



Powered by  
Arizona State University

# INGENIERÍA AUTOMOTRIZ

**Proyecto Previo a la Obtención del Título de Ingeniero en  
Mecánica Automotriz**

**Autores:** Miguel Ángel Arévalo Riccio

Anthony Josué Mendoza Ramírez

**Tutor:** Ing. Marco Noroña Merchán

**Estudio e Implementación de un Banco para Comprobación y**

**Diagnóstico de Módulos Automotrices**



### **Aprobación del Tutor**

Yo, Marco Vinicio Noroña Merchán certifico que conozco a los autores del presente trabajo siendo responsable exclusivo tanto de su originalidad y autenticidad, como de su contenido.

---

Ing. Marco Vinicio Noroña Merchán. MSc

Director del Proyecto

### **Certificación de Autoría**

Yo, Miguel Ángel Arévalo Riccio, juntos a mi compañero Anthony Josué Mendoza Ramírez, declaramos bajo juramento, que el trabajo aquí descrito es de nuestra autoría; que no ha sido presentado anteriormente para ningún grado calificación profesional y que se ha consultado la bibliografía detallada. Cedemos nuestros derechos de propiedad intelectual a la Universidad Internacional del Ecuador, para que sea publicado y divulgado en internet, según lo establecido en la Ley de Propiedad Intelectual, reglamento y leyes.

---

Miguel Ángel Arévalo Riccio

C.I: 0931292536

---

Anthony Josué Mendoza Ramírez

C.I: 0940642556

## **Dedicatoria**

El presente trabajo lo dedico con todo mi amor y cariño a Dios, a mi querida madre Martha Riccio por ese esfuerzo que hacías día a día para poder culminar mi carrera profesional, por sus palabras de aliento, por darme la mano cuando me caía, por siempre creer en mí y por ser el pilar fundamental en mi vida.

A mi padre Julio Arévalo que se encuentra en el cielo, gracias por cuidarme y guiarme desde allá arriba, gracias por todo lo que me enseñaste cuando estabas en vida y gracias por estar.

A mi abuelita Sofia Miranda y a mi tío Ángel Riccio que me han brindado su apoyo de manera incondicional.

Gracias por formar parte de mi vida, por sus consejos y palabras de alientos. Mis triunfos siempre llevaran sus nombres y una mirada al cielo.

***Miguel Ángel Arévalo Riccio***

## **Dedicatoria**

El presente trabajo se lo dedico a mis padres quienes siempre me estuvieron apoyando a lo largo de mi carrea como profesional.

A mi hermano Fausto Mendoza quién fue un apoyo incondicional a lo largo de estos años, qué no fue fácil, pero gracias a su ayuda pude seguir con mi formación profesional.

Gracias por ayudarme en este proceso que me ayuda a completar una de mis metas principales, sus palabras de aliento y consejos fueron de gran ayuda en este recorrido.

***Anthony Josué Mendoza Ramírez***

## **Agradecimiento**

Agradezco a Dios por permitirme tener una buena experiencia dentro de la UIDE, agradezco a la universidad por tan calurosa acogida desde el curso de nivelación, por haber sabido sobrellevar las clases en tiempo de pandemia, agradezco a mi tutor de tesis al Ing. Marco Noroña por haberme brindado la oportunidad de acudir a su capacidad y conocimientos que fueron de principal ayuda para el desarrollo de la tesis y agradezco a cada uno de los docentes que forman parte de tan noble institución como es la UIDE por transmitirme sus conocimientos desde el inicio de la carrera hasta mi último semestre.

***Miguel Ángel Arévalo Riccio***

## **Agradecimiento**

Agradezco a mi familia por depositar su confianza en mí, por darme la oportunidad de vivir una de las más importantes experiencias dentro de la UIDE, agradezco a la universidad quién demostró su total apoyo en los momentos más difíciles para poder seguir con mi proceso de estudios, al Ing. Marco Noroña por ser un guía y docente excelente impartiendo sus conocimientos y estar siempre al pendiente ante cualquier inquietud que tuviese.

*Anthony Josué Mendoza Ramírez*



## Resumen

En el presente proyecto el objetivo es proponer una nueva alternativa al remplazo de computadoras automotrices en el Ecuador mediante un banco de pruebas de ECU (Electronic Control Unit). Es por esto que el banco de pruebas está basado en un Software y Hardware de libre acceso llamado Arduino. Este banco de pruebas se usa para comprobar las señales de entrada y salida de la computadora automotriz hacia los actuadores del vehículo, lo que permite verificar si existe un fallo en la computadora o un fallo del actuador así se evita el reemplazo de la computadora innecesariamente en comparación a un solo componente que se pueda reemplazar y evitar las fallas en el vehículo. Se realiza las pruebas respectivas en computadoras automotrices de los vehículos “Chevrolet Sail”, donde en el primer análisis se detecta un fallo en el actuador del ventilador dando a entender que la computadora automotriz presenta una falla en esta señal, se realiza un segundo análisis con otra ECU y esta no presenta fallas, quiere decir que dicha computadora está en buen estado.

**Palabras clave:** ECU (Electronic Control Unit), Banco de Pruebas, Chevrolet Sail, Arduino, Software.

## **Abstract**

The objective of this project was pointed to propose a new alternative for replacing automotive computers in Ecuador through an ECU (Electronic Control Unit) test bench. This test bench is based on a freely accessible Software and Hardware called Arduino. This test bench is used to check the input and output signals from the automotive computer to the vehicle's actuators, it allows to verify if there is a computer or an actuator failure, this will avoid unnecessary computer replacements in the future, It works as a single component which can be replaced in order to prevent vehicle failures. The respective tests were carried out in automotive computers of "Chevrolet Sail" vehicles, In the first analysis a failure in the fan actuator was detected, implying that the automotive computer presented a failure in this signal, a second analysis was carried out with another ECU and this one had no faults, it meant that computer is working well.

**Keywords:** ECU (Electronic Control Unit), Test Bench, Chevrolet Sail, Arduino, Software.

## Índice de Contenido

Dedicatoria .....	v
Agradecimiento .....	vii
Resumen .....	ix
Abstract .....	x
Índice de Figuras .....	xviii
Índice de Tablas.....	xxiii
Capítulo I Antecedentes .....	24
1.1 Tema de Investigación .....	24
1.2 Planteamiento, Formulación y Sistematización del Problema.....	24
1.2.1 <i>Planteamiento del Problema</i> .....	24
1.2.2 <i>Formulación del Proyecto</i> .....	25
1.2.3 <i>Sistematización del Problema</i> .....	25
1.3 Objetivos de la Investigación.....	25
1.3.1 <i>Objetivo General</i> .....	25
1.3.2 <i>Objetivos Específicos</i> .....	25
1.4 Justificación y Delimitación de la Investigación .....	26
1.4.1 <i>Justificación Teórica</i> .....	26

1.4.2	<i>Justificación Metodológica</i> .....	26
1.4.3	<i>Justificación Practica</i> .....	26
1.4.4	<i>Delimitación Temporal</i> .....	26
1.4.5	<i>Delimitación Geográfica</i> .....	27
1.4.6	<i>Delimitación del Contenido</i> .....	27
1.5	Hipótesis .....	27
1.6	Variables de Hipótesis .....	27
1.6.1	<i>Variables Independientes</i> .....	27
1.6.2	<i>Variables Dependientes</i> .....	28
1.7	Metodología de la Investigación.....	28
1.7.1	<i>Métodos</i> .....	28
1.8	Tipo de Estudio.....	28
1.8.1	<i>Investigación Exploratoria.</i> .....	28
1.8.2	<i>Investigación de Campo</i> .....	28
1.8.3	<i>Investigación Aplicada</i> .....	29
	Capítulo II Marco de Referencia.....	30
2.1	Marco Teórico.....	30

2.2	Conceptos Preliminares .....	30
2.2.1	<i>Unidad de Control Electrónico (ECU)</i> .....	30
2.2.2	<i>Tipos de Módulo de Control</i> .....	31
2.2.3	<i>Tipos de Memorias</i> .....	31
2.2.4	<i>Estructura</i> .....	32
2.3	Fases de Procesamiento de Datos en la ECU .....	32
2.3.1	<i>Interfaz de Entrada y Salida</i> .....	32
2.3.2	<i>Filtrado de Señales</i> .....	33
2.3.3	<i>Convertidor Analógico y Digital</i> .....	33
2.3.4	<i>Unidad de Procesamiento</i> .....	34
2.3.5	<i>Etapa de Potencia</i> .....	34
2.4	Fallas y Posibles Averías en la ECU .....	34
2.5	Programación.....	35
2.6	Arduino .....	36
2.7	Comprobación.....	37
2.8	Diagnóstico .....	37
2.9	Implementación .....	38

2.10 Banco de Pruebas.....	38
2.11 Partes de un Banco de Pruebas .....	38
2.12 Sensores del Banco de Prueba .....	42
2.12.1 <i>Sensor FRP</i> .....	42
2.12.2 <i>Sensor ECT</i> .....	42
2.12.3 <i>Sensor MAP</i> .....	43
2.12.4 <i>Sensor TPS</i> .....	43
2.12.5 <i>Sensor O2</i> .....	44
2.12.6 <i>Sensor MAF</i> .....	44
2.12.7 <i>Sensor IAT</i> .....	45
2.13 Funcionamiento de la ECU.....	45
2.14 Posibles Fallas y Averías en la ECU .....	46
2.15 Ventajas de Plataforma Arduino (Que es Arduino).....	46
Capítulo III Implementación del Banco de Pruebas.....	48
3.1 Desarrollo de la Programación de Arduino .....	48
3.1.1 <i>Programación de la Pantalla LCD en Arduino</i> .....	48
3.1.2 <i>Programación de Vehículos Chrysler en Arduino</i> .....	49

3.1.3	<i>Señales del Vehículo Chrysler</i> .....	49
3.1.4	<i>Programación de Vehículos Audi en Arduino</i> .....	50
3.1.5	<i>Señales de Vehículo Audi</i> .....	50
3.1.6	<i>Programación de Vehículos BMW en Arduino</i> .....	51
3.1.7	<i>Señales de Vehículos BMW</i> .....	51
3.1.8	<i>Programación de Vehículos Ford en Arduino</i> .....	52
3.1.9	<i>Señales de Vehículos Ford</i> .....	52
3.1.10	<i>Programación de Vehículos Toyota en Arduino</i> .....	53
3.1.11	<i>Señales de Vehículos Toyota</i> .....	53
3.1.12	<i>Programación de Vehículos Volkswagen en Arduino</i> .....	54
3.1.13	<i>Señales de Vehículos Volkswagen</i> .....	54
3.1.14	<i>Programación de Vehículos Nissan en Arduino</i> .....	55
3.1.15	<i>Señales de Vehículos Nissan</i> .....	55
3.1.16	<i>Programación de Vehículos Chevrolet en Arduino</i> .....	56
3.1.17	<i>Señales de Vehículos Chevrolet</i> .....	56
3.1.18	<i>Programación de Vehículos con Sistema Diesel</i> .....	57
3.1.19	<i>Programación de Sensor CKP</i> .....	57

3.1.20	<i>Programación del Menú de Marcas.</i>	58
3.1.21	<i>Simulación de RPM.</i>	58
3.2	<i>Circuito Eléctrico del Banco de Pruebas</i>	59
3.2.1	<i>Sistema de Inyectores</i>	59
3.2.2	<i>Sistema de Batería de 12V y Sistema de Ignición.</i>	59
3.2.3	<i>Sistema de Bobinas</i>	60
3.2.4	<i>Sistema de ASD y Sistema Check Engine</i>	60
3.2.5	<i>Sistema Ventiladores, Tierra de Sensores y Alternador</i>	60
3.2.6	<i>Sistema Voltaje Sensores</i>	61
3.2.7	<i>Válvula IAC</i>	61
3.2.8	<i>Sistema de Bomba.</i>	62
3.2.9	<i>Conector DB25 B y Conector OBD2 16 Pin</i>	62
3.3	<i>Pin Out Chevrolet Sail</i>	67
3.4	<i>Conexiones del Banco de Pruebas a la ECU de Sail</i>	70
<b>Capitulo IV Análisis de Funcionamiento del Banco de Pruebas.</b>		<b>76</b>
4.1	<i>Análisis de la ECU con Falla al Banco de Prueba</i>	76
4.2	<i>Análisis de la ECU en Buen Estado al Banco de Pruebas.</i>	78



Conclusiones .....	80
Recomendaciones.....	81
Bibliografía.....	82
Anexos.....	84

## Índice de Figuras

Figura 1 <i>Computadora Automotriz (ECU)</i> .....	30
Figura 2 <i>Estructura de ECU</i> .....	32
Figura 3 <i>Entrada y Salida de una ECU del Vehículo Sail</i> .....	33
Figura 4 <i>Filtrado de Señales</i> .....	33
Figura 5 <i>Etapa de Potencia</i> .....	34
Figura 6 <i>Corrosión en una ECU</i> .....	35
Figura 7 <i>Programación de una ECU</i> .....	36
Figura 8 <i>Placa Arduino</i> .....	36
Figura 9 <i>Comprobación de ECU</i> .....	37
Figura 10 <i>Diagnóstico de la ECU</i> .....	37
Figura 11 <i>Implementación</i> .....	38
Figura 12 <i>Pantalla LCD</i> .....	38
Figura 13 <i>Arduino Mega 2560</i> .....	39
Figura 14 <i>Diodo Led</i> .....	39
Figura 15 <i>Condeador Electrolítico</i> .....	40
Figura 16 <i>Resistencia de 1k ohmios</i> .....	40

Figura 17 <i>Transistor</i> .....	40
Figura 18 <i>Pulsador</i> .....	41
Figura 19 <i>Relé</i> .....	41
Figura 20 <i>Diodo Semiconductor</i> .....	41
Figura 21 <i>Cargador 2A - AC</i> .....	42
Figura 22 <i>Sensor FRP</i> .....	42
Figura 23 <i>Sensor ECT</i> .....	43
Figura 24 <i>Sensor MAP</i> .....	43
Figura 25 <i>Sensor TPS</i> .....	43
Figura 26 <i>Sensor O2</i> .....	44
Figura 27 <i>Sensor MAF</i> .....	44
Figura 28 <i>Sensor IAT</i> .....	45
Figura 29 <i>Funcionamiento Correspondiente de la ECU</i> .....	45
Figura 30 <i>Posibles Fallas y Averías de la ECU</i> .....	46
Figura 31 <i>Placa Electrónica de Hardware</i> .....	47
Figura 32 <i>LCD – Programación</i> .....	48
Figura 33 <i>Chrysler Programación</i> .....	49

Figura 34 Señales de Vehículos Chrysler.....	49
Figura 35 Audi Programación.....	50
Figura 36 Señales Audi.....	50
Figura 37 BMW Programación.....	51
Figura 38 Señales BWM.....	51
Figura 39 Ford Programación.....	52
Figura 40 Señales Ford.....	52
Figura 41 Toyota Programación.....	53
Figura 42 Señales Toyota.....	53
Figura 43 Volkswagen Programación.....	54
Figura 44 Señales Volkswagen.....	54
Figura 45 Nissan Programación.....	55
Figura 46 Señales Nissan.....	55
Figura 47 Chevrolet Programación.....	56
Figura 48 Señales Chevrolet.....	56
Figura 49 Diesel Programación.....	57
Figura 50 CKP Programable.....	57

Figura 51 <i>Menú de Marcas</i> .....	58
Figura 52 <i>Simulación de RPM</i> .....	58
Figura 53 <i>Sistema Inyectores</i> .....	59
Figura 54 <i>Sistema de Batería 12 V y Sistema de Ignición</i> .....	59
Figura 55 <i>Sistema de Bobinas</i> .....	60
Figura 56 <i>Sistema ASD y Sistema Check Engine</i> .....	60
Figura 57 <i>Sistema Ventiladores, Sistema Tierra Sensores y Sistema Alternador</i> .....	61
Figura 58 <i>Sistema Voltaje Sensores</i> .....	61
Figura 59 <i>Sistema Válvula IAC</i> .....	62
Figura 60 <i>Sistema Bomba</i> .....	62
Figura 61 <i>Conector DB25 B y conector OBD2 16 Pin</i> .....	63
Figura 62 <i>Conector DB25 A y Conector DB89 C</i> .....	63
Figura 63 <i>Sistema 5V y Resistencia Can H L</i> .....	64
Figura 64 <i>Sistema TPS, Sistema ECT, Sistema MAP, Sistema MAF y Sistema FRP</i> .....	64
Figura 65 <i>Sistema O2 y Sistema IAT</i> .....	65
Figura 66 <i>Conector DB89 D tablero, Conector DB25 F llave &amp; Conector DB89 G</i> .....	65
Figura 67 <i>Sistema Start 12V e Inmovilizador</i> .....	66

Figura 68 Solenoides de Transmisión .....	66
Figura 69 Conexión Arduino Mega .....	67
Figura 70 Pin Out del Vehículo Sail.....	68
Figura 71 Puertos ECU A y ECU B .....	71
Figura 72 Orden del Pin ECU A .....	73
Figura 73 Orden del Pin ECU B.....	73
Figura 74 Banco de Pruebas Apagado.....	76
Figura 75 Apagado del Led del Check Engine .....	76
Figura 76 Apagado del Led Inmovilizador.....	77
Figura 77 Prueba #1 de Ecu al Banco de Pruebas .....	77
Figura 78 Falla de Fan en ECU del Sail.....	78
Figura 79 Falla de Variación de Voltaje en Sensores.....	78
Figura 80 Correcto Funcionamiento ECU #2 Sail.....	79
Figura 81 Variación de Voltaje en Sensores Dentro del Rango Permitido .....	79

## Índice de Tablas

Tabla 1 <i>Significado de Cada Uno de los Pines Out del Sail</i> .....	68
Tabla 2 <i>Detalle de ECU A</i> .....	71
Tabla 3 <i>Detalles de ECU B</i> .....	72
Tabla 4 <i>Conexión de ECU A – ECU del Sail</i> .....	73
Tabla 5 <i>Conexión de ECU B a la ECU del Sail</i> .....	74

## Capítulo I

### Antecedentes

#### 1.1 Tema de Investigación

Estudio e implementación de un banco para comprobación y diagnóstico de módulos automotrices.

#### 1.2 Planteamiento, Formulación y Sistematización del Problema

El sector automotriz en la ciudad de Guayaquil presenta números desafíos que deben ser abordados con urgencia. Uno de ellos es la comprobación del funcionamiento de computadoras automotrices, producto de falta de equipos y conocimientos. Por aquello consideran cambiar la computadora automotriz del vehículo.

##### 1.2.1 Planteamiento del Problema

A nivel mundial existen 1.420 millones de automóviles, de todos estos 1.060 millones corresponden a vehículos personales, mientras que 360 millones corresponden a comerciales. (Marnato, 2022)

En el Ecuador los vehículos livianos presentan un crecimiento interanual del 12%, mientras que los vehículos comerciales mostraron un crecimiento del 21%. En enero del 2022 se comercializaron 9.605 vehículos nuevos, dónde se puede evidenciar un crecimiento mensual del 0.6% con relación al mes de diciembre del 2021. (Over, 2022)

En la ciudad de Guayaquil circulan alrededor de 629.642 automóviles, El informe también hace referencia que la edad promedio del parque automotor ecuatoriano es de 16 años. (Dono, 2022). Es evidente que año a año se van sumando más automóviles en la ciudad de Guayaquil, sin embargo los dueños de sus vehículos cuando estos presentan un fallo en sus computadoras automotrices deciden cambiar el componente completo, esto debido a que la ciudad de Guayaquil carece de personas capacitadas y de equipos que puedan diagnosticar computadoras automotrices, algunas personas decidirán comprar este elemento de un alto valor,



puesto que no les dan más alternativas, sin embargo existirán personas que no puedan adquirirlo y tendrán su auto paralizado por un largo periodo.

### **1.2.2 Formulación del Proyecto**

¿El proyecto para el estudio e implementación de un banco de diagnóstico automotriz permitirá verificar el funcionamiento de ECUS en un vehículo Sail, lo que logrará darles una alternativa a las personas propietarias de vehículos en la ciudad de Guayaquil?

### **1.2.3 Sistematización del Problema**

- ¿Cómo se desarrollará la implementación del banco de diagnóstico automotriz?
- ¿Se logrará implementar un equipo banco de diagnóstico intuitivo, que resulte fácil manejo para las demás personas?
- ¿Qué tan beneficioso resultaría la implementación de un banco de diagnóstico automotriz que permitan verificar el funcionamiento de Ecus para los talleres en la ciudad de Guayaquil?

## **1.3 Objetivos de la Investigación**

### **1.3.1 Objetivo General**

Implementar un banco de diagnóstico automotriz mediante la plataforma Arduino para la verificación del funcionamiento del módulo de control electrónico del sistema de inyección electrónica de un vehículo Sail.

### **1.3.2 Objetivos Específicos**

- Investigar acerca de los principios de funcionamiento de los bancos de comprobación de módulos y partes que lo conforman.
- Implementar la programación haciendo uso de la plataforma Arduino en base a los requerimientos necesarios para uso del banco de diagnóstico.
- Realizar pruebas de verificación del funcionamiento correcto del banco de diagnóstico una vez acoplado al módulo de control electrónico del vehículo considerado.

- Elaborar guías prácticas sobre el banco de comprobación mediante estrategias didácticas.

#### **1.4 Justificación y Delimitación de la Investigación**

Teniendo en cuenta los objetivos planteados dentro del tema de estudio, la apología de la investigación se abordará desde una perspectiva teórica, metodológica y práctica.

##### ***1.4.1 Justificación Teórica***

Se basa en la investigación de equipos que ayuden comprobando el estado de las computadoras automotrices. Programar usando herramientas de acceso libre, donde se apoya de equipos existentes en el mercado que puedan aportar al proyecto.

##### ***1.4.2 Justificación Metodológica***

La implementación y estudio de un banco de diagnóstico automotriz para verificar el funcionamiento de ECUS en un vehículo Sail por medio de la plataforma Arduino, teniendo en cuenta todos los aspectos técnicos, mediante un proceso sistemático y práctico. nuestro estudio acerca de un equipo de comprobación para computadoras automotrices, pueden permitir una alternativa en los talleres en la ciudad de Guayaquil.

##### ***1.4.3 Justificación Práctica***

A nivel internacional, existen diferentes tipos de equipos para la comprobación de computadoras automotrices.

El resultado de investigación permite conocer sobre estos equipos y sus valores, los mismos que se mantiene en cuenta para la elaboración de nuestro equipo y poder abaratar costos en comparación a uno de los equipos existentes en el mercado.

##### ***1.4.4 Delimitación Temporal***

El trabajo se desarrollará desde el mes de junio del 2022, hasta junio del 2023, tiempo que permitirá realizar la investigación, así como diseñar y construir el banco de pruebas.

### **1.4.5 Delimitación Geográfica**

El trabajo se desarrollará en la Escuela de Ingeniería Automotriz en la Universidad Internacional del Ecuador con sede en Guayaquil.

### **1.4.6 Delimitación del Contenido**

En el primer bloque se encuentra un mapa conceptual, estructurado por conceptos necesarios para la discusión tales como computadoras automotrices, vehículos que se encuentran en la facultad donde posteriormente se hará uso de éstos para poder comprobar el funcionamiento de la Ecu, plataforma que ayuden con la programación de comprobación para computadoras automotrices.

En el segundo bloque se visualiza el diseño y programación del equipo bajo la plataforma Arduino.

El tercer bloque está constituido por fallas comúnmente presentadas en las Ecus, y además las pruebas que se realizarán con el equipo, para luego realizar las guías prácticas.

## **1.5 Hipótesis**

La implementación de un banco de diagnóstico de ECUS en Guayaquil es un proceso que permitirá obtener resultados positivos en cuanto a factores económicos y académicos con la reparación de computadoras automotrices y capacitaciones.

Este proyecto consiste en implementar un banco de diagnóstico automotriz para verificar el diagnóstico de las computadoras automotrices en Guayaquil, en un horizonte temporal de implementar el banco de diagnóstico automotriz para computadoras automotrices, con el fin de diagnosticar y reparar las computadoras automotrices.

## **1.6 Variables de Hipótesis**

### **1.6.1 Variables Independientes**

- Implementación de un banco de pruebas para computadoras automotrices

### **1.6.2 Variables Dependientes**

- Sail
- Diagnóstico y análisis de Ecu
- Banco de diagnóstico de Ecus por medio de la plataforma Arduino

## **1.7 Metodología de la Investigación**

### **1.7.1 Métodos**

En un principio se empezará con la recopilación y análisis del banco de prueba de computadoras automotrices, seleccionando de artículos científicos, blogs automotrices, revistas, libros, documentos, tesis y sitios web con excelente información, su inicio, principales elementos, entre otros.

Para obtener una mejor información sobre el banco de prueba de computadoras automotrices, en el presente trabajo se pretende utilizar una metodología que permita la aplicación de varios métodos y técnicas con el propósito de compilar información sobre el banco de pruebas: Plataforma Arduino, Ecus, Mazda 3, costos, etc. en este trabajo se recurrirá a la bibliografía especializada sobre el tema estudiar.

## **1.8 Tipo de Estudio**

El estudio pragmático se define bajo un concepto de la práctica y experimentación donde la realización de acciones son puntos clave para evaluar la efectividad.

### **1.8.1 Investigación Exploratoria.**

Se realizará investigaciones y estudios acerca de los tipos de plataformas que ayuden con la programación del banco de pruebas para poder definir cual el más oportuno.

### **1.8.2 Investigación de Campo**

Se adhiere este tipo de investigación, dado que mediante ella podemos informarnos equipos similares existentes en el mercado.

### ***1.8.3 Investigación Aplicada***

Mediante ella se puede poner en práctica los diferentes tipos de estudios para la implementación del banco de pruebas de computadoras automotrices, y a su vez poner en práctica el aprendizaje durante los años de estudio

## Capítulo II

### Marco de Referencia

#### 2.1 Marco Teórico

Durante mucho tiempo el sector automovilístico ha ido incrementado, misma que se ha podido presenciar en las diferentes ciudades principales del Ecuador; es una realidad que debido al cambio climático que sufre la ciudad de Guayaquil los ingresos de los vehículos en los talleres automotrices por motivos eléctricos son abundantes, donde se presentan problemas en la ECU y como única solución que ofrecen los talleres al cliente es el cambio total de la misma.

#### 2.2 Conceptos Preliminares

La computadora automotriz es la unidad de control electrónico (ECU), es el elemento principal de los vehículos, la cual registrará diversos parámetros sobre el funcionamiento del vehículo. (Panadero, 2012)

##### 2.2.1 Unidad de Control Electrónico (ECU)

Se la conoce con varios nombres, entre ellos: computadora automotriz, centralita, recibe la información de los sensores y ésta a su vez envía la orden a los actuadores, como se observa en la figura 1 a la computadora automotriz. (Den, 2022)

#### Figura 1

*Computadora Automotriz (ECU)*



Fuente: (D'Addario, 2017)

### 2.2.2 *Tipos de Módulo de Control*

Existen diferentes tipos de módulo de control, entre ellos:

- PCM (Powertrain Control Module), módulo de control del tren de potencia.

La función que cumple es la de almacenar y controlar los datos de la transmisión y la del motor.

- ECM (Engine Control Module), módulo de control electrónico, módulo de control electrónico.

La función es controlar y almacenar los códigos de falla que se generan mediante los actuadores del motor y sensores.

- VCM (Vehicle Control Module), módulo de control del vehículo.

La función que cumple es de almacenar y controlar los datos de los sistemas de frenos ABS, la transmisión y del motor. (Villa, 2016)

### 2.2.3 *Tipos de Memorias*

Existen diferente tipo de memorias, entre ellas:

- RAM (Random Acces Memory)

Es la encargada de almacenar de manera temporal los datos y valores la cual los sensores estarán enviando, esta necesita una alimentación de voltaje constante para que pueda funcionar, cada vez que el interruptor que se encuentre en apagado esta perderá su información.

- PROM (Programable ROM)

Es una memoria de tipo programable, es decir una memoria solo de lectura, donde si la programación de esta presenta algún error deberá ser desechada.

- EEPROM (Electrically Programmable Read Only Memory)

Es una memoria que se encuentra soldada a los circuitos de la ECM, memoria solo de lectura y se puede programar, se borra de manera electrónica.

- EPROM (Erasable- Programable Read Only Memory)

Es una memoria que se puede programar y borrar según lo que decida el usuario, nos permite almacenar datos indefinidamente.

- ROM (Read Only Memory)

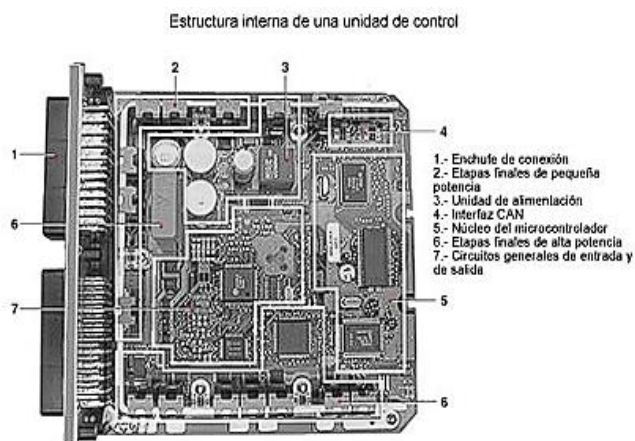
Es una memoria que no necesita energía, no se pueden borrar los datos puesto que solo es de lectura, permite guardar información de manera permanente. (Villa, 2016)

#### 2.2.4 Estructura

La Ecu contiene una carcasa metálica la cual la protege, la comunicación y alimentación de esta se realiza mediante un conector multipolar, como se detalla en la figura 2 con sus respectivas partes. (Noroña, 2018).

### Figura 2

#### Estructura de ECU



Fuente: (Villa, 2016)

### 2.3 Fases de Procesamiento de Datos en la ECU

A continuación, se detalla las fases de procesamiento de una computadora automotriz:

#### 2.3.1 Interfaz de Entrada y Salida

Los sensores proceden a enviar la señal a la ECU o centralita y este a su vez envía la señal a los actuadores, en la figura 3 se observó el pin out del vehículo Sail y la cual posee 100 pines.



### Figura 3

*Entrada y Salida de una ECU del Vehículo Sail*

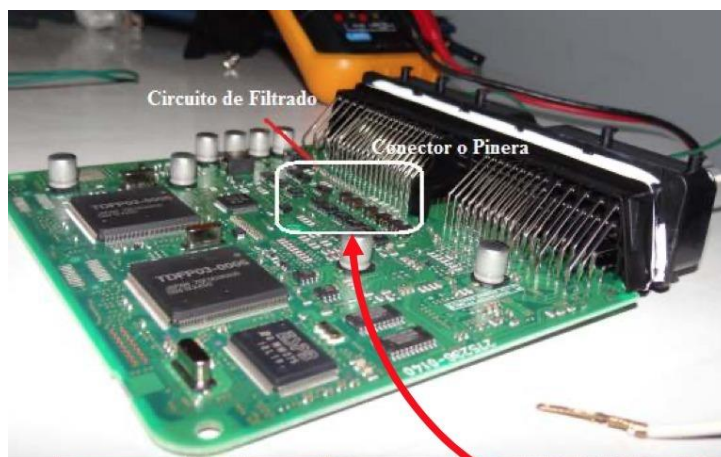


#### 2.3.2 Filtrado de Señales

Tiene como misión eliminar la interferencia y ruido la cual se pudo haber producido en la transmisión de la señal de un sensor, en la figura 4 se especificó el circuito de filtrado y el conector.

### Figura 4

*Filtrado de Señales*



Fuente: (Villa, 2016)

#### 2.3.3 Convertidor Analógico y Digital

La ECU solo interpreta señales digitales, por lo tanto, las señales de los sensores de tipo analógico se convierten en digital.

### 2.3.4 Unidad de Procesamiento

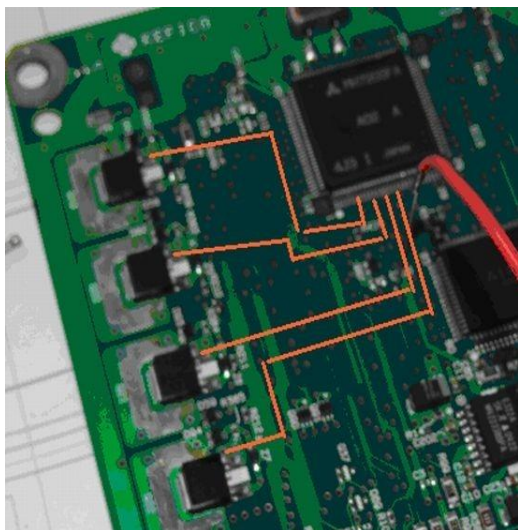
Es la encargada de recibir las señales enviadas por los sensores para que su vez este pueda encausar la información y realizar los cálculos correspondientes de tal manera pueda decidir cuáles son los actuadores que se usaran para que el motor funcione en perfectas condiciones.

### 2.3.5 Etapa de Potencia

Estos se ubican en el integrado de la ECU, mismos que son los encargados de evitar la sobrecarga de los componentes de esta, en la figura 5 se detalló las etapas de la potencia donde se presencia transistores y microprocesadores.

#### Figura 5

#### *Etapa de Potencia*



Fuente: (Mellard, 2017)

## 2.4 Fallas y Posibles Averías en la ECU

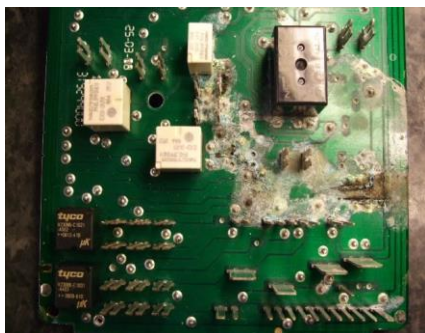
A continuación, se detalla las posibles fallas físicas:

- *Pista quemada*: Se dice que una pista esta quemada cuando este genera sobrecarga la cual interrumpe el paso de la corriente y a la vez perjudica el funcionamiento de la Ecu.

- *Corrosión*: Se puede encontrar el deterioro del material en los componentes eléctricos de la ECU, la cual producen una interferencia en la parte electrónica o conexiones que afectan al funcionamiento del vehículo, en la figura 6 se observó como la corrosión daño el interior de una ECU.

## Figura 6

*Corrosión en una ECU*



Fuente: (Villa, 2016)

- *Transistor averiado*: Esta falla suele presentarse cuando los transistores reciben un sobre voltaje, para corregir esta falla se debe ubicar el transistor y remplazarlo sí este fuese necesario.

- *Fallas de condensador*: Cuando un condensador pierde la capacidad de acumular energía se debe a que el mismo se encuentra sometido a temperaturas muy elevadas; y también se puede hallar otra falla producida por la deformación en la parte física.

- *Fallas del microprocesador*: Es considera una de las partes más críticas de la ECU debido a que no se encuentra fácilmente, en caso de avería se debería cambiar el microprocesador o remplazar la ECU.

## 2.5 Programación

La programación, aunque no se vea físicamente está en todos los equipos electrónicos que se usa, se puede definir como el conjunto de indicaciones que llevarán a la computadora a

realizar un trabajo específico, se observó en la figura 7 como la ECU está conectada al banco y así se verificó su funcionamiento. (Gil, 2021)

### **Figura 7**

#### *Programación de una ECU*



Fuente: (Silva, 2020)

## **2.6 Arduino**

Es una plataforma electrónica de código abierto basada en hardware y software fáciles de usar. las placas Arduino pueden leer entradas (lucen un sensor, un dedo en un botón o un mensaje de Twitter) (Morán, 2022). Para posteriormente convertirlo en salida, es decir encender un led o publicar algo en línea; en la figura 8 se detalló la placa Arduino mega 2560 que es apropiada para el uso del banco. (Walack, 2018)

### **Figura 8**

#### *Placa Arduino*



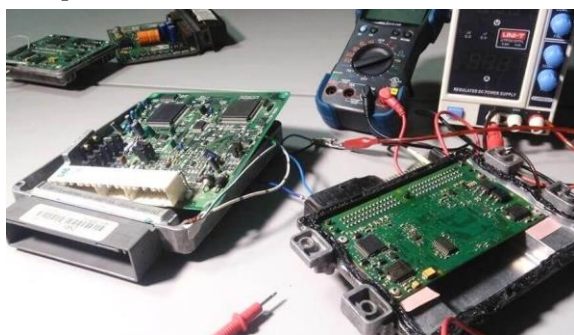
Fuente: (Fernandez, 2012)

## 2.7 Comprobación

Se refiere a la confirmación o prueba de la existencia, en pocas palabras a la veracidad o exactitud de una cosa, en la figura 9 se detalló con claridad como es el mecanismo para la comprobación de una ECU. (Snold, 2022)

### Figura 9

*Comprobación de ECU*



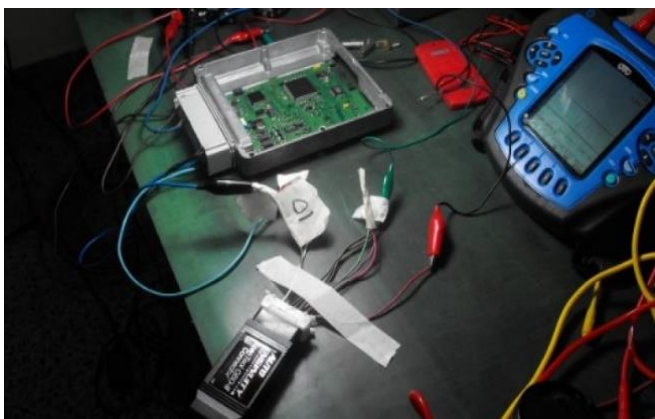
Fuente: (D'Addario, 2017)

## 2.8 Diagnóstico

Se le denomina diagnóstico al proceso de reconocimiento, análisis y evaluación de una computadora automotriz para determinar sus tendencias de tal manera solucionar un problema, en la figura 10 se observa cómo se puede realizar un diagnóstico de una ECU. (Tarlo, 2022)

### Figura 10

*Diagnóstico de la ECU*



Fuente: (Luque, 2008)

## 2.9 Implementación

Se basa en aplicar métodos, aplicar medidas para llevar algo a cabo, en la figura 11 se detalla la implementación para el proceso de comprobación de una ECU. (Bello, 2021)

### Figura 11

*Implementación*



Fuente: (Alonso, 2003)

## 2.10 Banco de Pruebas

Un banco de pruebas es un dispositivo o equipo que ayuda a diagnosticar y saber el estado de algún aparato electrónico. (Bello, 2021)

## 2.11 Partes de un Banco de Pruebas

*LCD*: Es una pantalla plana formada por pixeles y números, que se emplea para visualizar imágenes fijas y en movimiento, en la figura 12 se detalla la pantalla LCD (Den, 2022)

### Figura 12

*Pantalla LCD*



Fuente: (Den, 2022)



*Arduino mega 2560*: Es una placa del microcontrolador ATmega2560, dispone de 54 entradas y salidas, y de estas se pueden usar 15 como salidas (PWM), una conexión USB, un conector de alimentación y un botón RESET. En la figura 13 se aprecia el Arduino mega 2560. (Gil, 2021)

### Figura 13

*Arduino Mega 2560*



Fuente: (Gil, 2021)

*Diodo Led*: Es un emisor de luz que emite fotones cuando este recibe una corriente eléctrica de baja intensidad. En la figura 14 se aprecia el foco led. (Gil, 2021)

### Figura 14

*Diodo Led*



Fuente: (Gil, 2021)

*Condensador Electrolítico*: Es un componente eléctrico con dos terminales con la capacidad de acumular energía eléctrica. En la figura 15 se aprecia el condensador. (Alonso, 2003)

### Figura 15

*Condensador*



Fuente: (Alonso, 2003)

*Resistencias de 1k ohmios +- 5%:* La resistencia es semiconductor y su principal función es de restringir el paso de la corriente. En la figura 16 se aprecia la resistencia de 1k ohmios.

(Gil, 2021)

### Figura 16

*Resistencia de 1k Ohmios*



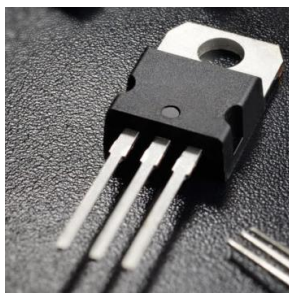
Fuente: (Gil, 2021)

*Transistores:* Es un dispositivo semiconductor y se compone de tres elementos: base, colector y emisor; funcionan como flujo de corriente actuando como amplificador e interruptor.

En la figura 17 se aprecia un transistor. (Gil, 2021)

### Figura 17

*Transistor*





Fuente: (Gil, 2021)

*Pulsador*: Es un dispositivo que permite interrumpir el paso de la corriente eléctrica en la figura 18 se aprecia un pulsador. (Luque, 2008)

### Figura 18

*Pulsador*



Fuente: (Luque, 2008)

*Relé*: Es un dispositivo electromagnético que permite abrir o cerrar contactos mediante electroimán. En la figura 19 se observa un relé (D'Addario, 2017)

### Figura 19

*Relé*

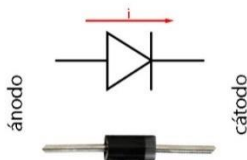


Fuente: (D'Addario, 2017)

*Diodo Semiconductor*: Es un dispositivo semiconductor que sirve como interruptor unidireccional para la corriente. En la figura 20 se aprecia un diodo. (Cajicasa, 2016)

### Figura 20

*Diodo*



Fuente: (Cajicasa, 2016)

*Cargador 2A - AC*: Elemento que sirve para poder alimentar el banco de pruebas. En la figura 21 se observa el cargador de 2A (Alonso, 2003)

### **Figura 21**

*Cargador de 2A - AC*



Fuente: (Alonso, 2003)

## **2.12 Sensores del Banco de Prueba**

### **2.12.1 Sensor FRP**

Por sus siglas Fuel Rail Pressure, en español es el sensor del riel el mismo que es el encargado de detectar la presión de la gasolina en el riel. En la figura 22 se observa el sensor FRP. (Den, 2022)

### **Figura 22**

Señor FRP



Fuente: (Den, 2022)

### **2.12.2 Sensor ECT**

Por sus siglas Engine Coolant Temperature, este sensor es el encargado de medir la temperatura del refrigerante. En la figura 23 se observa el sensor ECT (D'Addario, 2017)

**Figura 23**

Sensor ECT



Fuente: (D'Addario, 2017)

### 2.12.3 Sensor MAP

Por sus siglas Manifold Absolute Pressure, este sensor es el encargado de la medición de presión de aire que entra el motor mediante el múltiple de admisión. En la figura 24 se aprecia el sensor MAP (Luque, 2008)

**Figura 24**

Sensor MAP



Fuente: (Luque, 2008)

### 2.12.4 Sensor TPS

Por sus siglas Throttle Position Sensor, este sensor es encargado de controlar la mariposa, es decir que su función es pasar la señal a la Ecu para poder controlar la inyección de combustible. En la figura 25 se visualiza el sensor TPS (Bello, 2021)

**Figura 25**

Sensor TPS



Fuente: (Bello, 2021)

### **2.12.5 Sensor O2**

Es el encargado de verificar si la mezcla que realizan los motores y sale por el escape tiene exceso o tiene poco combustible. En la figura 26 se observa el sensor O2 (Alonso, 2003)

#### **Figura 26**

*Sensor O2*



Fuente: (Alonso, 2003)

### **2.12.6 Sensor MAF**

Por sus siglas Mass Air Flow, es el encargado de determinar la temperatura del aire de admisión en cualquier estado del auto (es decir em frio o caliente). En la figura 27 se observa el sensor MAF (Den, 2022)

#### **Figura 27**

*Sensor MAF*



Fuente: (Den, 2022)

### 2.12.7 Sensor IAT

Por sus siglas Intake Air Temperature, este sensor es el encargado de medir la temperatura que ingresa al colector de admisión esto ayuda a que el motor obtenga la cantidad de aire correcta. En la figura 28 se observa un sensor IAT (D'Addario, 2017)

#### Figura 28

*Sensor IAT*



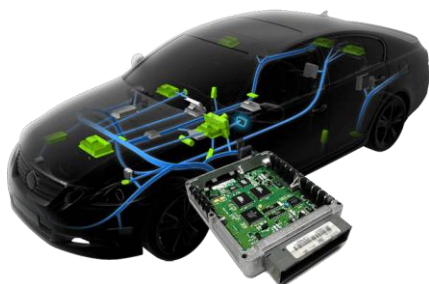
Fuente: (D'Addario, 2017)

### 2.13 Funcionamiento de la ECU

La unidad de control (ECU) también conocida como: ECM, ECC, ECCS, CPU, PCM, VCM, entre otros. Es quién recibe los datos de entrada de los dispositivos, tales como: interruptores y sensores, quiénes son los encargados de procesar la información, entre ellos: Temperatura de aire interior y exterior del vehículo, presión del colector, posición del árbol de levas, posición del acelerador, oxígeno dentro del motor, es él cigüeñal, control de ralentí, entre otros, con la finalidad de activar los actuadores, en la figura 29 se puede observar como la ECU envía información a los actuadores para su ejecución. En la figura 29 funcionamiento de la ECU (Cajicasa, 2016)

#### Figura 29

*Funcionamiento Correspondiente de la ECU*



Fuente: (Luque, 2008)

## 2.14 Posibles Fallas y Averías en la ECU

Dentro de las principales fallas se clasifican en: físicas y lógicas, en las cuales las fallas físicas pueden ser: pistas quemadas, punto de soldadura agrietado, corrosión por óxido, transistor averiado, falla de condensadores, fallas el microprocesador. la falla es lógica suelen producirse en la base de datos del programa interno, lo que no requiere que se desarme la computadora, estas fallas por lo general se suelen solucionar con una reprogramación de esta, en la figura 30 se detalla cómo se puede verificar y solucionar la reprogramación de una ECU. En la figura 30 se observa las posibles fallas y averías de la ECU (Cajicasa, 2016)

### Figura 30

*Posibles Fallas y Averías de la ECU*



Fuente: (Cajicasa, 2016)

## 2.15 Ventajas de Plataforma Arduino (Que es Arduino)

Una de las principales ventajas de la plataforma Arduino es su bajo costo que le permite a las personas el uso de dicho software, además se considera una plataforma fácil de usar, donde con pocos conocimientos de programación se puede interactuar con la misma. cuenta con un hardware y software de código abierto, además de gran variedad de placas que permiten crear proyectos de toda clase. En la figura 32 se observa la placa a utilizar en el banco de pruebas. (Walack, 2018)

**Figura 31**

*Placa Electrónica de Hardware*



Fuente: (Fernandez, 2012)

## Capítulo III

### Implementación del Banco de Pruebas

#### 3.1 Desarrollo de la Programación de Arduino

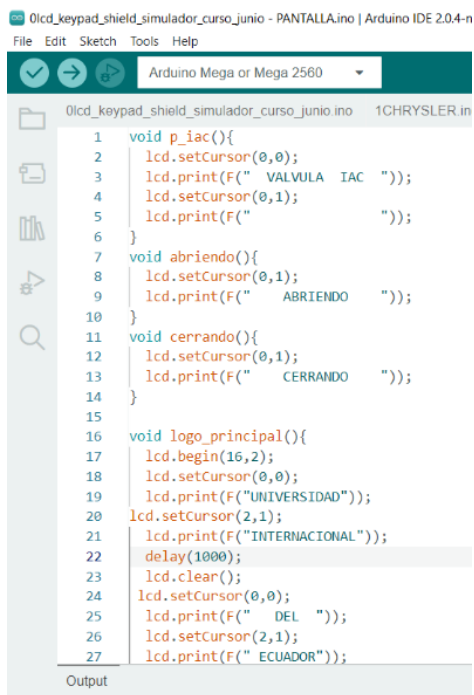
En este apartado se observa la programación detrás del banco de prueba mediante el software Arduino haciendo uso de una placa “Arduino mega 2560” ideal para este proyecto debido al contenido de programación que se le cargará.

##### 3.1.1 Programación de la Pantalla LCD en Arduino

En esta primera captura se encuentra con la programación de la pantalla led, es decir lo que se mostrará apenas se prenda nuestro banco de pruebas. En la imagen 32 se observa la programación detrás de una pantalla LCD

**Figura 32**

*LCD - Programación*



```
0lcd_keypad_shield_simulador_curso_junio - PANTALLA.ino | Arduino IDE 2.04-n
File Edit Sketch Tools Help
Arduino Mega or Mega 2560
0lcd_keypad_shield_simulador_curso_junio.ino 1CHRYSLER.in
1 void p_iac(){
2   lcd.setCursor(0,0);
3   lcd.print(F(" VALVULA IAC "));
4   lcd.setCursor(0,1);
5   lcd.print(F("          "));
6 }
7 void abriendo(){
8   lcd.setCursor(0,1);
9   lcd.print(F(" ABRIENDO "));
10 }
11 void cerrando(){
12   lcd.setCursor(0,1);
13   lcd.print(F(" CERRANDO "));
14 }
15
16 void logo_principal(){
17   lcd.begin(16,2);
18   lcd.setCursor(0,0);
19   lcd.print(F("UNIVERSIDAD"));
20   lcd.setCursor(2,1);
21   lcd.print(F("INTERNACIONAL"));
22   delay(1000);
23   lcd.clear();
24   lcd.setCursor(0,0);
25   lcd.print(F(" DEL "));
26   lcd.setCursor(2,1);
27   lcd.print(F(" ECUADOR"));
Output
```

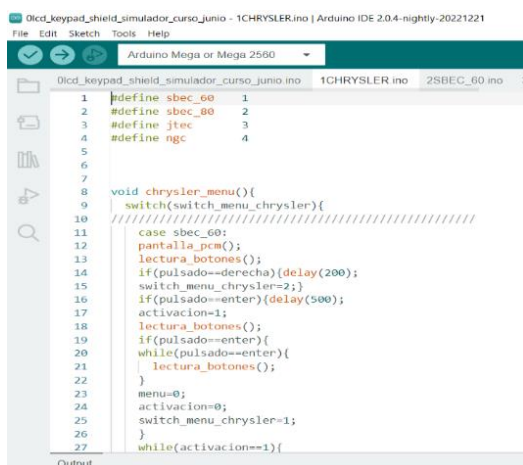


### 3.1.2 Programación de Vehículos Chrysler en Arduino

Luego de tener la programación de nuestra pantalla led se procede con el primero modelo de autos “Chrysler” donde será dividida en 2 partes para una mejor organización. En la figura 33 se detalla la programación de vehículos Chrysler.

**Figura 33**

#### Programación de Vehículos Chrysler



```

1  #define sbec_00  1
2  #define sbec_80  2
3  #define jtec    3
4  #define ngc     4
5
6
7
8  void chrysler_menu(){
9    switch(switch_menu_chrysler){
10   ///////////////////////////////////////////////////
11   case sbec_60:
12     pantalla_pcm();
13     lectura_botones();
14     if(pulsado==derecha){delay(200);
15     switch_menu_chrysler=2;}
16     if(pulsado==enter){delay(500);
17     activacion=1;
18     lectura_botones();
19     if(pulsado==enter){
20     while(pulsado==enter){
21     | lectura_botones();
22     }
23     menu=0;
24     activacion=0;
25     switch_menu_chrysler=1;
26     }
27     while(activacion==1){

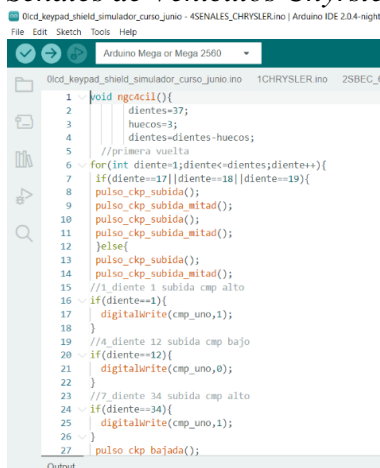
```

### 3.1.3 Señales del Vehículo Chrysler

En el siguiente apartado se observa la programación de las señales que se enviarán y recibirán el banco de prueba por parte de la marca “Chrysler”. En la figura 34 se aprecia la programación de las señales de los vehículos Chrysler.

**Figura 34**

#### Señales de Vehículos Chrysler



```

1  void ngc4c11(){
2    dientes=37;
3    huecos=3;
4    dientes=dientes-huecos;
5    //1era vuelta
6    for(int diente=1;diente<dientes;diente++){
7      if(diente==17||diente==18||diente==19){
8        pulso_ckp_subida();
9        pulso_ckp_subida_mitad();
10       pulso_ckp_subida();
11       pulso_ckp_subida_mitad();
12       }else{
13       pulso_ckp_subida();
14       pulso_ckp_subida_mitad();
15       //1_diente 1 subida cmp alto
16       if(diente==1){
17         digitalWrite(cnp_uno,1);
18       }
19       //4_diente 12 subida cmp bajo
20       if(diente==12){
21         digitalWrite(cnp_uno,0);
22       }
23       //7_diente 34 subida cmp alto
24       if(diente==34){
25         digitalWrite(cnp_uno,1);
26       }
27       pulso_ckp_bajada();

```

### 3.1.4 Programación de Vehículos Audi en Arduino

En la figura 35 se puede divisar la programación usada para los vehículos Audi.

**Figura 35**

#### Programación de Vehículos Audi



```

0lcd_keypad_shield_simulador_curso_junio - AUDLino | Arduino IDE 2.0.4-nightly
File Edit Sketch Tools Help
Arduino Mega or Mega 2560
0lcd_keypad_shield_simulador_curso_junio.ino 1CHRYSLER.inc
1 #define r_a419 1
2 #define r_a619 2
3 #define r_a624 3
4 #define r_q730 4
5
6 void audi_menu(){
7   switch(switch_menu_audi){
8
9     case 1:
10    pantalla_a419();
11    lectura_botones();
12    if(pulsado==derecha){delay(200);
13    switch_menu_audi=2;}
14    if(pulsado==enter){delay(500);
15    activacion=1;
16    lectura_botones();
17    if(pulsado==enter){
18    while(pulsado==enter){
19      lectura_botones();
20    }
21    menu=0;
22    activacion=0;
23    switch_menu_audi=1;
24  }
25  while(activacion==1){
26    senal_a419();
27  }
  }
}
Output

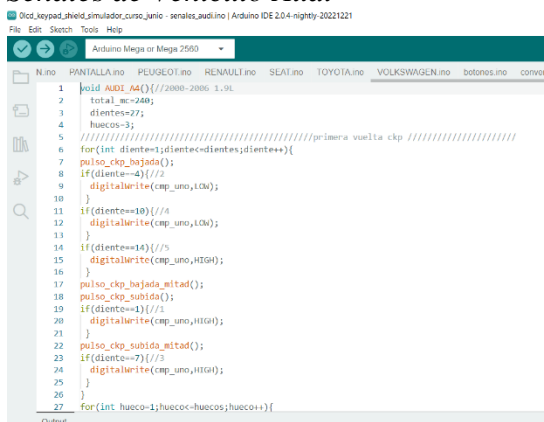
```

### 3.1.5 Señales de Vehículo Audi

En la figura 36 se observa la programación de las señales que se enviarán y recibirán el banco de prueba por parte de la marca “Audi”.

**Figura 36**

#### Señales de Vehículo Audi



```

0lcd_keypad_shield_simulador_curso_junio - senales_audi.ino | Arduino IDE 2.0.4-nightly-2021221
File Edit Sketch Tools Help
Arduino Mega or Mega 2560
N:ino PANIALLA.ino PEUGEOT.ino RENAULT.ino SEAT.ino TOYOTA.ino VOLKSWAGEN.ino botones.ino conve
1 void Audi_A4(){//2000-2006 1.9L
2   total_mc=240;
3   diente=27;
4   hueccos=3;
5   ////////////////////////////////////////////primera vuelta ckp ////////////////////////////////////////////
6   for(int diente=1;diente<dientes;diente++){
7     pulso_ckp_hajada();
8     if(diente==4){//1/2
9       digitalWrite(cmp_umo,LOW);
10    }
11    if(diente==10){//1
12    digitalWrite(cmp_umo,LOW);
13    }
14    if(diente==14){//1/5
15    digitalWrite(cmp_umo,HIGH);
16    }
17    pulso_ckp_hajada_mitad();
18    pulso_ckp_subida();
19    if(diente==1){//1
20    digitalWrite(cmp_umo,HIGH);
21    }
22    pulso_ckp_subida_mitad();
23    if(diente==7){//3
24    digitalWrite(cmp_umo,HIGH);
25    }
26  }
27  for(int huecco=1;huecco<hueccos;huecco++){
}
Output

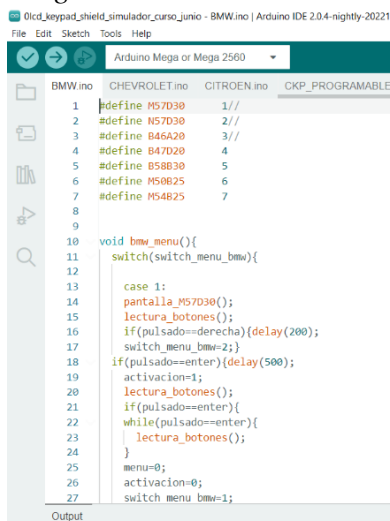
```

### 3.1.6 Programación de Vehículos BMW en Arduino

En la figura 37 se procede con el siguiente modelo de autos “BMW” donde será dividida en dos partes para una mejor organización.

**Figura 37**

#### Programación de Vehículos BMW



```

Arduino Mega or Mega 2560
File Edit Sketch Tools Help
BMW.ino CHEVROLET.ino CITROEN.ino CKP_PROGRAMABLE
1 #define M57D30 1//
2 #define M57D30 2//
3 #define B46A20 3//
4 #define B47D20 4
5 #define B58B30 5
6 #define M50B25 6
7 #define M54B25 7
8
9
10 void bmv_menu(){
11   switch(switch_menu_bmv){
12
13     case 1:
14       pantalla_M57D30();
15       lectura_botones();
16       if(pulsado==derecha){delay(200);
17         switch_menu_bmv=2;
18       if(pulsado==enter){delay(500);
19         activacion=1;
20         lectura_botones();
21         if(pulsado==enter){
22           while(pulsado==enter){
23             lectura_botones();
24           }
25         }
26         menu=0;
27         activacion=0;
28         switch_menu_bmv=1;

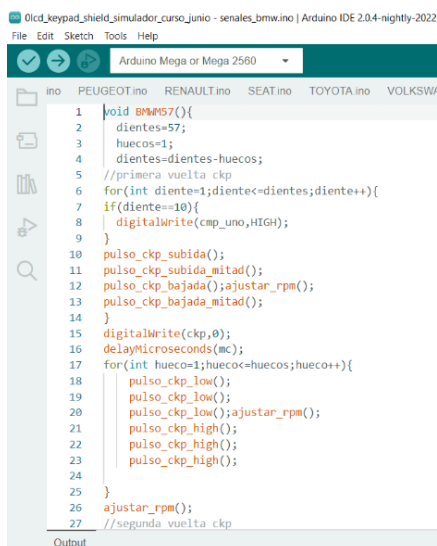
```

### 3.1.7 Señales de Vehículos BMW

En la figura 38 se observa la programación de las señales que se enviarán y recibirán el banco de prueba por parte de la marca “BMW”.

**Figura 38**

#### Señales de Vehículos BWM



```

Arduino Mega or Mega 2560
File Edit Sketch Tools Help
ino PEUGEOT.ino RENAULT.ino SEAT.ino TOYOTA.ino VOLKSWA
1 void BMW57(){
2   dientes=57;
3   huecos=1;
4   dientes=dientes-huecos;
5   //primera vuelta ckp
6   for(int diente=1;diente<=dientes;diente++){
7     if(diente==10){
8       digitalWrite(cmp_uno,HIGH);
9     }
10    pulso_ckp_subida();
11    pulso_ckp_subida_mitad();
12    pulso_ckp_bajada();ajustar_rpm();
13    pulso_ckp_bajada_mitad();
14  }
15  digitalWrite(ckp,0);
16  delayMicroseconds(mc);
17  for(int hueco=1;hueco<=huecos;hueco++){
18    pulso_ckp_low();
19    pulso_ckp_low();
20    pulso_ckp_low();ajustar_rpm();
21    pulso_ckp_high();
22    pulso_ckp_high();
23    pulso_ckp_high();
24  }
25  }
26  ajustar_rpm();
27  //segunda vuelta ckp

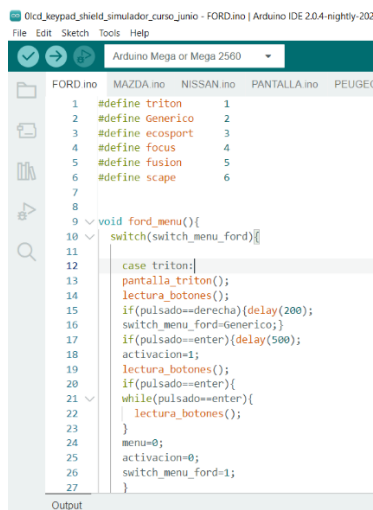
```

### 3.1.8 Programación de Vehículos Ford en Arduino

En la figura 39 se procede con el siguiente modelo de autos “Ford” donde será dividida en 2 partes para una mejor organización.

**Figura 39**

#### Programación De Vehículos Ford



```

1 #define triton 1
2 #define Generico 2
3 #define ecosport 3
4 #define focus 4
5 #define fusion 5
6 #define scape 6
7
8
9 void ford_menu(){
10 switch(switch_menu_ford){
11
12 case triton:
13 pantalla_triton();
14 lectura_botones();
15 if(pulsado==derecha){delay(200);
16 switch_menu_ford=Generico;
17 if(pulsado==enter){delay(500);
18 activacion=1;
19 lectura_botones();
20 if(pulsado==enter){
21 while(pulsado==enter){
22 lectura_botones();
23 }
24 menu=0;
25 activacion=0;
26 switch_menu_ford=1;
27 }

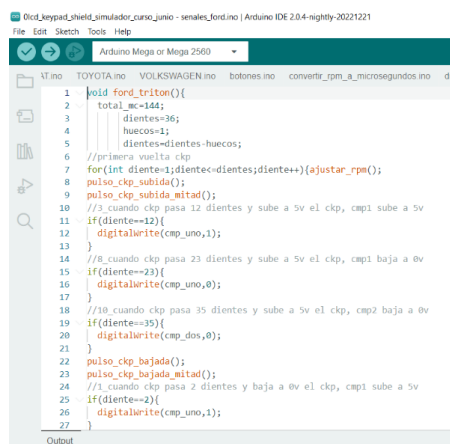
```

### 3.1.9 Señales de Vehículos Ford

En la figura 40 se observa la programación de las señales que se enviarán y recibirán el banco de prueba por parte de la marca “Ford”.

**Figura 40**

#### Señales de Vehículos Ford



```

1 void ford_triton(){
2 total_mc=144;
3 dientes=36;
4 huecos=1;
5 dientes=dientes-huecos;
6 //primera vuelta ckp
7 for(int diente=1;diente<dientes;diente++){ajustar_rpm();
8 pulso_ckp_subida();
9 pulso_ckp_subida_mitad();
10 //3 cuando ckp pasa 12 dientes y sube a 5v el ckp, cmp1 sube a 5v
11 if(diente==12){
12 digitalWrite(cmp_uno,1);
13 }
14 //8 cuando ckp pasa 23 dientes y sube a 5v el ckp, cmp1 baja a 0v
15 if(diente==23){
16 digitalWrite(cmp_uno,0);
17 }
18 //18 cuando ckp pasa 35 dientes y sube a 5v el ckp, cmp2 baja a 0v
19 if(diente==35){
20 digitalWrite(cmp_dos,0);
21 }
22 pulso_ckp_bajada();
23 pulso_ckp_bajada_mitad();
24 //1 cuando ckp pasa 2 dientes y baja a 0v el ckp, cmp1 sube a 5v
25 if(diente==2){
26 digitalWrite(cmp_uno,1);
27 }

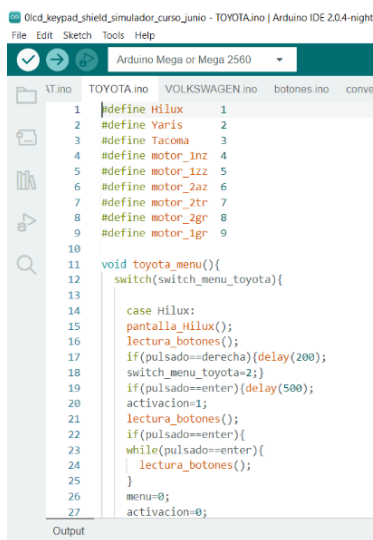
```

### 3.1.10 Programación de Vehículos Toyota en Arduino

En la figura 41 se procede con el siguiente modelo de autos “Toyota” donde será dividida en 2 partes para una mejor organización.

**Figura 41**

#### Programación de Vehículos Toyota



```

0lcd_keypad_shield_simulador_curso_junio - TOYOTA.ino | Arduino IDE 2.0.4-night
File Edit Sketch Tools Help
Arduino Mega or Mega 2560
TOYOTA.ino VOLKSWAGEN.ino botones.ino conve
1 #define Hilux 1
2 #define Yaris 2
3 #define Tacoma 3
4 #define motor_1nz 4
5 #define motor_1zz 5
6 #define motor_2az 6
7 #define motor_2tr 7
8 #define motor_2gr 8
9 #define motor_1gr 9
10
11 void toyota_menu(){
12   switch(switch_menu_toyota){
13
14     case Hilux:
15       pantalla_Hilux();
16       lectura_botones();
17       if(pulsado==derecha){delay(200);
18         switch_menu_toyota=2;}
19       if(pulsado==enter){delay(500);
20         activacion=1;
21         lectura_botones();
22         if(pulsado==enter){
23           while(pulsado==enter){
24             lectura_botones();
25           }
26         }
27         menu=0;
28         activacion=0;

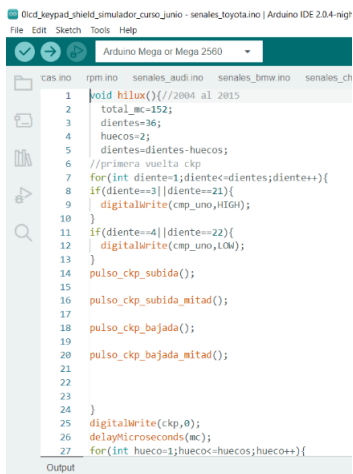
```

### 3.1.11 Señales de Vehículos Toyota

En la figura 42 se observa la programación de las señales que se enviarán y recibirán el banco de prueba por parte de la marca “Toyota”.

**Figura 42**

#### Señales de Vehículos Toyota



```

0lcd_keypad_shield_simulador_curso_junio - senales_toyota.ino | Arduino IDE 2.0.4-ngp
File Edit Sketch Tools Help
Arduino Mega or Mega 2560
cas.ino rpm.ino senales_audi.ino senales_bmw.ino senales_ch
1 void hilux(){//2004 al 2015
2   total_mc=152;
3   dientes=36;
4   huecos=2;
5   dientes=dientes-huecos;
6   //primera vuelta ckp
7   for(int diente=1;diente<dientes;diente++){
8     if(diente==3||diente==21){
9       digitalWrite(cmp_uno,HIGH);
10    }
11    if(diente==4||diente==22){
12      digitalWrite(cmp_uno,LOW);
13    }
14    pulso_ckp_subida();
15
16    pulso_ckp_subida_mitad();
17
18    pulso_ckp_bajada();
19
20    pulso_ckp_bajada_mitad();
21
22
23  }
24
25  digitalWrite(ckp,0);
26  delayMicroseconds(mc);
27  for(int hueco=1;hueco<huecos;hueco++){

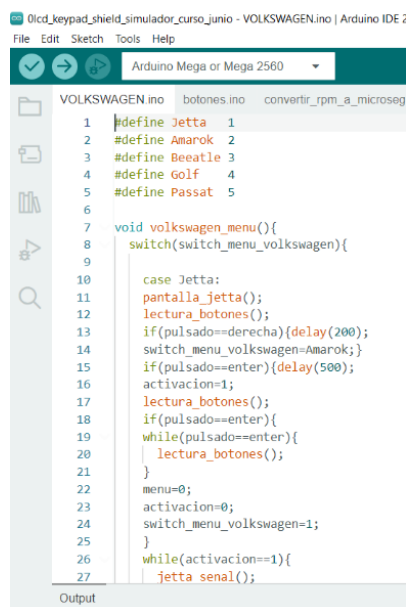
```

### 3.1.12 Programación de Vehículos Volkswagen en Arduino

En la figura 43 se procede con el siguiente modelo de autos “Volkswagen” donde será dividida en 2 partes para una mejor organización.

**Figura 43**

#### Programación de Vehículos Volkswagen



```

VOLKSWAGEN.ino  botones.ino  convertir_rpm_a_microseg

1  #define Jetta  1
2  #define Amarok  2
3  #define Beetle  3
4  #define Golf  4
5  #define Passat  5
6
7  void volkswagen_menu(){
8  switch(switch_menu_volkswagen){
9
10     case Jetta:
11     pantalla_jetta();
12     lectura_botones();
13     if(pulsado==derecha){delay(200);
14     switch_menu_volkswagen=Amarok;}
15     if(pulsado==enter){delay(500);
16     activacion=1;
17     lectura_botones();
18     if(pulsado==enter){
19     while(pulsado==enter){
20     lectura_botones();
21     }
22     menu=0;
23     activacion=0;
24     switch_menu_volkswagen=1;
25     }
26     while(activacion==1){
27     jetta_senal();

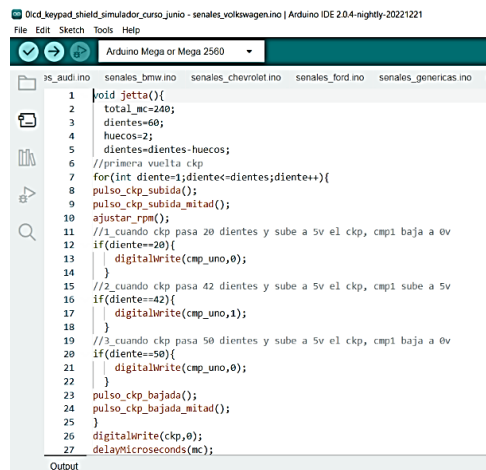
```

### 3.1.13 Señales de Vehículos Volkswagen

En la figura 44 se observa la programación de las señales que se enviarán y recibirán el banco de prueba por parte de la marca “Volkswagen”

**Figura 44**

#### Señales de Vehículos Volkswagen



```

js_audi.ino  senales_bmw.ino  senales_chevrolet.ino  senales_ford.ino  senales_genericas.ino
1  void jetta(){
2  total_mc=240;
3  dientes=60;
4  huecos=2;
5  dientes=dientes-huecos;
6  //primera vuelta ckp
7  for(int diente=1;diente<=dientes;diente++){
8  pulso_ckp_subida();
9  pulso_ckp_subida_mitad();
10  ajustar_rpm();
11  //1_cuando ckp pasa 20 dientes y sube a 5v el ckp, cmp1 baja a 0v
12  if(diente==20){
13  digitalWrite(cmp_uno,0);
14  }
15  //2_cuando ckp pasa 42 dientes y sube a 5v el ckp, cmp1 sube a 5v
16  if(diente==42){
17  digitalWrite(cmp_uno,1);
18  }
19  //3_cuando ckp pasa 50 dientes y sube a 5v el ckp, cmp1 baja a 0v
20  if(diente==50){
21  digitalWrite(cmp_uno,0);
22  }
23  pulso_ckp_bajada();
24  pulso_ckp_bajada_mitad();
25  }
26  digitalWrite(ckp,0);
27  delay(microseconds(mc);

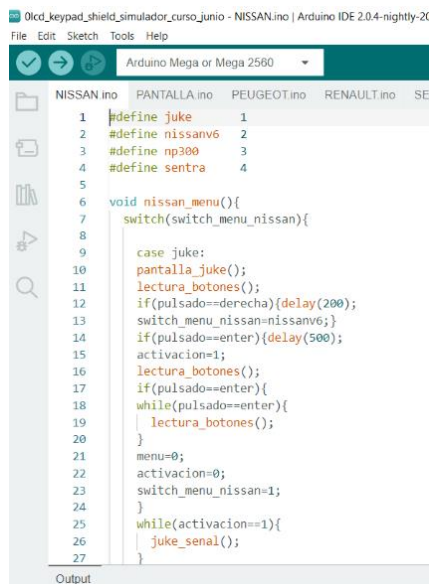
```

### 3.1.14 Programación de Vehículos Nissan en Arduino

En la figura 45 se procede con el siguiente modelo de autos “Nissan” donde será dividida en dos partes para una mejor organización.

**Figura 45**

#### Programación de Vehículos Nissan



```

1 #define juke 1
2 #define nissanv6 2
3 #define np300 3
4 #define sentra 4
5
6 void nissan_menu(){
7   switch(switch_menu_nissan){
8
9     case juke:
10    pantalla_juke();
11    lectura_botones();
12    if(pulsado==derecha){delay(200);
13    switch_menu_nissan=nissanv6;}
14    if(pulsado==enter){delay(500);
15    activacion=1;
16    lectura_botones();
17    if(pulsado==enter){
18    while(pulsado==enter){
19    | lectura_botones();
20    }
21    menu=0;
22    activacion=0;
23    switch_menu_nissan=1;
24    }
25    while(activacion==1){
26    | juke_senal();
27    }
28  }
29 }

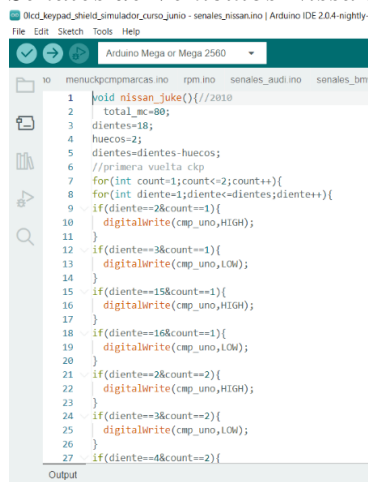
```

### 3.1.15 Señales de Vehículos Nissan

En la figura 46 se observa la programación de las señales que se enviarán y recibirán el banco de prueba por parte de la marca “Nissan”.

**Figura 46**

#### Señales de Vehículos Nissan



```

1 void nissan_juke(){//2010
2   total_ac=00;
3   dientes=18;
4   huecos=2;
5   dientes=dientes-huecos;
6   //primera vuelta ckp
7   for(int count=1;count<=2;count++){
8   for(int diente=1;diente<=dientes;diente++){
9   | if(diente==2&&count==1){
10  | digitalwrite(cmp_uno,HIGH);
11  | }
12  | if(diente==3&&count==1){
13  | digitalwrite(cmp_uno,LOW);
14  | }
15  | if(diente==15&&count==1){
16  | digitalwrite(cmp_uno,HIGH);
17  | }
18  | if(diente==16&&count==1){
19  | digitalwrite(cmp_uno,LOW);
20  | }
21  | if(diente==2&&count==2){
22  | digitalwrite(cmp_uno,HIGH);
23  | }
24  | if(diente==3&&count==2){
25  | digitalwrite(cmp_uno,LOW);
26  | }
27  | if(diente==4&&count==2){

```

### 3.1.16 Programación de Vehículos Chevrolet en Arduino

En la figura 47 se procede con el siguiente modelo de autos “Chevrolet” donde será dividida en 2 partes para una mejor organización.

**Figura 47**

#### Programación de Vehículos Chevrolet



```

0lcd_keypad_shield_simulador_curso_junio - CHEVROLET.ino | Arduino IDE 2.0.4-night
File Edit Sketch Tools Help
Arduino Mega or Mega 2560
CHEVROLET.ino CITROEN.ino CKP_PROGRAMABLE.ino
1 #define chevy 1
2 #define luvdmax 2
3 #define aveo 3
4 #define captiva 4
5 #define silverado 5
6 #define spark 6
7
8 void chevrolet_menu(){
9     switch(switch_menu_chevrolet){
10
11         case 1:
12             pantalla_chevy();
13             lectura_botones();
14             if(pulsado==derecha){delay(200);
15                 switch_menu_chevrolet=2;}
16             if(pulsado==enter){delay(500);
17                 activacion=1;
18                 lectura_botones();
19                 if(pulsado==enter){
20                     le(pulsado==enter){
21                         lectura_botones();
22                     }
23                 }
24                 menu=0;
25                 activacion=0;
26                 switch_menu_chevrolet=1;
27             }
28         }
29     }
30     while(activacion==1){

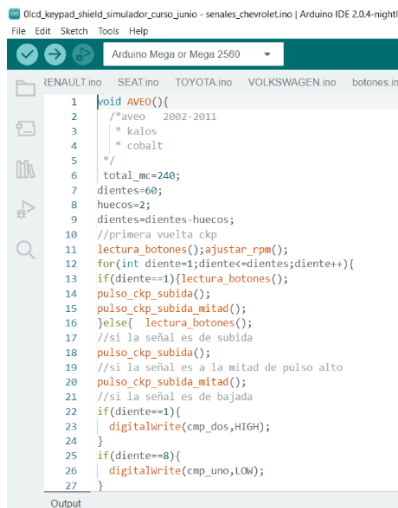
```

### 3.1.17 Señales de Vehículos Chevrolet

En la figura 48 se observa la programación de las señales que se enviarán y recibirán el banco de prueba por parte de la marca “Chevrolet”.

**Figura 48**

#### Señales de Vehículos Chevrolet



```

0lcd_keypad_shield_simulador_curso_junio - senales_chevrolet.ino | Arduino IDE 2.0.4-night
File Edit Sketch Tools Help
Arduino Mega or Mega 2560
RENAULT.ino SEAT.ino TOYOTA.ino VOLKSWAGEN.ino botones.ino
1 void AVEO(){
2     //aveo 2002-2011
3     * kalos
4     * cobalt
5     */
6     total_mc=240;
7     dientes=60;
8     huecos=2;
9     dientes=dientes-huecos;
10    //primera vuelta ckp
11    lectura_botones();ajustar_rpm();
12    for(int dientes=1;diente<dientes;diente++){
13        if(diente==1){lectura_botones();
14            pulso_ckpt_subida();
15            pulso_ckpt_subida_mitad();
16        }else{ lectura_botones();
17            //si la señal es de subida
18            pulso_ckpt_subida();
19            //si la señal es a la mitad de pulso alto
20            pulso_ckpt_subida_mitad();
21            //si la señal es de bajada
22            if(diente==1){
23                digitalWrite(cmp_dos,HIGH);
24            }
25            if(diente==8){
26                digitalWrite(cmp_uno,LOW);
27            }

```

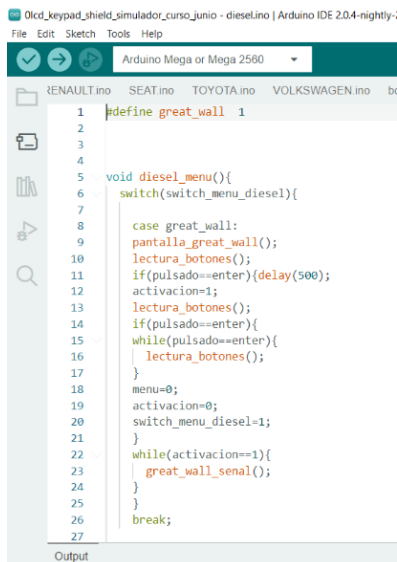


### 3.1.18 Programación de Vehículos con Sistema Diesel

En la figura 49 se procede con los modelos de autos que manejan un sistema distinto al de gasolina, es decir a los automóviles con sistema diésel.

**Figura 49**

#### *Programación para Vehículos de Sistema Diesel*



```

1  #define great_wall 1
2
3
4
5  void diesel_menu(){
6    switch(switch_menu_diesel){
7
8      case great_wall:
9        pantalla_great_wall();
10       lectura_botones();
11       if(pulsado==enter){delay(500);
12       activacion=1;
13       lectura_botones();
14       if(pulsado==enter){
15         while(pulsado==enter){
16           lectura_botones();
17         }
18         menu=0;
19         activacion=0;
20         switch_menu_diesel=1;
21       }
22       while(activacion==1){
23         great_wall_senal();
24       }
25     }
26     break;
27

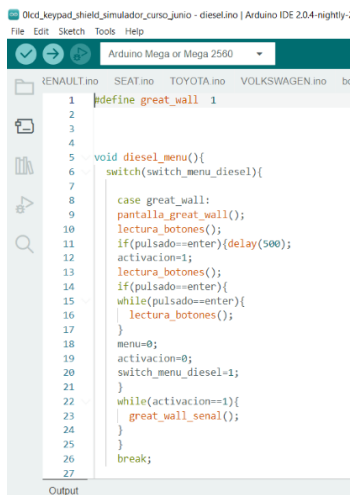
```

### 3.1.19 Programación de Sensor CKP

En la figura 50 se encuentra la programación detrás del sensor CKP con una configuración de rueda fónica de 36 – 2.

**Figura 50**

#### *Programación de sensor CKP*



```

1  #define great_wall 1
2
3
4
5  void diesel_menu(){
6    switch(switch_menu_diesel){
7
8      case great_wall:
9        pantalla_great_wall();
10       lectura_botones();
11       if(pulsado==enter){delay(500);
12       activacion=1;
13       lectura_botones();
14       if(pulsado==enter){
15         while(pulsado==enter){
16           lectura_botones();
17         }
18         menu=0;
19         activacion=0;
20         switch_menu_diesel=1;
21       }
22       while(activacion==1){
23         great_wall_senal();
24       }
25     }
26     break;
27

```

### 3.1.20 Programación del Menú de Marcas.

En la figura 51 se procede a crear un menú el cual facilitara seleccionar la marca que se requiere trabajar.

**Figura 51**

#### Menú de Marcas

```

1 void sensorckp_menu(){
2   switch(menu_marcas){
3
4     case 1: //caso chevrolet
5       chevrolet_principal(); // ejecuta la funcion chevrolet_principal(); ubicada en la pestaña PANTALLA
6       lectura_botones(); // lectura de botones
7       if(pulsado==derecha){delay(200); // pulsacion abajo entra a la marca toyota
8         menu_marcas=2;}
9       if(pulsado==enter){delay(300);
10        menu=1;
11        lectura_botones();
12        if(pulsado==enter){
13          while(pulsado==enter){
14            lectura_botones();
15          }
16          menu=-1;
17        }
18        while(menu==1){ // se mantiene en esa opcion ejecutando
19          chevrolet_menu(); // la funcion chevrolet_menu(); de la pestaña chevrolet
20        }
21        break;
22
23
24        case 2:
25          toyota_principal(); //se imprime en la pantalla
26          lectura_botones(); //se lee el estado de la entrada A0 del arduino
27          if(pulsado==izquierda){delay(200);

```

### 3.1.21 Simulación de RPM

En la figura 52 se realiza el apartado de simulación de las RPM de los vehículos.

**Figura 52**

#### Simulación de RPM

```

1 void ajustar_rpm(){lectura_botones();
2   if(pulsado==abajo){delay(100);
3     mc=mc-50;
4     convertir_mc_rp();
5     ajustarrpm();
6     if(rpm<999||rpm>=900){
7       lcd.setCursor(0,1);
8       lcd.print(F("      "));
9     }
10    pantalla_rpm();
11  }
12  if(pulsado==arriba){delay(100);
13    mc=mc+50;
14    ajustarrpm();
15    convertir_mc_rp();
16    if(rpm<999&rpm>=900||rpm<=99&rpm>=90){
17      lcd.setCursor(0,1);
18      lcd.print(F("      "));
19    }
20    pantalla_rpm();
21  }
22 }
23 void ajustarrpm(){
24   if(mc<50){
25     mc=50;
26   }
27 }

```

### 3.2 Circuito Eléctrico del Banco de Pruebas

En este apartado se puede encontrar de manera detallada cada circuito que se realizó para llevar a cabo el banco de pruebas.

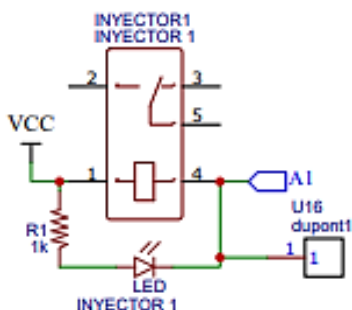
El diodo led es un elemento que encuentra en la mayoría del diagrama eléctrico puesto que es el encargado de cerrar el circuito, cuando se emita la señal desde la Ecu se manifiesta a través del diodo led.

#### 3.2.1 Sistema de Inyectores

Se comenzará con el apartado de inyectores, en la figura 53 se observa que el banco permite probar hasta 8 inyectores, donde la señal que emita la Ecu será reflejada en el diodo led correspondiente al número del inyector.

**Figura 53**

*Sistema Inyectores*

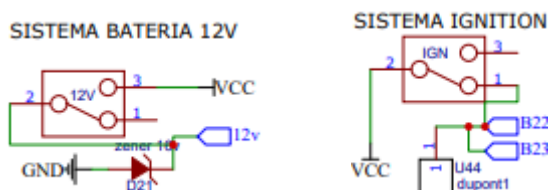


#### 3.2.2 Sistema de Batería de 12V y Sistema de Ignición

En la figura 54 se muestra el sistema de batería de 12V y sistema de ignición donde se observa una entrada de tierra a un diodo Zener mediante un interruptor para corriente continua o para la salida uno.

**Figura 54**

*Sistema de batería 12 V y sistema de ignición*

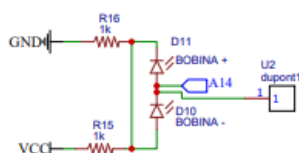


### 3.2.3 Sistema de Bobinas

En la figura 55 se observa el circuito que el banco permite comprobar hasta 8 bobinas, donde la señal que emita la Ecu será reflejada en los diodos leds correspondiente al número de la bobina.

**Figura 55**

*Sistema de Bobinas*



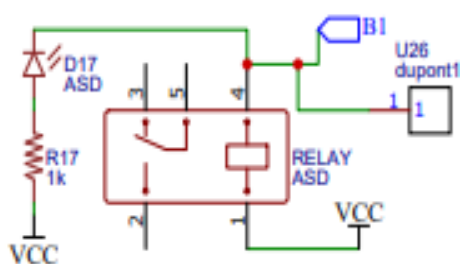
### 3.2.4 Sistema de ASD y Sistema Check Engine

En la figura 56 se observa el circuito del sistema ASD (Auto Shut Down) y del apartado del Check Engine donde para elaborar este circuito se hizo el uso de: Resistencia de 1k ohmios, diodo Zenner.

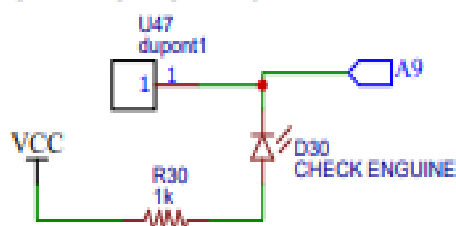
**Figura 56**

*Sistema ASD y Sistema Check Engine*

**SISTEMA ASD**



**SISTEMA CHECK ENGINE**

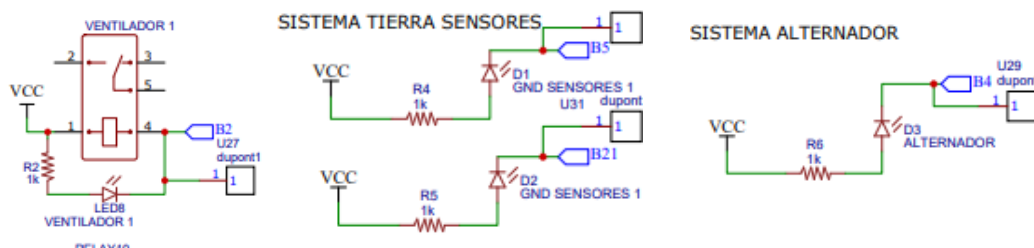


### 3.2.5 Sistema Ventiladores, Tierra de Sensores y Alternador

En la figura 57 se visualiza Sistema Ventiladores, Sistema Tierra Sensores y Sistema, se tiene una entrada de corriente continua que se dirige a una resistencia de 1K Ohmios y pasa por un diodo led.

**Figura 57**

*Sistema Ventiladores, Sistema Tierra Sensores y Sistema Alternador*



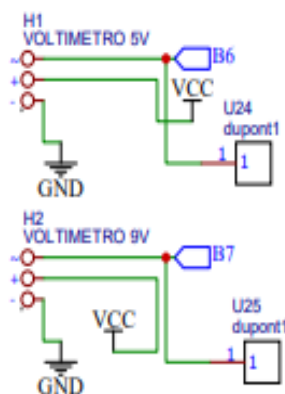
### 3.2.6 Sistema Voltaje Sensores

En la figura 58 Se conecta directamente con las pantallas led que se tiene, simulando un voltímetro de 5V y otro de 9V. Se visualiza una entrada de corriente continua, una tierra y una entrada B6 hacia el voltímetro 5V.

**Figura 58**

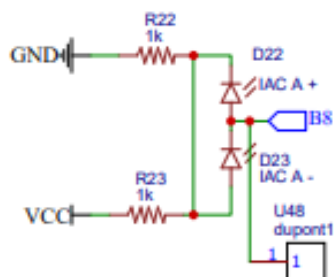
*Sistema Voltaje Sensores*

SISTEMA VOLTAJE SENSORES



### 3.2.7 Válvula IAC

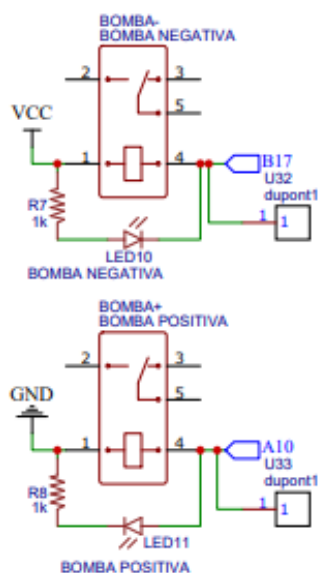
En la figura 59 se encuentra la válvula en la que trabaja el ralentí el vehículo, se logra visualizar una entrada de voltaje de corriente continua y una tierra, las dos entradas pasan por una resistencia 1k individualmente, en donde se conectan a un diodo led y a su vez al U48 dupont.

**Figura 59***Sistema Válvula IAC***3.2.8 Sistema de Bomba**

En la figura 60 se puede comprobar si la bomba de combustible se encuentra en buen estado o en mal estado, se puede comprobar bombas negativas y bombas positivas de un vehículo.

**Figura 60***Sistema Bomba*

SISTEMA BOMBA (+) \_ (-)

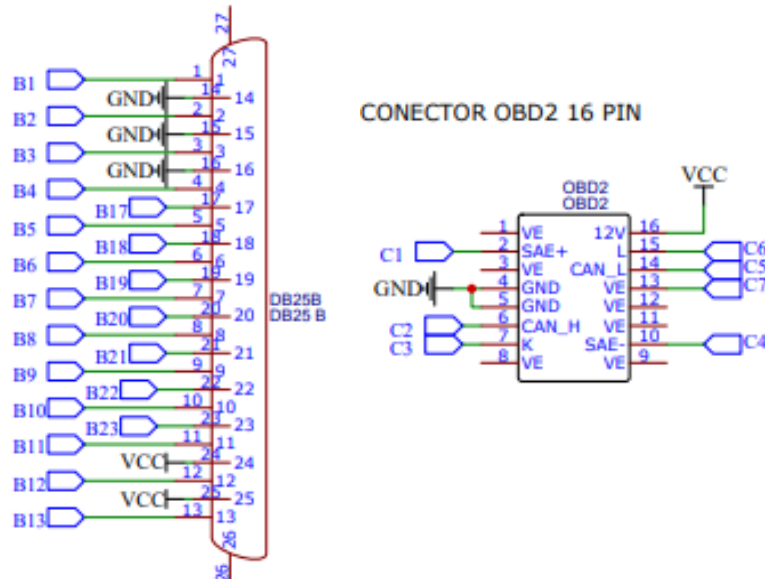
**3.2.9 Conector DB25 B y Conector OBD2 16 Pin**

En la figura 61 se encuentra los conectores del banco de prueba, y a su vez se visualiza la entrada de B1 hasta B25 y tres salidas de tierra y dos de corriente continua.

**Figura 61**

*Conector DB25 B y Conector OBD2 16 Pin*

CONECTOR DB25 B

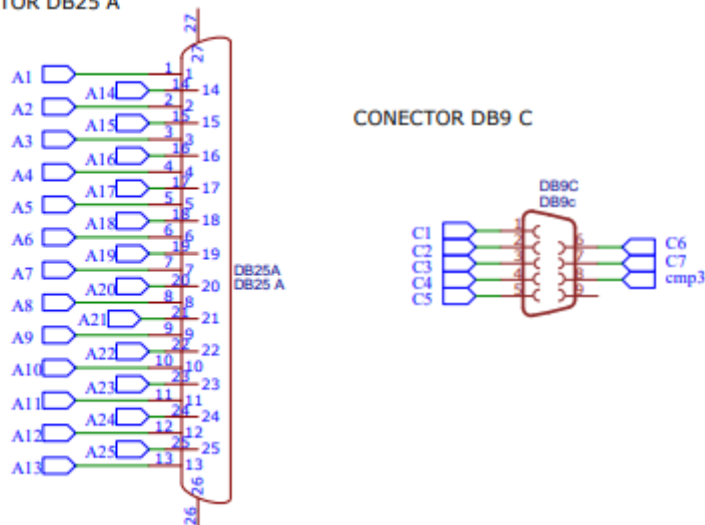


En la figura 62 se encuentra los conectores del banco de prueba, y a su vez se visualiza la entrada de B1 hasta B25, además se visualiza las entradas del conector DB89C.

**Figura 62**

*Conector DB25 A y Conector DB89 C*

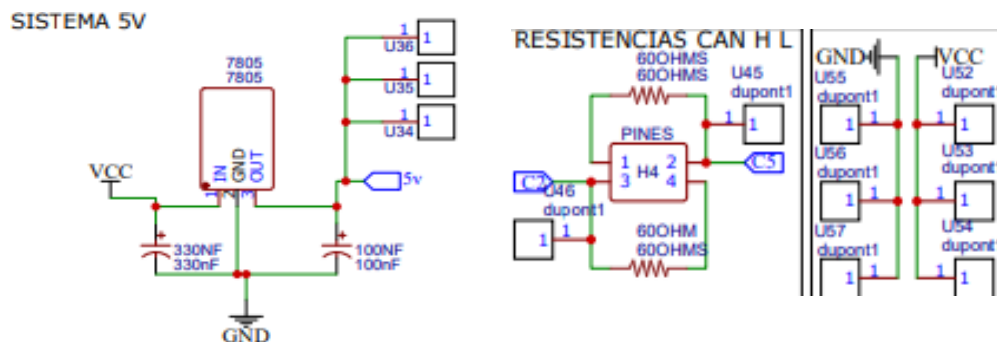
CONECTOR DB25 A



En la figura 63 se visualiza el sistema 5V y resistencias Can H L donde la entrada de 5 V se visualiza una entrada de corriente continua donde pasa por un capacitor de 330 nF y un capacitor de 100 nF.

**Figura 63**

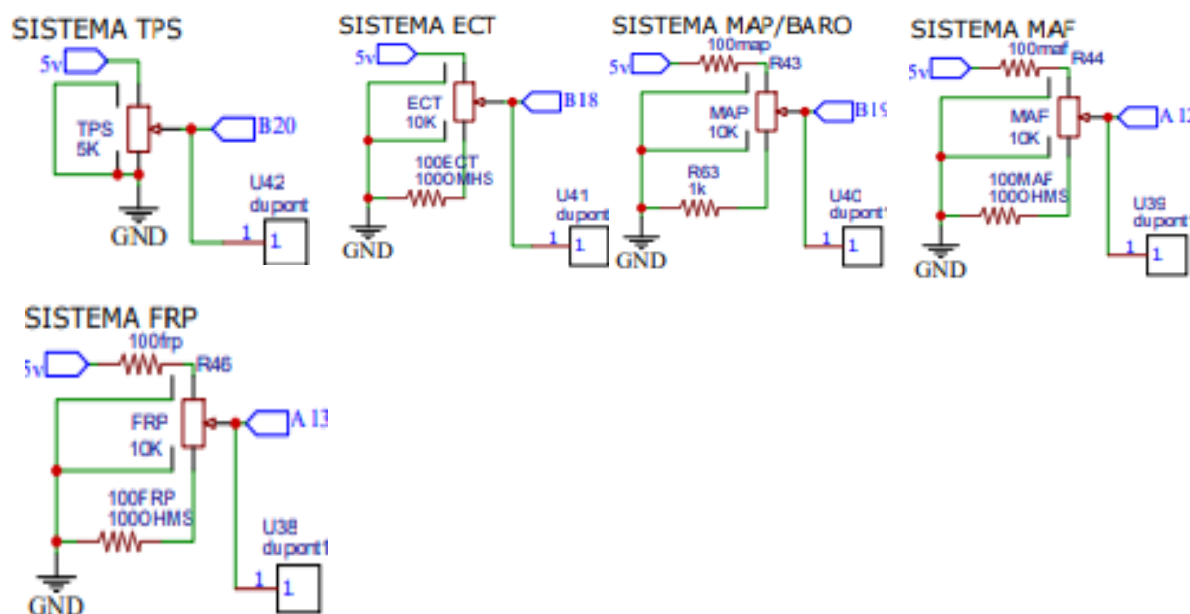
*Sistema 5V y Resistencias Can H L*



En la figura 64 se encuentra los diferentes circuitos de los sensores TPS, MAP, MAF, FRP, donde la entrada de todos los sensores se la hace por medio de 5V pasando por una resistencia y posee una salida de tierra.

**Figura 64**

*Sistema TPS, Sistema ECT, Sistema MAP, Sistema MAF y Sistema FRP*

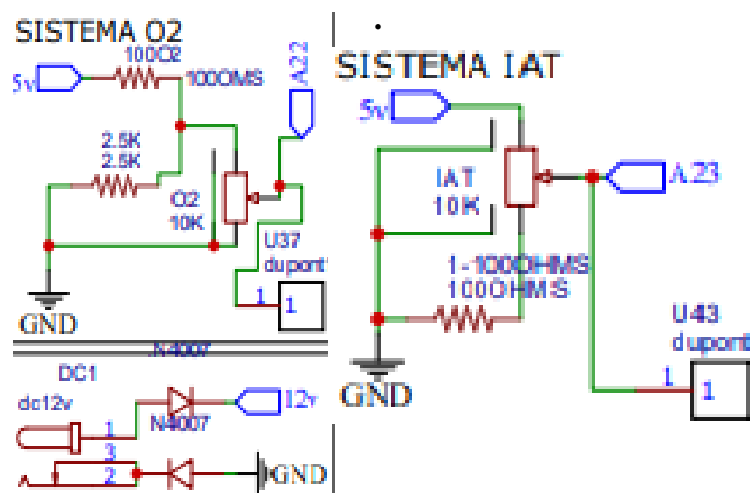


En la figura 65 se visualiza en el sistema O2 una entrada de 5v que pasa por una resistencia de 10DCR y por una resistencia de 2.5K terminando en tierra.



**Figura 65**

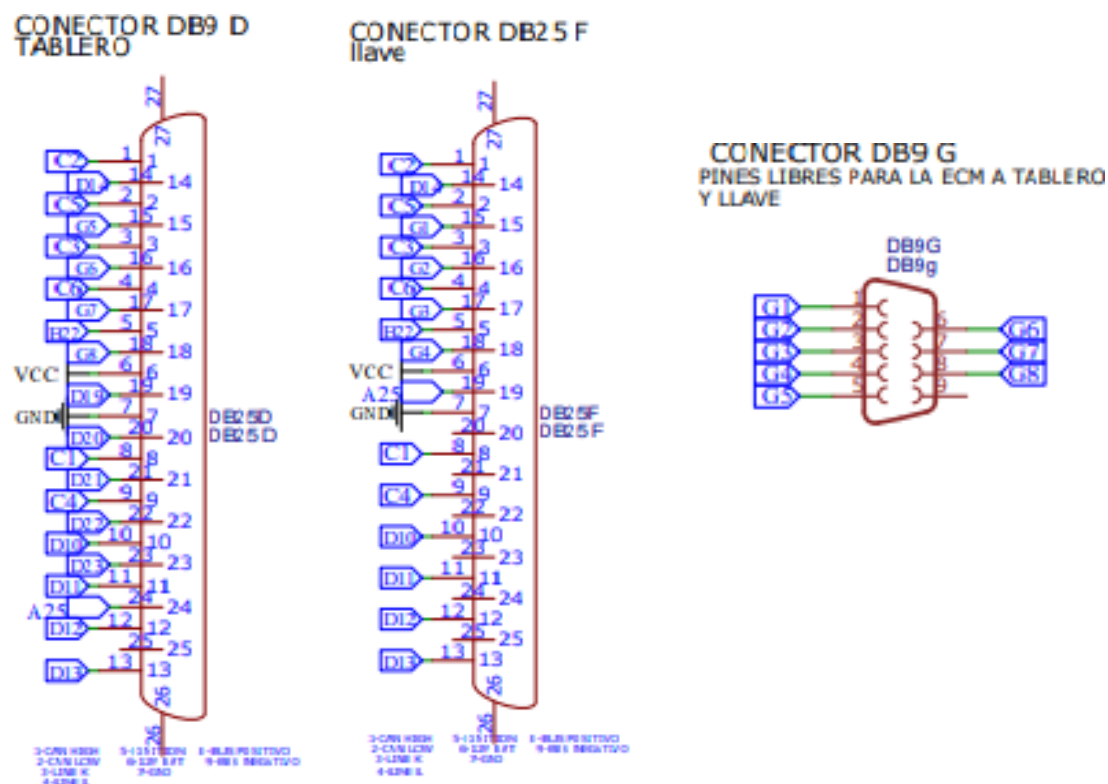
*Sistema O2 y Sistema IAT*



En la figura 66 se visualiza los conectores correspondientes a DB89D, DB25F, DB89G del banco de pruebas.

**Figura 66**

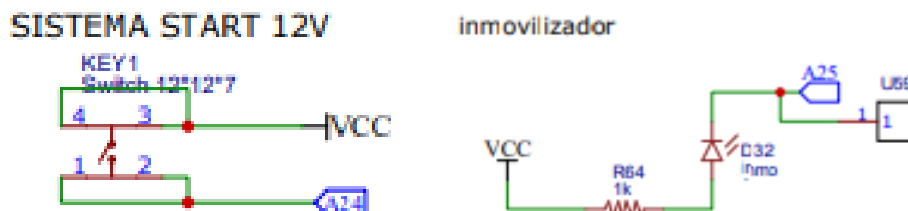
*Conector DB89 D tablero, Conector DB25 F llave & Conector DB89 G*



En la figura 67 se visualiza una entrada de corriente continua donde se realiza la conexión mediante cuatro puntos hacia un interruptor, en el inmovilizador se visualiza una entrada de corriente continua donde pasa por una resistencia de 1K y luego pasa por un diodo led.

**Figura 67**

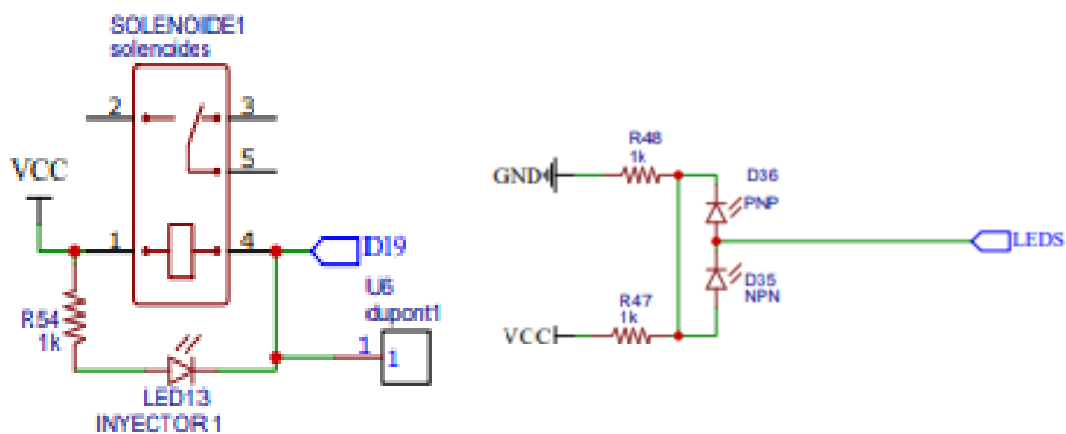
*Sistema Start 12V e Inmovilizador*



En la figura 68 se visualiza los solenoides de transmisión donde se realiza una entrada de corriente continua y pasa por una resistencia del 1K dando continuidad a un led correspondiente al inyector.

**Figura 68**

*Solenoides de Transmisión*

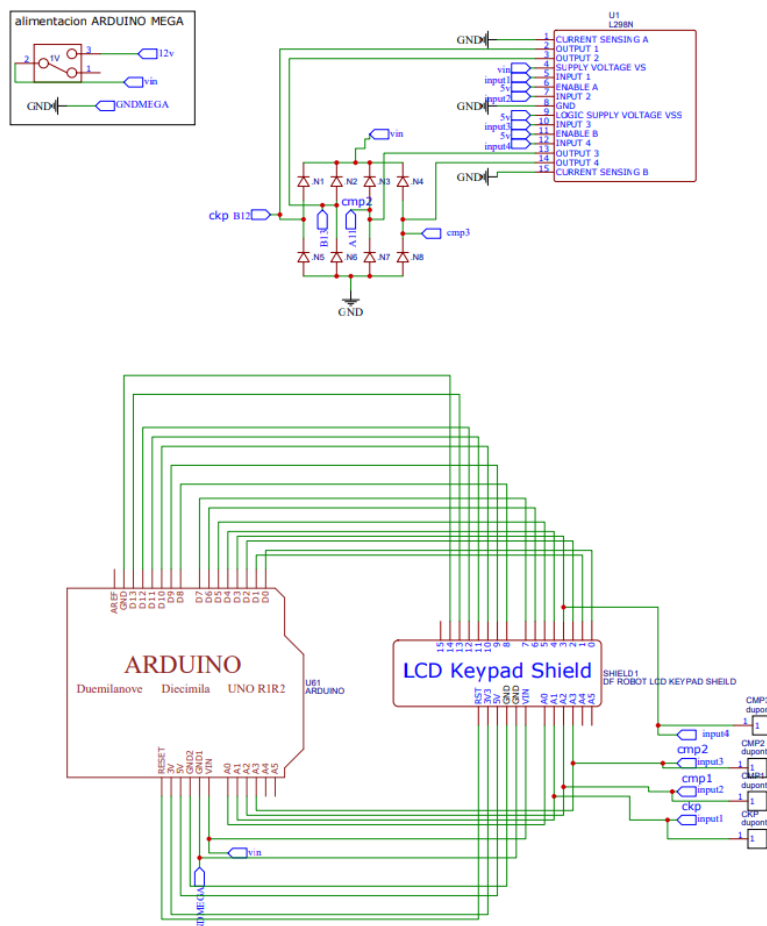


En la figura 69 se observa las conexiones que tiene la placa de Arduino mega con la pantalla LCD.

**Figura 69**

### Conexión Arduino Mega

CONEXION ARDUINO MEGA



Se debe tener en cuenta que cada sistema está conectado al Arduino el cual contiene la programación para poder activar cada señal que se requiera demostrar de la Ecu.

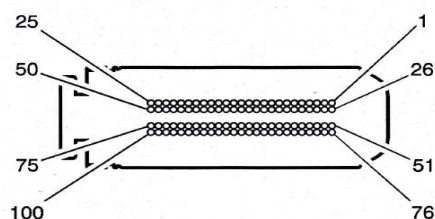
### 3.3 Pin Out Chevrolet Sail

El pin out facilita al momento de realizar alguna reparación de algún cortocircuito o simplemente para algún tipo de diagnóstico, por lo que son las conexiones eléctricas del vehículo; en la siguiente figura se puede observar con detalle cada pin out de la ECU y las conexiones, con el fin de facilitar la búsqueda por cortocircuitos o fallas en circuitos eléctricos grandes, la red CAN Bus nace como una solución a la comunicación entre los distintos dispositivos de los vehículos ya que es la encargada de llevar información a todas partes del

mismo, en la figura 70 se puede visualizar el pin out del Sail, posee 100 pines y en la tabla 1 el significado cada pin.

**Figura 70**

*Pin Out del Vehículo Sail*



Fuente: (Villa, 2016)

**Tabla 1**

*Significado de Cada Uno de los Pines Out del Sail*

Clavija	Tamaño	Circuito	Función	Opción
1	0.5	2123	Señal de control de ignición – Cilindro 4	-
2	0.5	2122	Señal de control de ignición – Cilindro 3	-
3	0.5	2124	Señal de control de ignición – Cilindro 2	-
4	0.5	150	Tierra	-
5	0.35	630	Señal del sensor de posición del árbol de levas	-
6	-	-	No se utiliza	-
7	0.35	919	Señal de sensor de temperatura del múltiple	-
8	0.5	435	Señal del sensor de posición de la válvula EGR	-
9	-	-	No se utiliza	-
10	0.5	908	Señal del sensor de oxígeno caliente (HO2S) trasero	-
11	0.35	432	Señal del sensor de presión absoluta del distribuidor	-
12	0.35	9469	Baja referencia	-
13	0.35	918	Referencia 5V	-
14	0.35	469	Baja referencia	-
15	0.5	1271	Baja referencia	-
16	0.5	1664	Referencia 5V	-
Clavija	Tamaño	Circuito	Función	Opción
17	0.5	140	Voltaje positivo de batería	-
18	0.35	573	Referencia 5V	-
19	0.35	916	Baja referencia	-
20	0.5	907	Baja referencia	-
21	0.5	3201	Referencia 5V	-
22-23	-	-	No se utiliza	-
24	0.5	7335	Control bajo del motor del actuador del acelerador	-
25	0.5	5292	Voltaje de encendido	-
26	0.5	2121	Señal de control de ignición – Cilindro 1	-
27	0.5	150	Tierra	-
28	05	150	Tierra	-

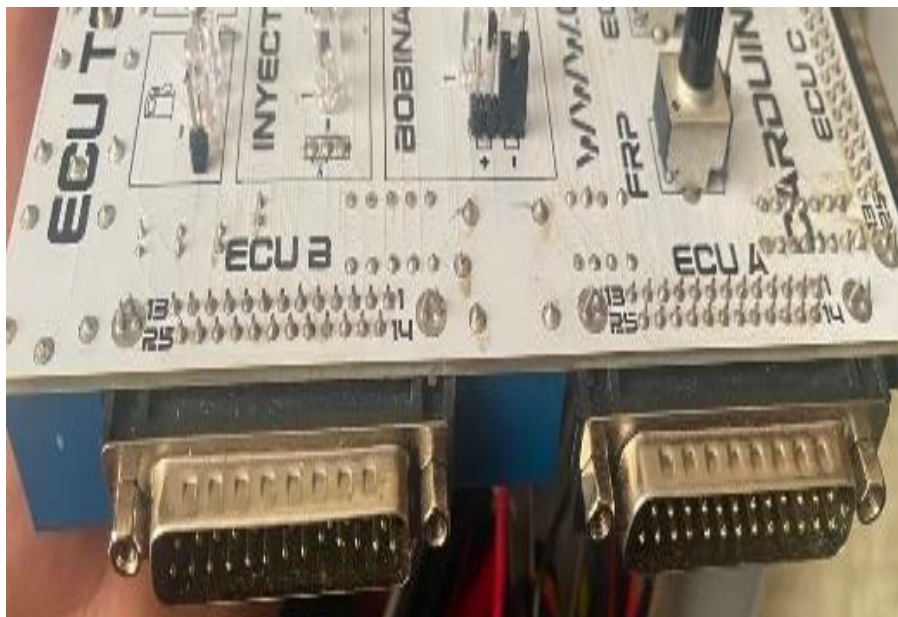
<b>Clavija</b>	<b>Tamaño</b>	<b>Circuito</b>	<b>Función</b>	<b>Opción</b>
29	0.5	150	Tierra	-
30	0.35	2832	Señal del sensor de posición del cigüeñal	-
31	-	-	No se utiliza	-
32	0.5	1661	Señal del sensor de posición del pedal	-
33	0.35	917	Baja referencia	-
34	0.5	1662	Señal del sensor de posición del pedal	-
35	0.35	3683	Señal del sensor de temperatura del evaporador de A/C	-
36	0.5	203	Señal del sensor de presión de A/C	-
37	0.5	30	Señal de sensor del nivel de combustible	-
38	-	-	No se utiliza	-
39	0.35	2761	Señal del sensor de temperatura del refrigerante	-
40	0.5	7352	Señal HO2S	-
41	-	-	No se utiliza	-
42	0.5	7355	Baja referencia	-
43	0.5	1274	Referencia 5V	-
44	0.5	1272	Baja referencia	-
45	0.35	915	Baja referencia	-
46	0.5	914	Referencia 5V	-
47	0.5	1739	Voltaje de ignición 2/3	-
48	0.5	340	Voltaje de ignición 1/2	-
49	0.5	7336	Control alto del motor del actuador del acelerador	-
50	0.5	5292	Voltaje de encendido	-
51	0.5	473	Control de relevador del ventilador de enfriamiento de velocidad alta	-
52	0.5	335	Control del relevador del ventilador de enfriamiento de velocidad baja	-
53	0.5	21	Señal del sensor de posición del pedal del freno	-
54	0.35	710	Datos seriales altos de CAN	-
55	0.35	1807	Datos seriales bajos de CAN	-
56	0.5	20	Señal del sensor de posición del pedal del freno	-
57	0.5	1319	Datos seriales	-
58	0.5	5270	Señal de la velocidad del vehículo	-
59	-	-	No se utiliza	-
60	0.5	66	Señal de solicitud de A/C	-
61-64	-	-	No se utiliza	-
65	0.5	486	Señal TPS 2	-
66	0.5	485	Señal TPS 1	-
67	0.5	3202	Baja referencia	-
68	0.5	817	Señal del sensor de velocidad del vehículo	-
69	0.35	410	Baja referencia	-
70	-	-	No se utiliza	-
71	0.35	1876	Señal del sensor de golpe	-
72	0.35	496	Señal del sensor de golpe	-
73	-	-	No se utiliza	-
74	0.75	150	Tierra	-
75	0.75	150	Tierra	-
76	0.5	7017	Señal de control de Puerto de admisión variable	-
77	0.5	459	Control del relevador del compresor de A/C	-
78	0.75	7013	Señal de control de inyector de combustible – Cilindro 2	-
79	0.75	7011	Señal de control de inyector de combustible – Cilindro 3	-
80	0.75	7010	Señal de control de inyector de combustible – Cilindro 1	-

Clavija	Tamaño	Circuito	Función	Opción
81	0.75	7012	Señal de control de inyector de combustible – Cilindro 4	-
82	-	-	No se utiliza	-
83	0.5	913	Señal de control de múltiple de admisión variable	-
84	0.5	817	Salida de señal de velocidad del vehículo	-
85	0.5	135	Salida de señal de transmisión controlada electrónicamente (ECT)	-
86	0.5	458	Señal de control de relevador de bomba combustible	-
87	0.5	902	Señal de control del relevador de arranque	EMT
88	0.5	9349	Control bajo del calentador del sensor de oxígeno delantero	-
89	-	-	No se utiliza	-
90	0.5	7354	Baja referencia de calentador	-
91	0.5	7414	Salida de señal de nivel de combustible	-
92	0.5	419	Control de la MIL	-
93	-	-	No se utiliza	-
94	0.5	5069	Señal de control del relevador principal	-
95	0.5	5372	Control de solenoide de purga de bote EVAP	-
96	-	-	No se utiliza	-
97	0.5	121	Señal de velocidad del motor	-
98	0.5	436	Control de la válvula de recirculación de gas de escape	-
99-100	-	-	No se utiliza	-

Fuente: (Villa, 2016)

### 3.4 Conexiones del Banco de Pruebas a la ECU de Sail

Para poder conectar una ECU al banco de pruebas es indispensable tener el pin out de la misma, en este caso se procede a trabajar con la ECU del Sail empleando los cables Dupont donde se realiza las conexiones desde el banco de pruebas haciendo uso de los puertos ECU A & ECU B del banco hacia los pines de la ECU, por ello es importante entender que cada pin del banco tiene una conexión específica al igual que la ECU tiene su propio pin out, en la figura 71 se puede observar las entradas de la ECU A y ECU B son entradas DB25.

**Figura 71***Puertos ECU A y ECU B*

A continuación, se detalla la entrada DB25 que se encuentra en el banco de pruebas que se denomina ECU A y se detallara la función de cada uno de los 25 pines, en la tabla 2 se puede observar el detalle de cada uno de los pines.

**Tabla 2***Detalle de ECU A*

<b>Número de Pin</b>	<b>Descripción</b>
Pin 1	Inyector 1
Pin 2	Inyector 2
Pin 3	Inyector 3
Pin 4	Inyector 4
Pin 5	Inyector 5
Pin 6	Inyector 6
Pin 7	Inyector 7
Pin 8	Inyector 8
Pin 9	Check Engine
Pin 10	Bomba Con Salida (+)
Pin 11	CMP2
Pin 12	Sensor MAF

<b>Número de Pin</b>	<b>Descripción</b>
Pin 13	Sensor FRP
Pin 14	Bobina 1
Pin 15	Bobina 2
Pin 16	Bobina 3
Pin 17	Bobina 4
Pin 18	Bobina 5
Pin 19	Bobina 6
Pin 20	Bobina 7
Pin 21	Bobina 8
Pin 22	Sensor O2
Pin 23	Sensor IAT
Pin 24	Señal Start
Pin 25	Led Inmovilizador

+

A continuación, se detalla la entrada DB25 que se encuentra en el banco de pruebas que se denomina ECU B y se especifica la función de cada uno de los 25 pines, en la tabla 3 se puede observar el detalle de cada uno de los pines.

**Tabla 3**

*Detalles de ECU B*

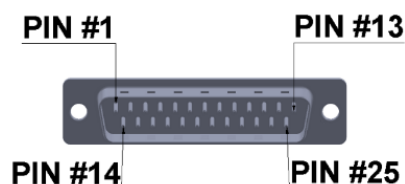
<b>Número de Pin</b>	<b>Descripción</b>
Pin 1	Asd/Main Relay
Pin 2	Fan Low
Pin 3	Fan High
Pin 4	Señal Alternador
Pin 5	Tierra Sensores I
Pin 6	5v A Sensores
Pin 7	9v A Sensores
Pin 8	A IAC
Pin 9	B IAC
Pin 10	C IAC
Pin 11	D IAC
Pin 12	CKP
Pin 13	CMP I
Pin 14	Negativo (-)
Pin 15	Negativo (-)
Pin 16	Negativo (-)
Pin 17	Bomba Con Salida (-)
Pin 18	Sensor ECT
Pin 19	Sensor MAP
Pin 20	Sensor TPS
Pin 21	Tierra Sensores 2



Número de Pin	Descripción
Pin 23	Ignition 12v
Pin 24	Bateria 12v
Pin 25	Bateria 12v

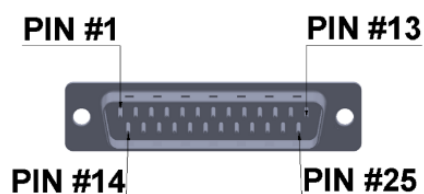
**Figura 72**

*Orden del Pin ECU A*



**Figura 73**

*Orden del Pin ECU B*



A continuación, se detalla los pines que se conectarán del banco de pruebas de la entrada ECU A a los pines de la ECU del Sail, como se detalla en la tabla 4.

**Tabla 4**

*Conexión de Ecu A - Ecu del Sail*

Conexiones del banco de prueba hacia la ECU			
Pines del banco de prueba ECU A	Descripción del banco de prueba ECU A	Pines de la ECU	Descripción de los pines de la ECU
Pin 1	Inyector 1	Pin 80	Señal de control de inyector de combustible – Cilindro 1
Pin 2	Inyector 2	Pin 78	Señal de control de inyector de combustible – Cilindro 2
Pin 3	Inyector 3	Pin 79	Señal de control de inyector de combustible – Cilindro 3

<b>Conexiones del banco de prueba hacia la ECU</b>			
<b>Pines del banco de prueba ECU A</b>	<b>Descripción del banco de prueba ECU A</b>	<b>Pines de la ECU</b>	<b>Descripción de los pines de la ECU</b>
Pin 4	Inyector 4	Pin 81	Señal de control de inyector de combustible – Cilindro 4
Pin 9	Check Engine	Pin 12	Baja referencia
Pin 10	Bomba Con Salida (+)	Pin 91	Salida de señal de nivel de combustible
Pin 11	CMP2	Pin 5	Señal del sensor de posición del árbol de levas
Pin 12	Sensor MAF	Pin 83	Señal de control de múltiple de admisión variable
Pin 13	Sensor FRP	-	
Pin 14	Bobina 1	Pin 26	Señal de control de ignición – Cilindro 1
Pin 15	Bobina 2	Pin 3	Señal de control de ignición – Cilindro 2
Pin 16	Bobina 3	Pin 2	Señal de control de ignición – Cilindro 3
Pin 17	Bobina 4	Pin 1	Señal de control de ignición – Cilindro 4
Pin 22	Sensor O2	Pin 88	Control bajo del calentador del sensor de oxígeno delantero
Pin 23	Sensor IAT	Pin 18	Referencia 5V
Pin 24	Señal Start	Pin 19	Baja referencia
Pin 25	Led Inmovilizador	Pin 20	Baja referencia

A continuación, se detalla los pines que se conectaran del banco de pruebas de la entrada ECU B a los pines de la ECU del Sail, como se detalla en la tabla 5.

**Tabla 5**

*Conexión de ECU B a la ECU del Sail*

<b>Conexiones del banco de prueba hacia la ECU</b>			
<b>Pines del banco de prueba ECU B</b>	<b>Descripción del banco de prueba ECU B</b>	<b>Pines de la ECU</b>	<b>Descripción de los pines de la ECU</b>
Pin 1	Asd/Main Relay	Pin 94	Señal de control del relevador principal

<b>Conexiones del banco de prueba hacia la ECU</b>			
<b>Pines del banco de prueba ECU B</b>	<b>Descripción del banco de prueba ECU B</b>	<b>Pines de la ECU</b>	<b>Descripción de los pines de la ECU</b>
Pin 2	Fan Low	Pin 52	Control del relevador del ventilador de enfriamiento de velocidad baja
Pin 3	Fan High	Pin 51	Control de relevador del ventilador de enfriamiento de velocidad alta
Pin 4	Señal Alternador	Pin 20	Baja referencia
Pin 5	Tierra Sensores I	Pin 4	Tierra
Pin 6	5v A Sensores	Pin 13	Referencia 5V
Pin 7	9v A Sensores	Pin 19	Baja referencia
Pin 8	A IAC	Pin 49	Control alto del motor del actuador del acelerador
Pin 12	CKP	Pin 30	Señal del sensor de posición del cigüeñal
Pin 13	CMP I	Pin 5	Señal del sensor de posición del árbol de levas
Pin 14	Negativo (-)	Pin 27	Tierra
Pin 15	Negativo (-)	Pin 28	Tierra
Pin 16	Negativo (-)	Pin 29	Tierra
Pin 17	Bomba Con Salida (-)	Pin 86	Señal de control de relevador de bomba combustible oxígeno delantero
Pin 18	Sensor ECT	Pin 39	Señal del sensor de temperatura del refrigerante
Pin 19	Sensor MAP	Pin 7	Señal de sensor de temperatura del múltiple
Pin 20	Sensor TPS	Pin 32	Señal del sensor de posición del pedal
Pin 21	Tierra Sensores 2	Pin 74	Tierra
Pin 22	Sensor MAP	Pin 48	Voltaje de ignición 1/2
Pin 23	Ignition 12v	Pin 47	Voltaje de ignición 2/3
Pin 24	Bateria 12v	Pin 50	Voltaje de encendido
Pin 25	Bateria 12v	Pin 25	Voltaje de encendido

## Capitulo IV

### 4.1 Análisis de la ECU con Falla al Banco de Prueba

Se procedió a conectar mediante cables Dupont el banco de pruebas a la Ecu del Sail tomando en cuenta en pin out de las misma, donde en primera instancia se conectó los pines que energizaban a la Ecu para luego conectar las bobinas, inyectores, alternador, tierra de los sensores, led inmovilizador, bomba de gasolina, y ventiladores.

En primera instancia se observa cómo está el banco de pruebas cuando esta apagado, en la figura 74 se observa que el banco de pruebas se encuentra apagado.

**Figura 74**

*Banco de Pruebas Apagado*



En la figura 75 se puede verificar que al momento de poner en funcionamiento el banco de prueba se puede visualizar que el check engine se apaga como se lo puede visualizar en el círculo rojo.

**Figura 75**

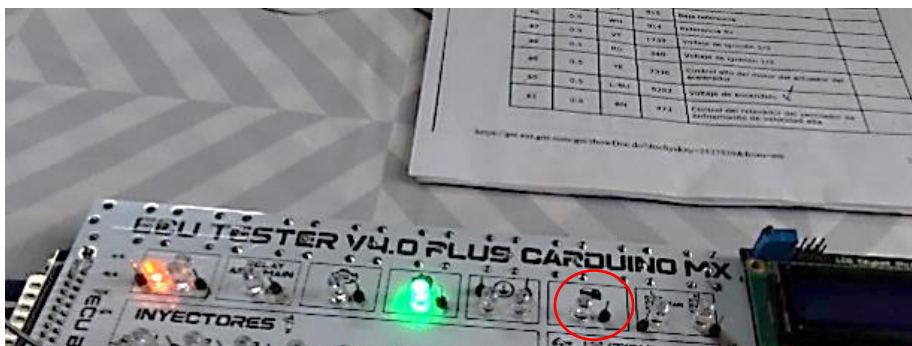
*Apagado del Led del Check Engine*



Posteriormente al darle al botón de Start se observa cómo se apaga la luz del led inmovilizador, como se visualiza en la figura 76.

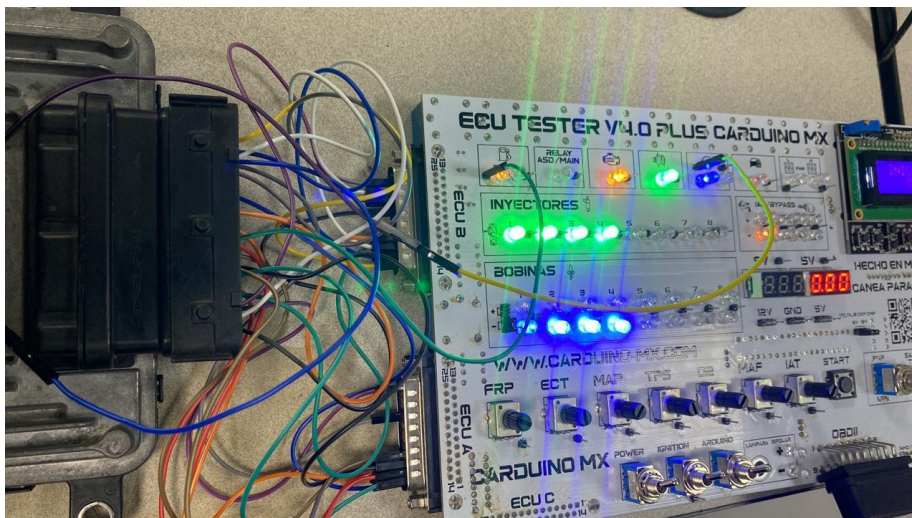
**Figura 76**

*Apagado del Led Inmovilizador.*



**Figura 77**

*Prueba #1 de ECU al Banco de Pruebas*



En esta primera prueba se logra divisar que el led de Fan low, es decir del ventilador del motor no se prendió por motivos a que la ECU no estaría mandando la señal para activarlo, esto producto a que algún componente de la ECU internamente podría estar dañado.

**Figura 78***Falla de Fan en ECU del Sail*

En segunda instancia se procedió a conectar los sensores donde se observa que no había alguna variación de voltajes al cambiar las RPM del motor, esto da a entender junto con el problema anterior que la Ecu tiene varios componentes internos dañados que producen un cortocircuito y no dejan que la Ecu envíe la señal correspondiente.

**Figura 79***Falla de Variación de Voltaje en Sensores*

Al no tener éxito en la primera prueba que se realiza, se decidió hacer la prueba con otra Ecu de Sail para descartar que fuese problema del banco de pruebas.

#### **4.2 Análisis de la ECU en Buen Estado al Banco de Pruebas**

Como se observa cuenta en esta Ecu sí enciende el led indicador del low fan lo que confirma de que la primera Ecu que se usó tenía defecto.



**Figura 80***Correcto Funcionamiento ECU #2 Sail*

Luego de comprobar la primera parte del banco de pruebas y ver que todo estaba funcional, se procede a usar la segunda parte del banco de pruebas que son los sensores, donde debe existir variaciones de voltaje cuando se modifica las RPM del motor.

También se puede visualizar que se encienden los diodos leds de los cuatros inyectores del banco de prueba que son de color verdes y los cuatros leds azules pertenecen a la bobina, a su vez se puede visualizar el funcionamiento de la bomba de combustible que se lo puede distinguir con el led de color naranja que se encuentra en la esquina superior izquierda, y el led superior de color azul corresponde a tierra de la ECU del Sail.

**Figura 81***Variación de Voltaje en Sensores Dentro del Rango Permitido*

## Conclusiones

Se concluye luego de haber realizado los diferentes diagnosticos en varias computadoras automotrices de un mismo modelo de vehículo lo siguiente:

Existen diferentes bancos de pruebas, lo que varía son: los componentes que lo forman, portabilidad y el software en el que están basados sin embargo cumplen una misma función, diagnóstico y estudio de una ECU.

El uso de un software libre en este caso Arduino permite agregar más funciones al banco de pruebas sin embargo es necesario tener conocimientos previos del mismo.

En la primera prueba se verifica un fallo interno en la ECU, dado que no presenta la señal del ventilador esto producto de un cortocircuito o componente interno en mal estado, mientras que en la segunda prueba haciendo uso de otra ECU del mismo modelo del vehículo se determina que la misma que se encuentra en buen estado.

Se realiza guías prácticas de libre acceso donde se puede dar a conocer sobre el manejo del banco de pruebas.



### **Recomendaciones**

Al momento de realizar las practicas con el banco de prueba y con la ECU que se vaya a usar, es importante seguir algunas recomendaciones como las siguientes:

- Comprobar que existe continuidad en los cables Dupont con la ayuda del multímetro.
- Evitar tocar varios puntos de soldadura con el multímetro o cualquier otro dispositivo.
- Obtener el correcto pin out de la Ecu que se vaya a utilizar.
- No colocar el banco de pruebas en algún lugar como: almohada, colchón, tela, preferible utilizar una mesa para evitar que el banco se caliente.
- Por seguridad no conectar con un solo cable dupont directamente de la ECU al banco de prueba, es preferible usar de dos a tres cables para dicha conexión.
- Conocer el número de dientes y huecos de un vehículo para poder añadir a las líneas de código de programación en Arduino.

## Bibliografía

Alonso, J. M. (2003). *Técnicas del automóvil*. Madrid.

Bello, M. (Abril de 2021). *Real Academia Española*. <https://dle.rae.es/implementar>

Cajicasa, J. V. (2016). *Diseño y construcción de un banco comprobador de ECUS de sistemas de inyección de gasolina usando una tarjeta adquisición Arduino Mega*. Azuay: Universidad del Azuay.

D'Addario, M. (2017). *Manual de mecánica del automóvil*.

Den. (20 de Abril de 2022). *Autodoc club*. Autodoc Club. Obtenido de <https://club.autodoc.es/magazin/que-es-la-ecu-y-como-funciona>

Dono, E. (8 de Enero de 2022). *Primicias Ec*.

<https://www.primicias.ec/noticias/economia/ecuador-parque-automotor-mas-autos-pichincha-guayas/>

Fernandez, F. B. (2012). *Proyectos de Arduino*.

Gil, M. O. (31 de Enero de 2021). *Unitec Blog*. <https://blogs.unitec.mx/vida-universitaria/para-que-sirve-la-programacion/>

Luque, P. (2008). *Ingeniería del Automóvil. Sistemas y Comportamientos Dinámico*. Madrid.

Marnato, E. (18 de Junio de 2022). *Todo Mecánica*.

<https://www.todomecanica.com/blog/1114-cuantos-coches-hay-en-el-mundo.html>

Mellard, T. (2017). *Automotive Electronic Systems*. Newnes.

<https://doi.org/https://doi.org/10.1016/C2013-0-04529-X>

Morán, G. (2022). *Diseño y construcción de un banco de emulación de estados de funcionamiento y comprobación de unidades de control electrónico de motor ECM usando Arduino*. UIDE.

Noroña, M. (2018). Estudio de emisiones contaminantes producidas por un motor MEP con transmisión automática y transmisión manual. *Innova Research Journal*, III(4), 120-128.

<https://doi.org/https://doi.org/10.33890/innova.v3.n4.2018.644>

Over, F. (20 de Febrero de 2022). *Automagazine*. <https://automagazine.ec/las-ventas-de-vehiculos-en-ecuador-a-enero-de-2022/#:~:text=La%20cifra%20provisional%20para%20el,lo%20registrado%20en%20diciembre%202021>.

Panadero, J. (3 de Julio de 2012). *Ecu qué es y porqué de su existencia*. Tecnovia: <https://www.diariomotor.com/tecmovia/2012/07/03/ecu-que-es-y-el-porque-de-su-existencia/>

Silva, N. I. (2020). *Guia Basica Mecanica Automotriz*. Amazon Digital Services.

Snold, F. (Julio de 2022). *Wordreference*.

<https://www.wordreference.com/definicion/comprobaci%C3%B3n>

Tarlio, G. (Enero de 2022). *Significados*. <https://www.significados.com/diagnostico/>

Villa, J. M. (2016). *Diseño y comprobacion de un banco de Ecus*. Cuenca: Universidad del Azuay.

Walack, J. (2 de Mayo de 2018). *Arduino*. <https://www.arduino.cc/en/Guide/Introduction>

## Anexos



Escuela de Ingeniería Automotriz

## Guía de Práctica #1

ASIGNATURA	RESPONSABLE	FECHA DE PRÁCTICA	DURACIÓN PRÁCTICA
	Docente		2-HORAS


LABORATORIO O TALLER	GUIA PRÁCTICA N°	NOMBRE DE LA PRÁCTICA
Taller	01	Proceso de conexión de la ECU del Sail al banco de pruebas.

<b>1.</b>	<b>OBJETIVO GENERAL</b>
	<ul style="list-style-type: none"> <li>Realizar correctamente las conexiones.</li> </ul>

<b>2.</b>	<b>OBJETIVOS ESPECÍFICOS</b>
	<ul style="list-style-type: none"> <li>Dar a conocer el proceso que se realiza para realizar las conexiones.</li> <li>Dar a conocer los implementos que se necesitan para poder llevar a cabo el proceso.</li> <li>Dar a conocer los pines del banco de prueba.</li> </ul>

<b>3.</b>	<b>RECURSOS</b>						
	<table border="1" style="width: 100%;"> <thead> <tr> <th>EQUIPOS</th> <th>MATERIALES</th> <th>INSUMOS</th> </tr> </thead> <tbody> <tr> <td>Banco de pruebas Ecu</td> <td>Cables dupont</td> <td>Pinout del vehículo</td> </tr> </tbody> </table>	EQUIPOS	MATERIALES	INSUMOS	Banco de pruebas Ecu	Cables dupont	Pinout del vehículo
EQUIPOS	MATERIALES	INSUMOS					
Banco de pruebas Ecu	Cables dupont	Pinout del vehículo					

<b>4.</b>	<b>DESARROLLO DE LA PRÁCTICA</b>																																																																								
	<p>1. Obtener el Pinout del vehículo que se desea trabajar, en este caso sería el Sail</p> <table border="1" style="width: 100%; font-size: small;"> <thead> <tr> <th>Clavija</th> <th>Tamaño</th> <th>Color</th> <th>Circuito</th> <th>Función</th> <th>Opción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0.5</td> <td>D-GN</td> <td>2123</td> <td>Señal de control de ignición - Cilindro 4</td> <td>-</td> </tr> <tr> <td>2</td> <td>0.5</td> <td>L-BU</td> <td>2122</td> <td>Señal de control de ignición - Cilindro 3</td> <td>-</td> </tr> <tr> <td>3</td> <td>0.5</td> <td>VF</td> <td>2124</td> <td>Señal de control de ignición - Cilindro 2</td> <td>-</td> </tr> <tr> <td>4</td> <td>0.5</td> <td>BR</td> <td>150</td> <td>Tierra</td> <td>-</td> </tr> <tr> <td>5</td> <td>0.35</td> <td>VE</td> <td>630</td> <td>Señal del sensor de posición del árbol de levas</td> <td>-</td> </tr> <tr> <td>6</td> <td>-</td> <td>-</td> <td>-</td> <td>No se utiliza</td> <td>-</td> </tr> <tr> <td>7</td> <td>0.35</td> <td>WH</td> <td>919</td> <td>Señal de sensor de temperatura del múltiple</td> <td>-</td> </tr> <tr> <td>8</td> <td>0.5</td> <td>BN</td> <td>435</td> <td>Señal del sensor de posición de la válvula EGR</td> <td>-</td> </tr> <tr> <td>9</td> <td>-</td> <td>-</td> <td>-</td> <td>No se utiliza</td> <td>-</td> </tr> <tr> <td>10</td> <td>0.5</td> <td>BN</td> <td>908</td> <td>Señal del sensor de oxígeno caliente (HO2S) trasero</td> <td>-</td> </tr> <tr> <td>11</td> <td>0.35</td> <td>D-GN</td> <td>432</td> <td>Señal del sensor de presión absoluta del distribuidor</td> <td>-</td> </tr> </tbody> </table>	Clavija	Tamaño	Color	Circuito	Función	Opción	1	0.5	D-GN	2123	Señal de control de ignición - Cilindro 4	-	2	0.5	L-BU	2122	Señal de control de ignición - Cilindro 3	-	3	0.5	VF	2124	Señal de control de ignición - Cilindro 2	-	4	0.5	BR	150	Tierra	-	5	0.35	VE	630	Señal del sensor de posición del árbol de levas	-	6	-	-	-	No se utiliza	-	7	0.35	WH	919	Señal de sensor de temperatura del múltiple	-	8	0.5	BN	435	Señal del sensor de posición de la válvula EGR	-	9	-	-	-	No se utiliza	-	10	0.5	BN	908	Señal del sensor de oxígeno caliente (HO2S) trasero	-	11	0.35	D-GN	432	Señal del sensor de presión absoluta del distribuidor	-
Clavija	Tamaño	Color	Circuito	Función	Opción																																																																				
1	0.5	D-GN	2123	Señal de control de ignición - Cilindro 4	-																																																																				
2	0.5	L-BU	2122	Señal de control de ignición - Cilindro 3	-																																																																				
3	0.5	VF	2124	Señal de control de ignición - Cilindro 2	-																																																																				
4	0.5	BR	150	Tierra	-																																																																				
5	0.35	VE	630	Señal del sensor de posición del árbol de levas	-																																																																				
6	-	-	-	No se utiliza	-																																																																				
7	0.35	WH	919	Señal de sensor de temperatura del múltiple	-																																																																				
8	0.5	BN	435	Señal del sensor de posición de la válvula EGR	-																																																																				
9	-	-	-	No se utiliza	-																																																																				
10	0.5	BN	908	Señal del sensor de oxígeno caliente (HO2S) trasero	-																																																																				
11	0.35	D-GN	432	Señal del sensor de presión absoluta del distribuidor	-																																																																				

<p>2. Luego de haber obtenido el Pinout del vehículo que se desea trabajar, se debe conocer el Pinout del banco de pruebas, mismo que se encuentra desde la tabla dos hasta la tabla siete en el presente documento.</p>	<p>HR.</p> <table border="1"> <thead> <tr> <th>Número de Pin</th> <th>Descripción</th> </tr> </thead> <tbody> <tr><td>Pin 1</td><td>inyector 1</td></tr> <tr><td>Pin 2</td><td>inyector 2</td></tr> <tr><td>Pin 3</td><td>inyector 3</td></tr> <tr><td>Pin 4</td><td>inyector 4</td></tr> <tr><td>Pin 5</td><td>inyector 5</td></tr> <tr><td>Pin 6</td><td>inyector 6</td></tr> <tr><td>Pin 7</td><td>inyector 7</td></tr> <tr><td>Pin 8</td><td>inyector 8</td></tr> <tr><td>Pin 9</td><td>Check Engine</td></tr> <tr><td>Pin 10</td><td>Bomba Cos Salida (+)</td></tr> <tr><td>Pin 11</td><td>CMP2</td></tr> <tr><td>Pin 12</td><td>Sensor MAF</td></tr> <tr><td>Pin 13</td><td>Sensor EBP</td></tr> <tr><td>Pin 14</td><td>Hobana 1</td></tr> <tr><td>Pin 15</td><td>Hobana 2</td></tr> <tr><td>Pin 16</td><td>Hobana 3</td></tr> <tr><td>Pin 17</td><td>Hobana 4</td></tr> <tr><td>Pin 18</td><td>Hobana 5</td></tr> <tr><td>Pin 19</td><td>Hobana 6</td></tr> <tr><td>Pin 20</td><td>Hobana 7</td></tr> </tbody> </table>	Número de Pin	Descripción	Pin 1	inyector 1	Pin 2	inyector 2	Pin 3	inyector 3	Pin 4	inyector 4	Pin 5	inyector 5	Pin 6	inyector 6	Pin 7	inyector 7	Pin 8	inyector 8	Pin 9	Check Engine	Pin 10	Bomba Cos Salida (+)	Pin 11	CMP2	Pin 12	Sensor MAF	Pin 13	Sensor EBP	Pin 14	Hobana 1	Pin 15	Hobana 2	Pin 16	Hobana 3	Pin 17	Hobana 4	Pin 18	Hobana 5	Pin 19	Hobana 6	Pin 20	Hobana 7
Número de Pin	Descripción																																										
Pin 1	inyector 1																																										
Pin 2	inyector 2																																										
Pin 3	inyector 3																																										
Pin 4	inyector 4																																										
Pin 5	inyector 5																																										
Pin 6	inyector 6																																										
Pin 7	inyector 7																																										
Pin 8	inyector 8																																										
Pin 9	Check Engine																																										
Pin 10	Bomba Cos Salida (+)																																										
Pin 11	CMP2																																										
Pin 12	Sensor MAF																																										
Pin 13	Sensor EBP																																										
Pin 14	Hobana 1																																										
Pin 15	Hobana 2																																										
Pin 16	Hobana 3																																										
Pin 17	Hobana 4																																										
Pin 18	Hobana 5																																										
Pin 19	Hobana 6																																										
Pin 20	Hobana 7																																										
<p>3. Se realiza la conexión pin por pin mediante cables dupont teniendo en cuenta el Pinout de la ECU y el Pinout del banco de pruebas.</p>																																											

## Guía de Práctica #2

ASIGNATURA	RESPONSABLE	FECHA DE PRÁCTICA	DURACIÓN PRÁCTICA
	Docente		2-HORAS

LABORATORIO O TALLER	GUIA PRÁCTICA N°	NOMBRE DE LA PRÁCTICA
Taller	01	Diagnostico de ECU del Sail






<b>1.</b>	<b>OBJETIVO GENERAL</b>
	<ul style="list-style-type: none"> <li>Realizar diagnostico mediante el uso del banco de pruebas</li> </ul>

<b>2.</b>	<b>OBJETIVOS ESPECÍFICOS</b>
	<ul style="list-style-type: none"> <li>Verificar si la ECU se encuentra en buen estado.</li> <li>Dar a conocer la veracidad del banco de pruebas.</li> <li>Demostrar que es un banco de pruebas de fácil uso.</li> </ul>

<b>3.</b>	<b>RECURSOS</b>						
	<table border="1" style="width: 100%;"> <thead> <tr> <th>EQUIPOS</th> <th>MATERIALES</th> <th>INSUMOS</th> </tr> </thead> <tbody> <tr> <td>ECU Sail Banco de pruebas</td> <td>Cables dupont</td> <td>Pinout del vehículo Pinout del banco de pruebas</td> </tr> </tbody> </table>	EQUIPOS	MATERIALES	INSUMOS	ECU Sail Banco de pruebas	Cables dupont	Pinout del vehículo Pinout del banco de pruebas
EQUIPOS	MATERIALES	INSUMOS					
ECU Sail Banco de pruebas	Cables dupont	Pinout del vehículo Pinout del banco de pruebas					

<b>4.</b>	<b>DESARROLLO DE LA PRÁCTICA</b>
	<p>1. Luego de haber realizado las conexiones como se muestra en la guía practica uno, procedemos a encender el banco de pruebas, y seleccionar la marca que deseamos trabajar, en este caso sería Chevrolet sail.</p>



<p>2. Una vez encendido le pulsamos al botón de start, posteriormente se observa cómo se apaga la luz del check engine</p>	
<p>3. Luego de apagarse el testigo/ led de check engine también se apaga el led del inmovilizador</p>	
<p>4. Sin embargo, se visualiza que el led del "fan" se encuentra apagado, esto producto de algún corto en alguna línea o componente interno en la ECU</p>	
<p>5. En este caso se procede a realizar la prueba con una segunda ECU del vehículo Sail, para comprobar si de verdad es problema de la ECU o del banco de pruebas, dando como resultado que la ECU que se usó en la primera prueba esta defectuosa puesto que en esta segunda prueba sí funciona.</p>	
<p>6. En ultima instancia se realiza la prueba de sensores, siendo este la válvula IAC donde se observa una variación de voltaje.</p>	

## Programación de Arduino

### LCD

```

#include <LiquidCrystal.h>
LiquidCrystal lcd(8,9,4,5,6,7);

#include <Stepper.h>
#define STEPS 300
Stepper stepper(STEPS, A1, A2, A3, A4);
int menu_valvulas;
long rpm;          //variable para rpm
int x;
int pulsado=3000;  // variable para guardar que tecla se pulso
int dientes;
int huecos;
long mc=500;       //variable para microsegundos
byte menu_marcas;
byte menu_principal;
byte switch_menu_diesel;
byte switch_menu_chevrolet;
byte switch_menu_nissan;
byte switch_menu_toyota;
byte switch_menu_volkswagen;
byte switch_menu_ford;
byte switch_menu_chrysler;
byte switch_menu_mazda;
byte switch_menu_hyundai;
byte switch_menu_bmw;
byte switch_menu_ngc;
byte switch_menu_citroen;
byte switch_menu_seat;
byte switch_menu_renault;
byte switch_menu_peugeot;
byte switch_menu_audi;
byte menu_pcm_switch;
byte activacion;
int menu;
int total_mc=240;
#define sensorckp 1
#define valvulaiac 2
#define valvulabypass 3
#define ckp      A1
#define cmp_uno  A2
#define cmp_dos  A3
#define cmp_tres A4
#define chevrolet 1

```



```

#define toyota      2
#define volkswagen 3
#define nissan      4
#define chrysler   5
#define ford        6
#define diesel      7
#define mazda       8
#define bmw         9
#define citroen     10
#define seat        11
#define renault     12
#define peugeot     13
#define audi        14
//#define honda
//#define kia
//#define mitsubishi
//#define fiat
//#define suzuki

#define derecha     1
#define izquierda  2
#define arriba      3
#define abajo       4
#define enter       5
void pulso_cmp_subida(){
    digitalWrite(cmp_uno,HIGH);
}
void pulso_cmp_bajada(){
    digitalWrite(cmp_uno,LOW);
}
void setup(){

    convertir_mc_rp();
    pinMode(ckp,OUTPUT);
    pinMode(cmp_uno,OUTPUT);
    pinMode(cmp_dos,OUTPUT);
    pinMode(3,OUTPUT);
    stepper.setSpeed(40);
    menu_pcm_switch=1;
    logo_principal();
    activacion=0;
    menu_marcas    = 1;
    menu=-1;
    menu_principal=1;
    //se inician en la primera opcion
    switch_menu_diesel    = 1;
    switch_menu_chevrolet = 1;

```

```

switch_menu_toyota      = 1;
switch_menu_nissan     = 1;
switch_menu_volkswagen = 1;
switch_menu_ford       = 1;
switch_menu_chrysler   = 1;
switch_menu_mazda      = 1;
switch_menu_hyundai    = 1;
switch_menu_ngc        = 1;
switch_menu_bmw        = 1;
switch_menu_citroen    = 1;
switch_menu_seat       = 1;
switch_menu_renault    = 1;
switch_menu_peugeot    = 1;
switch_menu_audi       = 1;
}
void loop(){
switch(menu_principal){

    case 1:                //caso chevrolet
        sensorckp_principal();          // ejecuta la funcion
        chevrolet_principal(); ubicada en la pestaña PANTALLA
        lectura_botones();              // lectura de botones
        if(pulsado==derecha){delay(200); // pulsacion abajo entra a la marca
        toyota
        menu_principal=valvulaiac;}
        if(pulsado==enter){delay(200); //pulsado derecha hace que
        menu_chevrolet=1 sea 1
        menu=0;}
        while(menu==0){              // se mantiene en esa opcion ejecutando
            sensorckp_menu();          // la funcion chevrolet_menu(); de la
            pestaña chevrolet

        }
        break;

    case 2:                //caso chevrolet
        valvulaiac_principal();          // ejecuta la funcion
        chevrolet_principal(); ubicada en la pestaña PANTALLA
        lectura_botones();
        if(pulsado==izquierda){delay(200); // pulsacion abajo entra a la marca
        toyota
        menu_principal=sensorckp;}// lectura de botones
        if(pulsado==enter){delay(200); //pulsado derecha hace que
        menu_chevrolet=1 sea 1
        menu=0;}
        while(menu==0){              // se mantiene en esa opcion ejecutando

```

```

        valvulaiac_menu(); // la funcion chevrolet_menu(); de la
pestaña chevrolet
    }
    break;

```

```

lectura_botones();
}

```

## Menu

```

#define ngcv4 1
#define ngcv6 2
#define ngcv8 3

void ngc_menu(){
    switch(switch_menu_ngc){

        case ngcv4:
            pantalla_ngcv4();
            lectura_botones();
            if(pulsado==derecha){delay(200);
            switch_menu_ngc=2;}
            if(pulsado==enter){delay(500);
            activacion=1;
            lectura_botones();
            if(pulsado==enter){
            while(pulsado==enter){
                lectura_botones();
            }
            menu=0;
            activacion=0;
            switch_menu_ngc=1;
            }
            while(activacion==1){
                ngc4cil_senal();
            }
            }
            break;
        case ngcv6:
            pantalla_ngcv6();
            lectura_botones();
            if(pulsado==derecha){delay(200);
            switch_menu_ngc=3;}
            if(pulsado==izquierda){delay(200);
            switch_menu_ngc=1;}
            if(pulsado==enter){delay(500);
            activacion=1;
            lectura_botones();

```

```

    if(pulsado==enter){
    while(pulsado==enter){
        lectura_botones();
    }
    menu=0;
    activacion=0;
    switch_menu_ngc=1;
    }
    while(activacion==1){
        ngc6cil_senal();
    }
    }
    break;
    case ngcv8:
    pantalla_ngcv8();
    lectura_botones();
    if(pulsado==izquierda){delay(200);
    switch_menu_ngc=2;}
    if(pulsado==enter){delay(500);
    activacion=1;
    lectura_botones();
    if(pulsado==enter){
    while(pulsado==enter){
        lectura_botones();
    }
    menu=0;
    activacion=0;
    switch_menu_ngc=1;
    }
    while(activacion==1){
        ngc8cil_senal();
    }
    }
    break;
}
}
void ngc4cil_senal(){
    lectura_botones();
    if(pulsado==enter){delay(500);
        activacion=2;
        lectura_botones();
    if(pulsado==enter){
    while(pulsado==enter){
        lectura_botones();
    }
    }
    activacion=0;
}

```

```

    }
    lcd.clear();
    pantalla_rpm();
    pantalla_activa();
while(activacion==2){
    lectura_botones();
    ajustar_rpm();
    // ACCION DE CKP
    ngc4cil();
if(pulsado==enter){delay(200);
    activacion=1;

}
}
}
else{
    pantalla_desactiva();

}
}
void ngc6cil_senal(){
    lectura_botones();
if(pulsado==enter){delay(500);
    activacion=2;
    lectura_botones();
if(pulsado==enter){
while(pulsado==enter){
    lectura_botones();
}
}
activacion=0;

}
    lcd.clear();
    pantalla_rpm();
    pantalla_activa();
while(activacion==2){
    lectura_botones();
    ajustar_rpm();
    // ACCION DE CKP
    ngc6cil();
if(pulsado==enter){delay(200);
    activacion=1;

}
}
}
else{

```

```

        pantalla_desactiva();

    }
}
void ngc8cil_senal(){
    lectura_botones();
    if(pulsado==enter){delay(500);
        activacion=2;
        lectura_botones();
        if(pulsado==enter){
            while(pulsado==enter){
                lectura_botones();
            }
        }
        activacion=0;

        }
        lcd.clear();
        pantalla_rpm();
        pantalla_activa();
    while(activacion==2){
        lectura_botones();
        ajustar_rpm();
        // ACCION DE CKP
        ngc8cil();
        if(pulsado==enter){delay(200);
            activacion=1;

        }
    }
}
else{
    pantalla_desactiva();

}
}
}

```

### Chrysler

```

#define sbec_60    1
#define sbec_80    2
#define jtec       3
#define ngc        4

void chrysler_menu(){
    switch(switch_menu_chrysler){

```

```
////////////////////////////////////
```

```

case sbec_60:
pantalla_pcm();
lectura_botones();
if(pulsado==derecha){delay(200);
switch_menu_chrysler=2;}
if(pulsado==enter){delay(500);
activacion=1;
lectura_botones();
if(pulsado==enter){
while(pulsado==enter){
    lectura_botones();
}
}
menu=0;
activacion=0;
switch_menu_chrysler=1;
}
while(activacion==1){
    chrysler_sbec60();
}
}
break;

```

```
////////////////////////////////////
```

```

case sbec_80:
pantalla_sbec_negro();
lectura_botones();
if(pulsado==derecha){delay(200);
switch_menu_chrysler=3;}
if(pulsado==izquierda){delay(200);
switch_menu_chrysler=1;}
if(pulsado==enter){delay(500);
activacion=1;
lectura_botones();
if(pulsado==enter){
while(pulsado==enter){
    lectura_botones();
}
}
menu=0;
activacion=0;
switch_menu_chrysler=1;
}
while(activacion==1){
    chrysler_sbec60();
}
}
break;

```

```
////////////////////////////////////
```

```

case jtec:
pantalla_jtec();
lectura_botones();
if(pulsado==derecha){delay(200);
switch_menu_chrysler=4;}
if(pulsado==izquierda){delay(200);
switch_menu_chrysler=2;}
if(pulsado==enter){delay(500);
activacion=1;
lectura_botones();
if(pulsado==enter){
while(pulsado==enter){
    lectura_botones();
}
}
menu=0;
activacion=0;
switch_menu_chrysler=1;
}
while(activacion==1){
    chrysler_sbec60();
}
}
break;
////////////////////////////////////
case ngc:
pantalla_ngc();
lectura_botones();
if(pulsado==izquierda){delay(200);
switch_menu_chrysler=3;}
if(pulsado==enter){delay(500);
activacion=1;
lectura_botones();
if(pulsado==enter){
while(pulsado==enter){
    lectura_botones();
}
}
menu=0;
activacion=0;
switch_menu_chrysler=1;
}
while(activacion==1){
    ngc_menu();
}
}
break;
////////////////////////////////////

```



## Chevrolet

```

#define chevy      1
#define luvdmax   2
#define aveo      3
#define captiva   4
#define silverado 5
#define spark     6

void chevrolet_menu(){
    switch(switch_menu_chevrolet){

        case 1:
            pantalla_chevy();
            lectura_botones();
            if(pulsado==derecha){delay(200);
            switch_menu_chevrolet=2;}
            if(pulsado==enter){delay(500);
            activacion=1;
            lectura_botones();
            if(pulsado==enter){
                while(pulsado==enter){
                    lectura_botones();
                }
            }
            menu=0;
            activacion=0;
            switch_menu_chevrolet=1;
        }
        while(activacion==1){
            senal_chevy();
        }
    }
    break;

    case 2:
        pantalla_luvdmax();
        lectura_botones();
        if(pulsado==izquierda){delay(200);
        switch_menu_chevrolet=1;}
        if(pulsado==derecha){delay(200);
        switch_menu_chevrolet=3;}
        if(pulsado==enter){delay(500);
        activacion=1;
        lectura_botones();
        if(pulsado==enter){
            while(pulsado==enter){
                lectura_botones();
            }
        }
    }
}

```

```

menu=0;
activacion=0;
switch_menu_chevrolet=1;
}
while(activacion==1){
    luvdmax_senal();
}
}
break;
case 3:
pantalla_aveo();
lectura_botones();
if(pulsado==izquierda){delay(200);
switch_menu_chevrolet=2;}
if(pulsado==derecha){delay(200);
switch_menu_chevrolet=4;}
if(pulsado==enter){delay(500);
activacion=1;
lectura_botones();
if(pulsado==enter){
while(pulsado==enter){
    lectura_botones();
}
}
menu=0;
activacion=0;
switch_menu_chevrolet=1;
}
while(activacion==1){
    aveo_senal();
}
}
break;

case 4:
pantalla_captiva();
lectura_botones();
if(pulsado==izquierda){delay(200);
switch_menu_chevrolet=3;}
if(pulsado==derecha){delay(200);
switch_menu_chevrolet=5;}
if(pulsado==enter){delay(500);
activacion=1;
lectura_botones();
if(pulsado==enter){
while(pulsado==enter){
    lectura_botones();
}
}
}

```

```
menu=0;
activacion=0;
switch_menu_chevrolet=1;
}
while(activacion==1){
    captiva_senal();
}
}
break;

case 5:
pantalla_silverado();
lectura_botones();
if(pulsado==izquierda){delay(200);
switch_menu_chevrolet=4;}
if(pulsado==derecha){delay(200);
switch_menu_chevrolet=6;}
if(pulsado==enter){delay(500);
activacion=1;
lectura_botones();
if(pulsado==enter){
while(pulsado==enter){
    lectura_botones();
}
}
menu=0;
activacion=0;
switch_menu_chevrolet=1;
}
while(activacion==1){
    silverado_senal();
}
}
break;

case 6:
pantalla_spark();
lectura_botones();
if(pulsado==izquierda){delay(200);
switch_menu_chevrolet=5;}
if(pulsado==enter){delay(500);
activacion=1;
lectura_botones();
if(pulsado==enter){
while(pulsado==enter){
    lectura_botones();
}
}
menu=0;
```

```

    activacion=0;
    switch_menu_chevrolet=1;
}
while(activacion==1){
    spark_senal();
}
}
break;
}
}
void senal_chevy(){
    lectura_botones();
    if(pulsado==enter){delay(500);
        activacion=2;
        lectura_botones();
        if(pulsado==enter){
            while(pulsado==enter){
                lectura_botones();
            }
        }
        activacion=0;
    }
    lcd.clear();
    pantalla_rpm();
    pantalla_activa();
    // PROGRAMAR DIENTES Y HUECOS
    dientes=60;
    huecos=2;
    dientes=dientes-huecos;
    while(activacion==2){
        lectura_botones();
        ajustar_rpm();
        // ACCION DE CKP
        ckp_programable();
        if(pulsado==enter){delay(200);
            activacion=1;
            pantalla_rpm();
        }
    }
}
else{
    pantalla_desactiva();
}
}
void luvdmax_senal(){
    lectura_botones();
    if(pulsado==enter){delay(500);
        activacion=2;

```

```

        lectura_botones();
    if(pulsado==enter){
    while(pulsado==enter){
        lectura_botones();
    }
    activacion=0;

}

    lcd.clear();
    pantalla_rpm();
    pantalla_activa();
while(activacion==2){
    lectura_botones();
    ajustar_rpm();
    // ACCION DE CKP
    LUVDMAX();
if(pulsado==enter){delay(200);
    activacion=1;

}
}
}
else{
    pantalla_desactiva();

}
}
}
void aveo_senal(){
    lectura_botones();
    if(pulsado==enter){delay(500);
        activacion=2;
        lectura_botones();
        if(pulsado==enter){
        while(pulsado==enter){
            lectura_botones();
        }
        activacion=0;

    }

        lcd.clear();
        pantalla_rpm();
        pantalla_activa();
while(activacion==2){
    lectura_botones();
    ajustar_rpm();
    // ACCION DE CKP
    AVEO();

```

```

    if(pulsado==enter){delay(200);
        activacion=1;

    }
    }
    }
    else{
        pantalla_desactiva();

    }
    }
}
void captiva_senal(){
    lectura_botones();
    if(pulsado==enter){delay(500);
        activacion=2;
        lectura_botones();
        if(pulsado==enter){
            while(pulsado==enter){
                lectura_botones();
            }
            activacion=0;

        }

        lcd.clear();
        pantalla_rpm();
        pantalla_activa();
        while(activacion==2){
            lectura_botones();
            ajustar_rpm();
            // ACCION DE CKP
            CAPTIVA();
            if(pulsado==enter){delay(200);
                activacion=1;

            }

        }
    }
    }
}
void silverado_senal(){
    lectura_botones();
    if(pulsado==enter){delay(500);
        activacion=2;
        lectura_botones();

```

```

    if(pulsado==enter){
    while(pulsado==enter){
        lectura_botones();
    }
    activacion=0;

}
    lcd.clear();
    pantalla_rpm();
    pantalla_activa();
while(activacion==2){
    lectura_botones();
    ajustar_rpm();
    // ACCION DE CKP
    SILVERADO();
if(pulsado==enter){delay(200);
    activacion=1;

}
}
}
else{
    pantalla_desactiva();

}
}
}
void spark_senal(){
    lectura_botones();
if(pulsado==enter){delay(500);
    activacion=2;
    lectura_botones();
if(pulsado==enter){
while(pulsado==enter){
    lectura_botones();
}
}
    activacion=0;

}
    lcd.clear();
    pantalla_rpm();
    pantalla_activa();
while(activacion==2){
    lectura_botones();
    ajustar_rpm();
    // ACCION DE CKP
    SPARK();
if(pulsado==enter){delay(200);

```

```

        activacion=1;

    }
    }
    }
    else{
        pantalla_desactiva();

    }
    }

```

## CKP

```

/*COPIAR EN CASO DE NESECITAR CKP SOLAMENTE
void senal_(){
    lectura_botones();
    if(pulsado==enter){delay(200);
        activacion=2;
        lcd.clear();
        pantalla_rpm();
        pantalla_activa();
        // PROGRAMAR DIENTES Y HUECOS
        dientes=8;
        huecos=4;
        dientes_restados=dientes-huecos;
    while(activacion==2){
        lectura_botones();
        // MODIFICAR RPM + -
        ajustar_rpm();
        // ACCION DE CKP
        ckp_programable();
        if(pulsado==enter){delay(200);
            activacion=1;
        }
    }
    }
    }
    else{
        pantalla_desactiva();
        if(pulsado==izquierda){delay(200);
            activacion=0;}
        }
    }
}
*/
void ckp_programable(){

for(int diente=1;diente<=dientes;diente++){ajustar_rpm();
pulso_ckp_subida();ajustar_rpm();
pulso_ckp_subida_mitad();ajustar_rpm();

```



```

    pulso_ckp_bajada();ajustar_rpm();
    pulso_ckp_bajada_mitad();ajustar_rpm();
}
    digitalWrite(ckp,0);
    delayMicroseconds(mc*2);
for(int hueco=1;hueco<=huecos;hueco++){
    pulso_ckp_low();
}
}
void pulso_ckp_subida(){
    digitalWrite(ckp,1);
    delayMicroseconds(mc/2);
}
void pulso_ckp_subida_mitad(){
    delayMicroseconds(mc/2);
}
void pulso_ckp_bajada(){
    digitalWrite(ckp,0);
    delayMicroseconds(mc/2);
}
void pulso_ckp_bajada_mitad(){
    delayMicroseconds(mc/2);
}
void pulso_ckp_low(){
    digitalWrite(ckp,0);
    delayMicroseconds(mc);
}
void pulso_ckp_high(){
    digitalWrite(ckp,1);
    delayMicroseconds(mc);
}
}

```

### Convertir RPMS a microsegundos.

```

void convertir_mc_rp(){
    rpm=mc*total_mc;
    rpm=60000000/rpm;
}

```

### Lectura RPMS

```

void ajustar_rpm(){lectura_botones();
    if(pulsado==abajo){delay(100);
    mc=mc+50;
    convertir_mc_rp();
    ajusterpm();
    if(rpm<=999||rpm<=99){
        lcd.setCursor(0,1);
        lcd.print(F("          "));
    }
}

```

```

    pantalla_rpm();
}
if(pulsado==arriba){delay(100);
mc=mc-50;
ajusterpm();
convertir_mc_rp();
if(rpm<=999&rpm>=900||rpm<=99&rpm>=90){
    lcd.setCursor(0,1);
    lcd.print(F("          "));
}
pantalla_rpm();
}
}
void ajusterpm(){
    if(mc<50){
        mc=50;
    }

    if(mc>2500){
        mc=2500;
    }
}
Pantalla
void p_iac(){
    lcd.setCursor(0,0);
    lcd.print(F(" VALVULA IAC "));
    lcd.setCursor(0,1);
    lcd.print(F("          "));
}
void abriendo(){
    lcd.setCursor(0,1);
    lcd.print(F(" ABRIENDO "));
}
void cerrando(){
    lcd.setCursor(0,1);
    lcd.print(F(" CERRANDO "));
}

void logo_principal(){
    lcd.begin(16,2);
    lcd.setCursor(0,0);
    lcd.print(F("UNIVERSIDAD"));
    lcd.setCursor(2,1);
    lcd.print(F("INTERNACIONAL"));
    delay(1000);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(F(" DEL "));
}

```

```

    lcd.setCursor(2,1);
    lcd.print(F(" ECUADOR"));
    delay(1000);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(F("  INGENERIA  "));
    lcd.setCursor(2,1);
    lcd.print(F(" AUTOMOTRIZ"));
    delay(3000);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(F("  ANTHONY  "));
    lcd.setCursor(0,1);
    lcd.print(F("  MENDOZA  "));
    delay(3000);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(F("  MIGUEL  "));
    lcd.setCursor(0,1);
    lcd.print(F("  AREVALO  "));
    delay(3000);
    lcd.clear();
}
// menu principal
void sensorckp_principal(){
    lcd.setCursor(0,0);
    lcd.print(F(" INICIAR PRUEBA "));
    lcd.setCursor(0,1);
    lcd.print(F("  CKP CMP  >"));
}
void valvulaiac_principal(){
    lcd.setCursor(0,0);
    lcd.print(F(" INICIAR PRUEBA "));
    lcd.setCursor(0,1);
    lcd.print(F("< VALVULA  IAC  "));
}
// menu marcas
void chevrolet_principal(){
    lcd.setCursor(0,0);
    lcd.print(F("SELECCIONA MARCA"));
    lcd.setCursor(0,1);
    lcd.print(F("  CHEVROLET  ->"));
}
void toyota_principal(){
    lcd.setCursor(0,0);
    lcd.print(F("SELECCIONA MARCA"));
    lcd.setCursor(0,1);

```

```
    lcd.print(F("<- TOYOTA ->"));
}
void volkswagen_principal(){
    lcd.setCursor(0,0);
    lcd.print(F("SELECCIONA MARCA"));
    lcd.setCursor(0,1);
    lcd.print(F("<- VOLKSWAGEN ->"));
}
void nissan_principal(){
    lcd.setCursor(0,0);
    lcd.print(F("SELECCIONA MARCA"));
    lcd.setCursor(0,1);
    lcd.print(F("<- NISSAN ->"));
}
void chrysler_principal(){
    lcd.setCursor(0,0);
    lcd.print(F("SELECCIONA MARCA"));
    lcd.setCursor(0,1);
    lcd.print(F("<- CHRYSLER ->"));
}
void ford_principal(){
    lcd.setCursor(0,0);
    lcd.print(F("SELECCIONA MARCA"));
    lcd.setCursor(0,1);
    lcd.print(F("<- FORD ->"));
}
void diesel_principal(){
    lcd.setCursor(0,0);
    lcd.print(F("SELECCIONA MARCA"));
    lcd.setCursor(0,1);
    lcd.print(F("<- DIESEL ->"));
}
void mazda_principal(){
    lcd.setCursor(0,0);
    lcd.print(F("SELECCIONA MARCA"));
    lcd.setCursor(0,1);
    lcd.print(F("<- MAZDA ->"));
}
void hyundai_principal(){
    lcd.setCursor(0,0);
    lcd.print(F("SELECCIONA MARCA"));
    lcd.setCursor(0,1);
    lcd.print(F("<- HYUNDAI ->"));
}
void honda_principal(){
    lcd.setCursor(0,0);
    lcd.print(F("SELECCIONA MARCA"));
```

```

lcd.setCursor(0,1);
lcd.print(F("<-  HONDA  ->"));
}
void kia_principal(){
lcd.setCursor(0,0);
lcd.print(F("SELECCIONA MARCA"));
lcd.setCursor(0,1);
lcd.print(F("<-  KIA  ->"));
}
void mitsubishi_principal(){
lcd.setCursor(0,0);
lcd.print(F("SELECCIONA MARCA"));
lcd.setCursor(0,1);
lcd.print(F("<- MITSUBUSHI ->"));
}
void fiat_principal(){
lcd.setCursor(0,0);
lcd.print(F("SELECCIONA MARCA"));
lcd.setCursor(0,1);
lcd.print(F("<-  FIAT  ->"));
}
void renault_principal(){
lcd.setCursor(0,0);
lcd.print(F("SELECCIONA MARCA"));
lcd.setCursor(0,1);
lcd.print(F("<-  RENAULT  ->"));
}
void suzuki_principal(){
lcd.setCursor(0,0);
lcd.print(F("SELECCIONA MARCA"));
lcd.setCursor(0,1);
lcd.print(F("<-  SUSUKI  ->"));
}
void bmw_principal(){
lcd.setCursor(0,0);
lcd.print(F("SELECCIONA MARCA"));
lcd.setCursor(0,1);
lcd.print(F("<-  BMW  ->"));
}
void citroen_principal(){
lcd.setCursor(0,0);
lcd.print(F("SELECCIONA MARCA"));
lcd.setCursor(0,1);
lcd.print(F("<-  CITROEN  ->"));
}
void seat_principal(){
lcd.setCursor(0,0);

```

```

lcd.print(F("SELECCIONA MARCA"));
lcd.setCursor(0,1);
lcd.print(F("<- SEAT ->"));
}
void renaul_principal(){
lcd.setCursor(0,0);
lcd.print(F("SELECCIONA MARCA"));
lcd.setCursor(0,1);
lcd.print(F("<- RENAULT ->"));
}
void peugeot_principal(){
lcd.setCursor(0,0);
lcd.print(F("SELECCIONA MARCA"));
lcd.setCursor(0,1);
lcd.print(F("<- PEUGEOT ->"));
}
void audi_principal(){
lcd.setCursor(0,0);
lcd.print(F("SELECCIONA MARCA"));
lcd.setCursor(0,1);
lcd.print(F("<- AUDI      "));
}
////////////////////////////////////chevrolet
void pantalla_chevy(){
lcd.setCursor(0,0);
lcd.print(F("''''''MODELO''''''"));
lcd.setCursor(0,1);
lcd.print(F("      CHEVY ->"));
}
void pantalla_luvdmax(){
lcd.setCursor(0,0);
lcd.print(F("''''''MODELO''''''"));
lcd.setCursor(0,1);
lcd.print(F("<- LUVDMAX ->"));
}
void pantalla_aveo(){
lcd.setCursor(0,0);
lcd.print(F("''''''MODELO''''''"));
lcd.setCursor(0,1);
lcd.print(F("<- SAIL ->"));
}
void pantalla_captiva(){
lcd.setCursor(0,0);
lcd.print(F("''''''MODELO''''''"));
lcd.setCursor(0,1);
lcd.print(F("<- CAPTIVA ->"));
}

```

```

void pantalla_silverado(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<- SILVERADO ->"));
}
void pantalla_spark(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<- SPARK      "));
}
//activacion de señal
void pantalla_activa(){
    lcd.setCursor(0,0);
    lcd.print(F(" SIMULANDO...  "));
}
void pantalla_rpm(){
    lcd.setCursor(0,1);
    lcd.print(F(" RPM ="));
    lcd.setCursor(9,1);
    lcd.print(rpm);
}
void pantalla_desactiva(){
    lcd.setCursor(0,0);
    lcd.print(F(" PARA SIMULAR  "));
    lcd.setCursor(0,1);
    lcd.print(F(" PULSE ENTER  "));
}
////////////////////////////////////// volkswagen
void pantalla_jetta(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F(" JETTA ->"));
}
void pantalla_amarok(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<- AMAROK ->"));
}
void pantalla_beeatle(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<- BEETLE ->"));
}

```

```

}
void pantalla_golf(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<-    GOLF    ->"));
}
void pantalla_passat(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<-    PASSAT    "));
}
//////////////////////////////////////ford
void pantalla_triton(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("    TRITON    ->"));
}
void pantalla_generico(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<- EEC-V 104 ->"));
}
void pantalla_escort(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<-    ESCORT    ->"));
}
void pantalla_focus(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<-    FOCUS    ->"));
}
void pantalla_fusion(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<-    FUSION    ->"));
}
void pantalla_scape(){
    lcd.setCursor(0,0);

```



```

    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<-   SCAPE   "));
}
//pantalla chrysler
void pantalla_pcm(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("   SBEC-60   ->"));
}
void pantalla_sbec_negro(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<-   SBEC-80   ->"));
}
void pantalla_jtec(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<-   JTEC   ->"));
}
void pantalla_ngc(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<-   NGC   "));
}
////////////////////////////////////MOTORES CHRYSLER
void pantalla_case_2_0(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MOTOR''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("   2.0L   ->"));
}
void pantalla_case_2_4(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MOTOR''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<-   2.4L   ->"));
}
void pantalla_case_2_5(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MOTOR''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<-   2.5L   ->"));
}

```

```

}
void pantalla_case_3_0(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MOTOR''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<- 3.0L ->"));
}
void pantalla_case_3_2(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MOTOR''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<- 3.2L ->"));
}
void pantalla_case_3_3(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MOTOR''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<- 3.3L ->"));
}
void pantalla_case_3_5(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MOTOR''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<- 3.5L ->"));
}
void pantalla_case_3_8(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MOTOR''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<- 3.8L ->"));
}
void pantalla_case_3_9(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MOTOR''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<- 3.9L ->"));
}
void pantalla_case_4_0(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MOTOR''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<- 4.0L ->"));
}
void pantalla_case_4_7(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MOTOR''''''"));
    lcd.setCursor(0,1);

```

```

    lcd.print(F("<-    4.7L    "));
}
/*void pantalla_case_5_9(){
  lcd.setCursor(0,0);
  lcd.print(F("''''''MOTOR''''''"));
  lcd.setCursor(0,1);
  lcd.print(F("<-    5.9L    "));
}*/
////////////////////////////////////ngc
void pantalla_ngcv4(){
  lcd.setCursor(0,0);
  lcd.print(F("''''''MODELO''''''"));
  lcd.setCursor(0,1);
  lcd.print(F(" 4  CILINDROS ->"));
}
void pantalla_ngcv6(){
  lcd.setCursor(0,0);
  lcd.print(F("''''''MODELO''''''"));
  lcd.setCursor(0,1);
  lcd.print(F("<- 6 CILINDROS->"));
}
void pantalla_ngcv8(){
  lcd.setCursor(0,0);
  lcd.print(F("''''''MODELO''''''"));
  lcd.setCursor(0,1);
  lcd.print(F("<- 8 CILINDROS  "));
}

////////////////////////////////////nissan
void pantalla_juke(){
  lcd.setCursor(0,0);
  lcd.print(F("''''''MODELO''''''"));
  lcd.setCursor(0,1);
  lcd.print(F("    JUKE    ->"));
}
void pantalla_nissanv6(){
  lcd.setCursor(0,0);
  lcd.print(F("''''''MODELO''''''"));
  lcd.setCursor(0,1);
  lcd.print(F("<-  TSURU    ->"));
}
void pantalla_np300(){
  lcd.setCursor(0,0);
  lcd.print(F("''''''MODELO''''''"));
  lcd.setCursor(0,1);
  lcd.print(F("<-  NP300    ->"));
}

```

```

void pantalla_sentra(){
    lcd.setCursor(0,0);
    lcd.print(F("MODELO"));
    lcd.setCursor(0,1);
    lcd.print(F("<- SENTRA"));
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////toyota
void pantalla_Hilux(){
    lcd.setCursor(0,0);
    lcd.print(F("MODELO"));
    lcd.setCursor(0,1);
    lcd.print(F("HILUX ->"));
}
void pantalla_1nz(){
    lcd.setCursor(0,0);
    lcd.print(F("MODELO"));
    lcd.setCursor(0,1);
    lcd.print(F("<-MOTOR 1NZ_FE->"));
}
void pantalla_1zz(){
    lcd.setCursor(0,0);
    lcd.print(F("MODELO"));
    lcd.setCursor(0,1);
    lcd.print(F("<-MOTOR 1ZZ_FE->"));
}
void pantalla_2az(){
    lcd.setCursor(0,0);
    lcd.print(F("MODELO"));
    lcd.setCursor(0,1);
    lcd.print(F("<-MOTOR 2AZ_FE->"));
}
void pantalla_2tr(){
    lcd.setCursor(0,0);
    lcd.print(F("MODELO"));
    lcd.setCursor(0,1);
    lcd.print(F("<-MOTOR 2TR_FE->"));
}
void pantalla_2gr(){
    lcd.setCursor(0,0);
    lcd.print(F("MODELO"));
    lcd.setCursor(0,1);
    lcd.print(F("<-MOTOR 2GR_FE->"));
}
void pantalla_1gr(){
    lcd.setCursor(0,0);
    lcd.print(F("MODELO"));
    lcd.setCursor(0,1);

```

```

    lcd.print(F("<-MOTOR 1GR_FE->"));
}
void pantalla_Tacoma(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<- TACOMA ->"));
}
void pantalla_Yaris(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<- YARIS ->"));
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////diesel
void pantalla_great_wall(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F(" GREAT WALL "));
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////mazda
void pantalla_mazda6(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F(" MAZDA 6 >"));
}
void pantalla_mazdacx7(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("< MAZDA CX-7>"));
}
void pantalla_mazda626(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("< MAZDA 626 "));
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////BMW
void pantalla_M57D30(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F(" M57D30 ->"));
}

```

```

}
void pantalla_N57D30(){
    lcd.setCursor(0,0);
    lcd.print(F("MODELO"));
    lcd.setCursor(0,1);
    lcd.print(F("<- N57D30 ->"));
}
void pantalla_B46A20(){
    lcd.setCursor(0,0);
    lcd.print(F("MODELO"));
    lcd.setCursor(0,1);
    lcd.print(F("<- B46A20 ->"));
}
void pantalla_B47D20(){
    lcd.setCursor(0,0);
    lcd.print(F("MODELO"));
    lcd.setCursor(0,1);
    lcd.print(F("<- B47D20  "));
}
void pantalla_B58B30(){
    lcd.setCursor(0,0);
    lcd.print(F("MODELO"));
    lcd.setCursor(0,1);
    lcd.print(F("<- B58B30 ->"));
}
void pantalla_M50B25(){
    lcd.setCursor(0,0);
    lcd.print(F("MODELO"));
    lcd.setCursor(0,1);
    lcd.print(F("<- M50B25 ->"));
}
void pantalla_M54B25(){
    lcd.setCursor(0,0);
    lcd.print(F("MODELO"));
    lcd.setCursor(0,1);
    lcd.print(F("<- M54B25  "));
}
////////////////////////////////////citroen
void pantalla_xsara(){
    lcd.setCursor(0,0);
    lcd.print(F("MODELO"));
    lcd.setCursor(0,1);
    lcd.print(F(" XSARA ->"));
}
void pantalla_c4(){
    lcd.setCursor(0,0);
    lcd.print(F("MODELO"));

```

```

    lcd.setCursor(0,1);
    lcd.print(F("<-      C4      ->"));
}
void pantalla_c5(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<-      C5      ->"));
}
void pantalla_ds5(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<-      DS5      ->"));
}
void pantalla_berlingo(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<-  BERLINGO  ->"));
}
void pantalla_jumpy(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<-      JUMPY      "));
}
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////seat
void pantalla_ateca(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("      ATECA      ->"));
}
void pantalla_ibiza(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<-      IBIZA      "));
}
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////renault
void pantalla_master(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("      MASTER      ->"));
}

```

```

void pantalla_clio(){
    lcd.setCursor(0,0);
    lcd.print(F("MODELO"));
    lcd.setCursor(0,1);
    lcd.print(F("<- CLIO ->"));
}
void pantalla_duster(){
    lcd.setCursor(0,0);
    lcd.print(F("MODELO"));
    lcd.setCursor(0,1);
    lcd.print(F("<- DUSTER ->"));
}
void pantalla_scenic(){
    lcd.setCursor(0,0);
    lcd.print(F("MODELO"));
    lcd.setCursor(0,1);
    lcd.print(F("<- SCENIC ->"));
}
void pantalla_sandero(){
    lcd.setCursor(0,0);
    lcd.print(F("MODELO"));
    lcd.setCursor(0,1);
    lcd.print(F("<- SANDERO ->"));
}
void pantalla_laguna(){
    lcd.setCursor(0,0);
    lcd.print(F("MODELO"));
    lcd.setCursor(0,1);
    lcd.print(F("<- LAGUNA ->"));
}
void pantalla_megane(){
    lcd.setCursor(0,0);
    lcd.print(F("MODELO"));
    lcd.setCursor(0,1);
    lcd.print(F("<- MEGANE  "));
}
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////peugeot
void pantalla_207(){
    lcd.setCursor(0,0);
    lcd.print(F("MODELO"));
    lcd.setCursor(0,1);
    lcd.print(F(" 207 ->"));
}
void pantalla_407(){
    lcd.setCursor(0,0);
    lcd.print(F("MODELO"));
    lcd.setCursor(0,1);

```



```

    lcd.print(F("<-    407    ->"));
}
void pantalla_607(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<-    607    ->"));
}
void pantalla_807(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<-    807    ->"));
}
void pantalla_partner(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<- PARTNER ->"));
}
void pantalla_boxer(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<-    BOXER    "));
}
//////////////////////////////////////AUDI
void pantalla_a419(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("    A4  1.9L  ->"));
}
void pantalla_a619(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<-  A6  1.9L  ->"));
}
void pantalla_a624(){
    lcd.setCursor(0,0);
    lcd.print(F("''''''MODELO''''''"));
    lcd.setCursor(0,1);
    lcd.print(F("<-  A6  2.4L  ->"));
}
void pantalla_q730(){
    lcd.setCursor(0,0);

```

```
lcd.print(F("''''MODELO''''"));  
lcd.setCursor(0,1);  
lcd.print(F("<- Q7 3.0L  "));  
}
```