



Universidad Autónoma de Querétaro

Facultad de Ingeniería
Campus San Juan del Río
Electromecánica



TESIS

SOFTWARE-HARDWARE PARA PRUEBAS DE FALLA EN CABLES PLANO-FLEXIBLE

Que como parte de los requisitos para obtener el título de
Ingeniero Electromecánico
Línea terminal en
Diseño y Manufactura

Presenta:

Raúl Eduardo Villagrán Trejo

Dirigida por:

Dr. Martín Valtierra Rodríguez

San Juan del Río, Qro., Octubre de 2018



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
 FACULTAD DE INGENIERÍA
 CAMPUS SAN JUAN DEL RÍO



**SOFTWARE-HARDWARE PARA PRUEBAS DE FALLA
 EN CABLES PLANO-FLEXIBLE**

TESIS

Como parte de los requisitos para obtener el título de

INGENIERO ELECTROMECAÁNICO

Línea Terminal

DISEÑO Y MANUFACTURA

Presenta

RAÚL EDUARDO VILLAGRÁN TREJO

Dirigido por:

DR. MARTÍN VALTIERRA RODRÍGUEZ

SINODALES:

DR. MARTÍN VALTIERRA RODRÍGUEZ (11869)
 Presidente

DR. LUIS MORALES VELÁZQUEZ (6829)
 Secretario

DR. JESÚS ROONEY RIVERA GUILLÉN (8943)
 Vocal

DR. JUAN PRIMO BENÍTEZ RANGEL (2088)
 Suplente

San Juan del Río, Qro., Noviembre 2018.

Resumen

Durante la fabricación de distintos equipos eléctricos y electrónicos es necesario detectar los diferentes tipos de desperfectos que puedan presentar. Uno de los principales problemas que se pueden identificar durante la fabricación son las fallas eléctricas en alguno de sus conductores. En el presente proyecto se realizó un prototipo para la detección de dichas fallas eléctricas aplicadas a cables plano-flexibles. Este sistema es controlado por medio de una interfaz de usuario intuitiva comunicada mediante un microcontrolador Arduino, que permite la selección de tres distintos tipos de falla comunes en dichos cables, estas son: continuidad, cortocircuito y conductores cruzados. La tarjeta Arduino, por otro lado, controla la señal de salida y se encarga de leer la señal de respuesta por medio de los circuitos integrados; 74HC959 (registros de cambio) y CD4021 (Multiplexores), que permiten extender la cantidad de entradas y salidas permitidas por Arduino. Este sistema permite al usuario detectar específicamente que conductor de los que constituyen el cable presenta alguna falla, identificable por medio de señales visuales en la interfaz de usuario. Al mismo tiempo, permite el análisis de cables de distintos tamaños al incluir conectores con diferente número de pines. Finalmente, el probador de cables plano-flexible cuenta con la configuración en modo portátil para permitir al usuario el uso del equipo sin necesidad de una PC. Los resultados muestran la funcionalidad del dispositivo.

Palabras clave: Arduino, interfaz gráfica de usuario, cable plano-flexible, continuidad, cortocircuito, cables cruzados, Visual Studio.

Agradecimientos

A mis padres, José Raúl Villagrán Jiménez y Josefina Trejo Ferrusca, por su apoyo y amor brindado todos estos años.

A mi hermano Jesús Ángel Villagrán Trejo, porque sé que puedo contar con él en cualquier etapa de mi vida.

A mi novia Valeria Gaytán Molina, por su comprensión, cariño y consejos brindados a lo largo de todos estos años.

A mi asesor, el Dr. Martín Valtierra Rodríguez, por el apoyo, orientación y confianza para la realización de este proyecto.

Agradezco a la Universidad Autónoma de Querétaro, por permitirme formar parte de esta institución y darme las herramientas necesarias para desarrollarme como profesionalista.

A Dios, por permitirme lograr esta meta.

Índice

Capítulo 1	1
Introducción	2
1.1 Antecedentes	2
1.2 Descripción del problema.....	7
1.3 Justificación	9
1.4 Objetivos	10
1.4.1 Objetivo general.....	10
1.4.2 Objetivos particulares	10
1.5: Planteamiento general	10
Capítulo 2.....	15
Fundamentación teórica y tecnológica.....	16
2.1 Cables planos y sus aplicaciones	16
2.2 Tipos de fallas.....	16
2.3 Visual Studio y Windows forms application.....	19
2.3.1 Simbología y descripción de las herramientas de Windows Forms Application	20
2.4 Tecnología Arduino	22
2.4.1 Simbología y descripción de las herramientas de Arduino	23
2.4.2 Comandos de programación de Arduino	24
2.5 Registro de corrimiento serie-paralelo y paralelo-serie.....	25
2.6 Display LCD	30
Capítulo 3.....	32
Metodología	33
3.1 Diseño de prototipo.....	33
3.2 Circuito de acondicionamiento eléctrico.....	54
3.2.1 Circuitos de acondicionamiento para registros de cambio 74HC959.....	54
3.2.2 Circuitos de acondicionamiento para multiplexores CD4021.....	57
3.2.3 Circuito de acondicionamiento para Display LCD.....	60
3.2.5 Montaje de circuitos dentro del prototipo	62
3.3 Interfaz de usuario	64

Capítulo 4.....	74
4.0. Experimentación	75
4.1 Matriz de pruebas	76
4.2. Pruebas y resultados	77
4.2.1. Pruebas y resultados de continuidad.....	77
4.2.2. Pruebas y resultados de cortocircuito	79
4.2.3 Pruebas y resultados de inversión de cables.....	81
4.3. Resultados de pruebas en PC	83
4.3.1. Pruebas de Continuidad	85
4.3.2. Pruebas de cortocircuito	87
4.3.3. Pruebas de cables cruzados.....	88
4.3.4 Pruebas a cable sano	89
4.4 Resultados de pruebas en modo Portable	91
4.4.1. Pruebas de Continuidad	92
4.4.2. Pruebas de Cortocircuito	93
4.4.3. Pruebas de Cables cruzados	94
4.4.4. Pruebas en un cable sano	95
4.5. Matriz de confusión	96
4.6. Conclusiones.....	97
Referencias.....	99

Índice de Figuras

Capítulo 1

Fig. 1.1 Ejemplos de cables plano flexible	4
Fig. 1.2 Ejemplos de diferentes conectores.....	5
Fig. 1.3 Diagrama de conexión entre componentes	131
Fig. 1.4 Metodología general del proyecto	133

Capítulo 2

Fig. 2.1 Falla en cables debido a un corto circuito.	17
Fig. 2.2 Falla en cable debido a desconexión de terminales en cables.....	18
Fig. 2.3 Falla en cable debido a fracturas o cortes por el mal manejo	18
Fig. 2.4 Cables invertidos.	19
Fig. 2.5 Logo de Visual Studio, 2017.....	19
Fig. 2.6 Windows form application en Visual Studio 2017.....	20
Fig. 2.7 Arduino Mega2560.	22
Fig. 2.8 Ventana principal del software de Arduino.	23
Fig. 2.9 void setup.	24
Fig. 2.10 loop ().	24
Fig. 2.11 Diagrama de terminales CD4021.	26
Fig. 2.12 Diagrama de terminales 74HC959.	27
Fig. 2.13 Multiplexores en cascada.	29
Fig. 2.14. Registro de cambio en cascada.	30
Fig. 2.15 Diagrama de pines LCD 16x2.	31

Capítulo 3

Fig. 3.1 Diagrama de conexión entre componentes.....	33
Fig. 3.2: Circuitos de acondicionamiento junto con cable bajo prueba.....	34
Fig. 3.3 Ubicación de cada uno de los circuitos.	35
Fig. 3.4 Respuesta recibida por el programa y mostrada en pantalla la ubicación del botón presionado.....	36
Fig. 3. 5 Ensamble final de circuitos junto con PC mostrando interfaz de usuario.....	37
Fig. 3.6 Base de pines para conectar cables.....	37
Fig. 3.7 Cable plano-flexible montado sobre la base de pines.	38
Fig. 3.8 Pruebas de funcionamiento del display LCD.....	38
Fig. 3.9 Diseño de prototipo.	39
Fig. 3.10 Prototipo con cable de alimentación.....	40
Fig. 3.11 Vista Superior.....	40
Fig. 3.12 Vista previa de la sección "Modo".....	41
Fig. 3.13 Vista previa de la sección "Tipo de prueba".	42
Fig. 3.14 Vista previa de la sección "Estado".	43

Fig. 3.15 Vista previa de la sección "Central".....	43
Fig. 3.16 Vista trasera de tapa superior.	44
Fig. 3.17 Vista a la cara A.	45
Fig. 3.18 Vista a la cara B.	45
Fig. 3.19 Vista a la cara C.	46
Fig. 3.20 Vista a la cara D.	46
Fig. 3.21 Corte transversal.	47
Fig. 3.22 Corte transversal visto desde otro ángulo.	47
Fig. 3.23 Prototipo conectado directamente hacia la PC.....	48
Fig. 3.24 Cada una de las caras con sus respectivas perforaciones.....	49
Fig. 3.25 Cara superior después de realizar los cortes.	49
Fig. 3.26 Cara superior con los elementos instalados.....	50
Fig. 3.27 Plantilla de las etiquetas.....	50
Fig. 3.28 Tapa superior con etiquetas.	51
Fig. 3.29 Elaboración de base para placa de circuitos.	52
Fig. 3.30 Base de soporte finalizada.	52
Fig. 3.31 Base montada sobre la cara superior.....	53
Fig. 3.32 Resultado final de prototipo.....	53
Fig. 3.33 Diagrama de conexión de registros de cambio en cascada.	55
Fig. 3.34 Vista previa del Gerber para la realización de la PCB de los registros de cambio.....	56
Fig. 3.35 Modelo en 3D de la vista superior de la tarjeta PCB para registros de cambio.....	56
Fig. 3.36 Modelo en 3D de la vista inferior de la tarjeta PCB para registros de cambio.....	56
Fig. 3.37 Resultado final de PCB para registros de cambio, vista superior.....	57
Fig. 3.38 Resultado final de PCB para registros de cambio, vista inferior.....	57
Fig. 3.39 Diagrama de conexión de multiplexores en cascada.	58
Fig. 3. 40 Vista previa del Gerber para la realización de la PCB de los multiplexores.	58
Fig. 3.41 Modelo en 3D de la vista superior de la tarjeta PCB para multiplexores.....	59
Fig. 3.42 Modelo en 3D de la vista inferior de la tarjeta PCB para multiplexores..	59
Fig. 3.43 Resultado final de PCB para multiplexores, vista superior.	59
Fig. 3.44 Resultado final de PCB para multiplexores, vista inferior.....	60
Fig. 3.45 Vista previa del Gerber para la realización de la PCB del display LCD..	60
Fig. 3.46 Modelo en 3D de la vista superior de la tarjeta PCB para el display LCD.	61
Fig. 3.47 Modelo en 3D de la vista inferior de la tarjeta PCB para el display LCD.....	61
Fig. 3.48 Resultado final de PCB para el display LCD, vista superior.	61
Fig. 3.49 Resultado final de PCB para el display LCD, vista inferior.	62
Fig. 3.50 PCB montadas dentro del prototipo.....	62
Fig. 3.51 Identificación de circuitos.	63
Fig. 3.52 Vista lateral del prototipo con vista interior.	64

Fig. 3.53 Vista general de la interfaz de usuario.	66
Fig. 3.54 Panel de pruebas dividido por secciones.	67
Fig. 3.55 Prueba de continuidad antes de realizar la prueba.	67
Fig. 3.56 Prueba de continuidad después de realizar la prueba.....	68
Fig. 3.57 Prueba de cortocircuito antes de realizar la prueba.	69
Fig. 3.58 Prueba de cortocircuito después de realizar la prueba.....	69
Fig. 3.59 Prueba de inversión de cables antes de realizar la prueba.	70
Fig. 3.60 Prueba de inversión de cables después de realizar la prueba.	70
Fig. 3.61 Sección de configuración.	71
Fig. 3.62 Barra de progreso y botones de opciones.....	71
Fig. 3.63 Diagrama de flujo del principio de funcionamiento de la interfaz de usuario.	72

Capítulo 4

Fig. 4.1 Ensamble final software-hardware.	75
Fig. 4.2 Ensamble software-hardware con circuitos internos visibles.....	75
Fig. 4.3 Cable con falla de continuidad.	78
Fig. 4.4 Interfaz de usuario de pruebas durante el análisis de continuidad.	78
Fig. 4.5 Cable con falla de cortocircuito.	79
Fig. 4.6 Prueba realizada en un cable dañado.	80
Fig. 4.7 Prueba realizada en un cable sano.	81
Fig. 4.8 Inversión de cables.....	81
Fig 4.9 Prueba de cables cruzados en cable dañado.....	82
Fig. 4.10 Prueba realizada a un cable sano.	83
Fig. 4.11 Cable de pruebas, vista lateral.	84
Fig. 4.12 Cable de pruebas, vista superior.....	84
Fig. 4.13 Switch de modo en la posición de PC.	85
Fig. 4.14 Prueba de continuidad.....	86
Fig. 4.15 Prueba de continuidad realizada en modo PC.	86
Fig. 4.16 Prueba de cortocircuito.....	87
Fig. 4.17 Prueba de cortocircuito realizada en modo PC.	88
Fig. 4.18 Prueba de cortocircuito.....	89
Fig. 4.19 Prueba de cortocircuito realizada en modo PC.	89
Fig. 4.20 Prueba de cortocircuito.....	90
Fig. 4.21 Prueba de cortocircuito realizada en modo PC.	90
Fig. 4.22 Switch de modo en la posición de PC.	91
Fig. 4.23 Sección de tipo de prueba, se muestran indicadores luminosos y botones de pruebas.	92
Fig. 4.24 Prueba de continuidad en modo Portable.	93
Fig. 4.25 Resultado de prueba de Continuidad en modo Portable.....	93
Fig. 4.26 Prueba de cortocircuito en modo Portable.	94
Fig. 4.27 Resultado de prueba de Cortocircuito en modo Portable.....	94

Fig. 4.28 Prueba de Cables cruzados en modo Portable.	95
Fig. 4.29 Resultado de prueba de Cables Cruzados en modo Portable.....	95
Fig. 4.30 Prueba de Cables cruzados en modo Portable.	96
Fig. 4.31 Resultado de prueba de Cable sano en modo Portable.	96

Índice de tablas

Tabla 1	Tabla de funciones de pines para el CD4021.....	27
Tabla 2	Funciones de pines para el 74HC959.	28
Tabla 3	Funciones de los pines del display LCD 16x2.	31
Tabla 4	Matriz de pruebas de cables en modo PC.....	76
Tabla 5	Matriz de pruebas de cables en modo Portable.	77
Tabla 6	Matriz de confusión a pruebas realizadas a cables.	97

Capítulo 1

Introducción

Introducción

1.1 Antecedentes

Hoy día, los equipos y procesos de automatización son cada vez más complejos gracias al acelerado avance tecnológico que existe. Esto ayuda a aumentar la velocidad de producción en las empresas y a disminuir los costos de producción.

En general, los sistemas de automatización involucran la interconexión de varios equipos y tarjetas de control que se encargan de transferir información de forma bidireccional entre sensores y actuadores. En este sentido, es de suma importancia que todos los elementos del sistema puedan funcionar de manera correcta y eficiente. En particular, la interconexión de equipos se hace a través de cables de distintos tipos como pueden ser cable coaxial, cable de par trenzado, cables de fibra óptica, alambre de cobre, etc. A pesar de que estos cables suelen ser muy robustos, los estreses mecánicos y eléctricos de las condiciones de operación pueden generar fallas de corto circuito, sobrecarga o pérdida de aislamiento, incluso calentamiento indirecto sobre otros componentes. Una falla, por pequeña que sea, termina afectando a todo el sistema ya que cada cable cumple con una función necesaria para el correcto funcionamiento del equipo. Además, la presencia de falla, a corto, mediano o largo plazo, implicará costos por degradación del mismo equipo u otros que estén conectados al equipo, paros en líneas de producción e inclusive accidentes que podrían ser fatales para los operadores de los equipos.

Pero no todas las fallas son producidas por el desgaste de las máquinas o equipos cuando ya están en operación, sino que también pueden tener fallas que se presentaron desde el momento de su fabricación, ya sea por una mala instalación o por la mala calidad de los materiales del que fue fabricado. En este sentido, las empresas que fabrican e integran equipos no pueden permitirse que sus equipos salgan con defectos ya que, por una parte, pierden credibilidad y confiabilidad de sus clientes, lo cual es muy importante hoy en día debido a la alta competitividad.

Por otro lado, ellos también presentarían pérdidas económicas debido al regreso de equipos, a los costos de mantenimiento, etc. Sabiendo esto, las empresas invierten grandes cantidades monetarias con el objetivo de mantener y subir cada vez más sus estándares de calidad.

En específico, se sabe que en un proceso de revisión de calidad de cables, conductores o arneses estos se deben verificar varias veces con el objetivo de evitar errores humanos, así como de mejorar la confiabilidad por principios estadísticos. En este sentido, es lógico suponer que mientras más veces se revise un conductor, es decir que se revise si no presenta falla, mayor será la confiabilidad del diagnóstico; sin embargo, como se mencionaba al inicio, las empresas deben incrementar su productividad y ser competitivos, siendo imposible e incosteable invertir un tiempo exagerado en la revisión y diagnóstico de los conductores. Por lo tanto, es indispensable hacer un equilibrio entre rapidez de pruebas y la confiabilidad de estas. Un número excesivo de pruebas implicaría mayor tiempo, más horas hombre, mayor necesidad de personal, y menor rapidez en la producción. Otro punto que considerar es que estos conductores pueden ser de diversos tamaños, no solo en longitud, sino en la cantidad de hilos o conductores en un mismo cable. Por ejemplo, hay cables que pudieran tener 1, 2, 5, 10, 40 o cualquier otro número de conductores. Lo cual incrementa de cierta forma la complejidad y tiempos en las revisiones.

Aunque ya existen algunas soluciones comerciales para la prueba de falla de varios tipos de conductores, es decir, cables, pero con diferentes conectores, estos presentan ciertas desventajas con respecto a las necesidades cambiantes del sector industrial. Por ejemplo, los probadores de falla son de arquitectura cerrada, es decir ni su software ni su hardware puede ser modificado. Por lo tanto, la integración de tareas diferentes para las que ellos fueron fabricados no es posible, esto considerando obviamente tareas que pueden ser relacionadas. En palabras más sencillas, si el equipo solo puede con 3 tipos de conductores, el hardware no puede ser modificado para incluir nuevos tipos de conectores o, aunque el hardware

pueda ser modificado, el software de control no reconocería esta modificación. Esta cualidad es, en específico, la desventaja más seria para una industria que continuamente cambia las especificaciones de sus productos, ya que normalmente se involucran cambios en los cables y tipos de conductores a utilizar. Por lo tanto, la adquisición de equipos comerciales de arquitectura cerrada no es una opción eficiente ya que muy probablemente quedará obsoleta en función de los nuevos proyectos.

Desde este punto de vista, el contar con un sistema de arquitectura abierta (no dependiente de alguna empresa) que pueda ser reconfigurable tanto en software como hardware de una forma sencilla y práctica representaría una alternativa atractiva de solución a las tareas de prueba de falla en cables.

En base a lo anterior, en este trabajo se desarrolla un prototipo software-hardware para probar la condición física de cables plano-flexible. La Fig. 1.1 muestra un ejemplo de distintos cables planos flexible. También la Fig. 1.2 muestra ejemplos de cómo el conector puede cambiar, indicando que el si el sistema de pruebas no se actualiza o reconfigura para el nuevo conductor, el sistema quedaría obsoleto.

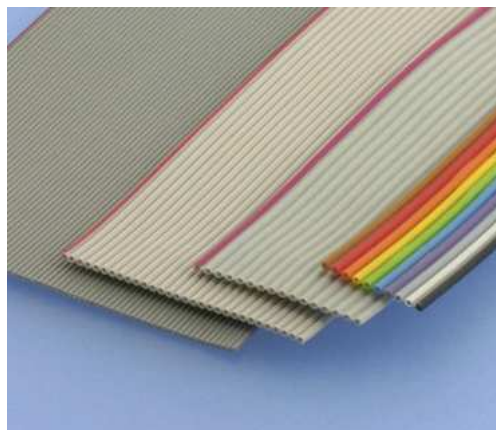


Fig. 1.1 Ejemplos de cables plano flexible.



Este tipo de cable es elegido principalmente por su fácil adquisición, es decir se puede adquirir prácticamente en cualquier tienda de electrónica; además, considerando que la principal motivación es desarrollar un software-hardware propio que pueda modificarse con relativa facilidad para disponer de otro tipo de conductores, el tipo de cable usado para la presentación de un prototipo puede considerarse no crítico.

En particular, el prototipo contará con una interfaz a computadora desde donde se tendrá acceso al tipo de prueba, tipo de conector (número de conductores). El hardware de control estará basado en una tarjeta microcontrolador Arduino, el cual que se encargará de enviar y recibir las señales para identificar si el cable está funcionando correctamente o no. Es importante mencionar que el software fundamentalmente considera número de conductores en vez de tipo de conector, esto con la finalidad de que el conector pueda ser modificado; de cualquier forma, este es irrelevante cuando se desea, por ejemplo, hacer una prueba de continuidad. Por otro lado, el hardware tiene el mismo principio, es decir considera la emisión de pulsos digitales por un conductor más que por un tipo de conector especial. El trabajo se dirige a cables flexible tipo plano, pero este solo es elegido para validar el software y el hardware.

Actualmente existen dispositivos utilizados para medir la continuidad en cables, como el multímetro, cuyo antecedente es el AVO (donde sus siglas vienen de las funciones que tenía: amperímetro, voltímetro y óhmetro), inventado por Donald Macadie con la finalidad de medir principalmente voltaje, corriente y resistencia (Gómez Belmontes, 2012-2016).

Además, existen varios probadores de continuidad de cables, un ejemplo de ellos es el probador de cables marca Intellinet multifuncional utilizado para medir continuidad en cables tipo UTP (del inglés *Unshielded Twisted Pair*), STP (del inglés *Shielded Twisted Pair*) y FTP (del inglés *Foiled Twisted Pair*) (Icintracom, 2016).

Otro ejemplo es el probador de cables y arneses CableEye, que puede identificar cables defectuosos o cruzados. El modelo M3U, tiene dos rangos de resistencia, uno para verificar la calidad de las conexiones hasta 0.3 ohms y el otro para medición de aislamiento hasta 10 Megaohms (CAMI Research Inc, 1996-2015).

De forma nacional, también se han desarrollado trabajos con tópicos relacionados a este trabajo. El trabajo de García (2013) con el título de “Planeación y control del mantenimiento en la línea de producción KU-400”, plantea un programa de mantenimiento para la línea de producción KU-400, en donde se fabrican arneses eléctricos, y surge por la necesidad de evitar paros en líneas por fallas en algunos de los componentes de las máquinas de producción, en el cual es muy útil para saber más sobre el mantenimiento de los cables. En el ámbito internacional, Castañeda (2001) realizó un manual de control de calidad de conductores eléctricos, donde, describe algunas especificaciones necesarias para la fabricación de distintos tipos de cables, como el material de fabricación del conductor y el aislamiento.

Por otro lado, refiriéndose al prototipo que se desea desarrollar, se encontró en la literatura que existen herramientas tanto de software como de hardware que pueden servir de base para este proyecto. Por ejemplo, Villarroel et al. (2010) presentan un sistema MSF basado en Visual Studio, en el cual describen ventajas,

especificaciones y características de usos de este software en entornos prácticos. Angamarca et al. (2015) presentan un proyecto de control usando un microcontrolador de la familia Arduino. En este trabajo, se describen las ventajas y funcionalidades que este microcontrolador puede ofrecer como dispositivo hardware de control. También, Crespo (2015) presenta en su proyecto de tesis las bases y fundamentos de diferentes protocolos de comunicación, resaltando el RS-232 como uno de los más sencillos y adecuados para un gran cantidad de aplicaciones. Por lo tanto y en base a los antecedentes discutidos, se decide usar estas herramientas (Visual Studio, Arduino y protocolo RS-232) como base del prototipo a desarrollar. Es importante mencionar que estas tecnologías permiten reconfiguración y una fácil adaptación en software para atender las diferentes necesidades del prototipo dependiendo el proyecto a desarrollar.

1.2 Descripción del problema

Como se mencionó anteriormente los cables son de suma importancia, ya que cumplen con el propósito de poder transferir corriente eléctrica, así como, señales de información. Generalmente, los cables están formados por uno o varios hilos hechos de un material conductor (cobre o aluminio), suelen estar recubiertos de un material aislante para evitar el paso de la corriente hacia afuera del cable y evitar cortocircuito entre ellos.

A lo largo del tiempo se les ha dado diferentes usos ya sea para la conducción de energía eléctrica, transmisión de datos y comunicaciones (telefonía, televisión, etc.). Esto ha generado distintos tipos de cables dependiendo del propósito para el que sean necesarios. Como ya se mencionó anteriormente, existen distintos tipos de cables que son por mencionar algunos: cable coaxial, de par trenzado, de fibra óptica, USB y el cable plano flexible.

Sin embargo, todos los cables son susceptibles a distintos tipos de fallas, por ejemplo: fallas de cortocircuito, generadas por falla en el aislante que recubre el

cable que termina por ocasionar contacto con otros conductores y generar daños a los equipos o a la instalación; fallas en la continuidad, ocasionadas principalmente por la fatiga mecánica ocasionada por la manipulación frecuente del conductor, así como su mala conexión; y la falla por inversión de cables, referida a una conexión intercambiada entre dos o más conductores de un mismo cable.

Para la detección de las fallas ya mencionadas, se requiere un sistema de monitoreo que facilite el proceso de detección de fallas, sea de fácil manejo y además permita una interpretación sencilla de los resultados arrojados por el software. Además de estas características, el sistema desarrollado debe permitir su adaptabilidad en software y hardware para distintos tipos de conectores (tanto en número como en forma), lo cual representa una ventaja con respecto a equipos comerciales que no permiten estas modificaciones. También, el hecho de que el software sea desarrollado desde cero y con base a un software libre permite adaptar e integrar funcionalidades específicas o nuevas para mantener una constante actualización de características y necesidades. Sin embargo, para poder llevar a cabo este sistema, se deben resolver una serie de cuestiones tanto en software como en hardware. En primer lugar, una problemática que se presenta en el desarrollo de este tipo de sistemas es el número de entradas y salidas que son necesarias en la unidad de control o prueba debido a la cantidad tan variable de conductores (hilos) que puede tener un cable. Por ejemplo, considerando una tarjeta de control con una cantidad limitada de pines digitales que pueden ser usados tanto de entrada como de salida de señales, es necesario recurrir a elementos o componentes electrónicos que permitan complementar la cantidad de conexiones posibles para cumplir con el número requerido; es decir, desarrollar o plantear estrategias de circuitos electrónicos que permitan con pocos pines digitales controlar un número mucho mayor de señales digitales. De la misma manera, el software debe ser desarrollado de tal manera que pueda adaptarse a las modificaciones del hardware ya que si no perdería funcionalidad, lo cual es muy común en los equipos comerciales.

1.3 Justificación

Con la intención de proveer una solución para la detección de fallas de calidad en las pruebas de continuidad eléctrica, fallas de cortocircuito, y fallas de inversión de cables, al ser estas las más típicas, es necesario contar con equipos que permitan de una forma sencilla y rápida la verificación del funcionamiento de los cables.

Es importante mencionar que ya existen equipos comerciales que realizan y proveen soluciones a algunas de las necesidades arriba mencionadas. Sin embargo, estos equipos además de presentar en su mayoría precios elevados y con un conjunto limitado de características, presentan una topología de arquitectura cerrada no libre de acceso. Esto es, la integración, modificación y actualización de tareas distintas a las que fue diseñado, no es posible, incluso con el mismo proveedor debido a la poca versatilidad del equipo. Esta condición es muy limitativa y poco competitiva para el sector industrial cambiante, ya que continuamente se enfrentan al desarrollo de nuevos prototipos y equipos que requieren de diferentes cables, conductores, etc., lo cual hace obsoleto un equipo con características limitadas. Por lo tanto, el contar con un software y un hardware de arquitectura abierta, versátil a las modificaciones y que además sea económico y fácil de usar, permitirá ser una opción de mejora de tiempos en la verificación de fallas en cables-conductores. También, permitirá que el sistema pueda ser modificado y ampliado según las necesidades actuales del proyecto

1.4 Objetivos

1.4.1 Objetivo general

Diseñar y desarrollar un sistema software-hardware para un probador de fallas de cortocircuito, continuidad e inversión de cables en cables tipo plano-flexible.

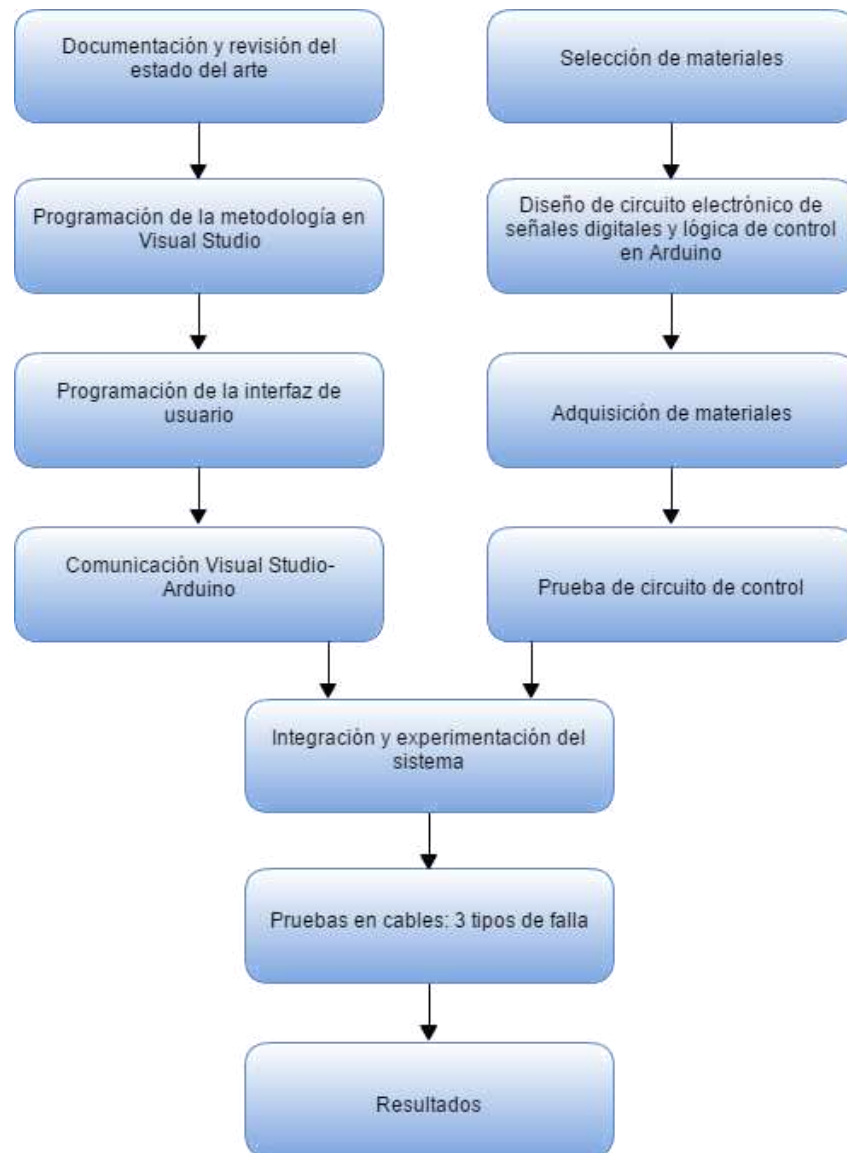
1.4.2 Objetivos particulares

- 1) Desarrollar un programa en Visual Studio para configuración y monitoreo de las pruebas.
- 2) Programar en Arduino la lógica y secuencia de las señales digitales requeridas para las pruebas de falla en cables tipo plano-flexible.
- 3) Desarrollar el circuito de control electrónico para el emisor y receptor de señales digitales.
- 4) Desarrollar un prototipo que integre el software y hardware desarrollados para pruebas de falla en cables tipo plano-flexible.
- 5) Probar el sistema con tres tipos de falla: cortocircuito, continuidad eléctrica e inversión de cables.

1.5: Planteamiento general

En esta sección, de manera general, se describirá el proceso que se llevará a cabo para el desarrollo del prototipo de pruebas en cables plano-flexible, además de una explicación de cada uno de los pasos que se muestran a continuación en la Fig. 1.3.

Basándose en este planteamiento, la interfaz de usuario, necesaria para este proyecto será desarrollada en Visual Studio, donde será una interfaz amigable e intuitiva para el usuario; además, será basada en lenguaje de programación C++ y de código abierto para modificaciones, adaptaciones y actualizaciones cuando sea necesario.



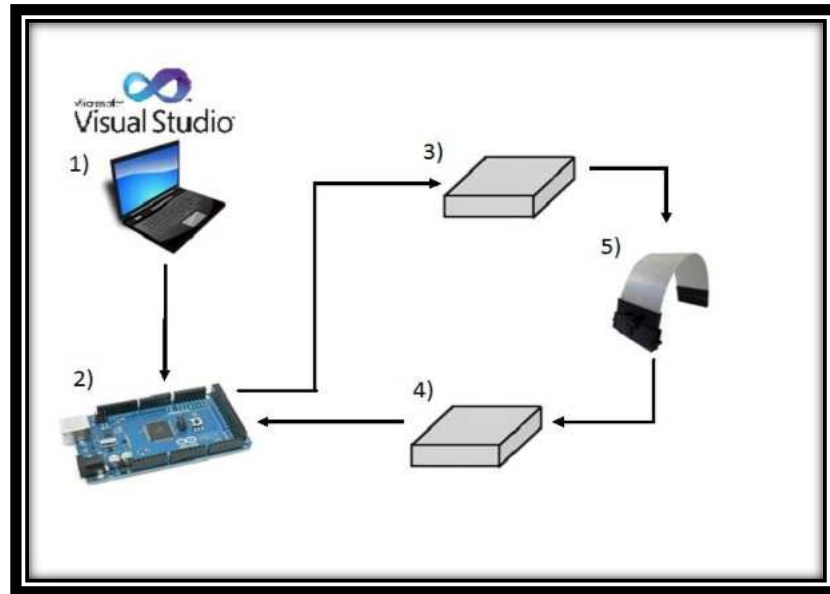
En el primer recuadro (Documentación y revisión del estado del arte) se comienza con la documentación bibliográfica sobre el tema de interés, también, se revisan términos relacionados con el trabajo a realizar como las distintas fallas en cables plano-flexibles y en cuales se va a enfocar este trabajo, además, de su importancia en la industria y algunos antecedentes relacionados al tema, incluyendo

trabajos con relación en Visual Studio, continuidad eléctrica, comunicación serial RS-232 y Arduino, donde, este microcontrolador por su facilidad de uso y bajo costo es el indicado para el desarrollo del software. En los dos siguientes recuadros (Programación de la metodología en Visual Studio y Programación de la interfaz de usuario) se desarrolla el software donde el usuario podrá conocer el resultado que fue enviado por el microcontrolador y así determinar si el cable está en buen estado o no. A continuación (Comunicación Visual Studio-Arduino), se hará la conexión entre el equipo y Arduino y poder obtener los resultados esperados de comunicación.

De forma paralela a los recuadros anteriores (Selección de materiales) se elegirán los materiales necesarios para la elaboración de este proyecto. El circuito electrónico de control será realizado con la finalidad de obtener el número de salidas y entradas necesarias que cumplan con las características de los cables utilizados en este proyecto y sin necesidad de ocupar demasiados pines en la tarjeta Arduino (Diseño de circuito electrónico de señales digitales y lógica de control en Arduino) y se adquirirá el material que fue seleccionado (Adquisición de materiales) en donde debido al mayor número de entradas comparado con otros modelos de la misma marca se seleccionará una tarjeta Arduino Mega y se harán las pruebas correspondientes para asegurarse de su funcionamiento (Prueba de circuito de control). Después, se integrará todo el sistema (Integración y experimentación del sistema), para posteriormente realizar las pruebas en los cables para realizar los tres tipos de pruebas, pruebas de continuidad, prueba de cortocircuito y prueba de inversión de cables (Pruebas en cables: 3 tipos de falla). Finalmente se documentarán los resultados obtenidos en estas pruebas (Resultados).

Es importante mencionar que el sistema desarrollado podrá operar de dos formas, una con modo PC y otro en modo portable (sin PC). En la primera se tendrá una mayor versatilidad de funciones mientras que la segunda solo serán las pruebas programadas pero sin la necesidad de contar con una PC (portable), desplegando la información en una pantalla LCD.

A continuación, se presentará el diagrama de conexión de cada uno de los componentes del sistema (Fig. 1.4), donde serán explicados brevemente.



- 1) PC: Se visualizará la respuesta obtenida en una interfaz de usuario realizada en Visual Studio.
- 2) Arduino: Es el microcontrolador que enviará un pulso hacia el receptor y a su vez recibirá una respuesta proveniente del circuito eléctrico de pruebas.
- 3) Emisor: En él será recibida la señal proveniente del microcontrolador y a su vez será transmitida hacia el cable para su prueba de falla.
- 4) Receptor: Es el circuito encargado de recibir la señal que pasó por el cable bajo prueba y enviarla hacia la tarjeta microcontroladora. .
- 5) Cable bajo prueba: Es el cable que será utilizado para saber si está en buen estado, en donde si el pulso eléctrico es capaz de circular a través de él. El cable se encuentra en buen estado o alguna condición de falla a determinar por el sistema.

En los siguientes capítulos se presenta la fundamentación teórica y tecnológica que se usa de base para este trabajo, la metodología seguida y el prototipo desarrollado, así como los resultados obtenidos.

Capítulo 2

Fundamentación teórica y tecnológica.

Fundamentación teórica y tecnológica.

En esta sección se describen los conceptos teóricos y tecnológicos usados en el presente trabajo con el objetivo de que los temas usados en la metodología sean visualizados de mejor manera.

2.1 Cables planos y sus aplicaciones

Cable FFC (cable plano flexible, del inglés *Flexible flat cable*) se refiere a cualquier variedad de cable electrónico que sea tanto plano como flexible y con múltiples conectores unidos a una superficie; sin embargo, el término FFC se refiere a un cable plano extremadamente delgado, a menudo es encontrado en diferentes dispositivos de alta densidad electrónica como computadoras, teléfonos celulares, cámaras y diversos dispositivos electrónicos.

El cable FFC consta de las siguientes especificaciones (FFC, 2018):

- Paso (*pitch*): es el espacio entre sus conductores, es decir, la distancia que hay desde el centro de un conductor hasta el centro de su conductor adyacente. Los cables plano flexible están disponibles con múltiples pasos, tales como 0.5 mm, 0.625 mm, 0.635 mm, 0.8 mm, 1.0 mm, 1.25 mm, 1.27 mm, 2.0 mm, 2.54 mm, donde, los más comunes son 0.5 mm y 1.0 mm.
- Tramo expuesto: tramo del contacto eléctrico que está expuesto al extremo del cable.
- Tensor: la mayor parte de los cables plano-flexibles tienen algún tipo de material adicional, ligado al lado opuesto del tramo expuesto del cable.

2.2 Tipos de fallas

En general, existen distintos tipos de fallas por las que un cable podría dejar de funcionar, pero en este caso solo se enfocará en las fallas por cortocircuito, por

circuito abierto e inversión de cables, las cuales son las demandadas por una de las problemáticas de la industria que se desea atender.

Un cortocircuito es un fenómeno eléctrico que ocurre cuando dos puntos donde existe una diferencia de tensiones tienen contacto entre sí y la magnitud de la corriente alcanza valores muy elevadas por encima del punto de falla, debido a una conexión por un circuito de baja impedancia. La magnitud de corriente de cortocircuito es mucho mayor que la corriente nominal o de carga que circula por el mismo. La Fig. 2.1 muestra un ejemplo de un conductor con falla de corto circuito.

Un cortocircuito puede ser de forma directa o indirecta. En el caso de un cortocircuito de forma directa, es cuando entran en contacto sin medios e intermediarios, es decir, una falla potencial-potencial o potencial-neutra. En cables planos puede ser provocado debido a que sus alambres internos se hayan cruzado o alguno de los conductores está teniendo contacto con otro conductor permitiendo que la corriente circule por otra parte donde no se desea; de forma indirecta, cuando existe un medio por el que pueda circular la corriente, por ejemplo, la carcasa del equipo o alguna barra metálica cercana, este tipo de fallas puede ocasionar desde que el equipo se queme y deje de funcionar, hasta incendios de gran magnitud.



Fig. 2.1 Falla en cables debido a un corto circuito.

Una falla por circuito abierto es cuando no hay flujo de corriente debido a una alta impedancia en el circuito, esto es ocasionado por el mal manejo o mala fabricación en cables o algún otro componente del circuito eléctrico, ocasionando

su fractura. Puede presentarse en el mismo conductor o a causa de una mala conexión en alguna terminal (ver Figs. 2.2 y 2.3). Algunas consecuencias por fallas en circuito abiertos son la pérdida de conexión con algunos componentes como sensores, leds, motores e incluso evitan que el equipo pueda encender.



Fig. 2.2 Falla en cable debido a desconexión de terminales en cables.



Fig. 2.3 Falla en cable debido a fracturas o cortes por el mal manejo

Una falla por inversión de cables ocurre cuando el extremo final de dos o más cables este cruzado (ver Fig. 2.4), es decir que el lugar donde está conectado el final de un conductor corresponde al de otro. Estos errores pueden provocar desde un mal funcionamiento del sistema hasta cortocircuitos que lo podrían dañar por completo.



Fig. 2.4 Cables invertidos.

2.3 Visual Studio y Windows forms application

Visual Studio es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C# y Visual C++ utilizan el mismo entorno de desarrollo integrado (IDE), que habilita el uso compartido de herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes utilizan las funciones de .NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios Web XML (Visual Studio, 2016). El logo de Visual Studio 2017 se muestra en la Fig. 2.5.



Fig. 2.5 Logo de Visual Studio, 2017.

Como los formularios son la unidad básica de la aplicación, es esencial realizar algunas consideraciones sobre su función y diseño. En última instancia, un formulario es una pizarra en blanco que el desarrollador mejora con controles para crear una interfaz de usuario y con código para manipular los datos. Para ello, Visual Studio proporciona un entorno IDE que ayuda a escribir código, así como un completo conjunto de controles escrito con NET Framework. Complementando la funcionalidad de estos controles con el código, puede desarrollar las soluciones que necesita de forma fácil y rápida (Visual Studio, 2016). En la Fig. 2.6 se muestra una vista general de la ventana principal de Visual Studio (*Windows forms application* en Visual C++).

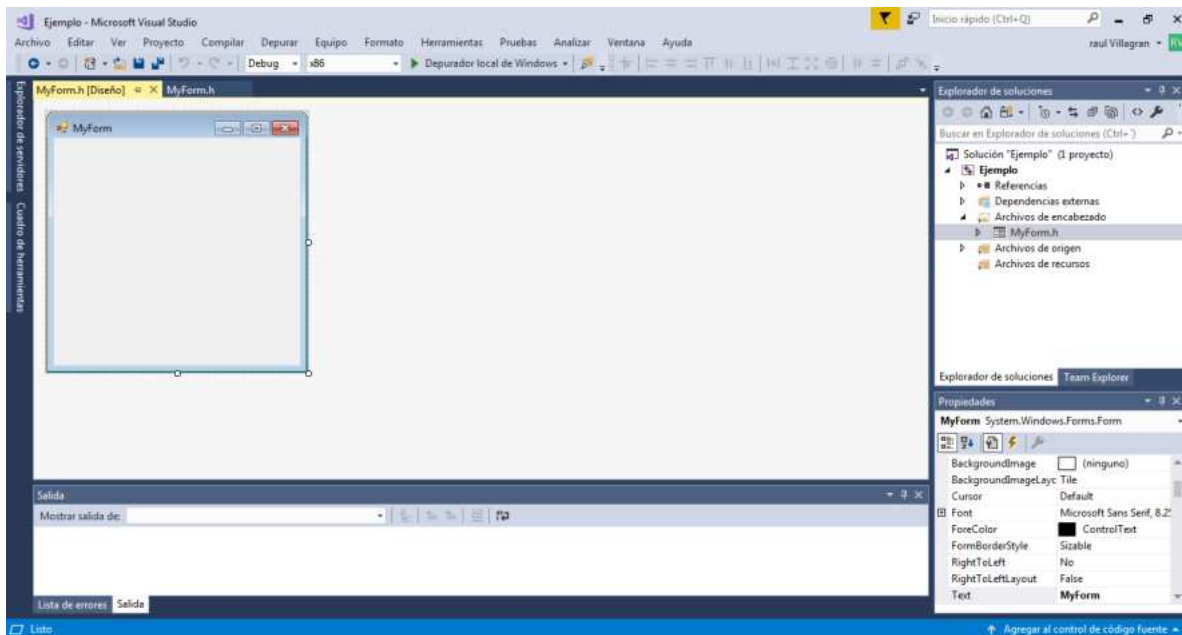
















Fig. 2.6 Windows form application en Visual Studio 2017.

2.3.1 Simbología y descripción de las herramientas de Windows Forms Application

A continuación, se dará una descripción breve de las herramientas útiles para el desarrollo de este proyecto (MSDN, 2018).

-  (Iniciar depuración): Compila y corre el programa.
-  (Guardar): Guarda las modificaciones hechas al programa.
-  (Cuadro de herramientas): En esta opción se encuentran las herramientas para el diseño de la interfaz de usuario.
-  Button (botón): desencadena un evento cuando un usuario hace clic.
-  Label (Etiqueta): proporciona información en tiempo de ejecución o texto descriptivo para un control.
-  RadioButton (RadioButton): Permite al usuario seleccionar una única opción de entre un grupo de opciones cuando están emparejadas con otros RadioButtons.
-  ProgressBar (Barra de progreso): muestra una barra que se va completando para indicar al usuario el progreso de una operación.
-  GroupBox (GroupBox): muestra un marco alrededor de un grupo de controles con un título opcional.
-  TextBox (TextBox): Permite al usuario especificar texto, así como funciones de edición de varias líneas y máscaras de caracteres para contraseñas.
-  TableLayoutPanel (TableLayoutPanel): Controla la presentación de sus componentes y los organiza en el formato de una tabla automáticamente.
-  SerialPort (SerialPort): representa un recurso de puerto serie.
-  (eventos): Le comunica a una aplicación que ha sucedido algo importante, los tipos de evento cambian según sea la herramienta utilizada.
- DataReceived: este evento se desencadena cada vez que se reciben datos del SerialPort.
-  OpenFileDialog (OpenFileDialog): muestra un cuadro de dialogo donde se pide al usuario que abra un archivo.
-  SaveFileDialog (SaveFileDialog): muestra un cuadro de dialogo donde se pide al usuario que seleccione una ubicación para guardar un archivo.

2.4 Tecnología Arduino

Arduino es una plataforma de prototipos de electrónica de código abierto (*open-source*) basada en hardware y software flexibles y fáciles de usar. Está pensado para artistas, diseñadores, como hobby y para cualquiera interesado en crear objetos o entornos interactivos. Arduino puede sentir el entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros artefactos. El microcontrolador de la placa se programa usando el *Arduino Programming Language* (basado en *Wiring*) y el *Arduino Development Environment* (basado en *Processing*) (Arduino, 2018).

Arduino cuenta con diferentes modelos en el mercado, como Arduino uno, Arduino 101, Arduino micro, Arduino Mega, entre varios modelos más, en donde este último mencionado será utilizado para la realización del probador de continuidad. En la Fig. 2.7 se observa una tarjeta Arduino Mega.



Fig. 2.7 Arduino Mega2560.

El software de Arduino es totalmente gratuito y se puede descargar desde la página oficial (<https://www.arduino.cc/en/Main/Software>) y está disponible para diferentes sistemas operativos.









Fig. 2.8 Ventana principal del software de Arduino.

En la Fig. 2.8 se muestra la vista general de la ventana principal de Arduino, consiste en varios iconos donde cada uno realiza una acción específica y una estructura de programación.

2.4.1 Simbología y descripción de las herramientas de Arduino

En la ventana de la Fig. 2.8 se pueden observar las siguientes herramientas para las cuales se describe su funcionalidad.

-  (Verificar): compila el código y detecta errores.
-  (Subir): Compila el código y lo sube a la tarjeta Arduino.
-  (Nuevo): Crea un archivo nuevo.
-  (Abrir): Abre un archivo ya existente.
-  (Salvar): Guarda las modificaciones realizadas al código.
-  (Monitor serie): Muestra en una ventana los datos enviados por el microcontrolador.

2.4.2 Comandos de programación de Arduino

La estructura básica del lenguaje de programación es bastante simple y se compone al menos de dos partes. Estas dos partes encierran bloques que contienen declaraciones, estamentos o instrucciones.

- *setup()*

```
void setup() {  
  // put your setup code here, to run once:  
}
```

Fig. 2.9 void setup.

La función `setup ()` mostrada en la Fig. 2.9 se invoca una vez cuando el programa empieza. Se utiliza para inicializar los modos de trabajo de los pines, o el puerto serie. Debe ser incluido en un programa, aunque no haya declaración que ejecutar.

- *loop()*

```
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

Fig. 2.10 loop ().

La función `loop ()`, mostrada en la Fig. 2.10, se ejecuta de forma cíclica, lo que posibilita que el programa este respondiendo continuamente ante los eventos que se produzcan en la tarjeta.

Para poder cumplir con su función de controlar, Arduino utiliza distintas funciones. Son instrucciones que permiten enviar y recibir señales. Algunas de las funciones son (Arduino, 2018):

- *PinMode* (pin, mode): se utiliza para declarar cierto pin como pin de entrada o salida, *INPUT* (entrada) y *OUTPUT* (salida).
- *digitalRead* (pin): Da el valor de una señal recibida por el pin, dando como respuesta *HIGH* (alto) si se recibió un pulso con valor de uno o *LOW* (bajo) si la respuesta es cero.
- *digitalWrite* (pin, value): Envía un valor de salida ya sea *HIGH* o *LOW* a un pin ya declarado previamente como salida.
- *analogRead* (pin): Recibe y lee el valor de una señal analógica con resolución de 10 bits, esta función solo es aplicable para los pines marcados como analógicos.
- *analogWrite*: envía una señal analógica por medio del procedimiento de modulación de ancho de pulso (PWM, del inglés *pulse width modulation*).
- *Serial.begin* (rate): Es utilizada para abrir un puerto serial y fija la velocidad en baudios para transmitir los datos, la velocidad típica para la comunicación con la computadora es de 9600 baudios.
- *Serial.Read* (): lee un carácter desde el puerto en serie, devolviendo el byte desde el puerto serie, en caso de no haber recibido ningún byte entrega el valor de -1.
- *Serial.write* (): Envía caracteres hacia el puerto en serie.
- *Serial.available* (): obtiene el número de bytes disponibles para su lectura desde el puerto serie.
- *Serial.write* (): Envía caracteres hacia el puerto en serie.

2.5 Registro de corrimiento serie-paralelo y paralelo-serie

Debido a la cantidad limitada de pines de conexión que incluye la tarjeta Arduino es necesario disponer de un elemento que sea capaz de aumentar el número de conexiones de entrada y salida requeridas en la tarjeta. Como solución a esta problemática, para las entradas se utilizó un registro de cambio de paralelo a serial donde puede obtenerse información de 8 entradas digitales más utilizando solamente tres pines del arduino. El elemento electrónico utilizado para la

recolección de información en la entrada es el registro de cambio de paralelo a serial CD4021B. Es capaz de recolectar la información de hasta 8 entradas conectadas al mismo tiempo; esto es llamado entrada asíncrona en paralelo. Las ocho salidas del registro de cambio son enviadas en forma de pulsos en alto y en bajo (*HIGH* y *LOW*) por medio de un pin de entrada conectado al arduino referido como *datapin*. Los datos enviados a través del *datapin* se conocen como “Salida serial síncrona”. El pin de reloj se encarga de mantener una comunicación síncrona entre el registro de cambio y el arduino, cada vez que el arduino cambia el valor de la señal de reloj de alto a bajo el registro de cambio modifica el valor del pin de salida serial. Finalmente, el *latchpin* se encarga de cada vez que detecte una transición de la señal de alto a bajo el registro de cambio lee las ocho entradas en paralelo en caso de detectar la señal en alto, en cambio si la señal está en bajo, lee la señal de reloj y pasa la información serialmente. A continuación, la Fig. 2.11 muestra el diagrama del CD4021 y la Tabla 1 su descripción.

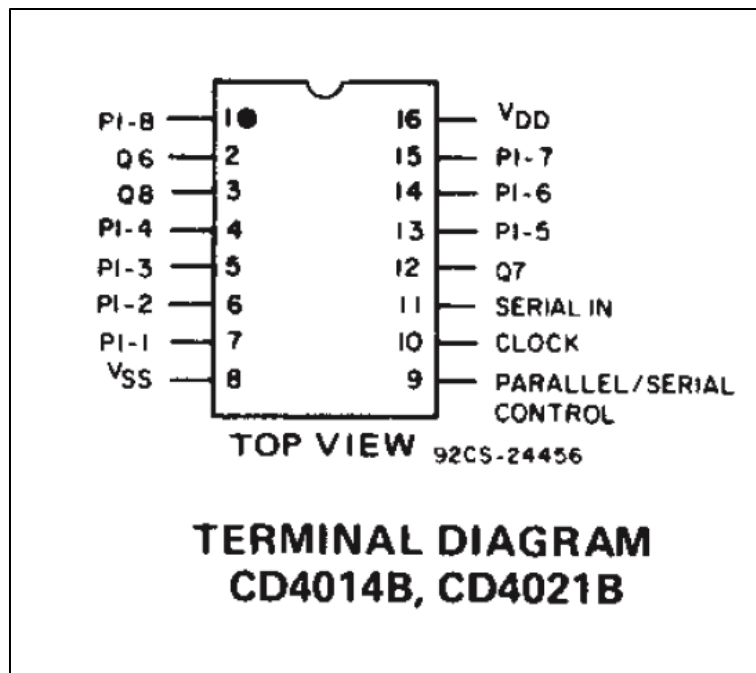


Fig. 2.11 Diagrama de terminales CD4021.

Tabla 1 Tabla de funciones de pines para el CD4021.

Pin	Función
1, 4-7, 13-15	Entradas en paralelo
2, 12, 3	Pines de salida serial de diferentes pasos en secuencia
8	GND (Tierra)
9	Control en paralelo/serial (latch pin)
10	Registro de cambio pin de reloj
6	Sincronización de lectura de datos
11	Entrada de datos serial
16	Alimentación de voltaje DC

Al igual que el circuito CD4021, también existen circuitos de desplazamiento para cuando sea necesario extender el número de pines de salida, este elemento es el 74HC959. A continuación, la Fig. 2.12 muestra el diagrama del 74HC959 y la Tabla 2 su descripción.

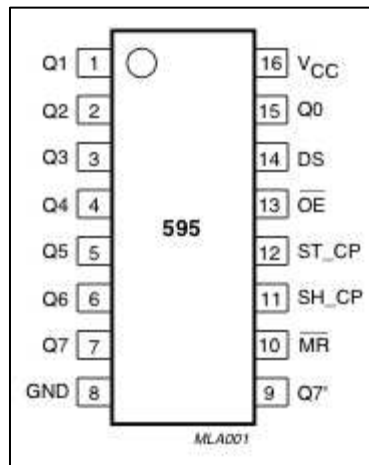


Fig. 2.12 Diagrama de terminales 74HC959.

Tabla 2 Funciones de pines para el 74HC959.

Pin	Función
1-7, 15	Pines de salida
8	GND (Tierra)
9	Salida serial
10	MR, activo en bajo
11	Pin de reloj de registro de cambio
12	Registro de almacenamiento del pin de reloj (latch pin)
13	Habilitar salida, activo en bajo
14	Entrada de datos en serie
16	Alimentación de voltaje DC

Las Figs. 2.13 y 2.14 se muestran los diagramas de conexión en cascada de los multiplexores y de los circuitos de desplazamiento, respectivamente. En la figura 2.13 se muestran los circuitos CD4021 en cascada, en donde se observa que están conectados entre sí por medio de los pines O7 (3) hacia el pin DS (11) de su consecutivo y que a su vez el pin O7 (3) se comunica hacia la entrada de dataPin del arduino en el primer CD4021 del arreglo, los pines PL (9) están conectados hacia la entrada de Latch, mientras que finalmente el pin CP (10) termina en la entrada clockPin, señal de reloj. De esta forma, los datos en paralelo de cada multiplexor son enviados por un solo pin (dataPin) hacia la tarjeta de recepción; por lo tanto, el número de entradas se puede ampliar bastante y el número de pines del microcontrolador no cambiaría. No obstante aspectos como cantidad de corriente y velocidad deben ser tomados en cuenta.

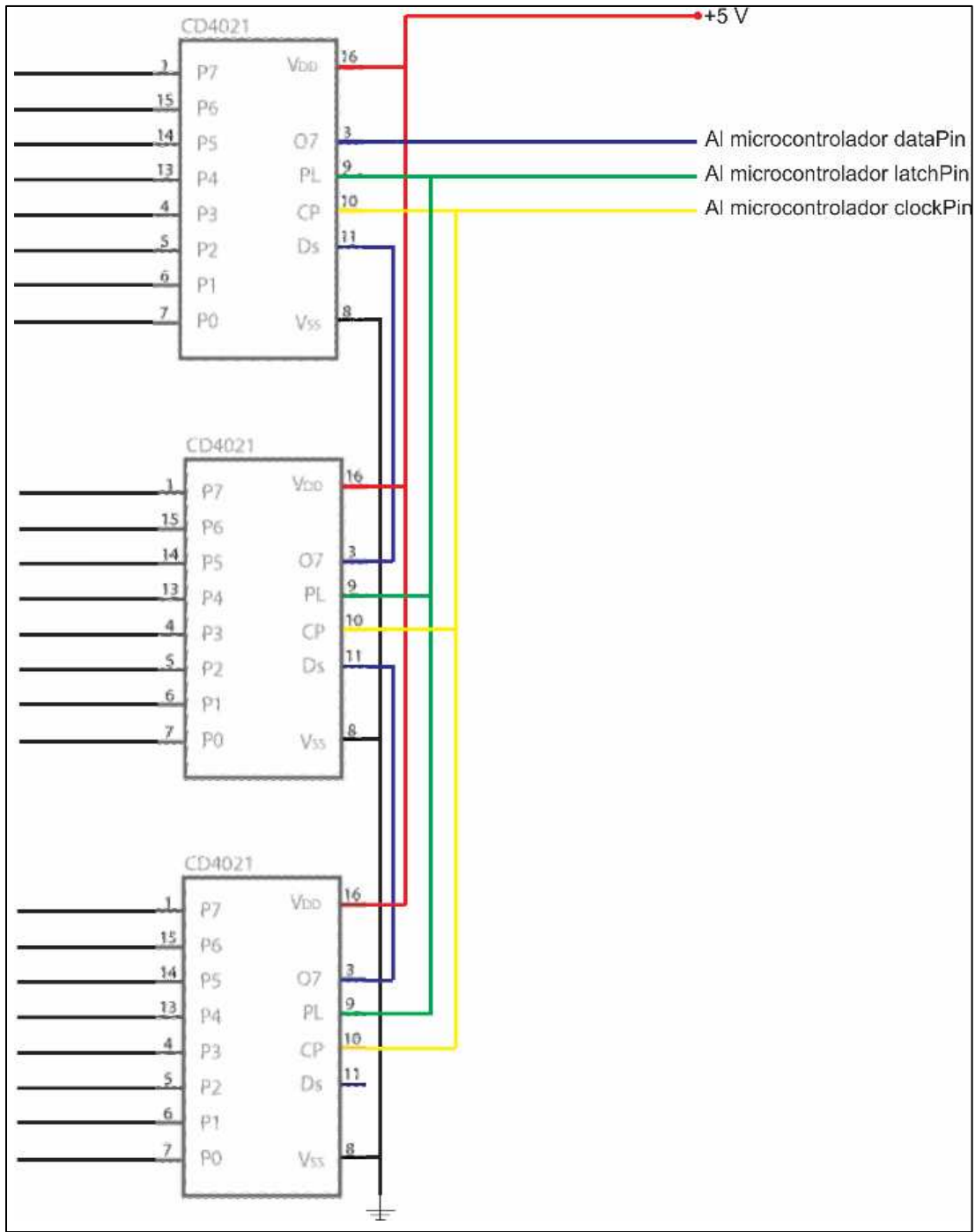


Fig. 2.13 Multiplexores en cascada.

Al igual que en las conexiones para los CD4021 en la Fig. 2.14 se muestra el diagrama de conexión de los circuitos multiplexores 74HC959. Se puede apreciar su conexión entre cada uno de ellos por medio de los pines Q7S hacia el pin DS de

su consecutivo. Los pines de salida hacia la tarjeta arduino de latchPin, clockPin y dataPin son comunicados por medio de los pines STCP (12), SHCP (11) y DS (14) únicamente en el primer 74HC959, correspondientes. Con propósito de simplicidad las salidas Q1-Q7 son conectadas a una barra de salidas indicando que de ahí se pueden tomar los datos en forma paralela.

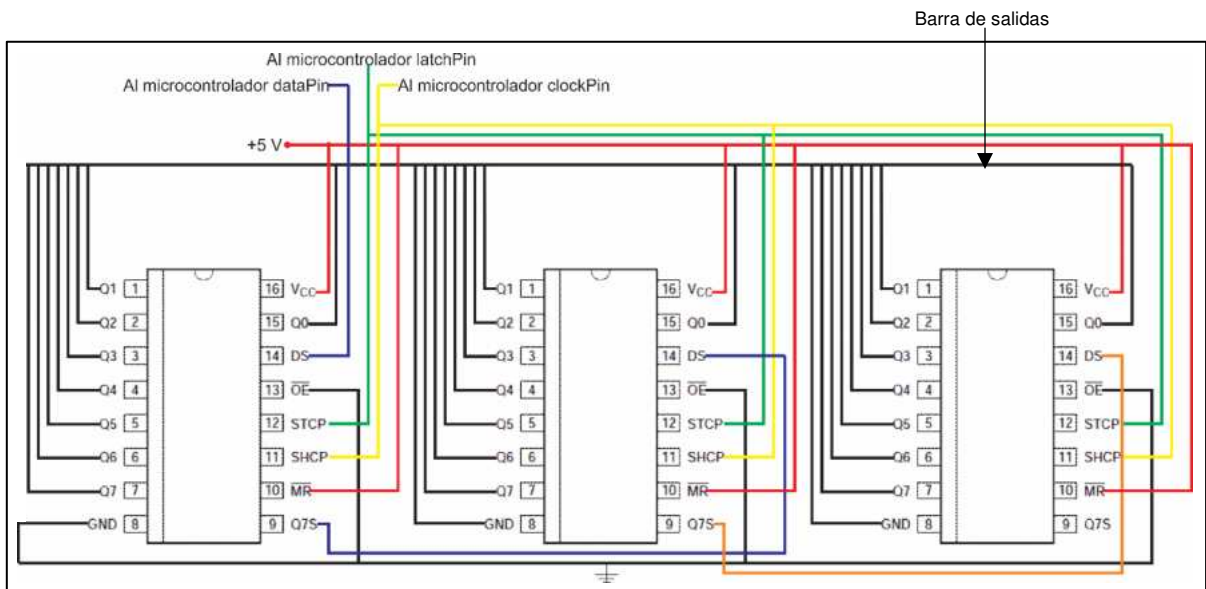


Fig. 2.14. Registro de cambio en cascada.

2.6 Display LCD

Para poder visualizar los resultados obtenidos cuando se efectúen pruebas en modo portable, fue necesaria la instalación de un display LCD 16x2. Por medio de esta pantalla se muestra el progreso del análisis, así como las fallas obtenidas del cable bajo prueba y cuando el análisis haya finalizado. A continuación, la Fig. 2.15 muestra un diagrama de los pines que están conformados en el display y la Tabla 3 su descripción.

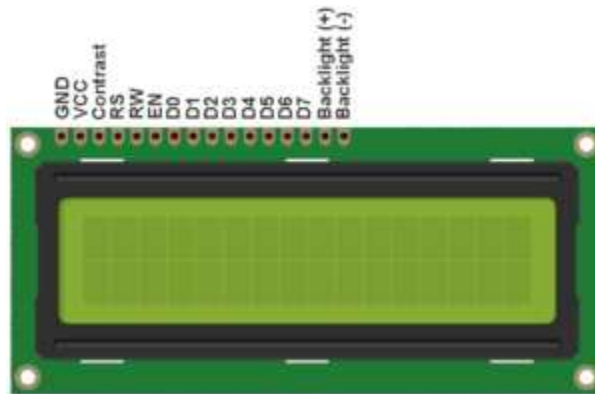


Fig. 2.15 Diagrama de pines LCD 16x2.

Tabla 3 Funciones de los pines del display LCD 16x2.

Pin	Función
1	GND (Tierra)
2	Entrada 5V
3	Control de contraste de pantalla
4	RS- Selector entre comandos y datos
5	RW- Escritura y lectura de comandos y datos
6	Sincronización de lectura de datos
7-14	Pines de datos de 8-bit
15	Alimentación luz de fondo (5V)
16	GND (Tierra) luz de fondo (0V)

Capítulo 3

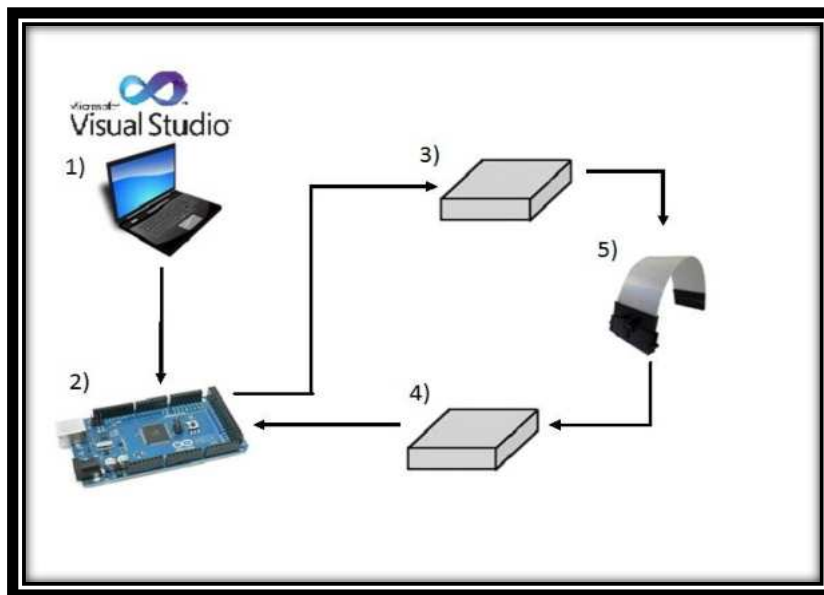
Metodología

Metodología

En este capítulo se describe el desarrollo del prototipo incluyendo el diseño base y pruebas preliminares (testeo en tarjetas de prototipado electrónico), diseño final exterior con cada una de sus secciones de las que consiste y los elementos por los que se conforma, los circuitos internos y su diseño, así como el principio de funcionamiento de la interfaz de usuario.

3.1 Diseño de prototipo

Antes de comenzar a describir el desarrollo del sistema se debe recordar el diagrama general de los componentes presentado en la Fig. 1.4, por conveniencia se vuelve a presentar en la Fig. 3.1. Donde los elementos 3 y 4 son bloques de circuitos de registro de desplazamiento y circuitos multiplexores (ver capítulo 2), respectivamente.



Un circuito a bloques más detallado de estas conexiones se muestra en la Fig. 3.2, donde se aprecia el número de componentes usados. En este caso se usan 5 registros de desplazamiento ($5 \times 8 = 40$ salidas) y 5 multiplexores ($5 \times 8 = 40$ entradas).

El área en gris representa el cable bajo prueba. Es conveniente mencionar que el cable de 40 pines es considerado en este trabajo como el más grande pero este puede ser expandido. Al ser la tarjeta de 40 pines, esta puede dar soporte a cables con un menor número de componentes. Cada pin digital cuenta con una resistencia tanto para limitar corriente como para mantener el estado lógico.

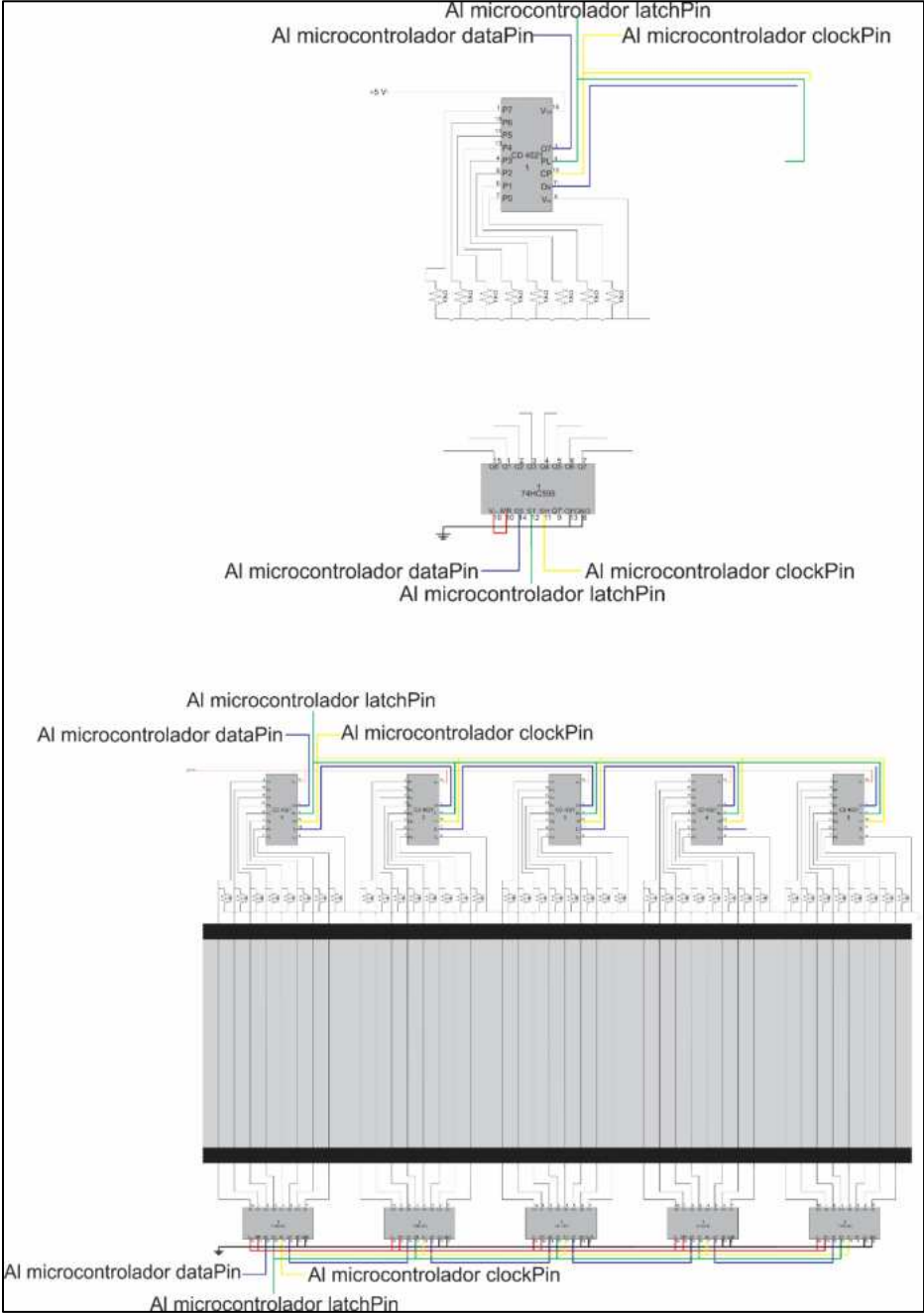


Fig. 3.2: Circuitos de acondicionamiento junto con cable bajo prueba.

Como se mencionó, el diseño del circuito fue probado antes de pasar a la parte de PCB. En este sentido, se realizaron pruebas a los circuitos ya armados, para poder asegurar su correcto funcionamiento antes de hacer las pruebas y así permitir realizarlas sin fallas en el hardware del prototipo final. La Fig. 3.3 muestra el circuito armado para pruebas manuales, es decir un botón mandará el pulso a través de los circuitos y el led indicará si este fue recibido en la posición correcta. En esta figura también se destaca cada uno de los componentes del circuito de control de pruebas para su identificación de cada uno: *push buttons* (interruptores de contacto), multiplexores, leds, registros de cambio (*shift registers*) y la tarjeta Arduino. La función de cada uno de estos componentes ya ha sido mencionada previamente.

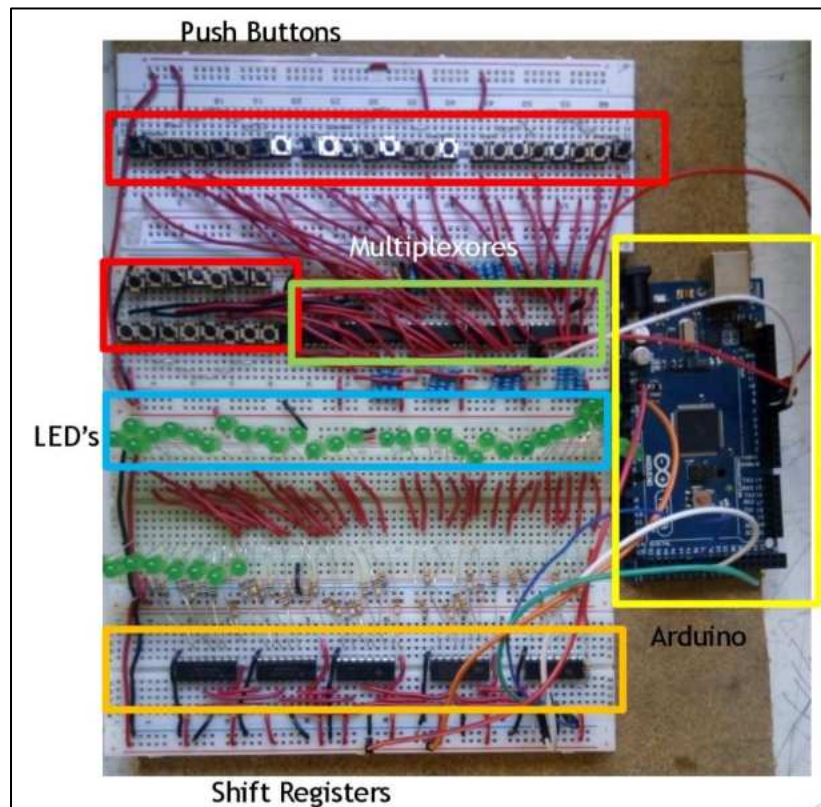


Fig. 3.3 Ubicación de cada uno de los circuitos.

Para comprobar el correcto funcionamiento de los circuitos, en un segundo paso, se creó un programa de prueba en Arduino, donde solo es necesario introducir el número de salida al pin al que se desea enviar la señal, permitiendo el paso de la señal a través del cable sometido bajo prueba para finalmente poder recibir la señal de respuesta. Este método debe de aplicar para los tres tipos de prueba requeridos. En la Fig. 3.4 se muestra el ejemplo aplicado para los multiplexores.

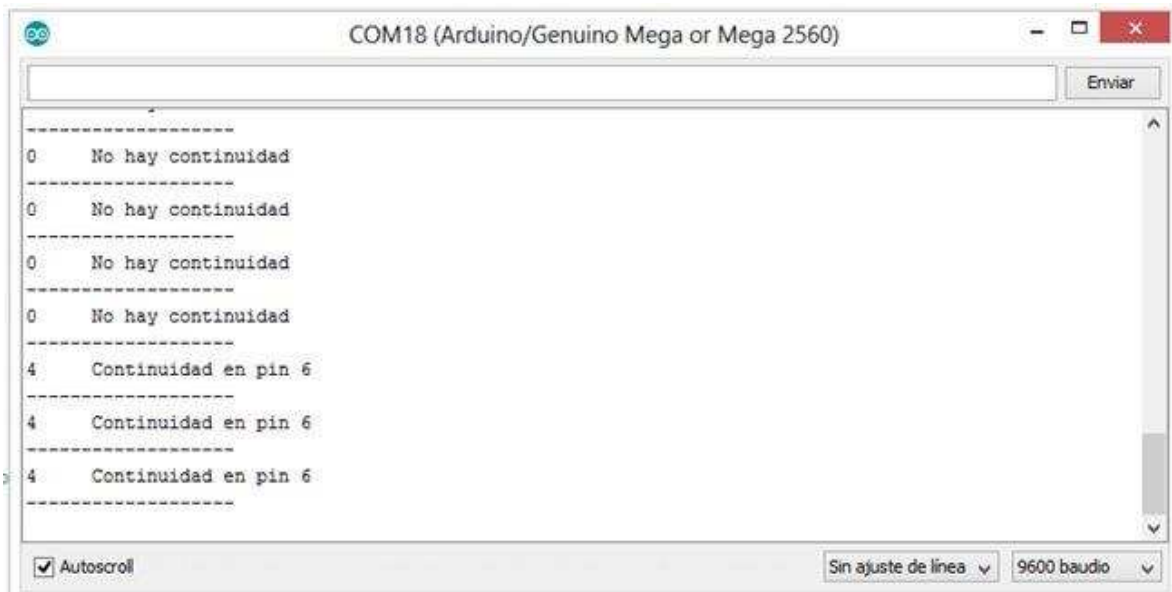
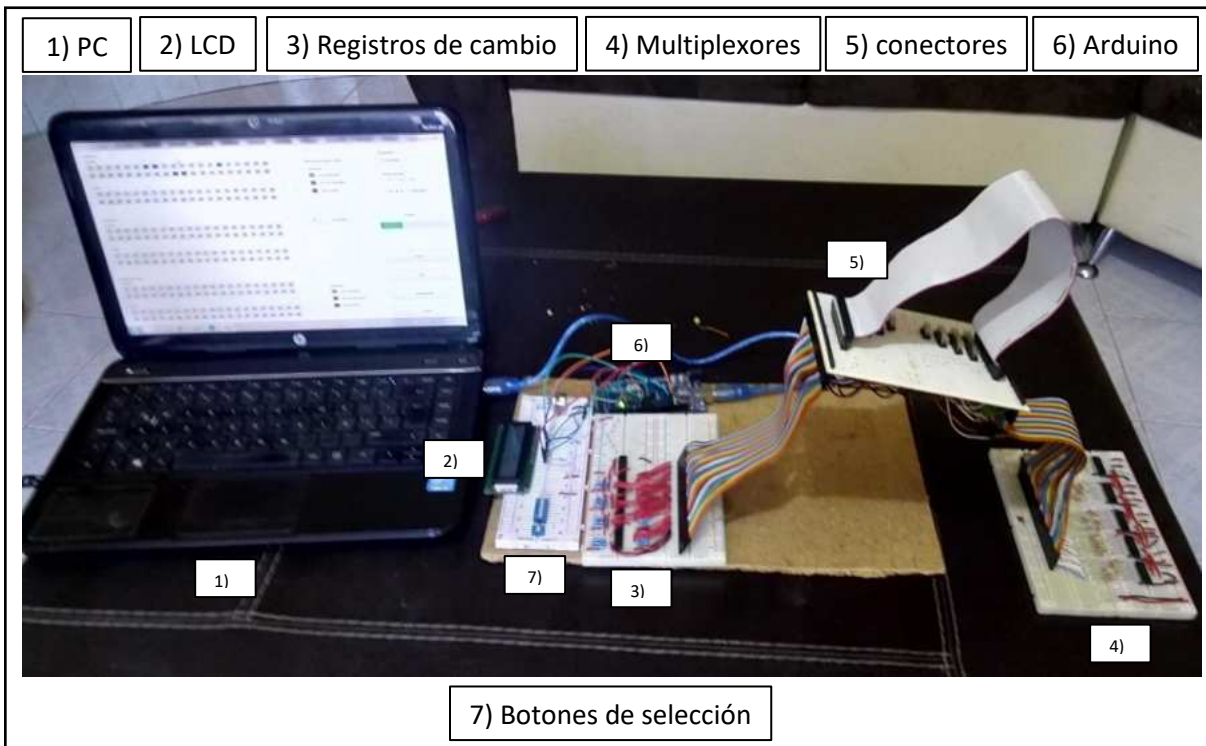


Fig. 3.4 Respuesta recibida por el programa y mostrada en pantalla la ubicación del botón presionado.

Una vez que se validó mediante software de Arduino, se procedió a integrar todo el sistema como se muestra en la Fig. 3.5. Ahí, se observa el ensamble de los circuitos. Se puede apreciar la conexión de los circuitos de control, display y Arduino ya ensamblados, incluyendo el cable plano-flexible que será sometido bajo las pruebas eléctricas que realizará el prototipo final. Cabe recordar que la base ha sido fabricada para distintos tamaños de cables, siendo el cable de 20x2 conductores el más grande y el de 4x2 el más pequeño, tal y como se muestra en la Fig. 3.6.



En la Fig. 3.7 (igual que la Fig. 3.6), los conectores de un menor número de pines están puenteados en la misma posición de los de un mayor número de pines, así el cableado/ruteo es mucho más simple; sin embargo, esto acarrea la desventaja de no poder hacer pruebas de más de un solo cable.

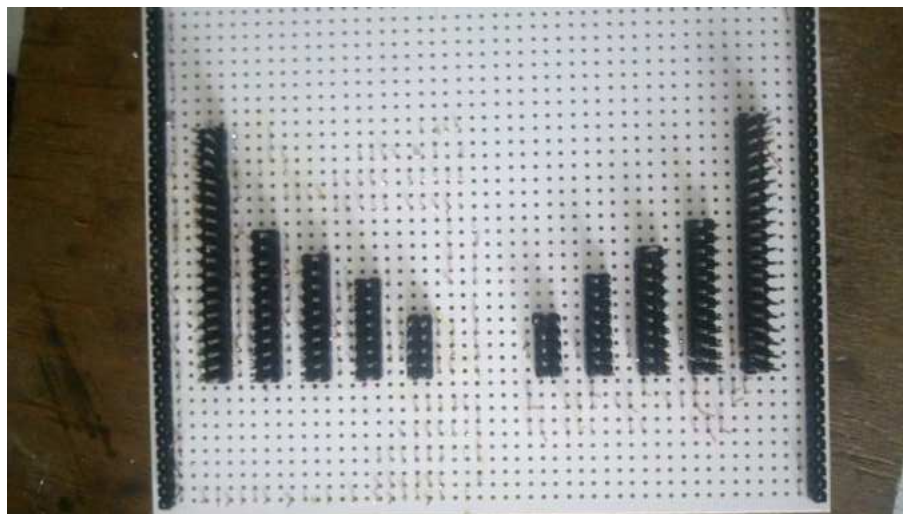


Fig. 3.6 Base de pines para conectar cables.



Fig. 3.7 Cable plano-flexible montado sobre la base de pines.

Por otro lado y sabiendo que el sistema debe tener modo portable (sin PC), es necesario un display que muestre información concisa acerca de la prueba y operación del sistema. La Fig. 3.8 muestra el circuito junto con el display ya instalado, su funcionamiento ya se puede apreciar con el texto “Probador de cables planos” en su pantalla.



Fig. 3.8 Pruebas de funcionamiento del display LCD.

Una vez que se validó todo lo anterior, se continúa con el desarrollo del prototipo final. Para esto, inicialmente se realizó el diseño del prototipo utilizando un software de dibujo en CAD (Inventor). El prototipo resultante se muestra en la Fig. 3.9.

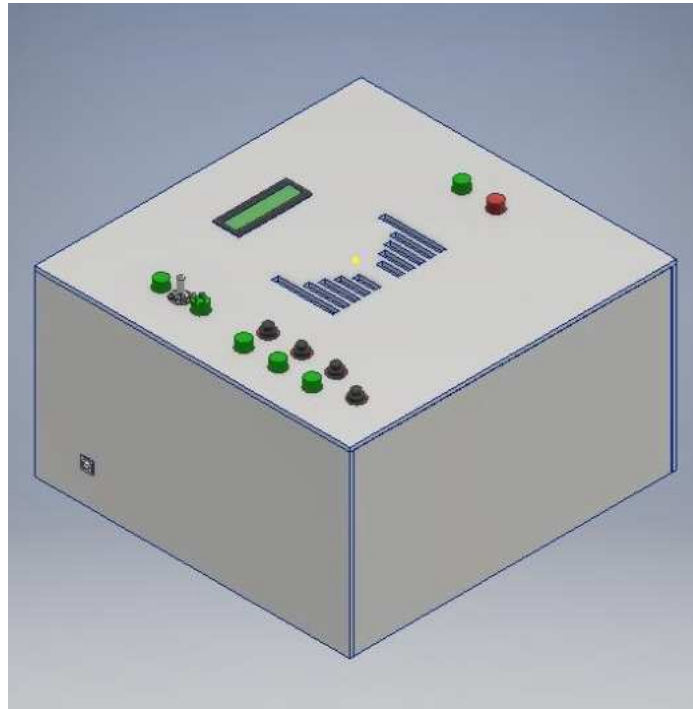


Fig. 3.9 Diseño de prototipo.

Este prototipo funcionará por medio de la interfaz en PC, pero además funciona con una versión portable en donde los resultados serán mostrados en una pantalla LCD. La sección de modo de operación será por medio de un botón, además de tener disponible las configuraciones para el número de pines y leds indicadores de encendido y de tipo de prueba. A continuación, se muestra el diseño final de cada una de las caras del prototipo incluyendo el cable de alimentación y cada uno de sus elementos (Fig. 3.10).

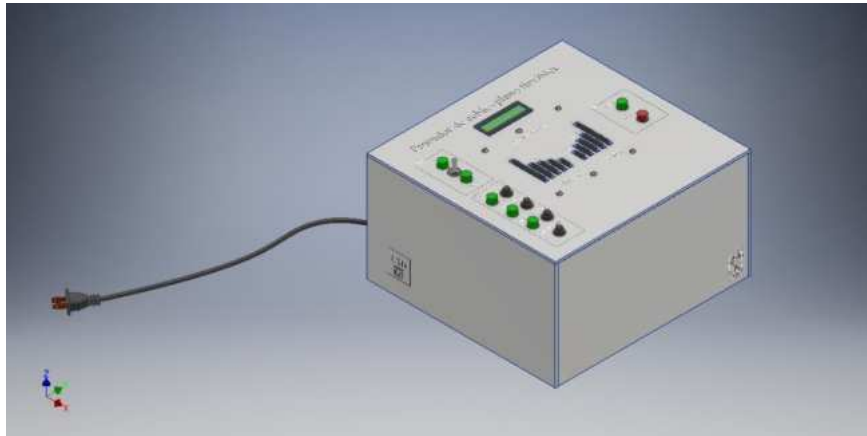


Fig. 3.10 Prototipo con cable de alimentación.

En la cara superior mostrada en la Fig. 3.11, se ubica el tipo de configuración a elegir por el usuario, así como el display LCD y los pines de conexión de entrada y salida para el cable sometido a prueba.

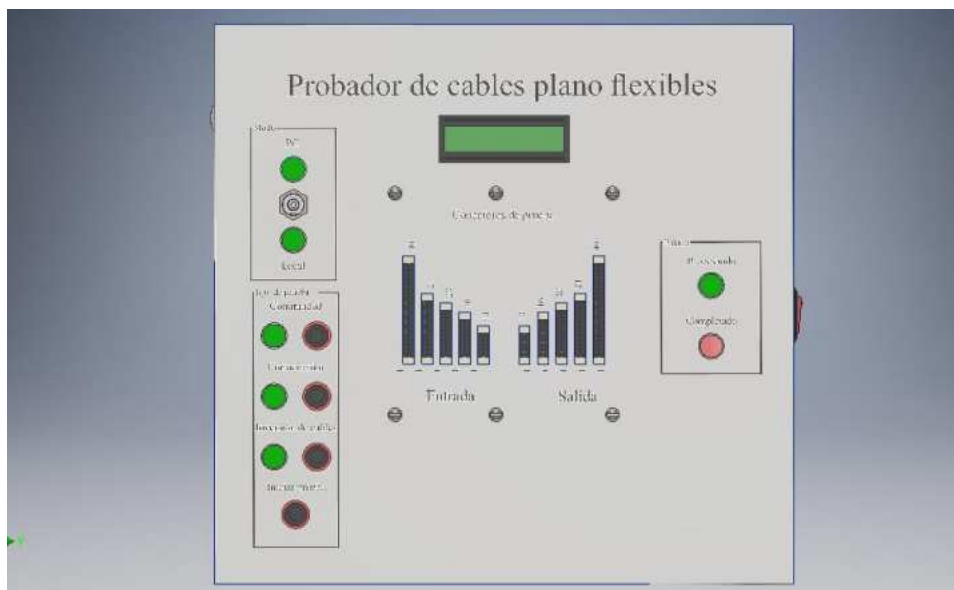


Fig. 3.11 Vista Superior.

La tapa superior se divide en diferentes secciones de acuerdo a la configuración a la que pertenece. A continuación, dichas secciones son explicadas más a detalle.

Del lado izquierdo superior, en la Fig. 3.11, se encuentra la sección “Modo” que incluye el switch de selección de configuración junto con los indicadores luminosos que facilitan la visualización de la posición en la que se encuentra el switch. Esta selección de modo establece si las pruebas serán realizadas de modo conectado a una PC o si se va a usar de modo Local, es decir de forma portable sin necesidad de tener una computadora (Fig. 3.12).

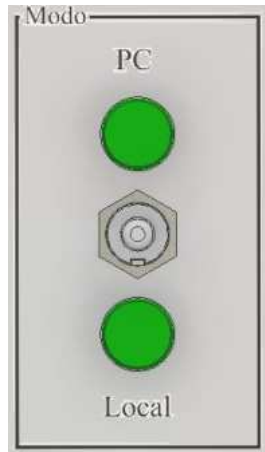


Fig. 3.12 Vista previa de la sección “Modo”.

En el lado izquierdo inferior se muestra la sección “Tipo de prueba”, mostrada en la Fig. 3.13. Esta sección consiste en los botones de selección de los tres diferentes tipos de pruebas y el botón para iniciar la prueba (tomando en cuenta que esta opción solamente funciona con la configuración en Local). En caso de estar en modo PC la selección se hace por medio de la interfaz de usuario. Si se tiene una conexión hacia la PC se utilizará la comunicación Arduino-PC mediante el cable USB. Por medio de esta configuración será posible poder acceder a las opciones que incluye la interfaz del usuario.



Fig. 3.13 Vista previa de la sección "Tipo de prueba".

La siguiente sección se ubica en el lado izquierdo, la sección "Estado", presentada en la Fig. 3.14, muestra la fase en la que se encuentra el análisis. El indicador "Procesando" se enciende cuando el software está obteniendo la información del estado del cable. Por otro lado, tenemos el indicador de "Completado", este indicador se enciende una vez que la prueba haya sido terminada y se haya obtenido la información requerida para generar los resultados de la prueba.



Fig. 3.14 Vista previa de la sección "Estado".

Finalmente, en la Fig. 3.15 se muestra la sección central, donde se alojan los conectores para los cables, haciendo indicación de cuales son de entrada y de salida e incluye la señalización del pin inicial y el final. Dentro de esta sección tambien se encuentra el display LCD, donde su función será descrita más adelante.

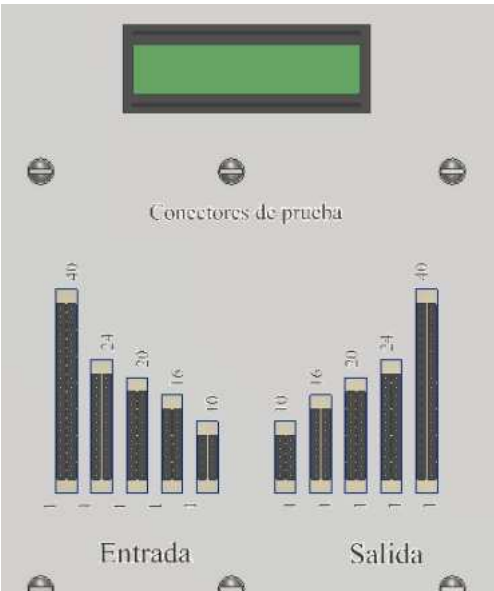


Fig. 3.15 Vista previa de la sección "Central".

Cabe mencionar que la cara posterior de la tapa superior en la Fig. 3.11 se pueden apreciar las salidas de los pines de salida de cada uno de los elementos que conforman esta cara y el soporte fabricado para sostener la placa de conexión de los cables tal y como se muestra en la Fig. 3.16; este soporte será retomado más adelante.

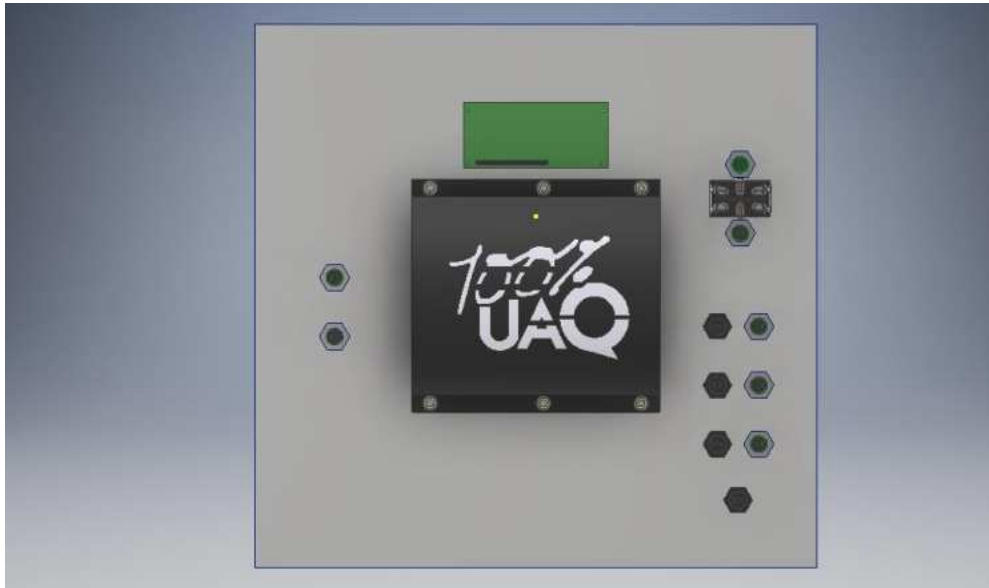


Fig. 3.16 Vista trasera de tapa superior.

Para identificar las caras laterales y así evitar confusiones en su descripción, se nombró a cada una con las etiquetas A, B, C y D.

En la cara A, Fig. 3.17, se muestra el interruptor de encendido y apagado. Este botón permite la alimentación por medio del modo portable al ser conectado a una fuente de energía.

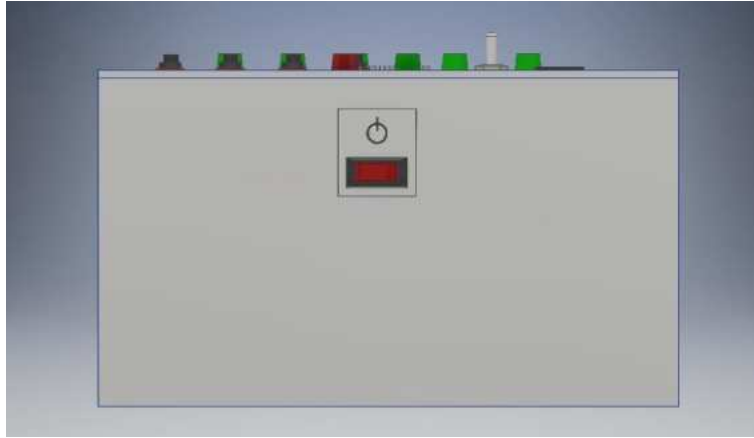


Fig. 3.17 Vista a la cara A.

En la cara que corresponde a la letra B, Fig. 3.18, solo se muestra el escudo de la Universidad Autónoma de Querétaro, mientras que en la cara C, mostrada en la Fig. 3.19, está constituida por la salida USB que establece la comunicación entre la PC hacia la tarjeta Arduino y viceversa. El cable de alimentación está ubicado sobre la cara D, presentada en la Fig. 3.20.



Fig. 3.18 Vista a la cara B.

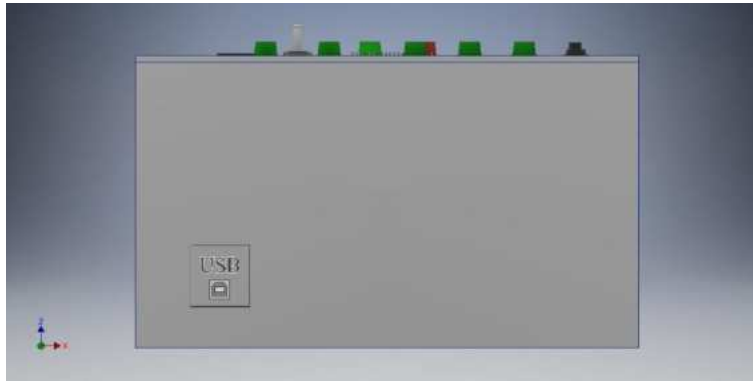


Fig. 3.19 Vista a la cara C.



Fig. 3.20 Vista a la cara D.

En la Fig. 3.21 se muestra el corte transversal donde muestra cada una de las partes internas del modelo, donde los dispositivos internos son descritos a detalle en las siguiente subsecciones. Al igual que en la figura previa, en la Fig. 3.22 se puede apreciar el mismo interior visto desde otro ángulo.

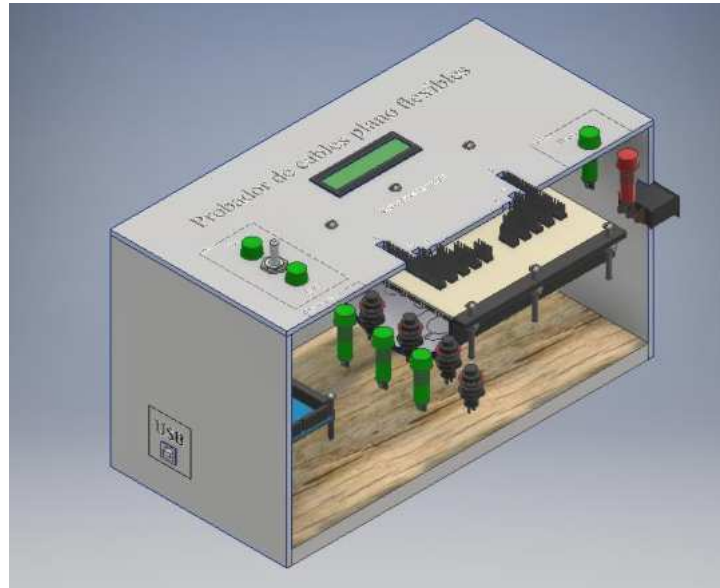


Fig. 3.21 Corte transversal.

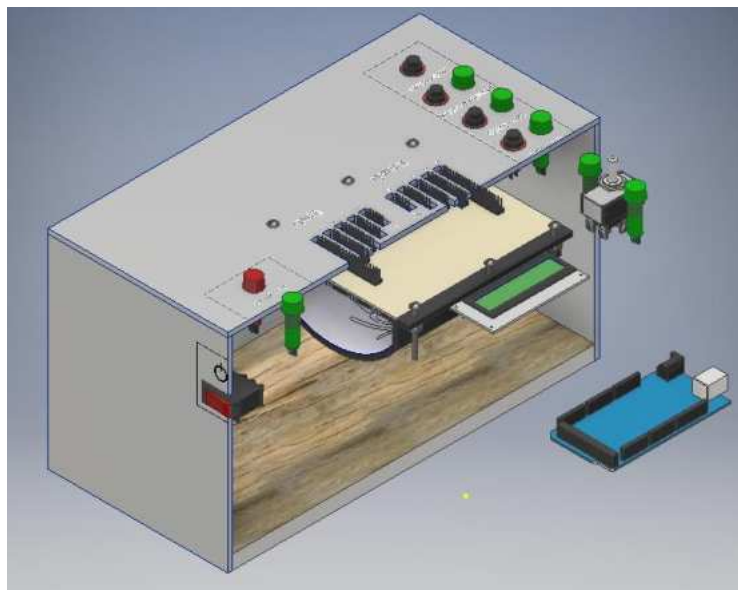


Fig. 3.22 Corte transversal visto desde otro ángulo.

Para concluir con el diseño exterior del prototipo se muestra el resultado final en la Fig. 3.23, incluyendo su conexión hacia la PC.

Cada uno de estos componentes se complementan con los circuitos internos, cuya función es establecer la comunicación entre cada uno de los elementos eléctricos y de control que conforman el proyecto final.

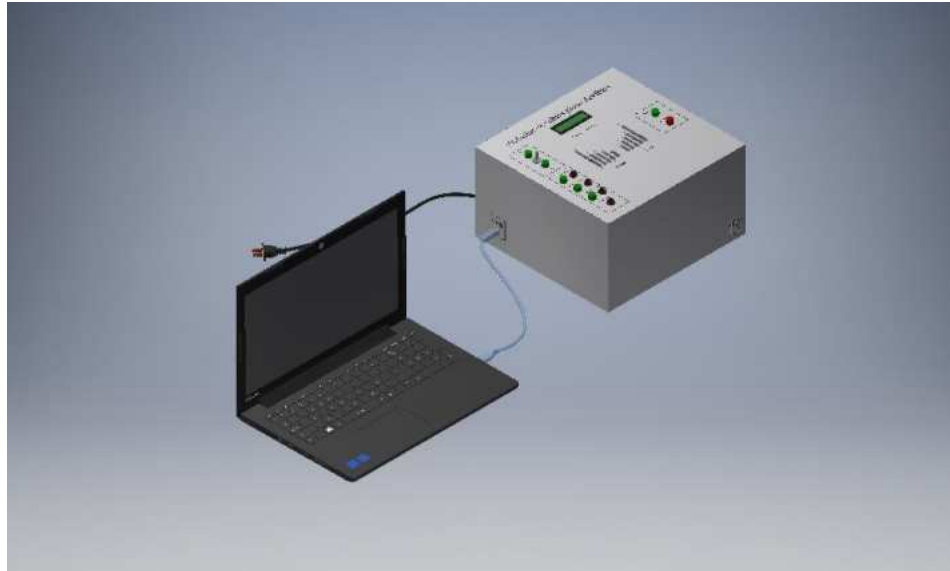


Fig. 3.23 Prototipo conectado directamente hacia la PC.

3.1.1 Elaboración exterior de prototipo.

Una vez que todo el sistema ha sido modelado/diseñado en CAD, se procede a su construcción. Para la carcasa se usó panel de aluminio (Figs. 3.24 y 3.25); también, se realizaron las perforaciones a la medida de cada uno de los elementos del cual está conformado.

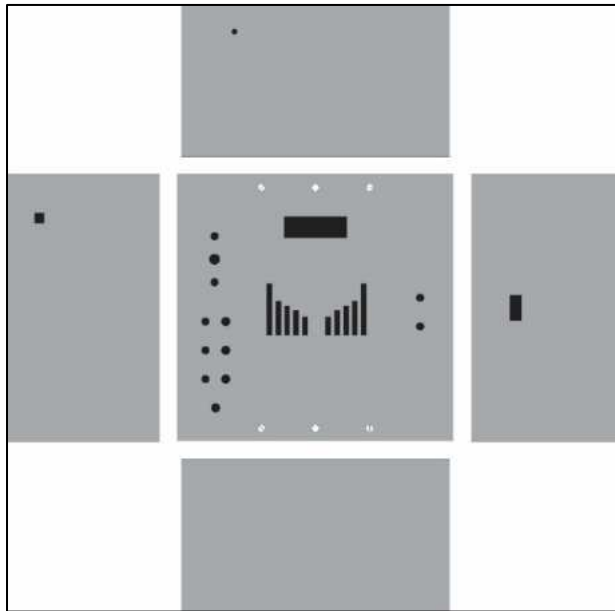


Fig. 3.24 Cada una de las caras con sus respectivas perforaciones.



Fig. 3.25 Cara superior después de realizar los cortes.

Cada uno de los elementos eléctricos por los que se conforman se colocó en las ranuras correspondientes, Fig. 3.26.

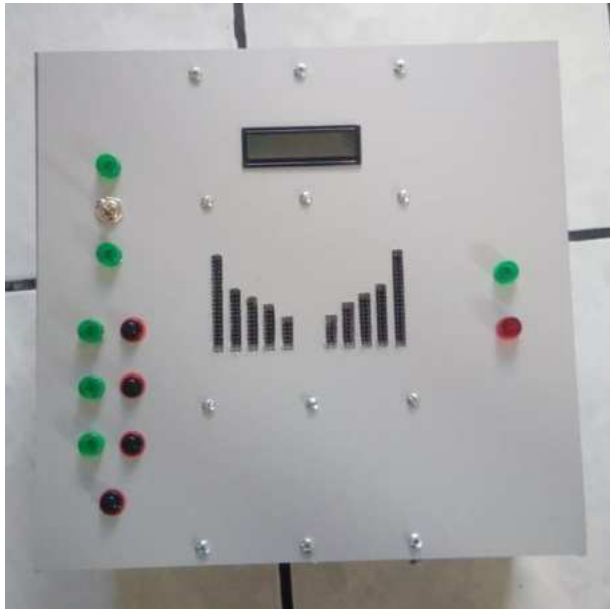


Fig. 3.26 Cara superior con los elementos instalados.

También se crearon etiquetas para la señalización de cada una de sus secciones repartidas en sus diferentes caras, colocando las etiquetas correspondientes mostradas en la Fig. 3.27.

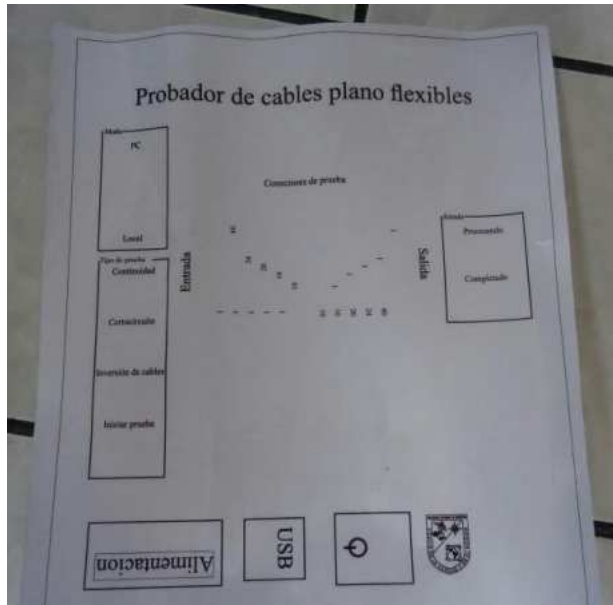


Fig. 3.27 Plantilla de las etiquetas.

Finalmente, se terminó de colocar las etiquetas a las caras restantes para darle la presentación exterior al prototipo. En la Fig 3.28 se presenta la cara superior ya finalizada.

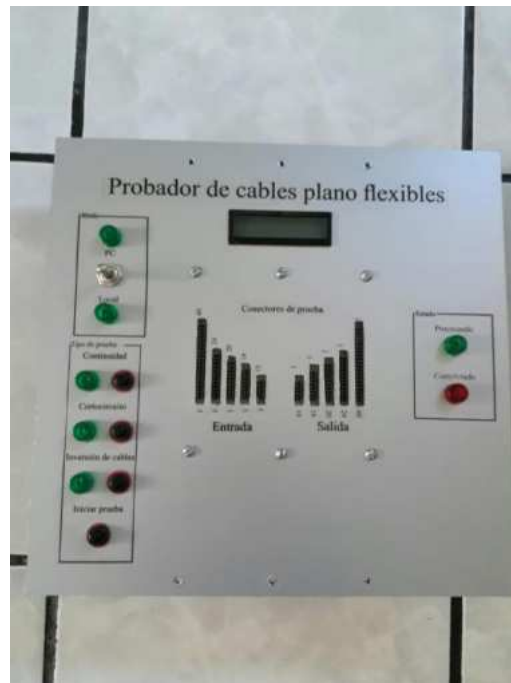


Fig. 3.28 Tapa superior con etiquetas.

Para la fijación de la placa donde se alojan los conectores de pines se realizó una base para brindarle soporte y evitar que la placa se flexione. Esta base fue elaborada mediante impresión 3D y utilizando como material, termoplástico PLA (ácido poliláctico), material derivado de diversos recursos renovables tales como almidón de maíz, tapioca o caña de azúcar y se puede degradar con facilidad bajo ciertas condiciones, tales como la presencia de agua y óxido de carbono. El proceso de la fabricación se aprecia en la Fig. 3.29. El diseño de la base de soporte, Fig. 3.30, fue realizado tomando de tal manera que permita la fácil conexión interna de las terminales de la placa con las terminales de los circuitos de acondicionamiento, asegurándose de que cumpla con la rigidez necesaria para soportar los esfuerzos generados en el material y en la placa debido a la conexión y desconexión de los cables bajo prueba (Figs. 3.31 y 3.32).



Fig. 3.29 Elaboración de base para placa de circuitos.



Fig. 3.30 Base de soporte finalizada.



Fig. 3.31 Base montada sobre la cara superior.



Fig. 3.32 Resultado final de prototipo.

3.2 Circuito de acondicionamiento eléctrico

Para establecer la conexión entre Arduino y las múltiples entradas y salidas requeridas, se realizó el circuito de acondicionamiento utilizando los multiplexores y registros de cambio ya mencionados anteriormente. Se estableció una conexión en cascada de cada uno de los registros de cambio y multiplexores para así poder obtener el número deseado tanto de entradas de información como de salidas.

Partiendo de los circuitos hechos inicialmente para pruebas y utilizando un software de diseño de circuitos electrónicos se fabricó una tarjeta PCB para cada uno de los circuitos de acondicionamiento ya mencionados. Las tarjetas PCB permiten el montaje de cada uno de los componentes electrónicos conectados eléctricamente por medio de pistas hechas de un material conductor sobre una placa fabricada de un material aislante. Dicho software, una vez terminado el diseño, permite la exportación del circuito en formato Gerber que incluye las instrucciones necesarias para realizar el maquinado del circuito sobre la placa fenólica. Las siguientes subsecciones describen esto a detalle.

3.2.1 Circuitos de acondicionamiento para registros de cambio 74HC959.

En la Fig. 3.33 se muestra la conexión en cascada de los registros de cambio y cada una de sus conexiones. Como se puede observar solamente se necesita disponer de tres pines de salida desde el Arduino (*dataPin*, *latchPin* y *clockPin*). Por el otro lado se tienen Leds para indicar en que pin se envía el dato (en la etapa de prueba), para la etapa final estos son eliminados.

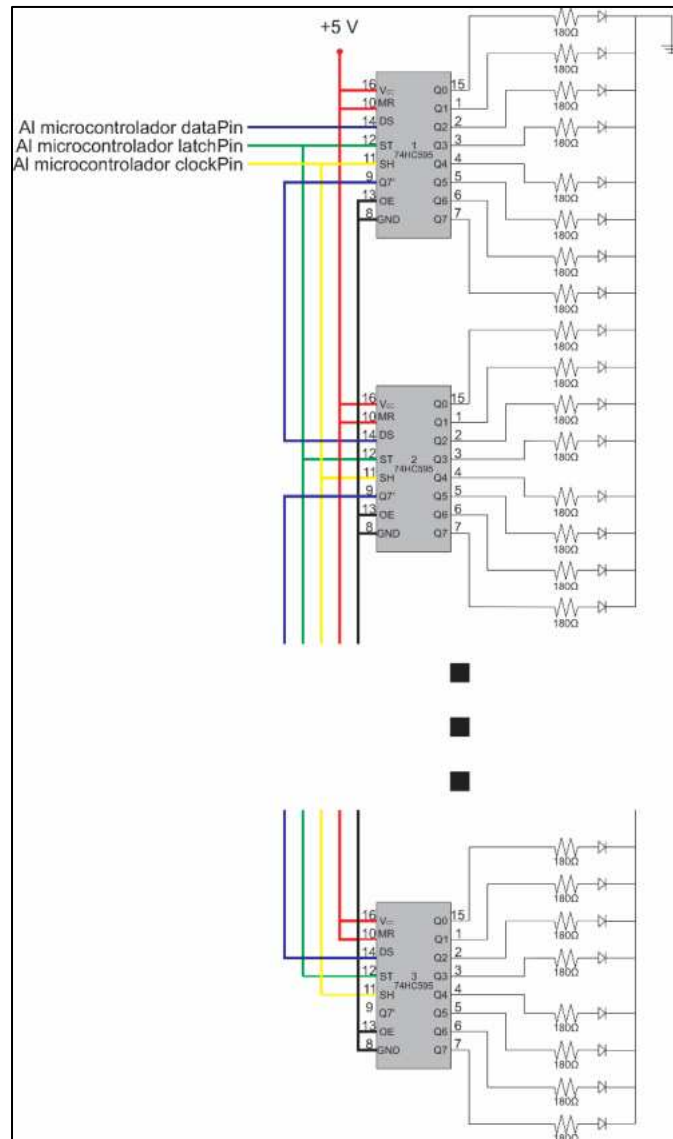


Fig. 3.33 Diagrama de conexión de registros de cambio en cascada.

Una vez establecido el diseño y la conexión del circuito, se desarrolló la PCB en la que se montaron cada uno de los componentes. La visualización del archivo Gerber se muestra en la Fig. 3.34, mientras que una vista del modelo elaborado en 3D del resultado final es mostrado en la Fig. 3.35, que muestra el montaje de los componentes sobre la placa, por otro lado en la Fig. 3.36 se observa la parte inferior donde se aprecia la conexión de las pistas.

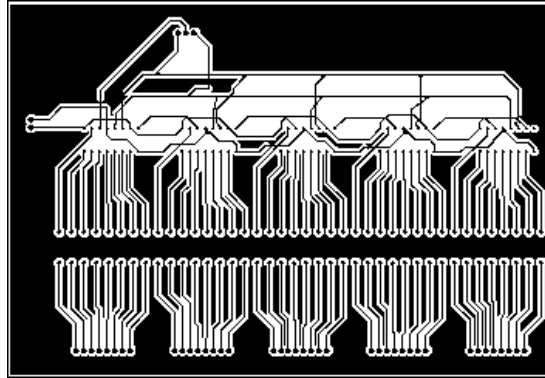


Fig. 3.34 Vista previa del Gerber para la realización de la PCB de los registros de cambio.

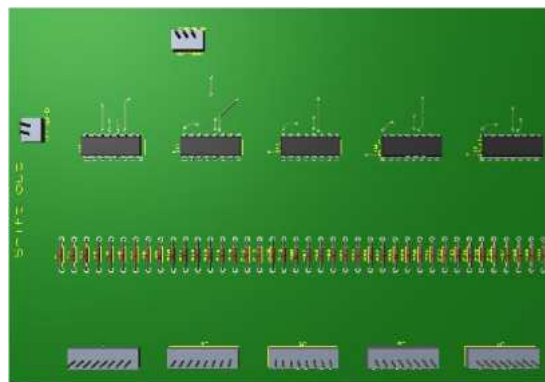


Fig. 3.35 Modelo en 3D de la vista superior de la tarjeta PCB para registros de cambio.

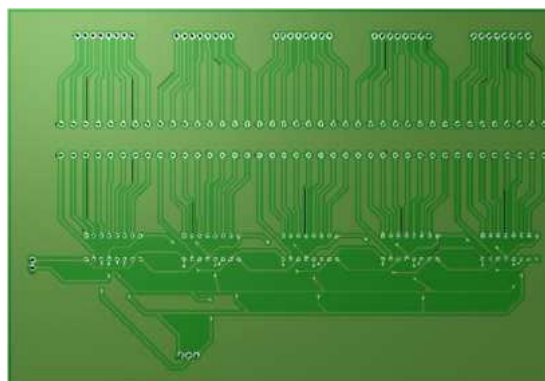


Fig. 3.36 Modelo en 3D de la vista inferior de la tarjeta PCB para registros de cambio.

Una vez validado el PCB, se prosiguió a elaborar el circuito previamente diseñado sobre la placa fenólica. El resultado final se muestra en la Fig. 3.37 y Fig. 3.38 con vistas superior e inferior respectivamente.

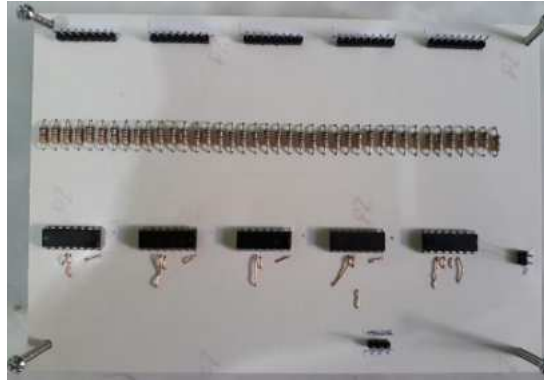


Fig. 3.37 Resultado final de PCB para registros de cambio, vista superior.

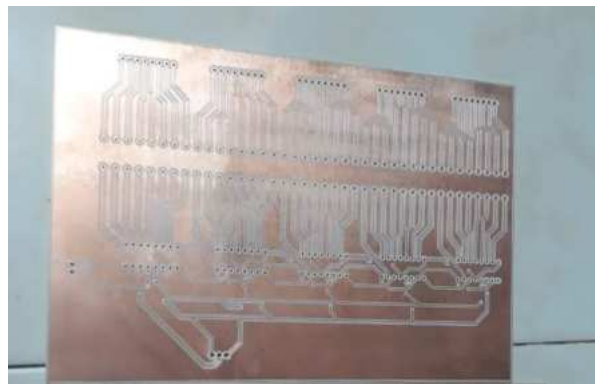


Fig. 3.38 Resultado final de PCB para registros de cambio, vista inferior.

3.2.2 Circuitos de acondicionamiento para multiplexores CD4021.

Para los multiplexores, circuito de la Fig. 3.39, se sigue un proceso similar al anterior. Se tienen las salidas que van dirigidas hacia las entradas del Arduino, con la diferencia de que ahora este arreglo cumple con la función de recibir el dato de forma manual a través de los interruptores de contacto instalados en sus entradas y como acción siguiente envía el dato recibido hacia el microcontrolador permitiendo conocer en qué posición se recibió la señal (esto en la parte de pruebas). Para el prototipo fina, estos componentes son removidos.

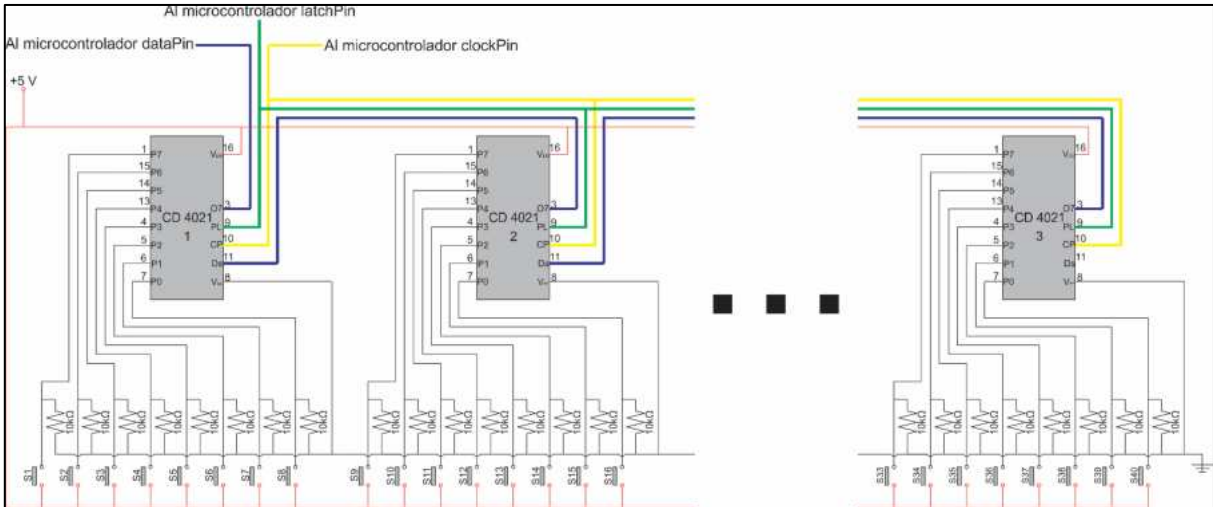


Fig. 3.39 Diagrama de conexión de multiplexores en cascada.

Para el desarrollo de la PCB de los multiplexores se siguió el mismo procedimiento usado en los registros de cambio. En las Fig. 3.40, Fig. 3.41 y Fig. 3.42 se exponen la visualización del archivo Gerber, la vista superior del modelo en 3D del circuito y la vista inferior del modelo 3D del circuito, correspondientemente.

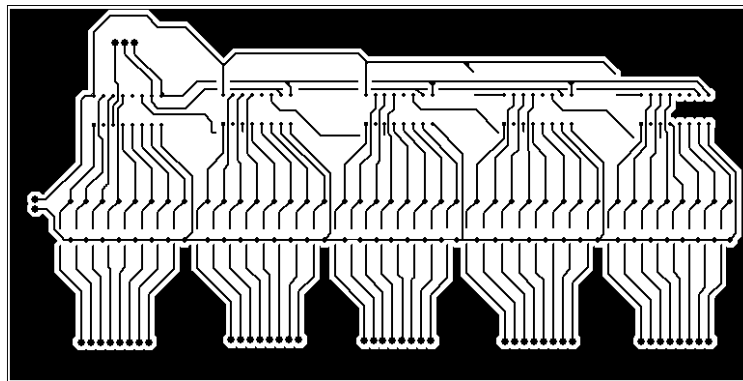


Fig. 3. 40 Vista previa del Gerber para la realización de la PCB de los multiplexores.

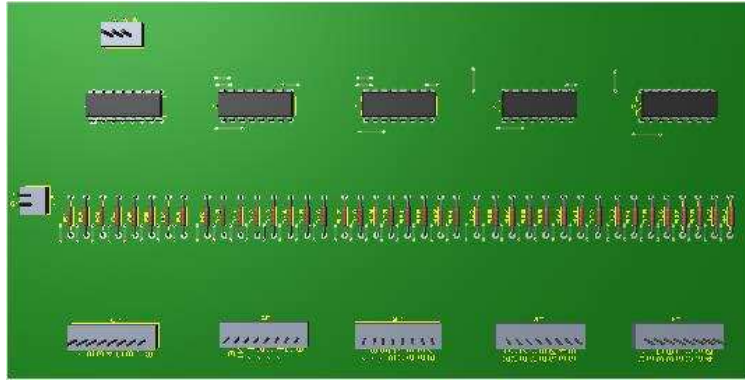


Fig. 3.41 Modelo en 3D de la vista superior de la tarjeta PCB para multiplexores.

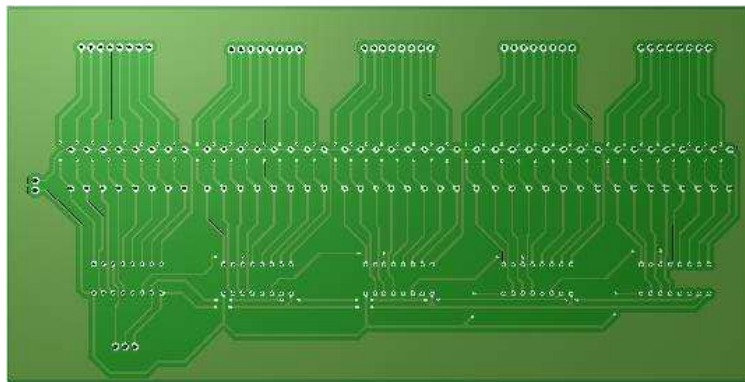


Fig. 3.42 Modelo en 3D de la vista inferior de la tarjeta PCB para multiplexores.

Terminado el diseño, se elaboró la manufactura de la PCB para los multiplexores. En la Fig. 3.43 se muestra el resultado final de la tarjeta vista desde la parte superior, mientras que en la Fig. 3.44 se puede observar la parte inferior en la que es posible observar las pistas que lo conforman.

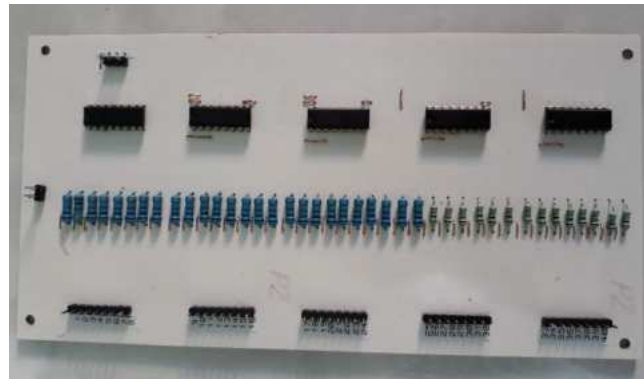


Fig. 3.43 Resultado final de PCB para multiplexores, vista superior.

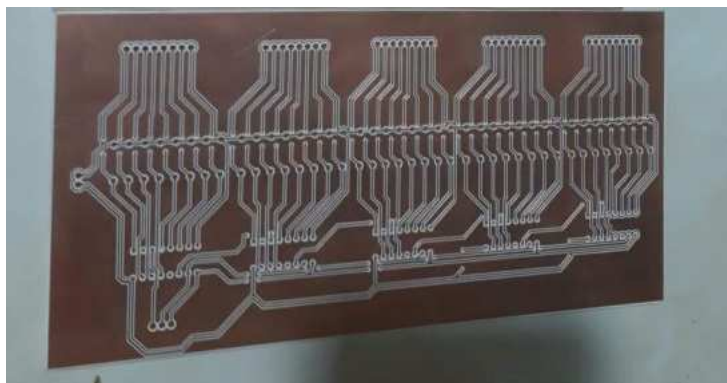


Fig. 3.44 Resultado final de PCB para multiplexores, vista inferior.

3.2.3 Circuito de acondicionamiento para Display LCD

Al igual que los elementos anteriores, es necesaria la elaboración de una tarjeta de circuito impreso para la correcta conexión del display LCD que forma parte del prototipo. El procedimiento que se realizó fue el mismo partiendo desde la exportación de los archivos Gerber mostrados en la Fig. 3.45, las vistas superior e inferior del modelo en 3D presentados en las Fig. 3.46 y Fig. 3.47 y finalizando con los resultados del maquinado sobre la placa fenólica con su vista superior mostrados en la Fig. 3.48 y la vista inferior en la Fig. 3.49.

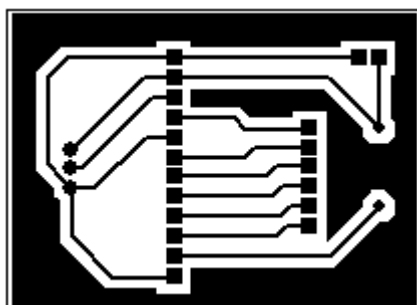


Fig. 3.45 Vista previa del Gerber para la realización de la PCB del display LCD.

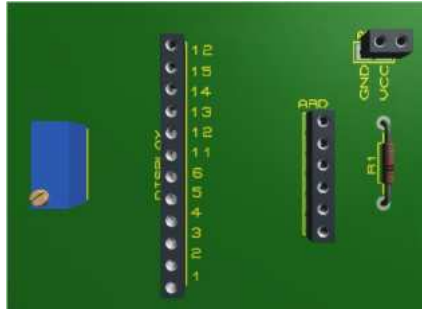


Fig. 3.46 Modelo en 3D de la vista superior de la tarjeta PCB para el display LCD.

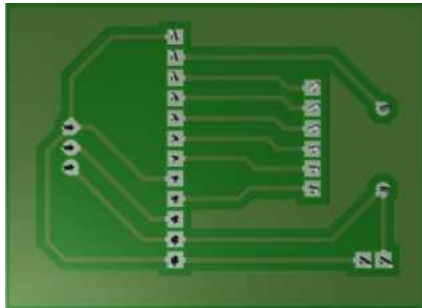


Fig. 3.47 Modelo en 3D de la vista inferior de la tarjeta PCB para el display LCD.



Fig. 3.48 Resultado final de PCB para el display LCD, vista superior.

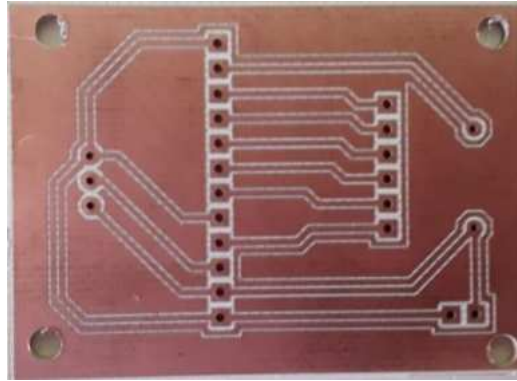


Fig. 3.49 Resultado final de PCB para el display LCD, vista inferior.

3.2.5 Montaje de circuitos dentro del prototipo

Finalmente, como se muestra en la Fig. 3.50, se realizó el montaje de los circuitos en el interior del prototipo, se acomodó cada una de las tarjetas PCB de forma que su ubicación facilite la conexión con los demás elementos correspondientes. Fue necesario agregar una placa fenólica extra para que funcione como el nodo que comunica a la alimentación y la tierra de cada una de las PCB hacia Arduino.



Fig. 3.50 PCB montadas dentro del prototipo.

Basándose en la Fig. 3.51, se realiza una descripción general para identificar cada uno de los PCB y su ubicación dentro del prototipo. De acuerdo con la numeración utilizada se identifica lo siguiente:

1. PCB para registros de cambio 74HC959.
2. Placa de alimentación (Voltaje y tierra).
3. PCB para multiplexores CD4021.
4. Arduino.
5. Circuito de alimentación y transmisión de datos para display LCD.

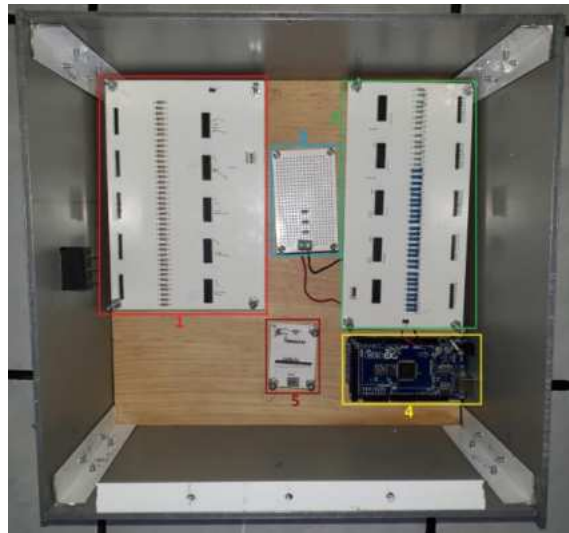


Fig. 3.51 Identificación de circuitos.

Finalmente, se conectaron los componentes con sus respectivos elementos de hardware. En la Fig. 3.52, se muestra una vista lateral que permite observar el cableado de las PCB.



Fig. 3.52 Vista lateral del prototipo con vista interior.

3.3 Interfaz de usuario

La interfaz de usuario es descrita como el medio en el cual se produce la interacción entre el usuario y la máquina con la finalidad de permitir un mayor control y un funcionamiento más eficiente del equipo, en otras palabras, permite establecer la comunicación entre dos sistemas que no hablan el mismo lenguaje. Se forma por diferentes elementos que pueden ser menús, botones, *radiobuttons*, barras de estado, etc., esto es en lo que refiere a la parte de software. De modo similar existen componentes de hardware que forman parte de la interfaz, como son el *mouse*, luces indicadoras, pantalla, teclado e inclusive sonidos emitidos al realizar alguna acción dentro de la interfaz. Es necesario cumplir con ciertos factores para poder crear una interfaz exitosa. La simplicidad es importante ya que al crear una interfaz intuitiva y de fácil manejo se está ofreciendo al usuario la capacidad de trabajar de forma más veloz. Otro factor importante es la experiencia que se le brinda al usuario, al ser una interfaz más amigable y estética se le ofrece al usuario una mejor experiencia sensorial.

Como ya se mencionó anteriormente, la interfaz puede ser utilizada para el control de ciertos procesos, por ejemplo, el apagado y encendido de algunos equipos, la velocidad a la que se desea que ocurran ciertos procesos y algunas otras opciones más, inclusive son útiles para poder monitorear el estado de un

objeto en cuestión (temperatura, velocidad, proximidad, etc.). Todo esto puede ser realizado a través de una interfaz gráfica de usuario (GUI, *graphical user interface*, por sus siglas en inglés), cuyas características son el empleo de objetos gráficos e imágenes para representar las acciones y la información disponible y así facilitar un entorno visual sencillo que permita esta comunicación usuario-máquina.

Acorde con lo ya mencionado, es importante tener un medio por el cual transmitir la información recopilada de los cables bajo prueba. Por lo cual se desarrolló una interfaz para monitorear el estado de dichos cables. La interfaz fue realizada utilizando el software Visual Studio, del cual ya se habló en el capítulo anterior. Para la elaboración de prueba a un cable plano es necesario indicar el número de pines y cuantas veces se desea repetir la prueba para comprobar la precisión del análisis. Por lo cual es importante que la interfaz contenga los elementos necesarios para permitir al usuario la capacidad de poder seleccionar las opciones deseadas y posteriormente obtener la información requerida por la prueba.

A continuación, en la Fig. 3.53 se muestra el diseño de la interfaz para un cable de 40 pines:

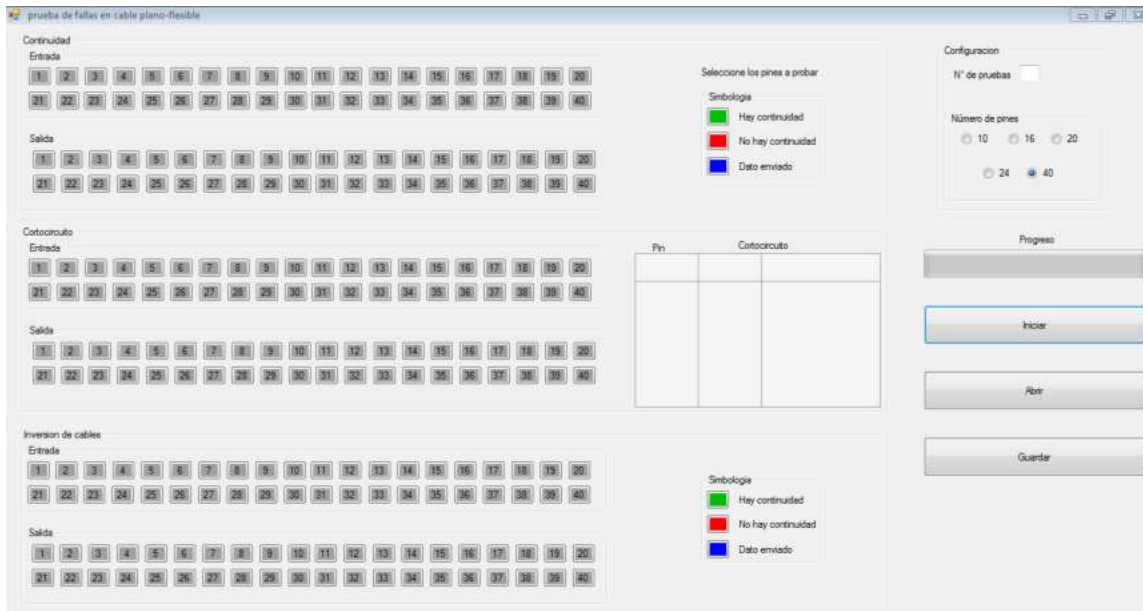


Fig. 3.53 Vista general de la interfaz de usuario.

Como se puede observar en la vista general de la interfaz de usuario, se muestra cada una de las pruebas que se realizan y por cada prueba se observa la cantidad de pines de entrada y salida a elegir por el usuario, incluyendo la simbología correspondiente permitiendo la interpretación de los resultados según el estado de los pines. Para una mejor explicación del contenido de la GUI se divide las secciones acorde a la Fig. 3.54.

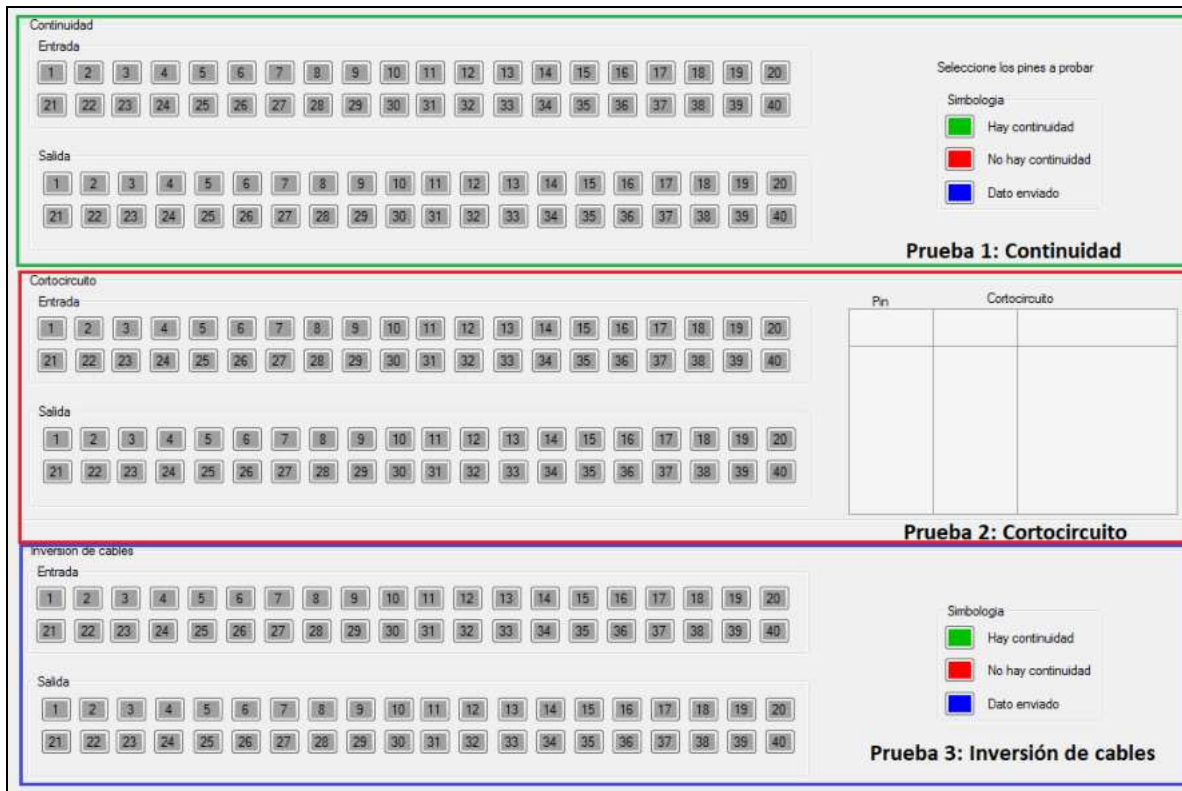


Fig. 3.54 Panel de pruebas dividido por secciones.

En primer lugar, la sección de continuidad incluye los pines de entrada que indicarán si el dato ha sido enviado hacia el cable bajo prueba, mientras que los pines de salida cambiarán su estado (correspondiente a la simbología anexa) si ya se ha detectado la señal de respuesta (Fig. 3.55).

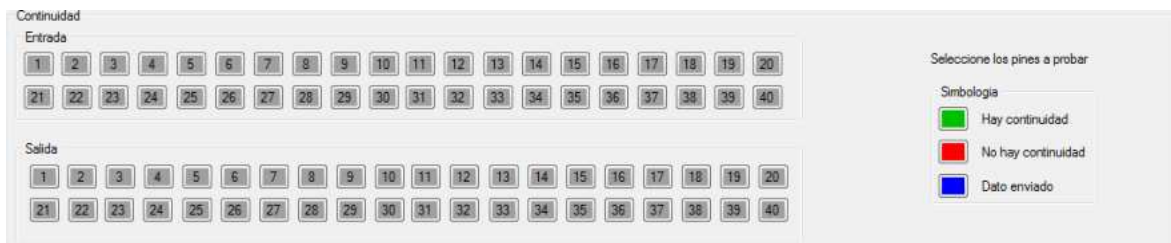


Fig. 3.55 Prueba de continuidad antes de realizar la prueba.

Si el pin de salida cambia a color verde significa que hay continuidad y rojo si no se detecta ninguna señal de salida, para los pines de entrada cuando cambia

a azul indica que ya se ha enviado el dato para su lectura, la Fig. 3.56 es un ejemplo de una prueba de continuidad ya realizada donde se presenta falla de continuidad en los pines 13 y 35; de acuerdo a la simbología los pines con falla se resaltan de color rojo y los conductores sanos se resaltan de verde. El color azul indica el estado de la señal de entrada hacia el cable, cambiando a ese color si la señal ya ha sido enviada.



Fig. 3.56 Prueba de continuidad después de realizar la prueba.

La segunda prueba, cortocircuito (Fig. 3.57), cumple con el mismo principio de enviar y recibir los datos y cambiando el color según su estado. Esta prueba también incluye una tabla en donde se despliega la información sobre la continuidad que tiene un pin respecto de otro, poniendo en evidencia de esa forma si existe alguna falla de cortocircuito.

El propósito de la tabla que se encuentra en el lado izquierdo de la Fig. 3.57 consiste en enviar la señal en cada uno de los pines individualmente e identificar si se detecta una única señal en el pin del otro extremo, asegurándose de que solo sea recibido en una salida y en el mismo número. En cambio, si se lee en la salida más de una señal, significa que hay contacto entre uno o más cables adyacentes.

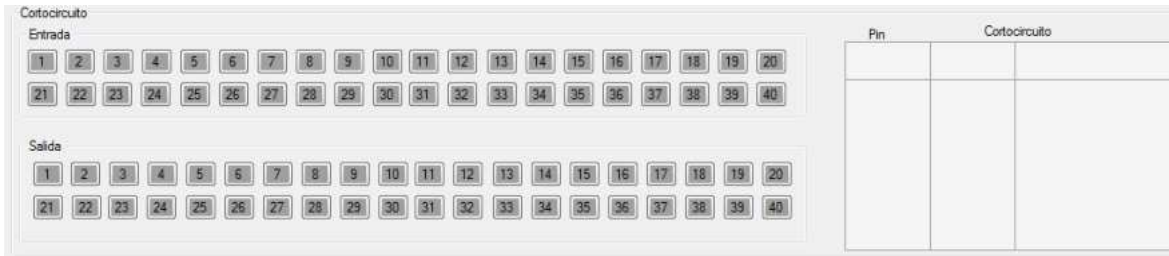


Fig. 3.57 Prueba de cortocircuito antes de realizar la prueba.

En la Fig. 3.58 se muestra un ejemplo del análisis, donde la señal ha sido enviada a través del pin 4, pero ha sido detectada en los pines de salida 4, 6 y 7, demostrando así que existe cortocircuito entre estos tres pines. En la tabla se muestra el pin de salida, en este caso el 4, contra las señales extra en la que se detectó el cortocircuito.



Fig. 3.58 Prueba de cortocircuito después de realizar la prueba.

Para la prueba de inversión de cables se tienen los pines de entrada y salida junto con la simbología ya antes mencionada, Fig. 3.59. El objetivo del análisis de la inversión de cables es el detectar si la posición de la señal de entrada es la misma en la salida. En caso de ser detectada una señal de salida, el botón del pin cambiará de acuerdo con la respuesta recibida.



Fig. 3.59 Prueba de inversión de cables antes de realizar la prueba.

En la Fig. 3.60 se muestra un ejemplo con el análisis finalizado. Simplemente se resalta el pin de salida invirtiendo los números de los cables en los que se detectó la falla. Este análisis permite detectar exactamente que conductores están cruzados entre sí.

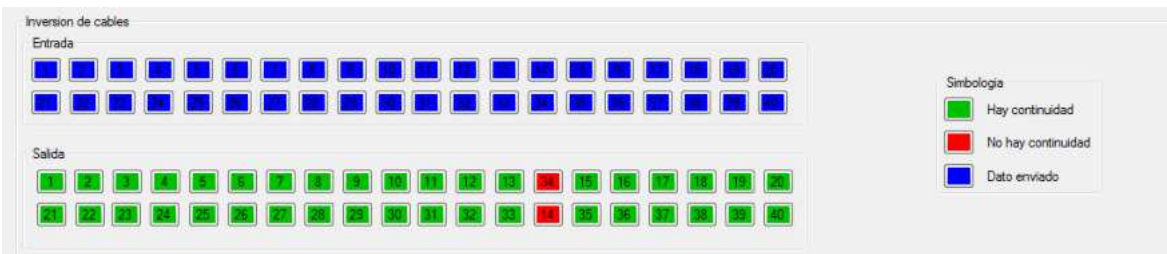


Fig. 3.60 Prueba de inversión de cables después de realizar la prueba.

En la sección de configuración, Fig. 3.61 se inserta el número de pruebas a realizar; dedicada a poder repetir el análisis la cantidad de veces que indique el usuario y así conseguir una respuesta más precisa. En número de pines se selecciona la medida del cable bajo prueba.

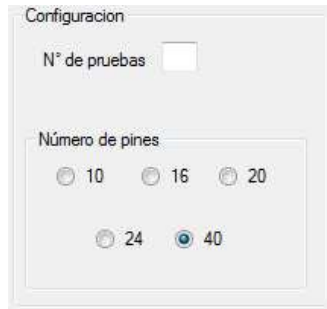


Fig. 3.61 Sección de configuración.

Finalmente, se ubica la barra de progreso; en la cual se denota el avance del análisis (Fig. 3.62). Los botones iniciar, para dar comienzo a la prueba, abrir, para abrir una prueba previa y el botón guardar que permite conservar los datos para una futura consulta.



Fig. 3.62 Barra de progreso y botones de opciones.

Ya una vez conociendo el diseño final, es importante conocer más a fondo el principio de funcionamiento del software a través del diagrama de flujo que se muestra a continuación. En la Fig. 3.63 se muestra dicho diagrama de flujo, el cual rige la lógica del diagnóstico.

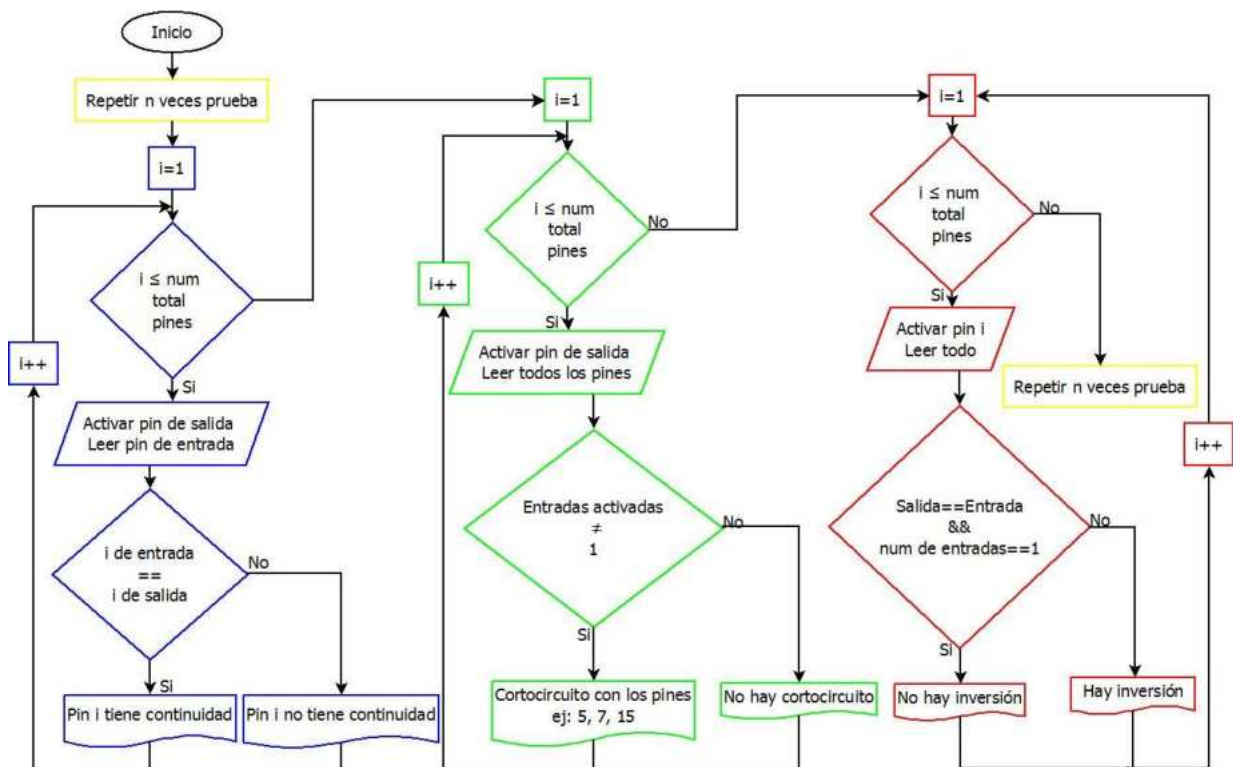


Fig. 3.63 Diagrama de flujo del principio de funcionamiento de la interfaz de usuario.

En primer lugar, al iniciar el programa es necesario insertar el número n de veces que se repetirá la prueba. Después se ejecuta la primera fase, prueba de continuidad, donde la variable i toma el valor inicial de $i=1$, se hace la comparativa si i es menor al número total de pines existentes en el cable, en caso de ser afirmativa se envía la señal de salida hacia la entrada del pin con el mismo valor de i e inmediatamente es leída en la salida en la misma posición. Se realiza la comparativa, si la señal de entrada es igual a la de salida se considera que existe continuidad entre los pines, en caso contrario no existe continuidad. A continuación, el valor de i aumenta en una unidad y regresa al condicional hasta que el valor de i sea mayor al número de pines para posteriormente pasar a la segunda prueba.

En segundo lugar está la prueba de cortocircuito, iniciando el valor de la variable $i=1$ para comenzar nuevamente con el barrido. A continuación, hace la comparativa de i con el número de pines. En caso de ser verdadera, se envía la señal hacia la entrada del pin correspondiente al valor de i , mientras que, en la salida se leen los valores de todos los pines y se compara el número de entradas activadas, si la cantidad de pines que recibieron una señal es mayor a uno, entonces existe cortocircuito entre varios pines. En contraste con la respuesta recibida, no existe cortocircuito, por ende tampoco hay falla. Finalmente el valor de i aumenta en una unidad y se repite el ciclo, hasta que i tome un valor mayor al número de pines para continuar con la siguiente prueba.

Finalmente, en la prueba de inversión de cables, se regresa el valor de $i=1$ y se realiza la comparativa con el número de pines ya mencionada anteriormente. Si es afirmativa dicha comparativa, se envía la señal hacia el pin con el valor de i en la entrada, mientras que se leen todas las salidas, si se recibe el dato en la misma posición de salida que de entrada, significa que no existe inversión, si se recibe respuesta y es en una posición diferente, entonces existe inversión. A continuación el valor de i aumenta en una unidad y se repite el ciclo hasta que alcance un valor mayor al número de pines. Cuando el valor de i sea mayor al número de pines, el valor de n aumenta en una unidad y se repite desde la primera prueba hasta que se repita n veces el proceso.

Capítulo 4

Pruebas y Resultados

4.0. Experimentación

En esta sección se describen las pruebas del sistema desarrollado. La Fig. 4.1 muestra el ensamble software-hardware final. Este ensamble final ya presenta a la PC conectada hacia el prototipo final; al mismo tiempo, en la Fig. 4.2 se muestra el mismo ensamble pero con la caja destapada para exponer los circuitos internos.



Fig. 4.1 Ensamble final software-hardware.

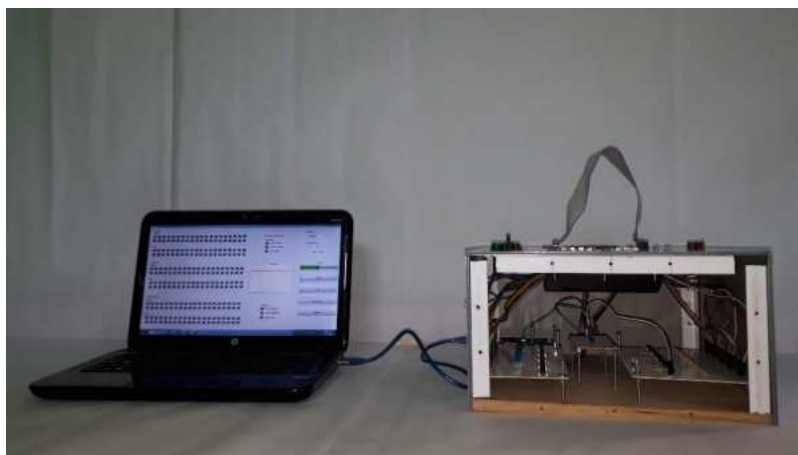


Fig. 4.2 Ensamble software-hardware con circuitos internos visibles.

4.1 Matriz de pruebas

Una vez realizado el software de prueba se realizó el código que permite la comunicación de la interfaz de usuario con el hardware. De este modo se realizaron los distintos tipos de pruebas a cables de los diferentes tamaños que se incluyen en el proyecto (40, 24, 20, 16 y 10 conductores).

Para documentar si la prueba fue terminada con éxito, se elaboró la matriz de pruebas (Tabla 4), en la que muestra el resultado obtenido al aplicar las pruebas de las tres fallas incluyendo el análisis a cables sanos en cinco diferentes cables con cierto número de pines (los resultados obtenidos se muestran en las siguientes subsecciones). De esta forma se verifica el correcto funcionamiento del software en modo PC controlado por medio de la interfaz de usuario. Para cada cable se hacen 5 pruebas y 4 condiciones (tres fallas y sano).

Tabla 4 Matriz de pruebas de cables en modo PC.

Matriz de pruebas de cables en modo PC						
Tipo de falla	Cantidad de pruebas	40 Pines	24 pines	20 pines	16 pines	10 pines
Sano	5	✓	✓	✓	✓	✓
Continuidad	5	✓	✓	✓	✓	✓
Cortocircuito	5	✓	✓	✓	✓	✓
Inversión de cables	5	✓	✓	✓	✓	✓

Por otro lado, en la matriz de pruebas expuesta en la Tabla 5, se realiza el mismo número de pruebas y condiciones de funcionamiento pero en el modo portable.

Tabla 5 Matriz de pruebas de cables en modo Portable.

Matriz de pruebas de cables en modo Portable						
Tipo de falla	Cantidad de pruebas	40 Pines	24 pines	20 pines	16 pines	10 pines
Sano	5	✓	✓	✓	✓	✓
Continuidad	5	✓	✓	✓	✓	✓
Cortocircuito	5	✓	✓	✓	✓	✓
Inversión de cables	5	✓	✓	✓	✓	✓

4.2. Pruebas y resultados

En esta sección se retoman las causas de las fallas a analizar para continuar con los resultados obtenidos en las pruebas de falla de cada cable en el uso del software de pruebas, mostrando el correcto funcionamiento de los circuitos en conjunto con el software.

4.2.1. Pruebas y resultados de continuidad

La pérdida de continuidad ocasionada por la fatiga mecánica es uno de los problemas más graves, ya que inicialmente genera calor en la zona donde empieza a fallar debido al daño ocasionado al conductor y finalmente termina por abrir el circuito, impidiendo la conducción de energía eléctrica y/o datos. Sin embargo, la fatiga mecánica no es la única causa de falla, también pueden ser generados por algún corte realizado en el conductor, sin importar que haya sido intencional o accidental.

En la Fig.4.3 se muestra un cable plano-flexible con falla de continuidad, se observa que parte de sus conductores ha sido seccionado impidiendo así la continuidad.



Fig. 4.3 Cable con falla de continuidad.

Dentro del software de experimentación expuesto en la Fig 4.4 se presentan los resultados obtenidos de dicho cable. Si un cable está en buen estado se imprime un texto indicando en que pin se detectó, en contraste con este resultado, si la respuesta es que no existe continuidad aparecerá el texto “No hay continuidad” haciendo mención en el número de pin donde existe la falla, del lado izquierdo se muestra el contador de cuantos conductores han presentado falla. Estos resultados son usando la interfaz de exhibición de Arduino, esto para comprobar en primera instancia la información que le llega a la GUI.

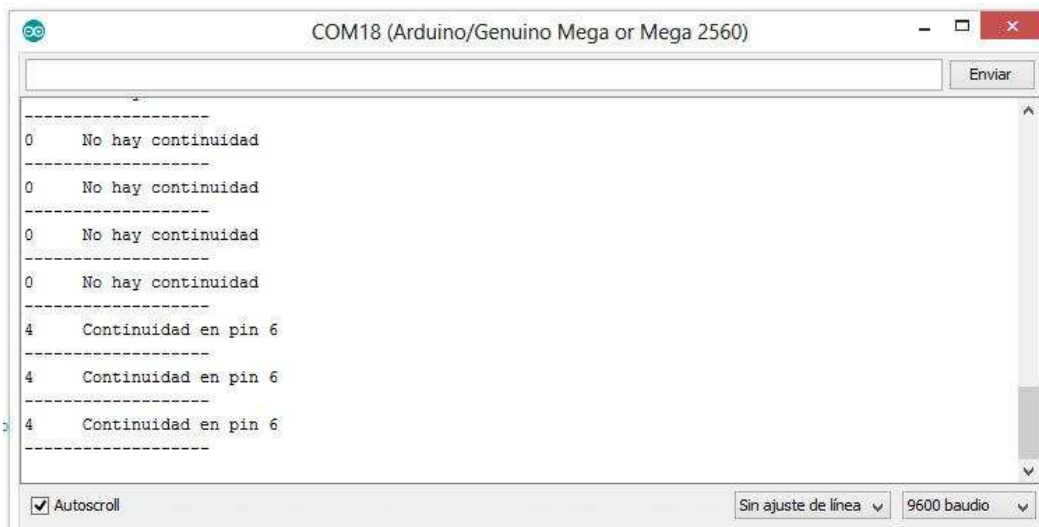


Fig. 4.4 Interfaz de usuario de pruebas durante el análisis de continuidad.

4.2.2. Pruebas y resultados de cortocircuito

Las fallas de cortocircuito, como se mencionó anteriormente, se generan debido a el contacto entre dos conductores, línea-línea o línea-neutro ocasionando un aumento brusco de la intensidad de la corriente (ver Fig. 4.5), dañando así los equipos eléctricos. Las causas más comunes de esta falla son por el deterioro del material aislante y descargas eléctricas por encima de la rigidez dieléctrica del material aislante.

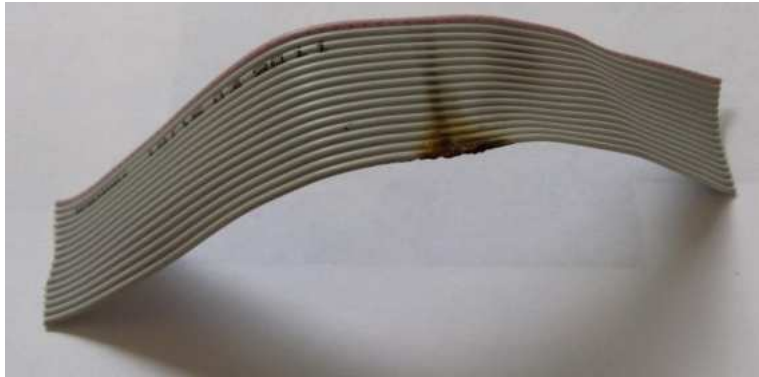


Fig. 4.5 Cable con falla de cortocircuito.

El software de pruebas envía la señal a una salida por vez para en el otro lado esperar la misma respuesta en un unico pin y que sea en la misma ubicación a la que se fue enviada. El software muestra el número de pin al que pertenece, así como el resultado obtenido; si el conductor se encuentra en buen estado indicará que el cable es “correcto” y si se detecta la señal en más de un conductor indicará que esta en “Cortocircuito”, este ejemplo se observa en la Fig. 4.6. Al final del renglón se muestra la ubicación del pin al que fue enviada la señal representada con 1 y 0, donde 1 significa el pin al que se detecto la señal y el 0 significa que no se detectó ningún tipo de señal. En la Fig. 4.6, se muestran los pines donde se detectaron fallas de cortocircuito. En este caso la falla se presentó en los pines 1 y 2 donde existe contacto entre ambos conductores igualmente que en los pines 15 y

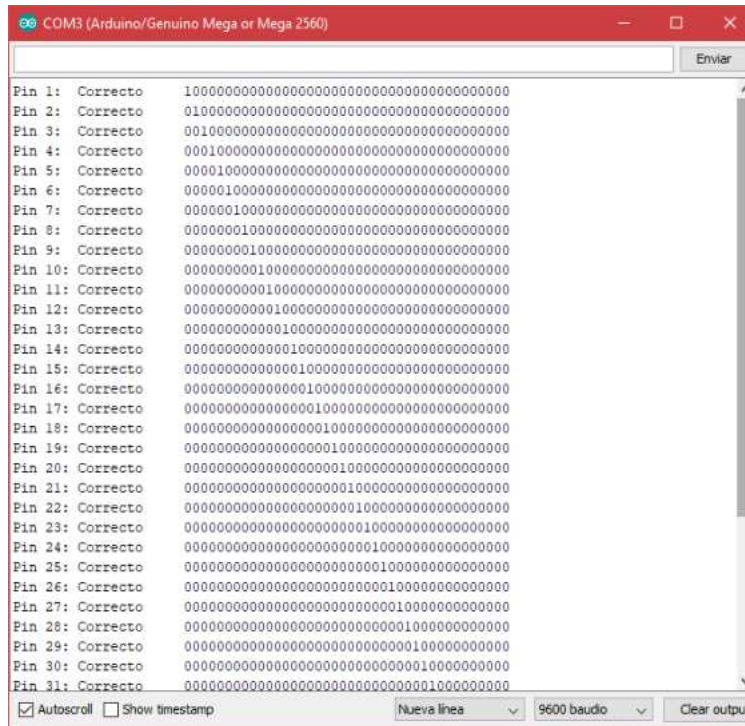


Fig. 4.7 Prueba realizada en un cable sano.

4.2.3 Pruebas y resultados de inversión de cables

Finalmente, se tiene la falla causada por inversión de cables, en la cual denota un intercambio entre uno o más conductores en el extremo final de un cable. Este tipo de falla es causado por una mala conexión y/o fabricación (ver Fig. 4.8).



Fig. 4.8 Inversión de cables.

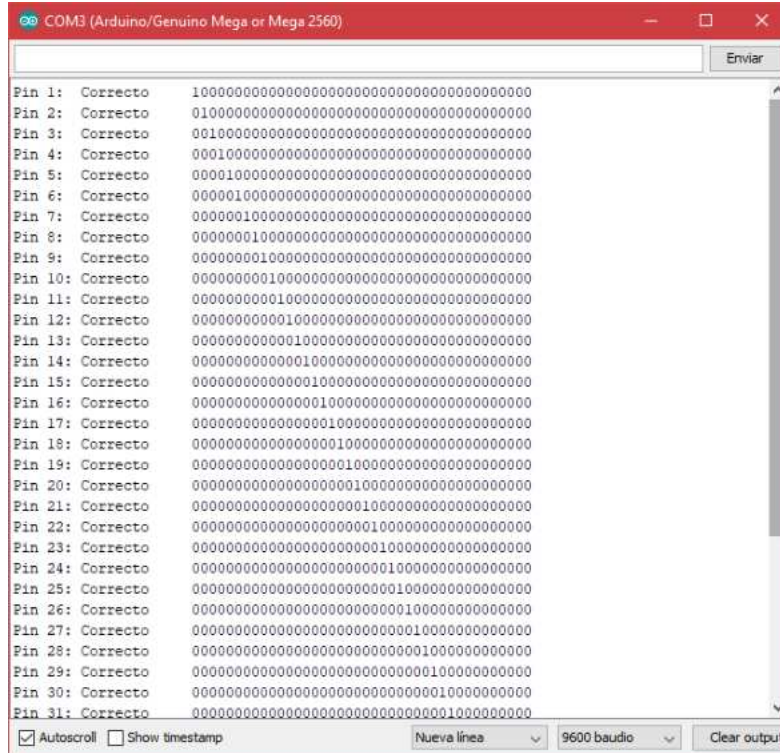


Fig. 4.10 Prueba realizada a un cable sano.

4.3. Resultados de pruebas en PC

Después de comprobar el funcionamiento del hardware, en esta sección se abordan los resultados finales aplicados en el modo PC. Incluyendo las evidencias del correcto funcionamiento del software junto con el hardware. Con fines demostrativos y de claridad, así como para no saturar el documento, solo se muestran los resultados y evidencias del cable de 40 pines y con los tres distintos tipos de falla para tener un control de la ubicación y tipo de falla que tendrán sus conductores. En los pines 1 y 2, se cortó la continuidad; en los pines 20 y 21 se provocó un cortocircuito en el cable y finalmente en los pines 39 y 40 se intercambiaron las posiciones de los cables. Las modificaciones hechas a dicho cable se muestran en las Fig. 4.11 y Fig. 4.12. Las figuras muestran los daños

inducidos al final de las pruebas, no obstante cada uno de ellos fue generándose en forma secuencial (uno a uno).



Fig. 4.11 Cable de pruebas, vista lateral.



Fig. 4.12 Cable de pruebas, vista superior.

Para activar el modo PC es necesario asegurarse de que el equipo esté conectado directamente a la PC y mover el switch ubicado en la sección Modo en la posición PC como se muestra en la Fig. 4.13.



Fig. 4.13 Switch de modo en la posición de PC.

4.3.1. Pruebas de Continuidad

La prueba de continuidad mostrada en la Fig. 4.14, presentó como resultado fallas en los pines 1 y 2 resaltando de color rojo los números correspondientes y de verde los pines que se detectaron sin falla; cumpliendo de esa forma con los resultados deseados.

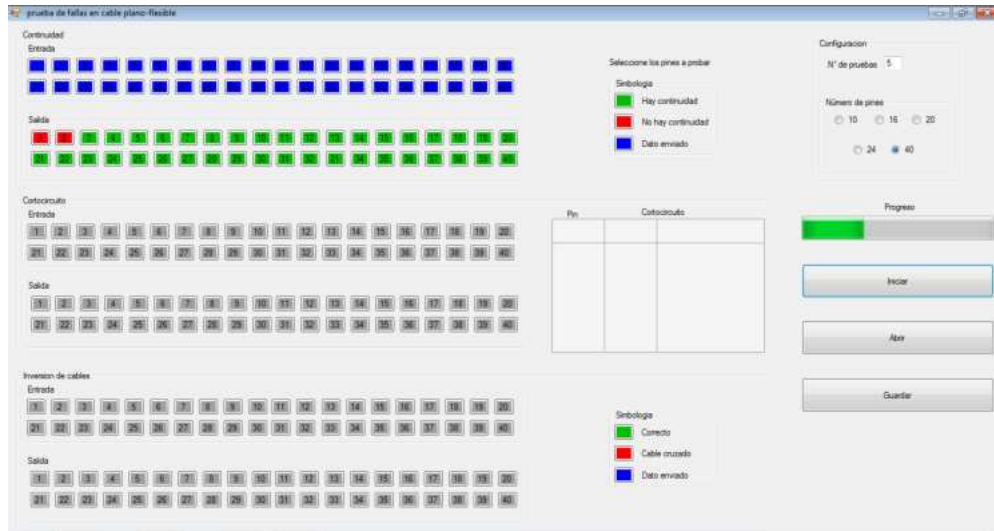


Fig. 4.14 Prueba de continuidad.

La Fig. 4.15 muestra el sistema funcionando, indicando el tipo de prueba que se realizó (primer indicador en verde, centrado en altura y a la izquierda).



Fig. 4.15 Prueba de continuidad realizada en modo PC.

4.3.2. Pruebas de cortocircuito

Como siguiente paso, se realizó la prueba de cortocircuito al cable obteniendo los resultados obtenidos en la Fig. 4.16. Las fallas presentadas en los pines 20 y 21 son mostradas en la interfaz de usuario una vez finalizada la prueba de cortocircuito. La Fig. 4.17 muestra el sistema funcionando, indicando el tipo de prueba que se realizó (segundo indicador en verde, centrado en altura y a la izquierda).

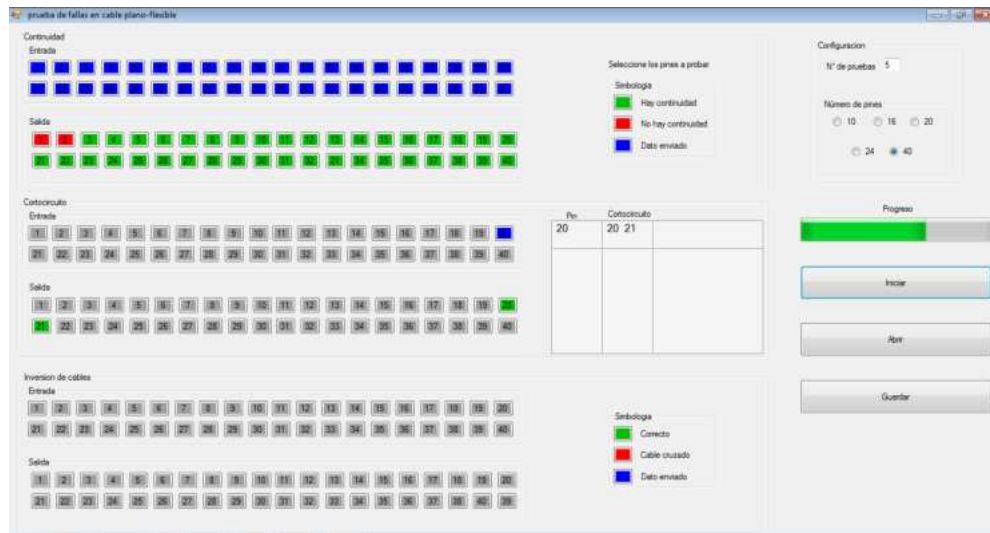


Fig. 4.16 Prueba de cortocircuito.



Fig. 4.17 Prueba de cortocircuito realizada en modo PC.

4.3.3. Pruebas de cables cruzados

Como siguiente paso, la prueba de cables cruzados fue realizada exitosamente. Mostrando fallas en los pines 39 y 40 cambiando la ubicación de los números hacia donde se detectó la señal de respuesta y cambiando a color rojo los números donde se presentaron las fallas (Fig. 4.18). De esta forma se comprueba cada una de las pruebas. La Fig. 4.19 muestra el sistema funcionando, indicando el tipo de prueba que se realizó (tercer indicador en verde, centrado en altura y a la izquierda).

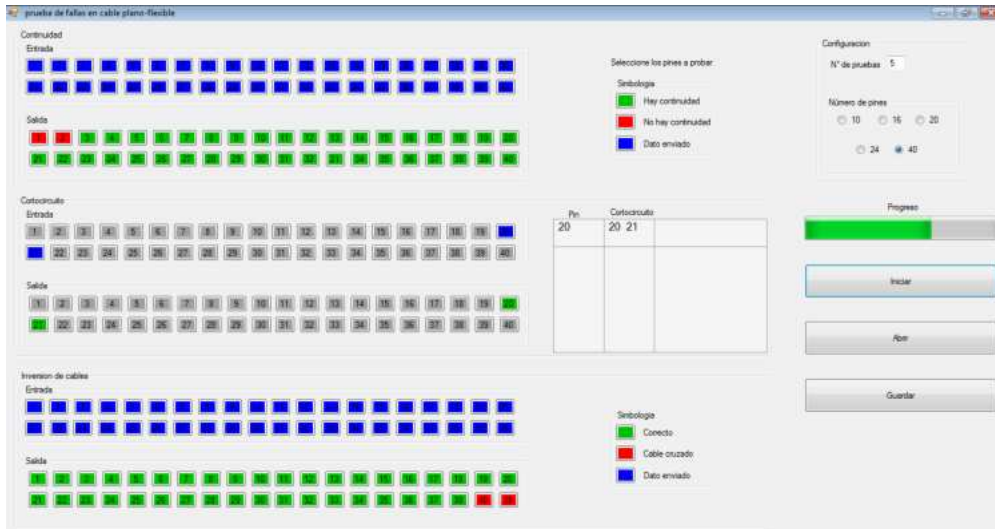


Fig. 4.18 Prueba de cortocircuito.

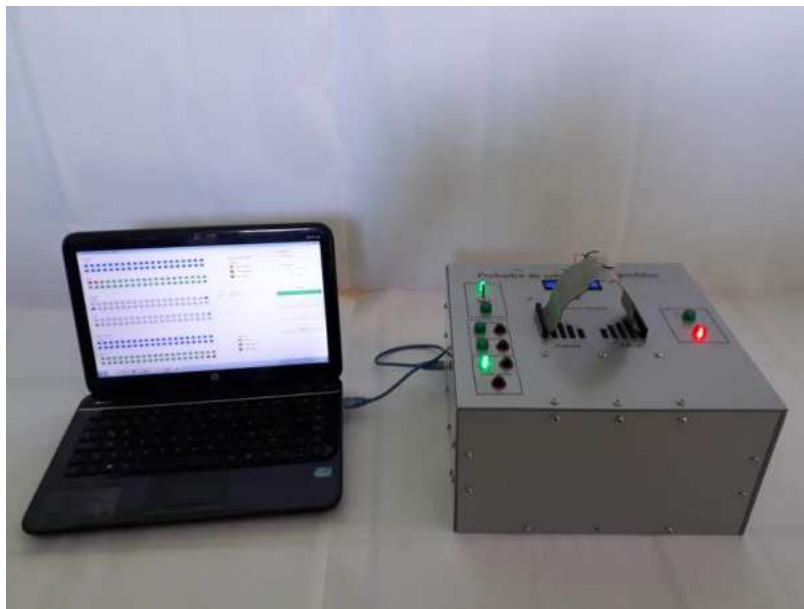


Fig. 4.19 Prueba de cortocircuito realizada en modo PC.

4.3.4 Pruebas a cable sano

Finalmente, en la Fig. 4.20 se demuestran los resultados que se obtienen al realizar pruebas a un cable sano. En esta prueba no se detecta ninguna falla en el cable (Fig. 4.21).

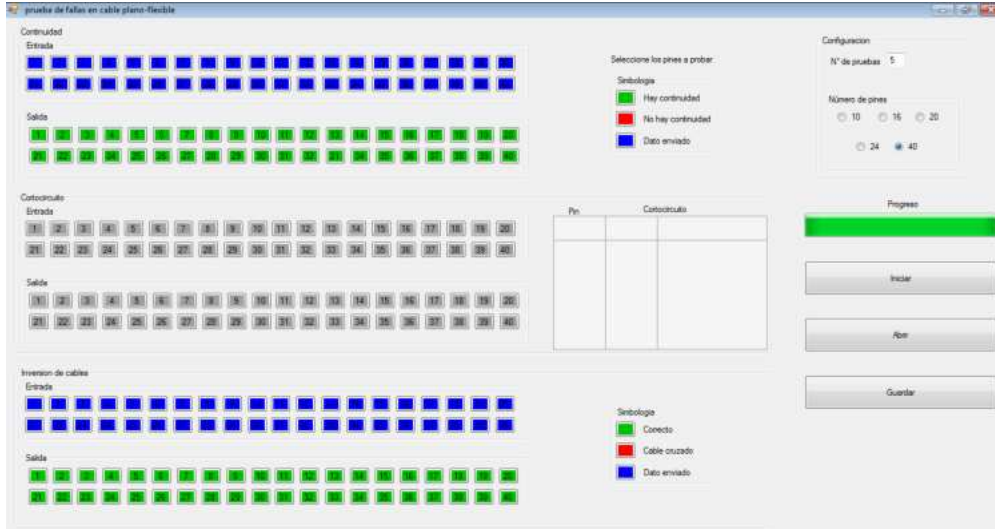


Fig. 4.20 Prueba de cortocircuito.



Fig. 4.21 Prueba de cortocircuito realizada en modo PC.

4.4 Resultados de pruebas en modo Portable

El uso de la interfaz de usuario por medio de una PC es un una gran ventaja. Sin embargo, algunas veces es necesario realizar pruebas en campo o en lugares donde no se pueda acceder a una PC, pero si con acceso a una toma de energía eléctrica o alguna batería portátil. Al presentarse estas situaciones se establece la configuración en modo portable moviendo el *switch* de modo en la posición Local como se muestra en la Fig. 4.22.



Fig. 4.22 Switch de modo en la posición de PC.

En esta sección se muestran los resultados de cada una de estas pruebas realizadas al mismo cable presentado en la sección 4.3 conociendo previamente sus fallas.

Para la selección del tipo de prueba que se desea realizar, basta con presionar el botón correspondiente al tipo de prueba que se encuentra en la sección de Tipo de prueba del prototipo (Fig. 4.23).



Fig. 4.23 Sección de tipo de prueba, se muestran indicadores luminosos y botones de pruebas.

Una vez seleccionada el tipo de prueba a realizar, oprimir el botón de “Iniciar prueba” para dar inicio al análisis del cable para detectar la falla seleccionada. Una vez que haya finalizado la prueba los resultados obtenidos se mostrarán en el display LCD.

A continuación se muestran los casos que se pueden obtener al realizar los tres tipos de pruebas e incluyendo el resultado al hacer pruebas a un cable sano

4.4.1. Pruebas de Continuidad

Para las pruebas de continuidad en modo local, los resultados son observados mediante el display LCD indicando la cantidad de fallas detectadas y las posiciones de dichos conductores (Fig. 4.24).



Fig. 4.24 Prueba de continuidad en modo Portable.

En la Fig. 4.25 se muestra la vista superior del prototipo tras realizar la prueba, mientras que la información desplegada en el display indica que se presentaron dos fallas en los pines 1 y 2.



Fig. 4.25 Resultado de prueba de Continuidad en modo Portable.

4.4.2. Pruebas de Cortocircuito

En la Fig. 4.26 se presenta el prototipo una vez finalizada la prueba de cortocircuito y en la Fig. 4.27 se observan los resultados obtenidos; la información desplegada indica una falla de cortocircuito en las posiciones 20 y 21 del cable.



Fig. 4.26 Prueba de cortocircuito en modo Portable.



Fig. 4.27 Resultado de prueba de Cortocircuito en modo Portable.

4.4.3. Pruebas de Cables cruzados

Al igual que las pruebas anteriores, en la Fig. 4.28 se muestra la vista superior de la prueba de Cables cruzados realizada en modo Portable. En la Fig. 4.29 se muestra la información obtenida del análisis al cable que presenta una falla de cables cruzados entre los conductores 39 y 40.



Fig. 4.28 Prueba de Cables cruzados en modo Portable.



Fig. 4.29 Resultado de prueba de Cables Cruzados en modo Portable.

4.4.4. Pruebas en un cable sano

Al igual que las pruebas ya descritas anteriormente, también es importante mostrar el resultado que se obtiene al realizar pruebas a un cable que no presenta ninguna falla en sus conductores. En la Fig. 4.30 se muestra el equipo después de realizar una prueba a un cable sano.



Fig. 4.30 Prueba de Cables cruzados en modo Portable.

En el display se puede observar que no existe falla y por lo tanto en la posición se obtiene el texto “NA” debido a la inexistencia de alguna falla. Este resultado se mostrará siempre que no se detecte falla en el cable bajo prueba sin importar el tipo de prueba que se le aplica.



Fig. 4.31 Resultado de prueba de Cable sano en modo Portable.

4.5. Matriz de confusión

Para presentar de mejor manera el correcto funcionamiento del prototipo, se realizaron las cinco pruebas a los diferentes cables por falla. La matriz de confusión mostrada en la Tabla 6 muestra el resultado de estas pruebas realizadas obteniendo

una eficiencia del 100%. Como se observa ninguna prueba es diagnosticada en otra condición. Por ejemplo, las cinco pruebas de falla de continuidad fueron clasificadas como fallas de continuidad. Los mismos resultados son obtenidos en modo local y modo PC para los distintos cables (diferente número de pines).

Tabla 6 Matriz de confusión a pruebas realizadas a cables.

	Prueba de Continuidad	Prueba de Cortocircuito	Prueba de Inversión de cables	Prueba a Cable Sano
Falla de Continuidad	5			
Falla de Cortocircuito		5		
Falla de Inversión de cables			5	
Cable sano				5
Eficiencia: 100%				

4.6. Conclusiones

En este proyecto se desarrolló un prototipo enfocado a realizar pruebas eléctricas para detectar fallas en cables plano-flexibles de manera rápida y eficiente debido a la cantidad de conductores que este tipo de cables puede presentar. Las fallas detectables son: falla de continuidad, falla de cortocircuito y falla ocasionada por cables cruzados. Estas fallas son analizadas ya que representan un problema en la industria a la cual se desea atender con este trabajo.

Se desarrolló una interfaz de usuario que permite la visualización de los resultados obtenidos gracias a un microcontrolador Arduino Mega que se encarga de obtener la información del estado del cable utilizando circuitos de acondicionamiento formados por registros de cambio 74HC959 y Multiplexores CD4021, los cuales permiten la expansión del número de salidas y entradas hacia el microcontrolador, permitiendo así la capacidad de aceptar cables con cierto número de conductores superior a las permitidas por el microcontrolador.

Este proyecto cuenta también con una configuración en modo portable que permite el uso del proyecto sin necesidad de estar conectado hacia una PC,

mostrando los resultados obtenidos a través de un display LCD. El mismo prototipo cuenta con elementos indicadores para conocer también el estado del análisis, así como indicadores para indicar el modo seleccionado (PC o Portable) y el tipo de prueba que se está realizando.

El hecho de desarrollar el software en Visual Studio C++ y Arduino permite el acceso al código de forma gratuita, lo cual permite su reconfigurabilidad y adaptabilidad a otros proyectos sin la necesidad de una inversión fuerte o completa, ya que en muchos casos el software de los equipos comerciales pierde vigencia y/o pasa a ser obsoleto ya que no permite su actualización y adaptación. Por otro lado, los circuitos desarrollados son modulares y pueden expandirse a un número mayor o menor de canales en caso de ser necesario.

Como prospectivas del proyecto, queda abierto a la modificación del hardware para que sea capaz de ser compatible con otros tipos de conectores ya que en un cable no solo cambia su composición sino también la forma de su conector, ejemplo: cuadrado, circular, diferentes orientaciones de los pines, etc. Para atender esta necesidad, se vislumbra la posibilidad de desarrollar un sistema de conectores genéricos que por un lado tengan una base universal y por otra tengan el conector hembra o macho requerido por el conductor bajo prueba. También queda la posibilidad de mejorar el despliegue de resultados mostrados en el display LCD al realizar las pruebas en el modo portable, permitiendo obtener más información después de haber realizado las pruebas correspondientes, así como incluir algún generador de reportes de pruebas, por ejemplo a través de un documento que se pueda almacenar e imprimir.

Referencias

- Angamarca, M., & Geovanny, D. (2015). Construcción del prototipo de una ruteadora cnc para la fabricación de pcbs con arduino (Bachelor's thesis).
- Arduino (2018). <http://arduino.cl> , consultado en Octubre 2018.
- Castañeda Pozo, H. R. (2001). *Manual del control de calidad en la fabricación de conductores eléctricos aplicación a la Empresa Fabricables* (Doctoral dissertation, Quito: EPN, 2001.).
- Crespo Vargas, E. M. (2015). Diseño e implementación de un dispositivo conversor con recepción Usb, Ethernet y RS-232 a enviar mediante Ethernet, Wi-Fi y Usb con una tarjeta Raspberry Pi; para el monitoreo de los equipos del Laboratorio de circuitos electrónicos de la Universidad de las Fuerzas Armadas-ESPE Extensión Latacunga (Bachelor's thesis, Universidad de las Fuerzas Armadas ESPE Extensión Latacunga. Carrera de Ingeniería en Electrónica e Instrumentación).
- FFC (2018). https://es.wikipedia.org/wiki/Cable_plano_flexible, consultado en Octubre 2018.
- García, M. (2013). Planeación y control del mantenimiento en la línea de producción KU-400. Licenciatura, Universidad Tecnológica de Querétaro.
- MSDN (2018). <https://msdn.microsoft.com/es-mx/library/60k1461a.aspx> , consultado en Octubre 2018.
- Villarroel Gonzalez, L. R., & Montalvo Yépez, C. A. (2010). Aplicación de la metodología MSF V4. 0 a la definición e implementación de arquitecturas orientadas a objetos en Visual Studio. Net 2005, caso práctico G5 Sharing Files (Bachelor's thesis, SANGOLQUÍ/ESPE/2010).
- Visual Studio (2016). <https://msdn.microsoft.com> , consultado en Octubre 2018.
- Icintracom (2016).<http://www.icintracom.com>. consultado en Noviembre 2018.
- CAMI Research Inc (2018). <https://www.camiresearch.com>, consultado en Noviembre 2018