en Sign Up, en la página que emerge para elegir la versión para estudiantes, por último, se llena nuestros datos como usuario, correo electrónico y contraseña, señalando que nuestro proyecto no es para uso comercial, dar clic en crear cuenta gratis.

Figura 46

Formulario de creación de cuenta Ubidots



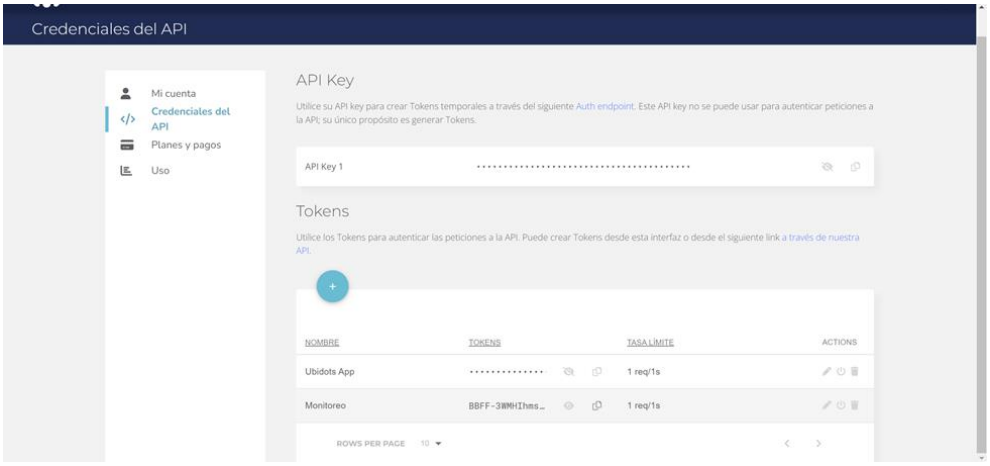
Integración del Backend de SIGFOX y Callback (Ubidots)

Se debe realizar el traspaso de información desde el backend de SIGFOX a el Callback debido a que en el backend de SIGFOX no se puede procesar los datos que se encuentran en formato hexadecimal, para lo cual se utiliza los Callback que en este caso es Ubidots, donde se transforma los datos hexadecimales a decimales, que son el resultado escalar después de multiplicarlo por su escala y restarle la compensación.

Para enviar los datos del Backend a Ubidots se copia el token que se crea que se encuentra en la pestaña de credenciales del API.

Figura 47

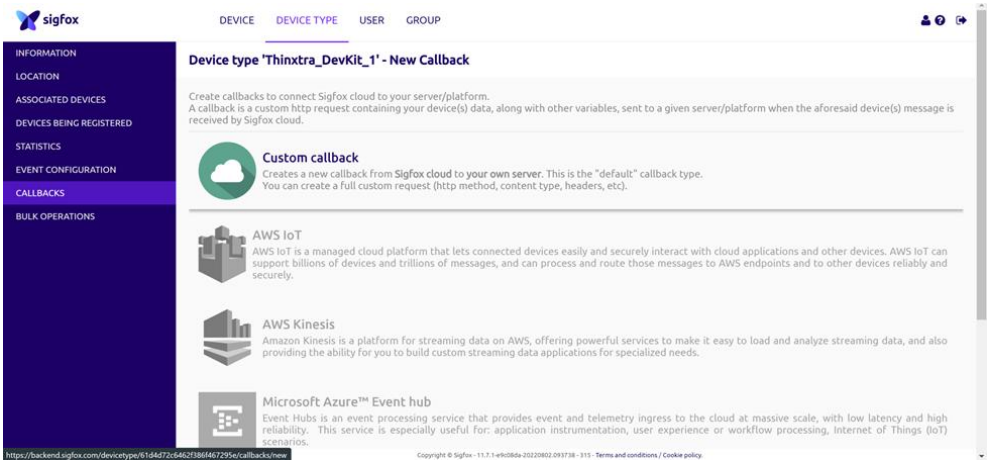
Credenciales API Ubidots



Ingresar al backend de Sigfox, en la pestaña device type, en el apartado Callbacks, se elige el Custom Callback para enlazarlo con Ubidots que va a mostrar los datos en el tablero que lo personaliza después.

Figura 48

Callbacks en el backend de SIGFOX

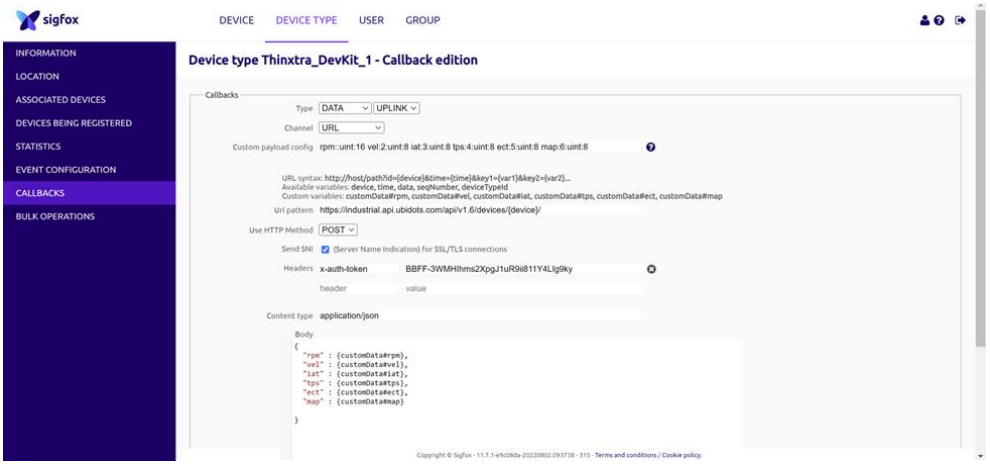


Al dar clic en el Callback seleccionado se despliega el formulario donde se llena los datos para configurar él envió de datos hacia Ubidots donde se llena lo siguiente:

- Tipo de Callback: indica el tipo de solicitud que se tiene, en nuestro caso es Data, también se escoge el tipo de comunicación que es Uplink.
- Custom Payload Config: establecer el formato de las variables que se envía y declarar las variables utilizadas, según los bytes que ocupe la información.
- URL pattern: es el enlace que corresponde al Callback que realiza la solicitud para obtener los mensajes. En Ubidots se encuentra el siguiente enlace https://industrial.api.ubidots.com/api/v1.6/devices/{device}/.
- Headers: aquí se coloca el medio de identificación del Callback, se utiliza autenticación por token y se escribe el token que se obtiene de las credenciales API de Ubidots.
- JSON Body: en este espacio se coloca en formato JSON (Formato de intercambio de datos) las variables que se decodifica en el payload, esta es la información que se va a enviar a la plataforma Ubidots.

Figura 49

Configuración del callback



La configuración que se tiene para el Callback es la siguientes:

- Custom payload config:

  `rpm::uint:16 vel:2:uint:8 iat:3:uint:8 tps:4:uint:8 ect:5:uint:8 map:6:uint:8`

- URL pattern: https://industrial.api.ubidots.com/api/v1.6/devices/{device}/
- Content Type:

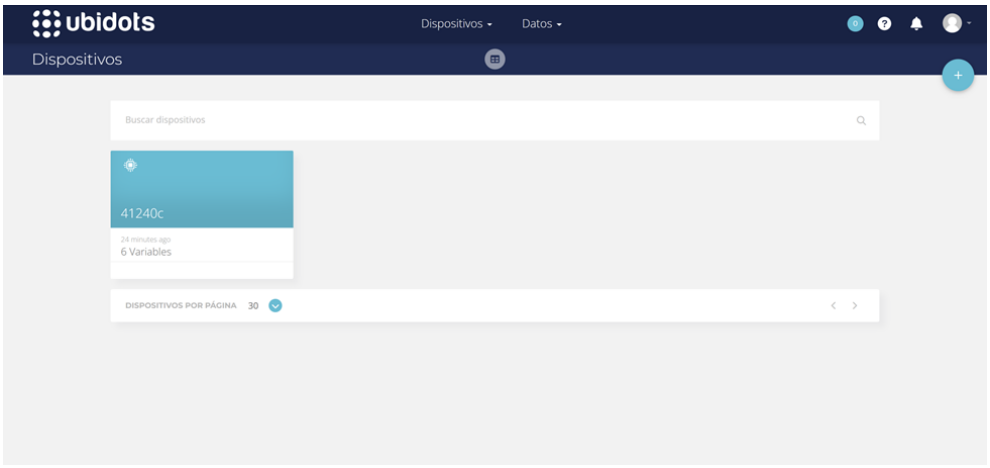
```
{
  "rpm" : {customData#rpm},
  "vel" : {customData#vel},
  "iat" : {customData#iat},
  "tps" : {customData#tps},
  "ect" : {customData#ect},
  "map" : {customData#map}
}
```

Procesamiento de datos

Si se realizó la configuración de una manera adecuada el Callback en el Backen de Sigfox al ingresar a Ubidots, en la pestaña dispositivos se encuentra el dispositivo que esta enlazado, con las variables que se decodifica en el Custom payload.

Figura 50

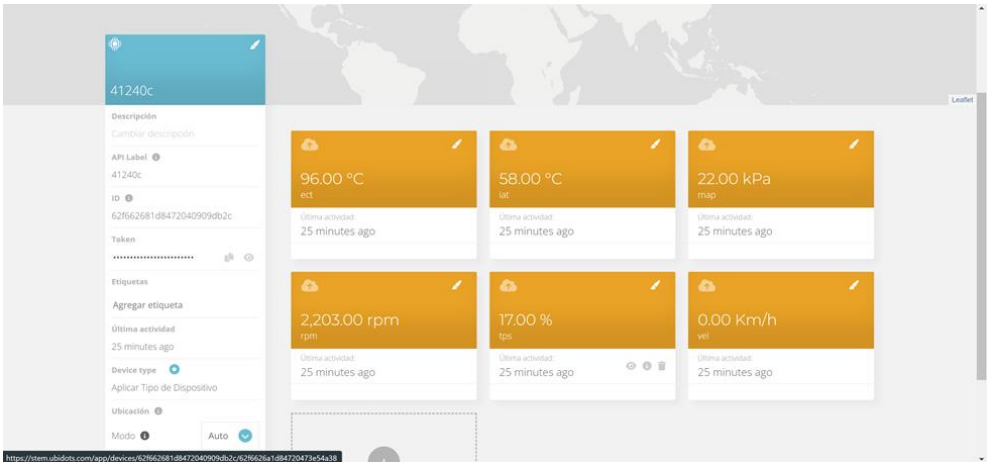
Dispositivos en Ubidots



Dar clic en el dispositivo, de esta forma se despliegan las variables que se envían desde el Backen, se muestra el último dato recibido, también se ve la información del dispositivo.

Figura 51

Datos recibidos en Ubidots

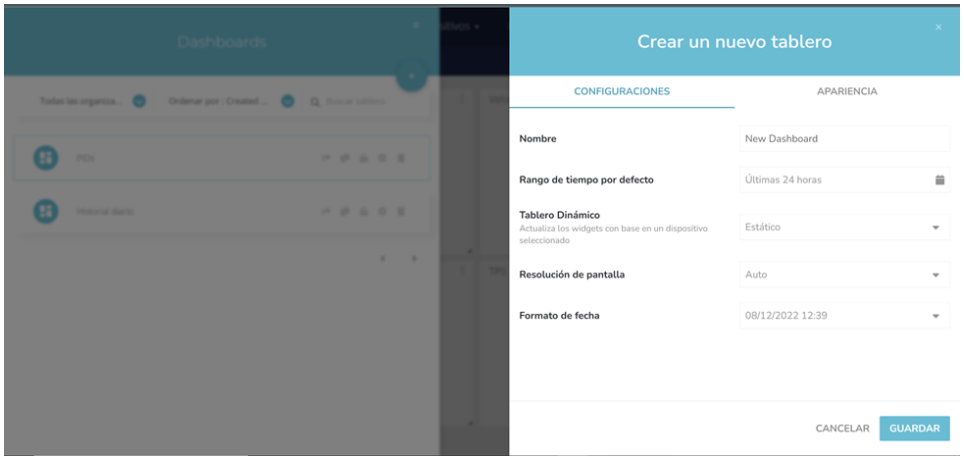


Tableros de información

Para crear los tableros de información se da clic en datos y se selecciona tablero, crear un nuevo tablero, dar un nombre para guardarlo.

Figura 52

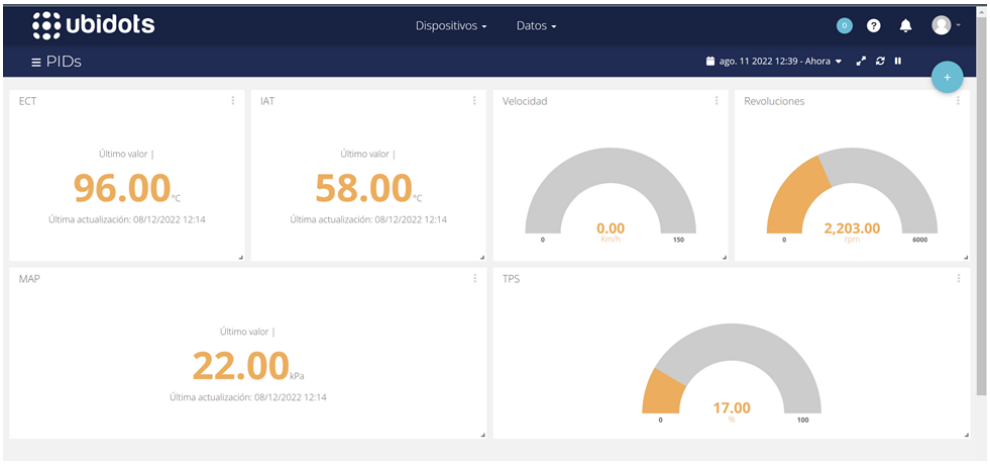
Creación de tablero



Con el tablero creado, configurar las variables a ser mostradas con sus unidades, existe widgets que indican solo el número y otros como reloj estos son los que se utiliza según el dato de los PIDs.

Figura 53

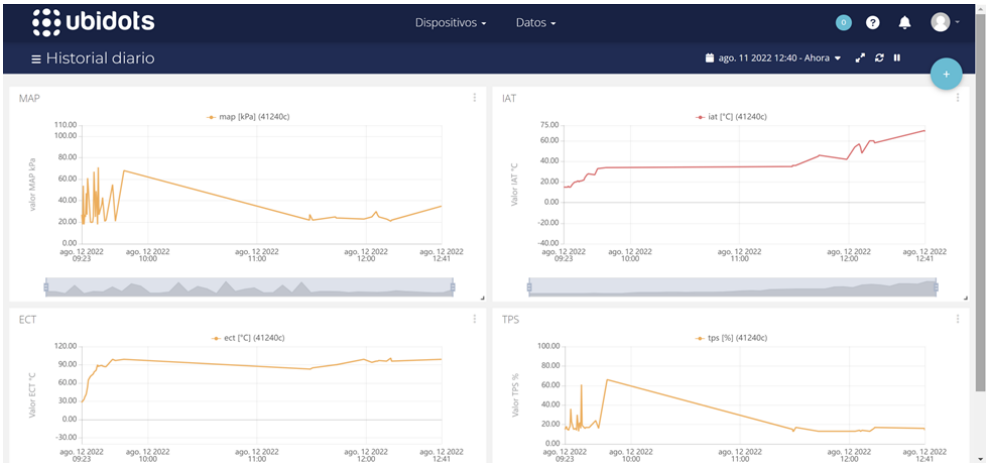
Tablero PIDs



Para el siguiente tablero se siguen los mismos pasos, pero en este se muestra la gráfica donde se indican los valores históricamente de las últimas 24 horas, de forma que se puedan analizar estos datos.

Figura 54

Tablero historial diario



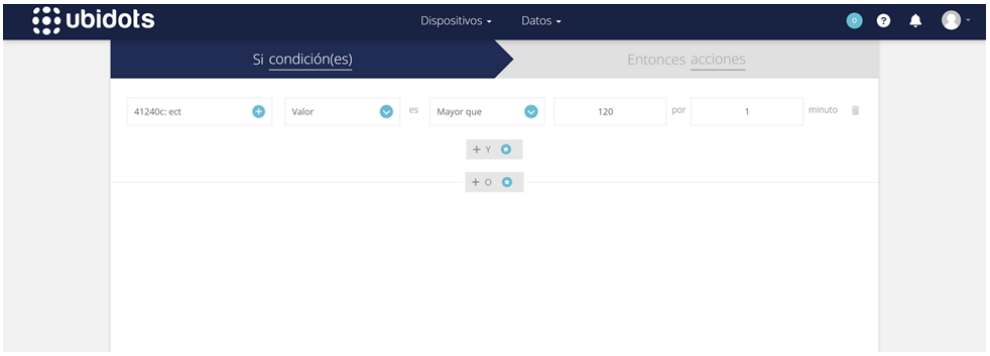
Mensajes de advertencia

En el caso de que algún sensor falle se va a enviar un mensaje de advertencia que indica que está fallando, esto se lo puede realizar directamente desde la plataforma Ubidots,

esta función se encuentra en la pestaña de eventos, se debe crear uno nuevo y en la siguiente pestaña agregar las condiciones en las que se envía el mensaje.

Figura 55

Configuración mensaje de alerta

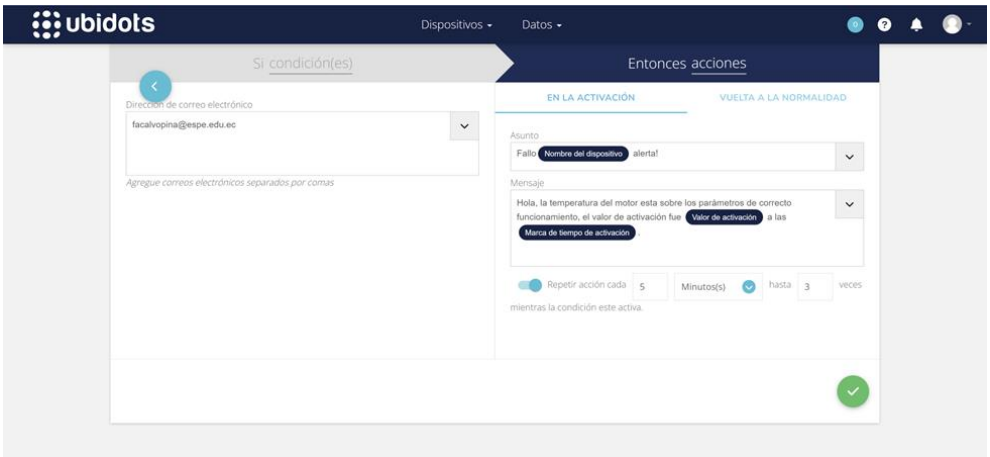


En este caso se configura para monitorear la señal del sensor ECT, en el caso de que pase de 120°C debe enviar el mensaje de alerta.

Consiguientemente se configura las acciones que se realizan si el valor llega a ser mayor a 120°C, colocar el correo electrónico al cual se envía el mensaje, el asunto y se redacta el mensaje, dar clic en el visto verde para guardar.

Figura 56

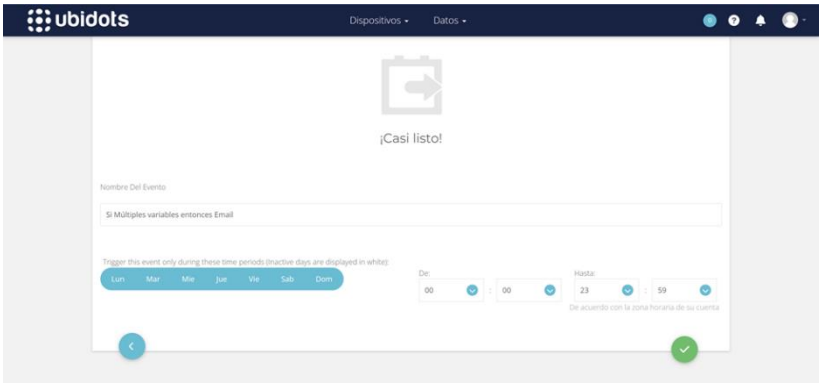
Estructura del mensaje de alerta



Con todo configurado poner el horario en que va a estar activo este evento, pero como no se tiene un horario establecido, se procede a configurarlo para que funciones las 24 horas de los 7 días de la semana, dar clic en el visto verde y nuestro evento se encuentra funcionando con los datos recibidos.

Figura 57

Horario actividad de evento

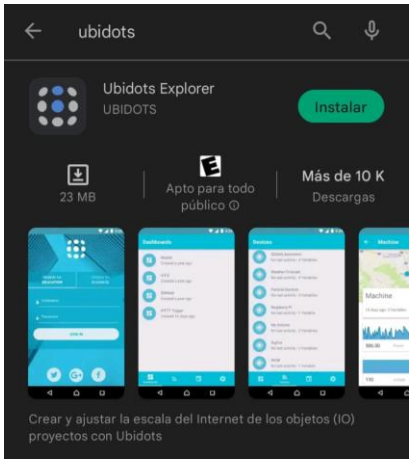


Aplicación móvil Ubidots

La aplicación se la puede obtener en la Play Store, con el nombre de Ubidots Explorer, dar clic en instalar, posterior a esto esperara que se instale en nuestro dispositivo móvil como se indica en la figura 58.

Figura 58

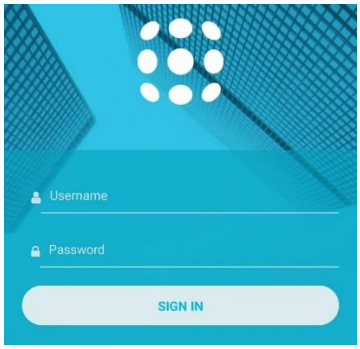
Ubidots Explorer



Con la aplicación instalada, ingresar con el usuario y contraseña con los que se creó la cuenta, dar clic en SIGN IN de forma que se inicia la sesión, se puede observar la información en los tableros antes creados desde nuestro dispositivo móvil sin tener que ingresar a la computadora.

Figura 59

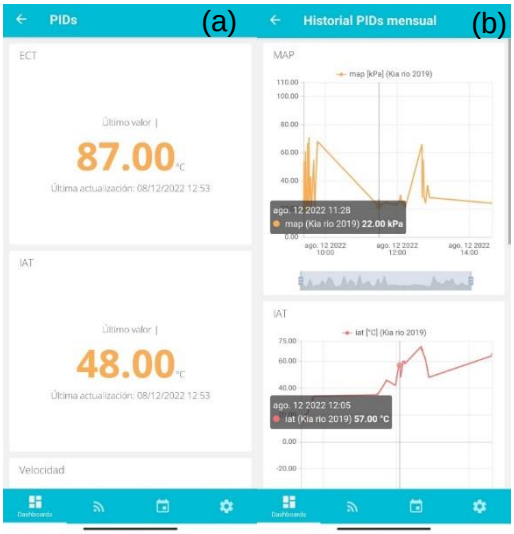
Ingreso a la aplicación



En la pestaña dashboard se elige cualquier tablero, para mostrar la última información recibida, haciendo la monitorización del vehículo más fácil, en la pestaña se indica el último valor de los sensores como se ve en la figura 60(a) y en la otra se observa la gráfica con todos los valores recibidos durante un mes como se ve en la figura 60(b).

Figura 60

Tableros de información Ubidots Explorer



Plan de mantenimiento

Para crear el plan de mantenimiento se define los elementos que se tiene en cuenta como son:

- Aceite de motor
- Filtro de aceite
- Aceite de la transmisión
- Bujías
- Filtro de aire
- Refrigerante de motor
- Pastillas de freno
- Forros de freno
- Liquido de freno y embrague
- Neumáticos
- Batería

De los elementos se obtendrá la tabla individual donde se indican sus características y cada cuanto deben ser revisados o sustituidos, así se obtiene el plan de mantenimiento del vehículo Kia Rio 2019.

Tabla 9

Características aceite de motor

Características	Valor
Viscosidad	SAE 5W-30
Volumen utilizado	3.6 L
API	SN
Vida útil	10.000 Km

Tabla 10

Características del filtro de aceite

Características	Valor
Tipo de filtro	Filtro enroscable
Media de rosca	M20 x 1.5
Presión válvula de desviación	1 bar
Vida útil	10.000 Km

Tabla 11

Características de aceite de transmisión

Características	Valor
Viscosidad	SAE 70W
Volumen utilizado	1.7 L
API	GL-4
Vida útil	60.000 Km

Tabla 12

Características de bujías

Características	Valor
Tipo	Electrodo de masa
Medida de rosca	M12 x 1.25
Piezas necesarias	4
Vida útil	75.000 Km

Tabla 13

Características del filtro de aire

Características	Valor
Tipo	Filtro de recirculación aire
Forma	Pentagonal
Vida útil	30.000 Km

Tabla 14

Características del refrigerante de motor

Características	Valor
Especificación	G12
Rango de temperatura	-35 – 145 °C
Volumen utilizado	5.5 L
Vida útil	30.000 Km

Tabla 15

Características de las pastillas de freno

Características	Valor
Tipo de freno	Disco
Ubicación	Eje delantero
Piezas necesarias	2 pares
Vida útil	60.000 Km

Tabla 16

Características de los forros de freno

Características	Valor
Tipo de freno	Tambor
Ubicación	Eje trasero
Piezas necesarias	2 pares
Vida útil	80.000 Km

Tabla 17

Características del líquido de freno y embrague

Características	Valor
Especificación	DOT 3 o DOT 4
Volumen utilizado	0.7 - 0.8 L
Punto de ebullición seco	325 °C
Vida útil	30.000 Km

Tabla 18

Características de los neumáticos

Características	Valor
Rin	15
Presión de inflado	2.3 bar
Piezas necesarias	4
Inspección	Cada 5.000 km
Vida útil	Según el uso

Tabla 19

Características de la batería

Características	Valor
Voltaje	12 V
Capacidad de la batería	52 Ah
Corriente arranque en frio	470 A
Tipo	Plomo ácido
Inspección	Cada 4 meses
Vida útil	4 años

Con las características mostradas en las tablas anteriores, se procede a realizar el plan de mantenimiento teniendo en cuenta los elementos antes mencionados, en los cuales se realiza inspecciones o reemplazos según corresponda en función del kilometraje o tiempo.

Tabla 20

Plan de mantenimiento Kia Rio 2019

Elemento	Inspección	Reemplazo
Aceite de motor	-	10.000 Km
Filtro de aceite	-	10.000 Km
Aceite de transmisión	-	60.000 Km
Bujías	25.000 Km	75.000 Km
Filtro de aire	-	30.000 Km
Refrigerante de motor	10.000 Km	30.000 Km
Pastillas de freno	20.000 Km	60.000 Km
Forros de freno	20.000 Km	80.000 Km

Elemento	Inspección	Reemplazo
Líquido de freno y embrague		30.000 Km
Neumáticos	5.000 Km	Según el uso
Batería	4 meses	4 años

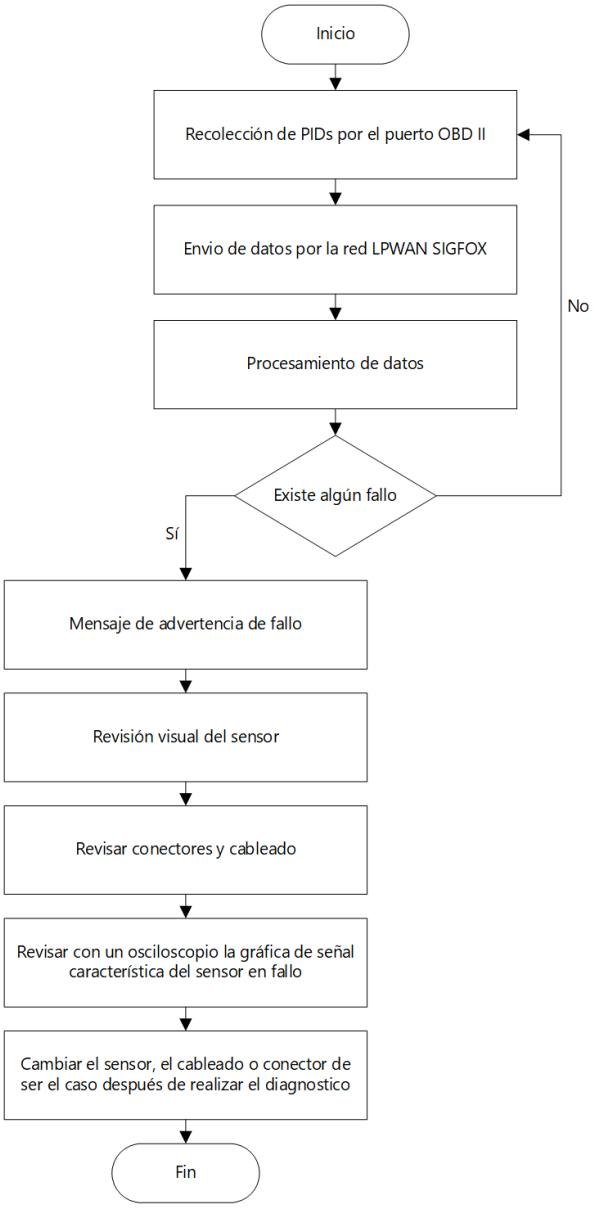
Plan de reparación

El plan de reparación se centra en el sistema de inyección electrónico en específico los sensores, este plan indica en el diagrama de flujo donde se ve los pasos a seguir cuando se amerite una reparación dentro del sistema.

Este plan de reparación va a ser utilizado cuando el módulo envía el mensaje de advertencia al correo electrónico, los sensores van a estar monitorizados en tiempo real y sus datos van a ser guardados en la gráfica para poderlos analizar, en el diagnóstico se utiliza herramientas de apoyo como osciloscopio y scanner, de forma que la avería pueda ser localizada, para la reparación se sigue los pasos que se detalla en la figura 61.

Figura 61

Diagrama de flujo del plan de reparación



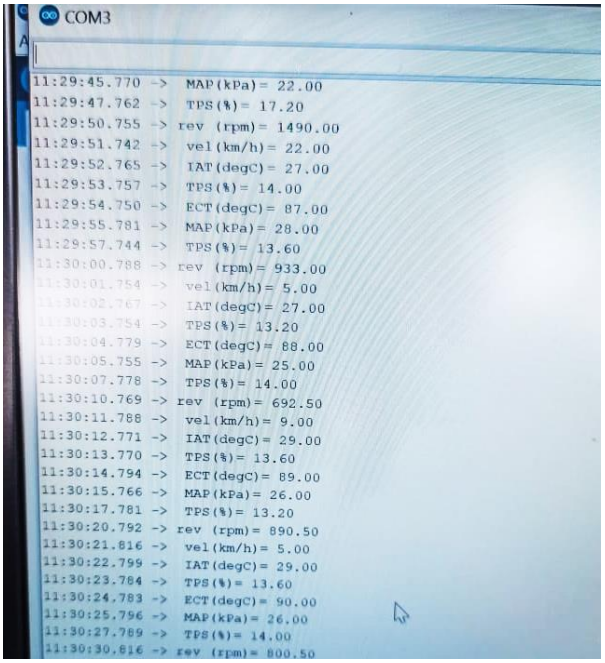
Prueba de envío de datos

Prueba de envío de datos OBD2 – Shield Can bus V2.0.

Para comprobación de envío y recepción de datos entre los parámetros que registra la ECU y el módulo Shield Can Bus V2.0 se realiza a través de la herramienta monitor serie de la aplicación de Arduino, el cual proporciona la entrada y salida de datos en forma de mensajes de texto, donde se puede visualizar la recepción de estos. Como se observa en la figura 62

Figura 62

Pantalla de Monitor Serie



Los datos corresponden al intervalo de tiempo 11:29:45 – 11:30:30, habiendo recolectado por 45 segundos datos entregados por la ECU de los parámetros antes mencionados.

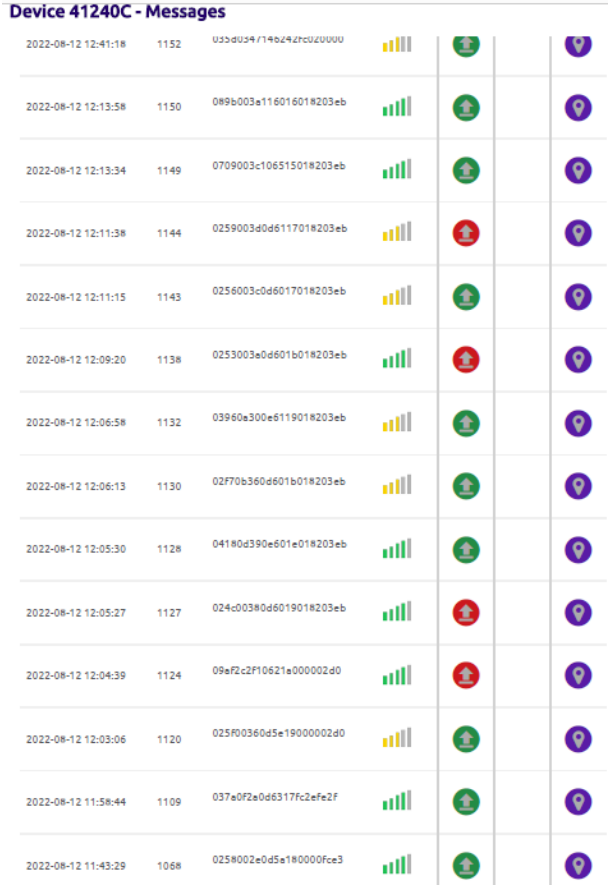
Prueba de envió de datos Placa thinxtra developed kit al Backend Sigfox

Con el objetivo de comprobar que los datos recolectados por la placa Shield Can Bus V2.0 sean transmitidos por el módulo Thinxtra hacia la nube, se utiliza la herramienta web que

proporciona dicho módulo pudiendo verificar que el envío de datos fue exitoso, cuando estos lleguen al backend como se observa en la figura 63 en el intervalo de prueba 11:43:29 hasta 12:13:58 siendo un tiempo de 30 minutos y 29 segundos.

Figura 63

Mensajes Obtenidos del Backend de Sigfox



Estos datos se representan en la siguiente tabla transformando del valor hexadecimal al valor real con las fórmulas antes mencionadas.

Tabla 21

Datos hexadecimales recibidos por el backend

Fecha y hora	Datos
12/8/2022 12:13	089b003a116016018203eb
12/8/2022 12:13	0709003c106515018203eb
12/8/2022 12:11	0259003d0d6117018203eb
12/8/2022 12:11	0256003c0d6017018203eb
12/8/2022 12:09	0253003a0d601b018203eb
12/8/2022 12:06	03960a300e6119018203eb
12/8/2022 12:06	02f70b360d601b018203eb
12/8/2022 12:05	04180d390e601e018203eb
12/8/2022 12:05	024c00380d6019018203eb
12/8/2022 12:04	09af2c2f10621a000002d0
12/8/2022 12:03	025f00360d5e19000002d0

Nota. Los datos presentados en la tabla son los mensajes que llegan al backend antes de ser transformados a valores reales en el dashboard.

Durante este lapso se mantuvo el vehículo en ralentí aproximadamente a 2000 rpms para comparar los datos con los que se van a obtener a continuación con el escáner automotriz. Se puede observar los datos convertidos en valores reales en el ANEXO 1.

Valores de referencia Con escáner automotriz.

Para asegurar que los datos obtenidos mediante los módulos antes mencionados son los valores reales, correctos y recopilados por la ECU se realizó la comparación con un equipo

de escáner automotriz tomando en cuenta dos consideraciones debido a que no se puede conectar ambos equipos al mismo tiempo porque usan el conector DLC para la recopilación de datos, entonces se procede a tomar los datos en dos condiciones específicas que son en ralentí y con el régimen de motor a 2000 rpm a un tiempo de 30 segundos.

En el primer reporte realizado en ralentí por el escáner automotriz se obtienen los datos que se pueden observar en la tabla 22 extraídos del ANEXO 2.

Tabla 22

Valores Obtenidos del Escáner en ralentí

Nombre	Valor	Unidad
Adaptación masa de aire	-3,16	%
Ángulo del sensor de posición del acelerador	0,85	V
Ángulo de bordes del árbol de levas de admisión en relación con el cigüeñal	630,40	Grados
Apertura de la mariposa	3,53	%
Presión del colector de admisión (MAP)	240,78	hPA
Temperatura de admisión de aire	71,25	Grados centígrados
Temperatura del agua	95,25	Grados centígrados
Velocidad del motor	587	rpm

En la siguiente prueba realizada con el escáner automotriz se mantuvo el motor a un régimen de giro de 2000 rpm aproximadamente y se obtiene el reporte ANEXO 3 de donde se extraen los datos de la tabla 23.

Tabla 23

Valores Obtenidos del Escáner a aproximadamente 2.000 rpm.

Nombre	Valor	Unidad
Adaptación masa de aire	-3,16	%
Ángulo del sensor de posición del acelerador	0,85	V
Ángulo de bordes del árbol de levas de admisión en relación con el cigüeñal	271	Grados
Apertura de la mariposa	8,24	%
Presión del colector de admisión (MAP)	228,2	hPA
Temperatura de admisión de aire	68,25	Grados centígrados
Temperatura del agua	99	Grados centígrados
Velocidad del motor	2.182	rpm

Caja contenedora para los módulos

Para proteger los módulos de agentes externos se realiza el diseño de la caja contenedora impresa en 3D, esta caja tiene dos partes, una de ellas es el contenedor de los módulos y la otra parte es la tapa.

El contenedor de los módulos tiene las siguientes características que se va a indicar en la tabla 24 y el diseño con sus dimensiones se ve en la figura 64.

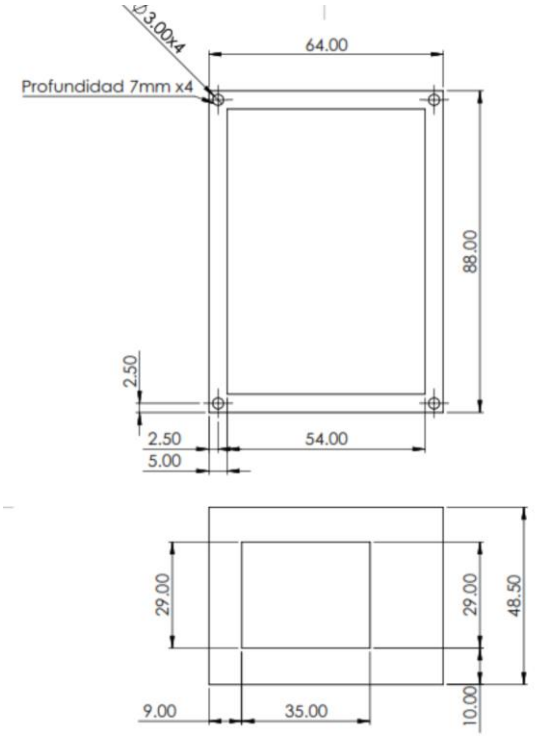
Tabla 24

Características de la caja contenedora

Nombre	Valor
Alto	48,5 mm
Ancho	54 mm
Largo	88 mm
Espesor	5 mm
Material	PLA

Figura 64

Dimensiones de la caja contenedora



La tapa de la caja tiene las siguientes características que se indican en la tabla 25 y sus dimensiones se pueden ver en la figura 65.

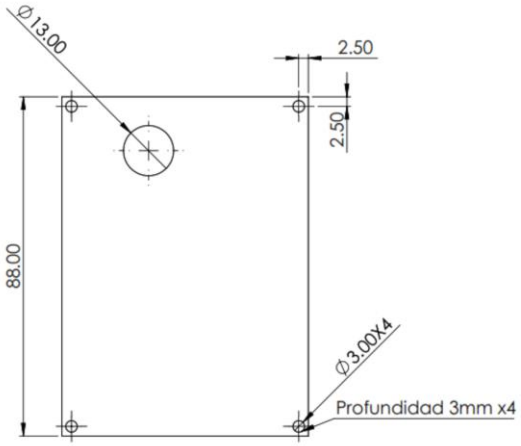
Tabla 25

Características de la tapa

Nombre	Valor
Ancho	54 mm
Largo	88 mm
Espesor	3 mm
Material	PLA

Figura 65

Dimensiones de la tapa

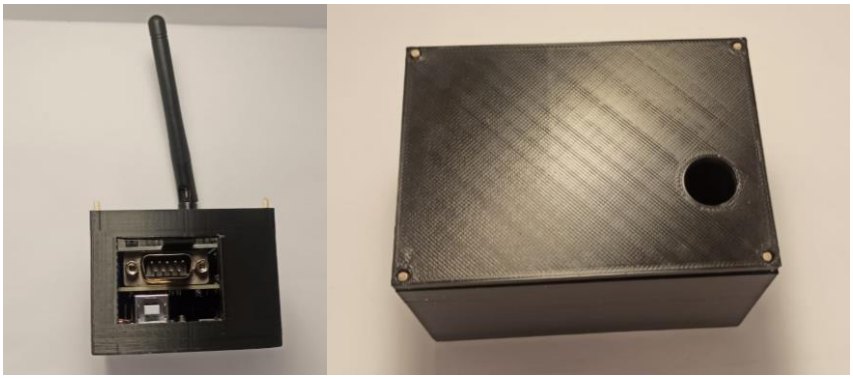


Ubicación de los módulos dentro de la caja

Los módulos van dentro de la caja como se ve en la figura 30, el orificio en la tapa es para que la antena del módulo Sigfox esté por fuera, en la caja contenedora hay un corte de forma que se pueda conectar el cable de datos del Arduino y el conector serial de Shield Can Bus.

Figura 66

Ubicación de los módulos dentro de la caja



Capítulo IV

Análisis de resultados

Prueba de Ruta

Para la recopilación de todos los parámetros entregados por el vehículo hacia la nube se realiza la prueba de ruta que contemple diferentes condiciones de manejo, para lo cual se establece la siguiente ruta como muestra la figura 67. Siendo este intervalo de 14 minutos con 1 segundo empezando a las 11:44 am el periodo de captura de datos que se almacenan en el dashboard y que posteriormente serán analizados.

Figura 67

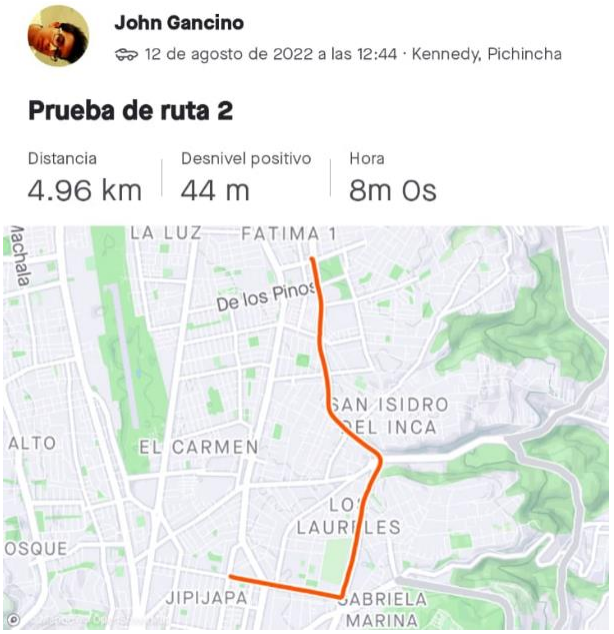
Prueba de ruta 1

Nota. La figura muestra el recorrido realizado desde el sector colinas del norte hasta la Hacienda Victoria.

Con el objetivo de recopilar más datos dentro de la zona urbana se efectúa la segunda prueba de ruta iniciando a las 12:44 y andando un tiempo total de 8 minutos

Figura 68

Prueba de ruta 2



Nota. La figura muestra el recorrido realizado desde el sector de la Hacienda Victoria hasta el sector de la avenida Jipijapa.

Datos obtenidos

La captura de datos se realiza en el intervalo de tiempo total de 22 minutos con 1 segundo, obteniendo un número de 12 mensajes debido a la configuración de la frecuencia de envío de estos a 1 mensaje cada 30 segundos. A continuación, se observa parte de los datos obtenidos a partir de las pruebas anteriores.

Los mensajes recopilados en la prueba de ruta 1 fueron en total 3 como se indica en la tabla 26 la que indica los valores que entrega.

Tabla 26

Valores Prueba de ruta 1

	RPM	VEL (km/h)	IAT (°C)	TPS (%)	ECT(°C)	MAP (kPa)
11:43	**599**	**0**	**45**	**13**	**99**	**23**
11:43	**600**	**0**	**46**	**13**	**90**	**24**
11:58	**890**	**15**	**42**	**13**	**90**	**25**

Con el fin de obtener la mayor cantidad de datos se realiza una segunda prueba donde se obtienen los datos que se pueden observar en la tabla 27.

Tabla 27

Valores Prueba de ruta 2

	RPM	**VEL (km/h)**	**IAT(°C)**	**TPS (%)**	**ECT(°C)**	**MAP (kPa)**
12:44	2.668	32	65	8	96	28
12:45	594	0	66	5	95	25
12:46	1.032	18	64	6	96	24
12:46	2.339	55	63	7	96	24
12:47	878	22	63	5	92	24
12:47	598	0	65	5	93	25
12:50	987	12	56	6	87	37
12:53	3.123	55	48	8	87	28
13:00	619	0	56	5	95	26

Comparación de datos obtenidos con valores ideales

Para poder determinar si los sensores están trabajando correctamente dentro de los parámetros establecidos en la tabla 4, se realiza la comparación entre los valores obtenidos de la prueba de ruta y dicha tabla, dando como resultado la diferencia que se puede observar en las siguientes tablas.

Tabla 28

Valores de la velocidad del motor

Velocidad del motor (rpm)	Rango de medición
599	0 – 16.384 rpm
600	
890	
2.668	
594	
1.032	
2.339	
878	
598	
987	
3.123	
619	

Nota. Como se puede observar, los valores obtenidos utilizando el hardware empleado, se establece que estos están dentro del rango de medición.

Tabla 29

Valores de la velocidad del vehículo

Velocidad (km/h)	Rango de medición
0	0 – 255 km/h
0	
15	
32	
0	
18	
55	
22	
0	
12	
55	
0	

Nota. Los valores presentados en esta tabla representan la velocidad alcanzada en el vehículo durante las pruebas, recopiladas desde el dashboard y dentro de los rangos de medición normales.

Tabla 30

Valores del sensor Intake Air Temperature.

IAT (°C)	Rango de medición
45	-40 a 215 °C
46	
42	
65	
66	
64	
63	
63	
65	
56	
48	
56	

Nota. El contenido de la tabla representa a la temperatura de aire de admisión por lo que se puede observar el registro que realiza el dashboard que está dentro de los rangos previstos.

Tabla 31

Valores del Throttle position sensor.

TPS	Rango de medición
5	0-100%
5	
5	
8	
5	
6	
7	
5	
5	
6	
8	
5	

Nota. Como se observa en la tabla presentada, el rango de medición y los valores obtenidos están acorde al funcionamiento del sensor. Además, se ve registrado la variación de estos al abrirse o cerrarse la mariposa de aceleración.

Tabla 32

Valores del Engine Coolant Temperature.

ECT (°C)	Rango de medición
99	-40 a 255 (°C)
90	
90	
96	
95	
96	
96	
92	
93	
87	
87	
95	

Nota. La temperatura del refrigerante del motor trabaja normalmente en el rango de 90 a 100 grados centígrados por lo que se puede deducir que los valores obtenidos se encuentran dentro del rango de medición y además el buen funcionamiento del sistema de refrigeración del motor.

Tabla 33

Valores del Mass Air Presure.

MAP (kPa)	Rango de medición
23	0 – 255 (kPa)
24	
25	
28	
25	
24	
24	
24	
25	
37	
28	
26	

Nota. Los valores registrados en esta tabla concuerdan con el rango permitido, por lo que el registro realizado en las pruebas es correcto.

Pruebas en otros vehículos

El proyecto se probó en otros vehículos como el Volkswagen T-cross, se utiliza el programa de obtención de PIDs, en el cual no se obtuvo respuesta de la ECU a la solicitud de PIDs que envía el módulo.

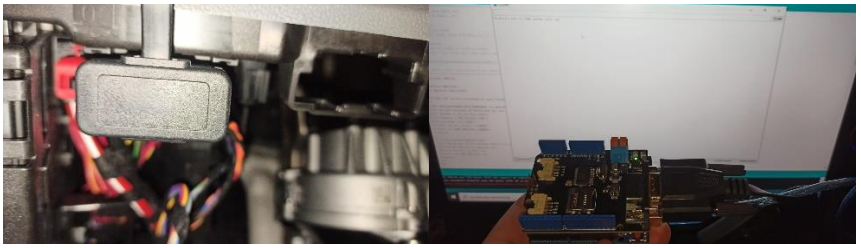
Figura 69

Volkswagen T-cross

Conectar el módulo a el puerto OBD II, para ver si arroja datos, para lo cual se abre el monitor serial en el programa de Arduino.

Figura 70

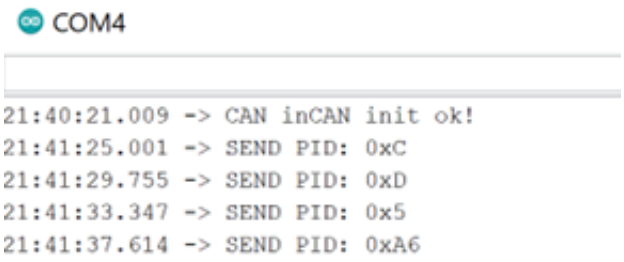
Conexión módulo y puerto OBD II



En el monitor serial se observa que la comunicación Can Bus se inicia, se ingresa los ID de los PIDs que se necesitan, a lo cual no se obtiene respuesta por parte de la ECU. Véase figura 71.

Figura 71

Monitor serial prueba T-cross



Con esta prueba se ve que el módulo no funciona para todos los vehículos debido a que los ID de la red Can Bus no se encuentran normados, al igual que los ID de los PIDs, las marcas designan diferentes identificadores.

Capítulo V

Marco administrativo

Para el proyecto de investigación se tuvieron en cuenta para su desarrollo y ejecución algunos parámetros como los administrativos, técnicos, financiamiento, de forma que se obtiene su factibilidad y el desarrollo en la etapa de investigación.

Factibilidad del proyecto

Los recursos que se utilizan en el proyecto son los tecnológicos, humanos, financieros que fueron considerados dentro de la investigación para delimitar la viabilidad, con el fin de que en la ejecución no existan percances, a su vez tomar en cuenta que el proyecto solo puede ser utilizado dentro de la zona de cobertura de la red Sigfox.

Recursos humanos

En el desarrollo del proyecto los recursos humanos son importantes ya que cada uno de ellos aporta con ideas y conocimiento que se utilizan en la investigación para cumplir con los objetivos propuestos.

Tabla 34

Recursos Humanos

Orden	Descripción	Cantidad	Función
1.	Fausto Calvopiña	1	Investigador
2.	John Gancino	1	Investigador
3.	Ing. Germán Erazo	1	Colaborador Científico

Recursos materiales

Los materiales indicados fueron utilizados en el desarrollo y ejecución del proyecto.

Tabla 35

Recursos Materiales

Orden	Cantidad	Descripción
1.	30 gal	Combustible
2.	2	Alimentación Movilización
3.	1.000	Copias e impresiones
4.	1	Disco duro (1TB)
5.	2	Vehículos de prueba

Recursos tecnológicos

En el desarrollo del proyecto se utiliza los siguientes recursos tecnológicos que se encuentran detallados a continuación.

Tabla 36

Recursos Tecnológicos

Orden	Detalle	Cantidad
1.	Arduino uno	2
2.	Módulo SIGFOX	2
3.	Shield Can Bus	2
4.	Cable OBDII	2
5.	Internet	1
6.	Computadora portátil	2

Presupuesto

El presupuesto estimado por los investigadores para la elaboración del proyecto es de 700 $.

Recursos financieros

Detalle de cada recurso económico y financiero que se utiliza en la elaboración del proyecto.

Tabla 37

Materiales utilizados en el desarrollo del proyecto

Orden	Detalle	Cantidad	Valor unitario USD	Valor total USD
1.	Arduino uno	2	27	54
2.	Módulo SIGFOX	2	36,25	72,50
3.	Shield Can Bus	2	34	68
4.	Cable OBDII	2	12	24
	Total			218,5

Tabla 38

Costo de pruebas realizadas

Orden	Detalle	Cantidad	Valor unitario USD	Valor total USD
1.	Combustible	20 gal	2,40	48
	Total			48

Tabla 39

Gastos imprevistos

Orden	Detalle	Cantidad	Valor unitario USD	Valor total USD
1.	Gastos extras	1	250	250
	Total			250

Tabla 40

Costo total del proyecto

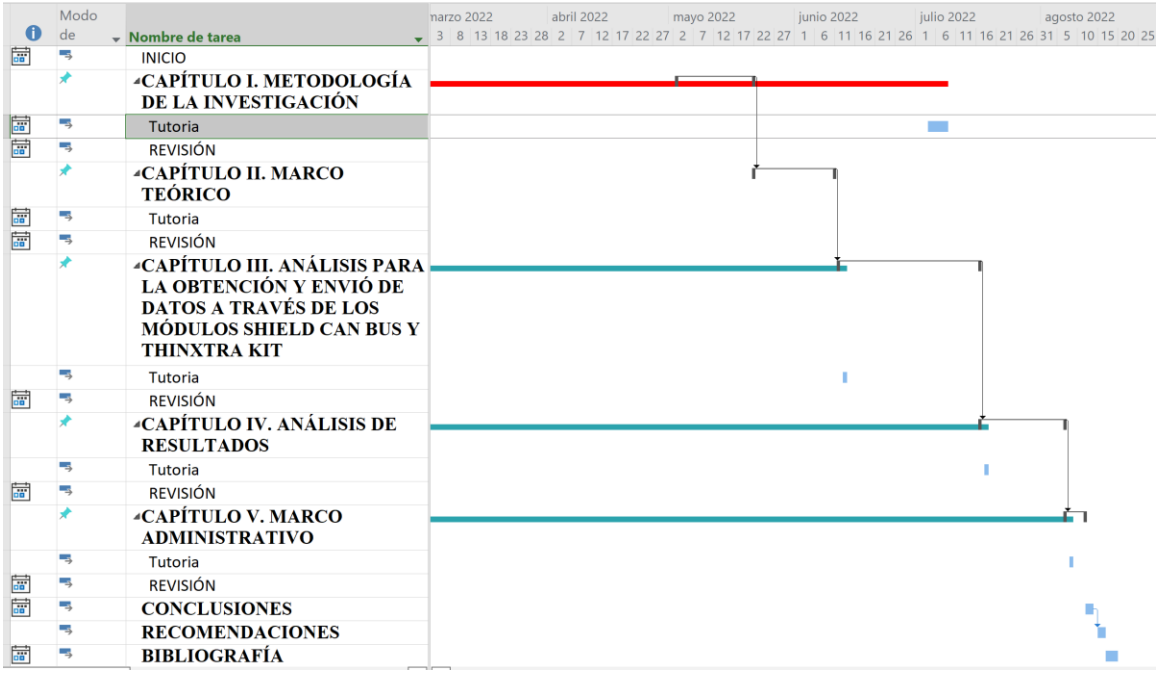
Orden	Detalle	Valor total USD
1.	Materiales	218,5
2.	Pruebas	48
3.	Imprevistos	250
	Total	516,5

Cronograma

Detalle de las actividades realizadas durante el desarrollo del proyecto de investigación.

Figura 72

Cronograma



Conclusiones.

- Se desarrolló el diagnóstico avanzado en sistemas de inyección de gasolina mediante el uso de la red LPWAN - SIGFOX para determinar planes de mantenimiento y reparación en tiempo real.
- Se obtuvo información sobre la aplicación de los conceptos de IoT en conjunto con la tecnología LPWAN – SIGFOX y el módulo Shield Can Bus V2.0, que facilitó el diagnóstico, la realización de planes de mantenimiento y reparación para el sistema de inyección electrónica a gasolina utilizando temas relacionados con el proyecto como: libros, artículos científicos, investigaciones científicas, páginas de internet de contenido confiable.
- Con la ayuda de fuentes bibliográficas y fichas técnicas de los fabricantes se fundamentó teóricamente los conceptos a utilizar en la investigación como el IoT, redes de comunicación, programación, módulo Shield Can Bus V2.0, placa SIGFOX RCZ4 DEV KIT, Arduino uno, sistemas de inyección de gasolina, OBD-II, planes de mantenimiento y reparación.
- Para la obtención de los datos y envío a la nube de estos, se realizó el código de programación para el funcionamiento del módulo Shield Can Bus V2.0, juntamente con la placa de Arduino UNO y el módulo SIGFOX RCZ4 DEV KIT.
- Una vez conocidos los sensores de los que se va a obtener los datos se generó la base de datos de los rangos de funcionamiento permitidos en los sensores del sistema de inyección según el tipo de vehículos a los que se vaya a monitorear.
- Se procesó la información obtenida de los sensores del sistema de inyección electrónica a gasolina extraída con el módulo Shield Can Bus V2.0 y enviada backend de Sigfox.

- Con los datos en el backend de Sigfox se configuró el Callback de forma que se puedan observar los datos obtenidos de la ECU mediante la utilización de widgets en el dashboard.
- Se revisó que los valores obtenidos se encuentren dentro de los parámetros correctos de funcionamiento, para esto se comparó la información recopilada del OBD-II mediante el uso del dashboard con los valores de la base de datos previamente investigada.
- Se analizó los valores obtenidos de los componentes del sistema de inyección del vehículo con la base de datos para determinar el procedimiento que se deberá seguir al momento de reparar las averías.
- El plan de reparación se creó para el caso de que los sensores del sistema de inyección funcionen fuera del rango correcto, se deben seguir los pasos especificados en este para poder arreglar la avería.
- Para que un vehículo tenga una vida útil larga se debe tener en cuenta un plan de mantenimiento del sistema de inyección para los vehículos que se diseñó en función del kilometraje y/o estado de los componentes, este plan de mantenimiento no se puede utilizar dentro de la aplicación debido a que el PID del odómetro no se pudo obtener.

Recomendaciones.

- Utilizar la información sobre IoT de las diferentes ramas y clasificarlas de forma que sean útiles en el área automotriz.
- Para las pruebas hay que revisar que el vehículo tenga protocolo de comunicación SAE 15765-4, se observa que en el conector OBD II se tengan los pines 6 y 14.
- En la recolección de datos revisar que la conexión del módulo con el puerto OBD II se encuentre bien de forma que se puedan obtener los valores de los sensores que se está analizando.
- Cuando existan fallas en el sistema de inyección y el módulo envié el mensaje de advertencia, se deberá utilizar otros equipos para el diagnóstico como son el osciloscopio y el scanner.
- Ubicar el módulo en un lugar donde no tenga interferencia en la antena, para lo cual se sitúa sobre el tablero de instrumentos, cerca del parabrisas.

Bibliografía.

Allauca, F., & Bryan, D. (2020). Desarrollo de un Sistema de Parqueadero Inteligente Mediante una Red LPWAN. 1(1), 1–171. http://etd.eprints.ums.ac.id/14871/%0Ahttps://doi.org/10.1016/j.cell.2017.12.025%0Ahttp://www.depkes.go.id/resources/download/info-terkini/hasil-riskesdas-2018.pdf%0Ahttp://www.who.int/about/licensing/%0Ahttp://jukeunila.com/wp-content/uploads/2016/12/Dea

Atmel. (2015). Inspiring Smart and Secure Connected Designs. 11, 28.

Barelli, L., Bidini, G., Cinti, G., Zhang, H. H., Wang, L., Van, J., Mar, F., Desideri, U., Khalil, A., Tauler, C. M., Pantou, S., Nr, S., Ouyang, L., Ma, M., Huang, M. S., Duan, R., Wang, H., Sun, L., Zhu, M., … Intl, S. (2018). Propuesta de un Modelo de Aplicación de IoT y Telemetría en los Procesos de Servicios de Taller para Empresas Concesionarias Automotrices. In Energies (Vol. 6, Issue 1). http://journals.sagepub.com/doi/10.1177/1120700020921110%0Ahttps://doi.org/10.1016/j.reuma.2018.06.001%0Ahttps://doi.org/10.1016/j.arth.2018.03.044%0Ahttps://reader.elsevier.com/reader/sd/pii/S1063458420300078?token=C039B8B13922A2079230DC9AF11A333E295FCD8

Bermeo, J. (2019). Diseño, desarrollo e implementación de una solución IoT para medir los niveles de radiación solar, usando una red (SIGFOX) de cobertura amplia de baja potencia (LPWAN) en la facultad de Ingeniería de la Universidad Católica de Santiago de Guayaquil. 120. http://repositorio.ucsg.edu.ec/bitstream/3317/12587/1/T-UCSG-PRE-ING-CIS-213.pdf

Blum, J. (2019). Exploring arduino too: Tools and Techniques for Engineering Wizardry.

Bonilla-Fabela, I., Tavizon-Salazar, A., Morales-Escobar, M., Guajardo-Muñoz, L., &

Laines-Alamina, C. (2016). Iot, El Internet de las Cosas y la Innovación de sus Aplicaciones. VInculaTégica EFAN, 2(1), 2313–2340. http://www.web.facpya.uanl.mx/Vinculategica/Revistas/R2/2313-2340 - Iot, El Internet De Las Cosas Y La Innovacion De Sus Aplicaciones.pdf

Correa, H. (2013). DISEÑO DE UN SIMULADOR DE SEÑALES BÁSICAS PARA UN SISTEMA DE INYECCION ELECTRONICA DE GASOLINA.

CSSELECTRONICS. (2022a). csselectronics.pdf. OBD2 Explicado: Una Introducción Simple. https://www.csselectronics.com/pages/obd2-explained-simple-intro

CSSELECTRONICS. (2022b). csselectronics.pdf. OBD2 Explicado: Una Introducción Simple. csselectronics 2.pdf. (n.d.).

Denton, T. (2016). Advanced Automotive Fault Diagnosis, 4th ed. In Advanced Automotive Fault Diagnosis, 4th ed (Quinta). Routledge. https://doi.org/10.4324/9781315856612

Guanoluisa, C., & Paucar, O. (2020). Universidad Politécnica Salesiana Sede Quito [UNIVERSIDAD POLITÉCNICA SALESIANA SEDE QUITO]. In Tesis. http://dspace.ups.edu.ec/bitstream/123456789/5081/1/UPS-CYT00109.pdf

Kia. (2019). Student Learning Guide and Workbook.

López, J. (2014). Diseño De Escáner Automotriz Obdii Multiprotocolo. In Universidad de San Carlos de Guatemala. https://docplayer.es/1287942-Diseno-de-escaner-automotriz-obdii-multiprotocolo.html

Microchip Technology Inc. (2007). Stand-Alone CAN Controller With SPI Interface. Writing, 1–84.

Montero, C., & Paguay, F. (2021). “ESTUDIO E IMPLEMENTACIÓN DE UN SISTEMA DE INYECCIÓN ELECTRÓNICA PROGRAMABLE PARA EL AUMENTO DEL RENDIMIENTO Y DISMINUCIÓN DE GASES DE ESCAPE CONTAMINANTES

EN UN VEHÍCULO SUZUKI FORSA GA.” UNIVERSIDAD POLITECNICA SALESINA SEDE CUENCA.

Obregón, J. (2016). ANÁLISIS DE WAVE FORMS EN SENSORES Y ACTUADORES DEL SISTEMA DE CONTROL ELECTRÓNICO SOBRE LA INYECCIÓN DE COMBUSTIBLE DEL MOTOR S4A DEL VEHÍCULO CHEVROLET SAIL .

SAE Standard. (2001). SAE J1962: Diagnostic Connector Equivalent to ISO/DIS 15031. SAE Standard, 552(1), 1806–5. https://law.resource.org/pub/us/cfr/ibr/005/sae.j1962.2002.pdf

SAE Standard. (2002). SAE J1979: E/E Diagnostic Test Modes. SAE Standard, 552(1).

SAE Standard. (2016). INTERNATIONAL STANDARD Road vehicles — Diagnostic communication over Controller Area iTeh STANDARD PREVIEW iTeh STANDARD PREVIEW. 2016.

Torrente, Ó. (n.d.). Arduino curso práctico de formación.

Torres, A. (2017). Sistema de diagnóstico automotriz mediante el análisis de emisiones contaminantes con la aplicación de redes neuronales para la detección de fallas. UNIVERSIDAD POLITÉCNICA SALESIANA SEDE QUITO.

Anexos