

INSTITUTO TECNOLÓGICO SUPERIOR DEL SUR DE GUANAJUATO



BANCO DE PRUEBAS PARA PIEZAS AUTOMOTRICES

Opción 2: Tesis Integral - Tesis Profesional

Elaborada por:

Orlando Balcázar García

Que presenta para obtener el título de:

INGENIERO ELECTRÓNICO

Asesor:

Ing. Netzahualcóyotl Martínez Cázares

Uriangato, Gto.

Noviembre de 2022

“BANCO DE PRUEBAS PARA PIEZAS AUTOMOTRICES”

Elaborada por:

Orlando Balcázar García

Aprobado por.

Ing. Netzahualcóyotl Martínez Cázares

Docente de la carrera de Ingeniería Electrónica

Asesor de la tesis profesional.

Revisado por.

M.C. Susana Violeta Martínez Hernández

Docente de la carrera de Ingeniería Electrónica

Revisora de la tesis profesional.

Revisado por.

M.C. Mariano Braulio Sánchez

Docente de la carrera de Ingeniería Electrónica

Revisor de la tesis profesional.

Revisora de la tesis profesional.



LIBERACIÓN DE PROYECTO PARA LA TITULACIÓN INTEGRAL

Uriangato, Gto., 10/11/2022

Asunto: Liberación de proyecto para la titulación integral

Ing. J. Trinidad Tapia Cruz
Director Académico
ITSUR
PRESENTE

Por este medio informo que ha sido liberado el siguiente proyecto para la titulación integral:

Nombre de estudiante y/o egresado(a): Orlando Balcazar Garcia	
Carrera: Ingeniería en electrónica	Núm. de control: E15120265
Nombre del proyecto: "Banco de pruebas para piezas autométricas".	
Producto: Tesis profesional	

Agradezco de antemano su valioso apoyo en esta importante actividad para la formación profesional de nuestras y nuestros egresados.

ATENTAMENTE

Dr. J. El Eduardo González Durán
Jefe de división de Ingeniería Electrónica
ITSUR



La comisión revisora ha tenido a bien aprobar la reproducción de este trabajo.

Ing. Netzahualcóyotl Martínez Cazarez Asesor de la tesis profesional	M.C. Susana Violeta Martínez Hernandez Revisor 1	M.C. Mariano Braulio Sanchez Revisor 2



Instituto Tecnológico Superior del Sur de Guanajuato
Dirección Académica
División de Ingeniería Electrónica

DEPARTAMENTO ACADEMICO	CLAVE:ITEIT0002E IELE-EGR-2022/14
---------------------------	--------------------------------------

Uriangato, Guanajuato, **11/noviembre/2022**

Asunto: Aprobación de impresión de trabajo profesional

C. Orlando Balcazar García
PRESENTE:

Por medio de este conducto, le comunico a usted que después de haber sido revisado su trabajo de titulación bajo la del cual se derivó la Monografía Titulada:

“ Banco de pruebas para piezas automotrices .”

La comisión revisora, ha tenido a bien aprobar la impresión de este trabajo.

ATENTAMENTE
"Excelencia en Educación Tecnológica"
"Tecnología y Calidad para la Vida"

Dr. J. Eli Eduardo González Durán
Jefe de División de Ingeniería Electrónica

Instituto Tecnológico Superior
del Sur de Guanajuato



**COORDINACIÓN INGENIERÍA
ELECTRÓNICA**

c.p Unidad de Servicios Escolares
C.c.p Coordinación de Ingeniería Electrónica
C.c.p Archivo Consecutivo



Ave. Educación Superior No. 2000, Col. Benito Juárez, Uriangato, Guanajuato, C.P. 38980
Tels. (445) 45 7 74 68 al 71 Ext. *137, e-mail: electronica@itsur.edu.mx
tecnm.mx | itsur.edu.mx



2022 Flores
Año de Magón
PRESENCIA DE LA EDUCACIÓN TECNOLÓGICA

Resumen:

En el presente documento se muestra un banco de piezas automotrices similar al que existe en las líneas finales de ensamblaje. Un banco de piezas automotrices pone a prueba la calidad y el funcionamiento de los artículos fabricados, que generalmente son pequeños productos como switches, botones, interruptores, etc., cuyo destino es el mercado de piezas automotrices.

La finalidad de un banco de pruebas es demostrar que dichas piezas de fabricación funcionan bien, y en este documento se expone la creación de uno de estos dispositivos que es igual de eficiente a los que ya existen en el mercado y es menos costoso. Este banco de pruebas para piezas automotrices será de gran utilidad ya que por medio de este se podrá observar el correcto funcionamiento en las piezas automotrices, que consiste en tres tipos: funcionamiento electrónico, funcionamiento mecánico e iluminación en los diferentes tipos de piezas. Puede ser utilizado en diferentes empresas con tecnología relativamente económica a comparación de un probador final de línea de piezas automotrices que usualmente compran estas empresas para dicho uso, con máquinas muy grandes y robustas de probadores finales y cuyo costo va de 1 millón hasta 5 millones de pesos, a comparación de este banco de pruebas que su costo aproximado es de 230,000 pesos.

En este banco de pruebas para piezas automotrices se llevará a cabo un proceso en el cual será sometida la pieza a diferentes verificaciones de calidad tales como inspección visual de la pieza en general para detectar malformaciones en ella, así como el buen iluminado de la pieza, si esta presenta una buena soldadura de estaño en las superficies de cobre donde se soldan los componentes electrónicos en una tarjeta de circuito impreso, mejor conocidas como pad's; y en las terminales de los componentes electrónicos en general, inspección mecánica al momento de recorrer los botones y/o presionarlos que todo esté en el correcto ensamble, se muestran también las mediciones eléctricas como lo son voltajes, corrientes, y resistencias de la pieza. Así mismo el análisis de las secuencias que debe de hacer la pieza en su correcto funcionamiento, dado que la secuencia puede variar según las variantes de la pieza en sí.

El presente documento muestra también la documentación realizada para hacer el banco de pruebas de piezas automotrices en el cual se hace el diseño eléctrico, con el que se van a hacer los distintos tipos de conexiones con el hardware utilizado y el software correspondiente para cada tipo de pieza automotriz requerida. Los resultados se obtienen tomando como base los distintos bancos de prueba que ya se tienen en las diferentes empresas actualmente, ya que en todas ellas existe un área de calidad para inspeccionar y aprobar el buen funcionamiento de estas piezas, ya que son sometidas a distintas verificaciones que se inspeccionan, con regularidad.

Palabras claves

Banco de pruebas, Línea de producción, Probador final de línea, Plan de pruebas, Pruebas, Inspección.

Abstract:

This paper shows an automotive parts bench similar to the one that exists on final assembly lines. The purpose is to demonstrate that it works well and is less expensive. This test bench for automotive parts will be very useful because it will be possible to observe the correct operation of the parts, which consists of three types: electronic operation, mechanical operation and lighting in the different types of parts. It can be used in different companies with relatively inexpensive technology compared to a final tester of automotive parts line that usually buy these companies for such use, with very large and robust machines of final testers and whose cost ranges from 1 million to 5 million pesos.

In this test bench for automotive parts will be carried out a process in which the part will be subjected to different quality checks such as visual inspection of the part in general, malformations in it, as well as the good lighting of the piece, if it has a good tin soldering in the pad's and terminals of the electronic components in general, mechanical inspection when going through the buttons and / or press them that everything is in the correct assembly, electrical measurements such as voltages, currents, and resistances. Also the analysis of the sequences that the part must make in its correct operation, since the sequence may vary according to the variants of the part itself.

This document also shows the documentation made to make the test bench of automotive parts in which the electrical design is made, with which the different types of connections are going to be made with the hardware used and the corresponding software for each type of automotive part required. The results are obtained based on the different test benches that already exist in the different companies at present, since in all of them there is a quality area to inspect and approve the proper functioning of these parts, since they are subjected to different verifications that are inspected on a regular basis.

Key Words:

Testing bench, Production line, End of line tester, Test plan, Tests, Inspection.

Agradecimientos

Agradezco primeramente a Dios por brindarme una vida llena de aprendizajes, bendiciones y felicidad.

Agradezco a mis padres y hermanos por su apoyo incondicional hacia mí, consejos y bendiciones, por estar a mi lado día a día.

A mi mamá por su apoyo incondicional por su comida y siempre apoyarme con sus sagrados alimentos y sus buenos consejos, por siempre estar pendiente de mis horarios y por siempre poner mis aprendizajes y estudios académicos por sobre lo más importante.

A mi papá por su incondicional apoyo, por siempre brindarme lo mejor y apoyarme en todas las cosas que hago, y siempre estar a mi lado por su gran valentía y fuerza de siempre estar trabajando y nunca darse por vencido.

A la memoria de mi abuelita Flora Guzmán González por haberme dado todo su amor, su cariño, sus bendiciones todo el apoyo incondicional desde niño hasta adulto, siempre te recordare con cariño sonriendo, porque hiciste de mi vida algo bonito, te querré siempre.

Agradezco mucho a la Maestra Susana Violeta Martínez Hernández y al Maestro Netzahualcóyotl y por su incondicional apoyo durante toda mi formación y en especial por el cumplimiento de fecha y forma de esta tesis y siempre haberme apoyado ética y moralmente durante toda mi formación en la universidad.

A la memoria del Maestro Fernando Torres Rangel le agradezco incondicionalmente por haberme guiado desde el principio de la carrera y siempre ser mi motivación, mi maestro, mi amigo y haberme apoyado en la carrera con sus consejos y fuerza de voluntad cuando no tenía motivación, siempre estaré infinitamente agradecido con usted maestro (vivir en los corazones que dejamos atrás no es morir).

Agradezco en especial y con suma atención a toda la academia de electrónica a la maestra Melisa Torres Rangel, al maestro Leonel Estrada Rojo al maestro Julio Ortega Alejos al Dr. Carlos Alberto Fuentes Hernández al Dr. José Eli Eduardo Gonzales Durán al maestro Mariano Braulio Sánchez y agradezco mucho al maestro Rogelio Orozco Martines por todo su apoyo brindado para mis estudios y en la realización de este proyecto de tesis, gracias por la ayuda y por creer en mí, por abrirme las puertas al conocimiento, darme la oportunidad de estudiar y alcanzar un peldaño más en mi vida profesional. Gracias a todos los maestros que intervinieron en mi formación porque con sus conocimientos y apoyos me prepararon y me dieron herramientas muy completas para enfrentar la competencia del mundo laboral.

Orlando Balcázar García

Dedicatoria

Dedico este proyecto a todas las personas que me apoyaron, no solo en mi desarrollo, si no en la posibilidad de estar en el lugar que estoy, a todas esas personas que con sus consejos y ánimos me ayudaron a progresar con mención especial a mis padres y hermanos, incluso en los tiempos más difíciles, esas personas que creyeron en mí y en mis capacidades para seguir adelante, cada una de esas personas con su grano de arena impulsaron a el resultado que se ve hoy en día, este proyecto no es un logro solo mío, sino de todos ellos, y sin más sigamos mejorando.

Orlando Balcázar García

Tabla de Contenido

Capítulo 1	1
Introducción.....	1
Capítulo 2.....	1
Marco teórico (Antecedentes).....	1
Industria Automotriz:	1
Línea de producción:.....	1
Fases de prueba y bancos de prueba.....	2
Fase de pruebas del desarrollo del banco de pruebas para piezas automotrices.....	2
Banco de prueba Bogotá.....	5
Banco de pruebas para comprobación y verificación de computadoras automotrices con inyección electrónica a gasolina.....	6
Vortex-dV Probador de par motorizado.....	7
Bancos de pruebas para arboles de transmisión:	7
Capítulo 3.....	1
Planteamiento del problema.....	1
3.1. Identificación.....	1
3.2. Justificación. 1	
3.3. Alcance. 2	
Capítulo 4.....	1
Objetivos	1
4.1. Objetivo General	1
4.2. Objetivos específicos	1
Capítulo 5.....	2
Metodología.....	2
Arduino	4
¿Qué es Arduino?	4
Software Arduino (IDE)	6
Hardware ARDUINO MEGA 2560.....	7
ShieldBuddy para AURIX TC275	9
¿Qué se ha integrado en la placa?	10

HMI 4D Systems	11
Workshop4	13
Genie Arduino	14
Protocolos de comunicación.....	14
Protocolo de comunicación CAN.....	14
Protocolo de comunicación LIN.....	16
Convertor analógico digital	17
Bancos de prueba	19
“Assist” de “KIA”	19
“Rear” de “KIA”	33
Capítulo 6	45
Resultados	45
Nuevos Proyectos con “KIA”	45
“Head Lamp” de “General Motors”	47
Capítulo 7	49
Análisis de Resultados	49
Capítulo 8	52
Conclusiones y trabajo a futuro	52
Capítulo 9	54
Referencias bibliográficas	54
Anexos	56

Índice de Figuras

Figura 1 Representación línea de producción. (Freepng, 2022)	2
Figura 2 Banco de prueba de computadora automotriz (ECU) (Autosoporte, s.f.).	4
Figura 3 Fuente de alimentación DC programable (Carecarcenter, 2021).	5
Figura 4 Banco de pruebas (Eduardo, 2006).	6
Figura 5 Tarjeta multímetro Keysight m9182A (Mecmecin, 2021).	7
Figura 6 Bancos de pruebas del funcionamiento para arboles de transmisión (Blumnovotest, s.f.).	8
Figura 7 Arduino IDE.	7
Figura 8 Arduino Mega 2560.	8
Figura 9 ShieldBuddy TC275.	10
Figura 10 Pantalla GEN4-ULCD-70DT (Systems, 4D Systems turning technology into art, 2022).	13
Figura 11 Software Workshop4 (Systems, 4D Systems turning technology into art, 2022)	14
Figura 12 . Conversión análogo-digital (Huircán, s.f.).	18
Figura 13 . Logo de la empresa KIA (KIA, 2021).	19
Figura 14 Pieza "Assist" de KIA.	20
Figura 15 Nido de la fixtura "Assist".	21
Figura 16 Bloqueo y desbloqueo de puertas.	22
Figura 17 Subir y bajar ventanas.	22
Figura 18 Secuencia de subir-bajar ventana.	23
Figura 19 Interfaz de inicio.	23
Figura 20 . Interfaz para probar la pieza.	25
Figura 21 Añadiendo candado blanco a las imágenes del programa.	25
Figura 22 Añadiendo candado verde a las imágenes del programa.	26
Figura 23 Lecturas de "ADCs" y "HC".	27
Figura 24 Pieza "Rear" de KIA.	34
Figura 25 Fixtura que trabaja con una tarjeta "ShieldBuddy TC275".	34
Figura 26 Botón calefactor.	35
Figura 27 Interfaz para seleccionar el tipo de pieza ("LC" o "HC").	36
Figura 28 Interfaz para probar la pieza de "LC".	37
Figura 29 Interfaz para probar la pieza de "HC".	38
Figura 30 Fixtura "Assist" trabajando en contención.	46
Figura 31 Fixtura "Rear" trabajando.	47
Figura 32 Fixtura "Driver" trabajando en contención.	47
Figura 33 Fixtura "Head lamp" trabajando.	48
Figura 34 Menú para seleccionar la pieza a probar.	50
Figura 35 . Probando la pieza "Mini ICP" en la fixtura.	50
Figura 36 Probando la pieza "CX430 en la fixtura".	51

Índice de Tablas

<i>Tabla 1 Especificaciones técnicas del Arduino MEGA 2560 (Arduino, 2022).</i>	9
<i>Tabla 2 Valores de accionamiento de las piezas LC y HC</i>	27
<i>Tabla 3 Valores de accionamiento de las piezas LC yHC</i>	39

Capítulo 1

Introducción.

Actualmente las tecnologías avanzan a pasos agigantados a causa de las necesidades ya sea en la calidad de los productos, como por la propia competencia entre las distintas empresas en la industria. En todos los sectores es necesario tener un crecimiento tecnológico, así como una mejora y optimización permanente de todos los procesos o productos que se crean, para que la tecnología avance, y con ella la sociedad. Esto significa realizar cambios o innovar dispositivos o maquinaria ya creada, para optimizar el funcionamiento haciendo más fácil su manejo, abaratando costos y ahorrando tiempo. Este es el caso del dispositivo presentado en este documento.

Es necesario por parte de las grandes industrias hacer investigaciones e inversiones en sus plantas para tener siempre productos con tecnología de punta y mayor calidad. Para cumplir los altos estándares de calidad y una constante optimización tecnológica, es necesario migrar a sistemas automatizados. Un ejemplo claro es la industria automotriz, en la cual día con día se buscan estrategias y metodologías que optimicen las líneas de producción, algunas de estas estrategias son: uso de sistemas de visión artificial, sistemas automáticos de mediciones eléctricas, HAPTIC (que es un estándar de prueba de calidad). De manera que con cada uno de estos procesos se deben agilizar las línea de producción, y en el caso de que sigúan trabajando con tecnologías obsoletas emplear estos bancos para pruebas automotrices, gracias a estos bancos para pruebas automotrices se plantea la mejora del proceso actual, dando como resultado una línea de producción mucho más ágil y de optimizada, lo cual se traduce en beneficio monetario por el aumento de producción y por la reducción de posibles reclamos de cliente por la mejora de calidad a la hora de realizar tanto las pruebas funcionales mecánica y eléctricamente. Una herramienta ampliamente utilizada en la industria, para la inspección de calidad de piezas es el uso de técnicas de visión artificial.

En este documento se presenta el desarrollo y proceso de creación de los bancos de prueba para pruebas de piezas, este banco de pruebas para piezas automotrices servirá porque con él se podrá observar el correcto funcionamiento en las piezas automotrices.

La creación de sistemas que no solo mejoren dispositivos que tienen las funciones arriba descritas, sino que disminuyan los costos y faciliten el trabajo; es lo que se describe a través de este documento, el cual muestra el proceso de creación de un banco de pruebas en estos se probaran piezas automotrices que tiene la finalidad de pasar los estándares de calidad de la empresa, checando su óptimo funcionamiento de las piezas automotrices; aun cuando las compañías ya poseen estos tipos de sistemas, el que aquí se propone es económico y más eficiente.

La importancia del dispositivo radica en que realiza la inspección visual de la pieza en general, malformaciones en ella, así como el buen iluminado de la pieza, si esta presenta una buena soldadura de estaño en los pad's que son los puntos de contactación eléctricamente con los botones, también se conoce el estado de las terminales de los componentes electrónicos en general, inspección mecánica al momento de recorrer los botones y/o presionarlos que todo esté en el correcto ensamble, mediciones eléctricas como lo son voltajes, corrientes, y resistencias. Aunque existen desde hace años estos dispositivos, ciertamente implican un alto costo para las empresas y éstas no pueden prescindir de ellos; mientras que el dispositivo aquí expuesto puede competir debido a un menor costo y una mejora en el funcionamiento para verificar las piezas. Como todos sabemos la tecnología tiende a mejorar a medida que se perfeccionan los funcionamientos de sus mecanismos.

Por otro lado, esta innovación presentada por el nuevo dispositivo es importante debido a que reduce significativamente costos y espacio.

Capítulo 2

Marco teórico (Antecedentes).

Industria Automotriz: está compuesta por todas las empresas dedicadas a producir y comercializar vehículos terrestres de autopropulsión entre los que se encuentran los automóviles de pasajeros, autobuses, camiones, equipos agrícolas y otros vehículos comerciales. Los automotores han propiciado que el público pueda viajar largas distancias ya sea para trabajar, hacer compras o para realizar viajes de recreación, esta industria fomenta el desarrollo de los diferentes sistemas de carreteras. La industria automotriz se ha convertido en uno de los mayores consumidores de productos primarios dentro de la industria, como el acero. La gran cantidad de personas empleadas por ella, la convierten en un sector industrial de gran valor para la economía de los países en los que esta se asienta (Abadiano Abadiano, et al 2020).

Línea de producción: conjunto de operaciones secuenciales en las que se organiza un proceso para la fabricación de un producto. Para la fabricación de un gran número de unidades del mismo producto, se requiere organizar un montaje en serie de las distintas operaciones requeridas para su transformación de materias prima en producto. En la figura 1, se puede apreciar un ejemplo de una línea de producción.



Figura 1 Representación línea de producción. (Freepng, 2022)

Fases de prueba y bancos de prueba.

Durante muchos años, existen compañías que han estado ofreciendo servicios de certificación y pruebas automotrices nacionales e internacionales para piezas automotrices, con la finalidad de verificar la calidad en seguridad, confiabilidad, longevidad, durabilidad y funcionalidad de dichas piezas.

Los fabricantes de piezas automotrices ven con el compromiso de garantizar calidad en sus productos y por lo tanto contratan dispositivos de verificación de calidad, también llamados bancos de pruebas, con esta finalidad. El proceso de fabricación de cada producto debe atravesar por una fase de prueba que demuestre una fabricación óptima y la mejor calidad. Con este propósito se contratan bancos de pruebas.

Fase de pruebas del desarrollo del banco de pruebas para piezas automotrices.

La fase de pruebas en el desarrollo de un producto es absolutamente necesaria, ya que dependiendo de esta fase se definirá la calidad de este. Estos tipos de bases o bancos de prueba consisten en dispositivos que permiten revisar el funcionamiento de las piezas automotrices producidas en las fábricas para confirmar el control de calidad antes de su entrega a otras compañías. Las pruebas son diversas y dependerán de las características y diseño del producto, para ello se requieren bancos de pruebas ya sea automáticos o manuales (Autosoporte, s.f.).

Capítulo 2. Marco teórico (Antecedentes).

Existen vastos bancos de prueba en los diferentes tipos de sectores empresariales y de fabricación, como lo son; bancos de pruebas para helicópteros, para el sector aeronáutico, bancos de pruebas para motores eléctricos, esta tesis se centrará en el sector auto motriz, en específico en piezas electrónicas automotrices.

Se proporcionan soluciones a medida y completamente diseñadas con una ingeniería que se adapte para necesidades específicas del cliente en el sector automovilístico. Los requisitos clave para dichos bancos de pruebas son: amigable con el usuario, funcionalidad, flexibilidad y seguridad. Y por medio de soluciones avanzadas es posible ayudar a los clientes a cumplir con los requerimientos y estándares de calidad y buen funcionamiento y enfrentarse a nuevos desafíos por lo que respecta a los accionamientos eléctricos. También es posible proveer mantenimiento y asistencia de los dispositivos a las empresas que lo requieran.

De acuerdo a la empresa NIDEC (NIDEC, 2022), sus aplicaciones típicas incluyen:

- Bancos de prueba para vehículos eléctricos, con alimentación en CC para verificar el inversor del vehículo.
- Bancos de prueba para el ciclo de flexión, para ensayos de materiales.
- Bancos de prueba para convertidores de par para el desarrollo y el aseguramiento de la calidad.
- Bancos de prueba para motores para la producción de motores eléctricos y de combustión.
- Bancos de prueba para transmisiones/cajas de cambios para proveedores de la industria automovilística y de fabricación de maquinaria.
- Bancos de pruebas de frenos.
- Bancos de pruebas de rodillos para fabricantes de automóviles y motos.

Otros equipos de bancos de prueba son los siguientes:

Equipo de ECU Banco de Prueba de Computadora Automotriz (ECU): Es una caja plástica con canales y terminales a puntas como multímetro, que tiene una eficiencia en el diagnóstico

Capítulo 2. Marco teórico (Antecedentes).

de defectos; es el llamado VRS-565 Turbo III, este es un dispositivo creado por la empresa Autosoporte.

El VRS-565 Turbo III el cual se aprecia en la figura 2, es compatible para cualquier tipo de placa electrónica, atiende a autónomos, pequeñas, medianas y grandes empresas de todos los segmentos como Automotriz, Electrónica Industrial, Tecnologías de la Información, Naval, Aviación, Refrigeración, Textil. Gráficos, etc. Sus funciones especiales facilitan la prueba y el análisis de tarjetas electrónicas, lo que permite un diagnóstico más rápido y preciso, introduce una nueva cultura en el diagnóstico de placas electrónicas, lo que aumenta significativamente la productividad de su laboratorio. Es una opción con una buena relación calidad-precio. Es un instrumento virtual compuesto por un software y un módulo de adquisición conectado al USB de la computadora. Localiza el defecto comparando las curvas características de la tarjeta defectuosa con las curvas de una tarjeta igualmente buena. El usuario puede memorizar las placas, permitiendo almacenar los conocimientos adquiridos en archivos. Creando su propio banco de curvas, que se puede utilizar en futuras reparaciones, eliminando la necesidad de tener siempre una buena placa física.



Figura 2 Banco de prueba de computadora automotriz (ECU) (Autosoporte, s.f.).

Características:

- Número de canales con mayor precisión.

Capítulo 2. Marco teórico (Antecedentes).

- VeRSis Curve Bank.
- Herramientas de análisis e informes para mayor documentación o análisis.
- Ayudas gráficas.
- Funciones de edición de archivos.
- Navegación de archivos.
- Guía de sonido y ayuda contextual

Banco de prueba Bogotá (Carecarcenter, 2021): El banco de prueba de chasis se usa para restaurar las medidas originales del chasis de un vehículo, este proceso se realiza mediante fuerzas hidráulicas sobre los anclajes del chasis, las medidas de alineación de un vehículo se expresan mediante tres parámetros: Camber o caída, caster o avance, convergencia o toe el cual se aprecia en la figura 3.



Figura 3 Fuente de alimentación DC programable (Carecarcenter, 2021).

Características:

- Corrección Caídas (Camber).
- Corrección Avance (Caster).
- Corrección Convergencias (Toe).
- Reconstrucción Puntas Chasis.
- Soldaduras MIG.

Banco de pruebas para comprobación y verificación de computadoras automotrices con inyección electrónica a gasolina (Eduardo, 2006): Este banco de pruebas diseñado tiene la característica principal de generar señales de accionamiento por medio de puntos de prueba generalizados con el objetivo de poner en funcionamiento la UCE de tecnología EEC-IV que va a ser diagnosticada. Como se puede apreciar en la figura 4, todos los valores y señales de cada componente del banco de pruebas están previamente analizado y tomado de los respectivos catálogos de servicio de los sistemas EEC-IV con la finalidad de lograr un preciso y eficaz funcionamiento de todos los parámetros principales utilizados en las ECM aplicados en la inyección electrónica.

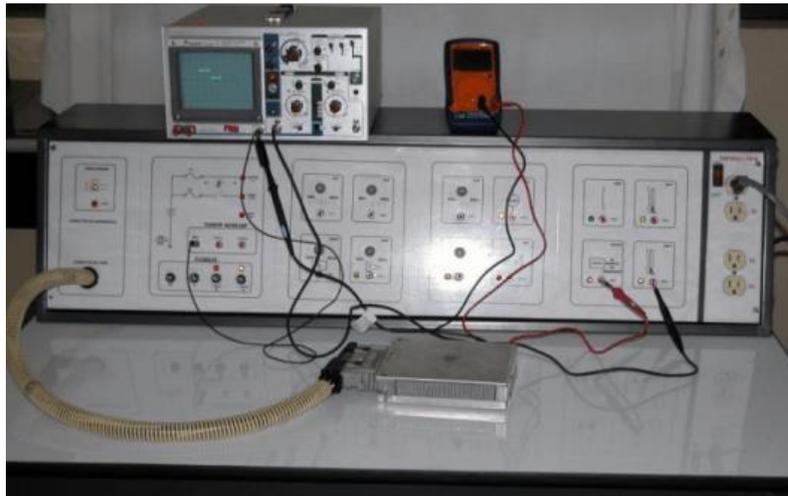


Figura 4 Banco de pruebas (Eduardo, 2006).

Características:

- Medición de aire y combustible.
- Medición de aire y combustible (Circuito inyector).
- Sistemas encendidos.
- Controles Auxiliares de emisiones.
- Velocidad y marcha lenta.
- Circuito de salida de la computadora.

Vortex-dV Probador de par motorizado (Mecmecin, 2021): Como se aprecia en la figura 5 es un probador de par autónomo, versátil, robusto y de 10 Nm. Fácil de usar, es adecuado para controles de calidad de rutina en un entorno de producción o laboratorio. Este sopórtese configura accionado por motor con un sensor de torque conectado a una unidad de visualización, una selección de accesorios de agarre y software opcional.



Figura 5 Tarjeta multímetro Keysight m9182A (Mecmecin, 2021).

Bancos de pruebas para arboles de transmisión: El banco de pruebas sirve para medir los valores de funcionamiento en articulaciones y árboles de transmisión. Con el banco de pruebas del funcionamiento se ejecutan diferentes ensayos de medición de la tecnología y mediciones relevantes para la seguridad.



Figura 6 Bancos de pruebas del funcionamiento para arboles de transmisión (Blumnovotest, s.f.).

Características de este banco de prueba mostrado en la figura 6 son:

- Momento de inercia.
- Diagrama de la trayectoria del ángulo.
- Ángulo de torsión admisible.
- Fuerza estática de desplazamiento sin par, sin número de revoluciones, con y sin ángulo de flexión.
- Momento de flexión sin par y sin número de revoluciones.

Capítulo 3

Planteamiento del problema

3.1. Identificación.

Dentro de las empresas se utilizan las fixturas (bancos de prueba) principalmente para las áreas como análisis o calidad (área encargada de revisar las fallas de todas las piezas producidas en la empresa), contención (área en la que se realiza una segunda validación de que la pieza funciona correctamente), refacciones (líneas de producción que solamente producen refacciones). Además de estas áreas, en muchas ocasiones las fixturas son llevadas a otros estados o países para validar que las piezas que se produjeron en dicha empresa mexicana funcionan correctamente.

3.2. Justificación.

Las razones fundamentales que conllevan a plantearse y desarrollar este proyecto tienen que ver directamente con los nuevos proyectos dentro de las empresas. Las compañías están produciendo nuevas piezas y las diferentes áreas de la empresa (análisis, contención, etc.) necesitan sujetadores, o comúnmente conocidos como fixturas; para probar estas piezas, también existe la necesidad de crear nuevas fixturas para piezas que ya se producían, ya que se necesita más de una fixtura dentro de la planta debido a que hay más de un área que puede necesitar la fixtura al mismo tiempo. Además, hay fixturas que necesitan ser modificadas debido a que hubo cambios en la pieza para la cual fueron creada, o simplemente la fixtura no funcionaba correctamente.

Realizar fixturas para las empresas sirve para que el personal que labora en las diversas actividades que implican el uso de estos bancos de prueba, pueda realizar el análisis eléctrico de alguna pieza, de una forma sencilla y logre alcanzar un mejor desempeño, ya que al momento de realizar los análisis utilizando dispositivos tales como multímetros y osciloscopios, pueden llegar a presentarse más errores, debido a que son dispositivos más complicados de usar que necesitarían de una capacitación para su correcto uso. Además de que las fixturas son un instrumento importante en el análisis eléctrico de las piezas, persiguen la mayor eficiencia en la ejecución del trabajo asignado al personal, permitiendo

aumentar la calidad, para alcanzar los objetivos de la empresa y propiciando la mejora continua de los productos.

3.3. Alcance.

Por todo lo ya mencionado este proyecto está enfocado en desarrollar los bancos de pruebas para piezas automotrices dirigido a las empresas la que lo vayan requiriendo de acuerdo con las necesidades que vayan surgiendo, como lo pueden ser:

- Nuevos proyectos: así se les llama a las nuevas piezas que empezarán a producir dichas empresas, las cuales no tienen fixturas listas para que utilicen los operadores por ser pieza nueva.
- Nuevas versiones: se empieza a producir una nueva versión de una pieza y esta pieza manda diferentes lecturas de voltaje o cambian sus pines de conexión.
- Demanda de fixturas: se necesita más de una fixtura para una misma pieza debido a que se está usando una fixtura permanentemente en alguna contención y el área de calidad o análisis necesita usarla.
- Obsolescencia: la fixtura que ya existe está muy vieja y puede estar fallando constantemente o ser muy pesada tener muchos aparatos viejos, grandes e innecesarios lo que la hace poco práctica para llevarla de un lugar a otro, en este acaso renovarla o cambiarla por una nueva y de materiales moderno.

Capítulo 4

Objetivos

4.1. Objetivo General

Desarrollar bancos de pruebas (fixturas) para la detección de fallas eléctricas o de comunicación en las piezas producidas en los diferentes tipos de empresas.

4.2. Objetivos específicos

- Crear un dispositivo (banco de pruebas) que sea fácil de usar a comparación de los existentes.
- Desarrollar un dispositivo (bancos de pruebas) que sea económicamente viable.
- Desarrollar un dispositivo (banco de pruebas) cuya creación implique poco tiempo y más eficiencia que los ya existentes.
- Probar que es posible diseñar bancos de pruebas funcionales, económicos y confiables.

Capítulo 5

Metodología

La realización del proyecto se llevó a cabo en la empresa Kostal Mexicana S.A de C.V, donde con el apoyo de los ingenieros del departamento de Calidad-Análisis, se desarrolló el proyecto “Bancos de pruebas para piezas automotrices” utilizando los conocimientos adquiridos en el periodo de residencias. Se contempló el cumplimiento de los objetivos en las fechas acordadas entre 24 de enero del 2022 al 22 de Julio del 2022.

El lugar donde se desarrollaron las actividades de este proyecto fue en el área “AQP1” mejor conocida como “Análisis”, en ésta hay un grupo de operadores e ingenieros trabajando. El grupo de operadores que laboran en esta área reciben las piezas que presentan las fallas en las líneas de producción de la planta, las revisan según la falla, si la falla fue por atascamiento de un botón, fuerza excesiva en un botón (fuerza aplicada fuera del rango), etc. los técnicos del área revisaron estas pieza externamente e internamente para ver qué es lo que estaban generando ese problema (que la pieza contenga malformaciones externamente, que la pieza contenga alguna contaminación internamente, que alguno de los botones este fuera de especificaciones, mal ensamble, etc.) y dieron una respuesta a la línea de producción para saber si fue un caso aislado o hay un riesgo de que más piezas presentaran la misma falla, además de que sirvió como retroalimentación si es que la falla la estaba generando un factor humano.

En el caso de que la falla hubiese sido por secuencia (un botón no mandó la señal eléctrica) o por iluminación (algún botón no ilumina correctamente), los operadores del área pudieron hacer el uso de los bancos de pruebas denominados “Fixturas”, que facilitan la detección de fallas rápidamente en las piezas.

Generalmente hay una fixtura para cada pieza que se produce en KOMEX II (empresa Kostal Mexicana), estas fixturas sirven para que prueben las piezas y mediante Leds, o pantallas (dependiendo la fixtura), puedan ver si la pieza está dando la secuencia de señales y ver si funciona correctamente la pieza, o en efecto tiene alguna falla. En el caso de la

iluminación, al conectar la pieza a la fixtura sus pines estarán alimentados por tanto la pieza se iluminará y esto servirá para checar si la iluminación es correcta.

▪ Aspectos que generalmente se verifican en las piezas por medio del Banco de pruebas:

➤ Garantías: Es cuando una pieza de KOSTAL ha sido utilizada por el cliente final (usuario del vehículo). Estas fallas pueden ser provocadas por contaminación, humedad, daños externos en las piezas, esto es catalogado como “Abuso del cliente”.

➤ Cero kilómetros: Son piezas que salieron de la planta y se enviaron al cliente automotriz. Al ser montadas en el vehículo presentaron alguna falla. Estas fallas pueden ser ocasionadas por mal manejo de las piezas por parte del cliente, mala instalación o un proceso interno de KOSTAL. Como residente se apoya en todas las actividades del área, las principales actividades son:

➤ Revisar las piezas que presentaron alguna falla en las líneas de producción y dar respuesta de la falla a las líneas.

➤ Atender los reclamos que llegan, los cuales implican;

✓ Realizar una inspección externa de la pieza.

✓ Hacer las pruebas de señales con las fixturas (a temperatura ambiente, a -40°C y a $+85^{\circ}\text{C}$).

✓ Realizar una inspección interna de la pieza para ver que el PCB, el “Silicone Pad” y los plásticos están en buenas condiciones.

- ✓ Hacer uso de herramientas más especializadas si la pieza lo requiere (microscopio, cámara de rayos X, etc.).
- ✓ Elaborar un reporte con toda la evidencia de las pruebas e inspecciones realizadas.
- Atender las fixturas, es decir,
- ✓ Dar mantenimiento.
- ✓ Hacer mejoras en la interfaz o en el código.
- ✓ Generar nuevas fixturas, debido a nuevos productos, nuevas versiones de un producto (puede cambiar sus pines o los valores de sus ADC'S), piezas que no tienen fixtura, fixturas faltantes en contención (segunda validación de que la pieza funciona correctamente).

Los aspectos mencionados con anterioridad fueron objeto de verificación en las pruebas llevadas a cabo con el prototipo del banco de pruebas diseñado y descrito en este documento.

El cerebro del banco de pruebas esta creado por medio de un Arduino:

Arduino

¿Qué es Arduino?

Arduino es una plataforma electrónica de código abierto basada en hardware y software fáciles de usar (Arduino, 2022). Las placas Arduino pueden leer entradas como los son (luz en un sensor, un dedo en un botón o un mensaje de Twitter) y convertirlo en una salida, activando un motor, encendiendo un LED, publicando algo en línea. Puede decirle a su

tablero qué hacer enviando un conjunto de instrucciones al microcontrolador en el tablero. Para hacerlo, utiliza el lenguaje de programación Arduino (basado en Wiring) y el Software Arduino (IDE), basado en Processing. A lo largo de los años, Arduino ha sido el cerebro de miles de proyectos, desde objetos cotidianos hasta complejos instrumentos científicos. Una comunidad mundial de creadores (estudiantes, aficionados, artistas, programadores y profesionales) se ha reunido en torno a esta plataforma de código abierto; sus contribuciones se han sumado a una increíble cantidad de conocimiento accesible que puede ser de gran ayuda tanto para principiantes como para expertos.

Arduino nació en el Ivrea Interaction Design Institute como una herramienta fácil para la creación rápida de prototipos, dirigida a estudiantes sin experiencia en electrónica y programación. Tan pronto como llegó a una comunidad más amplia, la placa Arduino comenzó a cambiar para adaptarse a las nuevas necesidades y desafíos, diferenciando su oferta desde placas simples de 8 bits hasta productos para aplicaciones de IoT, wearables, impresión 3D y entornos integrados. Todas las placas Arduino son completamente de código abierto, lo que permite a los usuarios construirlas de forma independiente y eventualmente adaptarlas a sus necesidades particulares. El software también es de código abierto y está creciendo gracias a las contribuciones de los usuarios de todo el mundo.

¿Por qué Arduino?

- Económico: Las placas Arduino son relativamente económicas en comparación con otras plataformas de microcontroladores. La versión menos costosa del módulo Arduino se puede ensamblar a mano, e incluso los módulos Arduino preensamblados cuestan menos de \$ 50 dolares.

- Multiplataforma: El Software de Arduino (IDE) corre en Windows, Macintosh OS X, y Linux. La mayoría de los sistemas de microcontroladores están limitados a Windows.

- Entorno de programación simple y claro: El software Arduino (IDE) es fácil de usar para principiantes, pero lo suficientemente flexible para que los usuarios avanzados también lo aprovechen. Para los profesores, se basa convenientemente en el entorno de programación

de procesamiento, por lo que los estudiantes que aprenden a programar en ese entorno estarán familiarizados con el funcionamiento del IDE de Arduino.

➤ Software de código abierto y extensible: el software Arduino se publica como herramientas de código abierto, disponibles para su extensión por programadores experimentados. El lenguaje se puede expandir a través de bibliotecas C ++, y las personas que quieran comprender los detalles técnicos pueden dar el salto de Arduino al lenguaje de programación AVR C en el que se basa. Del mismo modo, puede agregar código AVR-C directamente en sus programas Arduino si lo desea.

➤ Hardware de código abierto y extensible: Los planos de las placas Arduino se publican bajo una licencia de Creative Commons, por lo que los diseñadores de circuitos experimentados pueden crear su propia versión del módulo, ampliarlo y mejorarlo. Incluso los usuarios relativamente inexpertos pueden construir la versión de tablero del módulo para comprender cómo funciona y ahorrar dinero.

Software Arduino (IDE)

El entorno de desarrollo integrado Arduino, o software Arduino (IDE) mostrado en la figura 7, contiene un editor de texto para escribir código, un área de mensajes, una consola de texto, una barra de herramientas con botones para funciones comunes y una serie de menús. Se conecta al hardware Arduino para cargar programas y comunicarse con ellos.

Los programas escritos con el software Arduino (IDE) se denominan “sketches”. Estos “sketches” se escriben en el editor de texto y se guardan con la extensión de archivo “.ino”. El editor tiene funciones para cortar / pegar y para buscar / reemplazar texto. El área de mensajes proporciona comentarios al guardar y exportar y también muestra errores. La consola muestra la salida de texto del software Arduino (IDE), incluidos los mensajes de error completos y otra información. La esquina inferior derecha de la ventana muestra la placa configurada y el puerto serie. Los botones de la barra de herramientas le permiten verificar y cargar programas, crear, abrir y guardar bocetos y abrir el monitor en serie.

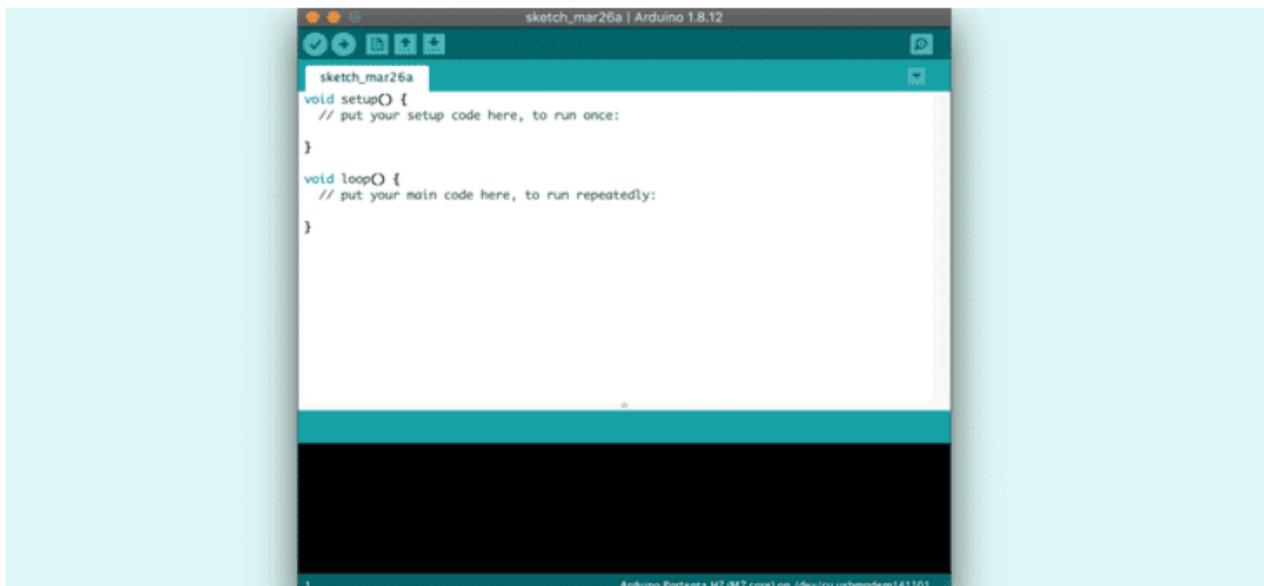


Figura 7 Arduino IDE.

Hardware ARDUINO MEGA 2560

El Arduino Mega 2560 mostrado en la figura 8 es una placa de microcontrolador basada en el ATmega 2560. Tiene 54 pines de entrada / salida digital (de los cuales 15 se pueden usar como salidas PWM), 16 entradas analógicas, 4 UART (puertos serie de hardware), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, un encabezado ICSP, y un botón de reinicio. Contiene todo lo necesario para soportar el microcontrolador; simplemente se conecta a una computadora con un cable USB o enciéndalo con un adaptador de CA a CC o una batería para comenzar. La placa Mega 2560 es compatible con la mayoría de los escudos diseñados para la Uno y las antiguas placas Duemilanove o Diecimila. El Mega 2560 es una actualización del Arduino Mega, al que reemplaza, todas las especificaciones se encuentran en la tabla 1.



Figura 8 Arduino Mega 2560.

Tabla 1 Especificaciones técnicas del Arduino MEGA 2560 (Arduino, 2022).

Especificación	Descripción
Microcontrolador	ATmega 2560
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12 V
Voltaje de entrada (límite)	6-20 V
Pines de E / S digitales	54 (de los cuales 15 proporcionan salida PWM)
Pines de entrada analógica	dieciséis
Corriente CC por pin de E / S	20 mA
Corriente CC para pin de 3.3V	50 mA
Memoria flash	256 KB de los cuales 8 KB utiliza el gestor de arranque
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz
LED BUILTIN	13
Largo	101,52 milímetros
Ancho	53,3 milímetros
Peso	37 g

ShieldBuddy para AURIX TC275

La empresa Hitex presenta la tarjeta ShieldBuddy TC275 (ShieldBuddy, 2018), que es una placa de desarrollo que dispone del procesador de tres núcleos AURIX Infineon TC275. Los procesadores Aurix se han desarrollado específicamente para aplicaciones ADAS de automoción donde la seguridad es muy importante.

La ShieldBuddy es ideal para profesionales, estudiantes o aficionados. Esta placa de desarrollo dispone de licencias de evaluación, y después recibirá acceso completo a todo un conjunto de herramientas de desarrollo mostrada en la figura 9.

La placa de desarrollo ShieldBuddy cuenta con el factor de forma Arduino Mega y proporciona acceso a una amplia gama de placas de accesorios y placas de extensión.

¿Qué se ha integrado en la placa?

- Procesador: MCU de seguridad de tres núcleos TriCore Infineon SAK-TC275T64F200W de 200 MHz.
- Memoria: memoria eFlash ECC de 4 MB, SRAM ECC de 472 kB.
- Seguridad: sistema de supervisión de seguridad en chip CIC61508 para su uso con SafeTlib.
- Alimentación: +5 V por USB o 9-12 V con conector cilíndrico de alimentación.
- Botón: Botón RESET.

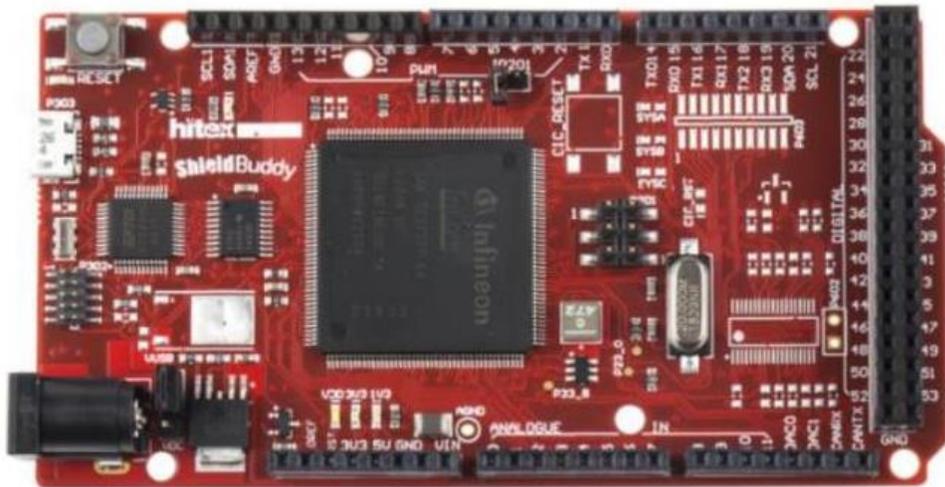


Figura 9 ShieldBuddy TC275.

Detalles técnicos

- Rango de temperaturas de funcionamiento de $-40\text{ }^{\circ}\text{C}$ a $+125\text{ }^{\circ}\text{C}$.
- Interfaz depuración directa USB.
- Dos de los núcleos tienen "sombras" en funcionamiento en bloqueo de reloj para detección de errores/fallos.

- Se suministra con controladores básicos para DS-ADC, SAR, GTM, GPT, CCU6, ASC, LIN, SPI, QSPI y Ethernet.
- Las herramientas de desarrollo disponibles están basadas en Arduino IDE y HighTec Free Aurix Toolchain.
- Compatible con: AUTOSAR V3.2 y V4.x .
- Todos los usuarios de AURIX (ya sean nuevos o futuros) reciben acceso inmediato a más de 300 placas de expansión.
- Hay una placa de control de motor de 30 A dc compatible disponible con el código RS 906-2874.
- Con la CPU TC275, puede obtener de 150 a 200 instrucciones de 32 bits por microsegundo.

HMI 4D Systems

GEN4-ULCD-70DT

Es un módulo de pantalla inteligente de la serie gen4 de 7.0 "con controlador de gráficos DIABLO16. Los módulos de pantalla integrados Diablo16 de 7.0" gen4 son parte de la última serie de módulos gen4 diseñados y fabricados por 4D Systems (Systems, 4D Systems Pty Ltd, 2022).

La serie gen4 fue diseñada específicamente para facilitar la integración y el uso, con una cuidadosa consideración de los requisitos de espacio y la funcionalidad.

Estos módulos gen4 específicos cuentan con una pantalla LCD TFT a color de 7.0" y vienen con opciones para el bisel de la lente de la cubierta (CLB), el tacto resistivo y el tacto

capacitivo. Están impulsados por el conocido procesador gráfico 4D Systems Diablo16, que ofrece una variedad de funciones y opciones para cualquier diseñador / integrador / usuario.

El módulo de pantalla integrado Diablo16 de 7.0" cuenta con una pantalla LCD TFT, es capaz de detección táctil, almacenamiento de memoria microSD, GPIO y comunicaciones, junto con temporizadores de resolución de varios milisegundos y generación de audio, esta pantalla es mostrada en la figura 10.

La serie gen4 es 100% compatible con Workshop4 IDE y sus 4 entornos de desarrollo diferentes, proporcionando al usuario una gran cantidad de opciones para programar y controlar

su sistema.

La serie gen4 de módulos de pantalla integrados cuenta con un zócalo ZIF de 30 pines, diseñado para un cable FFC de 30 pines, para una conexión fácil y simple a una aplicación o placa madre, o para conectar a placas accesorias para una variedad de avances de funcionalidad.

La serie de módulos gen4 se ha diseñado para minimizar el impacto de los circuitos relacionados con la pantalla y proporcionar una plataforma adecuada para la integración en un producto. Las placas de aplicación se pueden colocar al ras en la parte posterior del módulo de pantalla gen4 si es necesario, ya que los componentes electrónicos relacionados con la pantalla se encuentran dentro de la base de montaje de plástico, dejando la superficie de la placa de

aplicación despejada para los circuitos del usuario.

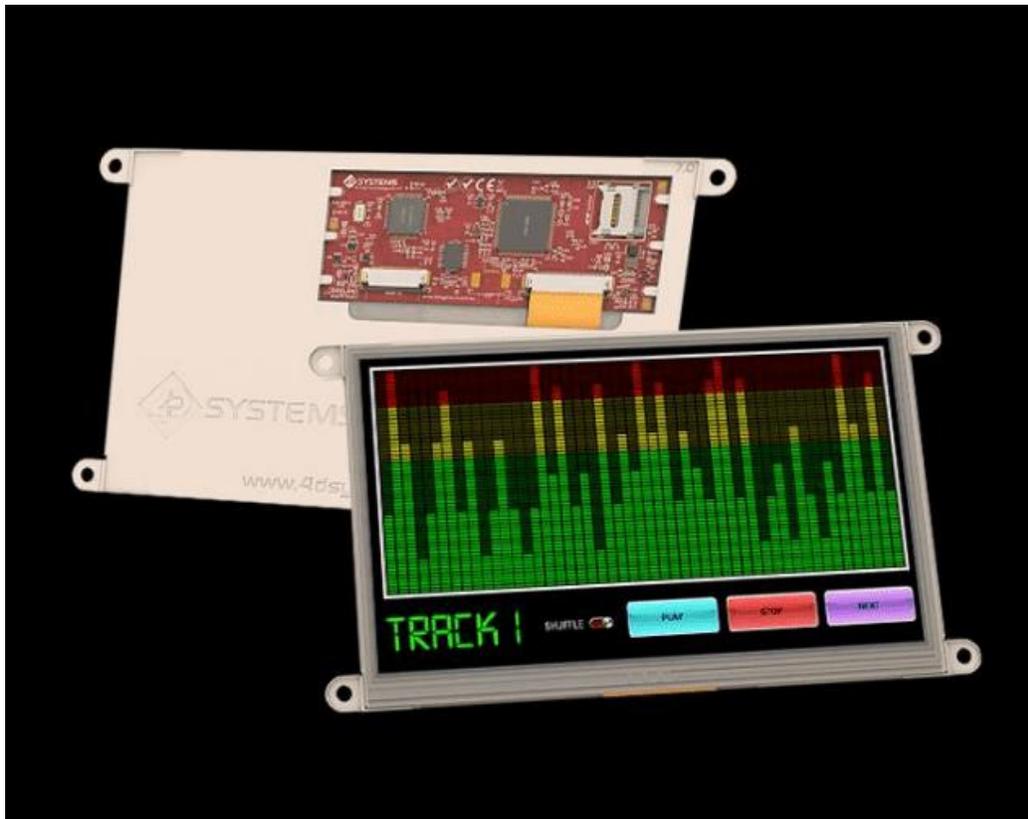


Figura 10 Pantalla GEN4-ULCD-70DT (Systems, 4D Systems turning technology into art, 2022)

Workshop4

Workshop4 es un IDE de software completo para Microsoft Windows que proporciona una plataforma de desarrollo de software integrada para toda la familia de procesadores y módulos 4D. El IDE combina el editor, el compilador, el vinculador y el descargador para desarrollar un código de aplicación 4DGL completo. Todo el código de la aplicación del usuario se desarrolla dentro del IDE Workshop4 (Systems, 4D Systems), mostrado en la figura 11.



Figura 11 Software Workshop4 (Systems, 4D Systems turning technology into art, 2022)

Genie Arduino

Esta es una biblioteca para Arduino IDE que permite la comunicación con un módulo de pantalla inteligente de 4D Systems, programado con el entorno ViSi-Genie, utilizando un Serial UART.

Protocolos de comunicación

Los protocolos de comunicación son de suma importancia en las empresas, este es un sistema de reglas que permiten que dos o más entidades de un sistema de comunicación se comuniquen entre ellas para transmitir información por medio de una variación de magnitud física.

Protocolo de comunicación CAN

El bus CAN (Controller Area Network) (L. Sánchez, 2016) es una representación de un bus serial diseñado por Bosh, en 1986. Para su uso en el desarrollo industrial y la aplicación en vehículos. Está basado en una topología bus para la transmisión de mensajes en entornos distribuidos.

CAN es el bus orientado a comunicación de sistemas para aplicaciones en la industria automotriz, como son los vehículos, tractocamiones, autobuses, trenes, y otros tipos de

vehículos; la razón fundamental fue la drástica reducción del complejo cableado de los sistemas de adquisición y sistemas de procesamiento presentes en vehículos modernos, que cuentan con múltiples microcomputadoras o microcontroladores basados en sistemas de instrumentación y control para el manejo del sistema de combustión, control de la suspensión, ABS, entre otros. El bus CAN ha ganado una gran aceptación en las redes de área local, no solo por en la industria automotriz, también por muchos otros usuarios en otros sectores de la industria. Hoy en día, existe una larga lista de productores de equipos para el bus CAN, entre ellos se encuentran: Alcatel, Motorola, National Semiconductor, Philips, Siemens, Texas Instrumentes, etc.

La red CAN puede ser usada en comunicación de sistemas embebidos para microcontroladores, como un sistema de comunicación abierta para dispositivos inteligentes.

Las principales características del protocolo CAN son las siguientes:

- Es una red serial de comunicación para soluciones embebidas
- Necesita solo dos cables CAN_H (Alto) y CAN_L (bajo).
- Opera en tasas de hasta 1 Megabit por segundo
- Soporta máximo 8 bytes por mensaje de trama
- No soporta ID de nodos, solo ID de mensajes. Una aplicación puede soportar múltiples ID de mensajes.
- Soporta prioridad de mensajes. El ID menor es el ID de mayor prioridad.
- Soporta dos tamaños de ID, uno de 11 bits (estándar) y otro de 29 bits (extendido)
- No experimenta colisión de mensajes (esto ocurre en otras tecnologías de comunicación serial)

➤ No demanda requerimientos de cableado, con solo tener un par de cables trenzados es suficiente.

Protocolo de comunicación LIN

El bus de Red Local de Interconexión (LIN) fue desarrollado para crear un estándar para comunicación multiplexada de bajo costo en redes automotrices. A pesar de que el CAN cubre la necesidad para alto ancho de banda, redes de manejo de error avanzado, los costos de hardware y software por la implementación de CAN se han vuelto costosos para dispositivos de menor rendimiento como controladores de potencia de ventanas y asientos. LIN proporciona comunicación rentable en aplicaciones donde el ancho de banda y la versatilidad de CAN no son requeridos. Puede implementar LIN prácticamente a un menor precio usando el transmisor/receptor estándar serial universal asincrónico (UART) embebido en la mayoría de los microcontroladores modernos de bajo costo, de 8 bits.

Las redes automotrices modernas usan una combinación de LIN para aplicaciones de bajo costo, principalmente en electrónicos, CAN para comunicación de tren de potencia y carrocería; y el bus FlexRay para comunicaciones de datos sincronizados de alta velocidad en sistemas avanzados, como suspensión activa.

El bus LIN es un bus con un solo dispositivo maestro y uno o más dispositivos esclavos. El dispositivo maestro contiene una tarea de maestro y una tarea de esclavo; cada dispositivo esclavo tiene solamente una tarea de esclavo. La comunicación a través del bus LIN está controlada completamente por la tarea de maestro en su dispositivo. La unidad básica de transferencia en el bus LIN es el marco, el cual está dividido en un encabezado y una respuesta.

El encabezado siempre es transmitido por el nodo maestro y consiste de tres diferentes campos:

- La interrupción, la sincronización (symc) y el identificador (ID). La respuesta es transmitida por una tarea de esclavo y puede residir ya sea en el nodo maestro o un nodo esclavo; consiste en una carga útil de datos y una suma de verificación.

Normalmente, la tarea de maestro consulta cada tarea de esclavo en un ciclo al transmitir un encabezado que consiste de una secuencia de interrupción sincronización-ID. Antes de comenzar el LIN, cada tarea esclava es configurada para publicar datos al bus o suscribir a datos en respuesta a cada ID de encabezado recibido. El intercambio de datos entre los diferentes sistemas de LIN en un vehículo se realiza respectivamente por medio de una unidad de control a través del CAN de datos.

El sistema permite el intercambio de datos entre una unidad de control LIN maestra, y hasta 16 unidades de control LIN esclavas.

Unidad maestra: es la unidad de control que van conectada al CAN es la que realiza las funciones de maestra en el LIN-Bus.

Unidad esclava: En un sistema de bus de datos LIN, la función de esclava la pueden realizar tanto una unidad de control como diferentes sensores o actuadores; por ejemplo, el sensor volumétrico de la alarma antirrobo. Los sensores llevan integrada una parte electrónica que analiza los valores medidos por el propio sensor. La transmisión de estos valores se realiza entonces a través del LIN-Bus en forma de señales digitalizadas. Varias unidades esclavas pueden ser conectadas a una sola terminal de la unidad de control maestra del LIN-Bus.

Convertor analógico digital

El objetivo básico de un convertor analógico digital (ADC) (Huircán, s.f.) es transformar una señal eléctrica análoga en un número digital equivalente, mostrado a detalle en la figura 12.

Según el tipo de componente y su aplicación existen distintos parámetros que lo caracterizan, éstos pueden ser: la velocidad de conversión, la resolución, los rangos de entrada, etc. Por ejemplo, una mayor cantidad de bit, implica mayor precisión, pero también

mayor complejidad. Un incremento en un solo bit permite disponer del doble de precisión (mayor resolución), pero hace más difícil el diseño del circuito, además, la conversión podría volverse más lenta.

El diagrama de bloques de la siguiente Figura muestra la secuencia desde que la variable física entra al sistema hasta que es transformada a señal digital (código binario). Para dicha señal ingrese al convertidor análogo - digital, ésta debe ser muestreada, es decir, se toman valores discretos en instantes de tiempo de la señal análoga, lo que recibe el nombre de "sampling".

Matemáticamente es el equivalente a multiplicar la señal análoga por una secuencia de impulsos de periodo constante. Como resultado se obtiene un tren de impulsos con amplitudes limitadas por la envolvente de la señal analógica.

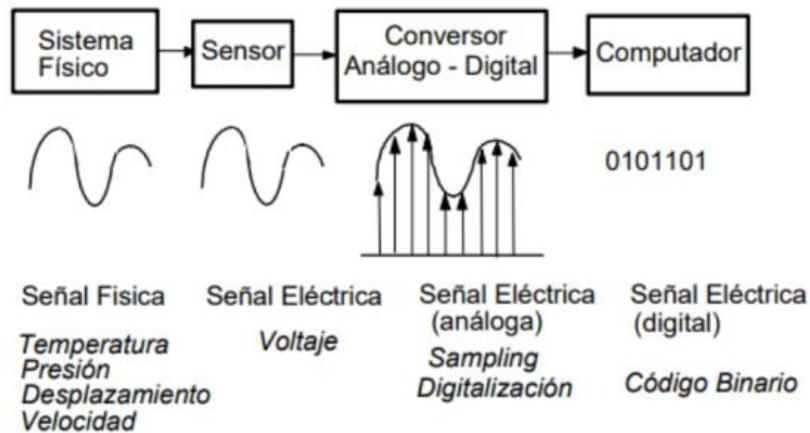


Figura 12 . Conversión análogo-digital (Huircán, s.f.).

Para garantizar la toma de muestra y la conversión de forma correcta se debe considerar la velocidad de muestreo, para lo cual el Teorema de Nyquist, establece que la frecuencia de muestreo "fs", debe ser como mínimo el doble que el ancho de banda de la señal muestreada.

Bancos de prueba

Un banco de prueba o fixtura es un dispositivo utilizado en la prueba de productos. Las fixturas brindan una forma de comprobación rigurosa, repetible y confiable. El propósito principal de las fixturas es identificar fallas de tipo eléctricas o de comunicación en los productos y poder atenderlas.

Un banco de pruebas se usa cuando un producto se prueba aparte del programa al que luego será agregado, para que el producto se comporte como si ya formara parte del programa más grande (el automóvil).

El proyecto de fixturas tiene algunos años que se empezó a implementar dentro de

Las empresas, por lo que ya existen una gran variedad de fixturas que han sido utilizadas a lo largo de los años para probar diferentes tipos de piezas.

Ejemplos de piezas para las cuales pueden ser diseñados estos bancos de pruebas automotrices.

“Assist” de “KIA”

Conociendo la pieza



Figura 13 . Logo de la empresa KIA (KIA, 2021).

A esta pieza de KIA, a la cual se le denomina “Assist” o “Pasajero” ya que es una pieza que va en el lado del copiloto. Lo primero que se hizo fue conocer la pieza para identificar los

accionamientos que tiene, los cuales se pueden apreciar en la figura 14. Esta pieza sube y baja la ventana del copiloto, así como también sirve para poner y quitar los seguros de las puertas.



Figura 14 Pieza "Assist" de KIA..

La fixtura para la pieza "Assist" fue diseñada para dos tipos de piezas, una a la cual se le llama de baja corriente (LC) y una de alta corriente (HC). La pieza LC trabaja con "ADCs" en un rango de 0-5V (Volts) y la HC trabaja mandando bajos y altos de 0 y 13.5 Volts. Gracias a este diseño las piezas LC pueden agregar una función extra la cual es "Auto-Up" o "Auto-Down" para subir o bajar automáticamente las ventanas. En la Figura siguiente se puede ver el nido de la fixtura, el cual tiene dos espacios para colocar las piezas ya mencionadas, mostrada en la figura 15.

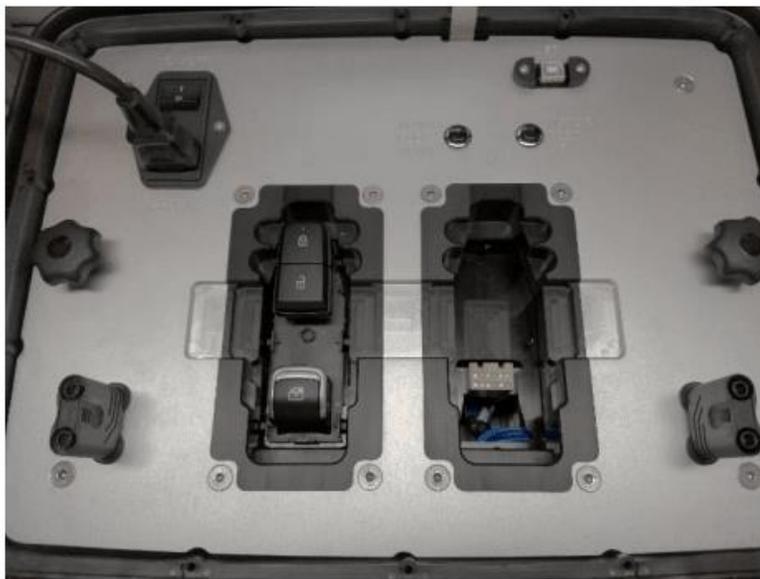


Figura 15 Nido de la fixtura "Assist".

Esta fixtura trabaja con una tarjeta "ShieldBuddy TC275" la cual se puede programar en la IDE de Arduino descargando un complemento para la misma y su programación es idéntica a la de un Arduino.

Esta fixtura trabaja con una tarjeta "ShieldBuddy TC275" la cual se puede programar en la IDE de Arduino descargando un complemento para la misma y su programación es idéntica a la de un Arduino.

Interfaz Gráfica

Una vez que se conocieron los accionamientos que tiene la pieza, se procedió a realizar la interfaz gráfica para lo cual primero se realizaron las imágenes de los accionamientos, como se muestra en la figura 17.

Para los seguros se tiene un candado en blanco el cual muestra que ese botón todavía no ha sido presionado, el color verde muestra que el botón está siendo presionado y finalmente el color azul muestra que la tecla ya se presionó, como se muestra en la figura 16.



Figura 16 Bloqueo y desbloqueo de puertas.

Para el botón de subir-bajar ventana se tiene una flecha en blanco la cual muestra que el botón no ha sido presionado, con el verde se tienen dos flechas como se aprecia en la Figura de abajo, una flecha en verde muestra que se ha presionado el botón para bajar o subir ventana según corresponda, y una doble flecha en verde muestra que se oprimió la función “Auto Up-Down” de la pieza.

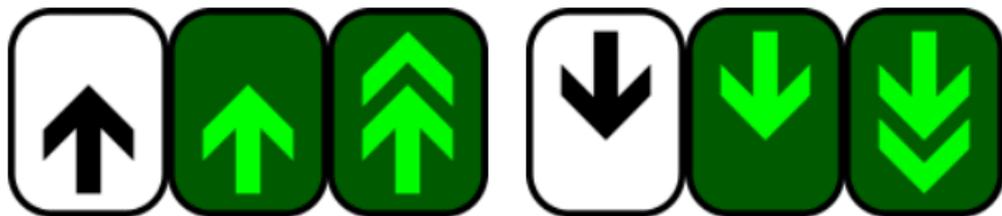


Figura 17 Subir y bajar ventanas.

La imagen blanca de la figura 18 se encontrará al lado de las activaciones de las ventanas, y cuando se haya realizado la secuencia de bajar o subir la ventana, esta imagen se pondrá en color azul para mostrar que ya se realizó la secuencia.

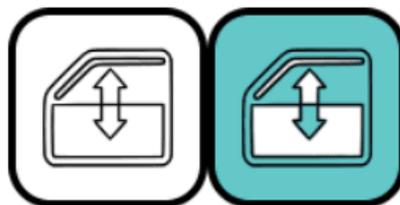


Figura 18 Secuencia de subir-bajar ventana.

Ya que se tuvieron las imágenes de los accionamientos de la pieza se procedió a modificar y agregar nuevas formas (Nuevas pantallas) a la interfaz que ya venía cargada en la fixtura, para esto se utilizó el software Workshop4, el cual nos permite agregar formas, imágenes, botones, contadores, etc. Para lograr tener una interfaz amistosa con el usuario. La Figura de abajo muestra la interfaz de inicio que tiene cargada la fixtura, esta interfaz la mandó el proveedor, únicamente se le agregó el botón de “START TEST” para que al presionar este botón la pantalla cambie a la forma de los accionamientos de la pieza, como es mostrado en la figura 19.



Figura 19 Interfaz de inicio.

Cuando se oprime el botón “START TEST” la pantalla los manda a la forma de la figura 20, la cuál es la interfaz para probar la pieza, esta interfaz cuenta con lo siguiente:

- Imágenes de la secuencia de los botones.
- Botón “Return” para regresar a la pantalla anterior.
- Botones “PIEZA OK” y “Pieza NOK” para llevar un conteo de cuantas piezas buenas y malas han probado.
- Dos contadores para las piezas buenas (Piezas OK) y las piezas malas (Piezas NOK).
- Una pantalla negra, la cual muestra un mensaje en caso de que se presiona el botón verde de “PIEZA OK” pero no se hayan presionado todos los botones aún.
- Logotipo de KOSTAL y de KIA.
- Dos leds “FEED DOWN” y “FEED UP” los cuales se activan con dos botones físicos que tiene la fixtura, para comprobar que la fixtura funciona bien.
- Imagen representativa de la pieza.

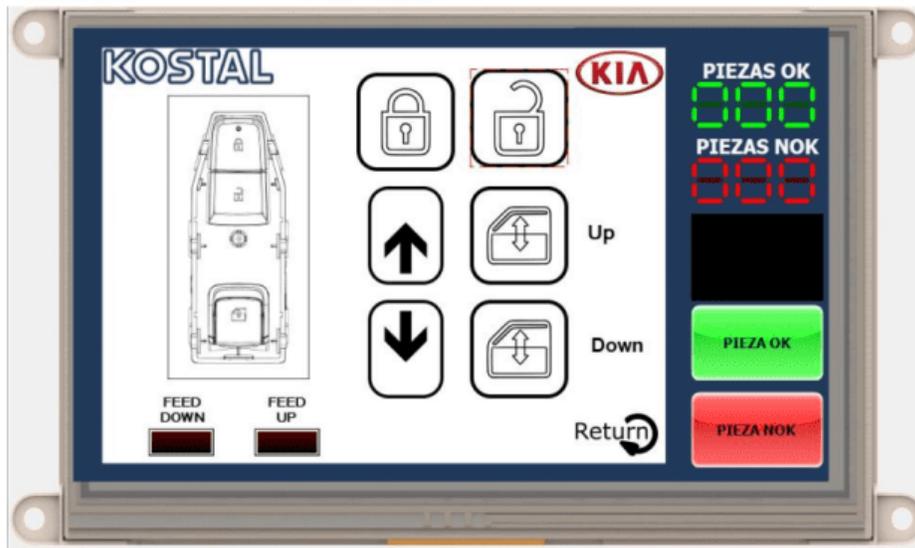


Figura 20 . Interfaz para probar la pieza.

Para lograr que cambien de color las imágenes de los botones en la pantalla, dentro del mismo programa de Workshop4, a cada imagen blanca se le agregan las imágenes que se quieran usar, y el programa les irá asignando un índice, estos índices son muy importantes ya que se utilizarán en el código al momento de programar, para hacer referencia a la imagen que se quiere mostrar en la pantalla. En la figura 21 y 22 se ejemplifica como se cargan las imágenes y como el programa les asigna un índice, para la figura 20 que es el candado blanco, se le asignó un “0” mientras que para el candado verde de la figura 22 se le asignó un “1”.

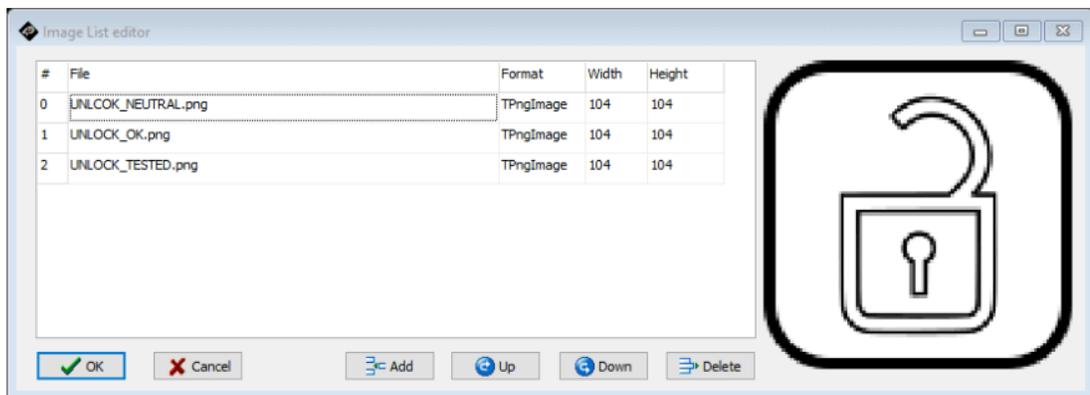


Figura 21 Añadiendo candado blanco a las imágenes del programa.

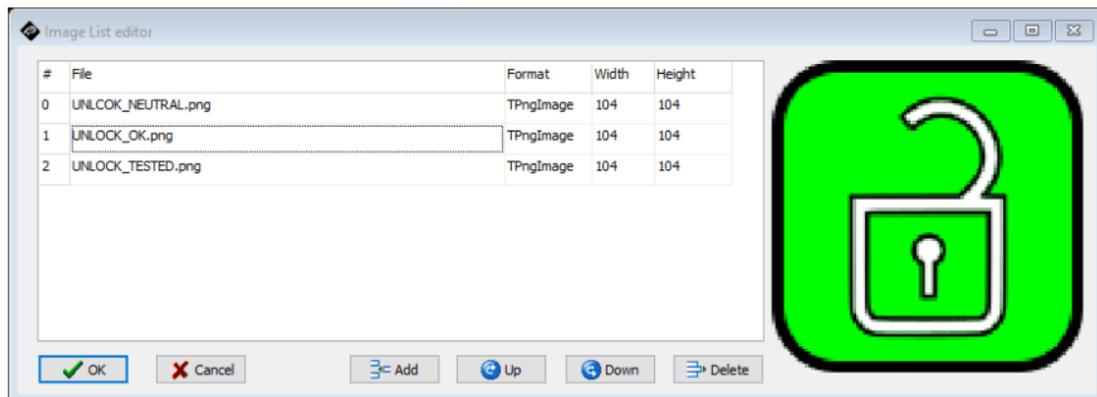


Figura 22 Añadiendo candado verde a las imágenes del programa.

Código

Para programar los accionamientos de la pieza, fue necesario primero conocer que lecturas de voltaje se tenían cada que se oprimía algún botón de la pieza, esto se logró conocer gracias a una función que incluyó el proveedor, en la figura 17 se aprecia que hay un botón que se llama “ADCs Debug Form”, al oprimir ese botón, la pantalla nos envía a la forma que se aprecia en la figura 23.



Figura 23 Lecturas de “ADCs” y “HC”.

Esta pantalla nos sirve para ver como varía cada ADC de la placa “ShieldBuddy TC275”. Lo que se hizo fue colocar las piezas en la fixtura y ver en la pantalla cuales eran los “ADCs” y “HC” que cambiaban de valor al presionar los botones de la pieza y se fue registrando el valor que arrojaban en la pantalla, para después programar los rangos con esos valores, los valores optenidos son los de la tabla 2.

Tabla 2. Valores de voltaje obtenidos de los accionamientos de las piezas “LC” y “HC” de “Assist”.

Tabla 2 Valores de accionamiento de las piezas LC y HC

		LC (V)	HC (V)	
Lock	ADC 2	0.8	0.8	ADC 0
Unlock	ADC 3	0.8	0.8	ADC 1
Window Down	ADC 4	3.85	-13.21	HC 0
Auto Window Down	ADC 4	2.78	NA	NA
Window Up	ADC 4	1.76	13.22	HC 0
Auto Window Up	ADC 4	0.8	NA	NA

En la tabla 2 se muestran los valores que arrojó cada accionamiento tanto de la pieza “LC” como la pieza “HC”, con estos valores se procedió a programar el código final de la fixtura de “Assist”. A continuación, se mostrarán fragmentos del código.

Primero se tiene la declaración de las variables que hacen referencia a las imágenes, botones y leds que se pusieron en la interfaz de la figura 18, para probar la pieza, y cada variable tiene un número asignado el cual debe corresponder con el número que se le asignó en el software de Worshop4.

```

//*****
//Form 6 - Forma de Pieza HC
//*****
const int F_Pieza           = 6;      //Esta forma es la número 6
const int UB_Return        = 0;      //Se declaran los UserButton, los WinButton, los
UserImage y los UserLed que hay en esa forma
const int WB_OK            = 13;
const int WB_NOK          = 12;
const int UI_FlechaUp      = 1;
const int UI_FlechaDown    = 2;
const int UI_VentanaDown   = 3;
const int UI_VentanaUp     = 4;
const int UI_Lock          = 13;
const int UI_Unlock        = 5;
const int UL_FeedUp        = 0;
const int UL_FeedDown      = 1;
const int S_General        = 0;      //Screen String de alta

```

También se declaran las variables que representan los cambios de estados, para cambiar el color de los candados y las flechas, estas variables tienen asignado un número que debe corresponder con el índice que se les asignó en Worshop4.

```

//*****
//Estados para las flechas
//*****
const int UIR_FlechaUp0    = 0;      //Se declaran todos los estados de las fechas
(Blanca, verde, verde doble flecha, azul), para arriba y para abajo.
const int UIR_FlechaUp1    = 1;
const int UIR_FlechaUp2    = 2;
const int UIR_FlechaUp3    = 3;
const int UIR_FlechaDown0  = 0;
const int UIR_FlechaDown1  = 1;
const int UIR_FlechaDown2  = 2;
const int UIR_FlechaDown3  = 3;

```

Se configuran los estados y cambios de estados para los seguros.

```
//*****  
//Estados para LOCK  
//*****  
const int UIR_Lock0          = 0;    //Se declaran todos los estados del botón LOCK  
(Blanco, verde y azul).  
const int UIR_Lock1          = 1;  
const int UIR_Lock2          = 2;
```

Después se declaran las variables que contienen los valores de voltaje de la Tabla 3.

```
//*****  
//Volt de Act en ADC4 LC  
//*****  
const float V_VoltajeVentanaDownLC      = 3.85;    //Se ponen los voltajes leídos en el  
debug (Bajar ventana, bajar ventana automatico, subir ventana, subir ventana automatico)para  
baja (ADC2)  
const float V_AutoVoltajeVentanaDownLC  = 2.78;  
const float V_VoltajeVentanaUpLC        = 1.76;  
const float V_AutoVoltajeVentanaUpLC     = 0.8;  
  
bool VoltajeVentanaDownL4                = false;  //Variables booleanas que se usarán para  
saber cuando ya se oprimió un botón  
bool AutoVoltajeVentanaDownL4            = false;  
bool VoltajeVentanaUpL4                  = false;  
bool AutoVoltajeVentanaUpL4              = false;  
  
//*****  
//Volt de Act en HCO  
//*****  
const float V_VoltajeVentanaDownHC      = -13.21;  //Se ponen los voltajes leídos en el  
debug (Bajar ventana, subir ventana),para alta (HCO y HC1)  
const float V_VoltajeVentanaUpHC        = 13.22;  
  
bool VoltajeVentanaDownH0                = false;  //Variables booleanas que se usarán para  
saber cuando ya se oprimió un botón  
bool VoltajeVentanaUpH0                  = false;
```

Se declaran las funciones que servirán para limpiar variables y volver a poner la interfaz en su forma inicial (con las imágenes de los accionamientos en blanco).

```
void clearImages()
{
  genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_FlechaUp, UIR_FlechaUp0);
  genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_FlechaDown, UIR_FlechaDown0);
  genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_VentanaDown, UIR_VentanaDown0);
  genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_VentanaUp, UIR_VentanaUp0);
  genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_Lock, UIR_Lock0);
  genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_Unlock, UIR_Unlock0);

  genie.WriteString(S_General, " ");
}
```

Después se declara la lógica del programa para la pieza de “HC” y la de “LC”, se ponen todas las condiciones para que con base a la lectura de voltaje que se tenga, el programa se comunique con la pantalla y cambie la imagen que corresponda, para esto se utiliza la librería “GenieArduino” la cual es una librería que nos facilita el proveedor de las pantallas para poder comunicar el microcontrolador con sus pantallas.

```
//Funciones de voltajes LC
//*****
void Window_ActLC() //Función que hará las comparaciones del botón ventana de baja
{
    if(Volts_TC275[4]<4.5){ //Condición para saber que está probando la pieza de baja con
    calefacción
        if(Volts_TC275[4] > V_VoltajeVentanaDownLC - tolerance && Volts_TC275[4] <
V_VoltajeVentanaDownLC + tolerance)
        {
            genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_FlechaDown, UIR_FlechaDown1);
            VoltajeVentanaDownL4 = true;
        }
        else if(Volts_TC275[4] > V_AutoVoltajeVentanaDownLC - tolerance && Volts_TC275[4] <
V_AutoVoltajeVentanaDownLC + tolerance)
        {
            genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_FlechaDown, UIR_FlechaDown2);
            AutoVoltajeVentanaDownL4 = true;
        }
        else if(Volts_TC275[4] > V_VoltajeVentanaUpLC - tolerance && Volts_TC275[4] <
V_VoltajeVentanaUpLC + tolerance)
        {
            genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_FlechaUp, UIR_FlechaUp1);
            VoltajeVentanaUpL4 = true;
        }
        else if(Volts_TC275[4] > V_AutoVoltajeVentanaUpLC - tolerance && Volts_TC275[4] <
V_AutoVoltajeVentanaUpLC + tolerance)
        {
            genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_FlechaUp, UIR_FlechaUp2);
            AutoVoltajeVentanaUpL4 = true;
        }
    }

    else if(Volts_TC275[4] >= 4.5 && Volts_HC[0] < 0.5 && Volts_HC[0] > -0.5) //Condición que
indica que ningún botón de baja se está probando
    {
        genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_FlechaUp, UIR_FlechaUp0);
        genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_FlechaDown, UIR_FlechaDown0);
        genie.WriteObject(GENIE_OBJ_USER_LED, 0, 0); //Se apagan los dos leds visuales
        genie.WriteObject(GENIE_OBJ_USER_LED, 1, 0);
    }

    if(VoltajeVentanaDownL4 && AutoVoltajeVentanaDownL4) //Si ya se oprimió las flechas, se
ponen las imagenes de ventana en azul.
    {
        genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_VentanaDown, UIR_VentanaDown2);
    }
    if(VoltajeVentanaUpL4 && AutoVoltajeVentanaUpL4)
    {
        genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_VentanaUp, UIR_VentanaUp2);
    }
}
}
```

Después se declara la función “myGenieEventHandler()”, la cuál es una función que está esperando a que ocurra algún evento en la pantalla, para realizar alguna acción, en este caso se agregaron las funciones del botón “START TEST”, el botón “Return”, “PIEZA OK” y “PIEZA NOK” y la lógica para que el botón “PIEZA OK” funcione correctamente, y solo contabilice la pieza como buena, cuando se haya completado toda la secuencia de la pantalla.

```
/**.....  
//Event Handler del LCD, aqui se leen botones y otro objetos  
//.....  
void myGenieEventHandler(void)  
{  
    //SerialASC.println(millis());  
}
```

Librerías y eventos mandados llamar.

```
genieFrame Event;  
genie.DequeueEvent(&Event);  
  
if (genie.EventIs(&Event, GENIE_REPORT_EVENT, GENIE_OBJ_WINBUTTON, WB_starttest)) //Si se  
oprimió el botón Start test entra aquí  
{  
    mode = F_Pieza;          //Mode toma el valor de F_Inicio para poder usarlo en el case  
    ADC_ON_OFF = 1;  
    genie.WriteObject(GENIE_OBJ_FORM, F_Pieza, true);    //Pone la forma F_Inicio en true, para  
que la pantalla cambie de forma  
}  
  
if (genie.EventIs(&Event, GENIE_REPORT_EVENT, GENIE_OBJ_WINBUTTON, WB_OK))//Si se oprimió  
el botón WB_OKHC que esta en la forma de alta, entra aquí  
{  
    if(VoltajeVentanaUpL4 == true && VoltajeVentanaDownL4 == true && LOCKL == true && UNLOCKL  
== true && AutoVoltajeVentanaUpL4 == true && AutoVoltajeVentanaDownL4 == true){    //Si ya  
se oprimieron los 3 botones de la pieza de calefacción entra aquí  
    Counter_OK++;          //Aumenta la variable contador  
    genie.WriteObject(GENIE_OBJ_LED_DIGITS, LD_OK, Counter_OK);    //Y le asigna el  
número de la variable contador al LedDigital (Pantalla de 7 segmentos)  
    clearImages();        //Pone todas las imagenes en blanco de nuevo  
}  
  
    else if(VoltajeVentanaUpH0 == true && VoltajeVentanaDownH0 == true && LOCKH == true &&  
UNLOCKH == true){ //Si ya se oprimieron los 3 botones de la pieza de calefacción entra aquí  
    Counter_OK++;          //Aumenta la variable contador  
    genie.WriteObject(GENIE_OBJ_LED_DIGITS, LD_OK, Counter_OK); //Y le asigna el número de la  
variable contador al LedDigital (Pantalla de 7 segmentos)  
    clearImages();        //Pone todas las imagenes en blanco de nuevo  
}  
  
    else{                  //Si oprime el botón WB_OKHC y no se completó la secuencia manda el  
mensaje de abajo en el Screen String  
        genie.WriteStr(S_General, "Secuencia no \ncompletada");  
    }  
}
```

Finalmente, dentro del “loop” se agrega un “switch” el cual sirve para saber en qué vista de la pantalla se encuentra actualmente el usuario y con base en eso poder usar las funciones de esa vista, en este caso dentro de la vista para probar la pieza se mandan a llamar todas las funciones para checar las activaciones de la pieza.

```
void loop()
{
  genie.DoEvents();
  switch (mode)
  {
    case F_Pieza: //Se agrega un case con la forma de la pieza de alta y baja
      Read_ADC_TC275(); //Se llaman a las funciones de lectura, para poder leer
      los voltajes de baja y de alta
      Read_ADC_ADS8688();
      Window_ActHC(); //Se llaman a las funciones que leerán los voltajes de
      alta (ventana, lock y unlock)
      Lock_ActHC();
      Unlock_ActHC();
      Window_ActLC();
      Lock_ActLC();
      Unlock_ActLC();
      break;
  }
}
```

Con esto se logró finalizar el código y las formas (vistas de la pantalla) de forma exitosa, para que la fixtura de “Assist” fuera utilizada por cualquier operador de la planta. En la sección de “Resultados” se mostrará la fixtura trabajando.

“Rear” de “KIA”

Conociendo la pieza

Esta fue la segunda pieza de KIA con la que trabajamos, a la cual se le denomina “Rear”, esta es una pieza que se encuentra en las puertas traseras del automóvil. Al igual que con “Assist” lo primero que se hizo fue conocer la pieza para identificar los accionamientos que tiene, los cuales se pueden apreciar en la figura 24. Esta pieza sube y baja la ventana de la puerta trasera, y algunas versiones tienen otro botón para la calefacción del asiento.



Figura 24 Pieza "Rear" de KIA.

La fixtura para la pieza "Rear" fue diseñada para cuatro versiones de esta pieza, una "LC" con botón calefactor, otra "LC" sin botón calefactor y dos más HC de igual forma una pieza con botón calefactor y una sin botón calefactor. En la figura 25, se puede ver el nido de la fixtura, el cual tiene dos espacios para colocar las piezas ya mencionadas.

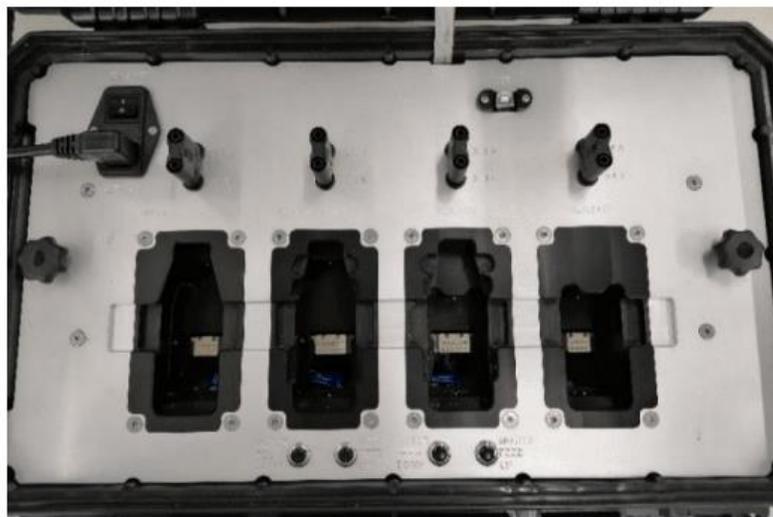


Figura 25 Fixtura que trabaja con una tarjeta "ShieldBuddy TC275".

Interfaz Gráfica

Ya que se conocieron los accionamientos que tiene la pieza, se procedió a realizar la interfaz gráfica para lo cual primero se realizaron las imágenes de los accionamientos. Debido a que el accionamiento para subir y bajar la ventana es el mismo que el de la pieza “Assist” se utilizaron las mismas imágenes de las figuras mostadas para la primera fixtura, y solamente se crearon las imágenes para el botón calefactor.

En la figura 26 se muestran las imágenes creadas para el botón calefactor, donde la imagen en blanco muestra que ese botón todavía no ha sido presionado, el color verde muestra que el botón está siendo presionado y el color azul muestra que la tecla ya se presionó.



Figura 26 Botón calefactor.

Ya que se tuvieran las imágenes de los accionamientos de la pieza se procedió a modificar y agregar nuevas formas a la interfaz que ya venía cargada en la fixtura. Para lograr tener una interfaz amistosa con el usuario.

La fixtura “Rear” al igual que la fixtura “Assist” venían cargadas con la misma interfaz, de las figuras de la primera fixtura, muestra la interfaz de inicio que tiene cargada la fixtura, esta interfaz la mandó el proveedor, únicamente se le agregó el botón de “START TEST”.

Cuando se toca el botón “START TEST” la pantalla los manda a la forma de la figura 25, la cuál es la interfaz para seleccionar el tipo de pieza que se quiere probar (“LC” o “HC”).

Esta forma contiene:

- Dos Botones con la imagen del tipo de pieza.
- Botón “Return” para regresar a la pantalla anterior.
- Boton “Debug” para mandar a la lectura de los “ADCs” la cual es idéntica la que podemos observar en la figura 23.
- Logotipo de “KOSTAL” y de “KIA” y un “string” con la leyenda “INICIAR PRUEBA”.



Figura 27 Interfaz para seleccionar el tipo de pieza (“LC” o “HC”).

Si seleccionan “Rear LC” en la figura 27 la pantalla los mandará a la forma de la figura 28.

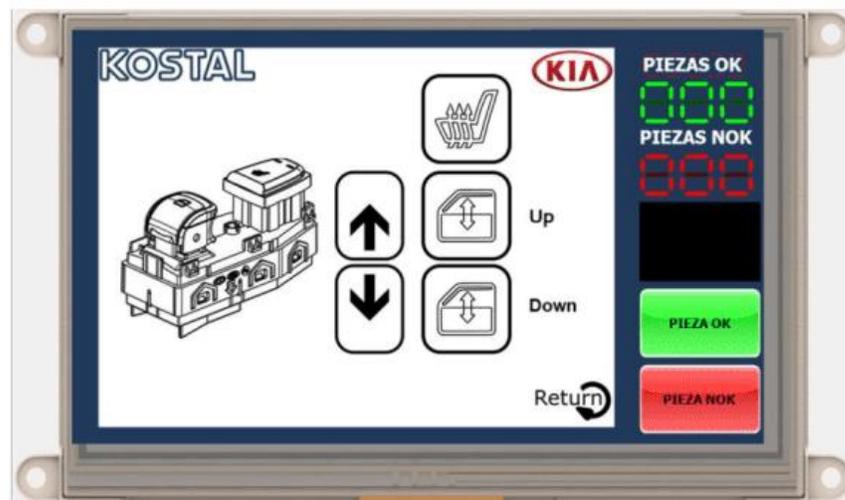


Figura 28 Interfaz para probar la pieza de “LC”.

Y si seleccionan “REAR HC” la pantalla los mandará a la forma de la figura 27. Las dos formas son muy parecidas, únicamente cambian los leds de “FEED DOWN”, “FEED UP” y la imagen que ilustra la pieza, lo cual ayuda para que el operador que utilice la fixtura, seleccione correctamente el tipo de pieza que quiere probar.

La interfaz de la figura 28 y 29 cuenta con lo siguiente:

- Imágenes de la secuencia de los botones.
- Botón “Return” para regresar a la pantalla anterior.
- Botones “PIEZA OK” y “Pieza NOK” para llevar un conteo de cuantas piezas buenas y malas han probado.
- Dos contadores para las piezas buenas (Piezas OK) y las piezas malas (Piezas NOK).
- Una pantalla negra, la cual muestra un mensaje en caso de que se presiona el botón

verde de “PIEZA OK” pero no se hayan presionado todos los botones aún.

- Logotipo de “KOSTAL” y de “KIA”.
- Dos leds “FEED DOWN” y “FEED UP” los cuales se activan con dos botones físicos que tiene la fixtura, para comprobar que la fixtura funciona bien.
- Imagen representativa de la pieza.

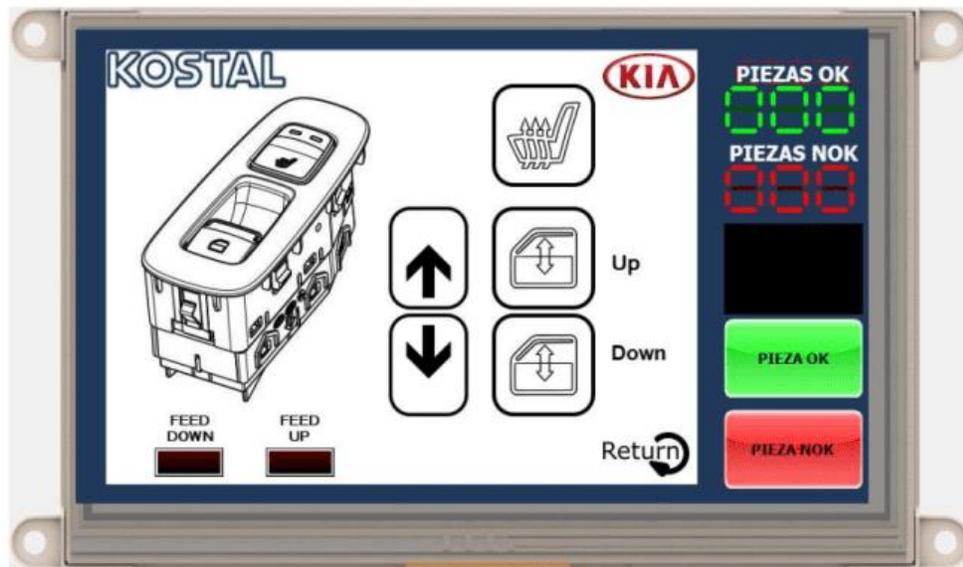


Figura 29 Interfaz para probar la pieza de “HC”.

Código

Al igual que con la fixtura “Assist” se utilizó el “ADCs Debug Form” que se puede observar en la figura 23 para obtener las lecturas de voltaje que daba cada accionamiento de la pieza y también conocer que “ADCs” estaba utilizando la pieza. Toda esta información se fue registrando en una tabla para después programar los accionamientos con esos valores.

Tabla 3. Valores de voltaje obtenidos de los accionamientos de las piezas “LC” y “HC” de “Rear”.

Tabla 3 Valores de accionamiento de las piezas LC yHC

		LC (V)	HC (V)	
Heating	ADC 3	0.77	0.74	ADC 0
Window Down	ADC 2	3.85	-13.21	HC 0
Auto Window Down	ADC 2	2.76	NA	NA
Window Up	ADC 2	1.74	13.21	HC1
Auto Window Up	ADC 2	0.77	NA	NA

En la tabla 3 se muestran los valores que arrojó cada accionamiento tanto de la pieza “LC” como la pieza “HC”, con estos valores se procedió a programar el código final de la fixtura de “Rear”. En la siguiente parte se muestran fragmentos del código.

Primero se tiene la declaración de las variables que hacen referencia a las imágenes, botones y leds que se pusieron en la interfaz de la figura 27, 28 y 29, cada variable tiene un número asignado el cual corresponde con el número que se le asignó en Worshop4.

```

//*****
//Form 7 - Forma de Pieza LC
//*****
const int F_PiezaLC           = 7;      //Esta forma es la número 7
const int UB_ReturnLC         = 1;      //Se declaran los UserButton, los WinButton y los
UserImage que hay en esa forma
const int WB_OKLC             = 17;
const int WB_NOKLC           = 16;
const int UI_PiezaLC          = 9;
const int UI_FlechaUpLC       = 5;
const int UI_FlechaDownLC     = 6;
const int UI_VentanaDownLC    = 8;
const int UI_VentanaUpLC      = 7;
const int UI_CalorLC          = 11;
const int S_LC                = 15;    //Screen String de baja

```

También se declaran las variables que representan los cambios de estados, para cambiar el color de los botones de calefacción y las flechas, estas variables tienen asignado un

número que debe corresponder con el índice que se les asignó en Workshop4.

```
//*****  
//Estados para las flechas  
//*****  
const int UIR_FlechaUp0      = 0;    //Se declaran todos los estados de las flechas  
//(Blanca, verde, verde doble flecha, azul), para arriba y para abajo.  
const int UIR_FlechaUp1      = 1;  
const int UIR_FlechaUp2      = 2;  
const int UIR_FlechaUp3      = 3;  
const int UIR_FlechaDown0    = 0;  
const int UIR_FlechaDown1    = 1;  
const int UIR_FlechaDown2    = 2;  
const int UIR_FlechaDown3    = 3;
```

Después se declaran las variables que contienen los valores de voltaje de la Tabla 4.

```
//*****  
//Volt de Act en ADC2 LC  
//*****  
const float V_VoltajeVentanaDownLC      = 3.85;    //Se ponen los voltajes leídos en el  
//debug para baja (ADC2)  
const float V_AutoVoltajeVentanaDownLC  = 2.76;  
const float V_VoltajeVentanaUpLC        = 1.74;  
const float V_AutoVoltajeVentanaUpLC     = 0.77;  
  
//*****  
//Volt de Act en HCC  
//*****  
const float V_VoltajeVentanaDownHC      = -13.21;   //Se ponen los voltajes leídos en el  
//debug (Bajar ventana, subir ventana), para alta (HC0 y HC1)  
const float V_VoltajeVentanaUpHC        = 13.21;
```

Se declaran las funciones que servirán para limpiar variables y volver a poner la interfaz en su forma inicial (con las imágenes de los accionamientos en blanco).

```
void clearImages()          //Función que limpiará las formas de alta y baja (Poner imagenes en  
//blanco, poner variables en false y limpiar Screen string)  
{  
    genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_FlechaUpHC, UIR_FlechaUp0);  
    genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_FlechaDownHC, UIR_FlechaDown0);  
    genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_VentanaDownHC, UIR_VentanaDown0);  
    genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_VentanaUpHC, UIR_VentanaUp0);  
    genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_CalorHC, UIR_Calor0);  
  
    genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_FlechaUpLC, UIR_FlechaUp0);  
    genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_FlechaDownLC, UIR_FlechaDown0);  
    genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_VentanaDownLC, UIR_VentanaDown0);  
    genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_VentanaUpLC, UIR_VentanaUp0);  
    genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_CalorLC, UIR_Calor0);  
}
```

En la siguiente parte se hace la lógica del programa para la pieza de “HC” y la de “LC”, se ponen todas las condiciones para que con base a la lectura de voltaje que se tenga, el

programa se comunique con la pantalla y cambie la imagen que corresponda, haciendo uso de la librería “GenieArduino”.

```
//*****  
//Funciones de voltajes HC  
//*****  
void Window_ActHC() //Función que hará las comparaciones del botón ventana de alta  
{  
  
    //-----HCO  
    if(Volts_HC[0]>0.5 || Volts_HC[0]<-0.5){ //Condición para saber que está probando la  
        pieza de alta con calefacción  
        if(Volts_HC[0] > V_VoltajeVentanaDownHC - tolerance && Volts_HC[0] < V_VoltajeVentanaDownHC  
+ tolerance) //Si se cumplen las condiciones, entra a este if  
        {  
            genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_FlechaDownHC, UIR_FlechaDown1); //Asigna  
una flecha verde al espacio de la flecha hacia abajo  
            VoltajeVentanaDownH0 = true; //Convierte la variable en true para saber que  
este botón ya se oprimió  
            genie.WriteObject(GENIE_OBJ_USER_LED, 1, 1); //Activa led visual 1  
        }  
    }  
}
```

```
    }
    else if(Volts_HC[0] > V_VoltajeVentanaUpHC - tolerance && Volts_HC[0] <
V_VoltajeVentanaUpHC + tolerance)
    {
        genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_FlechaUpHC, UIR_FlechaUp1);
        VoltajeVentanaUpH0 = true;
        genie.WriteObject(GENIE_OBJ_USER_LED, 0, 1);        //Activa led visual 0
    }
}

//-----HC1
else if(Volts_HC[1]>0.5 || Volts_HC[1]<-0.5){                //Condición para saber que está
probando la pieza de alta sin calefacción
    if(Volts_HC[1] > V_VoltajeVentanaDownHC - tolerance && Volts_HC[1] < V_VoltajeVentanaDownHC
+ tolerance)
    {
        genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_FlechaDownHC, UIR_FlechaDown1);
        VoltajeVentanaDownH1 = true;
        genie.WriteObject(GENIE_OBJ_USER_LED, 1, 1);
    }
    else if(Volts_HC[1] > V_VoltajeVentanaUpHC - tolerance && Volts_HC[1] <
V_VoltajeVentanaUpHC + tolerance)
    {
        genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_FlechaUpHC, UIR_FlechaUp1);
        VoltajeVentanaUpH1 = true;
        genie.WriteObject(GENIE_OBJ_USER_LED, 0, 1);
    }
}

if(Volts_HC[1] < 0.5 && Volts_HC[0] < 0.5 && Volts_HC[1] > -0.5 && Volts_HC[0] > -0.5)
    //Condición que indica que ningún botón de alta se está probando
    {
        genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_FlechaUpHC, UIR_FlechaUp0); //Se ponen flechas
blancas en los espacios de flecha arriba y abajo
        genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_FlechaDownHC, UIR_FlechaDown0);
        genie.WriteObject(GENIE_OBJ_USER_LED, 0, 0);        //Se apagan los dos leds visuales
        genie.WriteObject(GENIE_OBJ_USER_LED, 1, 0);
    }

if(VoltajeVentanaDownH0==true || VoltajeVentanaDownH1==true)        //Si ya se oprimieron las
flechas, se ponen las imagenes de ventana en azul.
    {
        genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_VentanaDownHC, UIR_VentanaDown2);
    }
if(VoltajeVentanaUpH0==true || VoltajeVentanaUpH1==true )
    {
        genie.WriteObject(GENIE_OBJ_USERIMAGES, UI_VentanaUpHC, UIR_VentanaUp2);
    }
}
```

Después se declara la función “myGenieEventHandler()”, la cual está esperando a que ocurra algún evento en la pantalla, para realizar alguna acción, en este caso se agregaron los eventos de los botones “START TEST”, el botón “Return”, “PIEZA OK”, “PIEZA NOK” y los botones para seleccionar el tipo de pieza “REAR LC” y “REAR HC” con sus respectivos comandos para asignar la imagen o forma que corresponda, también se agregó la lógica para que los botones “PIEZA OK” funcionen correctamente, y solo contabilicen las piezas como buenas, cuando se haya completado toda la secuencia de la pantalla.

```

//*****
//Event Handler del LCD, aqui se leen botones y otro objetos
//*****

void myGenieEventHandler(void)
{
    genieFrame Event;
    genie.DequeueEvent (&Event);

    if (genie.EventIs(&Event, GENIE_REPORT_EVENT, GENIE_OBJ_WINBUTTON, WB_starttest)) //Si se
    oprimió el botón Start test entra aquí
    {
        mode = F_Inicio; //Mode toma el valor de F_Inicio para poder usarlo en el case
        genie.WriteObject(GENIE_OBJ_FORM, F_Inicio, true); //Pone la forma F_Inicio en
        true, para que la pantalla cambie de forma
    }
    if (genie.EventIs(&Event, GENIE_REPORT_EVENT, GENIE_OBJ_USERBUTTON, UB_PiezaHC)) //Si se
    oprimió el botón de la imagen de la pieza de alta, entra aquí
    {
        mode = F_PiezaHC; //Mode toma el valor de F_PiezaHC para poder usarlo en el case
        ADC_ON_OFF = 1; //Se pone la variable en 1, para poder usar las funciones de
        lectura de los ADC's
        genie.WriteObject(GENIE_OBJ_FORM, F_PiezaHC, true); //Pone la forma F_PiezaHC en true,
        para que la pantalla cambie a esa forma
    }

    if (genie.EventIs(&Event, GENIE_REPORT_EVENT, GENIE_OBJ_WINBUTTON, WB_OKHC)) //Si se
    oprimió el botón WB_OKHC que esta en la forma de alta, entra aquí
    {
        if(VoltajeVentanaUpH0 == true && VoltajeVentanaDownH0 == true && CALORH == true){ //Si ya
        se oprimieron los 3 botones de la pieza de calefacción entra aquí
            Counter_OK++; //Aumenta la variable contador
            genie.WriteObject(GENIE_OBJ_LED_DIGITS, LD_OKHC, Counter_OK); //Y le asigna el
            número de la variable contador al LedDigital (Pantalla de 7 segmentos)
            clearImages(); //Pone todas las imagenes en blanco de nuevo
        }
        else if(VoltajeVentanaUpH1 == true && VoltajeVentanaDownH1 == true){ //Si se oprimieron
        los 2 botones de la pieza de alta sin calefactor entra aquí y hace lo mismo que la de arriba
        con calefactor
            Counter_OK++;
            genie.WriteObject(GENIE_OBJ_LED_DIGITS, LD_OKHC, Counter_OK);
            clearImages();
        }
        else{ //Si oprime el botón WB_OKHC y no se completó la secuencia manda el mensaje
        de abajo en el Screen String
            genie.WriteStr(S_HC, "Secuencia no \ncompletada");
        }
    }
}

```

Finalmente, dentro del “loop” se agrega un “switch” el cual sirve para saber en qué vista de la pantalla se encuentra actualmente el usuario y con base en eso poder usar las funciones de esa vista, en este caso dentro de la vista para probar la pieza se mandan a llamar todas las funciones para checar las activaciones de la pieza.

```
void loop()
{
  genie.DoEvents();
  switch (mode)
  {
    case F_PiezaHC: //Se agrega un case con la forma de la pieza de alta
      Read_ADC_TC275(); //Se llaman a las funciones de lectura, para poder leer
      los voltajes de baja y de alta
      Read_ADC_ADS8688();
      Window_ActHC(); //Se llaman a las funciones que leerán los voltajes de
      alta (ventana y calefactor)
      Warm_ActHC();
      break;

    case F_PiezaLC: //Se agrega un case con la forma de la pieza de baja
      Read_ADC_TC275(); //Se llama a la funcion de lectura, para poder leer el voltaje de baja
      Window_ActLC(); //Se llaman a las funciones que leerán los voltajes de baja
      (ventana y calefactor)
      Warm_ActLC();
      break;
  }
}
```

Con esto finalizó el código y las formas (vistas de la pantalla) de forma exitosa, para que la fixtura de “Rear” fuera utilizada por cualquier operador de la planta. En la sección de “Resultados” se mostrará la fixtura trabajando.

Capítulo 6

Resultados

Los resultados de todas las fixturas que se realizaron se desglosarán en seguida, pero es importante mencionar que todas las fixturas se realizaron de forma exitosa y quedaron funcionando dentro de “KOMEX II”. Los resultados obtenidos fueron tan favorables que todas las fixturas que se realizaron para este proyecto, se usan diariamente en la planta para probar piezas.

Nuevos Proyectos con “KIA”

Las tres fixturas de los nuevos productos “KIA” con las que se trabajó, se pueden observaren las figuras 30, 31 y 32, se puede ver cómo se están probando las piezas en la interfaz que se realizó y ahí mismo se puede apreciar que las imágenes cambian de color según los accionamientos que se realicen, es decir, las fixturas están respondiendo correctamente a los accionamientos de las piezas.

Ya que las líneas de “KIA” son nuevas dentro de “KOMEX II” es más probable que lleguen a presentar fallas y que algunas piezas con fallas no sean detectadas dentro de las líneas de producción, lo cual es un problema muy grave, ya que son piezas que recibirá el cliente (“KIA”) y esto ocasionaría reclamos en el futuro, por lo que a los pocos días de ir terminado las fixturas, se fueron realizando contenciones para hacer una segunda validación de que las piezas que salieron de las líneas de producción funcionaban correctamente.

Las contenciones siempre se realizan con fixturas, por lo que era necesario que como practicantes del área de “Análisis” termináramos de programar las fixturas, para que fueran usadas para este propósito. También las fixturas fueron usadas en el área de análisis, ya que cuando se empezó a producir los nuevos productos de “KIA” dentro de “KOMEX II”, empezó a haber fallas en algunas piezas y los operadores del área de “Análisis” tenían que revisar las piezas (“Assist” y “Rear”) usando multímetros y fuentes de voltaje, lo cual gasta más tiempo y es más susceptible a errores que usar una fixtura.

Capítulo 6. Resultados.

La fixtura de “Driver” se terminó poco antes de que la línea de “Driver” empezara a producir piezas, lo cual benefició mucho a los trabajadores del área de “Análisis”, ya que las piezas de “Driver” no podrían haber sido probadas con fuentes de voltaje y multímetros, ya que esas piezas funcionan por comunicación “LIN” y no por una variación de voltaje en los pines. Por todo lo mencionado, las fixturas de los nuevos proyectos de “KIA” fueron demasiado útiles, para evitar mandar piezas con fallas al cliente (lo cual es un tema muy crítico) mediante las contenciones, y para que todos en el área de “Análisis” (tanto operadores como ingenieros), pudieran probar las piezas que llegaran al área.



Figura 30 Fixtura "Assist" trabajando en contención.



Figura 31 Fixtura "Rear" trabajando.



Figura 32 Fixtura "Driver" trabajando en contención.

“Head Lamp” de “General Motors”

Se terminó la fixtura de “Head Lamp” y funcionó correctamente, en la figura 33 se puede observar la fixtura que se realizó, trabajando correctamente, y se ve cómo las funciones que se están seleccionando en la pieza se ven reflejadas en la pantalla. Esta fixtura se dejó en el área de análisis, para que los operadores del área pudieran probar todas las piezas que llegan de las líneas de “Head Lamp”.

Capítulo 6. Resultados.

La realización de esta fixtura fue muy importante sobre todo para los operadores del área de “Análisis”, ya que la fixtura que se tenía antes en el área fue enviada al extranjero, por lo que únicamente se tenía una fixtura para esta pieza dentro de toda la planta, y esta fixtura generalmente se tenía en contención, entonces cuando llegaba una pieza de “Head Lamp” con falla al área de “Análisis” los operadores tenían que desplazarse hasta la contención para poder probar la pieza y además se tenía que interrumpir el trabajo de la persona que estaba haciendo la contención. Para que diera permiso de probar una pieza de “Análisis”, pero ya con la fixtura que se realizó, ya no era necesario estarse desplazando hasta la contención para hacer esta prueba y tampoco interrumpir el trabajo del operador que estuviera en contención, por lo que se ahorra tiempo y se hace más eficiente tanto el trabajo de la contención como el de “Análisis”.

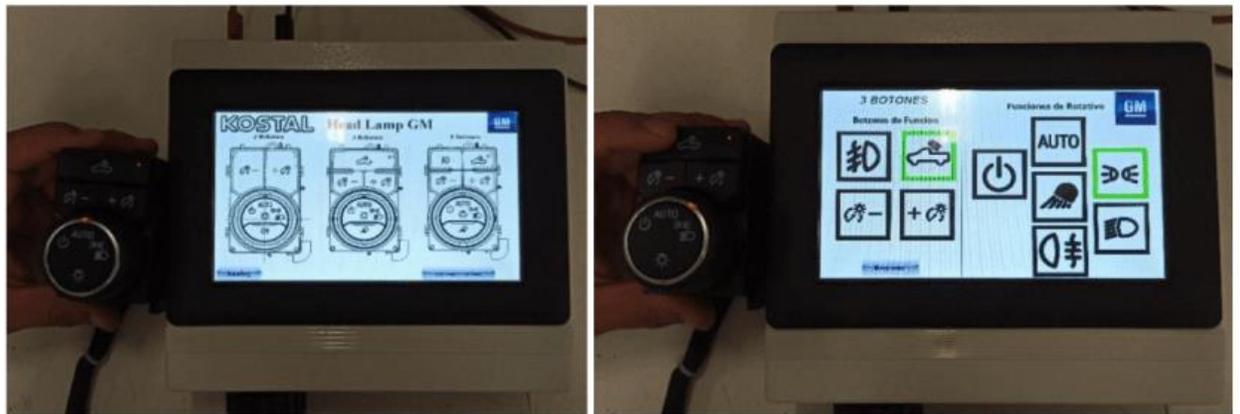


Figura 33 Fixtura "Head lamp" trabajando.

Capítulo 7

Análisis de Resultados

Los análisis de los resultados obtenidos recaen en especial sobre el banco de pruebas de “Mini ICP – CX430” de “Ford” los resultados de esta fixtura fueron muy gratificantes, debido a que antes se usaba una fixtura para probar la pieza “Mini ICP” y otra fixtura para probar la pieza “CX430”, pero ahora no solamente se logró reemplazar la fixtura de “CX430” que se envió con el cliente “Ford” sino que se creó una fixtura multifuncional que para fines prácticos es mucho más eficiente que estar probando con dos fixturas diferentes.

Los resultados de la fixtura se pueden observar en la figura 34, 35 y 36, donde se aprecia el menú de selección, para seleccionar la pieza a probar, y también se aprecia como se prueba una pieza de “Mini ICP” y otra de “CX430” y de acuerdo a la correcta programación e interfaz que se hizo, la activación de los accionamientos de los botones y las perillas se ven reflejados en la pantalla cuando son utilizados.

El haber realizado una fixtura pequeña, y que además cuente con la interfaz para probar dos piezas, fue de mucha ayuda en el área de “Análisis”, ya que la fixtura que se tenía de “CX430” se mandó con el cliente, por lo que cuando llegaba una pieza al área de “Análisis”, pasaba lo mismo que con “Head Lamp”, había que ir hasta la contención e interrumpir el trabajo del operador en turno, para poder probar la pieza. Además, la fixtura que se tenía de “Mini ICP” era pesada y necesitaba conectarse a una toma de aire, por lo que después de realizar esta nueva fixtura, tanto operadores como ingenieros solían utilizar más esta fixtura para probar las piezas de “Mini ICP” y “CX430”.



Figura 34 Menú para seleccionar la pieza a probar.



Figura 35 . Probando la pieza "Mini ICP" en la fixtura.



Figura 36 Probando la pieza "CX430 en la fixtura".

Capítulo 8

Conclusiones y trabajo a futuro

Después de trabajar en cada una de las fixturas ya expuestas anteriormente, y ver el uso que se les da dentro de la planta, se observó que las fixturas son una herramienta muy importante dentro de “KOMEX II” y son proyectos en los que se tiene que estar trabajando constantemente ya que siempre se necesitarán nuevas fixturas debido a la demanda por nuevos proyectos, fixturas faltantes, actualizaciones, etc.

Con cada fixtura que se trabajó en este proyecto se logró aportar algo a la empresa, las fixturas de “KIA” las usaron inmediatamente en contenciones para evitar que se fuera alguna pieza mala con el cliente, lo cual originaría reclamos en un futuro. Al ser las primeras fixturas con las que se trabajó, fue gratificante ver como el trabajo que se estaba realizando como practicantes del área de “Análisis” era muy importante, ya que esas fixturas de “KIA” que se programaron, era el último filtro para detectar fallas en las piezas, después de eso las piezas fueron enviadas con el cliente “KIA”.

La fixtura de “Head Lamp” era extremadamente necesaria en el área de análisis, ya que esta pieza tiene mucha demanda, por lo que hay dos líneas produciendo esta pieza, lo cual también implica mayor número de fallas y los operadores del área, necesitaban la fixtura para poder probar las piezas que trajeran para analizar.

La fixtura “Mini ICP – CX430” se convirtió en una de las más usadas, para probar las piezas, ya que la fixtura “Mini ICP” que se tenía, era más pesada y difícil de manipular, y ahora al tener una fixtura pequeña y que además sirve para probar dos piezas, es mucho más práctico para los operadores tomar esta fixtura. Además, al hacer la fixtura multifuncional, se logró ahorrar dinero en materiales, espacio para guardarla y el tiempo de elaboración, por lo que la tendencia en la elaboración de fixturas podría seguir este camino para futuros proyectos.

Finalmente, la fixtura de “C1XX” que fue la última que se pudo realizar, fue un gran aporte de nosotros como practicantes para la empresa, ya que ayudamos a brindar una solución 105 eficiente y confiable para un tema muy crítico como lo fue los “PCBs” con terminales levantadas.

Con la implementación de la fixtura se reduce la cantidad de operadores que se necesitan para validar que una “PCB” de “C1XX” funcione correctamente y aumenta la confiabilidad del resultado, así como también el tiempo que se necesita para validar esto, y más importante que ahorrar dinero para la empresa, el hecho de mandar tantas piezas malas con los clientes repercute directamente en la calidad de los productos de “KOMEX II”, lo cual puede generar que se pierdan certificaciones de calidad con las que ya cuenta la empresa y esto afecta a todo “KOMEX II”, por lo que fue muy gratificante poder aportar esta fixtura para atacar ese problema.

El objetivo del proyecto se cumplió de forma exitosa, y la repercusión del proyecto dentro de la empresa fue evidente y muy importante, ya que cada fixtura con la que se trabajó en este proyecto, se podía ver todos los días en funcionamiento dentro de la planta, eran herramientas de uso diario, ya fuera en el área de “Análisis”, en las contenciones, o en alguna otra área que la necesitara, y al momento en que se concluye este proyecto se estaba analizando la posibilidad de integrar la fixtura de “C1XX” a la línea de producción. Se trató de aprovechar al máximo el tiempo para trabajar con las fixturas y hacer el mejor trabajo posible, pero aún quedan muchas áreas de oportunidad para mejorar en lo que corresponde al tema de las fixturas, como lo ya mencionado con fixturas multifuncionales, por lo que se espera se le pueda seguir dando seguimiento a todos estos temas.

Capítulo 9

Referencias bibliográficas

- Abadiano Abadiano, J. M., Albán Toapanta, F. A., & Aldas Castillo, E. J. (2020). *Diseño e implementación de un banco comprobador de computadoras automotrices para vehículos Chevrolet Aveo activo*. Obtenido de SCRIBD: <https://es.scribd.com/document/462366233/Proyecto-Comprobador-de-computadores-verificado-2-3>
- Arduino. (2022). *Arduino*. Obtenido de Conceptos de Arduino: <https://www.arduino.cc/>.
- Automotriz, D. (2018). *Directorio Automotriz*. Obtenido de Directorio Automotriz: <https://www.directorioautomotriz.com.mx/listing/search/details/842>
- Autosoporte. (s.f.). *Banco de prueba de computadora*. Obtenido de Autosoporte.com: <https://autosoporte.com/lp/equipo-de-ecu-banco-de-prueba-de-computadora-automotriz-ecu/>
- Blumnovotest. (s.f.). *Bancos de pruebas para arboles de transmisión*. Obtenido de Bancos de pruebas para arboles de transmisión: <https://www.blumnovotest.com/mx/productos/prueftechnik/bancos-de-pruebas-para-arboles-de-transmision/>
- Carecarcenter. (2021). *Servicio técnico automotriz*. Obtenido de Banco de pruebas Bogota: <https://www.c3carecarcenter.com/banco-de-prueba-bogota/>
- Eduardo, F. (2006). *Diseño y construcción del banco de pruebas para comprobación de computadoras automotrices con inyección electrónica a gasolina*. Obtenido de bitstream: <http://repositorio.espe.edu.ec/xmlui/bitstream/handle/21000/4165/T-ESPEL-0222.pdf?sequence=1&isAllowed=y>
- Freepng. (2022). *Freepng.es*. Obtenido de Freepng.es: <https://www.freepng.es/png-2kia2m/>
- Huircán, J. (s.f.). *Convertidores Análogo-Digital y Digital-Análogo*. Obtenido de Conceptos básicos: <https://lc.fie.umich.mx/~azm/ad03.pdf>
- KIA. (2021). *KIA in MX*. Obtenido de KIA in MX: <https://www.kia.com/mx/main.html>
- KOSTAL. (2022). *KOSTAL GROUP*. Obtenido de KOSTAL GROUP: <https://www.kostal.com/en-gb>
- L. Sánchez, M. M. (2016). *Revisión documental del protocolo CAN como herramienta de comunicación y aplicación en vehículos*. Obtenido de Int.mxpublicaciontecnica.: <https://imt.mx/archivos/Publicaciones/PublicacionTecnica/pt474.pdf>
- Ldt, 4. S. (2022). *Arduino*. Obtenido de genieArduino.
- Mecmecin. (2021). *Bancos de pruebas motorizados*. Obtenido de Bancos de pruebas motorizados: <https://www.mecmesin.com/es/torque-systems/bancos-de-pruebas-motorizados>

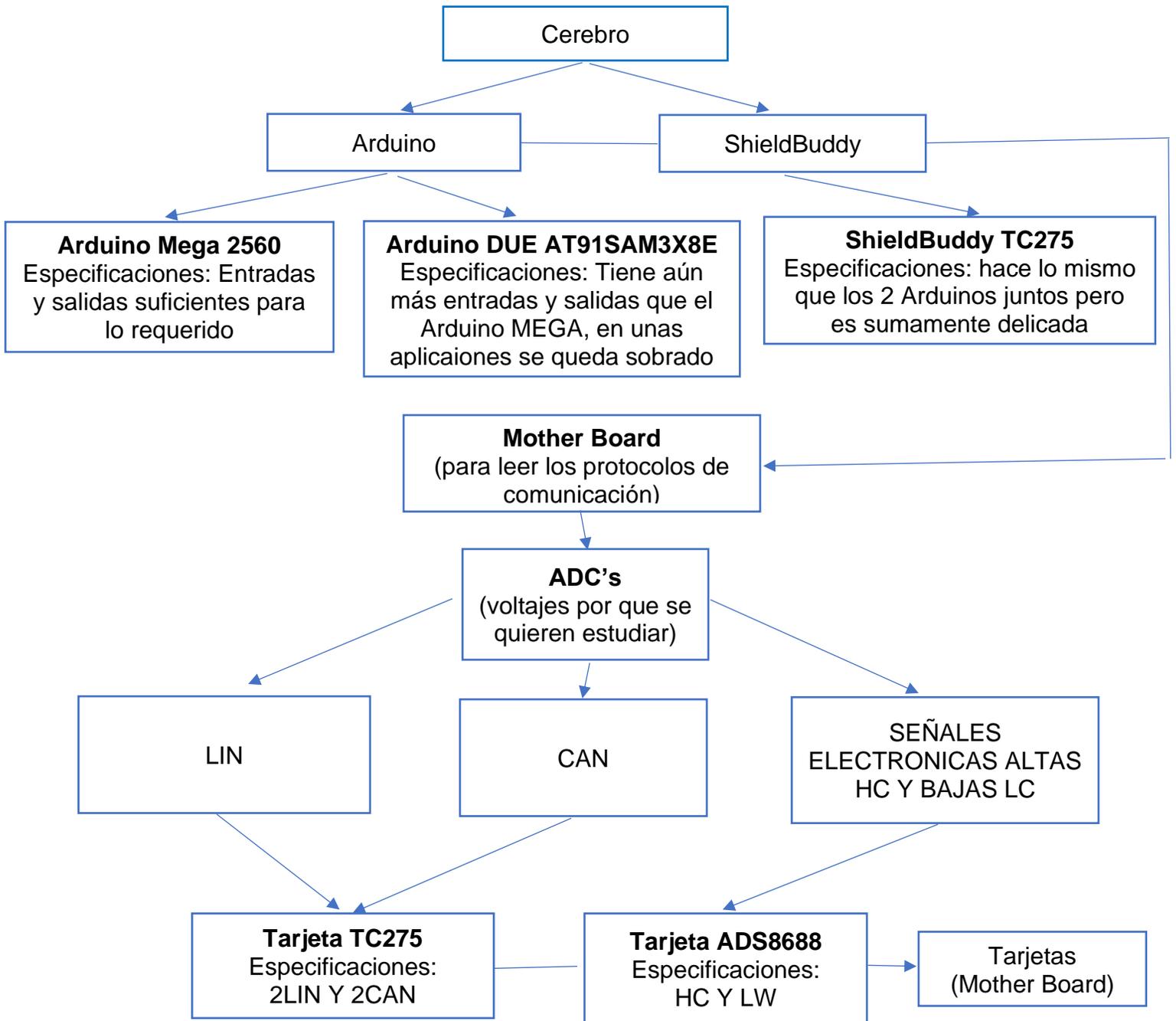
Referencias bibliográficas.

- NIDEC. (2022). *Soluciones Industriales*. Obtenido de Soluciones Industriales: <https://www.nidec-industrial.com/es/markets/bancos-de-pruebas/bancos-de-pruebas-para-el-sector-automovilistico/>
- ShieldBuddy, S. (2018). *Shield ShieldBuddy de Hiltex con nucleo TriCore*. Obtenido de RS Productos electronicos y mantenimiento: <https://es.rs-online.com/web/p/kits-de-desarrollo-de-microcontroladores/1245257/>
- Systems, 4. (2022). *4D Systems Pty Ltd*. Obtenido de genieArduino: <https://www.arduino.cc/reference/en/libraries/geniearduino/>
- Systems, 4. (2022). *4D Systems turning technology into art*. Obtenido de 4D Systems turning technology into art: <https://4dsystems.com.au/>

Anexos

Diagramas

Diagramas a bloques del banco de pruebas para piezas automotrices en general:



ADC: Los adc's o analog-digital coverter (convertidor de analógico a digital consiste en la conversión de señales analógicas en señales digitales, con el propósito de facilitar su procesamiento y hacer la señal resultante más inmune al ruido y otras interferencias a las que son más sensibles las señales analógicas.

Arduino Mega 2560: Es una tarjeta de desarrollo open-source construida con un microcontrolador modelo Atmega2560 que posee pines de entradas y salidas (E/S), analógicas y digitales. Esta tarjeta es programada en un entorno de desarrollo que implementa el lenguaje Processing/Wiring.

BootLoader: Un gestor de arranque, es un programa que permite elegir el código a ejecutar en el proceso de arranque. Habitualmente el gestor de arranque forma parte del cargador de arranque.

CA: Corriente alterna.

CC: Corriente continua.

Fixtura: Nombre adquirido del banco de pruebas en distintas empresas.

GEN4-ULCD-70FT: Es una HMI que se utilizó para el banco de pruebas es una pantalla táctil de 7 pulgadas capacitiva.

Hardware: Se refiere a las partes físicas, de un sistema informático, sus componentes eléctricos, electrónicos, electromecánicos internamente y externamente lo físico.

HC: High current, alta corriente.

HMI: El Interfaz Hombre-Máquina (HMI) es el interfaz entre el proceso y los operarios; es básicamente un panel de instrumentos del operario. Es la principal herramienta utilizada por operarios y supervisores de línea para coordinar y controlar procesos industriales y de fabricación.

ICSP: En la placas arduino se ven 6 pines que pone ICSP (In Circuit Serial Programming) que sirve para programar el BootLoader del Microcontrolador ATmega y asi poder cargar los programas que creamos en el IDE directamente en el microcontrolador sin tener que necesitar Programadores externos.

IDE: Es el entorno de desarrollo integrado o entorno de desarrollo interactivo, en inglés Integrated Development Environment, es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

Instrumento virtual: Es lo que permite al usuario interactuar con la computadora, como si se estuviera utilizando un instrumento electrónico justo a sus requerimientos, gracias a la utilización de un conjunto de software y hardware, agregado a un PC.

IoT: El Industrial IoT se define como la red de dispositivos, maquinaria y sensores conectados entre sí y a internet, con el objetivo de recoger datos y analizarlos para aplicar esta información a una mejora en los procesos.

Las principales aplicaciones del Industrial LOT son:

- Gestión y monitoreo automatizada y remota de los equipos.
- Mantenimiento predictivo.
- Aplicación de mejoras más rápida.
- Inventarios a punto.
- Control de calidad.
- Optimización de la cadena de suministro.
- Mejora en la seguridad en planta.

LC: Low current, baja corriente.

Licencia Creative Commons: Son una herramienta que permite a los usuarios usar obras protegidas por derecho de autor sin solicitar el permiso del autor de la obra.

Pantalla táctil capacitiva: Este tipo de pantalla utiliza el cuerpo humano como conductor de electricidad para el uso del táctil.

Pantalla táctil resistiva: Este tipo de pantalla funciona con un mecanismo de presión del dedo o un lápiz para el uso del táctil.

Pines: Cada una de las patillas metálicas de un conector.

Programación AVR-C: Son los microcontroladores AVR que contienen las dos placas usadas en este documento, esta es la tecnología preferida en muchas industrias por sus características únicas y sobresalientes. Estos dispositivos ofrecen una combinación equilibrada y única de rendimiento, flexibilidad de diseño y eficiencia energética. En este curso aprenderás a programar Microcontroladores AVR® a nivel de registros, configurar los diversos periféricos, utilizar librerías y explotar dichas características que te ayudarán en tu desarrollo profesional en diversas aplicaciones con Microcontroladores AVR sacando provecho de las poderosas herramientas de hardware y software incluidos en las nuevas tecnologías de Microchip/Atmel.

PWM: Pulse Width Modulation (Modulación por ancho de pulso), La modulación por ancho de pulsos de una señal es una técnica en la que se modifica el ciclo de trabajo de una señal periódica, ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

Software: Son el conjunto de los componentes lógicos necesarios que hace posible la realización de tareas específicas internamente lo virtual.

ShieldBuddy: Es una placa de desarrollo que dispone del procesador de tres núcleos AURIX Infineon TC275 esta placa de desarrollo dispone de licencias de evaluación, y después recibirá acceso completo a todo un conjunto de herramientas de desarrollo. La placa de desarrollo ShieldBuddy cuenta con el factor de forma Arduino Mega y proporciona acceso a una amplia gama de placas de accesorios y placas de extensión.

Workshop4: Es el IDE con el cual se programan las pantallas de los bancos de prueba.

Wiring: Es una plataforma para programación. Se inicio en 2004 por Hernando Barragán, alumno de Ben Fry y Casey Reas (los creadores de Processing) y la idea era crear una plataforma cuya capacidad de acción fuera más allá de los límites de la pantalla para llegar al mundo físico.