ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

“Diseño e implementación de un panel de control y monitoreo digital para activación de funciones y lectura de datos en tiempo real del vehículo Chevrolet Gemini”

Venegas Álvarez, David Andrés

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. Zambrano León, Víctor Danilo Mgtr.

13 de febrero del 2023

Latacunga

Reporte de verificación de contenido

Document Information

Analyzed document	"TESIS VENEGAS DAVID.pdf (D158565522)"
Submitted	2/13/2023 11:12:00 PM
Submitted by	Juan Carlos Altamirano
Submitter email	jc.altamiranoc@uta.edu.ec
Similarity	1%
Analysis address	jc.altamiranoc.uta@analysis.urkund.com

Sources included in the report

W	URL: https://www.dariomotor.com/2011/04/26/analogico-vs-digital-lo-moderno-frente-a-la-tradicion/ Fetched: 2/13/2023 11:12:00 PM	1
W	URL: https://www.autobodymagazine.com.mx/2018/09/12/ttd-vw-tablero-totalmente-digital/ Fetched: 2/13/2023 11:12:00 PM	1
W	URL: https://economipedia.com/definiciones/digitalizacion.html Fetched: 2/13/2023 11:12:00 PM	1
W	URL: https://aprendiendoarduino.wordpress.com/2015/03/26/lenguaje-de-programacion-de-arduino-estruc... Fetched: 2/13/2023 11:12:00 PM	2
W	URL: https://www.kionetworks.com/blog/data-center/protocolos-de-comunicaci%C3%B3n-de-redes Fetched: 2/13/2023 11:12:00 PM	2
W	URL: http://repositorio.uide.edu.ec/handle/37000/840 Fetched: 2/13/2023 11:12:00 PM	1
W	URL: https://oa.upm.es/45542/ Fetched: 2/13/2023 11:12:00 PM	1

Entire Document

Dedicatoria El proyecto de titulación va dedicado a Dios, por la sabiduría, inteligencia y paciencia que me ha entregado para lograr cumplir este objetivo. A mis padres, Mónica y Luis que me dieron el apoyo incondicional en cada momento del transcurrir de la carrera pues sin ellos no hubiera podido lograrlo. A mis hermanos, Daniel y Sebastián que fueron la inspiración para llegar a ser un excelente profesional como ellos y a mi familia más cercana tía, abuelito primas y sobrina.
Agradecimiento Agradezco infinitamente a Dios por sus bendiciones a lo largo de toda la carrera, desde que comenzó como un sueño hasta poder ver realizado el mismo. A mis padres, Mónica y Luis que sin su ejemplo y valores no hubiera logrado el objetivo que tenía propuesto hace muchos años, y enseñarme a perseverar como siempre lo han hecho. Mis hermanos y familia porque me han acompañado y apoyado en toda mi vida. Al Ing. Danilo Zambrano, por su apoyo y amistad en toda la carrera y su gran labor como docente. Amigos que estuvieron siempre presentes, con su amistad, conocimiento y motivación en el trascurso de la carrera.

Ing. Zambrano León, Víctor Danilo Mgtr.

C. C.: 0201740701

Departamento de Ciencias de la Energía y Mecánica

Carrera de Ingeniería Automotriz

Certificación

Certifico que el trabajo de titulación: **"Diseño e implementación de un panel de control y monitoreo digital para activación de funciones y lectura de datos en tiempo real del vehículo Chevrolet Gemini"** fue realizado por el señor **Venegas Álvarez, David Andrés** 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, 13 de febrero del 2023

Ing. Zambrano León, Víctor Danilo Mgtr.

C. C.: 0201740701

Departamento de Ciencias de la Energía y Mecánica

Carrera de Ingeniería Automotriz

Responsabilidad de Autoría

Yo, **Venegas Álvarez, David Andrés**, con cédula de ciudadanía N°1723488498, declaro que el contenido, ideas y criterios del trabajo de titulación: **Diseño e implementación de un panel de control y monitoreo digital para activación de funciones y lectura de datos en tiempo real del vehículo Chevrolet Gemini** es de mi 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, 13 de febrero del 2023

.....

Sr. Venegas Álvarez, David Andrés

C.C.: 1723488498

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Departamento de Ciencias de la Energía y Mecánica

Carrera de Ingeniería Automotriz

Autorización de Publicación

Yo, **Venegas Álvarez, David Andrés**, con cédula de ciudadanía N°1723488498, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Diseño e implementación de un panel de control y monitoreo digital para activación de funciones y lectura de datos en tiempo real del vehículo Chevrolet Gemini** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Latacunga, 13 de febrero del 2023

Sr. Venegas Álvarez, David Andrés

C.C.: 1723488498

Dedicatoria

El proyecto de titulación va dedicado a Dios, por la sabiduría, inteligencia y paciencia que me ha entregado para lograr cumplir este objetivo.

A mis padres, Mónica y Luis que me dieron el apoyo incondicional en cada momento del transcurrir de la carrera pues sin ellos no hubiera podido lograrlo.

A mis hermanos, Daniel y Sebastián que fueron la inspiración para llegar a ser un excelente profesional como ellos y a mi familia más cercana tía, abuelito primas y sobrina.

Agradecimiento

Agradezco infinitamente a Dios por sus bendiciones a lo largo de toda la carrera, desde que comenzó como un sueño hasta poder ver realizado el mismo.

A mis padres, Mónica y Luis que sin su ejemplo y valores no hubiera logrado el objetivo que tenía propuesto hace muchos años, y enseñarme a perseverar como siempre lo han hecho.

Mis hermanos y familia porque me han acompañado y apoyado en toda mi vida.

Al Ing. Danilo Zambrano, por su apoyo y amistad en toda la carrera y su gran labor como docente.

Amigos que estuvieron siempre presentes, con su amistad, conocimiento y motivación en el trascurso de la carrera.

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ECUACIONES

Resumen

En el presente proyecto de titulación se desarrolla una investigación de los sistemas digitales, aplicados en la industria automotriz en la actualidad. En primer lugar, se conoce sobre las GUI (graphical user interface) interfaz gráfica de usuario con la finalidad de entender la lógica de programación y conexión con el microcontrolador, igualmente se realiza la programación en la interfaz de la pantalla digital táctil con el propósito de conocer sus características, encontrar todas sus funciones y añadirlas al panel táctil con la interfaz que se tiene establecida. A continuación, se programa el algoritmo en el microcontrolador para la conexión con la pantalla táctil del fabricante NextionTech, la detección de datos, activación de actuadores, testigos de advertencia o emergencia; con la ayuda de los componentes eléctricos y electrónicos, se recopila datos como: velocidad, revoluciones por minuto, temperatura del motor, luminosidad, nivel de combustible entre otros, posteriormente se diseña y construye una placa PCB que integra todos los puertos de entrada y salida que el proyecto necesita. De igual manera se genera un plano eléctrico, que integra los nuevos componentes que se añaden al vehículo como el panel digital táctil, el microcontrolador, los sensores y actuadores en el vehículo Chevrolet Gemini. Finalmente se realiza pruebas y ensayos con la finalidad de comprobar el funcionamiento de los sensores y actuadores, así como la interacción entre el microcontrolador, la pantalla táctil y el conductor al momento de poner el vehículo en marcha, se sigue una ruta trazada con el objetivo de reportar novedades o mal funcionamiento de cada una de las los sensores o actuadores instalados en el vehículo y reportar mediante un informe el funcionamiento de todos los sistemas que se censaron o activaron.

Palabras clave: Sistemas digitales, interfaz gráfica, microcontrolador, pantalla táctil, electrónica.

Abstract

In the present titling project, an investigation of digital systems, applied in the automotive industry today, is carried out. In the first place, it is known about the GUI (graphical user interface) graphical user interface in order to understand the programming logic and connection with the microcontroller, programming is also carried out in the interface of the digital touch screen with the purpose of knowing its features, find all its functions and add them to the touch panel with the interface that you have established. Next, the algorithm is programmed in the microcontroller for the connection with the touch screen of the manufacturer NextionTech, the detection of data, activation of actuators, warning or emergency lights; With the help of electrical and electronic components, data such as: speed, revolutions per minute, engine temperature, luminosity, fuel level among others are collected, later a PCB board is designed and built that integrates all input and output ports. that the project needs. In the same way, an electrical plan is generated, which integrates the new components that are added to the vehicle such as the digital touch panel, the microcontroller, the sensors and actuators in the Chevrolet Gemini vehicle. Finally, tests and trials are carried out in order to verify the operation of the sensors and actuators, as well as the interaction between the microcontroller, the touch screen and the driver when starting the vehicle, a route is followed with the objective to report news or malfunctions of each of the sensors or actuators installed in the vehicle and to report, through a report, the operation of all the systems that were recorded or activated.

Keywords: Digital systems, graphical interface, microcontroller, touch screen, electronics.

Capítulo I

Generalidades

Antecedentes investigativos

El avance de la tecnología y la electrónica aplicada en el campo de la automoción, ha traído un desarrollo de los sistemas digitales en los vehículos, para generar una relación más estrecha entre la información del vehículo y el conductor.

Se puede establecer que el desarrollo de la electrónica en general empieza en la década de los 70´s, y poco a poco se fue introduciendo al mundo de los automóviles. De 1978 a mediados del año 2000, se integran en un vehículo nueve subsistemas: ocho combinan sensores, unidades de control electrónico y unidades de control hidráulico; uno — los frenos electromecánicos— combinan sensores, unidades de control electrónico y solenoides (Leen & Heffernan, 2002).

Existe también una evolución en cuanto a la transmisión de datos. En la primera evolución de los arneses la principal y exclusiva función de los cables era conducir electricidad. En la segunda se produce cuando los cables conducen electricidad e información; este sistema se caracteriza por un cable/una señal. Y la tercera se caracteriza porque el cable puede transmitir más de una señal (sistemas multiplexos) (Figueroa, 2015).

El fabricante en función de las condicionantes de cada momento diseña y fabrica un vehículo que solvente los requisitos sociales y legales utilizando las tecnologías disponibles, que den como resultado un mejor producto al costo más apropiado (María Fernández-Albalat, 2017).

Todos estos avances, han conducido a un ritmo acelerado de innovación del diseño y manufactura de los ECU, sensores, solenoides y arneses, así como a la administración del volumen de energía e información asociados a una red cada vez más extensa e

interdependiente (Villar Bonet, 2017).

Siempre hemos asociado el automóvil a evolución tecnológica y progreso, y sería ingenuo pensar que nos ha tocado vivir una época especial. Con mucha calma y buen criterio, muchos coches adoptarán fórmulas más innovadoras para mostrar su instrumentación (Seijo, 2011).

Una pantalla táctil ha sustituido a los interruptores, ahora hay perfiles de usuario y un ambiente personalizado en lugar de un entorno igual para todos. La digitalización en Volkswagen avanza a pasos agigantados. Hay una enorme diferencia entre el primer centro de información digital para el conductor de 1986 y la actual Active Info Display (Autobody, 2018).

Planteamiento del problema

Con el paso del tiempo, la utilización de la electrónica en los vehículos ha incrementado significativamente, pues cada vez se ha visto la necesidad de obtener datos específicos para mejorar el rendimiento de un vehículo y adaptarse a las exigencias del mercado local y mundial.

A raíz de ello, las empresas automotrices han migrado a simplificar las formas en las que se presentan datos inherentes del vehículo a través de sistemas electrónicos o digitales con la ayuda de pantallas digitales táctiles, pues con el cambio de motorización, ya sea en forma de hibridación o electrificación, las compañías encontraron que la actualización de la tecnología no solo debía concentrarse en la parte mecánica, sino también en el interior de los vehículos.

Con lo antes mencionado, en el mercado local y en las mallas curriculares no se encuentra un acercamiento a la programación y análisis del funcionamiento de los sistemas

digitales que usan las pantallas táctiles, donde fácilmente se podría encontrar un desconocimiento si existe un problema con esta nueva forma de interactuar entre el pasajero y el vehículo; en tal sentido los vehículos que ya circulan en el Ecuador y que sin lugar a dudas después de transcurrir un cierto tiempo, estas pueden fallar, el dueño se va a encontrar con que aún no existe gente capacitada para reparar dichos sistemas.

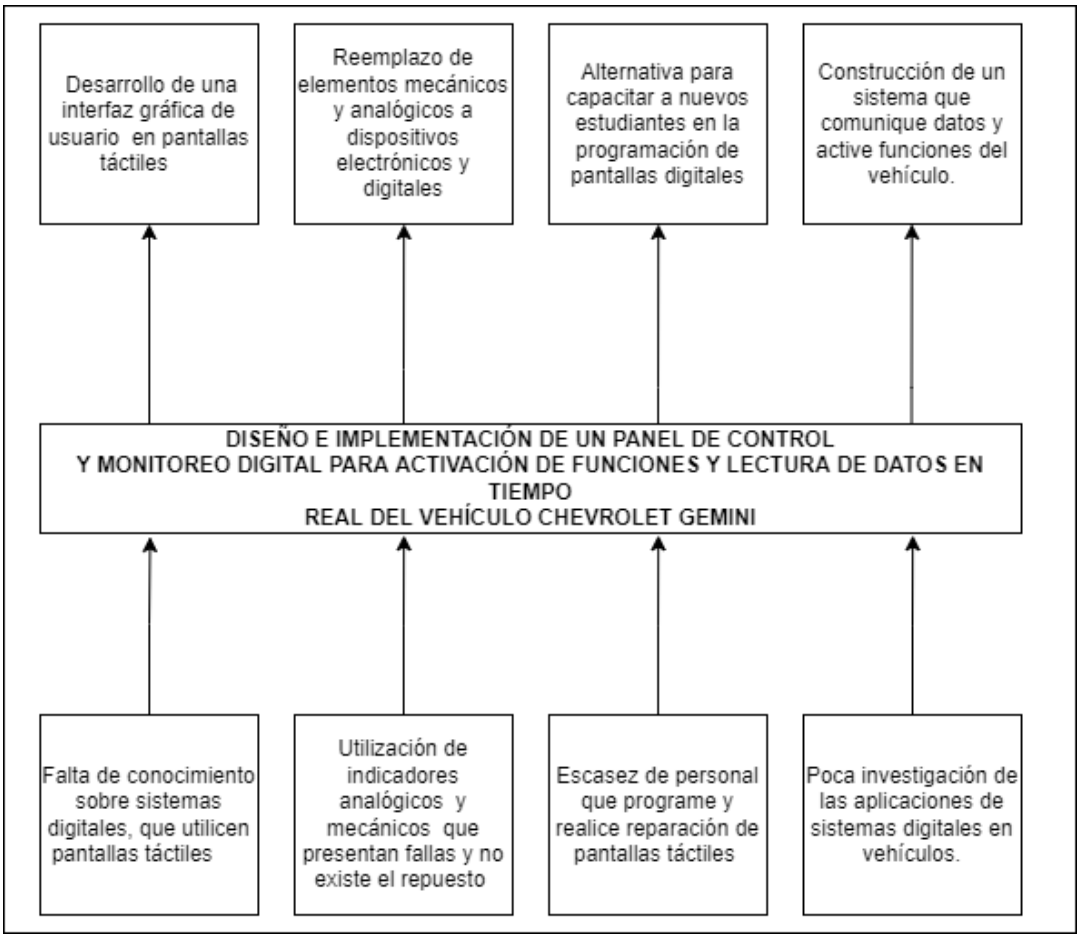
El vehículo al que se pretende realizar dichos implementación también ha sufrido varios daños en el panel de control donde algunos indicadores y actuadores están averiados y al ser un vehículo antiguo no se encuentra fácilmente en el mercado repuestos para suplir las fallas que ha sufrido y la opción de adaptar indicadores por fuera, requiere un costo mayor además de cambiar la estética del vehículo.

Por estos motivos el problema que se plantea investigar es el siguiente:

Diseño e implementación de un panel de control y monitoreo digital para activación de funciones y lectura de datos en tiempo real del vehículo Chevrolet Gemini

Figura 1

Árbol de problemas



Descripción detallada del proyecto

En el presente trabajo de titulación se desarrollarán los siguientes procesos:

Se recopilará información de todos los componentes eléctricos y electrónicos, así como sensores, actuadores, microcontroladores y pantalla digital táctil, que sean los más adecuados para la recopilación de datos como: velocidad, revoluciones por minuto, presión de aceite, temperatura del motor, luminosidad, etc.

Se realizará la programación en la interfaz de la pantalla digital táctil con el fin de conocer el lenguaje de programación, encontrar todas sus funciones y añadirlas al panel táctil en toda la interfaz que se tiene planteada.

Se programará el algoritmo en el microcontrolador para la detección de datos y posteriormente la activación de actuadores o indicadores.

Se generará un nuevo plano eléctrico con el cual se integre los nuevos componentes que se añadirán al vehículo.

Se implementará el panel digital táctil, el microcontrolador y los sensores y actuadores en el vehículo Chevrolet Gemini.

Se realizarán pruebas y ensayos con la finalidad de comprobar el funcionamiento de los sensores y actuadores, así como la interacción entre el microcontrolador y la pantalla táctil al momento de poner el vehículo en marcha y seguir una ruta trazada.

Se generará un informe del funcionamiento de todos los sistemas que se utilizaron en la prueba de ruta, para reportar novedades o mal funcionamiento de ser el caso, de cada una de las variables a analizar.

Justificación e importancia

La industria automotriz, se ha adaptado a todos los avances tecnológicos desde la creación de los microcontroladores y la electrónica aplicada, ya sea para emisiones menos contaminantes, eficiencia energética, confort del pasajero y la era digital y de telecomunicaciones que ha crecido desde inicios de este siglo.

En el pasado, el panel de instrumentos presentaba la información más básica del vehículo, sin la necesidad de entregar mayor información al conductor, pero eso ha cambiado y en la actualidad ya no es suficiente los datos que entrega el vehículo hacia el conductor, ahora

con el cambio y evolución progresiva de la tecnología debe también existir información de la interacción del vehículo con el entorno que le rodea, sean personas , objetos estáticos o móviles, la ubicación en tiempo real del vehículo, temperatura interna, alertas de tráfico cruzado, la música que se está reproduciendo, etc.

Con lo antes mencionado es necesario que exista un panel de control, que permita que la información sea entregada hacia al conductor de la manera más fácil y entendible posible; para recopilar toda esta información, las compañías automotrices vieron la importancia de transmitir al conductor toda esa data de una manera más amigable, no tan monótona como se lo realizaba antes como informar de un error solo encendiendo una lámpara o indicador luminoso, ahora es necesario que se evidencien figuras, colores, animaciones de lo que está ocurriendo en nuestro entorno al momento de manejar; precisamente la incorporación de las pantallas digitales táctiles permiten esta visualización con mayor detalle.

En otro aspecto, ante la avería de indicadores como temperatura y nivel de combustible, omisión de un indicador de revoluciones del motor y el daño de interruptores originales del vehículo para encender luces y limpiaparabrisas en el vehículo Chevrolet Gemini, se busca solucionar los problemas que el automóvil tiene y añadir funciones que mejoren la seguridad activa del vehículo, ya sea al indicar el límite de velocidad, señalar si el conductor tiene puesto el cinturón de seguridad, cierre de puertas automático, renovar los componentes eléctricos y mecánicos con otros que estén más acordes con la realidad de la industria automotriz.

El conocer cómo funciona la programación, la forma de comunicación y los nuevos desarrollos que se pueden aplicar a vehículos actuales que llegan al mercado nacional, así como el despliegue de la información a través de una pantalla digital táctil, aporta al perfil del ingeniero automotriz permitiéndole encontrar soluciones más eficientes a la vanguardia de la tecnología actual, existente en los automotores de hoy en día.

Objetivos del proyecto

Objetivo principal

Diseñar e implementar un panel de control y monitoreo digital para activación de funciones y lectura de datos en tiempo real del vehículo Chevrolet Gemini.

Objetivos específicos

- Investigar los diferentes componentes eléctricos y electrónicos como sensores y actuadores con el fin de encontrar los mejores elementos y prestaciones para la implementación del trabajo de titulación.
- Diseñar el panel de control y monitoreo en el software de interfaz gráfica de usuario de la marca NEXTION para la pantalla táctil Nextion NX8048P070.
- Programar el algoritmo para recepción de datos y activación de funciones en el microcontrolador Arduino e integrar con la pantalla táctil.
- Generar los nuevos planos eléctricos del vehículo Chevrolet Gemini.
- Instalar la pantalla táctil y efectuar pruebas de obtención de datos y activación de funciones a través de una prueba de ruta, para validar la información recibida.

Metas

- Actualizar al vehículo Chevrolet Gemini con nueva tecnología, con el fin de reemplazar los elementos mecánicos y analógicos por elementos eléctricos y digitales para mejorar la interacción entre el conductor y el vehículo, aumentar las funciones propias del vehículo además de mejorar su seguridad activa.
- Desarrollar un sistema que permita ser utilizado por estudiantes de la carrera de Ingeniería Automotriz, para la implementación en vehículos comerciales, prototipos de competencia, vehículos eléctricos, como precedente de las nuevas tecnologías que se utiliza en los automóviles en la actualidad.

Hipótesis

Al implementar un panel digital para activación de funciones y lectura de datos en tiempo real del vehículo Chevrolet Gemini permitirá, mejorar la interacción del vehículo con el conductor aumentando la seguridad del mismo.

Variables de investigación

Variable independiente

Panel de control digital

Tabla 1

Variables medibles del panel de control

Concepto	Categoría	Indicadores	Ítem	Técnicas	Instrumentos
El panel de control se encarga de indicar los valores característicos del vehículo e indicar si existe algún error	TÉCNICA TECNOLÓGICA	Temperatura del motor	t	Experimentación	Protocolo de pruebas
	TÉCNICA TECNOLÓGICA	Velocidad lineal	$\frac{Km}{h}$	Experimentación	Protocolo de pruebas
	TÉCNICA TECNOLÓGICA	Velocidad angular	rpm	Experimentación	Protocolo de pruebas
	TÉCNICA TECNOLÓGICA	Nivel de combustible	V	Experimentación	Protocolo de pruebas
	TÉCNICA TECNOLÓGICA	Presión de aceite	Pa	Experimentación	Protocolo de pruebas

Variable dependiente

Sistema digital de monitoreo y activación de funciones

Tabla 2

Operacionalización de la variable dependiente.

Concepto	Categoría	Indicadores	Ítem	Técnicas	Instrumentos
Se trata de implementar un panel digital táctil con el cual se pueda observar datos en tiempo real del vehículo, así como activar funciones	TECNICA TECNOLÓGICA	Voltaje de batería	V	Experimentación	Protocolo de pruebas
	TECNICA TECNOLÓGICA	Encendido de luces	V	Experimentación	Protocolo de pruebas
	TECNICA TECNOLÓGICA	Sensor de lluvia	Ω	Experimentación	Protocolo de pruebas
	TECNICA TECNOLÓGICA	Sensor de luminosidad	Ω	Experimentación	Protocolo de pruebas
	TECNICA TECNOLÓGICA	Cierre de puertas	V	Experimentación	Protocolo de pruebas
	TECNICA TECNOLÓGICA	Velocidad Máxima	$\frac{Km}{h}$	Experimentación	Protocolo de pruebas
	TECNICA TECNOLÓGICA	Alerta de cinturón de seguridad conductor	dB	Experimentación	Protocolo de pruebas
	TECNICA TECNOLÓGICA	Alerta de luces encendidas al apagar el motor	dB	Experimentación	Protocolo de pruebas

Capítulo II
Marco Teórico

Sistemas digitales

El gran desarrollo experimentado por la electrónica en los últimos años ha propiciado que la mayoría de los equipos actuales funcionen con sistemas digitales. Un sistema digital se caracteriza por utilizar señales discretas, es decir, señales que toman un número finito de valores en cierto intervalo de tiempo.

Digitalización del panel de control y salpicadero

El portal web Economipedia, define a la digitalización como: “el proceso por el cual procesos analógicos y objetos físicos se convierten al formato digital.” (Guillermo, 2021)

La actualidad de la industria automotriz ha permitido acercarse más a las necesidades del cliente que en años anteriores. No es suficiente que el vehículo tenga un diseño actual, sus motores sean eficientes o que emitan menos gases contaminantes, sino también la interacción entre el vehículo y el conductor tiene que verse reflejado de manera más orgánica y eficiente, dejando de lado los indicadores analógicos y físicos además de alertas que un vehículo solía tener.

De esta manera existe un cambio dentro de los vehículos a reemplazar botones, activadores, agujas que se mueven por un campo eléctrico manejado por voltaje, a botones representados en una pantalla táctil y números, para tener un valor exacto de diferentes datos del vehículo.

Lo que nos lleva a revisar más detalladamente la evolución de la digitalización dentro del vehículo, (Figura 2) en este caso se hace presente a través de paneles digitales, pantallas táctiles donde el conductor puede ver la información del vehículo y activar funciones de este, a través de la interfaz o programación que presenta cada vehículo.

Figura 2

Salpicadero convencional vs Digital

Nota. Ejemplo de cómo las empresas automotrices, adaptan a sus vehículos a los avances tecnológicos y el pedido que realizan los clientes.Tomado de *Evolución del tablero automotriz*, por AutobodyMagazine,2018.

Pantallas digitales táctiles

Las pantallas son dispositivos que aúnan las interfaces de comunicación de la máquina con el usuario y viceversa, es decir, permiten la interacción de la máquina con el usuario. Estos dispositivos aportan al usuario una interfaz fácil de emplear y principalmente una forma sencilla de efectuar las instrucciones u operaciones requeridas por el usuario de manera táctil, es decir, se da pie a la instrucción u operación empleando el tacto por lo que no se requiere un dispositivo externo para efectuar la comunicación con la máquina. (Moreno Díaz, Aller Pardo, & Mariano Pulido, 2001)

Las pantallas táctiles son dispositivos de entrada de datos sencillos de usar, proporcionando y brindando al usuario una respuesta rápida y una instrucción u operación sin errores, la entrada de datos al tacto es la mejor opción para la realización de proyectos innovadores, en donde intervienen los sistemas de información. (Lopez Cordero, 2006)

Figura 3

Interior del tesla Model S

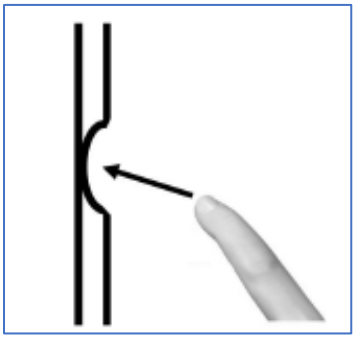
Nota. Los nuevos vehículos, en especial los eléctricos han cambiado completamente el interior, en la distribución y botonería. Tomado de *Tesla model S,* por Tesla, 2022.

Pantalla táctil resistiva

Compuesta por varias capas conductivas separadas por diversos puntos formando una matriz. Cuando se pulsa sobre la pantalla los puntos de las distintas capas se juntan produciéndose conducción eléctrica. Midiendo el voltaje se puede saber en qué punto el usuario pulsó la pantalla. Tiene la ventaja de que se puede utilizar con guantes o con cualquier objeto. (Moreno, 2014)

Figura 4

Pantalla táctil resistiva



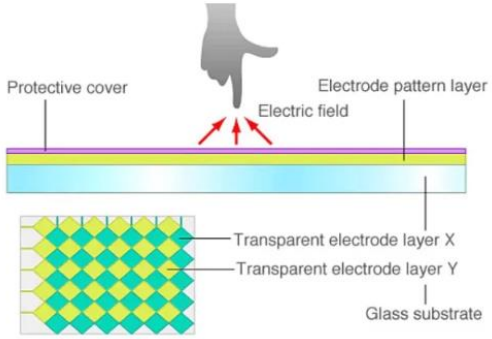
Nota. Funcionamiento de la pantalla táctil con tecnología resistiva. Tomado de *Pantallas táctiles* (p.225), por Moreno,2014.

Pantalla táctil capacitiva

Sólo reacciona al dedo desnudo y detecta la posición mediante la medida de capacitancia eléctrica y las variaciones de fase eléctrica producidas al interrumpirse la corriente que circula por la superficie de la pantalla. (Moreno, 2014)

Figura 5

Sensor táctil capacitivo



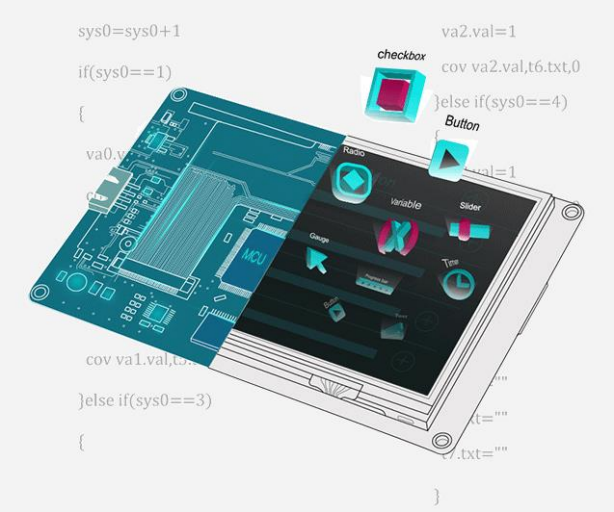
Nota. Tecnología aplicada en la pantalla táctil de la mayoría de los dispositivos táctiles en la actualidad. Tomado de *¿Qué es mejor: pantalla táctil resistiva o capacitiva ?*,por Orientdisplay,2014.

Nextion tech

Nextion es una solución de interfaz hombre-máquina (Human-Machine Interface, HMI) que combina un procesador integrado y una pantalla táctil con memoria con el software Nextion Editor para el desarrollo de proyectos de GUI HMI. (NEXTIONTECH, 2022)

Figura 6

Interfaz Nextion Tech



Nota. Muestra de la estructura que disponen las pantallas táctiles de la marca Nextion.Tomado de *Pantallas Nextion,* por NEXTIONTECH,2022.

Nextion GUI EDITOR

El concepto de GUI se define en inglés como: GRAPHIC USER INTERFACE que traducido al español es: La Interfaz Gráfica de Usuario, donde su definición es la siguiente:

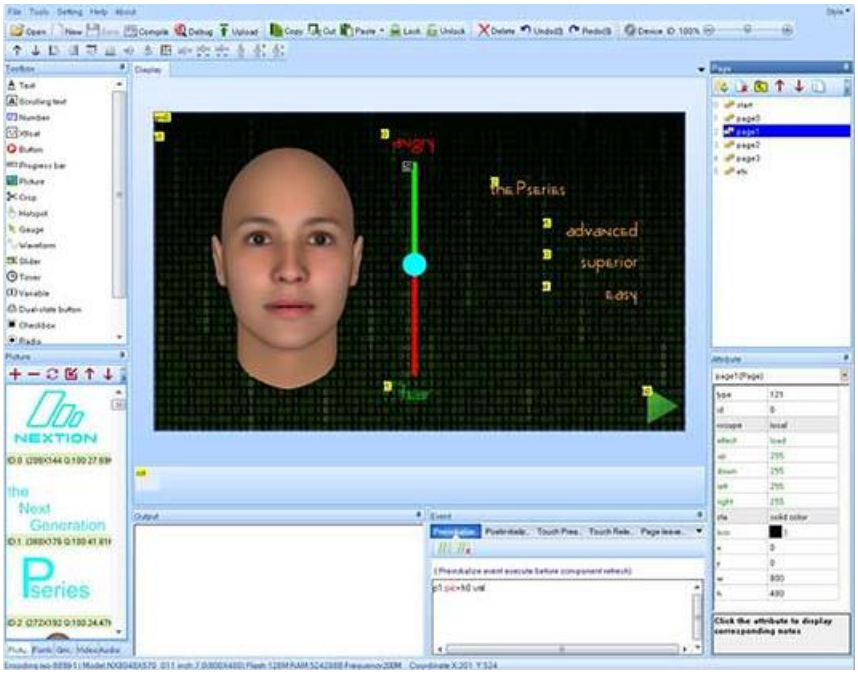
Una graphical user interface o GUI es una interfaz que se puede utilizar para controlar PC, tabletas y otros dispositivos. Las GUI utilizan elementos gráficos como iconos, menús e imágenes para facilitar el manejo del usuario humano. (NEXTIONTECH, 2022)

Con el software Nextion Editor, puede desarrollar rápidamente la GUI de HMI arrastrando y soltando componentes (gráficos, texto, botón, control deslizante, etc.) e

instrucciones basadas en texto ASCII para codificar cómo interactúan los componentes en el lado de la pantalla. (NEXTIONTECH, 2022)

Figura 7

Nextion GUI Editor



Nota. Para poder realizar la programación en la pantalla táctil, se necesita de un software, donde se puede editar los objetos y realizar la programación en la pantalla. Tomado de *Made for HMI GUI,* por NEXTIONTECH,2022.

Nextion serie inteligente

La marca de pantallas táctiles Nextion, ofrece en su catálogo varios tipos de pantallas con diferentes características desde una serie básica para proyectos con poca demanda de procesamiento, hasta la serie inteligente que ofrece mayores prestaciones para el tipo de proyecto que se pretende realizar.

Los productos de la serie inteligente tienen hardware más potente en términos de MCU, almacenamiento flash y SRAM en comparación con la serie básica y la serie mejorada. Las

funciones de reproducción de audio, video y animación enriquecen la interacción HMI del proyecto del usuario. La serie inteligente admite características y funciones de software avanzadas, como componente transparente, efecto de carga de página, mover y arrastrar componente, etc. (NEXTIONTECH, 2022)

Figura 8

7.0" Nextion Intelligent Series HMI Touch Display

Nota. Las pantallas Nextion tienen varias medidas de pantalla las cuales son usadas para diferentes propósitos. Tomado de *Nextion Intelligent Series HMI Touch Display,* por NEXTIONTECH,2002.

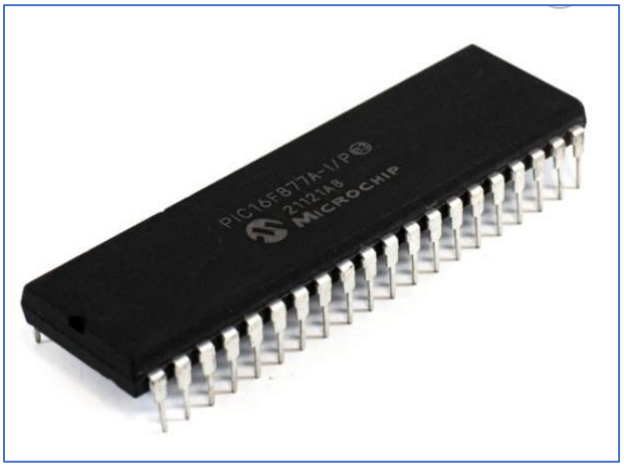
Microcontrolador

Un microcontrolador es un circuito integrado, en cuyo interior posee toda la arquitectura de un computador, esto es CPU, memorias RAM, EEPROM, y circuitos de entrada y salida. Un microcontrolador de fábrica no realiza tarea alguna, este debe ser programado para que realice desde un simple parpadeo de un led hasta un sofisticado control de un robot. Un microcontrolador es capaz de realizar la tarea de muchos circuitos lógicos como compuertas AND, OR, NOT, NAND, conversores A/D, D/A, temporizadores, decodificadores, etc.,

simplificando todo el diseño a una placa de reducido tamaño y pocos elementos. (Reyes , 2008)

Figura 9

Microcontrolador MICROCHIP



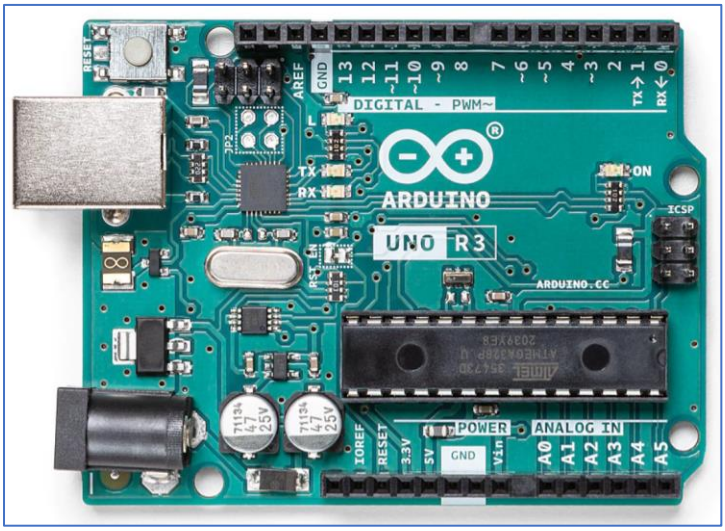
Nota. la marca microchip, es una de las empresas mas conocidas en el mundo de la electrónica y que tiene en su catálogo microcontroladores para desarrollo. Tomado de *PIC 16F877A-I-P,* por Microchip,2022.

Arduino

Una placa hardware libre que incorpora un microcontrolador reprogramable y una serie de pines-hembra (los cuales están unidos internamente a este dispositivo para su funcionamiento se compone de 3 bloques; puertos de entradas o salidas, CPU y memoria. El CPU o central de procesamiento se encarga de correr una secuencia de órdenes dictadas y almacenadas por las memorias para gestionar las salidas y entradas al dispositivo. (Torrente Artero, 2013)

Figura 10

Placa de hardware libre Arduino



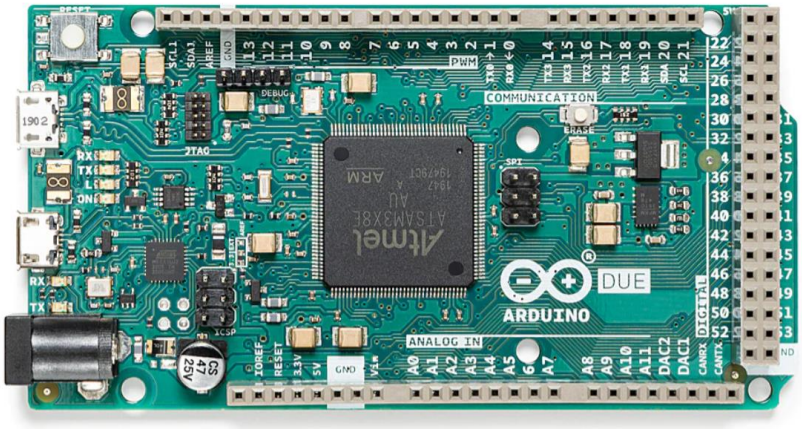
Nota. La placa Arduino Uno, es la más vendida de la empresa, con la cual se desarrolla proyectos de electrónica desde básicos hasta más avanzados. Tomado de *Placa Arduino UNO R3*, por Arduino,2021.

Arduino Due

Esta placa pertenece a una familia totalmente distinta de la del resto de placas Arduino. Incluye el microcontrolador SAM3X8E, el cual, aunque fabricado también por Atmel, es de una arquitectura interna muy diferente a la AVR (concretamente es de tipo ARM Cortex-M3) y además, sus registros son cuatro veces más grandes de lo habitual en las otras placas (concretamente, son de 32 bits). Su velocidad de reloj está también muy por encima del resto de placas Arduino (concretamente, es de 84 MHz). Además, el microcontrolador SAM3X8E dispone de mucha más memoria (concretamente, 96 KB de SRAM y 512 KB de memoria Flash) y también de un circuito especializado (llamado controlador “DMA”) que permite a la CPU acceder a la memoria de una manera mucho más rápida (Torrente Artero, 2013)

Figura 11

Placa de Arduino Due



Nota. Arduino Due la placa que utiliza un microcontrolador diferente, donde la característica principal es que las entradas y salidas utiliza y recibe 3.3 voltios a diferencia de las demás placas de la marca. Tomado de *Placa Arduino DUE*, por Arduino,2021.

Lenguaje de programación

Todos los microcontroladores o microprocesadores están basados en una lógica binaria de ceros y unos, estado alto o bajo o verdad o falso, pero cada tipo de microcontrolador para llegar a ese estado necesita de una programación y dependiendo de la marca cada uno tendrá su propio lenguaje de programación.

Para programar un Arduino, el lenguaje estándar es C++, aunque es posible programarlo en otros lenguajes. No es un C++ puro, sino que es una adaptación que proveniente de avr-libc que provee de una librería de C de alta calidad para usar con GCC en los microcontroladores AVR de Atmel y muchas funciones específicas para los MCU AVR de Atmel. (Aprendiendoarduino, 2015)

Protocolo de comunicación

Un protocolo es un conjunto de reglas: los protocolos de red son estándares y políticas formales, conformados por restricciones, procedimientos y formatos que definen el intercambio de paquetes de información para lograr la comunicación entre dos servidores o más dispositivos a través de una red. (KIONETWORK, 2022)

Existen dos formas de realizar una comunicación binaria, la paralela y el serial. La comunicación paralela como por ejemplo la comunicación, en donde los datos viajan simultáneamente a través de los 4 hilos, tiene la ventaja de que la transferencia de datos es más rápida, pero el inconveniente es que necesitamos un cable por cada bit de dato, lo que encarece y dificulta el diseño de las placas, otro inconveniente es la capacitancia que genera los conductores por lo que la transmisión se vuelve defectuosa a partir de unos pocos metros. La comunicación serial en cambio es mucho más lenta debido a que transmite bit por bit, pero tiene la ventaja de necesitar menor cantidad de hilos, y además se puede extender la comunicación a mayor distancia, por ejemplo; en la norma RS232 a 15 mts., en la norma RS422/485 a 1200mts y utilizando un MODEM, pues a cualquier parte del mundo. (Reyes , 2008)

Comunicación serial

El microcontrolador ATmega328P dispone de un receptor/transmisor serie de tipo TTL-UART que permite comunicar la placa Arduino UNO con otros dispositivos (normalmente, nuestro computador), para así poder transferir datos entre ambos. El canal físico de comunicación en estos casos suele ser el cable USB, pero también pueden ser los pines digitales 0 (RX) y 1 (TX) de la placa. Si se usan estos dos pines para comunicar la placa con un dispositivo externo, tendremos que conectar concretamente el pin TX de la placa con el pin RX del dispositivo, el RX de la placa con el TX del dispositivo y compartir la tierra de la placa con la tierra del dispositivo. Hay que tener en cuenta que, si se utilizan estos dos pines para la

comunicación serie, no podrán ser usados entonces como entradas/salidas digitales estándar. (Torrente Artero, 2013)

Capítulo III

Diseño e implementación del sistema

Selección de componentes eléctricos y electrónicos

Para este trabajo de titulación se utilizará diferentes componentes eléctricos y electrónicos que permitirán obtener datos análogos del vehículo como velocidad, temperatura, nivel de combustible y a su vez poder activar los actuadores o cargas eléctricas como las luces del vehículo, limpiaparabrisas entre otros.

Relay

El relay es un componente eléctrico, el cual actúa como un interruptor que se puede activar eléctricamente ya que dispone de dos pasos, el primero irá la señal de activación o control y el segundo pasará la mayor cantidad de voltaje y corriente. En el caso del proyecto serán utilizados para activar las luces de cruce, carretera, neblineros y motor del limpiaparabrisas.

Figura 12

Relay

Nota. Relay de 5 pines marca Bosch. Tomado de *Relay automotriz*, por MaxCar,2022.

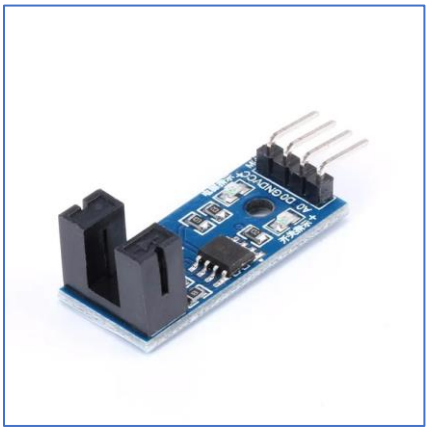
Sensor de velocidad

Al manejar la placa de desarrollo electrónico Arduino nos permite utilizar varios módulos, entre los cuales existe el módulo de detección de revoluciones por minuto, que utiliza

básicamente el principio de un opto acoplador, que, a través de un diodo infrarrojo, su luz llega al fototransistor, que permite el paso de corriente, y cuando hay una interrupción u obstáculo entre estos dos elementos, se detecta un cambio de estado, de esta manera se detecta las revoluciones del cable, se realiza el cálculo respectivo para la equivalencia de la velocidad del vehículo.

Figura 13

Módulo sensor velocidad de giro



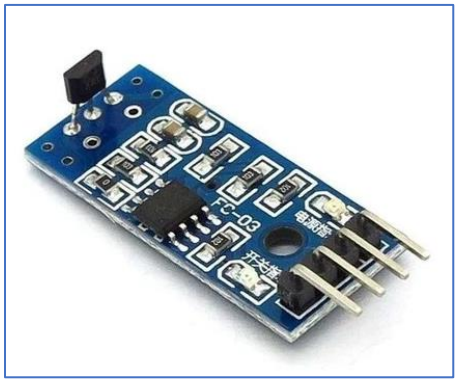
Nota. Módulo de sensor óptico digital. Tomado de *Modulo Sensor Medidor De Velocidad Rpm*, por Tecneu,2022.

Sensor de rpms

Para el caso de las rpms, el vehículo no posee un indicador análogo del cual se pueda tomar la señal, para realizar la transformación a digital, por ello existe la necesidad de instalar un sensor de efecto hall, en el cigüeñal, para realizar la toma respectiva de la información de rpm. Como se conoce, el efecto Hall su principio de funcionamiento es a través de un campo magnético, el cual se lo realiza por medio de un imán, de este modo para el desarrollo se utilizará un imán de neodimio, a instalarse en la polea del cigüeñal.

Figura 14

Modulo A3144e Sensor De Campo Magnético



Nota. Sensor de efecto Hall. Tomado de *Modulo A3144e Sensor De Campo Magnético Efecto Hall*, por Tecneu. 2022.

Sensor de temperatura

Para el censado de la temperatura de refrigerante, se utiliza el termo resistor tipo NTC, instalado originalmente en el vehículo, el cual estaba dispuesto para la aguja de la temperatura que se encuentra en el panel original pero que, por su tiempo, dejo de funcionar. La resistencia tiene un valor cercano a los 1500 ohm cerca de los 0 grados; el dato se obtuvo experimentalmente, con lo cual se realiza el cálculo de divisor de voltaje para obtener la señal digital.

Figura 15

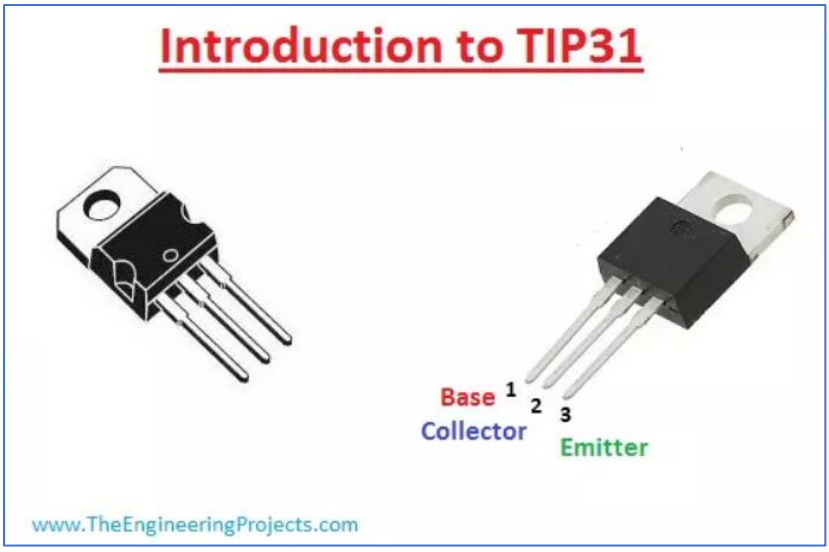
Sensor temperatura de refrigerante

Transistor de potencia

Los transistores de potencia se diferencian de un transistor NPN como el BC547, principalmente por permitir el paso de corrientes entre 1 a 5 A en su etapa de saturación o dependiendo de la necesidad que se desea aplicar, omitiendo el uso de relés y realizando la misma función. Para el presente proyecto se utiliza un transistor NPN TIP31, para activar la bomba de agua del limpiaparabrisas, así como luces de posición delanteras.

Figura 16

Transistor NPN TIP 31



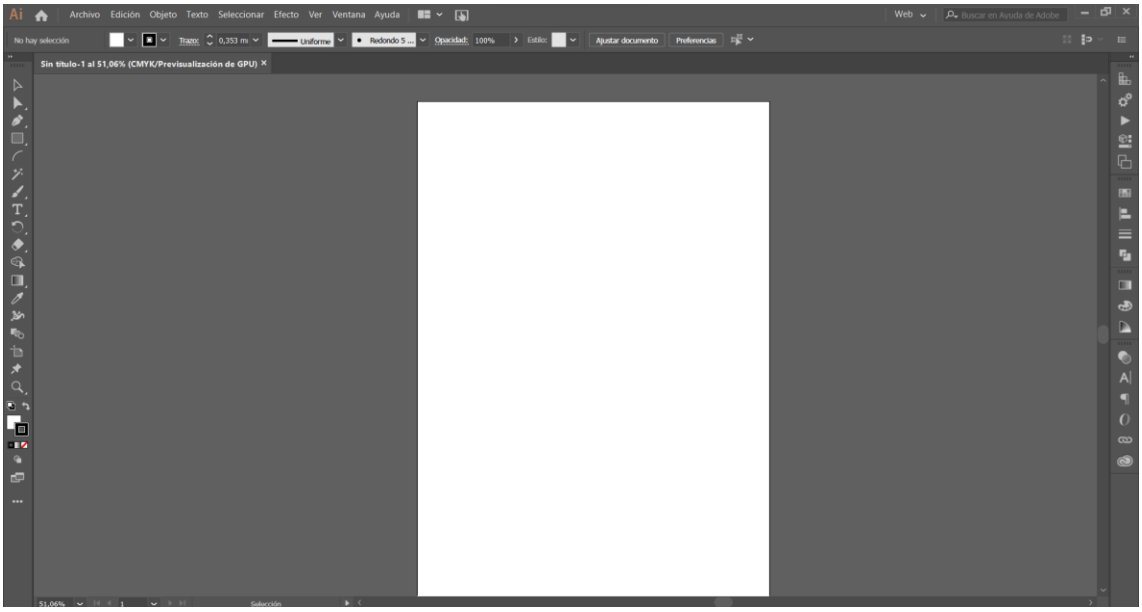
Nota. Transistor de potencia. Tomado de *Introduction to TIP31,* por Rahid,2019.

Diseño en forma vectorial del panel digital

Para desarrollar el diseño tipo ilustración de la pantalla que se ve en el dispositivo Nextion, se utiliza el software, Adobe Illustrator, el cual permite realizar ilustraciones de tipo vectorial, esto significa que su dimensión puede tomar cualquier valor sin que este se vea afectado en su diseño o calidad de imagen.

Figura 17

Software Adobe Illustrator



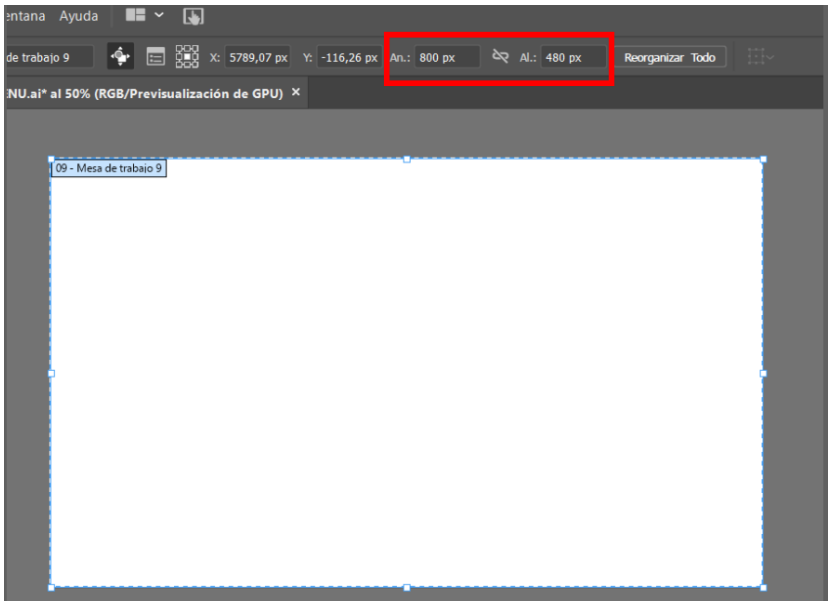
Diseño del panel de datos en tiempo real

Antes de empezar el diseño, primero se debe obtener la dimensión o pixeles que dispone la pantalla Nextion, para configurar la mesa de trabajo en el software Illustrator, para exportar la imagen diseñada al panel táctil.

La denominación de serie de la pantalla Nextion tiene un código de identificación (NX8048P070). El fabricante indica que los números que se encuentran entre las letras X y P son las dimensiones, de este modo se debe configurar la mesa de trabajo en un valor de 800 x480 pixeles.

Figura 18

Dimensionamiento de la mesa de Trabajo



Después de realizar la mesa de trabajo, se procede al diseño con el cual se va a manejar de forma táctil, la pantalla principal de datos en vivo y las pantallas para activar las diferentes funciones del vehículo.

Figura 19

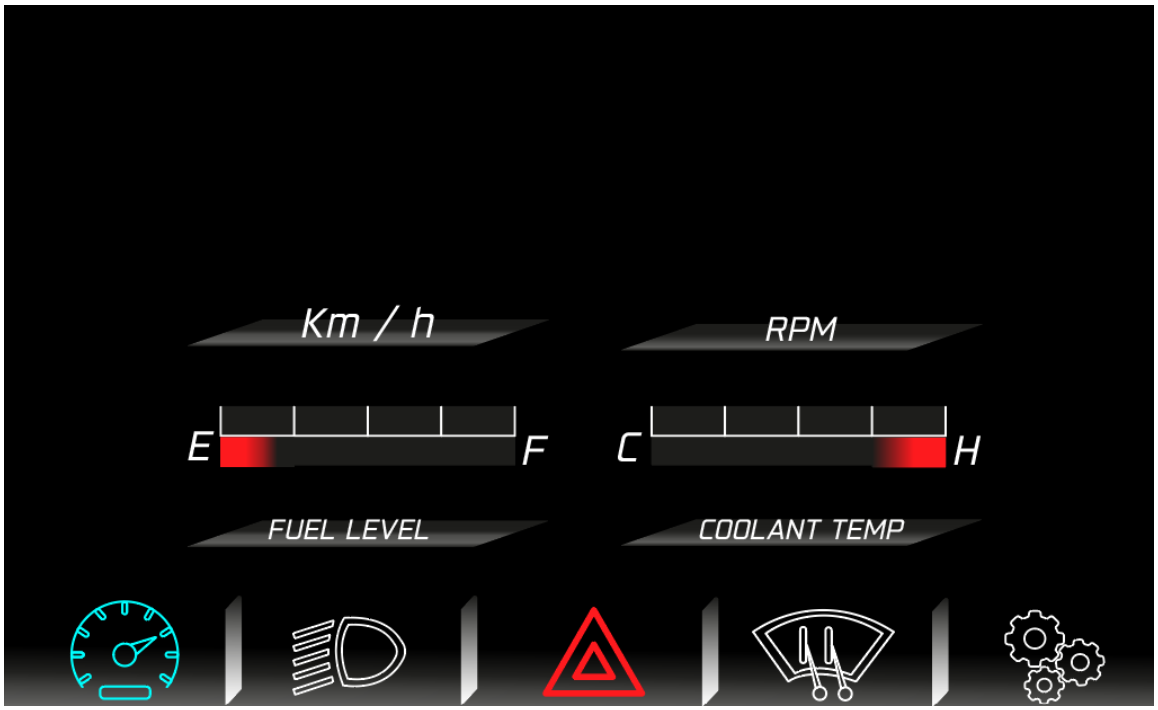
Diseño de botones táctiles

Nota. Este diseño se realizó con el motivo de que la pantalla tenga diferentes menús, por ello se aprecia 5 imágenes referenciales, las cuales son: datos en vivo, activación de luces, luces de emergencia, activación limpiaparabrisas y configuraciones.

Al tener el resto de área libre, se dispone los indicadores de velocidad, revoluciones por minuto que se programarán para que entreguen un valor digital, en cambio el nivel de combustible, y temperatura del refrigerante se verá como una barra horizontal que indica cuanto es el nivel, divido en rangos de 25% para conocer niveles máximos y mínimos.

Figura 20

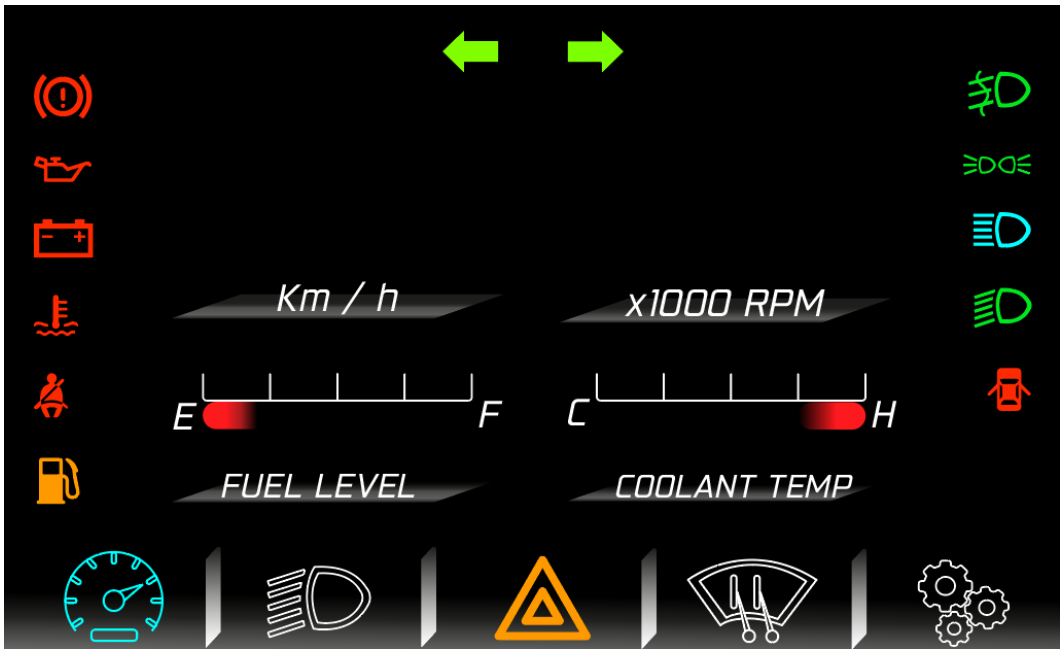
Panel de control datos en vivo



Además de lo antes mencionado se pondrá todos los testigos que normalmente tiene un panel de control como, direccionales, freno de mano, temperatura del motor etc. (Figura21)

Figura 21

Panel datos en vivo y testigos de advertencia



Nota. Al diseñar este panel de control, también se añadió varios testigos que el vehículo originalmente no posee y que se implementarán, como el testigo de cinturón de seguridad, el de puertas abiertas, luces de posición y luces de cruce.

Diseño del panel de control para activación de funciones

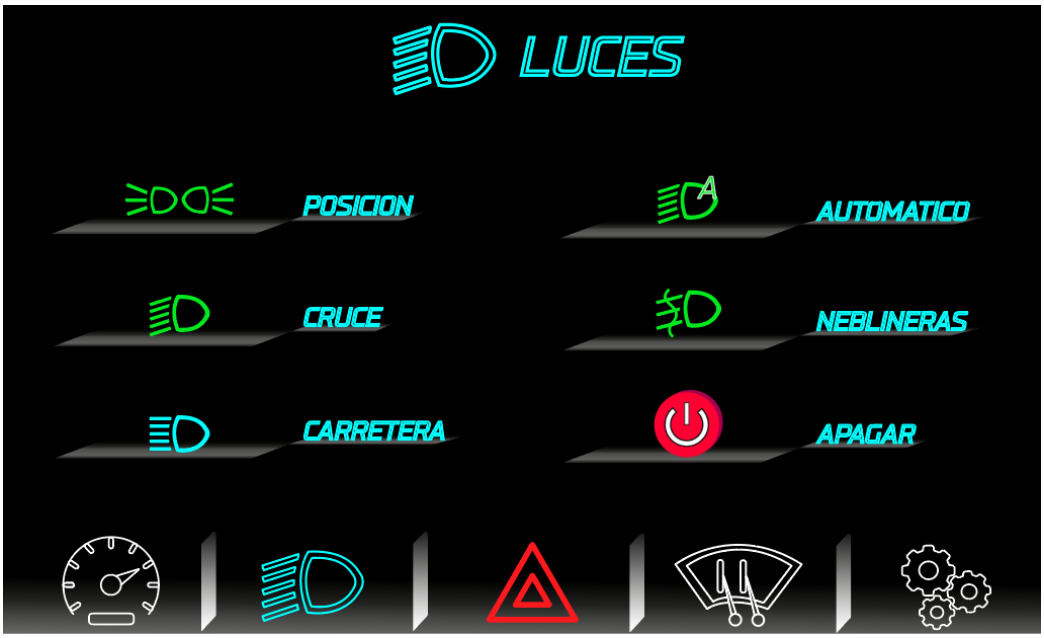
Como se mencionó anteriormente, los demás ilustraciones que se encuentran en la parte inferior de la mesa de trabajo, se utilizará como botones táctiles en la pantalla Nextion, que muestran los diferentes menús de activación de funciones para suplir los botones físicos a botones táctiles de activación.

Menú luces del vehículo

Originalmente el vehículo Chevrolet Gemini, disponía de un botón para cada una de las luces, pero al paso del tiempo ha tenido averías y adaptaciones para que siga funcionando, con esta premisa se ha diseñado un menú único para la activación de las luces, además de añadir una opción de luces automáticas.

Figura 22

Menú luces del vehículo



Nota. Se debe indicar que la activación de luces de carretera permanentemente se encuentra en el mecanismo de luces direccionales, a la izquierda del volante, pero este sufrió una avería al momento de dejar activas (solo permite la acción de destello) y se optó por añadir el botón de activación de dichas luces a este menú.

Menú limpiaparabrisas

El vehículo Chevrolet Gemini, como varios de los modelos japoneses de los años de 1986 hasta 1990, optaron por añadir los botones de activación de limpiaparabrisas al costado derecho del panel de control(ahora se encuentran al costado derecho de los volantes), de este modo realizando un mecanismo específico para activar los botones, provocando que por la antigüedad del vehículo, no se encuentre un repuesto y dejando de utilizar las función de agua para el parabrisas y la velocidad lenta del limpiaparabrisas.

Se procede a diseñar un menú específico para la activación de las funciones de limpiaparabrisas como se observa en la figura 23.

Figura 23

Menú de limpiaparabrisas

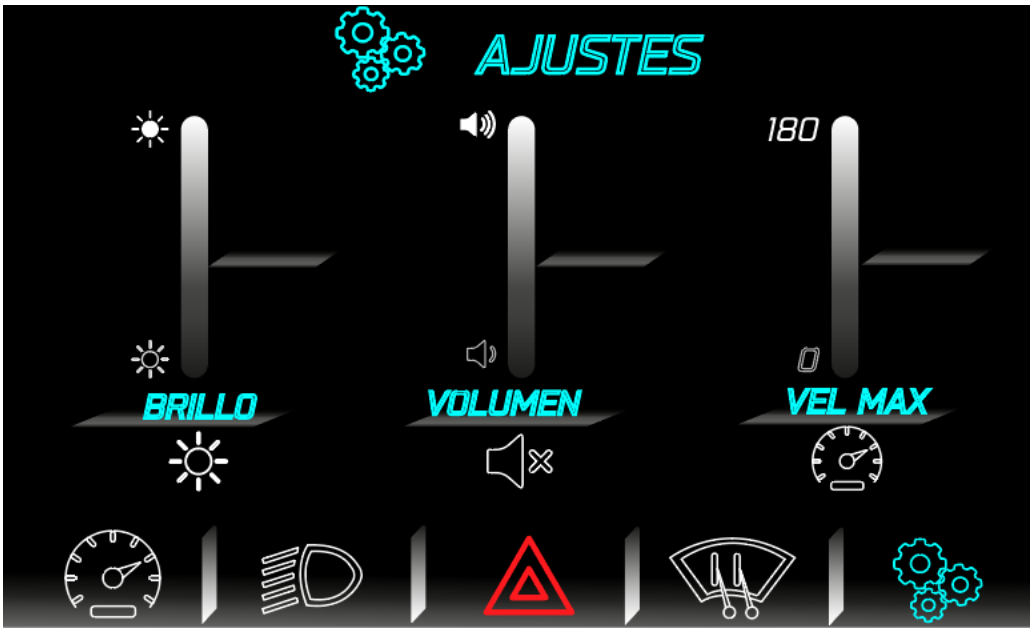
Nota. Además de volver a tener las funciones originales del automotor, se añade la función de encendido intermitente para darle mayor funcionalidad al menú y al vehículo.

Menú ajustes-configuraciones

Con la necesidad de ajustar varios parámetros de la pantalla táctil y el sistema, se diseña un tercer menú (Figura 24) que permita cambiar el nivel de brillo de la pantalla, el nivel de volumen para el parlante que posee la pantalla Nextion y por último el indicador de advertencia de exceso de velocidad.

Figura 24

Menú Ajustes



Nota. El indicador de advertencia de velocidad máxima no limita la velocidad del vehículo, solo está diseñado para marcar cuando la velocidad sobrepase el rango que el ocupante programe, de este modo se presentará el valor que se especifique en el panel de control principal en color rojo.

Programación

Parte fundamental del proyecto de titulación, es la programación en el microcontrolador Arduino y en la pantalla Nextion, ya que depende de varios factores y configuraciones específicas, en ambas que permite que funcione o exista una interconexión para la entrada y salida de datos en cada uno de ellas.

Con la ayuda del Nextion GUI Editor (software de programación de la pantalla Nextion) se puede realizar las acciones que se realizarán dentro de la pantalla sin que intervenga el microcontrolador, es decir la pantalla puede realizar funciones específicas, como bajar el nivel

de brillo de la pantalla, cambiar a otra página, generar mensajes de advertencia o bienvenida, aun si no está conectada con el microcontrolador.

Del mismo modo, la placa Arduino DUE tiene su propio software para programar donde su lenguaje de programación se basa en C++, con lo cual se han aplicado los conocimientos impartidos en la carrera.

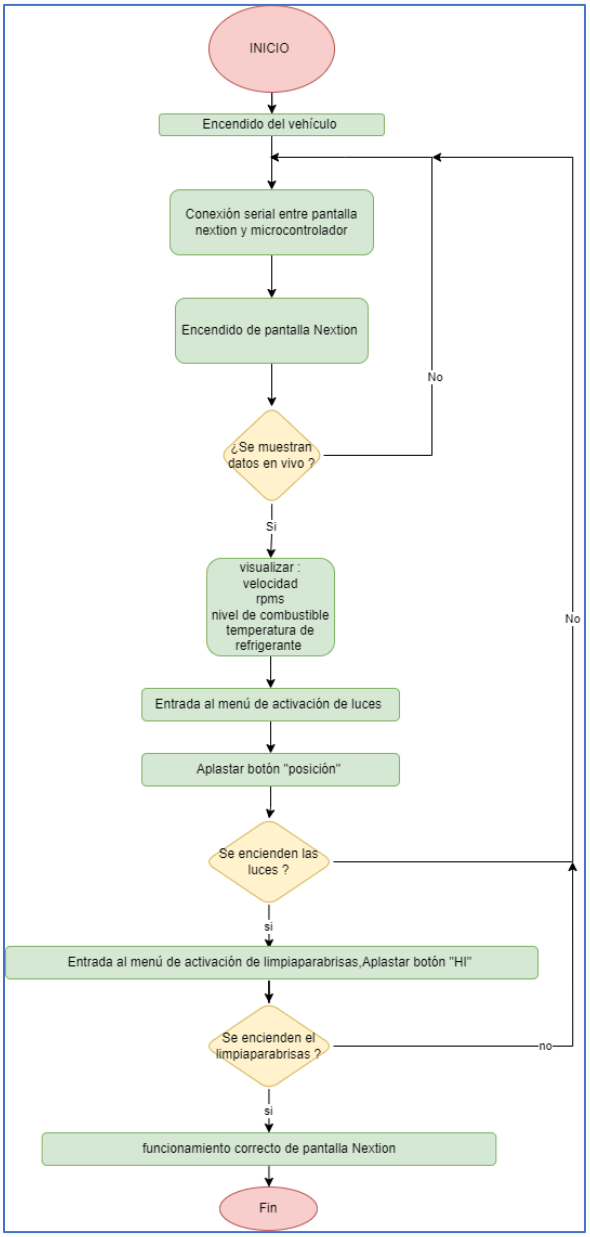
Diagrama de flujo del sistema

El diagrama de flujo indica el camino del código de programación que se realizó, ya sea en la pantalla Nextion o en el Arduino Due para completar una acción, este es usado para ver de forma gráfica lo que muchas veces no se puede entender en forma de código, de esta manera se observa la lógica de programación que se realiza al subir el código ya sea a la pantalla Nextion o placa Arduino.

El diagrama de flujo en el caso de la pantalla Nextion tiene el propósito de señalar el proceso que toma desde el encendido, como se visualiza los datos en vivo, y las funciones que cumple al aplastar los botones digitales(figura 25).

Figura 25

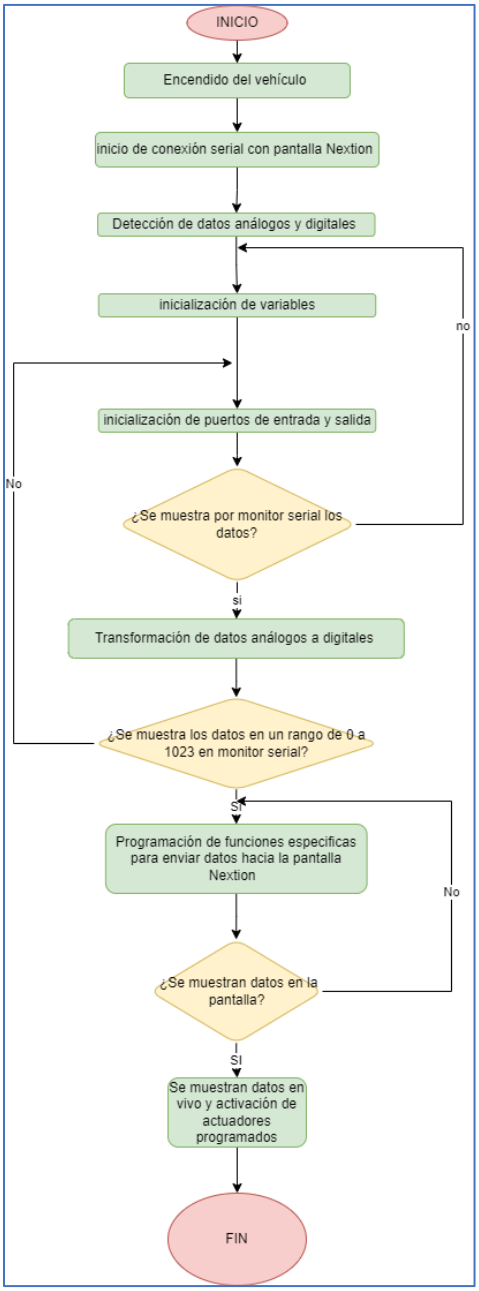
Diagrama de flujo de pantalla Nextion



Por otra parte, el microcontrolador Arduino también posee su código de programación, por ello se necesita de otro diagrama de flujo que represente de forma general cómo funciona de forma individual(figura 26).

Figura 26

Diagrama de flujo Arduino Due



Tabulación de los puertos de entrada y salida

Las entradas y salidas que se utilizan para el desarrollo de tabulación se presentan en la siguiente tabla 3:

Tabla 3

Tabulación puertos I/O Arduino DUE

N	FUNCION CADA PUERTO	NUM PUERTO	I/O	PUERTO
1	INTERRUPTOR ACC	22	***ENTRADA***	DIGITAL
2	INTERRUPTOR ON	23	***ENTRADA***	DIGITAL
3	----------	24	***I/O***	DIGITAL
4	----------	25	***I/O***	DIGITAL
5	INTERRUPTOR FRENO MANO	26	***ENTRADA***	DIGITAL
6	INTERRUPTOR PRESION ACEITE	27	***ENTRADA***	DIGITAL
7	INTERRUPTOR BATERIA	28	***ENTRADA***	DIGITAL
8	INTERRUPTOR LUCES ALTAS	29	***ENTRADA***	DIGITAL
9	INTERRUPTOR CINTURON //INPUT_PULLUP	30	***ENTRADA***	DIGITAL
10	INTERRUPTOR GASOLINA	31	***ENTRADA***	DIGITAL
11	INTERRUPTOR PUERTAS	32	***ENTRADA***	DIGITAL
12	SEÑAL DIGITAL RPMS	33	***ENTRADA***	DIGITAL
13	SEÑAL DIGITAL VELOCIDAD	34	***ENTRADA***	DIGITAL
14	----------	35	***I/O***	DIGITAL
15	----------	36	***I/O***	DIGITAL
16	ACTIVACION MOTOR DEPOSITO AGUA	37	***SALIDA***	DIGITAL
17	ACTIVACION LIMPIAPARABRISAS HI	38	***SALIDA***	DIGITAL
18	ACTIVACION LIMPIAPARABRISAS LOW	39	***SALIDA***	DIGITAL
19	ACTIVACION LUZ POSICION	40	***SALIDA***	DIGITAL
20	ACTIVACION LUZ CRUCE	41	***SALIDA***	DIGITAL
21	ACTIVACION LUZ CARRETERA	42	***SALIDA***	DIGITAL
22	ACTIVACION LUZ NEBLINERA	43	***SALIDA***	DIGITAL
23	ACTIVACION LUZ DIRECCIONAL DERECHA	44	***SALIDA***	DIGITAL
24	ACTICACION LUZ DRL DERECHA	45	***SALIDA***	DIGITAL
25	ACTIVACION LUZ DIRECCIONAL IZQUIERDA	46	***SALIDA***	DIGITAL
26	ACTICACION LUZ DRL IZQUIERDA	47	***SALIDA***	DIGITAL
27	----------	48	***I/O***	DIGITAL
28	----------	49	***I/O***	DIGITAL
29	----------	50	***I/O***	DIGITAL
30	----------	51	***I/O***	DIGITAL
31	----------	52	***I/O***	DIGITAL
32	----------	53	***I/O***	DIGITAL
33	SENSOR DE LUZ	A0	***ENTRADA***	ANALOGICA
34	SENSOR NIVEL COMBUSTIBLE	A2	***ENTRADA***	ANALOGICA
35	SENSOR TEMPERATURA MOTOR	A4	***ENTRADA***	ANALOGICA

Nota. Arduino DUE, dispone de 54 puertos digitales de entrada y salida, y 12 entradas para señales análogas. A diferencia de otras placas Arduino, la placa DUE se caracteriza por recibir y entregar como máximo una tensión de 3,3V, por ello es importante tomar en cuenta esta característica ya que se puede omitir y se puede producir daños a la placa.

Programación del algoritmo en Nextion GUI Editor

El software desarrollado por el fabricante de las pantallas Nextion, permite realizar funciones específicas que solo se controla en la pantalla.

Nextion GUI Editor, permite el desarrollo del proyecto y las funciones propias que se quieran dar a la pantalla táctil, esto es añadir botones táctiles, campos de texto, imágenes, barras de progreso y algunas otras que se enunciaran para el desarrollo del proyecto. (Figura 27)

Figura 27

Pantalla principal Nextion Editor

Nextion editor tiene como característica, manejar un lenguaje de programación simple y basado en las funciones básicas, como las condiciones “if”,”for”,’while’, de este modo se verá ejemplos aplicados en cada una de las páginas y botones que estarán en el desarrollo del tema.

Para comenzar el software en su pantalla principal, posee diferentes campos donde el desarrollador observa distintas funciones y características con las que se guía, de este modo se puede modificar, bloquear, cambiar de forma, color, nombre, etc.

La sección “DISPLAY”, (figura 28) se genera con las medidas predeterminadas como se explicó anteriormente, por lo tanto , es un área de 800x480 pixeles donde a diferencia del desarrollo vectorial, aquí se dispondrá de varias herramientas encima de la imagen para que tenga una funcionalidad especifica, pues en Adobe Illustrator, solo se diseñó una imagen, mas no tiene la capacidad de tener una acción si no se programa en el editor de Nextion.

Figura 28

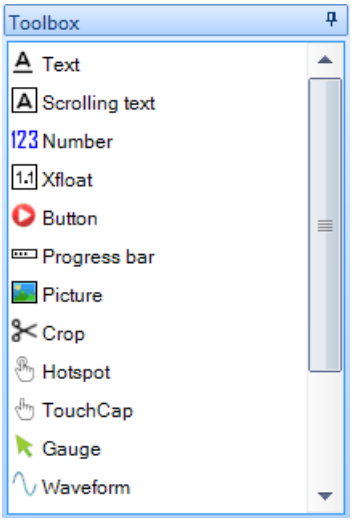
Display Nextion Editor

La caja de herramientas o “Toolbox”, (figura 29) es un listado de diferentes elementos digitales que se dispone para aplicar encima del “Display”, cada uno tiene una aplicación distinta y en el caso del menú principal, se han utilizado las herramientas

- scrollingText (Muestra un mensaje que se mueve de izquierda a derecha)
- number (Utilizado para mostrar la velocidad, es un digito entero)
- Xfloat (Utilizado para mostrar las revoluciones por minuto, es un digito decimal)
- Button (Utilizado para cambiar de página, tiene la similitud como un botón pulsador)
- ProgressBar (Utilizado para mostrar nivel de combustible y temperatura, es una barra de progreso calibrada de 0 a 100)
- Crop (Es un recorte de imagen, utilizado para esconder los testigos de advertencia)
- Dual State Button (Utilizado para las luces de parqueo, tiene la similitud de un botón que se mantiene aplastado, hasta que volver a aplastar)

Figura 29

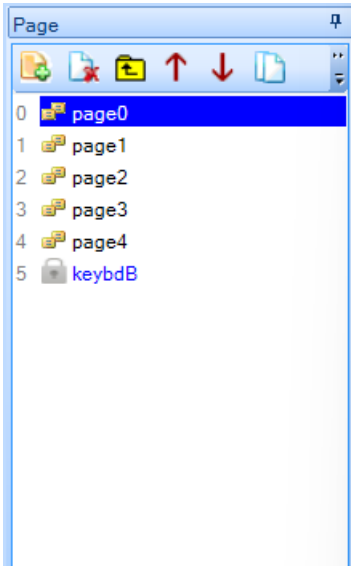
Toolbox



la sección de “Page”, (figura 30) se muestra todas las páginas que se desarrollan y que luego serán cargadas a la pantalla principal, es importante saber cuál página se está editando o desarrollando pues para la programación en el microcontrolador será necesario saber el número de la página.

Figura 30

Page Nextion Editor

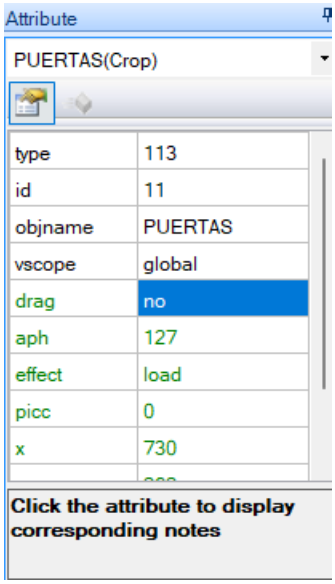


La sección “Attribute”, o atributos informa a la persona diferentes características que tienen la finalidad de modificarse si es el caso y otras que no se tiene permitido realizar cambios.

La parte fundamental de la sección atributo es conocer el “id”, representado en un único número, este se enumera automáticamente de acuerdo con las herramientas que se dispongan en el “Display”, el “objname” es utilizado para cambiar el nombre al atributo de acuerdo a la necesidad o lógica que el desarrollador crea pertinente y “vscope” se utiliza para que el atributo se pueda ver en todas las pantallas o solo en la que está programada, la cual tiene dos estados: local o global y se observa en la figura 31.

Figura 31

Attribute Nextion Editor

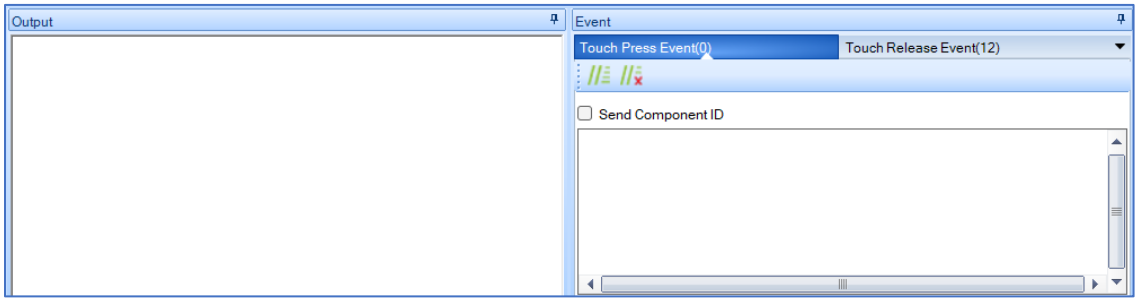


La sección “Output” y “Event” se usa en primer lugar para mostrar si el código ha compilado y evento es donde se programa las líneas de código, están estrechamente relacionadas ya que al momento de compilar y presentar algún error ya sea de tipo numérico o de forma como una letra demás o falta un paréntesis, la sección Output indica en que parte de la sección Event se encuentra el error y de esta manera solucionar la falla para que el código pueda compilar. (Figura 32)

Parte importante de la programación en Nextion Editor, y que se diferencia de la programación en Arduino, es que cada uno de las herramientas utilizadas, como un botón, o una barra de progreso tiene su propio lugar para programar, es decir se empieza un pequeño algoritmo en cada evento de las herramientas, y que tiene una similitud como si estuviera programado en paralelo ,cada elemento tiene su programación y no va ligado a un código general, como si se lo realiza en Arduino, el cual el algoritmo tiene que estar en serie, y cada línea de código es sucesiva.

Figura 32

Output y Event Nextion Editor



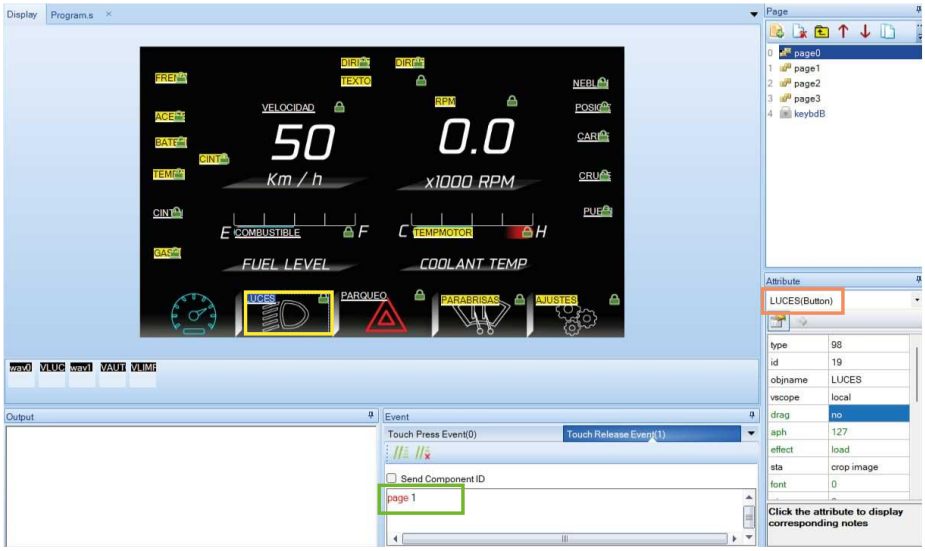
La importancia de la programación en el editor de Nextion, es que se pueda ver reflejado cada línea de código en una acción determinada en la pantalla táctil, por lo tanto cada acción que se verá cuando la pantalla este instalada en el carro, es realizada con una lógica y estructura que se programa con el objetivo de cumplir dicha acción, como se enunciaran a continuación en diferentes líneas de código que se programaron y que si bien cada uno de las herramientas que se utilizaron son distintas , tiene una acción particular, en general esta lógica se repite para las herramientas del mismo tipo, y que se pondrá ver todo el código en el anexo (A).

Button

Para la herramienta botón(figura 33), se utiliza una programación que permite realizar el cambio de página. Como se describió anteriormente, su característica es ser un botón que tiene un solo estado y que al presionar la acción programada se mantendrá activa en el tiempo.

Figura 33

Herramienta Button



Nota. El rectángulo amarillo que se puede apreciar es la herramienta Button, donde se presiona de forma táctil en la pantalla y eso producirá una acción que está ligada a la sección “Event”, donde se observa la línea de código en el rectángulo verde, que indica que haga un salto de página hacia page1, y el rectángulo de color rojo, es que nos indica la característica de la herramienta y como se muestra tiene su nombre y que tipo de herramienta es “LUCES(button)”. Con esta misma lógica se programó a los botones adyacentes como “PARABRISAS” y “AJUSTES”.

Dual-state button

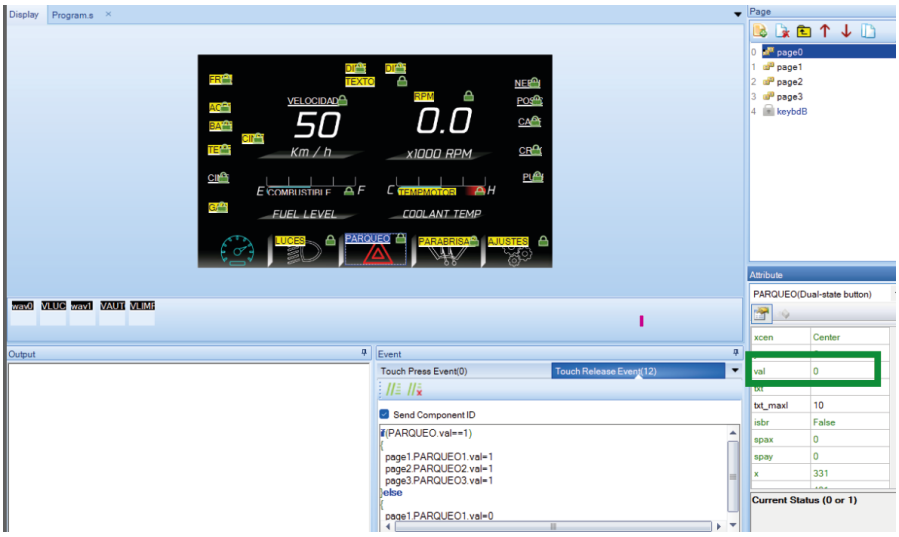
Anteriormente se mencionó que el botón es un símil a un interruptor por ello tiene dos estados o que se conecta y se desconecta, el botón se aplica con dos propósitos, el primero es realizar una acción que se va a ver reflejada o simulada en la pantalla al presionar y la segunda, es enviar el “Component ID”, esto significa que el botón envía una señal al microcontrolador a través de la conexión serial, para indicar que se aplastó el botón y si se vuelve a aplastar indicará otra acción con un simple valor de 1 y 0 respectivamente. El botón

cumple una función específica en cada página donde se añade dicha herramienta; no se permite que un botón sea copiado de página en página, porque puede tener funciones diferentes a menos que el desarrollador realice una programación similar en cada página.

En este caso en específico se utiliza para activar las luces de parqueo o emergencia y simula el botón que se tiene en todos los vehículos y se indica en la figura 34.

Figura 34

Dual-state button

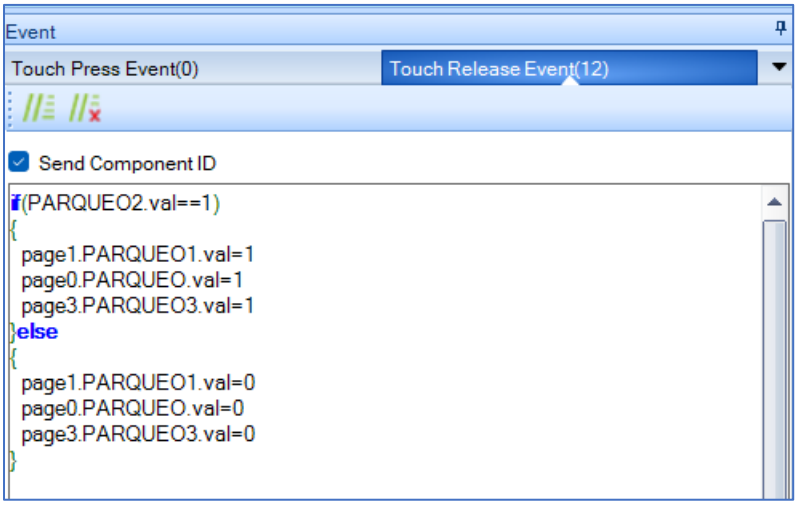


Nota. Se observa en los atributos “PARQUEO (Dual-state button)”, como se identifica la herramienta con su nombre y tipo, de esta manera se manejan todos los componentes en la pantalla, es importante mencionar es que no se puede repetir el nombre en la misma página, pero si tener el mismo nombre en las diferentes páginas que se crean.

Muy importante es hablar de las líneas de código que están presentes en la sección de “Event”, para este caso en específico, se programa la función que debe realizar dentro de la pantalla Nextion, como se observa en la figura 35:

Figura 35

Líneas de código Dual state button

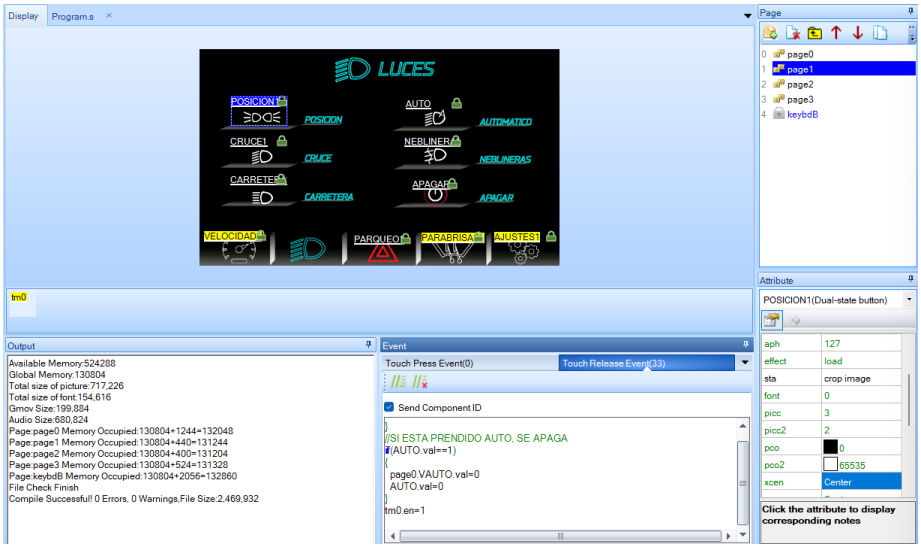


Nota. La programación indica que mediante la condición if, la herramienta dual state button con el nombre PARQUEO2, tiene una característica llamada “val”,(“val” tiene únicamente dos valores, cero y uno) y que al presionar tendrá el valor de uno y si llega a suceder ese evento, realiza las actividades que se muestran, que prácticamente es indicar que los botones que se llaman “PARQUEO1”hasta “PARQUEO3”,actúen como si se los presiona de forma táctil caso contrario los demás botones también permanecerán como si estuvieran apagados o desconectados, pues como se advirtió anteriormente, aunque el button tenga el mismo nombre o propósito, si el desarrollador no programa lo que desea que realice la herramienta, no verá ningún cambio o acción.

Otro de los casos para usar el dual state button, es poder encender las luces del vehículo en este caso las luces de posición, por ello se selecciona la pagina 1, a continuación se selecciona al botón POSICION1 como se muestra en la figura 36:

Figura 36

Botón luces posición



Nota. Como se observa esta es la manera en que se muestra, los atributos de cada herramienta y esta es necesaria para tener como un código de barras que debe ser una dirección único para cada elemento así sea del mismo tipo, sino puede ocurrir errores de interferencia o compilación en la sección de OUTPUT, pero como cada elemento esta con sus propias características, se puede apreciar que existe una compilación exitosa además de 0 errores y 0 advertencias.

La sección de eventos de este botón, ya nos muestra una programación distinta a la anterior, debido a que esta se enfoca en realizar acciones propias cuando se aplasta el botón, pero también de acciones que están realizando en otro botón, por lo tanto, ya no depende que acción quiero realizar con ese botón, sino de las opciones que exista en ese momento en su entorno.

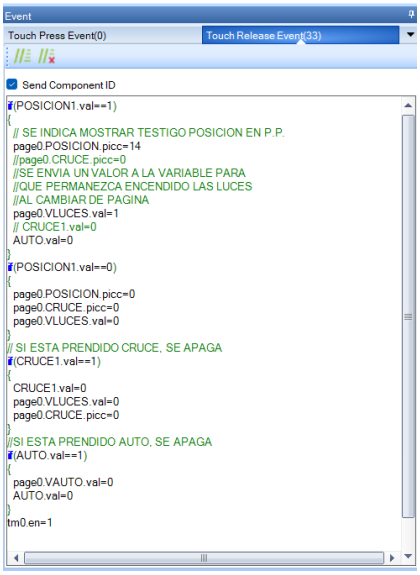
Como se conoce en los vehículos, el mecanismo de activación de luces en algunos vehículos es distinto a otros, como por ejemplo el corsa evolution su interruptor esta fuera del volante, a diferencia de un vehículo muy conocido como el Aveo, que su mecanismo esta

sostenido sobre columna de dirección. A pesar de la ubicación el principio de activación en cualquiera de los dos vehículos es el mismo, y sucede que existen dos situaciones, el mecanismo puede indicar encendido luces de posición o encendido de luces de cruce, pero no puede estar en ambos al mismo tiempo. En efecto lo que se programó en este algoritmo es prácticamente lo mismo, si esta aplastado el botón de luces de posición no puede estar aplastado al mismo tiempo el botón de luces de cruce, y viceversa, por eso se indicaba que esta programación no solo depende de la acción que realice el botón sino de su entorno.

En resumen(figura 37), si un dual state button ya sea de posición o cruce esta presionado y se desea presionar el otro botón, pues se desactivará automáticamente el primero y se mantendrá aplastado el siguiente botón, con esto se evita enviar los dos “Component ID” al mismo tiempo al microcontrolador.

Figura 37

Dual state button luz posición



Nota. En la parte inferior de la palabra Event se muestra “Touch Press Event” y “Touch Release Event”, esto significa que la acción se realice apenas se aplaste el botón o cuando se suelte el botón. En el desarrollo del proyecto siempre se programó en “Touch Release Event”, y

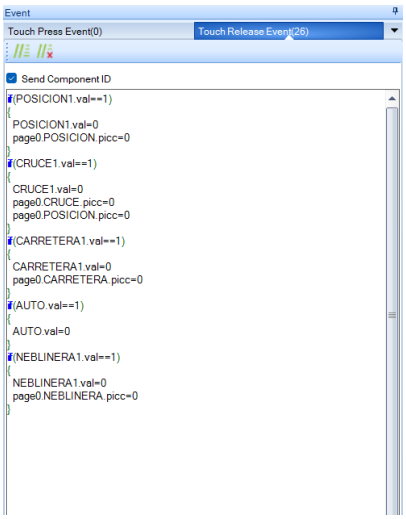
además esta activado el check list de “Send Component ID”, para enviar el dato al microcontrolador.

La programación en el Nextion Editor se basa en darle una acción determinada a cada herramienta, y que esta pueda verse reflejado en la misma pantalla o en el microcontrolador, así es el caso del botón de “APAGAR”, en la página 1.

Cuando el botón “APAGAR”, que es un botón de tipo pulsador pero digital, es presionado, entiende una sola acción a diferencia del botón de doble estado, ejecutando la acción o acciones escritas en el algoritmo pero con un propósito especifico de cambiar el estado de los demás botones de doble estado de la página 1 (figura 38), esto significa que puede que estén presionados todos los botones o solo uno de los diferentes botones(CRUCE,AUTOMATICO,CARRETERA,NEBLINERAS), el código realizará el cambio de estado a todos sin importar cual o cuantos están encendidos, como un conmutador en estado desconectado, con la finalidad de apagar automáticamente las luces sin necesidad de ir apagando uno por uno, facilitando el manejo del sistema de luces(figura 38).

Figura 38

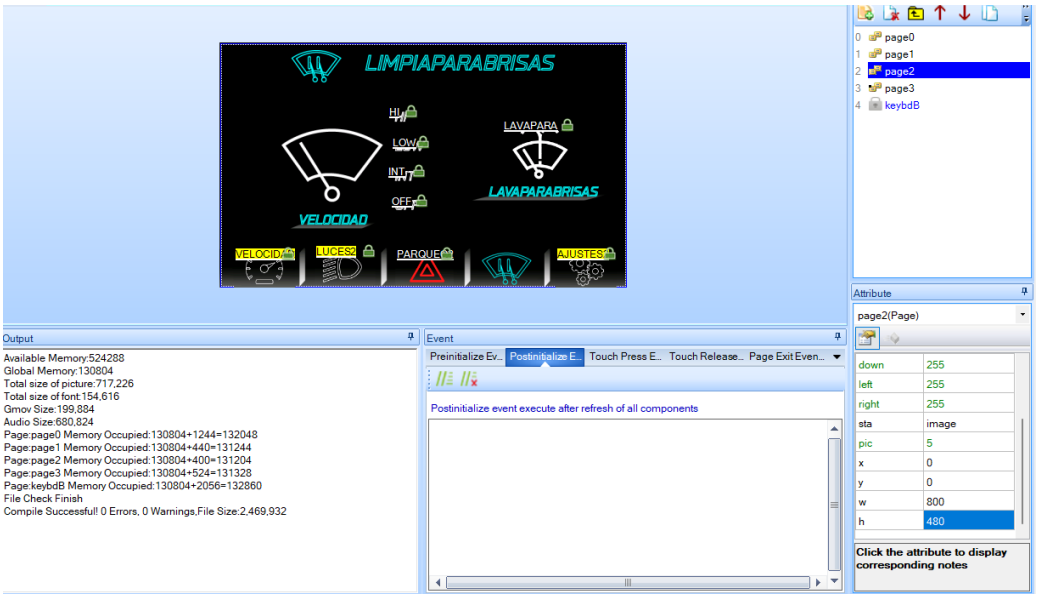
Código de programación Botón APAGAR



Todo lo descrito, anteriormente se usó con el mismo principio en la página 2 del proyecto, es así que los botones HI,LOW,INT, son botones de doble estado y OFF y LAVAPARA, son botones de un estado o pulsadores, de este modo, los botones de dos estados realizarán una activación o desactivación si se encuentra un botón de la misma característica presionado como paso en el botón de luces de cruce, y los botones de OFF y LAVAPARA, realizarán una sola acción ya sea apagar el motor del limpia parabrisas o encender el motor para limpiar el limpia parabrisas de este modo, los botones de dos estados son versátiles ya que se puede realizar más acciones dependiendo de su estado, mientras que los botones normales, ejecutan acciones específicas o una acción especifica de acuerdo a su programación.

Figura 39

Botones en pagina 2 Nextion Editor



Programación del algoritmo en Arduino Due

Después de describir todo el proceso que la pantalla Nextion necesita para realizar funciones específicas dentro de la misma sin que intervenga el microcontrolador, y a su vez lo que necesita el Arduino Due de la pantalla (Component ID) enviado a través del puerto serial al

momento de presionar el botón digital, se realizará el análisis de las funciones específicas para que esa señal enviada al microcontrolador derive en una acción que sea bien, de activación de alguna carga como luz del vehículo o un motor como el de limpia parabrisas y de igual modo la forma en la que el Arduino recibe información en vivo y se devuelve por el puerto serial a la pantalla para mostrar los datos en vivo.

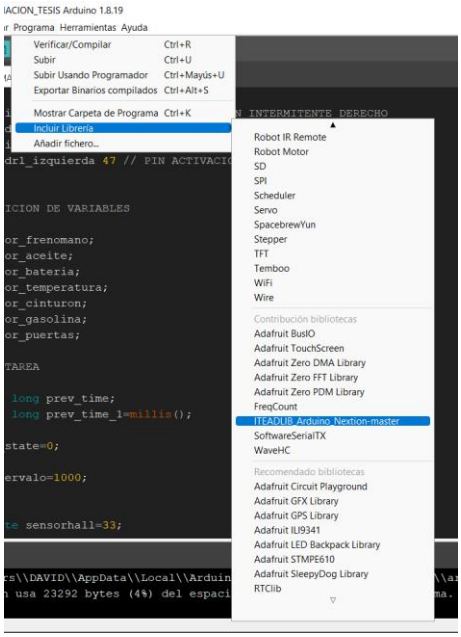
Librería NEXTION

Para la programación en la tarjeta Arduino, no es suficiente en ciertas ocasiones líneas de código que son propias de Arduino, sino depende también de las librerías que son desarrolladas para tareas específicas en la placa, como lectura de temperatura, mover motores paso a paso, audio, etc.

La librería que se utiliza en el código de Arduino para este proyecto es desarrollada por Nextion y su función principal es que se puedan enumerar bajo un formato, todos las herramientas que se añadieron en cada una de las páginas en el software Nextion GUI Editor, ya que sin realizar esta numeración o tabulación de cada una de las partes dispuestas en la pantalla, pues simplemente el microcontrolador toma como si no existe ese componente, y no puede realizar ninguna acción ni de regreso , ni cuando se envía desde la pantalla.

Figura 40

Librería Nextion



La estructura con la cual se debe llamar a cada herramienta que se utiliza es la siguiente:

Tipo_herramienta Nombre_Arduino=***Tipo_herramienta***(N°_pagina,ID,Nombre_Nextion);

Como indica la librería, es la forma correcta de llenar cada una de las herramientas y se puede asemejar como una dirección de un domicilio, donde lo importante es enunciar cada una de sus propiedades como se nombró en la sección correspondiente a “Attribute” de la pantalla Nextion.

El siguiente es un ejemplo semejante como está en el código en Arduino, donde es la primera herramienta en ser añadida a la página 0. Crop es un recorte de imagen que permite habilitar o deshabilitar una imagen en la pantalla dependiendo de varias condiciones las cuales serán programadas. Para el ejemplo se observa que la estructura empieza con la palabra “Nex” acompañado del tipo de herramienta que es, como se indicó en la figura 31, debajo de la

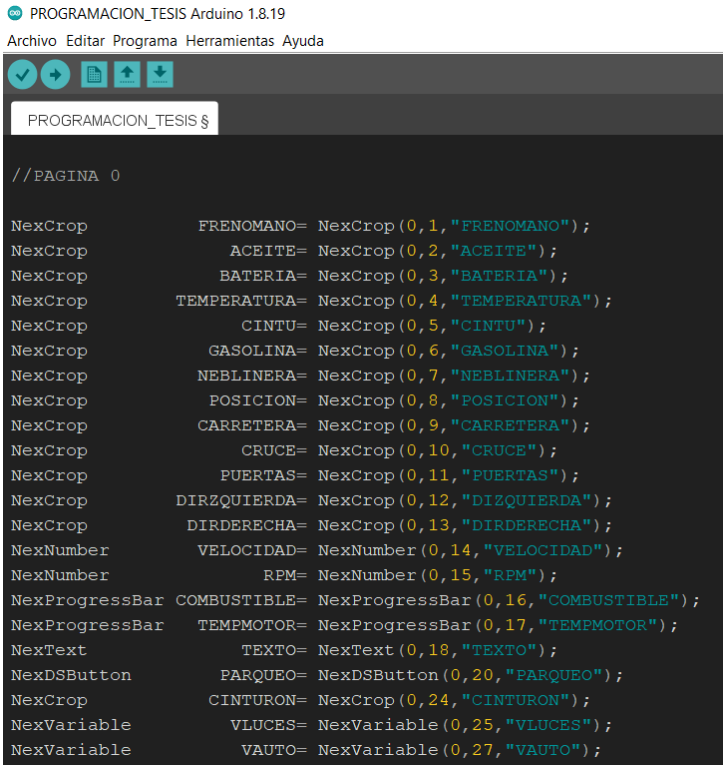
palabra “Attribute”, se observa el nombre de la herramienta y entre paréntesis el tipo de herramienta.

En segundo lugar se indica el nombre con el que se programará en el Arduino este puede ser distinto al nombre que se tiene en la pantalla Nextion ,(se recomienda que sea el mismo nombre para evitar equivocaciones) se acompaña del signo igual y nuevamente se nombra el tipo de herramienta y entre paréntesis se coloca en el número de página(figura 30), se separa a través de una coma y se escribe el ID como indica en la figura 31, nuevamente separada de una coma viene el nombre que es por defecto o el que se haya cambiado y debe estar entre comillas, como está a continuación :

NexCrop FRENOMANO= NexCrop(0,1,"FRENOMANO");

Figura 41

Declaración de herramientas de Nextion en Arduino



se presenta una tabla con todas herramientas que están dispuestas en todas las páginas de la pantalla Nextion:

Tabla 4

Tabulación de herramientas en pantalla Nextion

N°	TIPO DE HERRAMIENTA	ID	OBJNAME	PAGE
1	CROP	1	*FRENOMANO*	0
2	CROP	2	*ACEITE*	0
3	CROP	3	*BATERIA*	0
4	CROP	4	*TEMPERATURA*	0
5	CROP	5	*CINTURON*	0
6	CROP	6	*GASOLINA*	0
7	CROP	7	*NEBLINERA*	0
8	CROP	8	*POSICION*	0
9	CROP	9	*CARRETERA*	0
10	CROP	10	*CRUCE*	0
11	CROP	11	*PUERTAS*	0
12	CROP	12	*DIRIZQUIERDA*	0
13	CROP	13	*DIRDERECHA*	0
14	NUMBER	14	*VELOCIDAD*	0
15	XFLOAT	15	*RPM*	0
16	PROGRESS BAR	16	*COMBUSTIBLE*	0
17	PROGRESS BAR	17	*TEMPMOTOR*	0
18	SCROLLING TEXT	18	*TEXTO*	0
19	BUTTON	19	*LUCES*	0
20	DUAL STATE BUTTON	20	*PARQUEO*	0
21	BUTTON	21	*PARABRISAS*	0
22	BUTTON	22	*AJUSTES*	0
23	BUTTON	1	*VELOCIDAD1*	1
24	DUAL STATE BUTTON	2	*PARQUEO1*	1
25	BUTTON	3	*PARABRISAS1*	1
26	BUTTON	4	*AJUSTES1*	1
27	DUAL STATE BUTTON	5	*POSICION1*	1
28	DUAL STATE BUTTON	6	*CRUCE1*	1
29	DUAL STATE BUTTON	7	*CARRETERA1*	1
30	DUAL STATE BUTTON	8	*AUTO*	1
31	DUAL STATE BUTTON	9	*NEBLINERA1*	1
32	BUTTON	10	*APAGAR*	1
33	BUTTON	1	*VELOCIDAD2*	2
34	BUTTON	2	*LUCES2*	2
35	BUTTON	3	*AJUSTES2*	2
36	DUAL STATE BUTTON	4	*LIMHI*	2

N°	TIPO DE HERRAMIENTA	ID	OBJNAME	PAGE
37	DUAL STATE BUTTON	5	*LIMLOW*	2
38	DUAL STATE BUTTON	6	*INT*	2
39	BUTTON	7	*OFF*	2
40	BUTTON	8	*LAVAPARA*	2
41	DUAL STATE BUTTON	9	*PARQUEO2*	2
42	BUTTON	1	*VELOCIDAD3*	3
43	BUTTON	2	*LUCES3*	3
44	BUTTON	3	*PARABRISAS3*	3
45	SLIDER	4	*BRILLO*	3
46	SLIDER	5	*VOLUMEN*	3
47	SLIDER	6	*VELMAX*	3
48	NUMBER	7	*NBRILLO*	3
49	NUMBER	8	*NVOLUEN*	3
50	NUMBER	9	*NVEL*	3
51	DUAL STATE BUTTON	10	*PARQUEO3*	3

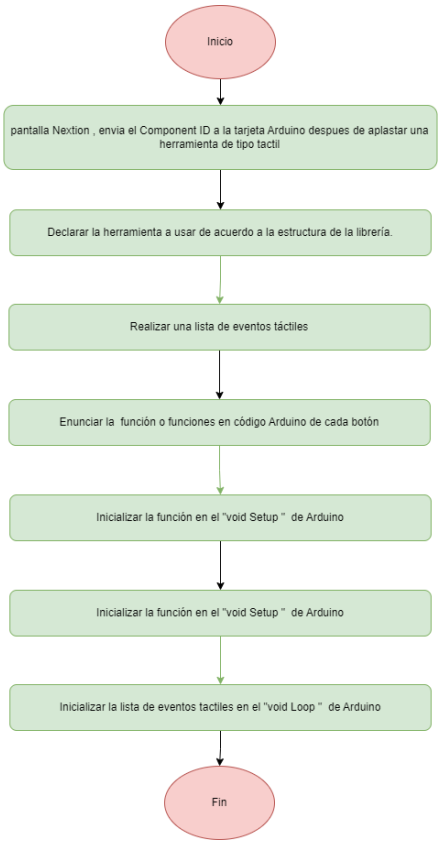
Nota. Como se detalla en la tabla, cada página tiene distintas herramientas con diferente identificación, que no permite interferencias o errores cuando se realiza la lista de herramientas en Arduino (figura 41), debido a que tendrá otro número de ID u otro número de página.

De esta manera es como se inicia el código en Arduino después de haber desarrollado la programación en Nextion Editor. Con esta tabla y la estructura presentada en la figura 41 es suficiente para desarrollar el código, si es que el desarrollador solo pretende enviar información a la pantalla y no realizar una etapa de control, sin embargo, existe una estructura para que una herramienta como Button, Dual-state Button o cualquier elemento que sea táctil realice una acción determinada dentro del código de Arduino.

Los pasos para que un botón digital de la pantalla Nextion realice una acción en Arduino se detalla en la figura 42.

Figura 42

Estructura de programación botón táctil en Arduino



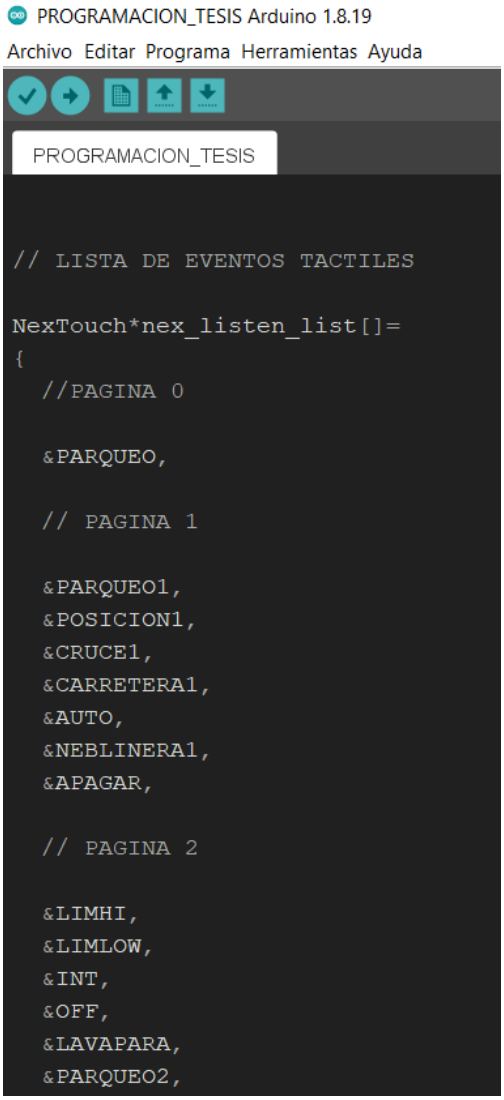
Como se observa en la figura 42, la estructura con la cual se programa una funcionalidad como encender un relé, un motor paso a paso, enviar una señal nuevamente al panel de control es a través de los pasos que se mencionan, y en conjunto con la programación que se lleve a cabo, se logra dar una funcionalidad adecuada a cada botón. Por otra parte, si esta estructura no llega a ser como se indica, tiene una similitud como una compuerta lógica “AND”, que solo funciona si todos los pasos están realizados, caso contrario no tendrá la función correspondiente y es importante nombrar debido a que en el desarrollo del mismo se presentó errores que se pudieron evitar siguiendo estos pasos.

Lista de eventos táctiles

Para continuar con la programación en el software Arduino, se utiliza la lista de eventos táctiles, la cual funciona a través de la librería Nextion. Esta lista permite enunciar todos los botones digitales que se añadieron a la pantalla en el editor de Nextion, sin este paso, pues simplemente el código entiende que no hay un botón, o que no existe un cambio de estado cuando se presiona el botón en la pantalla, de igual manera permite obtener el Component ID para usar a futuro. En la figura 43 se observa cómo está estructurada la lista.

Figura 43

Estructura lista de eventos táctiles



Nota. La lista de eventos táctiles tiene por característica tomar el nombre que se haya dado a la herramienta al momento de declarar como se muestra en la figura 41, y tiene que antecederle el signo “&”, y después del nombre acompañar de una coma, caso contrario marcará un error al momento de compilar el algoritmo.

Programación para herramienta “Button”

La herramienta button como se mencionó anteriormente, tiene una característica la cual es permanecer en un estado o realizar una acción con solo presionar una vez, o de otra manera, se puede presionar N veces, pero la función que va a realizar es una sola.

Esta herramienta se la usa para eventos que son a largo plazo, como por ejemplo dejar enclavado un relé sin importar las situaciones que pasen en el tiempo, esta va a seguir con la función que se determine realizar, así como por ejemplo se realizó para el botón de “APAGAR” del menú de luces (figura 44).

Figura 44

Algoritmo herramienta Button

```
void apagar_luces(void*ptr) {
  digitalWrite(luz_cruce,0);
  digitalWrite(luz_posicion,0);
  digitalWrite(luz_carretera,0);
  digitalWrite(luz_neblinera,0);
}
```

Nota. El botón APAGAR de la página 1 en la pantalla Nextion al momento de presionarlo, realiza las líneas de código que se muestran, este nos indica que todos los botones si alguno de ellos esta presionado, al presionar cambiará el estado en las salidas digitales del Arduino a 0 o a apagado de todas las luces.

Programación para herramienta “Dual-state Button”

La herramienta dual state button puede tener dos funciones, en el caso de los botones que se programaron sirven para activar y desactivar las salidas digitales del Arduino, ya sea para relés, motores, luces o diferentes funciones que el desarrollador desee realizar, también se puede mostrar mensajes de advertencia en la pantalla, permitiendo dar más funciones en un solo botón a diferencia de la herramienta “Button” que se necesitaría de dos botones para que haga la misma acción que este tipo de elemento, como se muestra a continuación:

Figura 45

Algoritmo herramienta Dual state-Button

```
void luces(){

  uint32_t estadolucesposicion;
  POSICION1.getValue(&estadolucesposicion);
  uint32_t estadolucescruce;
  CRUCE1.getValue(&estadolucescruce);
  uint32_t estadolucescarretera;
  CARRETERA1.getValue(&estadolucescarretera);


   // LUCES DE POSICION
  if(estadolucesposicion==1) {
  digitalWrite(luz_posicion,HIGH);
  }else if(estadolucesposicion==0)
  digitalWrite(luz_posicion,LOW);
  }

}
```

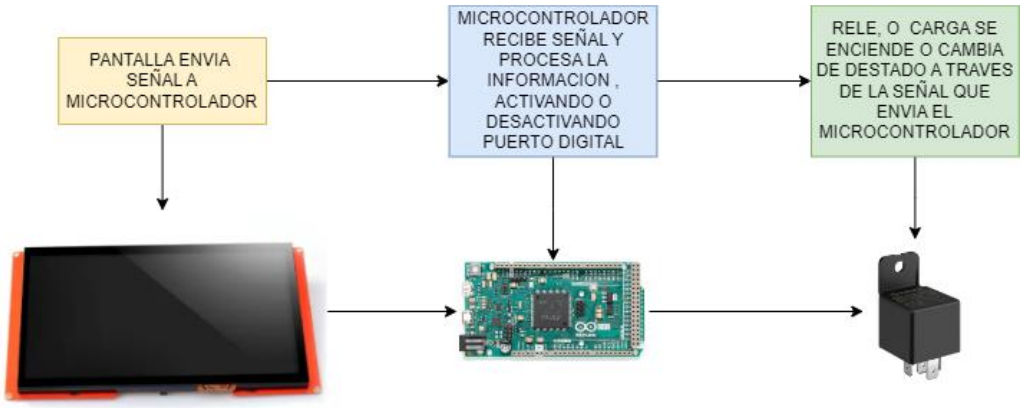
Nota. Para detector la variable que envía la pantalla, la estructura de programación recomienda usar el tipo de variable “uint32_t” para almacenar ese valor en el Arduino pues a pesar de ser un numero entero y que se puede declarar como” int” que será 1 o 0 , el tipo de número que se le atribuye con uint32_t permite que sea un número entero sin signo de 4 bytes.

Lo importante de usar los botones de doble estado es que se tiene que buscar el valor del botón, con el comando “getValue”, como se muestra en la figura 45, primero se llama a la herramienta POSICION1, que es un botón de doble estado, a continuación se pide a esa herramienta su valor para posteriormente almacenar en la variable de tipo uint32_t, antecediendo un “&” para que pueda tomar el valor numérico. Asimismo, se procede a realizar el algoritmo de comparación, donde se observa que, si llega a ser igual a “1” la variable, significa que el botón ha sido presionado y procede a encender, o enviar una señal digital “HIGH”, al puerto de salida “luz_posicion”, en cambio cuando se vuelve a presionar, va a detectar un cambio de variable igual a “0” y procederá, a enviar una señal digital “LOW”, al puerto luz posición. Este procedimiento se realiza para cada uno de los botones de doble estado, ya sea luces de cruce, limpiaparabrisas, etc. Por ello con la aclaración de este tipo de elemento, se sobre entiende cómo funciona el resto de los elementos similares al botón de doble estado.

Por lo tanto los botones digitales realizan una determinada acción, de acuerdo a la estructura de programación y el microcontrolador ejecuta el algoritmo para que este, sea usado para la salida de señales digitales(figura 46).

Figura 46

Activación de cargas a través de Arduino



Detección de señales lógicas

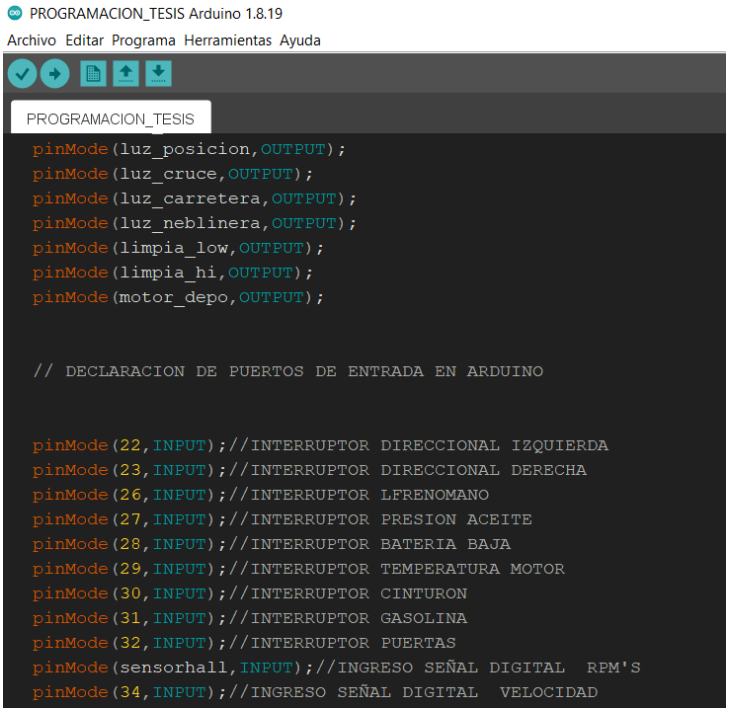
Continuando con la programación en Arduino, es necesario hablar de las señales lógicas que el Arduino detecta del exterior, principalmente cambios de estado ya sea de apertura de puertas, nivel de combustible, posicion del freno de mano, etc.

Anteriormente el Arduino fue utilizado para activar las señales digitales, ahora realizará el efecto contrario, al recibir una señal digital externa y utilizar dicha señal para mostrar una imagen, mensaje o información en la pantalla Nextion.

En primer lugar, se debe declarar la o las entradas en el “void Setup” de cada una de las señales lógicas que se pretenda detectar, pues sin antes anunciar al microcontrolador para que se va a usar ese puerto digital, este no va a entender que función quiere realizar a pesar de que la programación este bien relativamente, pues el algoritmo si compila en el editor de Arduino mas no entiende para que se va a usar dicho puerto si no está declarado (figura 47).

Figura 47

Declaración de puertos de entrada y salida



Nota. Los puertos digitales se declaran tanto como salidas o como entradas y siempre con la estructura determinada por pinMode (#puertodigital,I/O), pues esta estructura ayuda a que se conozca que función va a tener “X” numero de puerto digital, indicándole si es salida “OUTPUT”, o entrada “INPUT”, siempre con mayúsculas ya que si no se remite a la estructura predeterminada el código presenta un error.

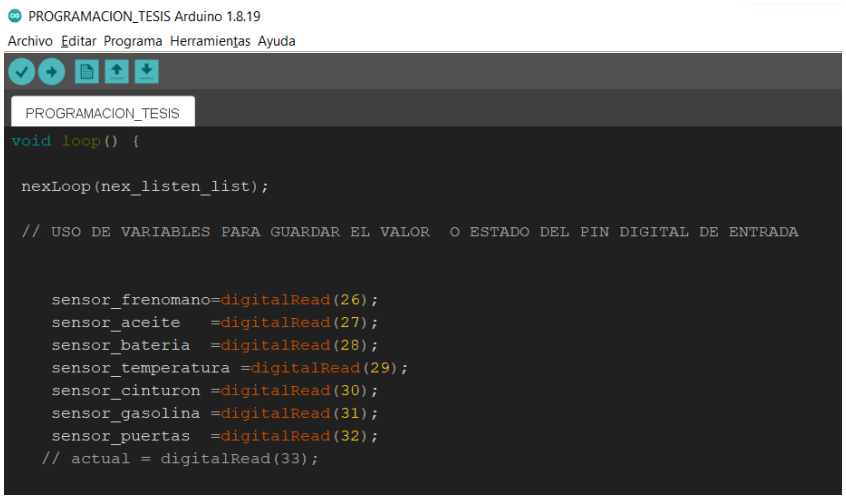
Posteriormente habiendo declarado los puertos digitales que funcionan tendrán, se necesita generar una variable de tipo entera para recibir el dato o cambio de estado lógico, pues el estar encendido un foco y luego apagarlo o una puerta abierta y luego cerrarla, el puerto entiende un cambio de estado que será interpretado a través de una variable y con la línea de código que inicia presentando el nombre de la variable, se acompaña de un igual donde se ingresa con el comando “digitalRead” y entre paréntesis el puerto de entrada que anteriormente se declaró.

Nombre_de_variable=**digitalRead**(#puerto_de_entrada)

Con esta estructura como se observa en la figura 48, se enumeran los diferentes puertos de entrada lógicas al Arduino DUE.

Figura 48

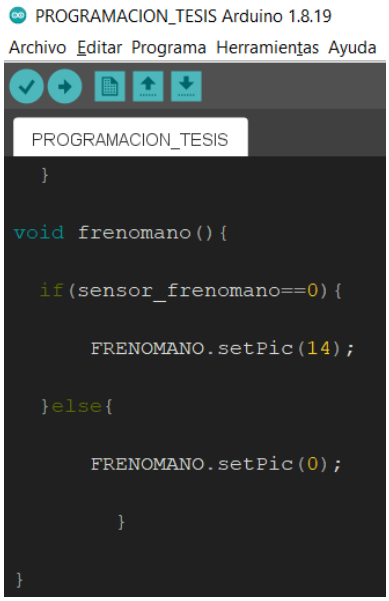
Variables de puertos digitales de entrada



Al seguir este proceso, declarando si el puerto digital será utilizado como entrada o salida, y a continuación tomando como una variable el cambio de estado en el pin digital de entrada, se logra tener una funcionalidad para la pantalla Nextion, pues se habilita una acción determinada a las herramientas de tipo “Crop”, que se tabularon en la tabla 4. La herramienta tipo Crop es un recorte de imagen, de las figuras 20 y 21, donde el código Arduino se programa para que muestre la primera imagen, o la segunda dependiendo el estado que tenga la variable del puerto de entrada digital. En la figura 49 se observa cómo se programa el algoritmo para que en la pantalla Nextion se muestre las figuras antes mencionadas, se utiliza el mismo principio para las diferentes entradas lógicas que tiene el proyecto.

Figura 49

Programación para mostrar cambio de figura en pantalla Nextion



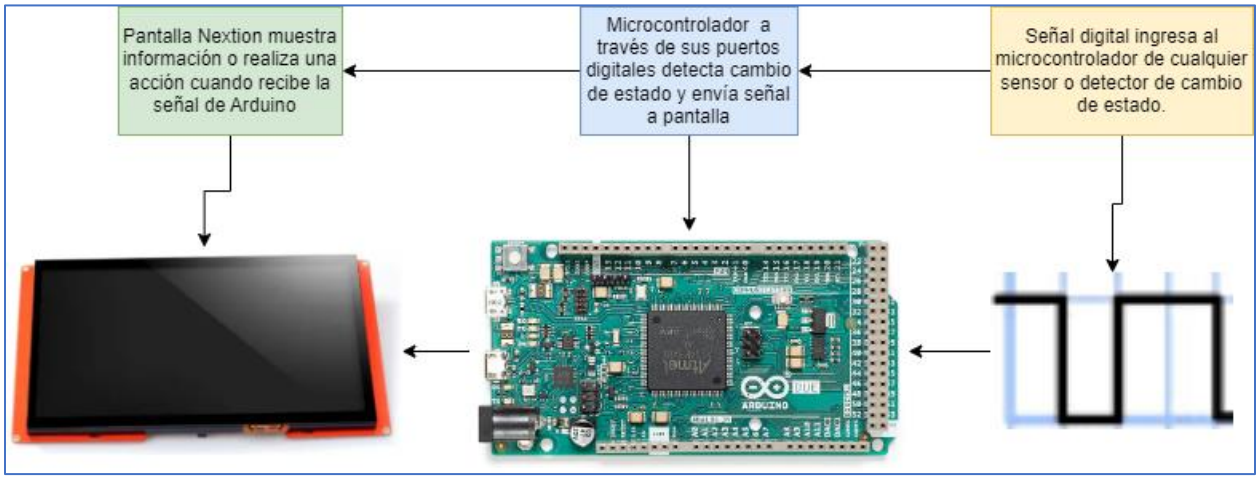
Nota. La programación para mostrar una imagen o figura en la pantalla Nextion, prácticamente se asemeja cuando se tenía al botón de doble estado, pero el botón activa una señal para que se vea reflejado exteriormente en cambio esta señal se verá activada en la pantalla. Lo importante es conocer el nombre de la herramienta ¨Crop¨, a continuación, añadir un punto y

escribir el comando ¨setPic¨, significa que muestra la imagen que está cargada en el Nextion editor con el número 14 y si existe un cambio de estado, muestra la imagen número 0.

Por último, se tiene el diagrama general del funcionamiento del Arduino en este caso como detector de señales lógicas y como realiza el trabajo en pantalla (figura 50).

Figura 50

Activación de pantalla Nextion a través de Arduino



Detección de señales analógicas

Arduino y como muchos microcontroladores, posee la entrada de puertos analógicos y que el microcontrolador trasforma a un valor digital. Un valor analógico se puede tratar del cambio de voltaje en el tiempo, como lo puede realizar un potenciómetro, un termo resistor, un sensor de presión de la atmósfera, y tal y como se presenta si se mide con un multímetro no puede trabajar el Arduino ya que necesita valores reales y no fraccionados, por ello Arduino en su interior tiene un conversor analógico digital de resolución de 10bits esto quiere decir, que puede tomar valores entre cero a 1023 (el valor de 1023 es el resultado de realizar la operación matemática del número dos elevado a la décima) datos en un rango de voltaje. En efecto Arduino al trabajar con un valor de 5V generalmente, tiene que ingresar señales que como máximo sean dicho voltaje así mismo Arduino ofrece únicamente un voltaje de salida de 5V, y

si en el caso de los vehículos que trabajan con 12V (voltaje de batería), se tiene que realizar un divisor de voltaje que prevea este tipo de voltajes que son diferentes al que necesita el Arduino.

Al tener un valor como 1023 y los 5V que utiliza Arduino significa al dividir el voltaje para la resolución ,se obtiene que Arduino puede detectar cambios de voltajes de 4,883 mV, cuando existe ingresa una señal analógica a los puertos del Arduino, y en el caso del Arduino DUE, puede detectar cambios mucho menores ya que al trabajar con un voltaje de 3.3V el cambio que detecta es de 3,222 mV, lo que permite que exista mejor apreciación de los datos de ser el caso que se necesite mayor precisión al momento de receptar los datos.

Para el caso del proyecto se necesita de 3 sensores analógicos, empezando por el sensor de temperatura, utilizando el mismo resistor tipo NTC del vehículo, el sensor de nivel de combustible, que se encuentra en el depósito y un resistor LDR para la aplicación de luces automáticas. En el caso de los dos primeros sensores, son los originales del vehículo, pero debido a su antigüedad los indicadores en el tablero las bobinas que muestran el estado de temperatura y nivel de combustible no funcionan adecuadamente, por ende a través de la programación que presenta a continuación, se logra recuperar la utilidad de los indicadores colocados en la pantalla digital Nextion y en el caso del resistor LDR sirve para la automatización de luces de posición, cruce y carretera a cierto nivel de luminosidad que se presente en el ambiente.

Para el caso del sensor de temperatura, tiene la salida de señal y realiza masa con la parte exterior del mismo, por ello se necesita un divisor de voltaje (en la parte de diseño de circuito se verá que resistencia se utilizó) y alimentado con el voltaje del Arduino DUE, para que este dentro del rango que acepta los puertos analógicos de la placa, además se añade mensajes preventivos y de emergencia que se muestran en la pantalla.

Para el caso de las luces automáticas, se toma el valor del puerto análogo A0, posteriormente se transforma con el comando map de un valor entre 0 a 1023 a un valor que este en el rango de 0 a 3300, que sería el valor de voltaje máximo que alcanzaría si hubiera una condición de luminosidad extrema, pero se realiza esta transformación para poder calibrar en un rango donde se extienda su nivel y de esta manera al código no este definido en valores cortos.

En el caso de las luces automáticas se realizaron 4 rangos para poder activar las luces de posicion, cruce, carretera y modo apagado. Una de las primeras medidas a tomar es que los cambios de luz en el ambiente cuando es en el día no varían entre un valor menor a 50 hasta 1000(los valores se ven en el monitor serial para poder calibrar los rangos), en dicho caso la opción de luces automáticas no tendrá ningún cambio, hasta que ya llega el anochecer, en ese caso se puede tomar que la luminosidad del ambiente varía entre el valor de 1000 a 1700, en este caso los valores se presentaron al caer la tarde en el horario de 6 de la noche en este caso ya se puede encender las luces de posición, a continuación cuando obscureció por medio del monitor serial los valores cambiaron de 1701 a 2500, donde se añade al encendido de las luces de cruce, donde este rango coexistía la noche y las luces que se encuentran en la calle o carretera, y por último, se aplica la última condición donde hay absoluta ausencia de luz, ni luces de las vías, y como este es el caso más extremo, sucede pocas veces, pues activaría las luces de carretera que no es recomendable que pasen encendidas por largo tiempo a menos que sean necesarias, o se esté en una vía con poca luminosidad, de esta manera se evita el destello a conductores que vengan de frente, este último rango se encuentra entre los 2500 y 3300,en este caso se debe llegar hasta el límite del rango debido a que si dejamos un rango menor, puede que ocurra un caso donde el valor sobre pase o este debajo del rango por algún factor en específico como ruido o una condición diferente de luz y el Arduino no va a saber que

acción realizar, por ello todo el rango de valores tiene que programarse en todo el código Arduino para que posea un código sin errores cuando ya esté en marcha el vehículo.

Figura 51

Código luces automáticas

Con el caso del nivel de gasolina, se conoce que el circuito para obtener la señal es a través de la utilización de un potenciómetro que se encuentra en el tanque de combustible del vehículo y a pesar que el indicador análogo no funciona, se comprobó a través de un multímetro que la señal siempre estuvo presente, pues en la vida real el indicador se mostraba como si estuviera el tanque lleno siempre y una de las dudas era si estaba atrancado el mecanismo que varía la resistencia en el tanque o estaba dañado el indicador y en efecto solo esta deshabilitado el indicador permitiendo a través de un circuito divisor de voltaje obtener el valor de la señal necesario para programar en la pantalla Nextion.

En este caso se utilizó el puerto “A2”, para obtener el valor de dicha señal y nuevamente se utilizó la función “Map”, con la diferencia de que el valor que se debe entregar a la pantalla Nextion solo debe estar entre 0 a 100 para poder utilizar la herramienta “NexProgressBar” la cual permite indicar un valor a través de una barra que se desplaza dependiendo de la cantidad indicada para dicho efecto. Se realizó varias condiciones para producir un mensaje en pantalla cuando el nivel de combustible llegue a menos de un cuarto o en el caso un valor menor a 15 hasta el 1 por ciento de combustible, indicando que se encienda el testigo de combustible y un mensaje preventivo para que el ocupante busque una gasolinera.

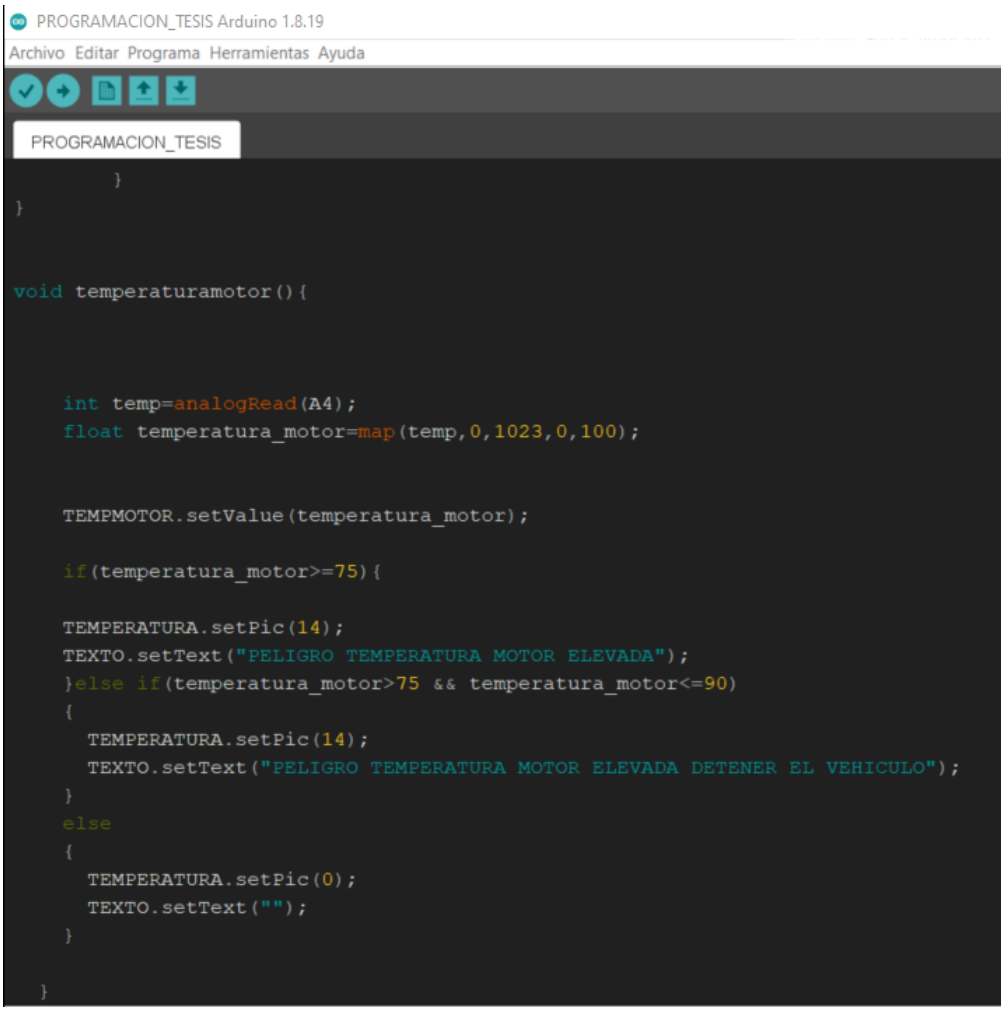
Figura 52

Código programación nivel de combustible

Para finalizar se tiene el nivel de temperatura, para este caso si se necesita de un divisor de voltaje y la señal debe ingresar por el puerto A4 analógico, con el cual al utilizar la función “Map”, obtenemos en un rango de 0 a 100, pues necesita la herramienta barra de progreso el valor ya indicado y con ello calibrar de acuerdo a los datos recibidos y observados en el puerto serial. Nuevamente se tiene una condición para mostrar un mensaje en la pantalla y encender testigo de temperatura, y está expresada en el rango que supere sobre el setenta y cinco por ciento cuando los grados de temperatura supere el rango normal de funcionamiento del motor.

Figura 53

Código de programación de Temperatura del Motor



Limpiaparabrisas

La función para accionar las plumas limpiaparabrisas está dividida en 3 velocidades, pero originalmente el vehículo solo incorporaba dos. La velocidad HI y la velocidad LOW eran las velocidades que vienen por defecto, y se añadió la velocidad o estado INT(intermitente) , la cual se acciona y deja de accionar cada segundo, de este modo se habilita las funciones que anteriormente no se tenía, ya que los botones originales del vehículo se encontraban deteriorados y rotos, por lo cual no se podía activar las velocidades adecuadamente. Para el desarrollo se utilizó 3 botones de doble estado, como se aprecia en la figura 36; el primer botón activa la velocidad HI, a través de un transistor realizando un control de masa, de igual modo el segundo activando la velocidad LOW y por último el botón INT tiene una programación diferente ya que al necesitar el algoritmo de una intermitencia se usó la función “millis()”, la que permite realizar funciones en cierto intervalo de tiempo de acuerdo como necesite el programador sin que el programa o la ejecución del código se vea detenido como realiza la función “delay()”, pues si se utiliza la dicha función, la interconexión entre la pantalla y las funciones a ejecutarse tienden a ralentizarse ocasionando que el programa no presente fluidez al momento de realizar dichas acciones. De este modo con la función millis() se torna el programa con mayor fluidez y las acciones al apretar los botones táctiles, presentan una mejor respuesta sin que este se vea afectado en la realización de las demás funciones.

Figura 54

Código de programación activación limpiaparabrisas

Velocidad y revoluciones por minuto

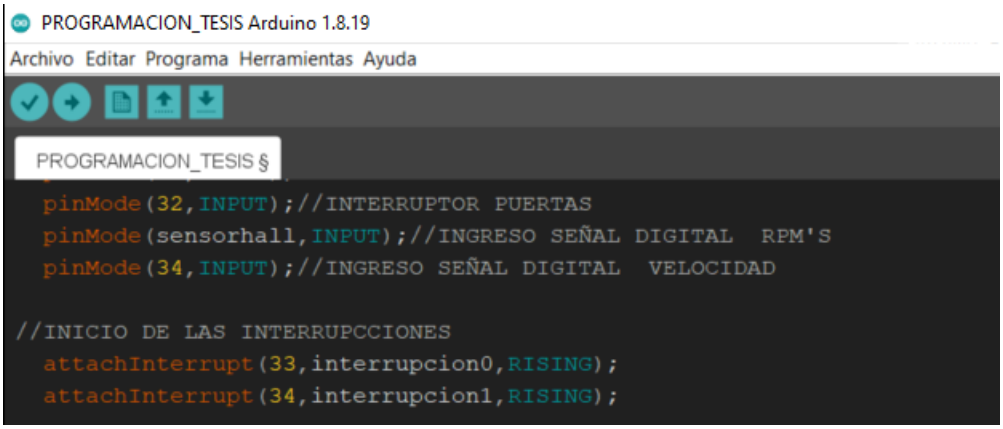
Para las señales lógicas de velocidad y revoluciones por minuto, se utilizó los módulos que la placa Arduino desarrolla para detectar la velocidad de giro, a través de un módulo de efecto hall. Para realizar el conteo de las revoluciones, se utiliza una interrupción que tiene por función obtener el valor de cambios de estado en un segundo, como se necesita las revoluciones por minuto, se multiplica por sesenta obteniendo un valor similar al que se encuentra en el motor.

A diferencia del sensor para la velocidad, este necesita de un valor constante X que se obtiene por experimentación y matematización, pues al depender de los giros que dé el eje de la rueda en un segundo, representará la velocidad en km/h que tenga en ese momento, no se puede realizar una función que interprete la velocidad que existe a través de las marchas

debido a que no se conoce la relación que existe para cada marcha ni un detector de la posicion de marcha como suelen tener autos más actuales, su información está detalla en los manuales del propietario o en las fichas técnicas que los fabricantes entregan a los clientes para informar de las características del vehículo.

Figura 55

Programación código detección de velocidad y rpm



Nota. La estructura para declarar la función de interrupción se estructura iniciando, en el “void Setup” , iniciando con el comando “attachInterrupt”, donde al abrir el paréntesis se dispone en primer lugar del puerto que se desea como interrupción, (en el caso de la tarjeta Arduino DUE, todos los puertos digitales pueden convertirse en interrupción a comparación de la tarjeta UNO que solo dispone de 2 interrupciones), a continuación se escribe el nombre de la función que buscará el código para realizar en cierto intervalo de tiempo además de añadir cuando se desea que la interrupción ocurra, en este caso se colocó la palabra “RISING” la cual detecta que ocurre un cambio de estado cuando la señal lógica pasa de LOW A HIGH.

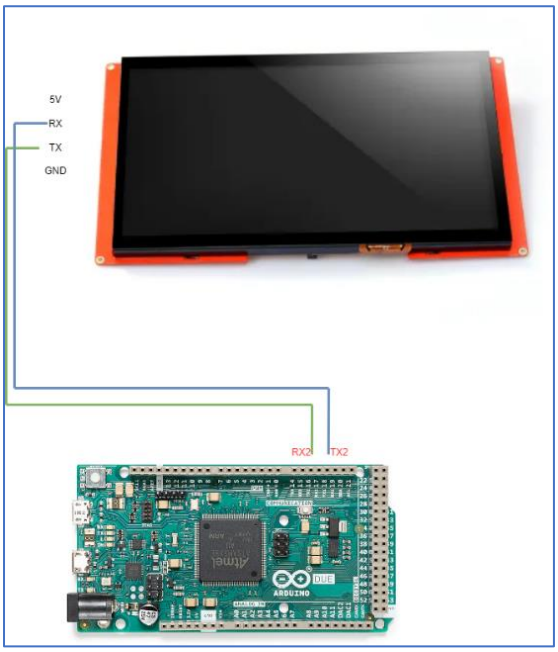
Conexión entre microcontrolador y pantalla táctil

La característica de las pantallas táctiles en los diferentes modelos que se investigó para la realización del proyecto se conforma de dos pines que están enumerados o nombrados como NX y TX, estos se caracterizan por realizar una comunicación serial con el

microcontrolador que se disponga para comunicarse entre ambos. A su vez el microcontrolador posee unos puertos nombrados también con el mismo nombre, para identificar la conexión que se tiene que realizar a fin de que exista una conexión entre la pantalla y la placa Arduino, de este modo el desarrollador no se confunda de pines puesto que no acepta otros pines que no sean los nombrados. Para el caso del proyecto desarrollado y la placa Arduino DUE al tener mayores puertos seriales (esta placa puede añadirse más de 3 conexiones seriales, puede ser bluetooth una de sus aplicaciones) el fabricante recomienda que se utilice en un principio los puertos RX2 y TX2. La forma como tiene que conectarse es de forma inversa entre los puertos de la pantalla y del Arduino, de manera que el puerto TX de la pantalla se conecta con el puerto RX2 del Arduino y el RX con el puerto TX2, de esta manera se genera la conexión serial entre los dispositivos teniendo en cuenta que si no se coloca de esta manera, no existe una comunicación serial adecuada y se podría pensar que está mal programado al no presentar valores en pantalla.

Figura 56

Conexión serial entre Pantalla Nextion y Arduino



Diseño de circuitos electrónicos

Para el diseño eléctrico y electrónico del sistema digital de pantalla táctil con el Arduino, se tuvo que diseñar varios circuitos electrónicos donde se implementen todos los componentes que ayuden a regular, controlar y activar los componentes tanto internos como externos. Mediante la simulación en el software Proteus se realiza todo el circuito eléctronico, el diseño de la PCB (Printed Circuit Board), la simulación en 3D y el plano eléctrico.

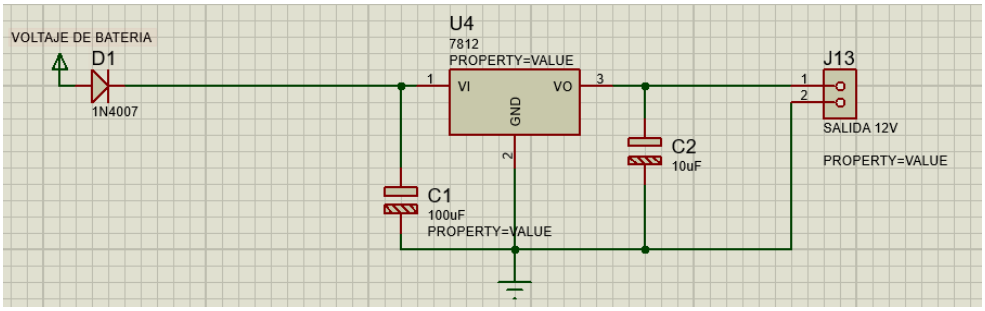
Diseño de circuito regulador de voltaje

Para efecto del circuito electrónico de regulación, se necesitó de 3 fases que comprendía en tener un voltaje constante de 12V, otro de 5V y uno de 3.3V, con la finalidad de tener un voltaje para cada uno de los elementos que se necesita en la aplicación del proyecto.

En fabricante en sus especificaciones técnicas indica o recomienda que el voltaje de alimentación externo cuando el Arduino no esté conectado al computador sea entre el rango de 7 a 12 voltios, con el fin de precautelar el regulador interno que posee el Arduino, evitando que este, se sobrecaliente al recibir mayor voltaje del recomendado sea 12 V o por el contrario que si está por debajo de 7 voltios este no tenga el suficiente voltaje para que sean utilizados los puertos de 5V que tiene el Arduino DUE, de este modo se vería como una falta de potencia de la placa creando una inconsistencia y generando un parpadeo y un reinicio a cada instante si no se encuentra el voltaje en el rango adecuado.

Figura 57

Regulador de voltaje 12V



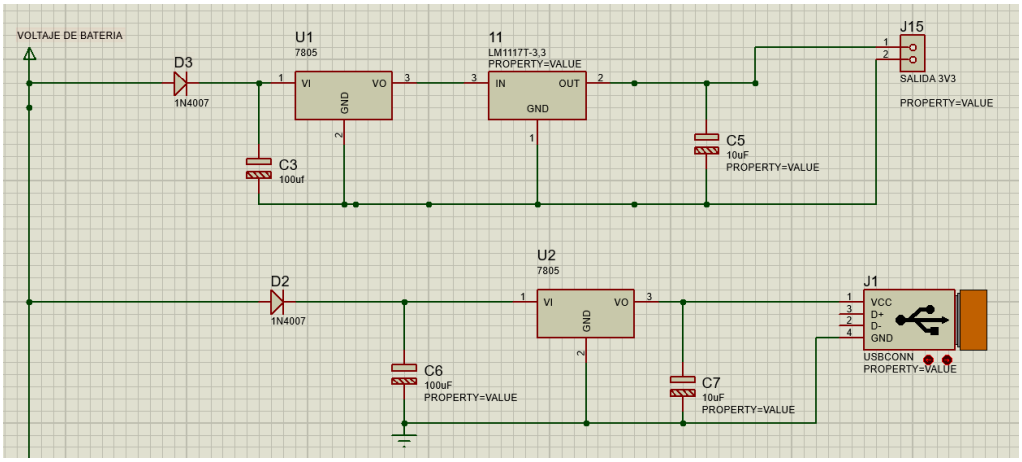
Nota. En el datasheet del componente 7812 recomienda poner dos capacitores electrolíticos o por lo menos el primero a la entrada de voltaje de batería para estabilizar los picos de voltaje que pueden ingresar al regulador, a modo de amortiguador del voltaje que está cambiando constantemente pues el conductor puede estar acelerando o desacelerando y el voltaje de batería estará en un rango que sobre pasa al que recomienda el fabricante Arduino además se incorpora un diodo N4007 para evitar si alguna vez se llega a colocar al revés los bornes de la batería, impidiendo el paso en el sentido contrario de la corriente.

De manera similar se realizó el circuito regulador de voltaje para la salida de 5V y 3.3V, obteniendo el voltaje de alimentación a partir del voltaje de batería y transformando a sus respectivos voltajes con la ayuda de un regulador de voltaje 7805 y un regulador LM1117 para la salida de 3.3 voltios, esto debido a que cada salida necesita de una aplicación indicada para cada uno, por ejemplo se necesita de la salida de cinco voltios para alimentar a la pantalla táctil y la salida de tres punto tres voltios para alimentar a los módulos de detección de velocidad de giro de motor y del eje de la rueda. La consideración con la cual se diseñó las 3 salidas de voltaje con la entrada de voltaje de batería en común es decir en paralelo la fuente de alimentación y no fue en serie, tomando el voltaje de la salida de cada regulador, es principalmente evitar las caídas de corriente de cada regulador, ya que solo permiten 1 amperio de paso en cada regulador, por ende si se toma en serie el voltaje de cada uno las

funciones, la tarjeta la pantalla y la PCB se pueden ver afectadas por la falta de corriente y empezar a fallar y otra de las razones es que se logre más salidas de voltaje si es que se necesita para otra aplicación que se desee implementar, de este modo no estará sobrecargado por ende las fuentes estarán adecuadas para las aplicaciones que necesiten.

Figura 58

Reguladores de Voltaje de 5 y 3.3 voltios



Diseño de circuito electrónico de control

El diseño del circuito electrónico de control tiene dos partes principales, la etapa de recepción de datos y la etapa de activación de actuadores.

Para la primera etapa, el Arduino DUE al necesitar de señales digitales y analógicas que no sobrepase el voltaje de operación, o como el fabricante nombra “OPERATING VOLTAGE” que es 3.3 voltios, se necesita bien de un regulador de voltaje o de un divisor de voltaje. En este caso se usó un divisor de voltaje con la finalidad de obtener la tensión ideal para utilizar en las entradas I/O, sin embargo es importante conocer que los microcontroladores que utilizan voltajes de 3.3 voltios, los valores donde su rango aún puede ser detectado como una señal HIGH, va desde los 2 hasta 3.3 voltios, de este modo se puede realizar un diseño del circuito divisor de voltaje con una tolerancia que evite necesariamente

tener el valor limite, en cambio al diseñar el divisor buscar que la tensión este por debajo de este y que sobrepase el nivel mínimo. Los puertos de Arduino pueden recibir y entregar hasta 40 mA y máximo 3.3 voltios en el caso del Arduino DUE, de esta manera se va a diseñar el circuito divisor de voltaje.

Ecuación 1

Divisor de Voltaje

$$V_{salida} = \left(V_{entrada}\frac{1}{R1 + R2}\right)R2$$

Donde:

Vsalida: representa la tensión de salida que se desea obtener, (V)

Ventrada: Es la tensión de la fuente que alimenta al circuito, (V)

R1: Es el resistor en serie antes del divisor de voltaje, (Ohm)

R2: Es el resistor en serie donde se obtiene el voltaje deseado, (Ohm)

Como se observa en la ecuación 1, para poder obtener el voltaje deseado, se ha dejado en valores de voltaje y resistencia, debido a que se puede despejar o reemplazar porque en un circuito en serie pasa la misma corriente. Del mismo modo se conoce el voltaje de entrada, que idealmente es 12V, el voltaje de salida se podría señalar que es 3.3 V pero en este caso vamos a optar por 3V, para tener cierto margen que permita que si la fuente de voltaje eleva el mismo, este pueda llegar a 3.3V, y para completar se ha dispuesto una resistencia de 100k Ohm con el fin de que la corriente que entre al circuito, este por debajo del límite de 40 mA y por otro lado utilizar resistores que son comerciales, ya que se conoce que no cualquier valor de resistencia que pueda salir en nuestro diseño será exactamente el valor a encontrar en la tienda electrónica.

Se reemplaza los valores en la ecuación 1 y se obtiene:

$$3\ V = \left(12\ V \frac{1}{100k + R2}\right) R2$$

$$300 + 3R2 = 12R2$$

$$9R2 = 300k$$

$$R2 = \frac{300k}{9}$$

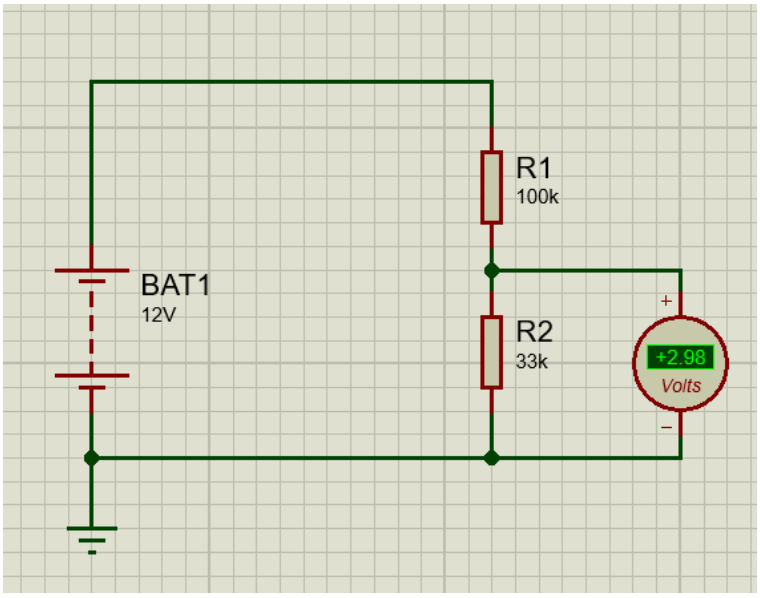
$$R2 = 33{,}33k\ ohm$$

Como resultado se obtiene una resistencia de 33,33 ohm para completar el circuito divisor de voltaje y que, en las tiendas electrónicas, si existe este valor comercial con lo cual no hay que cambiar el valor a otro que se acerque para realizar el diseño.

En la figura 59 se observa cómo está diseñado, las resistencias y voltaje de fuente para encontrar el valor que se pensó desde un inicio.

Figura 59

Circuito divisor de voltaje



De este modo ya se conoce las resistencias y como el voltaje ideal de 12V no va a ser constante, se debe conocer cuál es el valor máximo de voltaje que puede recibir el circuito para que no sobrepase los 3,3V por ello, se reemplaza en la Ecuación 1 con las resistencias y tensión de salida, obteniendo el siguiente resultado:

$$3.3\,V = \left(V_{salida}\,\frac{1}{100k + 33.3k}\right)33.3k$$

$$V_{salida} = 13{,}20\,V$$

De manera que la tensión de entrada para poder detectar un valor lógico, y que el Arduino no sufra sobre tensión ni se queme el puerto de entrada, no debe sobrepasar el valor de 13,20V, todo por debajo de ese voltaje es aceptable y funcionará de acuerdo con lo diseñado.

En segundo lugar, se encuentra la etapa de activación de actuadores, mencionada anteriormente, donde los actuadores poseen un voltaje mínimo de activación de 12V para que funcionen correctamente, se utiliza el concepto de control de masa , que permite activar un actuador o una carga solo con el uso de un transistor, en su etapa de saturación cuando funciona como un interruptor.

Con este fundamento, se aplica el voltaje que entrega las salidas del Arduino a la base del transistor siempre que se encuentre una resistencia entre ambos puertos, así pues colector y emisor actúan como interruptor y la corriente y voltaje de las cargas ya no estarán administradas por el Arduino sino por la fuente de voltaje, es decir la batería del vehículo.

Para ello se utilizará el transistor NPN BC547 el cual es utilizado normalmente para proyectos donde sus características son las siguientes:

Tabla 5

Características Transistor NPN BC547

DESCRIPTION	SYMBOL	TEST CONDITION	TYP	UNITS
COLLECTOR EMISOR SATURATION VOLTAGE	$V_{CE(Sat)}$	Ic=10 mA, Ib=0.5mA	0.09	V
		Ic=10 mA, Ib=5mA	0.2	V
		Ic=100mA,Ib=5mA	0.3	V

Con las características antes mencionadas en la tabla 5 se puede conocer la resistencia que necesita para que el transistor este en saturación.

Primero se debe conocer la ecuación de corriente de saturación del transistor:

Ecuación 2

Saturación de corriente del transistor

$$Ic_{(sat)} = \frac{V_{cc} - V_{CE(sat)}}{R_C}$$

Donde:

Ic(sat): corriente de saturación

Vcc: Voltaje

Vce(sat): voltaje de saturación

Rc: Resistencia

Para una condición de saturación con corriente de colector en 100 mA, se obtendrá el siguiente cálculo

$$Ic_{(sat)} = \frac{12 - 0.3}{1k}$$

$$Ic_{(sat)} = 11.7\ mA$$

Ahora se necesita saber si la corriente de base con la resistencia que se espera diseñar es suficiente para que el transistor este en saturación.

Ecuación 3

Corriente de la base del transistor

$$I_B = \frac{V_{BB} - V_{BE}}{R_B}$$

$$I_B = \frac{3.3 - 0.7}{5k}$$

$$I_B = 0.52\ mA$$

Se debe multiplicar por el factor de amplificación o ganancia de corriente para conocer si en efecto funciona el transistor de acuerdo al diseño planteado.

Ecuación 4

Corriente del Colector

$$I_C = \beta_{CD} I_B$$

$$I_C = 90 * 0.52\ mA$$

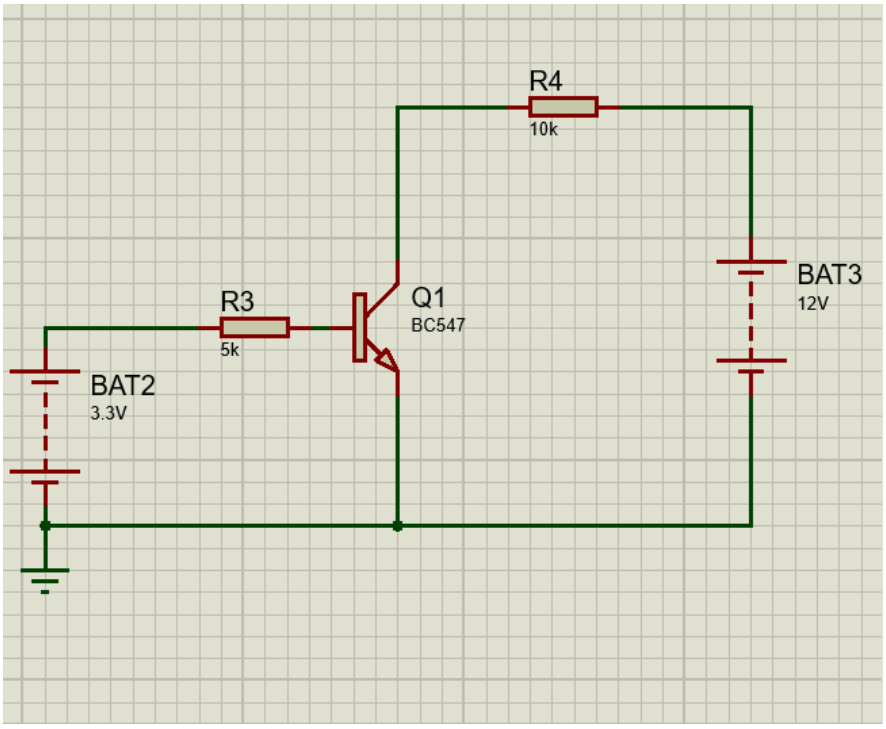
$$I_C = 46.8\ mA$$

Esto quiere decir que con el factor de ganancia de corriente especificado en el datasheet del transistor, esta corriente de base logra producir una corriente de colector “Ic” mayor a la corriente de saturación del colector “Ic(sat)”, por lo tanto, el transistor entra en saturación; esto significa que la corriente del colector no va a subir por encima de su corriente de saturación, así exista mayor corriente en la base, de otro modo siempre que se alcance la

corriente de saturación en la base no importa la cantidad que existe de corriente con el propósito de alcanzar la corriente de saturación.

Figura 60

Transistor en modo saturación



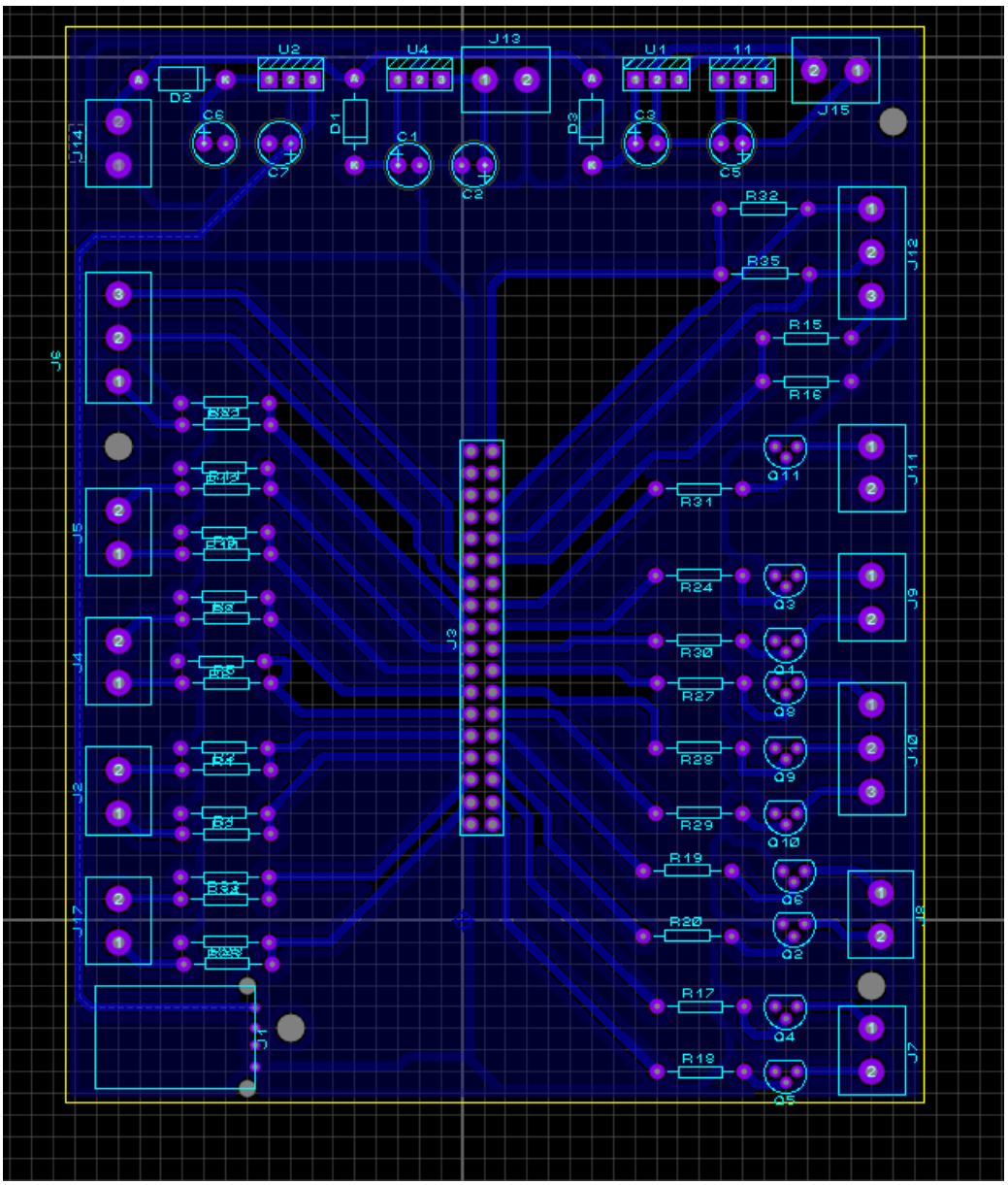
Diseño PCB

Al tener el diseño de la parte de control para activación de cargas, y el diseño del divisor de voltaje para recibir señales, se procedió a realizar una placa PCB, que permita integrar todos los componentes que se adjuntaron, así como los reguladores de voltaje para que funcione perfectamente en cada uno de los casos que se propuso alimentar, de este modo la PCB se caracteriza por tener entradas analógicas, entradas y salidas digitales que realizarán todas las acciones que estaban dispuestas en la programación. La placa está diseñada para que la parte izquierda se encuentren todas las señales digitales de entrada, como es el cinturón de seguridad, freno de mano, las señales de velocidad y RPM's, por la parte superior, todos los reguladores de voltaje , en la parte derecha la parte de control y activación de cargas y relés , y

la entrada de señales analógicas, para que pueda ser conectados mediante el diseño a un conjunto de pines que se asemeja a los pines del Arduino ,de esta manera se puede realizar mediante cables jumpers, la entrega de datos al Arduino.

Figura 61

Diseño de PCB

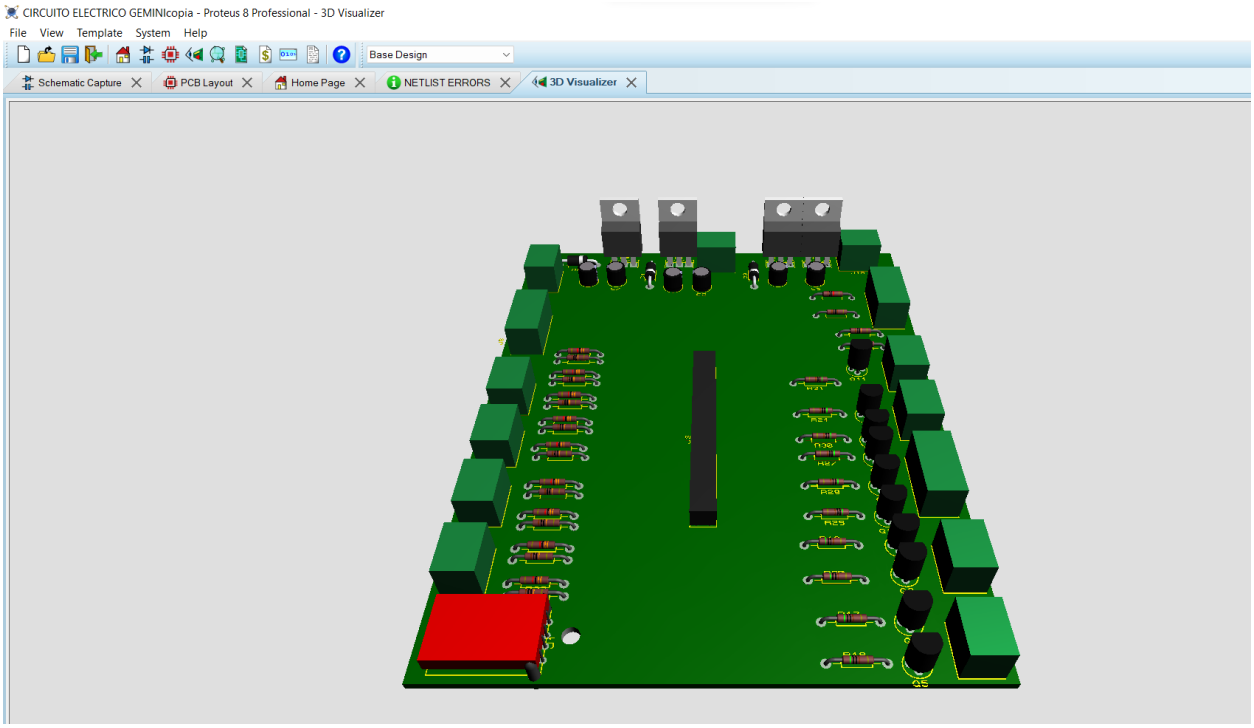


Nota. El diseño de la placa electrónica fue netamente realizado en Proteus, con todas las optimizaciones de espacio y leyes eléctricas para que funcione de acuerdo con lo diseñado en lo teórico, la simulación y por ende tener el funcionamiento requerido para el proyecto.

Otra de las ventajas del software Proteus es la ventaja de poder visualizar el diseño en 3D, para verificar si el sentido, dimensión, elemento y diseño está conforme la necesidad del proyecto.

Figura 62

Diseño de la PCB en 3D



Diseño de circuito electrónico de potencia

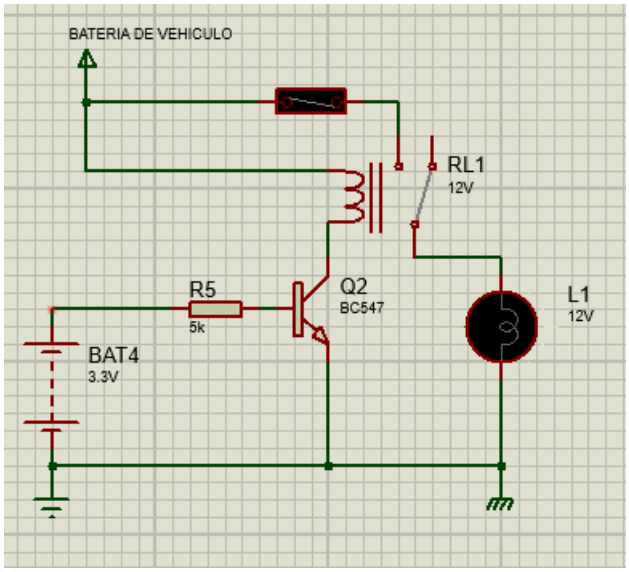
Con el principio del circuito electrónico de control, se basa el circuito de potencia donde se añade un relé para las cargas que necesitan de mayor corriente, voltaje o ambas.

Al realizar el circuito electrónico de control, se tiene que tomar en cuenta la salida del Arduino que realizará la activación del transistor para que este en modo saturación, este a su

vez, realizará el control de masa a la parte de la bobina del relé, permitiendo la activación de las cargas de mayor potencia, de esta manera se reemplaza la parte de batería de 12 V del circuito en la figura 60 con un relé automotriz de la siguiente manera. Los relés se encuentran en la parte exterior del habitáculo del vehículo, por ello en el diseño de la PCB no se añadió en la placa para optimizar el tamaño de este debido a que ocupa espacio y eleva el costo de su fabricación, para ello se reunió a los relés en un elemento que se adquirió para instalar en la parte interior donde se encuentra ubicado el motor.

Figura 63

Circuito electrónico de potencia



Instalación del sistema digital

El sistema digital se divide entre la pantalla digital táctil y la parte de control con el microcontrolador y el PCB, las instalaciones se realizará dentro y fuera del habitáculo del vehículo que se detalla a continuación.

Instalación de pantalla táctil

La pantalla táctil se encuentra dispuesta en la parte lateral del panel de control original, con el fin de que sea accesible para el piloto y copiloto, y esta junta a una base de acrílico que se une al vehículo de forma que este estable la pantalla en el habitáculo, su parte frontal es un borde impreso en 3D lo que ayuda que no ingrese polvo y proteger de los componentes internos.

Figura 64

Instalación de pantalla Nextion

Instalación de microcontrolador

El microcontrolador se instaló en la parte inferior del radio del vehículo para mayor accesibilidad de los componentes y del cableado, de este modo se puede esconder el paquete de cables debajo de la columna de dirección que tienen las conexiones de entrada y salida para el Arduino.

Figura 65

Cableado para las nuevas conexiones dentro del vehículo

A la placa PCB y el microcontrolador se colocó en una caja hecha de acrílico, que se adaptada a la estructura plástica que protege el radio del vehículo.

Figura 66

PCB impresa y soldada componentes eléctricos y electrónicos

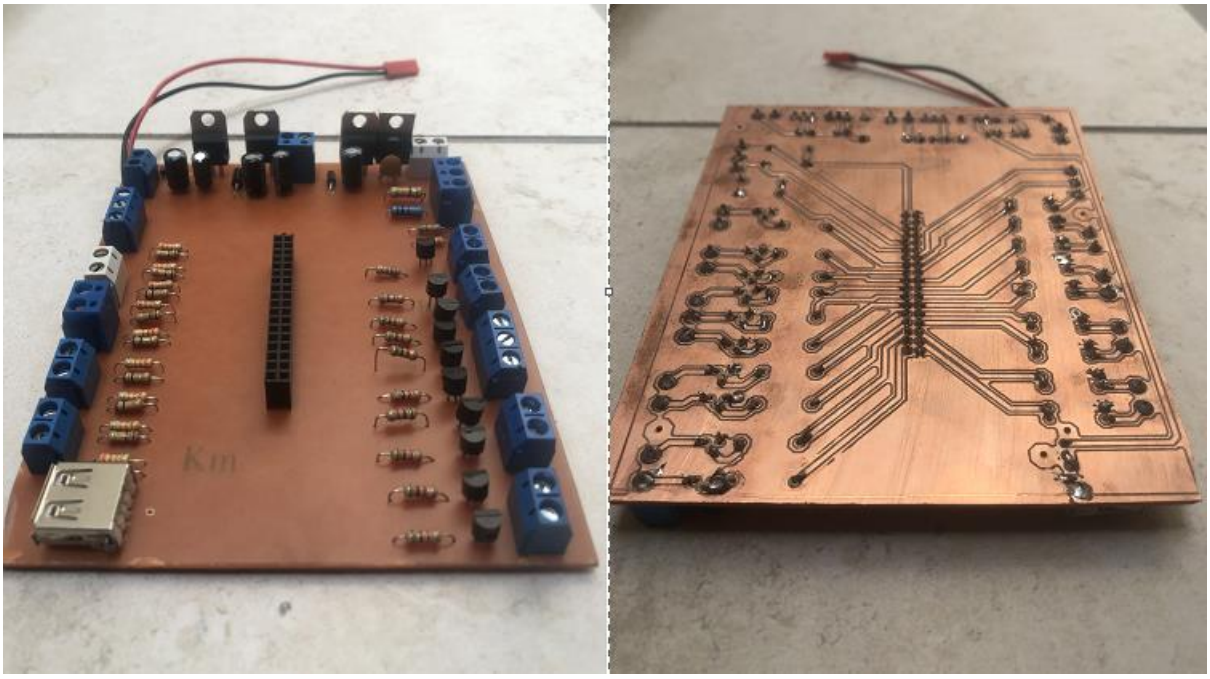


Figura 67

Estructura protección Microcontrolador y PCB

Figura 68

Estructura del microcontrolador instalada en el vehículo.

Instalación de sensores

Los sensores instalados en el vehículo son los ya mencionados de: temperatura, intensidad de luz, velocidad del vehículo y revoluciones por minuto.

El sensor de temperatura se encuentra en la entrada de agua del radiador, después del termostato que ayuda a entregar el valor de temperatura para el panel táctil, pero tiene función independiente del encendido del electroventilador debido a que se activa mediante un termoswitch.

Figura 69

Instalación sensor de temperatura

El sensor de detección de intensidad luminosa se encuentra junto a la pantalla táctil y el panel de control original, mediante una resistencia LDR.

Figura 70

Instalación de resistencia LDR

El módulo Arduino de efecto Hall para detectar las rpms, se colocó en la pared externa de la estructura donde van la banda de accesorios, ya que son las mismas revoluciones del motor.

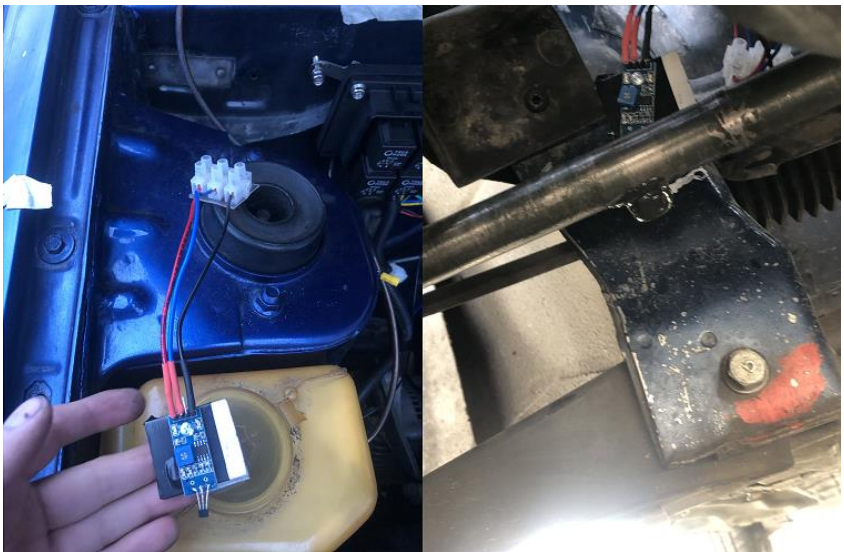
Figura 71

Instalación del módulo de detección de RPM

De igual manera se instaló el módulo de velocidad del vehículo en el eje que sale de la caja de cambio hacia la rueda, y detectar el giro de esta.

Figura 72

Instalación sensor de velocidad

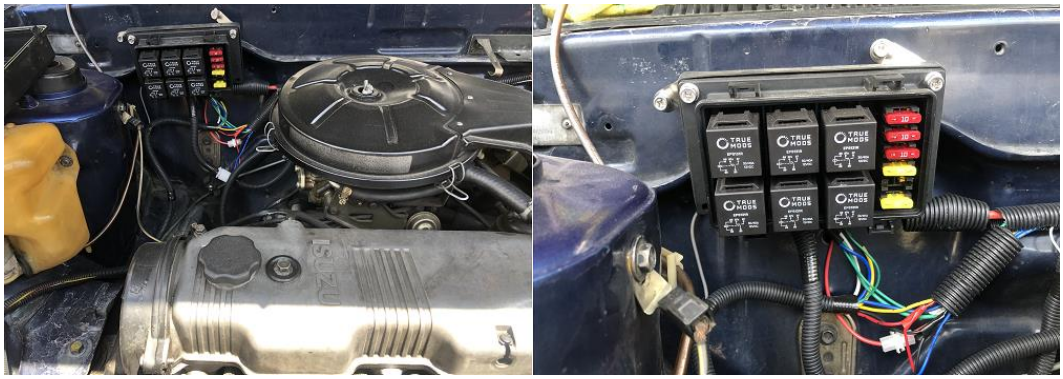


Instalación de actuadores

En el caso de los actuadores a utilizar, se colocó una caja de relés y fusibles en el interior del vehículo, donde se encuentran los componentes mecánicos, para habilitar las luces, y motor de limpiaparabrisas.

Figura 73

Instalación de actuadores

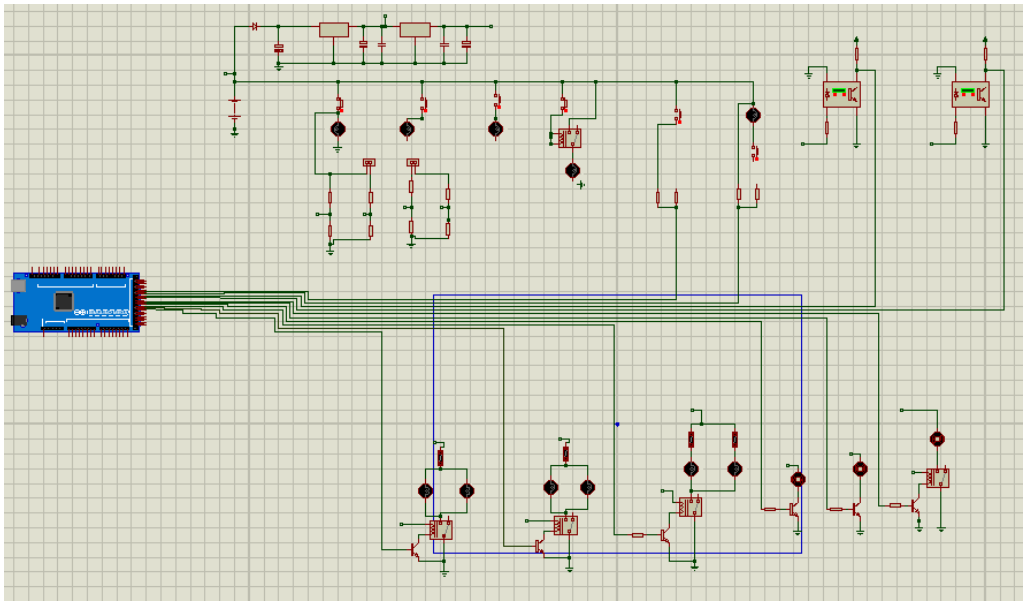


Conexión eléctrica de componentes

La conexión eléctrica de los componentes se expondrá en el anexo B, sobre la conexión y plano eléctrico del vehículo, sin embargo, este es el ejemplo de cómo estar realizado.

Figura 74

Conexión eléctrica de componentes



Capítulo IV

Prueba y análisis de resultados del sistema implementado

Pruebas de funcionamiento en sensores

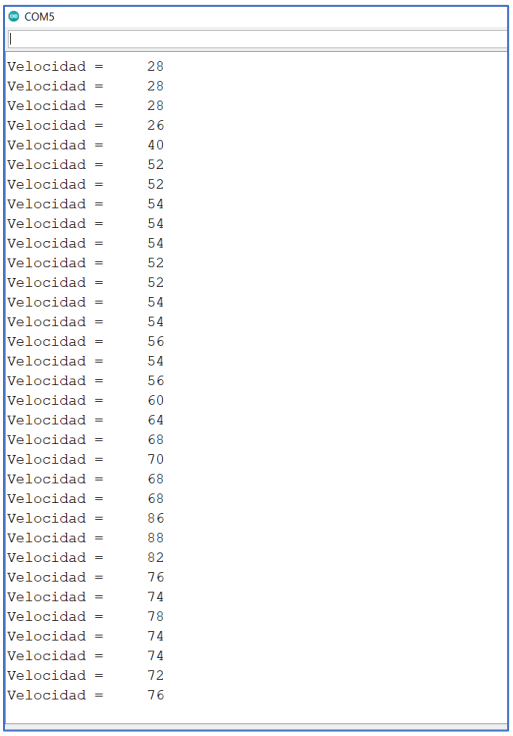
Para las pruebas de funcionamiento de los sensores se detalla la impresión de los datos a través del monitor serial, de acuerdo con la programación indicada para cada sensor.

Funcionamiento sensor de velocidad

El sensor de velocidad tiene una apreciación mínima de 2 km/h y a través del monitor serial presenta los siguientes resultados:

Figura 75

Valor sensor velocidad



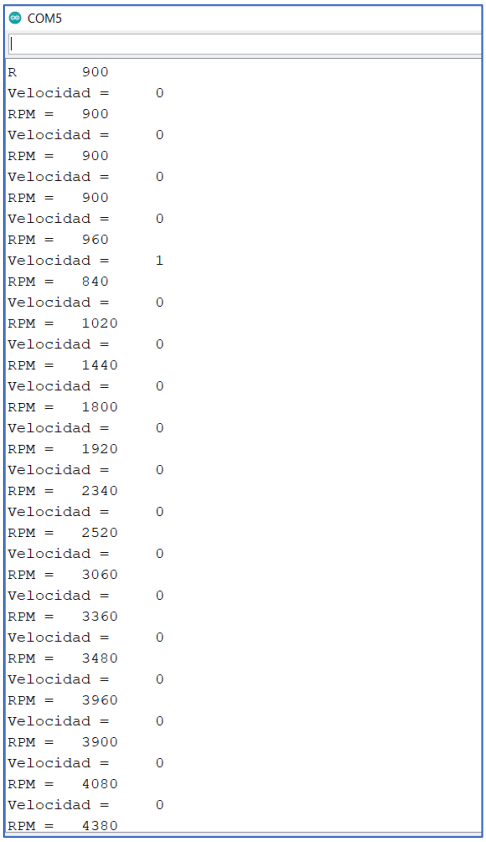
Funcionamiento sensor de revoluciones

El sensor de revoluciones por minuto tiene una apreciación mínima de 1 revolución por segundo y a través de la programación, la sumatoria de revoluciones que dé en un segundo se

multiplica por sesenta para obtener las revoluciones por minuto, y a través del monitor serial presenta los siguientes resultados:

Figura 76

Valores sensor de revoluciones por minuto

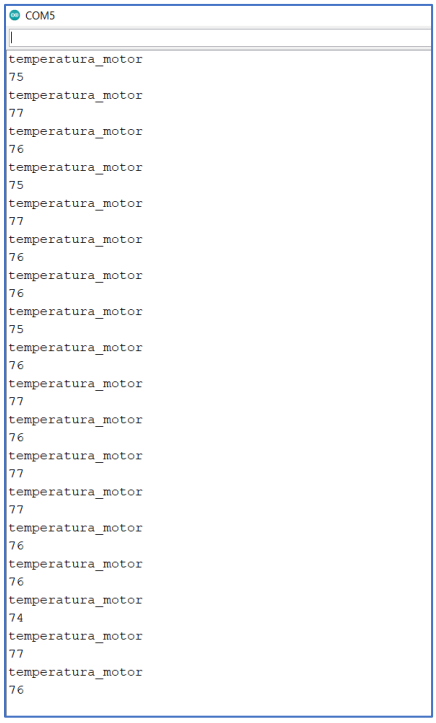


Funcionamiento sensor de temperatura

El sensor de temperatura esta comandado por un voltaje que ingresa al microcontrolador y se transforma en un valor digital, con la función map, como se detalló en el capítulo III. Los valores obtenidos se presentan a continuación:

Figura 77

Valores sensor de temperatura

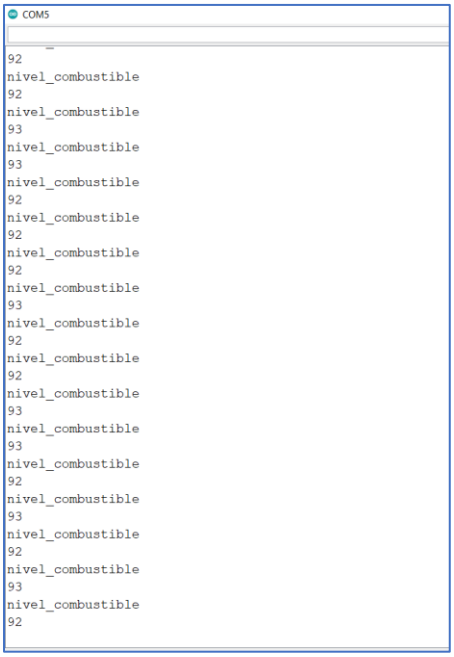


Funcionamiento sensor de nivel de combustible

El sensor de nivel de combustible, obtiene su valor de voltaje a través de la resistencia variable y la posicion que se encuentra la boya dentro del tanque de combustible. En el caso de la tensión cuando el tanque se encuentra lleno, presenta un valor mínimo de 0.50V, mientras sigue consumiéndose el combustible, este sigue aumentando el voltaje, por ello es necesario realizar una variación en la programación para que el porcentaje de combustible no empiece con un valor mínimo y se realiza una resta del 100 por ciento menos el valor que el porcentaje que entrega la función “map” detallado en la programación.

Figura 78

Valores nivel de combustible

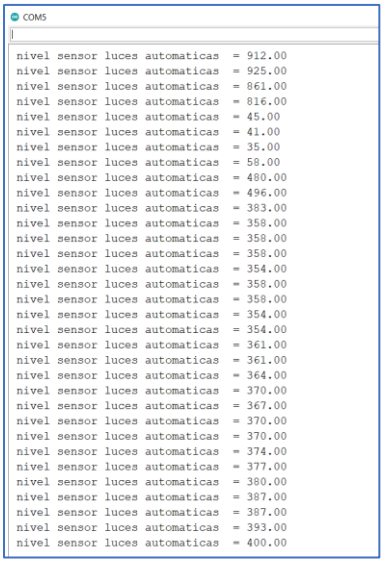


Funcionamiento sensor de luminosidad

El sensor de luminosidad su valor, muestra a través del monitor serial donde se presenta mayor o menor luminosidad, en el caso del sensor cuando existe más luminosidad se presenta un menor valor, mientras menor luminosidad, se entrega un mayor valor en el monitor serial.

Figura 79

Valores sensor de luminosidad



Pruebas de funcionamiento en actuadores

Las pruebas de funcionamiento, en los actuadores son destinadas al encendido de 5 relés y los limpiaparabrisas.

Prueba de funcionamiento en modo automático

El modo automático funciona únicamente, con las luces automáticas, en el caso se comprueba la activación de las luces de posicion, cruce y carretera dependiendo de las condiciones de luminosidad del ambiente.

Figura 80

Activación modo automático

Para la primera condición como se indicó en la programación necesita de poco brillo mas no, ausencia de luz, en ese caso se aprecia como se enciende las luces de posicion con la ayuda de una lampara se puede acerca o alejar dependiendo de la condición que se necesita

Figura 81

Activación automática luces posición

Para la segunda condición en sensor de luz detecta poca luz y está en el rango indicado, enciende las luces de posicion y cruce, en esta ocasión se apagó la linterna que se tenía para la prueba el sensor LDR, consigue el valor para después cumplir las condiciones del código de programación como se observa a continuación en la imagen

Figura 82

Activación automática luces cruce

Por último al existir poca luz ambiental y de la luminaria de la calle el sensor detecta la última condición de la programación, donde enciende luces de posicion, cruce y carretera, esta condición sucede comúnmente en la carretera donde no existe luminaria en la vía, o puede ser en caminos vecinales.

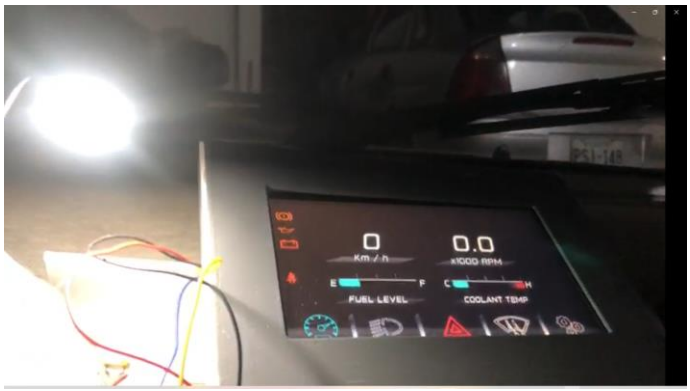
Figura 83

Activación automática luces carretera

Al final se puede observar cómo al acercar la lámpara al sensor, este detecta que existe mucha luminosidad y por ende apaga las tres luces para que se cumpla con la otra condición automática, que el vehículo este apagada las luces, si es que existe mucha luminosidad o simplemente es de día.

Figura 84

Desactivación automática de luces en condiciones de alta luminosidad



Prueba de funcionamiento en modo manual

El modo manual se realiza, con las luces y que estas puedan permanecer encendidas de igual manera con el motor del limpiaparabrisas en sus diferentes velocidades.

El primer caso, se necesita activa las luces de posicion, en el menú de luces, en el hay q señalar el símbolo de luces de posición mas no la palabra, de este modo se activan las luces de posicion tanto delanteras como traseras.

Figura 85

Encendido manual luces de posición

El segundo caso corresponde a las luces de cruce, y nuevamente se repite los pasos con las luces de cruce, carretera y neblineros.

Figura 86

Activación manual luces cruce

Figura 87

Activación manual luces carretera

Figura 88

Activación manual luces neblineras

Para el caso de los limpiaparabrisas se debe ingresar al menú de los limpiaparabrisas donde, se encuentra las distintas velocidades del motor, en este caso se presenta la figura 89 como se presiona la velocidad y se muestra el movimiento, por ello la activación sucede en todas las velocidades, pero no se puede apreciar el movimiento mayor o menor velocidad, por ello se indica una sola imagen.

Figura 89

Activación limpiaparabrisas

Prueba de voltajes y corrientes de consumo

La prueba de voltajes y corrientes, se efectúa encendiendo los actuadores, en la pantalla, y tomando los valores en la caja de relés y fusibles, donde el primer paso es retirar el fusible que le corresponde a cada relé con la ayuda del multímetro, en la opción de medición de amperaje, tomar el dato, y para el voltaje se vuelve a colocar el fusible, se ubica la opción de medición de tensión en corriente continua en el multímetro, seguidamente en uno de los extremos colocar la punta positiva del multímetro y la otra punta en el chasis del vehículo, de esa manera se obtiene el voltaje y los resultados se ubican en la siguiente tabla 6 :

Tabla 6

Prueba de voltajes y corrientes de consumo

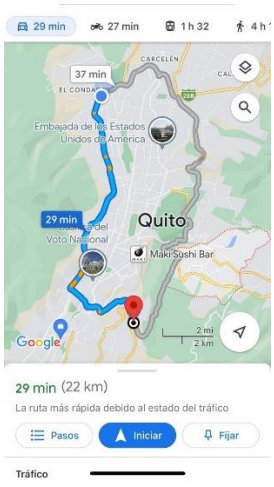
Actuador	Voltaje (V)	Amperaje (A)
Relé Luces Posición	13,87 V	1,56 A
Relé luces Neblineros	12,90	8,78 A
Relé motor Bomba limpia	13,4	1,51 A
Relé luces carretera	13,40	9,74 A
Relé luces cruce	13,53 V	8.94 A

Ensayo en ruta planificada

El ensayo en ruta planificada se realizó desde el sector del Condado en el norte de la ciudad de Quito, hasta el sector del peaje a los Chillos con una distancia de 22 km, donde se analizará en funcionamiento en conjunto de los actuadores y las mediciones de temperatura y nivel de combustible en pantalla, así como las revoluciones y medición de velocidad.

Figura 90

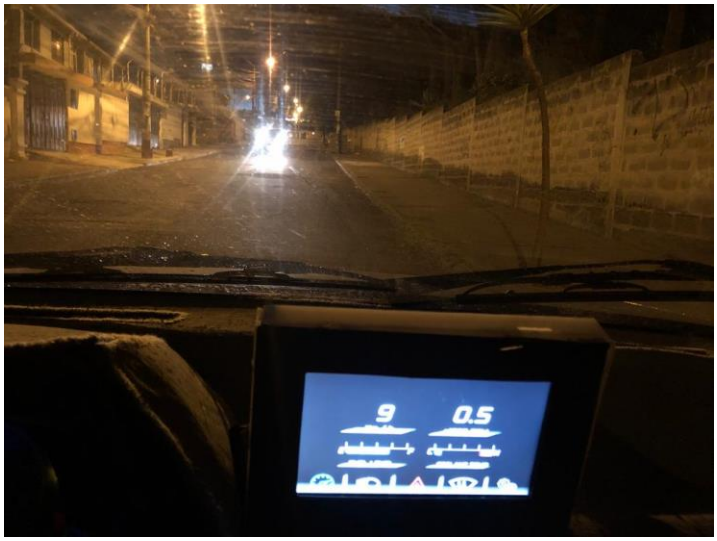
Mapa ruta planificada



En primer lugar, el viaje se realizó en la noche, con el fin de activar las luces y se pueda notar el funcionamiento del encendido, como se puede ver en la figura 91.

Figura 91

Estado apagado de luces



Posteriormente se activó las luces de cruce, en el menú de luces para poder continuar con el viaje

Figura 92

Activación de luces en ensayo

Con el paso del camino se tomó varias fotos para evidenciar como incrementaba la velocidad y las revoluciones, así se observa en la figura 93.

Figura 93

Prueba de ruta imagen de velocidad y rpm

Es importante tomar en cuenta que mientras este activado las luces, se muestran los testigos digitales que informan que las luces de posicion y cruce están encendidas, caso contrario, si no llega a mostrarse por algún motivo es que las luces no están activadas y el conductor tiene que verificar que se encuentre todo correcto.

Como la prueba se desarrolló en una ruta urbana, la velocidad a la que se encuentra el vehículo siempre fue dentro del rango permitido por la ley, de este modo se observa cómo funciona la velocidad y las revoluciones por minuto del vehículo, así mismo los testigos digitales que se encuentra prendidas las luces, y el medidor de combustible y temperatura marcan, de acuerdo con lo programado como indica la figura 94.

Figura 94

Activación permanente de luces y visualización de velocidad

Capítulo V

Marco administrativo

Recursos

Los recursos aplicados para la realización del proyecto a lo largo del periodo, fue necesario invertir en la parte económica, así como las personas que intervinieron en la realización de este.

Recursos humanos

El recurso humano que se necesita para el desarrollo del proyecto se observa en la tabla 7 :

Tabla 7

Recursos humanos

No.	*Descripción*	*Cantidad*	*Función*
1.	*Venegas Álvarez David Andrés*	*1*	*Investigador*
2.	*Ing. Zambrano Danilo*	*1*	*Colaborador Científico*

Nota. En la tabla se describe las personas que conforman el proyecto de investigación.

Recursos materiales

Los recursos materiales que el proyecto necesita se detallan en la tabla 8:

Recursos Materiales

Tabla 8

Recursos materiales

No	*Cantidad*	*Descripción*	*Valor unitario ($)*	*Valor total ($)*
1	*1*	*Pantalla Nextion NX8048P070-011C*	*$ 150,00*	*$150*
2	*1*	*Arduino DUE*	*$ 100*	*100*
3	*1*	*Sensor de velocidad*	*$ 45,75*	*$45,75*

No	*Cantidad*	*Descripción*	*Valor unitario ($)*	*Valor total ($)*
4	*1*	*Sensor de RPMs*	*$ 40,00*	*$40,00*
5	*1*	*Sensor de temperatura*	*$ 30,00*	*$60,00*
6	*1*	*Sensor de luminosidad*	*$ 5,00*	*$5,00*
7	*30*	*Alambre calibre 12awg /cm*	*$0,45*	*$13,50*
8	*6*	*Relé*	*$6,00*	*$36,00*
9	*2*	*Manguera protección cableado*	*$6,00*	*$12,00*
10	*1*	*Caja de fusibles*	*$55,00*	*$20,00*
11	*1*	*Soporte Pantalla*	*$25,00*	*$25,00*
12	*1*	*PCB*	*$ 20*	*$ 20*
13	*1*	*Componentes Electrónicos*	*$ 30*	*$ 30*
		Costo total		***$557,25***

Nota. En la tabla se detalla el presupuesto que se utilizará en la investigación citándose valores de materiales, equipos e instrumentos.

Recursos tecnológicos

Los recursos tecnológicos son indispensables para complementar el desarrollo del proyecto los cuales son detallados a continuación:

Tabla 9

Recursos tecnológicos

No	*Cantidad*	*Descripción*	*Valor unitario ($)*	*Valor total ($)*
1	*1*	*Computador Portátil*	*$ 600,00*	*$600*
2	*1*	*Curso de Programación pantalla Nextion*	*$ 40*	*$ 40*
		Costo total		***$640,00***

Presupuesto

Con lo detallado anteriormente se realiza una sumatoria del costo total del proyecto aproximadamente, con los valores a añadir de transporte y varios costos que se presentan en el desarrollo de este.

Tabla 10

Presupuesto Total

No	***Cantidad***	***Descripción***	***Valor unitario ($)***	***Valor total ($)***
1	*1*	*Recursos tecnológicos*	*$ 640,00*	*$640,00*
2	*1*	*Recursos materiales*	*$ 557,25*	*$ 557,25*
3	*1*	*Transporte*	*$ 2,55*	*$ 25,5*
4	*1*	*Varios*	*$ 50*	*$ 50*
		Costo total		***$1272,20***

Capítulo VI
Conclusiones y recomendaciones

Conclusiones

Se diseñó e implementó un panel de control y monitoreo digital para activación de funciones y lectura de datos en tiempo real del vehículo Chevrolet Gemini.

Se Investigó los diferentes componentes eléctricos – electrónicos, como sensores y actuadores, con el fin de seleccionar los elementos adecuados para la implementación del trabajo de titulación, los mismos que se utilizaron de manera eficiente en las distintas funciones de censado y activación del vehículo; a su vez se fabricó una PCB(Printed Circuit Board) que controla, regula y administra el voltaje necesario para cada sensor y actuador de acuerdo a las a los rangos de voltaje y corriente que el microcontrolador Arduino DUE necesita para su correcta operación.

Mediante el software Adobe Illustrator se diseñó la parte gráfica del panel de control y monitoreo, y en el software de interfaz gráfica de usuario de la marca NEXTION para la pantalla táctil Nextion NX8048P070 se programó el algoritmo de control de cada una de las herramientas táctiles a ser presionadas en la pantalla; así como el de los indicadores analógicos y digitales que existe en el diseño.

Para visualizar la temperatura del motor, nivel de combustible, velocidad en kilómetros por hora, las revoluciones por minuto y activación en pantalla de testigos gráficos digitales, se utilizó el software NEXTION EDITOR.

Se programó el algoritmo para recepción de datos y activación de funciones en el microcontrolador Arduino, obteniendo mediante el monitor serial los datos de cada uno de los sensores como temperatura, luminosidad, revoluciones por minuto, velocidad y nivel de combustible; y de esta manera se efectúan los comandos programados previamente.

Se generó el nuevo plano eléctrico del vehículo Chevrolet Gemini, con los componentes que se añadieron, así como los nuevos puertos de entrada de señales y puertos de salida para los actuadores que posee el vehículo.

Se instaló la pantalla táctil en la parte central del tablero del vehículo, se efectuó un ensayo de prueba en ruta para la obtención y verificación de datos obtenidos en tiempo real; además de la activación y desactivación de las diferentes funciones a través de los botones táctiles de la pantalla.

El proyecto desarrollado controla las luces de posición, cruce, carretera y neblinero en modo manual y automático, el motor del limpiaparabrisas enciende en velocidad hi, low, intermitente, como también la bomba para limpiaparabrisas y controlando los datos en vivo de velocidad, rpms, nivel de combustible y temperatura del vehículo, logrando reemplazar los elementos mecánicos y analógicos por elementos eléctricos y digitales para mejorar la interacción entre el conductor y el vehículo.

Recomendaciones

Se recomienda realizar una hoja de procedimientos donde se detalle cada paso que se realizó en la instalación de la pantalla y microcontrolador ,con el fin de evitar inconvenientes al momento de componentes eléctricos y electrónicos , a su vez que no sufran averías eléctricas por mala instalación.

Se recomienda en las uniones eléctricas entre dos cables, soldar y cubrir con aislante termo encogible, para evitar que se suelten o se aflojen debido al movimiento que tiene el carro, puede generar una interferencia o detectar ruidos en las entradas analógicas.

Con respecto a la instalación de cables se recomienda usar manguera para cables eléctricos, ajustar correctamente con correas de plástico o amarras, y ubicar en partes del vehículo donde se evite su visibilidad.

Es recomendable utilizar los módulos y accesorios originales de la placa Arduino, debido a que los fabricantes alternos, realizan una mala gestión de los voltajes y corrientes como por ejemplo en los módulos y eso impide conseguir datos reales y exactos de las variables analógicas que se pretenda utilizar.

Se recomienda implementar en varios proyectos a futuro la utilización de las pantallas táctiles y los microcontroladores como en prototipos eléctricos o de competición con el fin de desarrollar este tipo de tecnología que está en tendencia a evolucionar y ser aplicado en todos los vehículos.

Para un óptimo funcionamiento de la entrega de datos en vivo, se recomienda no utilizar la función “delay” en el código de Arduino, ya que este ralentiza la actividad de transmisión entre el microcontrolador y la pantalla, impidiendo que los datos sean presentados al instante y tengan un lapso hasta que se presenten nuevamente.

Bibliografía

Aprendiendoarduino. (26 de marzo de 2015). *Aprendiendoarduino*. Obtenido de https://aprendiendoarduino.wordpress.com/2015/03/26/lenguaje-de-programacion-de-arduino-estructura-de-un-programa/

Arduino. (2021). *Arduino*. Obtenido de Arduino Uno Rev3: https://store-usa.arduino.cc/products/arduino-uno-rev3?selectedStore=us

Autobody. (12 de septiembre de 2018). *Autobodymagazine*. Obtenido de https://www.autobodymagazine.com.mx/2018/09/12/ttd-vw-tablero-totalmente-digital/#:~:text=Una%20pantalla%20t%C3%A1ctil%20ha%20sustituido,Volkswagen%20avanza%20a%20pasos%20agigantados.

Figueroa, J. (2015). *Estudio y análisis del sistema multiplexado del vehículo híbrido toyota prius.[Tesis de Ingenieria,Universidad Internacional del Ecuador].* Repositorio Institucional, Guayaquil. Obtenido de http://repositorio.uide.edu.ec/handle/37000/840

Guillermo, W. (14 de junio de 2021). *Economipedia*. Obtenido de https://economipedia.com/definiciones/digitalizacion.html

KIONETWORK. (2022). *KIONETWORK*. Obtenido de ¿Qué son y para qué sirven los protocolos de comunicación de redes? : https://www.kionetworks.com/blog/data-center/protocolos-de-comunicaci%C3%B3n-de-redes#:~:text=Un%20protocolo%20es%20un%20conjunto,a%20trav%C3%A9s%20de%20una%20red.

Leen, G., & Heffernan, D. (2002). Expanding automotive electronic systems. *IEEE Xplore*, 88-93.

Lopez Cordero, C. (2006). *Diseño de un sistema prototipo de pantalla de estructura "Touch Screen" [Tesis de Ingeniería,Instituto Politécnico Nacional].* Repositorio Institucional ,

México D.F. Obtenido de https://es-static.z-dn.net/files/d51/b56515200bf861245c57ffbc2b14c252.pdf

María Fernández-Albalat, R. d. (2017). *Estudio de mercado de la incorporación de sistemas de confort, asistencia y seguridad en los vehículos [Tesis de Grado,Universidad Politecnica de Madrid].* Repositorio Institucional. Obtenido de https://oa.upm.es/45542/

MICROCHIP. (2022). *MICROCHIP*. Obtenido de PIC16F877A: https://www.microchip.com/en-us/product/PIC16F877A#

Moreno Díaz, Ó., Aller Pardo, J., & Mariano Pulido, I. (2001). *Domótica(Instalación y simulación)[Tesis de Ingeniería,Universidad Politécnica de Catalunya].* Repositorio Institucional, Barcelona. Obtenido de https://upcommons.upc.edu/bitstream/handle/2117/185610/40441-3452.pdf

Moreno, J. C. (2014). *Montaje y Mantenimiento de Equipos.* Madrid: RA-MA.

NEXTIONTECH. (03 de ENERO de 2022). *NEXTION*. Obtenido de https://nextion.tech/

Orientdisplay. (2014). *Orientdisplay*. Obtenido de orientdisplay. (2014). ¿Qué es mejor: pantalla táctil resistiva o capacitiva ? [Image]. Recuperado de https://www.orientdisplay.com/es/base-de-conocimientos/conceptos-b%C3%A1sicos-del-panel-t%C3%A1ctil/pantalla-t%C3%A1ctil-resistiva-o-capacitiva-que-es-me: https://www.orientdisplay.com/es/base-de-conocimientos/conceptos-b%C3%A1sicos-del-panel-t%C3%A1ctil/pantalla-t%C3%A1ctil-resistiva-o-capacitiva-que-es-mejor/

Rahid, A. (13 de JULIO de 2019). *THEENGINEERINGPROYECTS*. Obtenido de INTRODUCTION TO TIP31: https://www.theengineeringprojects.com/2019/07/introduction-to-tip31.html

Reyes , C. (2008). *Microcontroladores PIC.* Quito: RISPERGRAF.

Seijo, D. (26 de Abril de 2011). *Diariomotor*. Obtenido de https://www.diariomotor.com/2011/04/26/analogico-vs-digital-lo-moderno-frente-a-la-tradicion/

Tecneu. (2022). *Tecneu*. Obtenido de Modulo Sensor Medidor De Velocidad Rpm: https://www.tecneu.com/products/modulo-sensor-medidor-de-velocidad-rpm

Torrente Artero, Ó. (2013). *ARDUINO Curso práctico de formación .* México D.F: Alfaomega.

Villar Bonet, E. (2017). *La revolución Electrónica .* Santander : Editorial de la Universidad de Cantabria.

Anexos