

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería Electrónica.



Diseño de un sistema de comunicación para diagnóstico simultáneo de tarjetas de circuitos impresos.

Informe del Proyecto de Graduación para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura.

Stiven Sánchez Carranza.

Cartago, Noviembre de 2015.

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.


----- 2714-156 -----

Stiven Sanchez Carranza.

ID: 2-0714-0156

Cartago, 11 de Noviembre del 2015.

INSTITUTO TECNOLOGICO DE COSTA RICA

ESCUELA DE INGENIERIA ELECTRONICA

PROYECTO DE GRADUACIÓN

ACTA DE APROBACIÓN

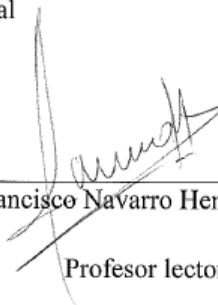
Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal



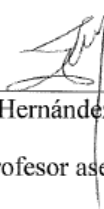
Ing. Hayden Phillips Brenes

Profesor lector



Ing. Francisco Navarro Henríquez

Profesor lector



Ing. Marvin Hernández Cisneros

Profesor asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica

Cartago, 11 de noviembre de 2015

Resumen

Este proyecto de graduación presenta el trabajo realizado para el diseño de un sistema de pruebas simultáneas para tarjetas de circuito integrado de la familia Flowserve; el cual tiene como objetivo brindar a los operadores del área de test de la empresa General Microcircuits de Costa Rica el resultado del diagnóstico general realizado a las respectivas tarjetas, indicando si ha existido un fallo durante la fabricación de las tarjetas; de ser así se procede con la reparación correspondiente.

Se desarrolló un sistema dentro del ámbito de proyectos internos de la empresa que busca el mejoramiento de la producción en el área de test, la cual hasta el momento presenta grandes inconvenientes en momentos críticos de la fabricación en los que se deben probar varias tarjetas al mismo tiempo, convirtiéndose esta área en un cuello de botella para la manufactura total de tarjetas. El proyecto se desarrolló mediante la consigna de hacer el área de test lo más eficiente posible en cuanto a cantidad de tarjetas probadas por operador en un tiempo dado, tomando en cuenta además los sistemas presentes en la actualidad en el área, los cuales fueron diseñados para la prueba de estas tarjetas de manera individual. Basado en estos aspectos se planteó un método de adquisición más eficiente, y que tomará en cuenta las variables ya conocidas en los antiguos aparatos de pruebas con la finalidad de no modificar los diseños de las tarjetas para el cliente.

El proyecto adicionalmente debía involucrar sistemas de adquisición de datos analógicos, así como la comunicación inalámbrica de los sistemas de pruebas con su respectiva computadora para facilitar el transporte de estos sistemas alrededor de la planta de producción, desde el computador el sistema recibe la secuencia programada de test basada en Labview así como una interfaz de operador tanto en el sitio como en el computador.

Por lo tanto, en el presente documento se describe el proceso de diseño y verificación del sistema que se propuesto.

Palabras clave: Labview, Modbus, RTU. Tarjeta de circuito impreso, adquisición de datos analógicos

Abstract

This paper presents the Project of graduation related of the design a simultaneously test fixture for the in circuit boards of the costumer FLOWSERVE. The general objective was give information to the manual operators of General Microcircuits of Costa Rica about the functional test of the boards. The system show them if exist any fail in the manufacturing production. So the operator can rework the board and solve the failure.

The system was developed in the area of internal projects of the company, and the idea was improve the production in the Test Area. Sometimes this area suffers saturation of production because it didn't have many fixtures for test such boards at the same time. So this area promoted delays in the production.

The project consisted to make the test area as efficient as possible, using a new method of data acquisition more efficiently than the actual. This method involved the test of 4 boards at the same time, using a slave system to access at which one of these. Also the system has a new programming based on Labview to make the diagnostics and a user interface for the operator.

Also the project involve the RF communication between the computer and the fixture using a 2.4 Ghz transceiver. That for facilitate the movement of the fixture around the production plant.

Finally this paper explain the process of design and verification of the system proposed.

Keywords: Labview, Modbus, RTU, in circuit Board, data acquisition.

ÍNDICE GENERAL

Capítulo 1. Introducción.	9
1.1. Entorno del problema.	9
1.2. Procedimiento para la ejecución del proyecto.	10
Capítulo 2. Meta y objetivos.	12
2.1 Meta.	12
2.1 Objetivo general.	12
2.2 Objetivos específicos.	12
Capítulo 3. Marco teórico.	14
3.1 Sistema actual de obtención de datos.	14
3.1.1 Bus de datos.	16
3.1.2 Protocolo de comunicación SPI.	17
3.1.3 NI USB 485.	18
3.1.4 RS232 Converter y DB9.	20
3.2 Sistema de adquisición de datos (DAQ).	21
3.2.1 NI SCXI 1000 Chasis.	22
3.2.2 NI SCXI 1600	23
3.2.3 SCXI 1125	25
3.2.4 SCXI 1313A bloque terminal.	28
3.2.5 NI MAX como programa de configuración para sistemas DAQ.	29
3.2.6 Transmisión Inalámbrica.	30
3.3 Protocolo de comunicación.	32
3.3.1 MODBUS.	32
3.3.2 Transmisión ASCII mediante Modbus.	36
3.3.3 Transmisión RTU mediante Modbus.	37

3.3.4	Descripción de funciones del protocolo Modbus.....	37
3.4	Programación del hardware.	42
3.4.1	Programación en Labview.	43
Capítulo 4.	Diseño de la solución.	45
4.1	Propuestas de solución.....	45
4.1.1	Acceso remoto y sistema con simultaneidad.	45
4.1.2	Separación de sistemas y comunicación inalámbrica.	46
4.1.3	Comparación de sistemas.....	47
4.2	Desarrollo de la propuesta seleccionada.	49
4.2.1	Diseño de conexión de equipos e interfaz de operador.....	50
4.2.2	Escogencia de microprocesador.....	54
4.2.3	Diseño de pruebas programadas.	56
4.2.4	Diseño de secuencias de Pruebas.	59
Capítulo 5:	Resultados y análisis.	66
Capítulo 6.	Conclusiones y Recomendaciones.	78
Referencias Bibliográficas.	81
Apéndice A.	Diagramas de Casos.....	85
Apéndice B.	Diagrama del Circuito.....	87
Apéndice C.	Programación en Labview.	88

ÍNDICE FIGURAS

Figura 3. 1. Diagrama general del sistema actual de diagnósticos.	15
Figura 3. 2. Bus de datos.	16
Figura 3. 3. Protocolo de comunicación SPI [1].	18
Figura 3. 4. Interfaz de comunicación RS485 full duplex [3].	19
Figura 3. 5. Conector DB9 encargado de la transmisión RS232 [5].	20
Figura 3. 6. Diagrama de canales para bloque terminal [12].	26
Figura 3. 7. Control de dispositivos DAQ.	29
Figura 3. 8. Configuración de canal.	30
Figura 3. 9. Generación del mensaje [16].	33
Figura 3. 10. Diagrama de tiempos en modo RTU [18].	37
Figura 3. 11. Barra de herramientas DAQmx. [21].	43
Figura 3. 12. Barra de herramientas Modbus. [21].	44
Figura 4. 1. Sistema de adquisición inalámbrico basado en SPI.	46
Figura 4. 2. Sistema Externo.	46
Figura 4. 3. Diagrama de conexiones Analog Board.	50
Figura 4. 4. Diagrama de conexiones DDC.	51
Figura 4. 5. Interfaz de operador.	53
Figura 4. 6. Diagrama interno del sistema.	53
Figura 4. 7. Listado de límites por cargar.	57
Figura 4. 8. Máquina de estados para la secuencia Analog Board.	61
Figura 4. 9. Máquina de estados para la secuencia DDC Board.	61
Figura 4. 10. Modificación para introducir diagnósticos.	63
Figura 4. 11. Acondicionamiento de señal por divisor de tensión.	64
Figura 5. 1. Petición de diagnósticos.	66
Figura 5. 2. Cambio de esclavos.	67
Figura 5. 3. Secuencia Holding Registers, en el registro 2021, petición.	70
Figura 5. 4. Secuencia Diagnostics, sub-función 34, petición.	70

Figura 5. 5. Verificación de Errores caso 1 tarjeta sin errores.....	72
Figura 5. 6. Verificación de Errores caso 1 tarjeta, con error.....	72
Figura 5. 7. Verificación de Errores caso 1 tarjeta, con error.....	73
Figura A. 1. Diagrama general de secuencias DDC.....	85
Figura A. 2. Diagrama Verify MA, Analog Board.	85
Figura A. 3. Verify Voltage, Analod Board, Secuencia general.	86
Figura A. 4. Secuencia General para cada tarjeta.	86
Figura C. 1 Diagnostics.	88
Figura C. 2 GET E2, Holding Registers.	88
Figura C. 3 Set E2, Write Registers.	89
Figura C. 4 Espera de Peticion 1.....	89
Figura C. 5. Verificar límites.....	89
Figura C. 6. Verify Voltage secuencia general	90
Figura C. 7. Programa general.....	90

ÍNDICE TABLAS

Tabla 3. 1. Rangos de ganancia para las diferentes entradas.	24
Tabla 3. 2. Ganancia total del sistema según el modulo terminal utilizado.	27
Tabla 3. 3. Señales de entrada versus rangos de ganancia del bloque terminal.	28
Tabla 3. 4. Organización de señales Modbus.	33
Tabla 3. 5. Código hexadecimal de las funciones Modbus.	34
Tabla 3. 6. Trama de caracteres en ASCII.	36
Tabla 3. 7. Trama de caracteres en RTU.	37
Tabla 3. 8. Trama de estado de bobinas (coils).	38
Tabla 3. 9. Respuesta de trama de estado de bobinas (coils).	38
Tabla 3. 10. Trama de Read input Status.	39
Tabla 3. 11. Trama de diagnóstico.	41
Tabla 3. 12. Sub-funciones de diagnóstico.	41
Tabla 4. 1. Costos de la primera propuesta	48
Tabla 4. 2. Costos de la segunda propuesta.	49
Tabla 4. 3. Elementos utilizados en la interconexión del sistema.	54
Tabla 4. 4. Requerimientos mínimos del microprocesador.	55
Tabla 4. 5. Instrucciones para el diagnóstico Analog Board.	60
Tabla 4. 6. Instrucciones para el diagnóstico DDC.	60
Tabla 5. 1. Funciones para la secuencia Verify Output Voltage.	68
Tabla 5. 2. Función Holding Registers para el registro 2021.	68
Tabla 5. 3. Función Holding Registers para el registro 2021, respuesta.	69
Tabla 5. 4 Beneficio en Tarjetas	76

Capítulo 1. Introducción.

El proyecto consistió en el diseño de un sistema de pruebas simultáneas para las tarjetas de circuito integrado de la familia Flowserve C597 y B563 utilizando sistemas inalámbricos de adquisición de datos, así como programación mediante Labview y Arduino para el desarrollo de estas pruebas.

1.1. Entorno del problema.

La empresa General Microcircuits de Costa Rica se dedica a la construcción y manufactura de tarjetas de circuito impreso, el desarrollo de estas tarjetas conlleva una serie de procesos, entre los que destaca el área de test, la cual se encarga de diagnosticar deficiencias en las tarjetas, con la finalidad de solucionar posibles errores de producción de manera inmediata y evitar que estas tarjetas sean enviadas al cliente con defectos eléctricos. Esta empresa transnacional se encuentra ubicada en la provincia de Alajuela, en el sector del Coyoil, específicamente en la zona franca Bes edificio 23.

El proyecto surge como una contrapuesta a diversos retrasos generados en la producción, ya que a pesar que muchas de estas tarjetas se generan rápidamente en las otras áreas de la empresa, estas se ven retenidas en gran medida en test debido a que el sistema actual de prueba de las tarjetas solo permite diagnosticar una tarjeta de manera individual.

Actualmente los sistemas de test que se encuentran en la empresa GMI.MFG Costa Rica permiten la evaluación de la funcionalidad de un circuito cuando el proceso de creación del mismo se ha concluido, esta función es de suma importancia ya que logra identificar diferentes fallos en cualquiera de las áreas de producción, así como de los materiales utilizados para el ensamble del mismo, permitiendo de este modo evitar devoluciones de parte de los clientes una vez que el producto ha sido enviado.

Parte de estos instrumentos de prueba son los creados por FLOWSERVE; cliente actual, los cuales proporcionan una bandeja de pruebas para los diferentes productos diseñados para dicha compañía entre los que destacan, Analog board C597, Devicenet B524, DDC B563 y Output board B503. Estas placas son diagnosticadas en un mismo sistema, el cual posee la programación y estructura física para realizar la prueba de cada una de ellas de manera individual.

La prueba de estas tarjetas se realiza desde una computadora de escritorio, la cual posee cada uno de los programas involucrados para el banco de ensayos del sistema, estos programas se encuentran basados en dos entornos de programación Test Stand y Labview ambos de National Instruments.

La idea fundamental del proyecto fue primeramente investigar los antiguos sistemas de prueba para identificar las variables a diagnosticar, así como los protocolos de comunicación utilizados para la prueba y el desglose de los programas específicos para cada tarjeta. Con estos datos se procedió al desarrollo de posibles soluciones para disminuir la latencia y aumentar el número de tarjetas diagnosticadas por un mismo operario, de esta manera se evaluó la solución que presenta mejores características y se procede al diseño de la misma.

Por lo tanto, fue de fundamental importancia el acceso al equipo de diagnóstico actual que estaba disponible en GMI, ya que permitió visualizar las variables de prueba, así como sus rangos de operación, además de la experimentación con el mismo para de esta manera brindar una solución al problema que se propuso. Además, la empresa brindó acceso a las plataformas de programación necesarias si la solución lo requería, de igual manera todas las facilidades en cuanto al uso de licencias y herramientas computacionales de acuerdo al formato de solución que el desarrollador establecía.

1.2. Procedimiento para la ejecución del proyecto.

Se procedió en primera instancia con la investigación del sistema antiguo con la finalidad de comprender todas las variables implicadas en la prueba ya que de estas depende la simultaneidad del nuevo sistema. Además se investigaron los protocolos de comunicación utilizados para la transferencia de datos al sistema.

Con estos datos claros se procedió al análisis de la programación existente así como a la modificación de la misma para obtener la prueba simultánea deseada desde la misma computadora para todos los instrumentos de prueba. De esta manera se generaron una serie de posibles soluciones, a partir de las cuales se selecciona la alternativa que genera mayores beneficios y además presenta un menor costo para la empresa.

Con esta solución como meta se procedió al diseño de los procesos de multiplexión de datos provenientes tanto para los sistemas de adquisición de datos como los protocolos de comunicación pertinentes a cada tarjeta testada. Se procede adicionalmente al desarrollo de la programación

paralela de los sistemas, con la finalidad de evidenciar disminución de los tiempos de prueba para las placas.

Se investigó en cuanto a los sistemas de adquisición de datos utilizados, con la finalidad de identificar sistemas con mejores características, así como la búsqueda de comunicación inalámbrica para los mismos. De esta manera se seleccionaron nuevos módulos para la adquisición de datos y se realizó la unión de estos al nuevo sistema.

Finalmente, se desarrolló una rutina general para la prueba simultánea de todos los sistemas de prueba inalámbricos, así como su respectiva interfaz de usuario Labview.

Capítulo 2. Meta y objetivos.

2.1 Meta.

Diseñar un sistema de pruebas simultáneas para tarjetas de circuito impreso, que diagnostique cuatro tarjetas iguales simultáneamente, que además permita trasladar los instrumentos de prueba a los diferentes sectores de la planta y comunicarlos de manera inalámbrica.

2.1 Objetivo general.

Diseñar un sistema de comunicación inalámbrica con simultaneidad para diagnóstico de tarjetas FLOWSERVE, de manera que estas puedan ser probadas desde cualquier punto de la planta de producción de forma paralela.

Indicador.

Abarcar un diseño completo del sistema de pruebas simultáneo, el cual incluya un método eficiente de separación y comunicación de las placas de prueba que minimice los costos, y disminuya los periodos de prueba por tarjeta.

2.2 Objetivos específicos.

Diseñar un método de separación para el sistema actual, de manera que se puedan comunicar las tarjetas externas con un sistema central de mando.

Indicador.

Visualización de la comunicación exitosa de cada uno de los sistemas planteados de manera independiente mediante una simulación teórica del proceso de pruebas solicitudes y muestra de resultados.

Desarrollar una programación orientada a crear un sistema inclusivo para el manejo de pruebas simultáneas mediante Labview y Arduino.

Indicador.

Creación de diagramas de flujo para la formación de cada una de las secuencias de pruebas, así como para la secuencia conjunta, además de la prueba de funcionamiento paralelo de los sistemas, así como una interfaz de usuario sencilla para el operador.

Crear un sistema de multiplexión de datos que permita la escalabilidad del diagnóstico a varias placas simultáneamente e incluya comunicación inalámbrica de las mismas hacia la computadora central.

Indicador.

Diagramas esquemáticos que muestren de manera de simulación la función del sistema de multiplexión de datos para los respectivos protocolos de comunicación utilizados.

Capítulo 3. Marco teórico.

3.1 Sistema actual de obtención de datos.

Las pruebas para las placas de manufacturas son de suma importancia ya que se encargan de verificar la funcionalidad y asegurar la calidad de los productos terminados, evitando de este modo que posibles fallos puedan llegar al cliente final. Estos diagnósticos pueden incluir un amplio rango de pruebas eléctricas, mecánicas o ambientales ya sean manuales o automatizadas.

Actualmente los diagnósticos realizados en las pruebas de funcionamiento para las tarjetas Flowserve conllevan una serie de requisitos de procesamiento los cuales son cubiertos por el sistema de pruebas actual. Es necesario ahondar un poco más en estos requisitos así como en la construcción del sistema, con la finalidad de evidenciar a fondo los requerimientos que presenta el nuevo sistema de pruebas simultáneas.

El sistema de pruebas, como se mencionó anteriormente, posee la capacidad de escoger entre 5 tipos de tarjetas por probar, sin embargo esto se hace de manera individual. Estas tarjetas son: Analog board (C597), Devicenet (B524), Output Board (B503), DDC (B563), Power Board (B823). Cada una de estas tarjetas cumple una función diferente en un ensamble final del sistema, y son controladas en su totalidad por una placa maestra llamada Main Board (B863).

La placa maestra o main board se encarga de girar las señales correspondientes al proceso que se desea realizar mediante las tarjetas en prueba, es decir, se encarga de la activación de otros módulos, señales de habilitación, señales de control de apertura de relays entre otros aspectos relacionados, además se encarga de la comunicación con cada una de estas tarjetas mediante protocolos SPI que luego se conocerán más a fondo.

El sistema cuenta con algunas tarjetas de adquisición de datos programables las cuales son controladas mediante el software NI Max, así como la programación de Labview, basadas en diferentes protocolos de comunicación conocidos como R232 y R485. De esta manera se puede evidenciar el diagnóstico de estas placas como un gran proceso de transferencia de datos, uso de protocolos, y adquisición de resultados en los que participan tanto agentes de hardware como de software. Para una mejor claridad de la conformación del sistema de diagnósticos se proporciona la Figura 3.1.

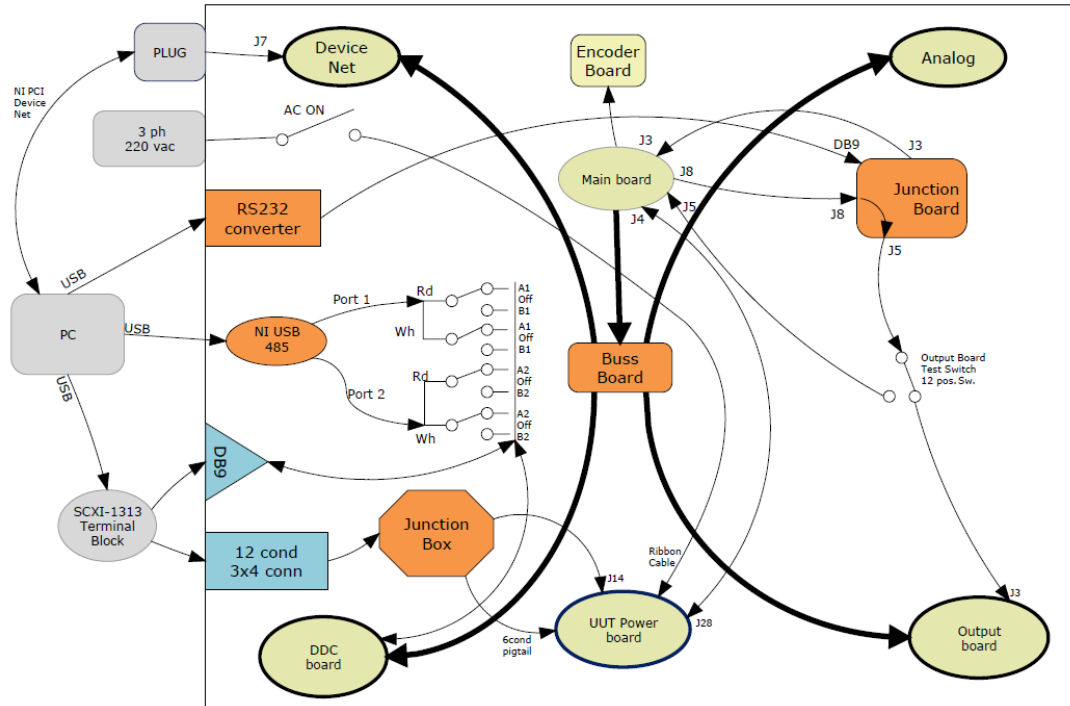


Figura 3. 1. Diagrama general del sistema actual de diagnósticos.

Tal como lo describe el diagrama anterior todas las tarjetas se encuentran conectadas por un mismo bus de datos, el cual se muestra en la imagen como Buss board mediante este, se realiza la comunicación por medio de los diferentes protocolos SPI con los que la tarjeta cuenta, así se envían una serie de comandos de parte de main board que indican a las placas diagnosticadas las funciones a realizar para responder correctamente a las pruebas programadas para las mismas.

Por otra parte se pueden visualizar una serie de interruptores, los cuales son activados de forma manual por el operador según las indicaciones brindadas por la interfaz de usuario que brinda el computador.

El sistema cuenta con tensión de entrada trifásica de 220 V la cual ingresa directamente tras pasar por un supresor de picos a una de las tarjetas, esta se encarga de transformar estas tensiones a valores más manejables para las demás tarjetas.

La adquisición de datos es realizada por medio de tarjetas programables del proveedor National Instruments, especializadas para los rangos de tensión requeridos, en su mayoría de datos analógicos. Además de protocolos de comunicación UART vía USB con el computador.

3.1.1 Bus de datos.

La transmisión interna de datos del sistema es uno de los aspectos que se debe tomar en cuenta para el desarrollo de la prueba simultánea, ya que este es un punto clave en cuanto a la capacidad de escalabilidad del sistema.

La placa maestra o main board cuenta con un conector universal para la comunicación con las diferentes placas diagnosticadas, por el pasan tanto datos de control como señales de alimentación y protocolos de comunicación; lo cual se muestra figura 3.2.

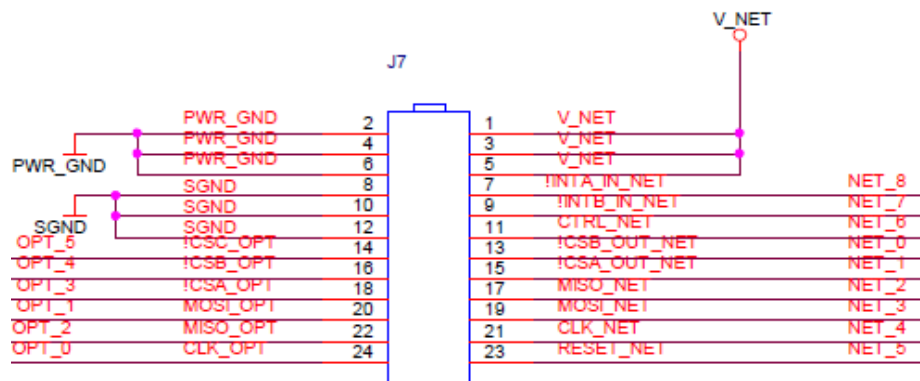


Figura 3. 2. Bus de datos.

En la figura anterior se muestra como se utilizan las diferentes líneas del conector, las líneas 2, 4, 6, 8, 10 y 12 son dedicadas a la utilización de las diferentes tierras de los sistemas con un valor de 0V. Las líneas 1, 3 y 5 son quienes brindan una tensión de 5V a las demás tarjetas excepto a la tarjeta de transformadores Power Board.

Por otra parte las líneas 14, 16 y 18 se encargan de seleccionar la tarjeta a la que se desea hacer referencia, como su nombre lo describe “!CSB_opt” indica que es una salida para habilitar un determinado sistema, además indica que la señal es negada. Algo semejante es lo que sucede con las líneas 13 y 15 las cuales son utilizadas para otros sistemas pero mantienen la misma función.

Las líneas 7 y 9 cumplen una función un poco diferente ya que encargan de indicar una interrupción al sistema, si estas están activas generan un paro en el proceso para ingresar un dato específico, estas líneas también se encuentran en lógica negativa.

La línea 23 se encarga de dar reinicio a la tarjeta en caso de ocurrir algún inconveniente o ser esto deseado de parte del programa de pruebas.

Finalmente nos encontramos con las líneas correspondientes a los 2 protocolos de comunicación utilizados por la main board para el traslado de información, en este caso se evidencian las líneas 20,22 y 24 para el primero de estos y las líneas 17,19 y 21 para el segundo. Cabe destacar que ambos protocolos son iguales, sin embargo el hecho en que se encuentren en diferentes líneas indica la capacidad del sistema de comunicarse con 2 tarjetas mediante el mismo protocolo de manera independiente. En la siguiente sección se estudiará más a fondo acerca de estos protocolos de comunicación.

3.1.2 Protocolo de comunicación SPI.

SPI es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos consta de un bus de tres líneas, sobre el cual se transmiten paquetes de información de 8 bits. Cada una de estas tres líneas porta la información entre los diferentes dispositivos conectados al bus. Cada dispositivo conectado al bus puede actuar como transmisor y receptor al mismo tiempo, por lo que este tipo de comunicación serial es full dúplex [1].

Cada una de las líneas posee una función específica la cual se encuentra ligada con su nombre, los nombres de las señales son definidos dentro del estándar como: SCLK reloj del bus, MOSI (Master Output Slave Input) salida de datos del maestro a esclavo, MISO (Master Input Slave Output) salida de datos del esclavo hacia el maestro [1].

Dependiendo del número de esclavos que se encuentren presentes en un bus de datos será necesaria la utilización de líneas adicionales para habilitar la selección de cada uno de estos por lo que se considera que el sistema permite la ubicación de múltiples esclavos de manera sencilla.

La totalidad de los datos transferidos son sincronizados por la línea de reloj del protocolo. Además cabe destacar que tan solo un bit es transferido por cada ciclo de reloj.

En la figura 3.3 se presenta un ejemplo de este tipo de protocolo de comunicación.



Figura 3. 3. Protocolo de comunicación SPI [1].

3.1.3 NI USB 485.

El sistema Ni USB 485 se encarga de transformar un puerto Universal serial bus (USB) en un puerto asincrónico para la comunicación serial con diferentes dispositivos, este utiliza la comunicación RS 485 como base de la transferencia de datos, este puede utilizar estos puertos estándar para diferentes aplicaciones, además se puede complementar mediante la escritura de aplicaciones como NI VISA en lo cual se profundizará más adelante.

El protocolo RS 485 se encuentra definido como un sistema en bus de transmisión multipunto diferencial, es ideal para transmitir a altas velocidades sobre largas distancias. Esta interfaz fue publicada como estándar en 1983 en donde se especifican las características eléctricas de un circuito de comunicación digital de salida ya sea el estado “0” o “1” lógicos mediante una señal diferencial más una señal de alta impedancia o “tercer estado” con lo cual es posible conectar más de un emisor en la red, motivo por el cual, es muy útil en redes locales industriales a dos hilos [2].

Permite la conexión de hasta 32 emisores con 32 receptores en transmisión doble simultánea full dúplex capaz de enlazar procesadores de comunicación principal (master) con procesadores subordinados (slaves) cuyo funcionamiento (acceso priorizado) está definido por los mismos arreglos topológicos de las redes de datos. Además presenta mayor resistencia a la interferencia electromagnética y mayor velocidad de transmisión que con la norma RS232 [2].

En cuanto a longitud y rapidez de transmisión, este protocolo presenta mejores condiciones que sus competidores permitiendo llegar hasta a 4000 pies de longitud y una velocidad de 10 megabits/segundo. La razón por la que RS-485 puede transmitir a largas distancias, es porque utiliza el balanceo de líneas. Cada señal tiene dedicados un par de cables, sobre uno de ellos se

encontrará un voltaje y en el otro se estará su complemento, de esta forma, el receptor responde a la diferencia entre voltajes. En la figura 3.4 se muestra este tipo de conformación de líneas.

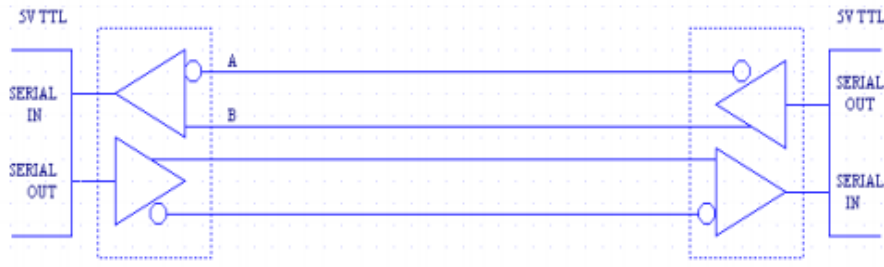


Figura 3. 4. Interfaz de comunicación RS485 full duplex [3].

Las interfaces típicas RS-485 utilizan una fuente de +5 Volts, pero los niveles lógicos de los transmisores y receptores no operan a niveles estándares de +5V o voltajes lógicos CMOS. Para una salida válida, la diferencia entre las salidas A y B debe ser al menos +1.5V. Si la interface está perfectamente balanceada, las salidas estarán desfasadas igualmente a un medio de la fuente de Voltaje [3].

En el receptor RS-485, la diferencia de voltaje entre las entradas A y B necesita ser 0.2V. Si A es al menos 0.2V más positiva que B, el receptor ve un 1 lógico y si B es al menos 0.2v más positivo que A, el receptor ve un 0 lógico. Si la diferencia entre A y B es menor a 0.2v, el nivel lógico es indefinido. Si esto ocurre habría un error en la transmisión y recepción de la información [3].

La diferencia entre los requerimientos del Transmisor y el Receptor pueden tener un margen de ruido de 1.3V. La señal diferencial puede atenuarse o tener picos de largo como de 1.3v, y aun así el receptor verá el nivel lógico correcto. El margen de ruido es menor que el de un enlace RS-232, no hay que olvidar que RS-485 maneja señales diferenciales y que cancela la mayoría del ruido a través de su enlace [3].

Así se presenta que el sistema NI USB 485 finalmente se convierte en un método de conversión de datos del bus USB al sistema full dúplex RS485 con el cual, como se observó con los datos anteriores presenta buenas características para la transferencia de datos con una serie de sistemas esclavos.

El elemento utilizado en el sistema de diagnósticos es proporcionado por National Instruments, entre sus especificaciones técnicas se indica que es compatible con cualquier aplicación que utiliza

puertos COM en PCs con Windows Vista (32 y 64 bits) /XP/2000, además presenta Velocidades de transferencia estándares hasta 460.8 kb/s, FIFOs de transmisión y recepción hasta 128MB, cuenta con 2 puertos RS485 completamente funcionales, incluyendo líneas de protocolo de sincronización de hardware RTS/CTS, opera en Modo automático de control transceptor de 2 cables, seleccionable por software mediante fácil programación utilizando NI VISA [4].

3.1.4 RS232 Converter y DB9.

Este protocolo es el más comúnmente utilizado en la industria, El nombre del estándar paso de RS-232 a EIA-232 al igual que otros elementos de la norma original han cambiado su denominación. Es un estándar completo de comunicación, puesto que no sólo especifica los niveles de voltaje y señal, sino que además especifica la configuración de pines de los conectores y una cantidad mínima de información de control entre equipos.

Es un protocolo de comunicación serial asíncrono, esto es, que no tiene un orden de envío de datos entre los dispositivos, por lo que se vuelve necesario el cuidar la sincronización del envío para evitar pérdidas de información o fallos en la comunicación. Otra de sus características principales es ser un protocolo punto a punto, lo que significa que solamente permite la comunicación de un dispositivo con respecto a otro empleando una terminal de comunicación determinada. No permite la creación de redes. El protocolo hace uso de conectores de tipo DB-25, sin embargo, es común observar dispositivos con conectores de tipo DB-9, de 9 pines debido a su menor costo [5]. Tal y como se puede observar en la figura 3.5.

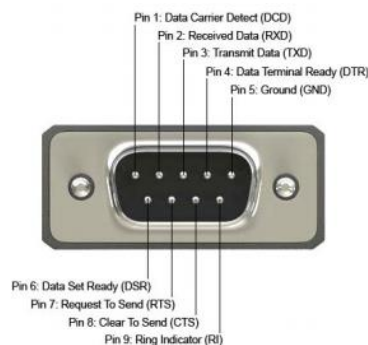


Figura 3. 5. Conector DB9 encargado de la transmisión RS232 [5].

La normativa define los rangos eléctricos del sistema como nivel alto a la salida del transmisor está definido como un voltaje entre +5 y +15 volts, mientras que un nivel bajo está definido como un voltaje entre -5 y -15 volts. La lógica del receptor fue diseñada para permitir un nivel de ruido de 2 volts. Así, un nivel alto para el receptor está definido en el rango +3 a +15 volts, mientras que un nivel bajo va desde los -3 a los -15 volts [6].

Este estándar también define un máximo slew rate o de máxima variación voltaje de 30[V/ μ s] para evitar el crosstalk, que es la inducción de las señales que viajan por un cable en los cables adyacentes. Las mínimas señales utilizadas en una comunicación bidireccional son: TD para transmitir datos y RD para recibir datos. Asimismo, si desea utilizarse control de flujo por hardware, se utilizan las de señales RTS (petición transmisión) y CTS (habilitado para transmitir). El control de flujo impide que un transmisor rápido sature a un receptor lento [6].

Los diferentes parámetros de la transmisión son programables, un caso es la velocidad que puede variar entre 50 y 19.200 baudios.

3.2 Sistema de adquisición de datos (DAQ).

Se entiende por adquisición de datos a la acción de medir variables, convertirlas a formato digital, almacenarlas en un computador y procesarlas en cualquier sentido. Este proceso necesita de una interfaz entre el mundo físico y el computador que se suele denominar como tarjeta de adquisición de datos [7].

Las tarjetas de adquisición deben poseer una serie de características que definen su funcionalidad y utilidad en respuesta a las variables que se desean analizar algunas de estas características se presentan a continuación [8].

- Especificación de entradas analógicas.
- Número de canales.
- Velocidad Máxima de muestreo.
- Resolución de los datos.
- Mecanismo de muestreo de las entradas analógicas.
- El bus de comunicación con la PC
- Aislamiento eléctrico del dispositivo
- Software controlador o driver para comunicarse con el equipo.

Una vez evidenciadas las capacidades mínimas necesarias se procede a la selección de la tarjeta adecuada a estas especificaciones.

El sistema actual de diagnósticos para tarjetas cuenta con una serie de tarjetas encargadas de la adquisición de los datos generados como respuesta a las demandas de los análisis predeterminados por la programación, estas señales pueden ser tanto de tipo analógico como digital, existen diferentes módulos que permiten la variación de parámetros al adquirir las señales deseadas, se estudiarán cada uno de estos módulos, así como la función que estos desempeñan.

En el sistema actual se encuentran presentes una serie de estos elementos, los cuales se estudiarán a continuación.

3.2.1 NI SCXI 1000 Chasis.

National Instruments cuenta con una serie de tarjetas que se encargan del sensado de señales tanto analógicas como digitales, con la finalidad de utilizar varios de estos módulos en un mismo sistema es cómo surge la necesidad de crear un sistema de encapsulado que soporte la colocación de estas varias de estas tarjetas.

El chasis SCXI aloja, alimenta y controla sus módulos SCXI y señales acondicionadas, además proporciona un medio de bajo ruido para la comunicación. La arquitectura única del chasis SCXI incluye el SCXIbus el cual envía señales analógicas y digitales y es el medio de comunicación entre módulos. El circuito de control del chasis administra este bus, al sincronizar el tiempo entre cada módulo y el dispositivo DAQ.

Con este sistema es posible escanear canales de entrada desde diversos módulos en distintos chasis a rangos de hasta 333 kS/s para cada dispositivo DAQ [9].

En este caso el sistema National Instruments utilizado posee la capacidad de alojar 4 tarjetas de adquisición, lo cual es de suma importancia debido a la capacidad de expandir el sistema a mayores cantidades de sistemas diagnosticados. Además la cantidad de chasis también puede ser escalable hasta a 16 chasis conectados en cadena, seleccionables mediante una dirección de ubicación en 4 bits.

En el sistema actual este chasis SCXI 1000 DC aloja 3 tarjetas principales las cuales también se estudiarán en esta sección y las cuales son SCXI 1600, SCXI 1125, SCXI 1313A.

3.2.2 NI SCXI 1600

El módulo de adquisición de datos USB SCXI 1600 adquiere datos y controla los módulos de acondicionamiento de señal instalados en el mismo chasis en el que reside, por lo cual convierte el chasis en un sistema completo de adquisición de datos, las señales acondicionadas de otros módulos terminales son dirigidas automáticamente al SCXI 1600 el cual digitaliza los datos y los transfiere al pc vía USB.

Este posee un sistema digitalizador de señales analógicas de 16 bits de resolución por lo que permite alta precisión en las mediciones realizadas, es capaz de controlar tanto la entrada como la salida de datos analógicos que son direccionados por otros módulos terminales [10].

Entre las características principales lo describen:

- 200KS/s para hasta 352 canales.
- 16-bit resolución en A/D.
- 10 V como rango de entrada en canales analógicos.
- USB 2.0 compatibilidad.
- Calibración Interna incluida.
- Fácil de configurar utilizando software.

Estas características muestran su capacidad de muestreo de datos una vez que estos ya han sido acondicionados.

Una de las características de limitación para esta tarjeta es su compatibilidad de sistemas Windows ya que solo se encuentra disponible para sistemas de 32 bits, lo cual genera alguna limitante en cuanto a capacidad de procesamiento, lo cual se refleja en eficiencia en cuanto a tiempo de muestreo.

El módulo SCXI-1600 tiene una ganancia programable por software para obtener el más alto nivel de precisión. Por lo que es posible acomodar el rango de señal de entrada con el rango de conversión del ADC mediante la modificación acertada de la ganancia. Específicamente el sistema cuenta con una gama de ganancias de entrada que rondan valores de 0,5, 1, 10, y 100 [11].

En la tabla 3.1 se muestran los rangos de estas ganancias de entrada y como estos generan una mayor resolución en la conversión de datos para el sistema.

Tabla 3. 1. Rangos de ganancia para las diferentes entradas.

Rango de configuración	Ganancia	Rango de entrada actual	Precisión
-5V a +5V	0.5	-10V a +10V	305.3uV
	1	-5V a +5V	152.6uV
	10	-500mV a +500mV	15.26uV
	100	-50mV a +50mV	1.53uV

Como se muestra en la tabla la resolución en cuanto a tensión censada aumenta de la misma manera que lo hacen los rangos de ganancia. En este caso el sistema actual maneja un rango de ganancia de 0.5 por lo cual la resolución corresponde a 305.3uV lo cual es bastante exacto debido a que los valores que se desean medir no son muy específicos.

Para calcular la máxima velocidad de escaneo para el sistema SCXI 1600 es necesario conocer la cantidad de canales a censar ya que esta velocidad es dependiente de los canales como lo muestra la fórmula siguiente.

$$velocidad\ de\ muestreo = \frac{1}{5\mu s * numero\ de\ canales}$$

Cabe destacar que el número de canales se refiere a los que se encuentran en el chasis.

El sistema funciona en modo multiplexado de manera que todos los canales de los demás módulos son multiplexados en un canal analógico único para el dispositivo DAQ. La operación de multiplexion se lleva a cabo en cada una de las otras tarjetas de adquisición no dentro del módulo 1600 ni en el chasis [11].

Cuando se escanea en modo de multiplexado el sistema mantiene una lista del orden de los canales que son inmediatamente prioridad a ser multiplexados según como fue programado el sistema y estos datos son dirigidos al módulo SCXI 1600 mediante el SCXI bus durante el escaneo de canales. Esta lista contiene los canales sin ser necesariamente almacenados en un orden físico específico, es decir este ordenamiento no es necesariamente secuencial, en cambio puede ocurrir el escaneo de varias veces en el mismo canal en la lista. Además la lista de escaneo posee un número arbitrario de canales por cada módulo de entrada presente en la misma, limitado a 352 canales por cada SCXI 1600 [11].

3.2.3 SCXI 1125

La tarjeta amplificador de aislamiento para 8 canales NI SCXI-1125 se encarga de brindar 300Vrms de aislamiento activo y filtrado de paso bajo para cada canal de entrada analógica. Cuenta con una arquitectura ideal para amplificación y aislamiento de fuentes con corrientes de entre 0 a 20 mA. Además este sistema permite la multiplexión de 8 canales de entrada a un solo canal para dispositivos DAQ.

El sistema cuenta también entre sus especificaciones con 12 opciones de programación de ganancia cuyos valores se encuentran en el rango entre 1 a 2000, así como 2 filtros programables entre valores igualmente de 4Hz a 10kHz todo esto para cada canal de entrada. Cada canal cuenta con un circuito externo de tierra lo cual es una buena opción para utilizarlo en la calibración [12].

La tarjeta permite la conexión de entradas analógicas de alta tensión, utilizando un terminal atenuador apropiado este valor podría rondar hasta los 1000Vdc. Algunas de estas terminales podrían ser:

- SCXI 1305
- SCXI 1313A
- SCXI 1320
- SCXI 1327
- SCXI 1328

Entre las anteriores destaca la utilizada por el sistema actual SCXI 1313A de la cual se comentará más adelante.

La función de aislamiento en la tarjeta SCXI 1125 colabora en funciones específicas como medir de manera precisa altas tensiones o bajas tensiones en las cuales están presentes voltajes de modo común, de esta manera se pueden proteger equipos sensibles conectados a este módulo.

El aislamiento permite además convertir una señal de alta tensión en un valor más bajo de manera que este pueda ser muestreada por otro sistema de adquisición de datos. Otra función es la de filtrado y ganancia de los valores de entrada, de esta manera se aumenta la resolución y precisión para la medición [12].

El sistema SCXI -1125 provee 2 filtros con respuesta de filtro Butterworth de cuatro polos, estos filtros pueden ser modificados entre los rangos de 4Hz a 10Khz, lo cual muestra que son filtro paso bajo, además de esto se puede realizar un filtro digital para complementar este sistema mediante la programación en Labview [12].

La multiplexación de este módulo depende de las características que se tienen para la tarjeta conectada a este, es decir si esta permite la entrada de más de un canal, el sistema envía la señal de entrada mediante el SCXIbus, de esta manera si el módulo de procesamiento es capaz de captar los 8 canales el sistema puede funcionar en forma paralela para estos, de no ser así se multiplexan los canales para proporcionar cada uno de los datos, esto se logra mediante una lista de canal.

El sistema permite ampliar su rango de medición de ± 5 V a ± 300 V utilizando terminales de atenuación, estas terminales pueden extender la ganancia del sistema hasta 1:100, en la siguiente tabla se muestran casos de esta atenuación [12].

Tabla 3. 2. Ganancia total del sistema según el módulo terminal utilizado.

Ganancia Total	Rango de entrada(V)	Ganancia SCXI-1125	Atenuación SCXI-1313A
0.01	-300	1	-----
0.02	-250	2	-----
0.002	-150	2	100
0.05	-100	5	100
0.1	-50	10	100
0.2	-25	20	100
0.5	-10	50	100
2.5	-2	250	100

De esta manera se puede observar que la capacidad máxima del sistema actual al poseer un bloque terminal se extiende hasta el valor de ± 150 V. Lo cual permite la medida de parámetros con rangos bastante altos.

Finalmente la tarjeta SCXI -1125 debe ser programada tanto en sus filtros como ganancia mediante el programa NI MAX, a partir del cual se escogen los parámetros deseados para estas 2 variables dependiendo de los rangos requeridos para la medición.

3.2.4 SCXI 1313A bloque terminal.

Este bloque terminal permite la atenuación de variables analógicas, consiste en un sistema de montaje frontal el cual puede extender la capacidad de entrada hasta 150 VDC para un sistema SCXI-1125.

El sistema permite mediante su atenuación programable en el sistema inicial para disminuir los valores de tensión que ingresan al mismo y hacerlos legibles por un sistema de rangos menores, además esta tarjeta proporciona un sensor de temperatura interna para el uso de éste en compensación de la unión fría con termopares [13].

La atenuación del bloque terminal SCXI 1313A es siempre constante y esta mantiene valores de 1:100 obtenida mediante divisores de tensión generados a partir de sistemas resistivos. Estos divisores de tensión se pueden omitir en casos en los que se realicen mediciones de baja tensión.

Para la habilitación y deshabilitación del atenuador cada canal cuenta con una configuración de herramientas a la cual se puede ingresar mediante NI MAX o ajustando los límites de entrada en la aplicación.

En la tabla 3.3 se muestra los rangos de entrada permitidos respecto a la atenuación del bloque terminal, así como el módulo.

Tabla 3. 3. Señales de entrada versus rangos de ganancia del bloque terminal.

Ganancia Total	Rango de Tensión	Ganancia del módulo	Ganancia del bloque terminal
0.01	300 Vrms o DC	1	0.01
0.02	250 Vpeak o DC	2	0.01
0.05	100 Vpeak o DC	5	0.01
0.1	50 Vpeak o DC	10	0.01
0.2	25 Vpeak o DC	20	0.01
0.5	10 Vpeak o DC	50	0.01
1	5 Vpeak o DC	1	1
2	2.5 Vpeak o DC	2	1
2.5	2 Vpeak o DC	250	0.01
5	1 Vpeak o DC	5	1
10	500 mVpeak o DC	10	1
20	250 mVpeak o DC	20	1

50	100 mVpeak o DC	50	1
100	50 mVpeak o DC	100	1
200	25 mVpeak o DC	200	1
250	20 mVpeak o DC	250	1
500	10 mVpeak o DC	500	1
1000	5 mVpeak o DC	1000	1
2000	2.5 mVpeak o DC	2000	1

Tal como se muestra en esta tabla existen diferentes rangos de capacidad para entrada analógica dependiendo de si se utilizan o no el sistema de atenuación para las tensiones de entrada.

3.2.5 NI MAX como programa de configuración para sistemas DAQ.

NI MAX es un asistente para la configuración de sistemas de adquisición de datos que posee grandes capacidades, este puede ser utilizado para configurar gran variedad de instrumentos de hardware de la marca National Instruments. Mediante el mismo es posible desarrollar diferentes opciones entre las que destacan la creación de canales, interfaces, instrumentos virtuales y diagnósticos de sistemas.

Además permite observar y configurar elementos de adquisición de datos conectados al sistema, permitiendo de esta manera la modificación de sus parámetros, como ubicación, ganancia, filtrado entre otros parámetros de los dispositivos conectados. En la figura 3.7 se muestra un ejemplo de la utilización de este software para el control de dispositivos DAQ.

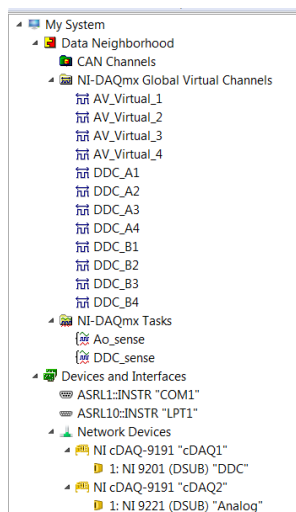


Figura 3. 7. Control de dispositivos DAQ.

En la figura anterior es posible observar cómo tanto los dispositivos de adquisición de datos así como los sistemas de comunicación que se encuentran actualmente en el sistema son referenciados por el programa NI MAX, a partir del cual es posible configurar los mismos a puertos específicos del computador, así como asignar las características deseadas.

De la misma manera es posible la formación de canales de medición tanto de manera global como local, los cuales pueden ser configurados según las necesidades de la medición accediendo a ganancias y filtros que modifican las características originales y preparan los parámetros deseados.

En la figura 3.8 se observa la configuración de un canal específico de medición.

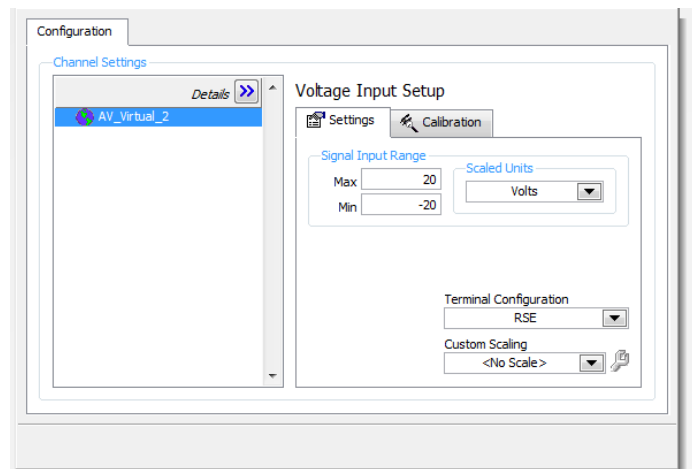


Figura 3. 8. Configuración de canal.

Cómo es posible divisar en la imagen, se pueden configurar los rangos de tensión de cada uno de los canales a medir. De la misma manera es posible utilizar una escala específica para casos en los cuales los valores de tensión de los sistemas sobrepasen los límites del dispositivo generando así una reducción de la medición.

3.2.6 Transmisión Inalámbrica.

La transmisión inalámbrica es sumamente utilizada en la actualidad debido a las ventajas que la misma presenta en cuanto a conectividad ya que permite la transmisión de información entre dos entes separados por diferentes distancias físicas. Este tipo de redes nos proporciona un sistema de comunicación muy flexible al eliminar por completo la utilización de cables.

El transmisor electrónico de RF es un dispositivo que transmite información en forma de ondas electromagnéticas a grandes distancias a través de la atmósfera, estas ondas son electromagnéticas

y son transmitidas en el nivel de la radiofrecuencia es decir sólo se transmiten ondas electromagnéticas cuyas frecuencias entran en este rango,(3Hz,300GHz) [22].

Las tareas principales de un transmisor electrónico son:

- Generar la señal portadora.
- Modular la señal portadora con el mensaje.
- Amplificar la señal modulada.
- Efectuar el filtrado de la misma.

Algunos de los sistemas transmisores y receptores digitales más comunes utilizan como frecuencias de funcionamiento 2.4 GHz y 433MHz. A continuación se presentan 2 casos muy claros de este tipo de sistemas de transmisión inalámbrica.

RF Serial Data Link UART, 2.4 Ghz.

Este sistema tiene su funcionamiento a una frecuencia de 2.4 GHz con el estándar Serial RS232, el mismo posee un Baud rate ajustable a 9600/4800/2400/19200 bps, además de la capacidad de enviar datos a una velocidad de 280Kbps o 500Kbps. Asimismo la de ganancia de funcionamiento es de 1dBm. Presenta interfaz con otros equipos vía RS232, es decir que mismo puede funcionar como instrumento para la comunicación entre dos computadores o microprocesador inalámbricamente tan solo mediante la conversión de a DB-9.[23]

Comprobación de error mediante CRC, rango de alcance de entre 30 -50 metros máximo. Posee sistema de intercambio automático entre modo de TX y RX. Tecnología de modulación FSK en half dúplex es decir solo transmite o recibe, no es posible ambas al mismo tiempo [23].

115200bps Transceiver Modules.

La frecuencia de funcionamiento de este sistema es de 434Mhz, además posee un rango de alcance de señal cercano a los 150 metros, igualmente es configurable si se desea en función TTL o RS232 mediante el conector DB-9, permitiendo con esto la fácil conexión con otros elementos. Posee detección automática del paquete de datos de hasta 96 bytes, Puede utilizarse en modo de “Operación transparente”, es decir que se comporte tan solo como una línea de información RS232.

Comprobación de error mediante CRC si se desea, El sistema presenta una serie de switches para la selección de características como el Baud Rate [24].

3.3 Protocolo de comunicación.

El protocolo, se define como las reglas para la transmisión de la información entre dos puntos. Un protocolo de red de comunicación de datos es un conjunto de reglas que gobiernan el intercambio ordenado de datos dentro de la red [14].

Existen diferentes tipos de protocolos, en este caso el protocolo utilizado para la transmisión de datos entre el software encargado de los diagnósticos y la placa diagnosticada cumple una serie de reglas pertinentes a la comunicación serial, en base a la cual se encierra el protocolo de comunicación Modbus, el cual se desarrollará a continuación.

En una interface serial los bits de datos son enviados de modo secuencial a través de un canal de comunicación o bus. Diversas tecnologías utilizan comunicación serial para la transferencia de datos, incluyendo las interfaces RS232 y RS485. Las normas que especifican los padrones a seguir en estas 2 interfaces, sin embargo, no especifican el formato ni la secuencia de caracteres para la transmisión y recepción de datos. En este sentido, además de la interfaz, es necesario identificar también el protocolo utilizado para la comunicación. Entre los diversos protocolos existentes, un protocolo muy utilizado en la industria es el protocolo Modbus [15].

3.3.1 MODBUS.

Modbus es un protocolo industrial que fue desarrollado en 1979 para hacer posible la comunicación entre dispositivos de automatización. El mismo consiste en de solicitud-respuesta implementado usando una relación maestro-esclavo. En una relación maestro-esclavo, la comunicación siempre se produce en pares, un dispositivo debe iniciar una solicitud y luego esperar una respuesta y el dispositivo de inicio (el maestro) es responsable de iniciar cada interacción. Por lo general, el maestro es una interfaz humano-máquina (HMI) o sistema SCADA y el esclavo es un sensor, controlador lógico programable (PLC) o controlador de automatización programable (PAC) [16].

En la especificación del protocolo están definidos dos modos de transmisión: ASCII y RTU. Los modos definen la forma como son transmitidos los bytes del mensaje. No es posible utilizar los dos modos de transmisión en la misma red.

El sistema de comunicación consta de tramas que están constituidas por un conjunto de caracteres que tienen una longitud en bits que depende del modo de transmisión que se utilice, la longitud varía hasta un máximo de 256 caracteres. Modbus maneja básicamente dos tipos de datos; bits individuales y palabras de 16 bits. Los bits individuales corresponden a entradas o salidas con estados ON/OFF y las palabras a registros de entrada o salida cuyos estados indican un valor analógico. En la tabla 3.4 se muestra estos tipos de datos [17].

Tabla 3. 4. Organización de señales Modbus.

Sector	Formato	Tipo de acceso	Controlador
Salidas discretas (coils)	Bits individuales	Lectura-escritura	Modificable por un programa de aplicaciones
Entradas discretas (Inputs)	Bits individuales	Solo lectura	Suministrados por un sistema de E/S
Registros de entrada (Input Registers)	Palabras de 16 bits	Solo lectura	Suministrado por un sistema de E/S
Registros de salida (Holding Registers)	Palabras de 16 bits	Lectura-escritura	Modificable por un programa de aplicación.

La red Modbus utiliza el sistema maestro-esclavo para el intercambio de mensajes. Permite hasta 247 esclavos, más solamente un maestro. Toda comunicación inicia con el maestro haciendo una solicitud a un esclavo, y este contesta al maestro el que fue solicitado. En ambos los telegramas (pregunta y respuesta), la estructura utilizada es la misma: Dirección, Código de la Función, Datos y Checksum. En la siguiente figura 3.9 se muestra las diferentes partes que componen el mensaje.

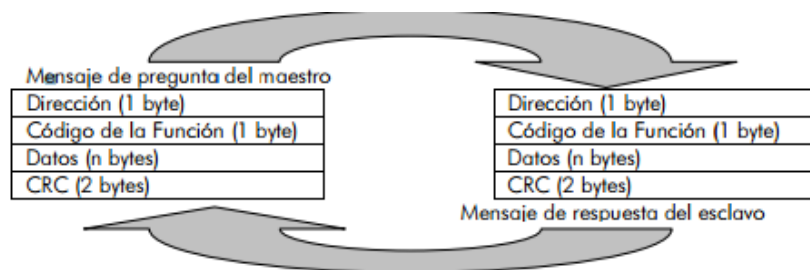


Figura 3. 9. Generación del mensaje [16].

A continuación se describen cada una de las partes que componen el mensaje.

Dirección: Modbus es un protocolo multipunto, lo que significa que el maestro puede comunicarse con múltiples esclavos utilizando la misma línea de comunicación lo cual es conocida como topología de bus. El maestro inicia la comunicación enviando un byte con la dirección del esclavo para el cual se destina el mensaje. Al enviar la respuesta, el esclavo también inicia el telegrama con él su propia dirección, posibilitando que el maestro conozca cuál esclavo está enviándole la respuesta. Se conoce como difusión (Broadcast) cuando el maestro se dirige a todos los esclavos de la red al mismo tiempo. En este caso el campo de dirección debe contener 00h. En este caso ninguno de los esclavos debe enviar una respuesta.

Campo de códigos de función: Este campo también contiene un único byte, donde el maestro especifica el tipo de servicio o función solicitada al esclavo (lectura, escrita, control.). De acuerdo con el protocolo, cada función es utilizada para acceder un tipo específico de dato. Las funciones de lectura y escritura se encargan de modificar los datos de los registros, mientras que las órdenes de control (RUN/STOP) se encargan de la carga y descarga de programas para la verificación de contadores. En la tabla 3.5 se muestran las funciones con su valor hexadecimal para cada una de las funciones.

Tabla 3. 5. Código hexadecimal de las funciones Modbus.

Código	Hex	Descripción
0	00	Control de estaciones esclavas.
1	01	Lectura de n bits de salida.
2	02	Lectura de n bits de entrada.
3	03	Lectura de n registros de entrada.
4	04	Lectura de n registros de salida.
5	05	Escritura de un bit.
6	06	Escritura de un registro.
7	07	Lectura rápida de 8 bits
8	08	Diagnóstico.

9	09	No utilizada
10	0A	No utilizada
11	0B	Solicitud de contador de eventos de comunicaciones.
12	0C	Solicitar diario de eventos de comunicaciones.
13	0D	No utilizada.
14	0E	No utilizada.
15	0F	Escritura de n bits (coils).
16	10	Escritura de n palabras (Holding Registers)

Campo de datos: Campo con tamaño variable. El formato y el contenido de este dependen de la función utilizada y de los valores transmitidos. El campo de datos se construye usando grupos de 2 dígitos hexadecimales en un rango de entre 00h a FFh lo cual se puede hacer a partir de caracteres ASCII o RTU de acuerdo al modo de transmisión en el que se esté trabajando. Aquí se contiene la información adicional que el esclavo debe usar para tomar la acción definida por el código de función como por ejemplo el registro específico en el que se desea leer o escribir.

Si no ocurre error, el campo de datos en el mensaje de respuesta contiene los datos solicitados. De lo contrario, este campo contiene un código de excepción que la aplicación del maestro puede usar para determinar la siguiente acción a tomar.

Chequeo de Errores (CRC):

El control de errores asegura que los dispositivos receptores no efectuarán acciones incorrectas debido a una modificación accidental del mensaje durante la transmisión, este depende del modo utilizado, en ASCII el chequeo consta de 2 caracteres los cuales son el resultado de un cálculo de redundancia. (LRC Redundancy).

Cuando se utiliza el modo RTU el campo de chequeo de error contiene un valor de 16 bits implementado como 2 bytes. El mismo es el resultado de chequeo de redundancia cíclica (CRC) aplicado al contenido del mensaje. El método utilizado es el CRC-16 (Cycling Redundancy Check). Este campo es formado por dos bytes, donde primero es transmitido el byte menos significativo (CRC-), y después el más significativo (CRC+)

Los dispositivos esclavos de la red no envían ninguna respuesta cuando detectan un error de CRC o LRC en la trama recibida [17].

Como se mencionó anteriormente, existen dos métodos de transmisión de estos datos los cuales poseen características similares, sin embargo es preciso evidenciar sus diferencias para realizar una comparación más acertada de los mismos.

3.3.2 Transmisión ASCII mediante Modbus.

La transmisión de datos vía Modbus mediante el método consiste en la toma de caracteres de esta tabla y la conversión de estos a su correspondiente a los datos en binario de estos caracteres.

Igualmente se debe tener presente la descripción de los campos correspondientes para la formación de tramas en Modbus, por lo que es necesario ajustar la colocación de los caracteres ASCII a estos espacios de trama. En la tabla 3.6 se muestra la generación de la trama.

Tabla 3. 6. Trama de caracteres en ASCII.

Inicio	Dirección	Función	Datos	LCR	FIN
1 Carácter :	2 Caracteres	2 Caracteres	N Caracteres	2 Caracteres	2 Caracteres

Mediante la imagen se tiene una mayor visualización de la trama mediante el modo ASCII, en este los mensajes comienzan con el carácter “:” cuyo hexadecimal es 3A y finaliza con el retorno cíclico o avance de línea CR/LF código en hexadecimal (0D o 0A) respectivamente, los cuales delimitan los extremos de la trama. Todos los dispositivos conectados a la red monitorizan el bus continuamente con la finalidad de detectar caracteres ASCII válidos. Una vez que estos dispositivos detectan el valor de “:” se inicia la comunicación con el maestro a la espera de la siguiente trama correspondiente a la dirección del esclavo seleccionado.

La longitud de un paquete de transmisión ASCII es de 10 bits, un bit de inicio, 7 bits de dato, un bit de paridad y un bit de parada. El hecho que se envíen sólo siete bits de información en un carácter en el modo ASCII implica la necesidad que es necesario enviar 2 caracteres para una sola instrucción. Por ejemplo para transmitir un mensaje como 06H se deben realizar 2 envíos, en el primero el 0 y en el segundo el 6 correspondiente al dato hexadecimal lo cual hace que la transmisión de información se realice más lentamente comparado por con el modo RTU [17].

3.3.3 Transmisión RTU mediante Modbus.

La transmisión mediante el modo RTU tiene como estructura principal el uso de datos de manera original, es decir se traslada el dato hexadecimal. La trama de comunicación se basa en un periodo de tiempo de inicio, una dirección a la que se hace referencia en este caso el valor hexadecimal del esclavo deseado, la función que se desea realizar al esclavo, los datos los cuales se envían en múltiplos de 8 bits, la comprobación de redundancia cíclica la cual consta de 16 bits y el periodo de silencio al final. Este periodo de silencio de la trama se compone a 3.5 veces el tiempo necesario para enviar un carácter. En la tabla 3.7 se muestra lo anteriormente mencionado.

Tabla 3. 7. Trama de caracteres en RTU.

Inicio	Dirección	Función	Datos	LCR	FIN
T1-T2-T3-T4	8 bits	8 bits	n* 8bits	16 bits	T1-T2-T3-T4

Por la presencia de estos periodos de silencio es necesario que los tiempos entre un carácter y otro no sobrepase los $1.5T_c$ donde T_c es tiempo de carácter, Si esto sucede se genera un mensaje de error, a partir del cual es esclavo decide no enviar ningún mensaje de respuesta. De esta manera el intercambio de datos de manera correcta se presenta en diagrama de tiempos de la figura 3.10.

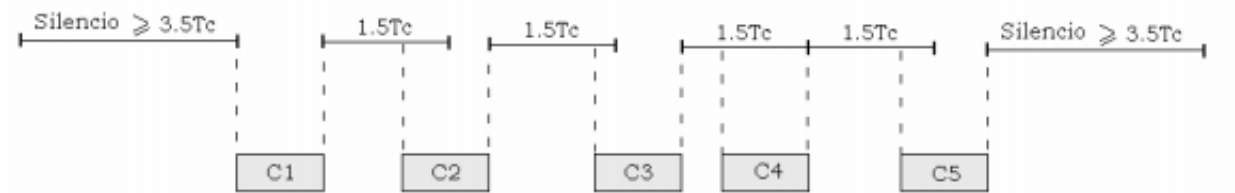


Figura 3. 10. Diagrama de tiempos en modo RTU [18].

Si existe algún tipo de error debido a tiempos de silencio en las tramas este será detectado durante el cálculo de redundancia cíclica (CRC) a partir del cual se decide la no respuesta de parte del esclavo y la comunicación del error al maestro.

3.3.4 Descripción de funciones del protocolo Modbus.

Las funciones del protocolo Modbus permiten la transmisión de órdenes o datos específicos según las necesidades de diagnóstico que se tengan. Existen 2 tipos básicos de órdenes.

- Órdenes de lectura/escritura de datos en los registros o en memoria del esclavo.

- Órdenes de control (RUN/STOP) encargadas de la carga y descarga de programas, verificación de contadores entre otros.

Cabe destacar además que cada una de las direcciones que ingresa en Modbus se direcciona a una posición con un valor inferior a la real, es decir que si se hace referencia a la bobina 127 decimal se debe colocar el valor hexadecimal de Modbus como 007 Eh el cual corresponde a la dirección 126 decimal.

Leer Estado de Bobinas (01H).

Esta función permite al maestro solicitar al esclavo el estado de las **salidas discretas** de un dispositivo remoto. Estas salidas poseen un estado binario ON/OFF. Las bobinas son accedidas según el orden planteado en la trama en los capítulos anteriores, en la tabla 3.8 se muestra un ejemplo de esta trama.

Tabla 3. 8. Trama de estado de bobinas (coils).

1	2	3	4	5	6	7
11	01	00	13	00	25	CRC

La figura anterior hace referencia a la petición al esclavo 17 de los estados de las bobinas de la 20 a la 56. Primeramente el maestro indica la dirección del esclavo en 1, la función se coloca en 2, la dirección inicial de las bobinas se coloca en 3 y 4 respectivos a la parte alta y baja del dato un valor de 19 en decimal, además en las posiciones 5 y 6 se coloca el número de salidas que en este caso es 37 bobinas. Finalmente se encuentra el CRC.

De esta manera la respuesta del esclavo contiene el estado de cada bobina empaquetado en bloques de 8 bits. Es decir el estado de 8 bobinas consecutivas. Ordenadas de manera ascendente es decir el LSB contiene el valor de la bobina inicialmente direccionada o bobina número 20 en este ejemplo. Si quedan datos restantes, al ser incluidos estos en un paquete de 8 bits los MSB son rellenados con 0. En la tabla 3.9 se presenta la trama de respuesta del esclavo.

Tabla 3. 9. Respuesta de trama de estado de bobinas (coils).

1	2	3	4	5	6	7	8	9
00	01	05	CD	6B	B2	0E	1B	CRC

Como se puede observar en la imagen que inicialmente el esclavo responde asegurando su dirección y la función que se realizó 1 y 2, luego este especifica la cantidad de bytes que va a enviar para completar todas las bobinas cuestionadas 3. De esta manera las posiciones 4,5,6,7 y 8 corresponden a la respuesta del estado de las bobinas, donde el LSB es en este caso el valor de CD donde se presenta el resultado de las primeras 8 bobinas. En este caso el orden es el siguiente: CD corresponde a 11001101 de esta manera la bobina 27 es el MSB de esta trama mientras que la bobina 20 es la LSB. De la misma manera sucede con los demás datos ubicados en las posiciones siguientes [18].

Estado de las entradas (02H).

Este permite al maestro solicitar el comportamiento de estado ON/OFF de un dispositivo remoto, mediante la **entradas discretas** del mismo, las entradas discretas se direccionan al igual que el caso anterior a partir de 0, la operación realizada también es conocida como Read input Status. En la tabla 3.10 se muestra con mayor claridad la secuencia de la trama enviada para solicitar al esclavo que envíe el estado de las entradas discretas del mismo [18].

Tabla 3. 10. Trama de Read input Status.

1	2	3	4	5	6	7
11	02	00	C4	00	16	-----

En este caso se observa que el esclavo es el campo 1, luego la función en el segundo campo, seguido a esto se tiene la dirección de inicio el cual se encuentra entre las posiciones 3 y 4, así como el número de salidas que se desean recibir de información de datos ubicado entre las posiciones 5 y 6. Finalmente la verificación de error de mensaje en la posición 7.

De la misma manera que se observó en el caso anterior, el esclavo responde la secuencia con la información respectiva, de manera que se ordenan los datos igualmente con su valor LSB en el primero de la secuencia en semejanza al de las bobinas.

Lectura de registros de salida (03H).

Esta función revela el estado binario de un registro de salida de cada uno de los esclavos, estos registros en su gran mayoría se encuentran compuestos de 16 bits de datos; también son conocidos como Holding Registers. El mensaje enviado al esclavo consta de 2 bytes de longitud para especificar el registro inicial a consultar, luego se indica la cantidad de registros que se deben enviar de respuesta de parte del esclavo. Al igual que en todos los casos anteriores la última secuencia es producto de la comprobación de error por redundancia cíclica.

El dato de respuesta procede de una manera semejante a como se mencionó en otros casos, este igualmente confirma la posición de esclavo, el código de la función, además especifica la cantidad de bytes que se enviarán como respuesta a lo solicitado por el maestro, estos bytes son correspondientes a 2 bytes por cada registro solicitado [18].

Lectura de registros de entrada (06H).

Esta función tiene como objetivo el conocer el estado de los registros de entrada de un esclavo, al igual que en el caso anterior estos registros poseen una longitud promedio de 16 bits y se comienzan a direccionar a partir del registro cero. El mensaje de solicitud de parte del maestro está compuesto por dirección del esclavo, código de función, registro inicial, número de registros a solicitar.

Por otra parte la respuesta está compuesta dirección de esclavo, función, cantidad de paquetes enviados y por dos paquetes correspondientes a cada registro consultado estos paquetes son ordenados, el primero con los bits más significativos (MSB) y el segundo con los LSB [18].

Función de diagnóstico (08H).

La función de diagnóstico es utilizada para la verificación de la correcta comunicación entre maestro y esclavo, inclusive para la comprobación de condiciones internas en el esclavo.

Cabe destacar que una función de diagnóstico utiliza en la consulta un código de subfunción de 2 bytes para definir la prueba realizada, de la misma manera el esclavo devuelve una respuesta normal en código de función y en subfunción. Cuando el maestro emite una función de diagnóstico a un esclavo, esta no afecta el programa actual que el mismo está procesando ya que la lógica de usuario, entradas y salidas discretas no pueden acceder desde este tipo de funciones de diagnóstico.

En ejemplo de este tipo de funciones se presenta en la tabla 3.11. En la cual se muestra una pregunta de diagnóstico [18].

Tabla 3. 11. Trama de diagnóstico.

1	2	3	4	5	6	7
11	08	00	00	A5	37	-----

Como es posible divisar en la figura anterior el maestro utiliza la función (08H) para referenciar que el procedimiento es diagnóstico, seguido a esto se coloca el código de subfunción, en este caso 00H, para lo cual se utilizan 2 bytes. La información devuelta de parte del esclavo es colocada en las posiciones 5 y 6, la cual es la respuesta correspondiente al diagnóstico deseado.

Como se mencionó anteriormente esta función posee espacio adicional para la colocación de subsunciones, las cuales permiten al maestro realizar diferentes a los esclavos lo cual se muestra en la tabla 3.12.

Tabla 3. 12. Sub-funciones de diagnóstico.

Sub-función	Datos (Consulta)	Datos (Respuesta)
00 0B	00 00	Total de mensajes detectados
00 0C	00 00	Total de mensajes con error CRC
00 0D	00 00	Total de mensajes de excepción
00 0E	00 00	Total de mensajes procesados
00 0F	00 00	Total de mensajes no respondidos
0010	00 00	Total NAK

La tabla anterior presenta los datos necesarios para que el maestro consulte al esclavo diversas situaciones contempladas en la tabla como mensajes detectados, mensajes de error , mensajes de excepción , entre otras respuestas que se presentan desde la última vez que se realizó un reinicio, última operación de limpieza de contadores o último arranque.

Códigos de excepción.

Los códigos de excepción suceden en su gran mayoría durante el ciclo consulta respuesta, en el cual pueden darse diferentes eventos que obligan al esclavo a enviar un mensaje de excepción.

Este tipo de mensaje se da cuando ha sucedido alguno de los siguientes eventos.

- El esclavo no soporta la función solicitada por el maestro (01h).
- Se solicita un bit o palabra inexistente (02h).
- Cuando el esclavo necesita tiempo adicional para ejecutar la función (05h).
- El número de datos sobrepasa el máximo soportado por el esclavo (03h).
- El esclavo está ocupado (06h).
- Error de ejecución de una función enviada en un programa (04h).
- Reconocimiento negativo (07h).
- Error de memoria extendida (08).

3.4 Programación del hardware.

El desarrollo del sistema diagnóstico incluye la presencia de programación con la finalidad del ordenamiento en la adquisición de datos así como el procesamiento de estos datos para la verificación de la funcionalidad de las tarjetas testeadas, es por esto que se procede a presentar una introducción a la programación existente en la actualidad y su función general.

Esta programación se encarga de llevar el hilo de la comunicación entre maestro y esclavo, además de solicitar los sistemas de adquisición de datos los respectivos valores censados. De la misma manera lo mencionan algunos autores.

Al estudiar profundamente la configuración de los sistemas de adquisición de datos modernos DAQ (Data Acquisition System), basados en equipos PC (Personal Computer), se aprecia que una de las partes que componen dichos sistemas, es el software quien controla y administra los recursos del computador, presenta los datos, y participa en el análisis.

Viéndolo de este modo, el software es un tópico muy importante que requiere de especial cuidado. Para los sistemas DAQ se necesita de un software de instrumentación, que sea flexible para futuros cambios, y preferiblemente que sea de fácil manejo, siendo lo más poderoso e ilustrativo posible [19].

3.4.1 Programación en Labview.

LabVIEW es un lenguaje de programación gráfico, adoptado en la industria y en la academia, como estándar para el desarrollo de sistemas de instrumentación virtual, dado su poderoso conjunto de bibliotecas de funciones que permite el desarrollo de algoritmos complejos de procesamiento de señales, además de permitir la comunicación con cualquier protocolo de comunicaciones e interfaces de hardware y software existentes

Adicionalmente, LabVIEW es la plataforma estandarizada para la adquisición de datos y el control de instrumentos que son pieza fundamental en los sistemas de instrumentación virtual. Estos sistemas de instrumentación virtual programados mediante el uso del software LabVIEW permiten el monitoreo y control de cualquier variable física que desee ser procesada por computadora, y realizan la implementación de algoritmos matemáticos con el objeto de visualizar resultados gráficamente, o comunicarlos con sistemas remotos mediante el uso de interfaces inalámbricas; todo esto engloba las características de cualquier sistema de instrumentación virtual. [20].

En el sistema actual de diagnóstico de tarjetas el lenguaje labview es utilizado para llevar a cabo la diferente toma de datos así como la preparación de la tarjeta para los mismos.

Labview cuenta con una serie de plataformas para la adquisición de datos y la comunicación con sistemas externas, entre las cuales destacan las herramientas DAQmx, Control Design and simulation, modbus protocol, VISA communication entre otras utilizadas. En la figura 3.11 se muestra la tabla de herramientas para la adquisición de datos mediante labview.

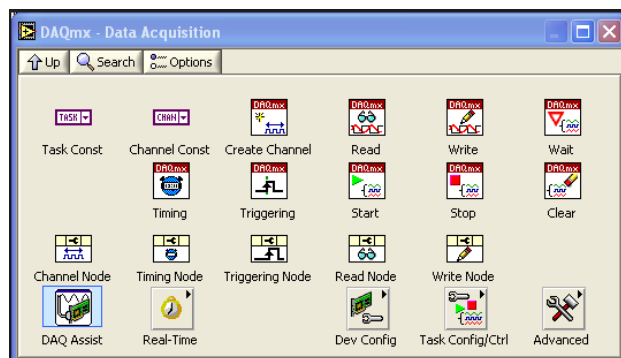


Figura 3. 11. Barra de herramientas DAQmx. [21].

En esta figura es posible observar que con el asistente de adquisición de datos que proporciona National Instruments es posible, crear el canal, leer un dato específico, escribir datos, esperar e

iniciar el diagnóstico entre otros, cabe destacar que para utilizar estas herramientas es necesario primeramente configurar el sistema desde NI MAX como se presentó en capítulos anteriores. Por otra parte en la figura 3.12 se presenta la barra de herramientas de modbus para la comunicación del maestro con un esclavo o viceversa.

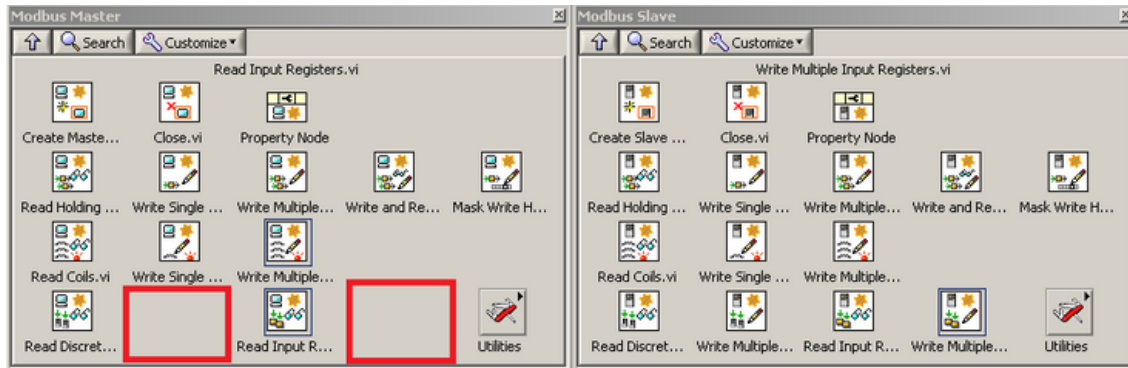


Figura 3. 12. Barra de herramientas Modbus. [21].

En la imagen anterior es posible evidenciar que existen diferentes opciones tanto para la creación de un sistema de maestro como para la creación de un sistema esclavo, en este caso la diferencia principal en la imagen es que en el sistema de maestro este no puede tener acceso a escribir múltiples registros ya que esta es una acción única del sistema esclavo.

Labview además proporciona la posibilidad de simular sistemas en base a otros programas de diseño de circuitos como por ejemplo Proteus o Multisim Circuit Design y utilizar sistemas programable como lo son Arduino, Raspberry pi, Beagle board entre otros.

Capítulo 4. Diseño de la solución.

4.1 Propuestas de solución.

Para el desarrollo del proyecto se propusieron 2 soluciones principales las cuales se discutirán a continuación evaluando las características de cada una de ellas con la finalidad de mostrar más detalladamente el por qué se escogió una de las mismas respecto a las demás.

4.1.1 Acceso remoto y sistema con simultaneidad.

Una de las ideas principales es la utilización de un sistema de acceso remoto para la prueba de tarjetas, el mismo consiste en la utilización del sistema actual, pero este implementando el diagnóstico varias de las tarjetas al mismo tiempo utilizando las capacidades que posee el sistema maestro para la comunicación con las tarjetas esclavas.

Este sistema está basado en el fundamento teórico de la utilización de diferentes protocolos de comunicación, ya que el sistema maestro cuenta con dos protocolos iguales de tipo SPI para comunicarse con cada una de las tarjetas. Esto se puede observar en la figura 3.2.

En esta figura se encuentran algunas líneas las cuales indican que es posible utilizar la placa maestra para la activación de esclavos mediante Chip Select. Debido a que existen tres líneas de este tipo se observa la posibilidad de controlar el mismo número de tarjetas mediante cada uno de los protocolos SPI.

Del mismo modo en la figura 4.1 se presenta un esquema de la idea de separación planteada inicialmente.

Tal y como se muestra en la figura anterior el sistema consiste en creación de nuevos sistemas de pruebas inalámbricos, los cuales se conectan mediante el protocolo SPI de la tarjeta maestra, a partir de la misma y mediante las líneas de esclavo se hace referencia a las tarjetas por diagnosticar.

La tarjeta main board es la responsable de comunicarse con la computadora inalámbricamente, al igual que lo hacen los sistemas de adquisición de datos DAQ y el sistema NI 485. Finalmente la placa PCI debe comunicarse con la tarjeta devicenet.

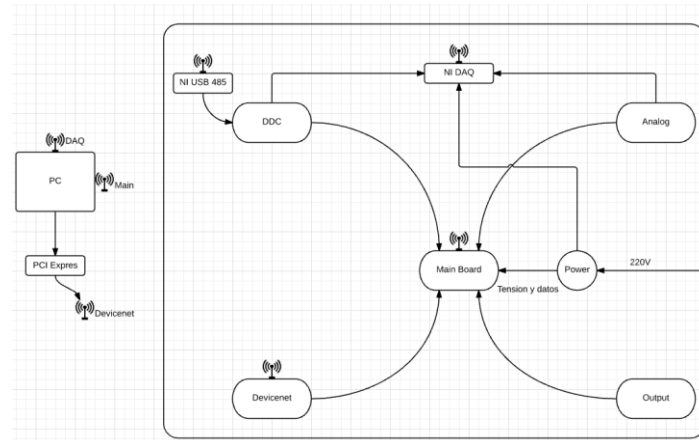


Figura 4. 1. Sistema de adquisición inalámbrico basado en SPI.

4.1.2 Separación de sistemas y comunicación inalámbrica.

Este caso se encuentra más dirigido a las necesidades de la empresa ya que se focaliza en las tarjetas con mayor demanda de parte de los clientes, las cuales corresponden a la DDC y Analog Board quienes son solicitadas en grandes cantidades y con mucha frecuencia.

El sistema se planteó para la prueba simultánea de estas dos tarjetas de manera separada al resto del sistema, es decir la creación de nuevos métodos de diagnóstico en los cuales se prueban cuatro tarjetas idénticas, y se presentan los resultados al operador de las mismas. Esto sucede para ambos casos mencionados, en la figura 4.2 se muestra un esquema general.

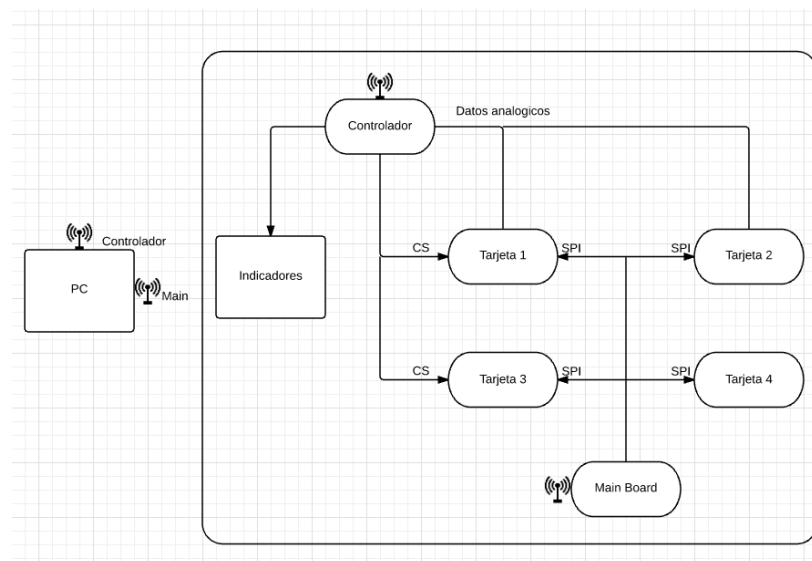


Figura 4. 2. Sistema Externo.

En la figura anterior es posible evidenciar la presencia de un microcontrolador, el cual se encargaría de presentar los mensajes para el operador, además es posible observar que este mismo microcontrolador realiza la selección de la tarjeta a la que desea tener acceso, esto mediante la multiplexión de las mismas.

El protocolo SPI utilizado para todas las tarjetas es el mismo, es decir, todas las tarjetas se encuentran conectadas a la misma línea y son seleccionadas a partir del microcontrolador.

De esta manera la comunicación inalámbrica solamente se presenta entre computador y microcontrolador, así como entre la placa maestra y el mismo.

Los datos analógicos son recolectados por el microcontrolador, lo cual se logra a partir de una reducción del valor analógico de manera que este pueda ser utilizado en la lectura mediante el microcontrolador, es decir se acondicionan los valores analógicos para que estos estén disponibles para el convertidor analógico digital del microcontrolador.

4.1.3 Comparación de sistemas.

Para la continuación del proyecto es preciso la selección de uno de estas dos propuestas, de manera que se desarrolle el diseño del mismo, por ende se pondrán en prueba los beneficios que poseen cada una de estas propuestas.

En la primera de estas se muestra una solución que implica la puesta en marcha de un nuevo banco de pruebas en el cual se deben colocar una serie de instrumentos de medición y recopilación de datos de alto costo, además, se pretende la obtención de computadoras adicionales para los mismos, es decir el sistema pretende la reproducción del método actual mediante el sistema Wireless, sin embargo a pesar de esto, el mismo beneficiaría en gran manera la producción brindando un fixture adicional para el testeado de tarjetas y la posible reproducción del mismo.

Por otra parte la segunda propuesta visualiza un método más acercado a la realidad que vive la empresa, se focaliza en las necesidades actuales y del futuro mediante la reproducción del sistema de pruebas de una tarjeta a 4 tarjetas iguales. Esto mediante la multiplexión, además por diagnosticarse la misma tarjeta se tiene una mayor especialización tanto en programación del mismo como en diseño de sus requerimientos.

Como se mencionó en el entorno del problema entre las tarjetas que poseen mayor demanda en producción destacan la DDC y la Analog board, cuyas órdenes implican aproximadamente 1000 unidades mensualmente, por lo que al utilizar la segunda propuesta se reduciría la cantidad de diagnósticos en un 75% y con esto la duración de estas tarjetas en el aérea de test. Mientras tanto con el primero de los métodos este número no sería tan beneficioso.

Otro aspecto relevante a tomar en cuenta es el costo generado por el nuevo sistema. En el primero de los casos el costo se muestra en la tabla 4.1.

Tabla 4. 1. Costos de la primera propuesta.

Elemento	Costo
PC Intel i5 3GHz, 8Gb	\$ 309.95
NI SCXI 1000	\$ 419.99
NI SCXI 1125	\$ 824.49
NI SCXI 1600	\$ 1,770.00
NI SCXI 1313	\$ 570.00
USB to RS232	\$ 2.00
Rs232 to TTL	\$ 1.50
NI USB 485	\$ 460.00
Fixture Box	\$ 323.68
Cables, Conectores, otros	\$ 200.00
Costo Total	\$ 4,881.61

A partir de la tabla se evidencia que el precio se extiende hasta casi los 5000 dólares, cabe destacar que este dato es manteniendo los sistemas de recolección de datos actuales los cuales además de su alto costo se encuentran descontinuados por el proveedor, sin embargo estos son necesarios para la toma de datos en este caso ya que poseen altos rangos de atenuación en sus mediciones necesarios para la medición de la Power board elemento incluido en el sistema.

Además la presencia de una computadora encargada de cada uno de estos módulos es otro aspecto que incrementa en gran parte el costo por unidad.

Para realizar la comparación respectiva se presentarán los costos de la segunda propuesta. En la tabla 4.2 se muestran estos datos.

Tabla 4. 2. Costos de la segunda propuesta.

Elemento	Costo
PC intel i5 3GHz, 8Gb	\$ 309.95
USB to RS232	\$ 2.00
Rs232 to TTL	\$ 1.50
NI USB485	\$ 460.00
Multiplexer	\$ 200.00
Fixture Box	\$ 323.68
Cables, conectores, otros	\$ 200.00
Costo Total	\$ 1,497.13

Como es posible observar el costo total de esta idea es mucho menor al presentado en el primer caso, ya que no implica la conservación del sistema recolección de datos analógicos NI DAQ a menos que este sea deseado por la empresa para una mayor resolución en la toma de datos.

Además la presencia del microcontrolador y sistemas de comunicación no implican un gasto mayor que en el mercado estos rondan entre los 20 a 100 dólares.

De esta manera una vez expuestas los beneficios que se obtendrán con cada uno de los casos de estudio así como los potenciales costos de estos sistemas se concluye que el segundo caso posee mejores características, a las necesidades actuales de la empresa.

4.2 Desarrollo de la propuesta seleccionada.

En base a lo discutido en la sección anterior, se seleccionó como propuesta a realizar la idea de los sistemas de diagnóstico de placas específicas a las necesidades de la empresa, tanto por su bajo costo como por los beneficios que estas generarían.

Esta propuesta abarca 2 tarjetas específicas de las 4 disponibles por diagnosticar, enfocando los procedimientos a la obtención de nuevos sistemas de pruebas con capacidad de test de 4 tarjetas desde el mismo mediante la comunicación inalámbrica con una computadora, dicha computadora se utiliza para ambas tarjetas de manera que cualquiera de estas puede ingresar y realizar una petición de test, incluso ambas pueden funcionar como hilos de programación paralelos.

Así se inició con el diseño del sistema tanto en sus conexiones como en programación y simulaciones para ambos bancos de pruebas.

4.2.1 Diseño de conexión de equipos e interfaz de operador.

El diseño físico de conexión del sistema es de suma importancia ya que previene la posibilidad de errores al tratar de adaptar funcionalmente los diferentes equipos, indicando de manera exacta los elementos necesarios desde las placas esenciales, hasta los elementos convertidores de señales y demás.

Las conexiones iniciales a tomar en cuenta son las provenientes de la computadora encargada de que se siga un proceso de diagnóstico, estas fueron expuestas a grandes rasgos en la figura 3.22. Primeramente se evidencia la presencia de una antena o sistema de comunicación inalámbrico en cual en este caso corresponde a un transceiver 2.4Ghz. El cual es capaz de establecer conexión con cualquier otro elemento del mismo tipo, adicionalmente existiría otro muy semejante en la tarjeta maestra. En este caso es necesario adicionalmente un convertidor de Rs232 a TTL para mantener los rangos de tensión entre los límites esperados por la tarjeta maestra.

La configuración es diferente para cada una de las placas debido a que involucran diferente cantidad de canales así como elementos adicionales en el caso de la DDC. En la figura 4.3 se muestra un diagrama de conexiones para la tarjeta Analog Board.

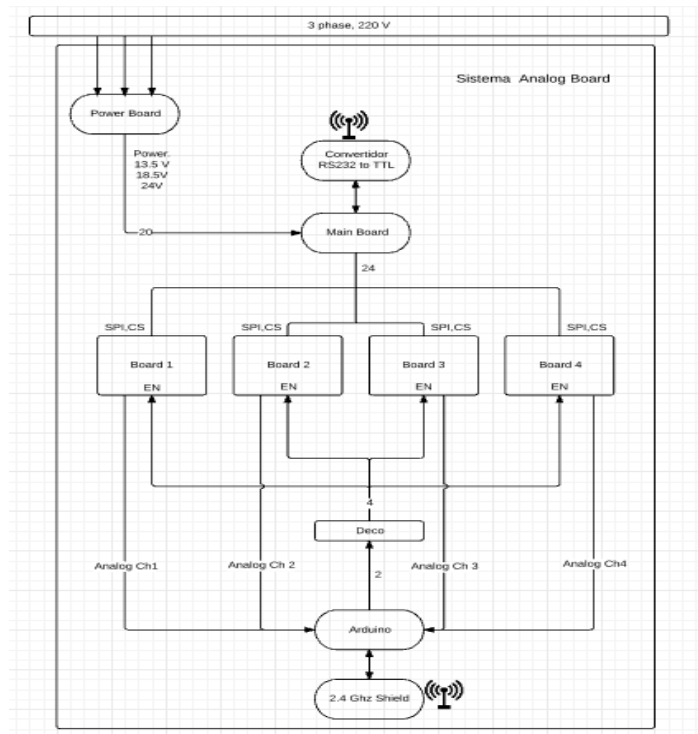


Figura 4. 3. Diagrama de conexiones Analog Board.

Se identifica que el sistema debe poseer una tarjeta adicional para la configuración de las tensiones adecuadas 24 V, 18V, 13.5V AC, tensiones necesarias para la alimentación de la tarjeta Main, así como las tarjetas diagnosticadas. Power Board consta de una serie de transformadores reductores, los cuales poseen diferentes relaciones entre sus devanados permitiendo diferentes tensiones de salida en corriente alterna. Por otra parte la Main board se comunica con las diferentes tarjetas mediante un conector de 24 pines entre los cuales brinda líneas de tensión VDC, líneas de selección !CS, así como las líneas MOSI, MISO, SCLK,y Reset.

Además es posible observar como el decodificador se encarga habilitar el esclavo mediante la selección de ENA de las tarjetas al que se hace referencia según lo solicitado por el programa, este mediante el microprocesador. Asimismo el microprocesador convierte los datos analógicos provenientes de las tarjetas a digital mediante un ADC interno incluido para todos los canales.

El convertidor Rs232 to TTL se encarga de acondicionar las señales de transmisión para que estas no presenten problemas al ser introducidas a la tarjeta maestra ajustando los rangos de tensión entre 0-3.3V a 0-5V, evitando errores tanto en recepción como en transmisión de datos.

Algo semejante sucede con la tarjeta DDC figura 4.4, la cual cuenta de la misma manera con el sistema de transformadores para alimentación Power Board, así como la Main board como master.

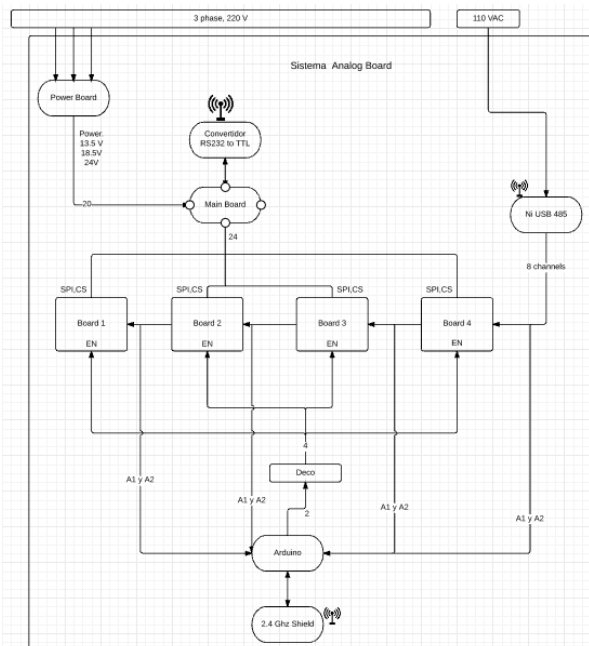


Figura 4. 4. Diagrama de conexiones DDC.

Tal y como se muestra en la figura este sistema posee un sistema adicional de comunicaciones serial NI-485 se encarga de una parte de la secuencia de pruebas. Más específicamente de realizar las secuencias de comunicación “COM A y COM B” relativos a la función de “Status” que se observará más adelante en la cual es necesario modificar el puerto así como el maestro quien en este caso será la misma tarjeta diagnosticada, es decir, esta función se realiza directamente sobre la tarjeta DDC. Finalmente es posible observar que en este caso se poseen 2 canales analógicos por cada una de las tarjetas.

De igualmente el computador se comunicara vía 2.4 GHz con el microcontrolador de manera que este reciba las órdenes y envíe las solicitudes especificadas para el diagnóstico, los elementos necesarios para la comunicación se presentan en la tabla 4.3.

Por otra parte, cada uno de los sistemas diseñados contará con una interfaz de operador en la parte superior, en la cual el mismo podrá realizar las solicitudes necesarias y visualizar los resultados de los mismos. Esta interfaz cuenta con una serie de elementos necesarios para su funcionamiento, que se muestran en el diagrama de la figura 4.5.

En esta figura se puede evidenciar la presencia de elementos como leds, display lcd y 7 segmentos, que permiten al operador clara guía de la secuencia a seguir durante el proceso de diagnóstico. Además el operador genera la solicitud a partir de 4 botones, a partir de los cuales se indica la cantidad de tarjetas que el mismo desea testear. Parte importante adicional es la presencia de un lector de código de barras ya que cada tarjeta posee un código de barras asignado a la misma por lo que es necesario obtener el mismo para guardar los resultados de las pruebas para futuras eventualidades.

Cabe destacar que cada uno de estos elementos antes mencionados es dirigido por el microcontrolador, de manera que este recibe la petición, códigos de barras, número de tarjetas e indica en los leds y display los mensajes correspondientes al estado del diagnóstico así como a la verificación de fallas.

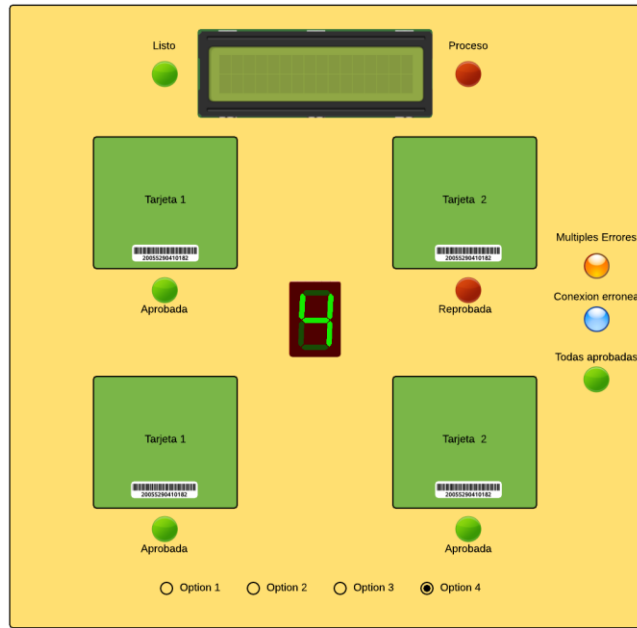


Figura 4. 5. Interfaz de operador.

Una vez conocido la interfaz de operador en la figura 4.6 se muestra el diagrama de conexión interna de esta interfaz.

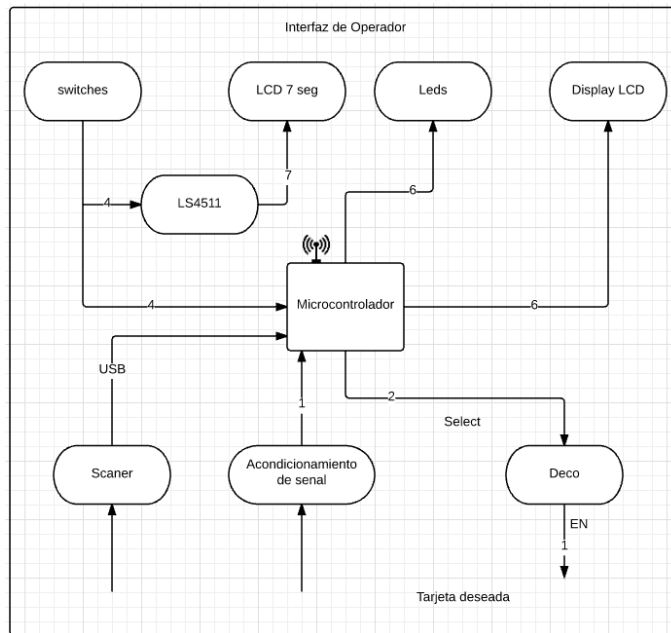


Figura 4. 6. Diagrama interno del sistema.

En la imagen anterior se puede evidenciar que el microcontrolador además de controlar los sistemas de interfaz con el operador se encarga de la recolección de los datos analógicos una vez que estos han sido acondicionados a las necesidades del ADC interno del mismo. Por otra parte es quien se encarga de la selección de esclavo mediante un decodificador permitiendo controlar cada una de las tarjetas mediante la habilitación de las mismas mediante CS o ENA.

Es importante destacar la presencia de un codificador BCD a 7 segmentos, cuya función principal es mostrar al operador la cantidad de tarjetas seleccionadas. El lector de códigos de barras también forma parte del diagrama ya que implica la existencia de un módulo adicional para la conexión USB microcontrolador al scanner.

De esta manera la tabla 4.3 se realiza una recopilación de los elementos involucrados, y las cantidades en que los mismos son necesarios.

Tabla 4. 3. Elementos utilizados en la interconexión del sistema.

Cantidad	Parte
1	Display LCD
1	Display 7 Segmentos
4	2.4Ghz Transceiver
4	Compuertas logicas AND, NAND, OR, NOT.
1	Scanner de códigos de barra
1	Decodificador 2X4, 74LS139
1	Arduino Mega
1	Computadora
20	Resistencias varias
15	Switchs interruptores y LED's
5	Cables Ribbon
1	Convertidor USB to 485
1	Convertidor RS232 to TTL

En esta lista de elementos se debe incluir la caja de pruebas la cual contiene todos los sistemas.

4.2.2 Escogencia de microprocesador.

La presencia de un microprocesador que posea la características necesarias es vital para el buen funcionamiento del sistema, por lo fue preciso el análisis de los requerimientos que el mismo debe de cumplir en cada uno de los aparatos de pruebas.

El mismo debe poseer como características mínimas 6 pines digitales designados como salidas para la conexión del display LCD, 6 salidas digitales también para mostrar el estado actual del sistema mediante los led de tarjetas, error, proceso, múltiples errores y secuencia completada.

Adicionalmente para el control del multiplexor de esclavo el microprocesador debe contar con 2 salidas adicionales utilizables para la selección.

En cuanto a los requerimientos de comunicación, este debe poseer al menos un protocolo SPI para la utilización del sistema transceiver 2.4Ghz ya que este es controlable mediante este protocolo.

Además debe poseer posibilidad de conexión vía USB para la introducción del Scanner lector de códigos de barras. Esta conexión realizada ya sea directamente o mediante un Shield que pueda adicionarse al mismo.

Analógicamente el sistema debe poseer un una serie de canales de entrada con alta resolución para evitar que pequeños rangos de tensión puedan alterar los resultados de las tarjetas que son analizadas. En este caso el número de entradas analógicas depende de la tarjeta, en el caso de la Analog Board el sistema necesita tan solo 4 entradas analógicas una correspondiente a cada tarjeta, sin embargo para el caso de DDC el sistema necesita 8 entradas analógicas.

En la tabla 4.4 se realiza un resumen de los requerimientos generales del microcontrolador.

Tabla 4. 4. Requerimientos mínimos del microprocesador.

Cantidad	Descripción
1	Puerto USB serial
1	Protocolo SPI
18	E/S Digitales
8	E/S Analógicas

Otro aspecto importante a tomar en cuenta es el precio del mismo debido a que el presupuesto es limitado para ambos sistemas.

Arduino es una plataforma que posee gran capacidad de conexión y manejo de sistemas de comunicación, además presenta gran cantidad de módulos que pueden ser adicionados para la centralización del sistema y utilización de protocolos comunes como serial USB.

De esta manera se seleccionó el Arduino mega como microprocesador ya que este cumple con los requerimientos mínimos y lo sobrepasa permitiendo la adición de otros sistemas o extensión del diseño actual. Arduino mega posee 54 pines digitales, adicionalmente 16 canales analógicos, y protocolos de comunicación SPI, I2C y 4 UART o serial.

En cuanto a resolución de ADC el Arduino mega provee 10 bits de resolución en un rango de 0 a 5V por lo que cada 5mV se verá reflejado un cambio en el dato al ser trasladado este al valor digital.

Por otra parte es posible obtener un Shield para el conector USB, USB host shield for arduino, el cual posee un valor en el mercado de 12 dólares, esto aunado al precio del Arduino no supera los 20 dólares por lo que la solución es sumamente económica.

4.2.3 Diseño de pruebas programadas.

Tal como se mencionó en los capítulos anteriores, se procedió al diseño de un banco de pruebas nuevo para estas tarjetas, igualmente se utilizó como lenguaje de programación Labview, esto debido a la gran cantidad de herramientas de interconexión que el mismo brinda, así como sus librerías específicas para los diversos protocolos.

La elaboración de diagramas de flujo y casos para la implementación de las rutinas de programación fue fundamental para realizar el trabajo de una forma ordenada y sistemática, estos diagramas dependen siguen una secuencia lógica correspondiente a las necesidades del sistema.

A continuación se describen primeramente las secuencias globales para ambas tarjetas y finalmente se enfoca en los casos específicos de cada una de ellas.

El procedimiento general para el diagnóstico de las tarjetas consistirá de varios elementos esenciales entre los cuales se encuentran:

- Al iniciar el sistema desde el computador se cargan automáticamente los límites entre los cuales se deben encontrar cada una de las pruebas (Carga de límites).
- El sistema se encuentra en un proceso de espera de parte de cualquiera de los sistemas de diagnóstico, cuando alguno de estos se accione se introducirá a la secuencia propia del mismo (Espera de petición).

- Con la solicitud anterior el sistema adquiere la cantidad de esclavos a los que debe tener acceso y procede a diagnosticarlos (Cambio de esclavo).
- Inicio de comunicación con cada uno de los esclavos (Iniciar Modbus).
- Realización de secuencia de pruebas (Secuencia de Pruebas).
- Finalización de la comunicación con cada uno de los esclavos (Finalizar Modbus).
- Se procede a la verificación que los rangos obtenidos en cada una de las pruebas en las diferentes tarjetas son los correctos (Verificación de Resultados).
- Además en caso de error en conexión este también se reportará en el sistema (Error de conexión).
- Envío de resultados obtenidos en el diagnóstico para que estos puedan ser visualizados en el sistema de diagnósticos. (Envío de resultados).

Para una explicación más detallada de estas se analizarán una a una las mismas. Cabe destacar que los diagramas de flujos correspondientes a estas secuencias se encuentran en el Apéndice A.

Carga de límites.

Esta secuencia abarca la carga de los límites necesarios para asegurar el funcionamiento de las tarjetas a diagnosticar, estos límites son cargados desde un archivo de Excel, el cual contiene los valores correspondientes a los datos de respuesta que deben enviar los diferentes registros una vez que estos sean solicitados, además de los rangos de tensión entre los cuales debe operar el sistema para cada prueba. En la figura 4.7 se muestran algunos de los valores correspondientes a los límites.

12	<Step Name>	Limits.High	Limits.Low
13	Verify AO1 Output Control Initial	33	33
14	Verify AO1 Output Control Final	55	55
15	Verify AO1 Low Ref	0	0
16	Verify AO1 Fault Initial	0	0
17	Verify AO1 Value Low	0.1	-0.1
18	Verify AO1 Fault Low	0	0
19	Verify AO1 Sense Low	50	0

Figura 4. 7. Listado de límites por cargar.

Espera de petición.

En esta secuencia el programa se encuentra esperando una petición de parte de cualquiera de los sistemas de prueba, lo cual se realiza mediante una interrupción al programa, Labview provee la posibilidad de utilizar las interrupciones mediante VISA Wait on event, de manera que el computador no se encuentra en espera de un dato, si no que si este ingresa se realiza una

interrupción al procesamiento haciendo de este se dirija a la petición solicitada. Como parte de esta secuencia se ingresa dentro de la petición el número de tarjetas que el operador ingreso y se solicitan los códigos de barras de las mismas.

Cambio de esclavo.

La secuencia del programa se encarga de modificar la ubicación del esclavo al que se desea realizar la prueba, dependiendo de la cantidad de tarjetas ingresadas, cada vez que se realiza la verificación de una de estas se debe continuar con la siguiente, esta activación del esclavo es realizada mediante el la línea de ENABLE, la cual es controlada por el microprocesador mediante la selección en un multiplexor, por lo que la secuencia consta del envío de esclavo al que se desea tener acceso y la confirmación de la indicación.

Iniciar Modbus.

Labview como sistema de programación gráfica proporciona una serie de librerías encargadas de comunicación y utilización de diferentes protocolos entre los que destacan VISA y Modbus. De esta manera la inicialización de Modbus es posible de manera sencilla mediante una función de la librería Modbus llamada Modbus Serial Init, la cual se encarga de preparar al esclavo para la comunicación mediante este protocolo brindando los datos a utilizar como Baud rate, paridad, bits de parada, el tipo de comunicación RTU o ASCII entre otros . Sin embargo se debe verificar que el sistema ha sido accedido correctamente por lo que se realizan adicionalmente el diagnóstico de la inicialización.

Secuencia de Pruebas.

Indica las funciones que se deben llevar a cabo para el análisis de las tarjetas implicando modificación de registros, diagnósticos, obtención de estas modificaciones y comprobación de CRC entre otras. Esta etapa es de suma importancia y se presentará más detalladamente adelante.

Finalizar Modbus.

Esta consta de la terminación de la comunicación con los diferentes esclavos una vez que se han culminado las pruebas correspondientes.

Verificación de Resultados.

Esta secuencia se encarga de la comparación directa de los límites cargados desde el archivo de Excel con los datos adquiridos desde las tarjetas mediante las pruebas. Esta identifica las fallas y la tarjeta en las cuales se generaron y devuelve un mensaje correspondiente encriptado el cual consta de un arreglo de datos binarios que indican la o las tarjetas con fallos, así como el fallo correspondiente en cada una de ellas. Si se da la existencia de más de uno de estos fallos el sistema esta secuencia contendrá un arreglo String con la palabra “Múltiples Errores” indicando al operador que debe dirigirse al reporte para obtener más información de estos errores.

Error de Conexión.

Esta secuencia es utilizada solamente en el caso que el sistema presentara algún error durante el proceso de envío, de recepción de datos, o durante las secuencias misceláneas. Implica la presentación de un error en la pantalla del computador así como una advertencia de error en el LCD del microprocesador, en el cual se indicará el Error ocurrido con código y descripción.

Envío de resultados.

Encargada de la transmisión de la información al microprocesador para que esté presente la información de la verificación mediante leds así como mediante el Display lcd indicando al operador el resultado de la prueba. La secuencia alberga tanto el proceso de envío de datos como de respuesta que indican que la recepción de estos ha sido correcta.

4.2.4 Diseño de secuencias de Pruebas.

Las secuencias de prueba programadas son las encargadas de brindar al sistema maestro las indicaciones de los registros, bobinas, funciones y subfunciones de diagnóstico que se deben evaluar en las tarjetas esclavas, por ende son de suma importancia para la correcta prueba de funcionalidad en las tarjetas. En ambas tarjetas se presentan secuencias de casos diferentes así como un orden específico por seguir en las mismas. En las tablas 4.5 y 4.6 se muestran las respectivas solicitudes e instrucciones que el maestro debe brindar al esclavo.

Tabla 4. 5. Instrucciones para el diagnóstico Analog Board.

Instrucciones para esclavo	Cód. Función	Dirección de registro/subfunción	Dato introducido
RAM CRC OFF	08	33	Ninguno
A01 Output Control	06	2021	X25
A01 Output Control	03	2021	Ninguno
A01 Lo Refence	06	2022	X291
A01 Lo Refence	03	2022	Ninguno
RAM CRC ON	08	34	Ninguno
A01 HDWR Fault Initial	03	4608	Ninguno
A01 4MA	08	38	Ninguno
A01 Out sense data	03	4609	Ninguno
A01 12MA	08	39	Ninguno
A01 20MA	08	40	Ninguno
A01 A02 Normal	08	45	Ninguno
A01 Disable	08	306	Ninguno

Tabla 4. 6. Instrucciones para el diagnóstico DDC.

Instrucciones para esclavo	Cód. Función	Dirección de registro/subfunción	Dato introducido
SM Stream IR	08	45059	Ninguno
DDC Multidrop Mode	08	101	Ninguno
NET Relay feedback A	03	4716	Ninguno
DDC CH Terminate	08	108	Ninguno
DDC CH Bias	08	106	Ninguno
DDC Repeater Mode	08	103	Ninguno
DDC Relay Watch OFF	08	84	Ninguno
DDC CH Unterminate	08	109	Ninguno
DDC CH UnBias	08	107	Ninguno
DDC Node Bypass	08	105	Ninguno
Status	05	8	Ninguno

En ambas tablas se presentan las ubicaciones de memoria a las que se desea hacer referencia ya sea para diagnosticar, para introducir un valor o incluso para obtener el dato guardado en el mismo en un determinado momento. Estas instrucciones deben ser brindadas por la placa maestra Main al esclavo el cual en este caso sería la tarjeta deseada. Las secuencias de casos correspondientes a cada tarjeta se presentan en las figuras 4.8 y 4.9.

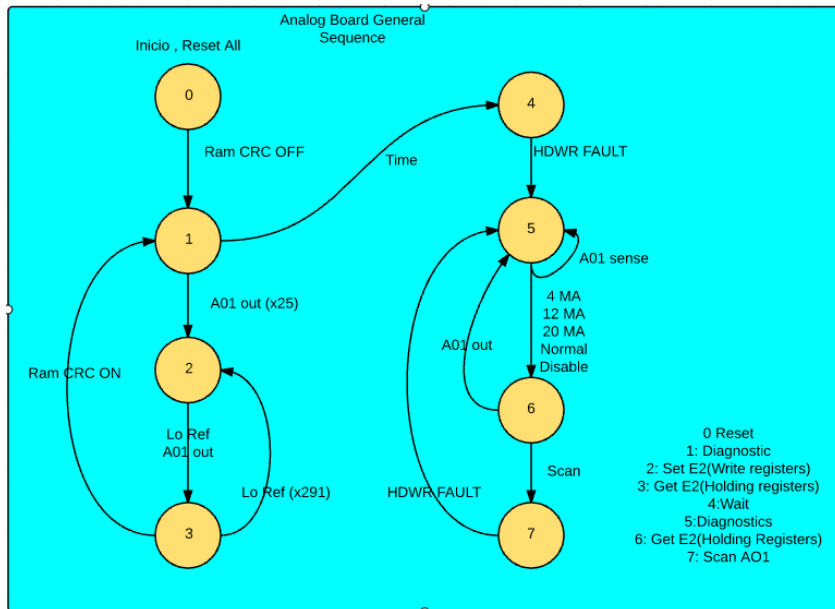


Figura 4. 8.Máquina de estados para la secuencia Analog Board.

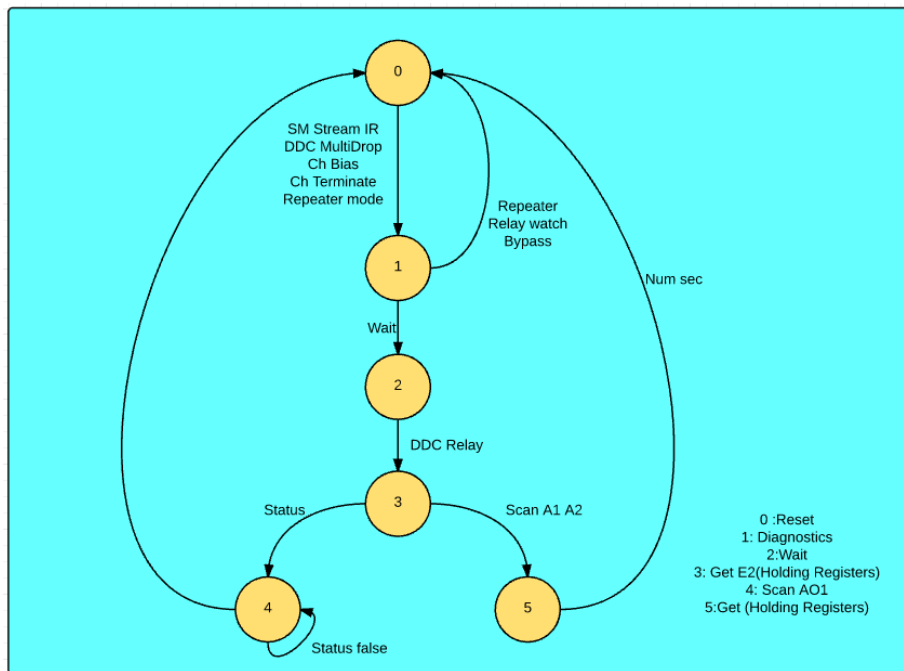


Figura 4. 9. Máquina de estados para la secuencia DDC Board.

En la figura 4.6 se mostró la máquina de estados general correspondiente a las secuencias de prueba para la tarjeta Analog Board. Esta consiste en primeramente la colocación de todas las variables en sus valores nulos o iniciales, luego se realiza la solicitud de diagnóstico de parte del maestro,

una vez para verificar que la redundancia cíclica se encuentra sin errores, posteriormente se introducen nuevos parámetros al registro 2021 mediante la instrucción Write register “06”.

Una vez ingresados los nuevos datos se verifica que estos han sido correctos mediante Holding Registers “03”, igualmente se modifica el registro 2022 y se verifica su cambio de estado, luego se procede nuevamente con los diagnósticos mediante diferentes subfunciones, tal y como se muestra en la tabla 4.4, cambiando de esta manera entre los diferentes modos de operación de la tarjeta e identificando los valores de respuesta en los registros mediante las funciones de Holding Registers, así como su valor analógico de salida mediante la función Scan, quien es realizada por el Arduino a partir del ADC interno que el mismo posee para las entradas analógicas.

Por otra parte en la figura 4.7 muestra el comportamiento de las diferentes secuencias realizadas para la tarjeta DDC board, en este caso al existir varias funciones que poseen un procedimiento similar se utiliza un valor numérico para identificar la secuencia en la que se encuentra, primeramente se realiza el Reset de todas las variables involucradas, luego se procede a realizar el diagnóstico respectivo a SM Stream Ir para averiguar el estado actual de la tarjeta, este se realiza mediante la sub-función “45059”, luego según el estado en que se encuentre se realiza un nuevo diagnóstico cambiando así los parámetros internos y forzando el sistema a un comportamiento específico. Una vez que se ha dado esto se presenta una pequeña espera y se procede a verificar el estado del registro “4716” y se guarda este dato para la comparación de límites. Finalmente se obtienen los valores analógicos generados a partir de este modo de funcionamiento.

Tal y como se observó en el marco teórico el protocolo Modbus consiste en el ordenamiento de estos datos de manera que la secuencia sea entendible por el esclavo asegurando la llegada y entendimiento de la instrucción y brindando la respuesta correspondiente al mismo.

Labview proporciona una librería especializada en el uso de este protocolo facilitando de esta manera la transmisión de las instrucciones. En esta librería se utilizó el módulo de maestro “MB Serial Master Query (poly)” un módulo polimórfico, es decir que puede realizar diversas funciones entre las cuales se encuentran Read coils, Holding Registers, and Write Registers. Todas estas funciones aplican según el protocolo Modbus explicado en la sección de marco teórico, sin embargo este sistema polimórfico no posee la función de diagnostics función 08 y la cual es muy utilizada en para la petición de estado de la tarjeta en las secuencias deseadas por para el diagnóstico del sistema, es por esto que se procedió a modificar la librería para introducir esta nueva función de manera alterna. En la figura 4.10 se muestra la modificación de la librería.

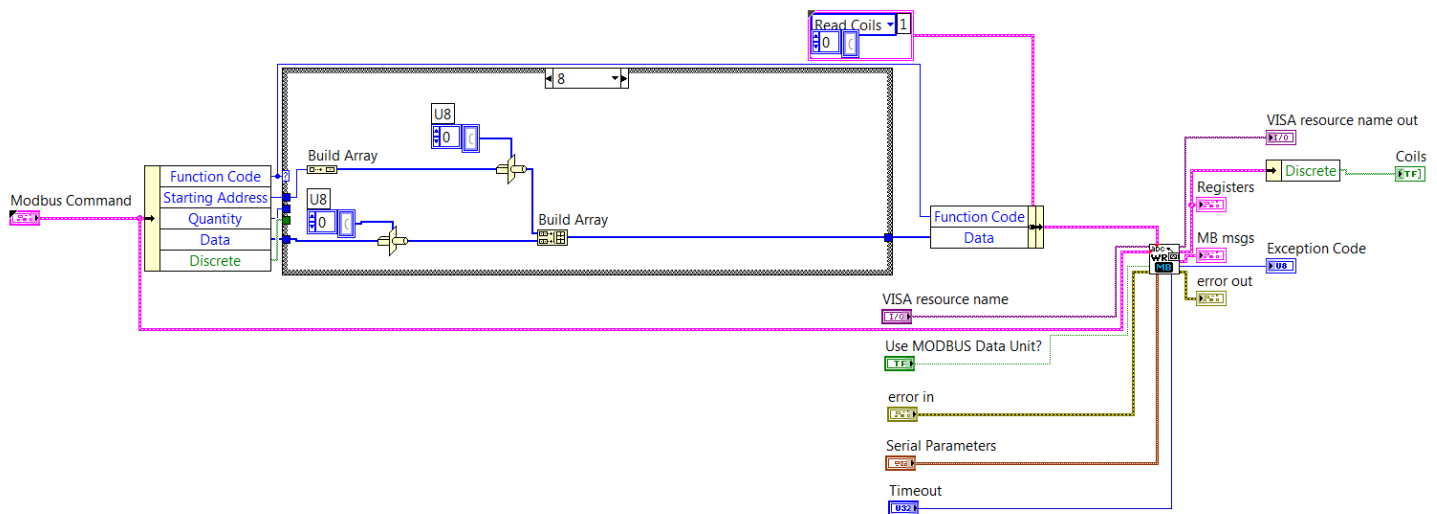


Figura 4. 10. Modificación para introducir diagnósticos.

Tal y como es posible observar en la figura se introdujo una secuencia de casos a partir de la cual se pueden introducir las funciones necesarias a realizar a partir de su código de función y este ingresa al sistema unificando el dato según corresponda mediante el manejo de arreglos, en las primeras posiciones la ubicación de la dirección a la que se desea hacer referencia, y seguido a este el dato que se desea ingresar. De esta manera se forma la secuencia de función diagnostics en lo que respecta a la parte de datos, aunado a esto se le traslada al sistema el código de función, estos datos sustituyen los datos del sistema actual permitiendo que los mismos sean ingresados de manera directa a la escritura realizada sobre el esclavo, el cual responde según lo solicitado aunado CRC al paquete, así como el estatus solicitado.

En el caso de diagnostics se coloca primeramente la función a realizar en este caso “08” luego la subfunción la cual cambia dependiendo de lo que se desea realizar para esto se poseen cuatro bytes, por ejemplo para el caso de RAM CRC OFF “0021” que es el correspondiente a la subfunción, los datos que en todos los casos de diagnósticos corresponden a “0000” y finalmente la comprobación de redundancia cíclica CRC. Cabe destacar que la respuesta del esclavo posee la misma forma que la secuencia enviada por el maestro.

También se le agregó al sistema un indicador tanto del mensaje enviado como el recibido el cual se identifica como MB msgs debido a la necesidad de comprobar que ambos poseen el mismo mensaje, ya que tal y como se comentó anteriormente cuando se realiza un diagnóstico el esclavo debe responder con el mismo mensaje al maestro.

Finalmente para este caso se coloca el booleano de ingreso en True para indicar al módulo que se realizará el ingreso de los datos directamente como cadena “Modbus Data Unit”.

De esta manera se realizan las secuencias para las diferentes funciones con un orden específico para realizar modificaciones en los registros así como la verificación de estas modificaciones y de la afectación de estas en tensiones de salida.

Otro aspecto a destacar es la utilización del microprocesador para la recolección de estos datos analógicos, en este caso se modificaron los rangos de tensiones de entrada para asegurar que estos fueran legibles por el sistema de conversión analógico digital del microprocesador.

En la figura 4.11 se muestran estos acondicionamientos de señal mediante divisor de tensión.

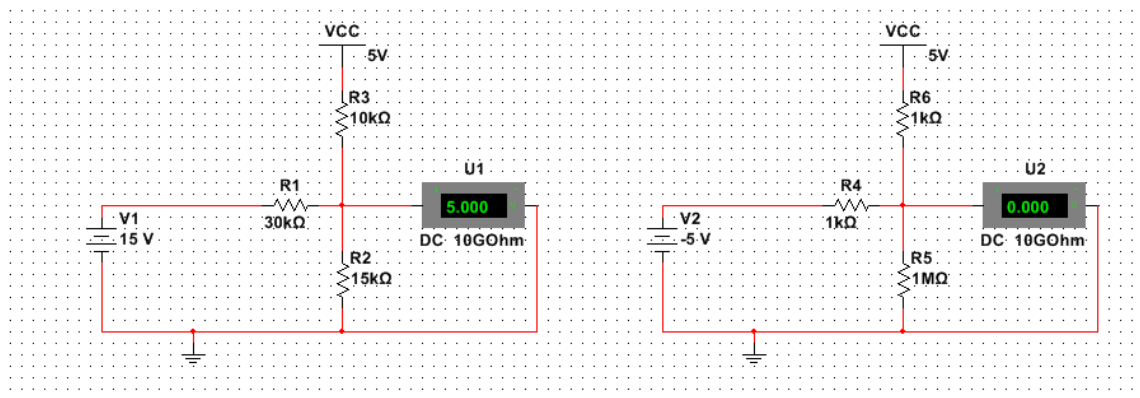


Figura 4. 11. Acondicionamiento de señal por divisor de tensión.

En la figura anterior se muestra el acondicionamiento de señal la cual modifica los rangos de analógicos de cada una de las tarjetas de +15 y -15V a 0-5V para el caso de la Analog Board, así como de +5 y -5V a 0-5V en la tarjeta DDC. Cabe destacar que este acondicionamiento de la señal podría generar algún tipo de pérdida de resolución por lo que en la sección de conclusiones y recomendaciones se hace referencia a algunos instrumentos de adquisición de datos con conexión inalámbrica que podrían acoplarse fácilmente al sistema.

Capítulo 5: Resultados y análisis.

Para llegar a resultados concluyentes del funcionamiento del sistema, tanto en programación como en hardware se realizaron una serie de simulaciones que muestran posibles casos a los cuales el sistema podría enfrentarse, las cuales se presentan a continuación, analizando el por qué estas respuestas son o no correctas. Además, estas pruebas permitieron anotar algunas observaciones para hacer recomendaciones sobre el diseño, e incluso algunas correcciones.

En la primera etapa se procedió a la prueba de recepción de petición realizada por el esclavo y como el programa recibe esta solicitud, para esto se utilizó una simulación por medio de Proteus y su posibilidad de conectividad con Labview mediante puertos virtuales de manera se procedió a realizar una solicitud cuyo procedimiento consiste en la selección de la cantidad de tarjetas que se desean probar, luego que se ha comprobado que la cantidad es la deseada se procede realizar la lectura de los códigos de barra mediante el escáner. Cabe destacar que estos códigos de barras constan de 7 caracteres entre números y letras, luego se oprime el botón de inicio con lo que el Arduino toma el dato y mediante comunicación serial procesa la petición transmitiendo al mismo tiempo el dato en forma hexadecimal. Luego se envían los códigos de barras respectivos a las tarjetas y estos son colocados según el orden en el que fueron ingresados por el operador. Tanto los esquemáticos correspondientes como la programación en Arduino y Labview se encuentran en el Anexo B.

De esta manera en el panel de control del sistema en Labview se presentan tanto la cantidad de tarjetas seleccionadas como los códigos de barra que cada una de estas posee. En la figura 5.1 se muestra los resultados de simulación para la selección de 4 tarjetas y sus respectivos códigos de barras.

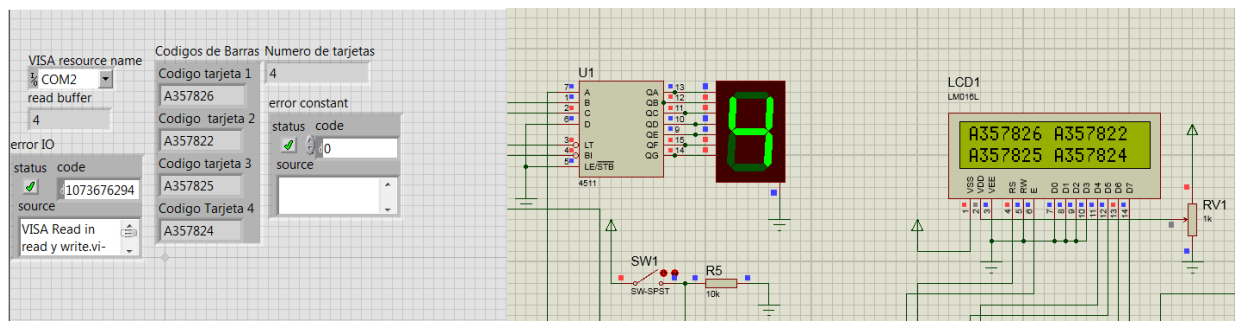


Figura 5. 1. Petición de diagnósticos.

Como se puede observar en la figura los datos tanto de la cantidad de tarjetas como de sus respectivos códigos de barra han sido transferidos de manera correcta del sistema al programa, además es fácil observar que estos se han ordenado de manera correcta. La espera de petición es un método eficiente debido a que introduce el concepto de interrupción al programa haciendo que este no se encuentre ocupado únicamente en la espera de un dato si no que él mismo se puede encontrar realizando otros algoritmos hasta que se realice la interrupción.

Cabe destacar que el dato se envía de manera hexadecimal mediante el protocolo VISA y una vez que este ingresa se guarda como una variable Enum en el programa en Labview ya que el mismo será utilizado en secuencias siguientes.

Otra de las secuencias que se deben evaluar es la que implica el cambio de esclavo, el cual se realiza mediante la selección del esclavo al que se desea tener acceso por parte del microprocesador. Este procedimiento se debe realizar una cantidad de veces correspondiente a la cantidad de tarjetas introducidas, el programa consiste básicamente en el envío del esclavo que se desea habilitar, este esclavo se envía mediante una String y este es recibido por el Arduino, dirige al mismo como un bit y se modifican las entradas de selección del decodificador. En la figura 5.2 se observa el funcionamiento del sistema, en este caso se realiza una representación de la línea de habilitación de las tarjetas ENA con cada uno de los Leds. De esta manera se habilita la tarjeta deseada para la carga de las secuencias de diagnósticos, escritura y lectura de registros entre otros.

De esta manera se comprueba la funcionalidad teórica de la selección de esclavo.

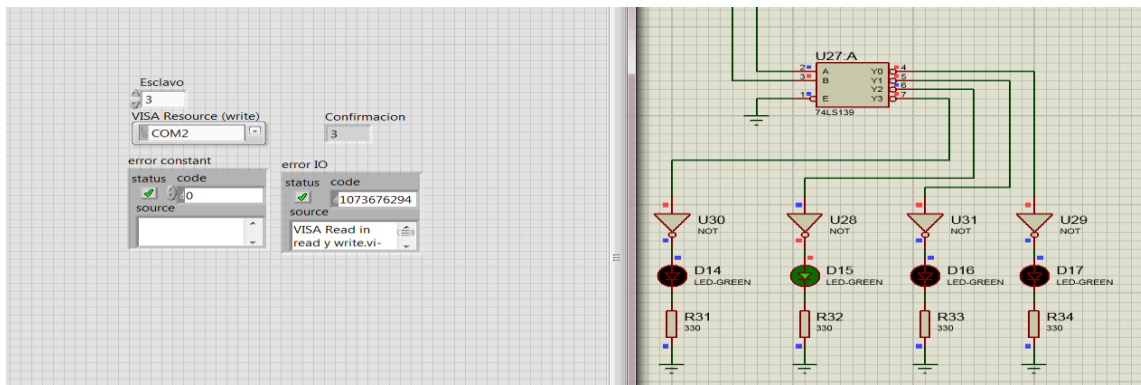


Figura 5. 2. Cambio de esclavos.

Después de realizar la solicitud de petición y la asignación de esclavo el sistema, procede a realizar las secuencias correspondientes a la lectura escritura de registros, así como obtención de los valores analógicos de respuesta y diagnostics.

Para cada una de estas secuencias se presentarán ejemplos que comprueban su correcto funcionamiento. Primeramente para la función de verificar tensiones de salida para la tarjeta Analog se debe seguir un orden específico de peticiones el cual se muestra en la máquina de estados general para esta tarjeta, refiérase a la figura 4.8. En la tabla 5.1 se muestran cada una de las Funciones a realizar y su conformación para él envío según el protocolo Modbus.

Tabla 5. 1. Funciones para la secuencia Verify Output Voltage.

Instrucciones para esclavo	Cód. Función	Dirección de registro	Dato introducido	Código Hexadecimal
RAM CRC OFF	08	33	Ninguno	000800210000D7
A01 Output Control	06	2021	X37	000607E50037DB
A01 Output Control	03	2021	Ninguno	000307E5000110
A01 Lo Refence	06	2022	X00	000607E63000DD
A01 Lo Refence	03	2022	Ninguno	000307E600010F
RAM CRC ON	08	34	Ninguno	000800220000D6
A01 HDWR Fault Initial	03	4608	Ninguno	000312000001EA
A01 4MA	08	38	Ninguno	000800260000D2
A01 Out sense data	03	4609	Ninguno	000312010001E8
A01 12MA	08	39	Ninguno	000800270000D1
A01 20MA	08	40	Ninguno	000800280000D0
A01 A02 Normal	08	45	Ninguno	0008002D0000CB
A01 Disable	08	306	Ninguno	000801320000C4

En la tabla anterior se presentan la conformación de las funciones para él envío de datos mediante el protocolo Modbus RTU. Un ejemplo a seguir es la secuencia GET E2 Output Control, en esta se desea conocer el estado del registro 2021 decimal; para esto la conformación de la función se realiza de la siguiente manera: 1 byte de dirección de esclavo, 1byte de código de función, 2 bytes de dirección de registro, 2 bytes de cantidad de registros por consultar, 1 byte de comprobación de redundancia cíclica (CRC). Esta conformación se muestra en la tabla 5.2

Tabla 5. 2. Función Holding Registers para el registro 2021.

Esclavo	Cód. función	Dirección inicial	Cantidad	CRC
01	03	07E5	0001	0F

Según el protocolo Modbus cuando el maestro realiza una petición de Holding Registers el esclavo debe responder de una forma semejante, primeramente 1 byte de dirección de esclavo, 1 byte de código de función, luego un contador de bytes el cual indica la cantidad de bytes de los cuales se encuentra conformado el mensaje de respuesta, normalmente 2 Bytes por cada registro consultado. La sección de datos coloca los registros de forma ascendente en el mensaje, es decir los primeros 2 byte corresponden a el primer de los registros consultados, y así sucesivamente. Finalmente se envían los datos de Comprobación de redundancia Cíclica. En la tabla 5.3 se muestra el posible mensaje de respuesta para esta función.

Tabla 5. 3. Función Holding Registers para el registro 2021, respuesta.

Esclavo	Cód. función	Contador de bytes	Bytes	CRC
01	03	01	21	5D

Una vez que se conocen tanto la función de envió como de respuesta se procede a observar el comportamiento del sistema mediante simulaciones el cual se muestra en la figura 5.3.

En este desde Labview se procede a enviar un solo mensaje con la indicación de solicitud de datos dentro del registro específico, el sistema se configura la ubicación del sistema en el puerto en este caso virtual “COM2”, en el esclavo número uno, el cual corresponde a la tarjeta 2, con transmisión de tipo ASCII; cabe destacar que esta también se puede configurar como RTU, además se selecciona la función a realizar la cual en este caso es Output Control, la cual se encarga de obtener los datos respectivos al registro de salida para el control analógico de la tarjeta.

Como se puede observar en la misma los datos transferidos son los mismos a los esperados transferir por lo que el sistema cumple a cabalidad las peticiones de información hacia los registros especificados, lo cual es parte importante de la secuencia de pruebas ya que estos datos de respuesta de los registros son quienes son almacenados y comparados con los límites específicos para cada una de las tarjetas. Los valores hexadecimales de estos registros son quienes indican al sistema que la tarjeta responde de manera correcta a las peticiones y solicitudes para las cuales fue diseñada que en finalidad es lo que se espera mediante las pruebas de funcionamiento.

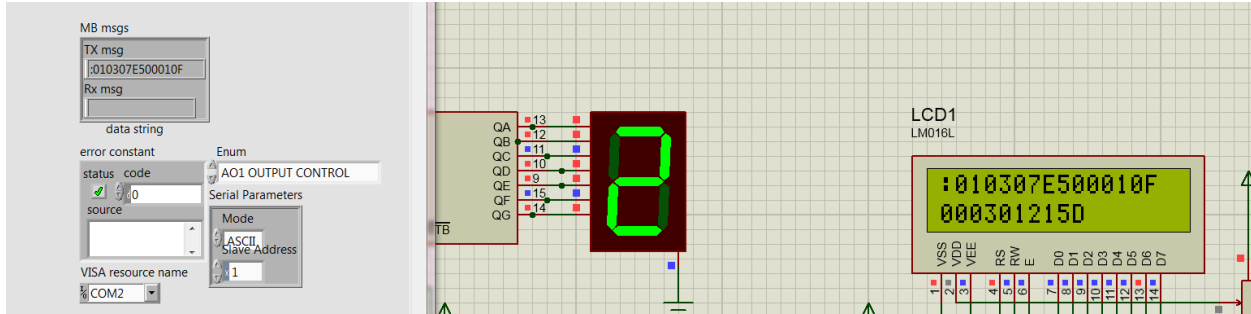


Figura 5. 3. Secuencia Holding Registers, en el registro 2021, petición.

Igualmente otra de las secuencias importantes es la de diagnóstico la cual se encarga de solicitar al esclavo información respectiva a diversas situaciones que él mismo experimenta durante el proceso de pruebas, entre estas destacan restablecer contadores, realizar consulta acerca de errores de conexión maestro esclavo entre otros, en este caso se presenta una función de diagnóstico específica para la consulta de CRC en el sistema. “Diagnostics CRC ON”, la misma corresponde al código hexadecimal de envío “000800220000D6” debido a que se coloca como subfunción “34 decimal” correspondiente a “0022” como valor hexadecimal en la posición de 2 bytes correspondientes a código de subfunción de la misma manera se encuentran en los 2 bytes siguientes lo que son datos que en este caso no se envían. La respuesta del esclavo para los casos de diagnóstico típicamente es exactamente igual a el valor ingresado por lo como respuesta se obtendrá la misma función “000800220000D6”. En la figura 5.4 se muestra el comportamiento obtenido mediante simulaciones de esta función.

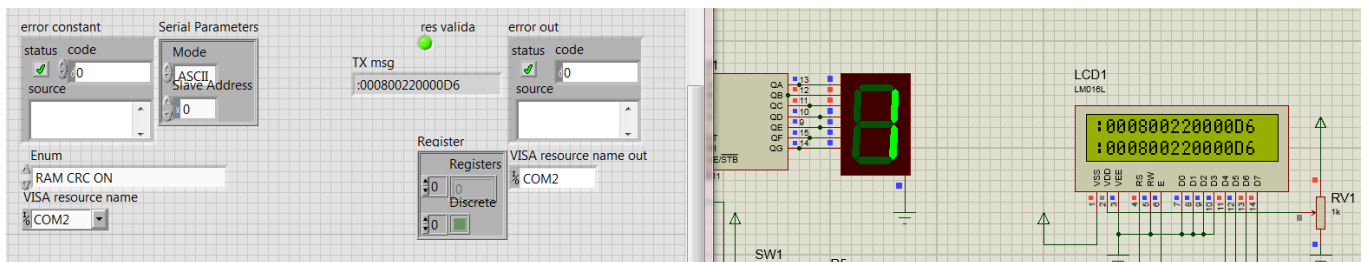


Figura 5. 4. Secuencia Diagnostics, subfunción 34, petición.

En la figura anterior se muestra como el mensaje ingresa correctamente por medio de la comunicación el mismo se puede visualizar en la pantalla del display en proteus, al igual que como este es retornado, por una secuencia formada por el mismo mensaje ya que este es uno de los requerimientos de la función de diagnóstico. Por otra parte en Labview se observa el mensaje de

salida, así como el indicador “Res valida” el cual especifica que la respuesta obtenida por Modbus es igual que la respuesta transmitida.

De esta manera se puede identificar la funcionalidad de las secuencias de diagnósticos implementadas en el sistema general. Cumpliendo con las expectativas propuestas para esta específica función.

Finalmente y una de las partes más importantes de esta transmisión es la verificación de resultados, ya que permite al operador enterarse del funcionamiento de las tarjetas, indicando si se ha dado un resultado favorable para cada una de ellas o si algunas presentan errores en su conformación. Por lo cual deben revisarse sus componentes. En este se presenta al operador tanto la tarjeta con fallos como también la ubicación de este fallo indicando el registro del que fue proveniente el mismo, lo que agiliza la búsqueda de los errores. A continuación en la figura 5.5 se muestra la simulación de esta secuencia de verificación de fallas, para esto se colocaron manualmente los valores de la falla en Labview para poder observar y comparar cómo esta refleja un mensaje al operador.

Como se observa en la imagen 5.5 primeramente se seleccionó el número de tarjetas introducidas, en este caso una, luego se introdujeron los parámetros respectivos a los resultados manualmente los cuales en este primer caso corresponden a los límites reales y por tanto se puede observar que el arreglo de salida correspondiente al estado de las tarjetas se es “1111” ya que como no se utilizan las demás estas se encuentran como default en “1” además si se realiza al traslado al esquemático este presenta en los leds tres instrucciones; en la parte superior el led verde indica que ya se terminó de probar las tarjetas, el led verde de la parte derecha indica que todas que se encontraban en prueba aprobaron la prueba , y los led verdes en la parte inferior indican que se ha pasado la prueba en todas las tarjetas.

El operador debe inferir que los demás leds se encuentran activos, a pesar que no se introdujeran las tarjetas, esto para no generar ningún tipo de confusión en el mismo.

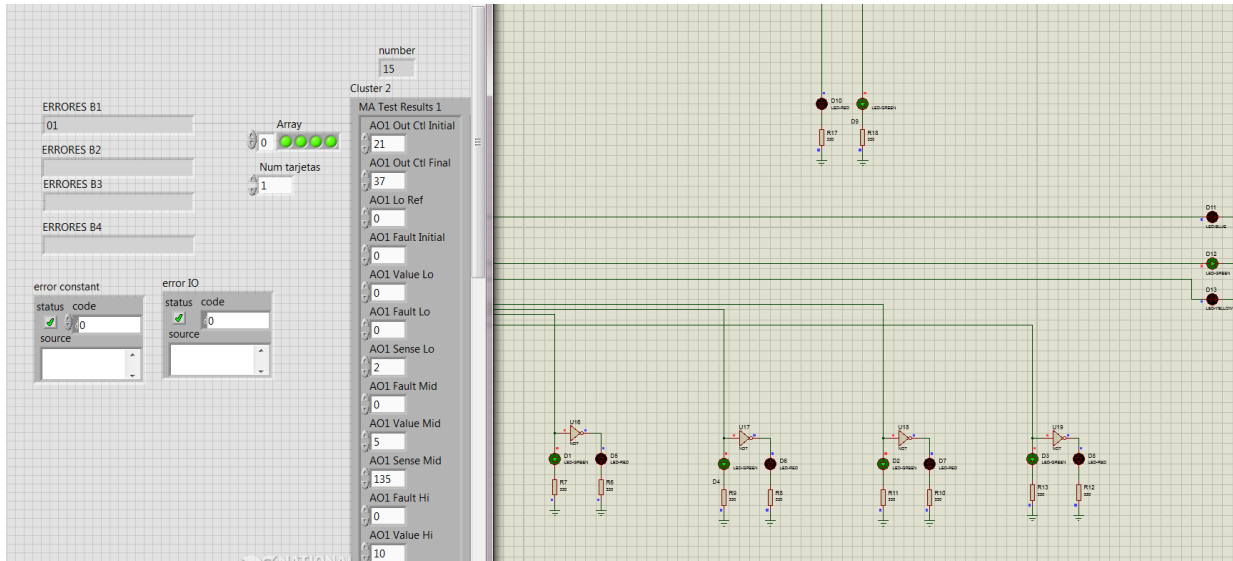


Figura 5. 5. Verificación de Errores caso 1 tarjeta sin errores.

En la figura 5.6 se mostrará el caso inverso en que la tarjeta presenta un error, esto con la intención de observar lo que sucede en este caso.

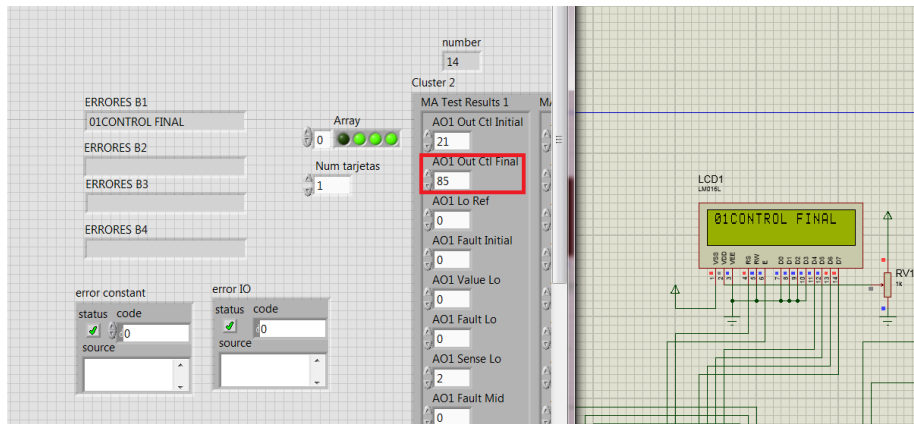


Figura 5. 6. Verificación de Errores caso 1 tarjeta, con error.

Como se puede observar en la figura 5.6 se formuló el caso en que la tarjeta presenta un error, el cual se encuentra enmarcado en un cuadro rojo, si este se compara con la figura 5.5 se puede observar las diferencias.

En este caso el error ocurrido generó que la tarjeta 1 no aprobará el diagnóstico y por lo tanto el led indicador que corresponde a esta tarjeta se encuentra en rojo refiérase a la figura 5.7. Además

se le muestra al operador mediante el display LCD la ubicación del error que produjo que la tarjeta fuese rechazada para que este pueda encontrar de manera más rápida el error.

Por otra parte el led azul a la derecha de la imagen muestra que se produjo un error en alguna de las tarjetas.

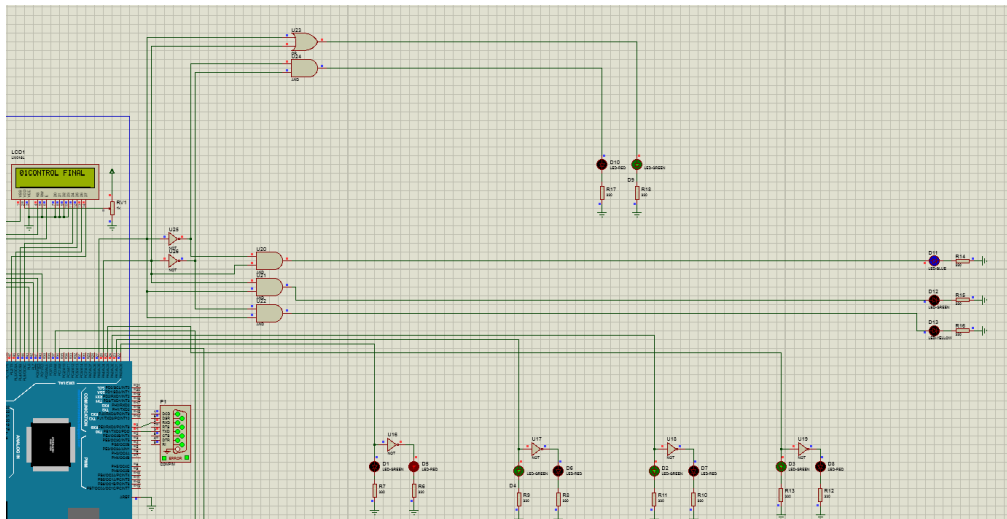


Figura 5. 7. Verificación de Errores caso 1 tarjeta, con error.

De este modo es posible visualizar que se cumplen las características necesarias esperadas para esta función ya que proporciona al operador las herramientas necesarias para ubicar posibles errores en las tarjetas. De manera que este puede resolver estos defectos directamente sin tener que trasladarse al Área de Test de la empresa para observar el reporte de la misma. Lo cual ahorra tiempo de re-trabajos, el cual es uno de los problemas que agobia esta Área específica y que ralentiza la producción en la misma.

Como se puede observar a lo largo del capítulo el diseño del sistema ha involucrado diversos conceptos teóricos que se han comprobado a partir de las pruebas de simulación realizadas anteriormente, es importante discutir los resultados de estas debido a la importancia que las mismas poseen en el proyecto.

Primeramente la realización de un sistema de conexión inalámbrica el cual permite a los sistemas de prueba tener la capacidad ser manejables a lo largo de la empresa es uno de los puntos críticos debido a que de estos depende la colocación de células de trabajo donde se realiza la formación de una tarjeta desde su despanelizado, limpieza de la misma, soldado de componentes adicionales,

colocación de conectores, etiquetado, hasta su prueba funcional, la capacidad de movilización de los aparatos de prueba permitirá la formación de este tipo de células, desde las cuales el operador podrá realizar los diagnósticos y obtener los resultados de los mismos. Es por ende que la utilización de sistemas como el que se propuso mediante transferencia Rf a 2.4Ghz conectados a los microprocesadores cumple las expectativas debido a sus rangos de alcance los cuales sobrepasan los 200 pies de longitud, además este Shield para Arduino proporciona una herramienta relativamente sencilla de usar para la transferencia de información.

Otro aspecto importante a discutir la creación del sistema con múltiples tarjetas ya que este permite realizar de manera más rápida las pruebas, ya que el operador ahorra el tiempo de setup o preparación entre las tarjetas. Actualmente el proceso de testeo de una de estas tarjetas conlleva en promedio 2 minutos con 15 segundos para el caso de la tarjeta DDC y 1 minuto 30 segundos para el caso de la tarjeta Analog board, además el tiempo de preparación entre cada tarjeta toma al menos 30 segundos. La meta fijada en este caso para el proyecto es realizar la prueba simultánea y que este procedimiento se realizará en un tiempo menor que las pruebas actuales, por tanto uno de los beneficios de centralizar las tarjetas ahorra el tiempo tanto en proceso de setup como en proceso de prueba. Ya que si sumamos las duraciones de las tarjetas en promedio 4 tarjetas DDC serían testeadas en 10 minutos y 30 segundos el cual es el tiempo a superar. Además un aspecto a tomar en cuenta es que las programaciones anteriores realizaban de manera más general, por lo que el sistema necesitaba ahondar más en la programación para encontrar la prueba específica a realizar, además la presencia de mayor cantidad de casos, ciclos hacen que en proceso se tarde más en la búsqueda de las funciones y los registros a acceder.

Al ser el sistema de pruebas inalámbrico, este debe evitar la necesidad de movilización del operador desde la célula de trabajo al área de test, por lo tanto el aparato de pruebas debe ser totalmente autónomo e independiente en cuanto a los diagnósticos, procesamiento de datos, generación de solicitudes y muestra de resultados.

En este aspecto tal como se pudo observar en los diferentes ejemplos el sistema es capaz de procesar solicitudes desde su ubicación permitiendo al operador seleccionar la cantidad de tarjetas que desea ingresar, además permite la introducción de códigos de barras ligados a estas tarjetas con la finalidad de poseer un reporte más específico dirigido a cada tarjeta testada, además el

mismo guía al operador entre las diferentes procesos indicando exactamente en qué periodo de la prueba se encuentra; “inicialización”, “secuencias”, “verificación de fallas”, entre otras.

Una vez que el proceso de diagnósticos finaliza el sistema presenta una serie de indicadores capaces de brindar al operador los resultados de las pruebas mostrando las tarjetas que presentan problemas, así como los problemas respectivos para que este realice las reparaciones en su puesto de trabajo. La indicación de error de comunicación es muy importante ya que advierte al operador que se debe reiniciar el sistema debido a que este ha perdido toda comunicación con la computadora y por ende las tarjetas podrían presentar errores inexistentes en las mismas.

En fin el sistema es completamente autónomo ya que brinda al operador conocimiento de todas las variables presentes en el diagnóstico, realizando peticiones y brindando resultados de manera clara y concisa mediante la interfaz.

La funcionalidad de los sistemas es sin duda el aspecto más importante a evaluar, debido a que de este depende la puesta en marcha del proyecto a nivel físico. La funcionalidad depende específicamente si las secuencias de funciones realizan la modificación de datos así como captura de resultados de forma correcta, como se pudo observar tanto en los diagnósticos, como en solicitud de datos a los registros, el sistema se acoge a las reglas impuestas en el protocolo Modbus, permitiendo la modificación de variables mediante la función SET “05” realización de diagnósticos mediante la función “08”, y su respectiva subfunción, finalmente la recopilación de estos datos modificados o de registros predeterminados mediante la función GET E2 “03”. Las cuales siguen una secuencia lógica ordena mediante máquinas de estados presentadas anteriormente. De esta manera es posible evidenciar que el funcionamiento general del sistema en cuanto a secuencias respecta cumple con las expectativas permitiendo de este modo realizar las pruebas y obtener los resultados deseados para las mismas.

El costo de un proyecto sin duda decide su factibilidad, es por esto que se debe tomar en cuenta este factor para el diseño del mismo reduciendo los gastos en la mayor cantidad posible. Como se pudo observar en la sección de marco teórico el sistema antiguo posee métodos de adquisición de datos cuyo costo es bastante elevado, por lo que se observó la necesidad de obtener un método de bajo costo para la toma de las señales analógicas que las tarjetas generan y cuyos valores es importante tomar en cuenta para verificar la funcionalidad de las mismas, es por esta razón que se

decidió utilizar el mismo microprocesador utilizado para comunicación para la toma de datos analógicos. Para esto se procedió a la realización de acoplamientos de señal que permitieron al ADC del Arduino capturar los valores correspondientes, sin embargo en este proceso de modificación y acondicionamiento de la señal fue necesario el sacrificio de la resolución de estos datos ya que los DAQ de National Instruments poseían 12 bits de resolución aplicados a 30 volts, mientras que el sistema propuesto presentará una resolución de 10 bits aplicados igualmente a 30 volts, sin embargo en este caso con el acoplamiento esta resolución se ve reducida. Por lo tanto se da una confrontación entre presupuesto y resolución de datos. Ya que como se observa en las tablas 4.1 y 4.2 las diferencias entre los precios del sistema incluyendo estos sistemas y sin incluir los mismos es abrumante obteniéndose inclusive un precio tres veces menor mediante la utilización del microprocesador como sistema de recopilación de datos analógicos también. Sin embargo en la sección de conclusiones y recomendaciones se presentarán posibles artefactos de adquisición de datos analógicos que se podrían adquirir, sumamente novedosos e inalámbricos para brindar mayor capacidad de resolución al sistema.

En cuanto a los beneficios obtenidos por la empresa con la aplicación de este nuevo sistema de pruebas son considerables, ya que la capacidad de prueba de tarjetas por operador se ve beneficiado en un 400% ya que este ahora prueba 4 tarjetas solamente el mismo. Por ende si se realizaran estos sistemas para todos los proyectos la capacidad aumentaría inclusive hasta un 2000% de la actual. En números esto se refleja según la tabla 5.4.

Tabla 5. 4 Beneficio en Tarjetas

Cantidad sistemas	Operadores	Tarjetas/mes
0	4	1000
1	4	4000
2	4	8000
3	4	12000
4	4	16000

En la tabla anterior se puede observar que con la misma cantidad de operadores de producción la capacidad se ve mejorada de 1000 tarjetas mensuales a 16000 tarjetas mensuales.

Una vez analizados todos los aspectos anteriores se puede identificar que el sistema cuenta con la capacidad de brindar al operador un ambiente claro para la realización de pruebas evitando la

movilización de este hacia el área de test , además presenta grandes beneficios en cuanto aumento de capacidad de pruebas se refiere, y con esto la capacidad total de la planta, el costo de este sistema se redujo en gran manera, sin embargo se perdió resolución en los datos lo cual podría generar problemas en cuanto a verificación de límites. Sin embargo el sistema cumple con las expectativas propuestas para la continuación y desarrollo del mismo.

Capítulo 6. Conclusiones y Recomendaciones.

Del trabajo realizado durante 16 semanas para el desarrollo de un sistema inalámbrico para pruebas de diagnóstico de tarjetas del cliente FLOWSERVE, específicamente las tarjetas con mayor demanda por parte de este cliente Analog Board y DDC board, se puede concluir que el diseño abarca la totalidad de los objetivos planteados en un inicio, indicando a la empresa la posibilidad de aumentar su capacidad de producción si el mismo se decidiera realizar. Ya que los beneficios económicos que el mismo plantea en cuanto ahorro de tiempo de operadores, así como el aumento de productividad supera en gran manera al precio que el desarrollo del mismo generaría.

El diseño del mismo contempla algunos sistemas disponibles en la empresa, como base para el mismo además herramientas de software como Labview y Proteus así como software libre como Arduino utilizados tanto para el desarrollo del sistema como para simulación del mismo.

Con base a diagramas del sistema general que se encuentra en la empresa si como listas de funciones a evaluar en las tarjetas se procedió a formular un sistema de diagnósticos específico para la evaluación de varias tarjetas simultáneamente a partir de la multiplexación de las mismas y el control de esclavos además este sistema permite el traslado de las tarjetas a cualquier punto de la planta gracias a la comunicación inalámbrica generado por los módulos RF implementados.

Se desarrolló un sistema eficiente de separación, para la generación de sistemas de pruebas independientes de bajo costo que permiten la prueba de 4 tarjetas simultáneamente, reduciendo los costos en la prueba mediante implementación de sistemas alternativos de toma de datos analógicos, y eliminando los sistemas de adquisición de alto precio.

Se diseñó un sistema de mando encargado del desarrollo y control de pruebas desde el cual el operador puede seleccionar la cantidad de tarjetas a evaluar, así como sus respectivos códigos de barras para el control de resultados y el módulo central se encarga de realizar las pruebas a cada una de las 4 tarjetas así como presentar los resultados respectivos a cada una de ellas, manteniendo control absoluto sobre las pruebas, manejo de variables y solicitudes durante toda la prueba.

Se realizaron secuencias de programación para diferentes funciones de diagnósticos para las tarjetas, en total la cantidad de funciones fueron 33 implementadas en 4 secuencias básicas y 3 generales, las cuales permiten la comprobación de diferentes errores en las tarjetas, identificando

el sector causante de la misma. Cada una de estas funciones fue comprobada en cuanto a funcionamiento y comparada con la esperada teóricamente demostrando su funcionalidad.

Se desarrolló una interfaz de usuario amigable, la cual se encuentra basada en indicadores y muestra de mensajes mediante un display LCD, dichos mensajes guían al operador para la realización de las pruebas y muestran los resultados respectivos una vez que el proceso ha culminado. Además el sistema brinda la información de aprobación o rechazo para cada una de las tarjetas mediante indicadores justamente bajo ellas. Así como indicación de error de conexión en caso que este suceda.

Finalmente se desarrolló un sistema de cambio de esclavo el cual permite al sistema la utilización la formulación de las pruebas para cada una de las tarjetas de manera eficiente, así como almacenar los resultados pertinentes a cada uno de los esclavos y presentar los mismos al final de la prueba así como el almacenaje de estos datos.

Recomendaciones.

Luego de las pruebas de simulación y análisis de resultados así como del diseño de los mismos se pueden mencionar algunas recomendaciones que beneficiarían en gran parte el proyecto.

1. Como se pudo observar en el análisis de resultados con la finalidad de reducir los costos del proyecto se decidió utilizar un método alternativo para la adquisición de datos analógicos el cual proporcionara una lectura de los mismos, sin embargo para esto fue necesario sacrificar la resolución del sistema en cuanto recolección de datos, por ejemplo para el caso de la tarjeta DDC esta posee límites de tensión analógicos los cuales se deben encontrar entre valores como (0.585V y 0.799 V) los cuales son muy cercanos por ende podrían generar problemas de entrecruzamiento del valor esperado tanto hacia el límite superior como inferior y con esto haciendo que la prueba sea rechazada sin una razón real. Es por esto se recomienda el análisis de la compra de nuevos módulos de adquisición de datos los cuales poseen adicionalmente la comunicación inalámbrica (NI - 9201 y NI -9221).

2. En el proyecto se utilizó un sistema de National Instruments para la comunicación RS /485 el sistema NI USB 485, este caso se recomienda obtener sistemas de menor valor y con características semejantes los cuales podrían formar parte de una solución mejorada.

3. En caso que se desee utilizar las secuencias realizadas para las demás tarjetas del Flowserve, estas permiten la adición de casos para cada una de las secuencias tanto diagnostics, como Holding Registers y Write registers, además la secuencia principal también permite la adición de casos para otras secuencias específicas.

Referencias Bibliográficas.

- [1] López, E. (2005, 09 de Agosto). *Protocolo SPI(Serial Peripheral Interface)* Recuperado el 01 de Septiembre del 2015, de <http://www.i-micro.com/pdf/articulos/spi.pdf>
- [2] Forero, N. (Ed.). (2012, 18 de Mayo). *Normas de Comunicación en Serie: RS-232, RS-422 y RS-485* Recuperado el 01 de Septiembre del 2015, de <http://www.unilibre.edu.co/revistaingeniolibre/revista-11/art13.pdf>
- [3] López, E. (2014, 12 de Octubre). *Ingeniería en Microcontroladores: Protocolo RS-485* Recuperado el 01 de Septiembre del 2015, de <http://www.electronica60norte.com/mwfls/pdf/rs-485.pdf>
- [4] National Instruments (2003, 11 de Junio). *Serial Instrument Control Interfaces for USB.* Recuperado el 01 de Septiembre del 2015, de http://www.ni.com/pdf/products/us/usb232_485_whitepaper.pdf
- [5] Ramírez, A. (2013, 02 de Junio). *Comunicación por protocolo RS-232* Recuperado el 01 de Septiembre del 2015, de http://extropynow.weebly.com/uploads/1/6/4/1/16411724/mikroc-u_i-rs232-protocol.pdf
- [6] Kusch, M. (2005, 05 de Octubre). *Diseño del módulo RS-232* Recuperado el 01 de Septiembre del 2015, de <http://www2.elo.utfsm.cl/~lsb/elo311/labs/docs/rs232.pdf>
- [7] Configuración del sistema de adquisición de datos mediante PC (2009, 05 de Octubre). Recuperado el 01 de Septiembre del 2015, de http://www.detcp.upct.es/Personal/Vgarceran3/Practica_1B.pdf
- [8] Ramírez, F. (2009, 21 de Octubre). *Sistemas de adquisición de datos de sensores analógicos y digitales.* Recuperado el 01 de Septiembre del 2015, de

<http://tesis.ipn.mx:8080/xmlui/bitstream/handle/123456789/6296/SISTEMADQUISICION.pdf?sequence=1>

[9] National Instruments (2005, 16 de Mayo). *SCXI Chassis* Recuperado el 01 de Septiembre del 2015, de http://www.ni.com/pdf/products/us/4scxisc333_334_335_336.pdf

[10] National Instruments (2015, 01 de Septiembre). *SCXI Data Acquisition Systems. 16-Bit, 200 kS/s USB Data Acquisition Module* Recuperado el 01 de Septiembre del 2015, de http://www.ni.com/pdf/products/us/2004_3940_321_101_d.pdf

[11] National Instruments (2006, 19 de Octubre). *SCXI-1600 User Manual* Recuperado el 01 de Septiembre del 2015, de <http://phubner.eng.ua.edu/Files/NI%20SCXI-1600%20Manual.pdf>

[11] National Instruments (2007, 20 de Julio). *SCXI* Recuperado el 01 de Septiembre del 2015, de <http://www.ni.com/pdf/manuals/322853a.pdf>

[12] National Instruments (2009, 02 de Noviembre). *SCXI-1125 User Manual* Recuperado el 01 de Septiembre del 2015, de <http://www.ni.com/pdf/manuals/372425c.pdf>

[13] National Instruments (2007, 20 de Julio). *SCXI-1313A High Voltage Attenuator Terminal Block* Recuperado el 01 de Septiembre del 2015, de <http://www.ni.com/pdf/manuals/322853a.pdf>

[14] Guzmán , M. (2012, 08 de Mayo). Optimización de ancho de banda para sistemas GSM. Recuperado el 29 de Septiembre del 2015, de <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/734/A6.pdf?sequence=6>

[15] WEG (2012, 01 de Diciembre). *Manual de la Comunicación Modbus-RTU* Recuperado el 29 de Septiembre del 2015, de <http://ecatalog.weg.net/files/wegnet/WEG-srw01-manual-de-la-comunicacion-modbus-rtu-10000521680-4.0x-manual-espanol.pdf>

- [16] National Instruments (2014, 16 de Octubre). *The Modbus Protocol In-Depth - National Instruments* Recuperado el 29 de Septiembre del 2015, de <http://www.ni.com/white-paper/52134/es/>
- [17] Torres salazar, R. (2006). *Modbus RTU. Implementación del protocolo en un microcontrolador.* Recuperado el 29 de Septiembre del 2015, de <http://repositorio.uis.edu.co/jspui/bitstream/123456789/3176/2/119444.pdf>
- [18] MODICON, Inc., Industrial Automation Systems (s. f.). *Modicon Modbus Protocol* Recuperado el 29 de Septiembre del 2015, de http://modbus.org/docs/PI_MBUS_300.pdf
- [19] Trejo, R. (1997). *SISTEMA DE ADQUISICIÓN DE DATOS PARA INGENIERÍA BIOMEDICA* Recuperado el 29 de Septiembre del 2015, de <http://www.etitudela.com/entrenadorcomunicaciones/downloads/labviewmanual.pdf>
- [20] Pineda, A. (2011). *Instrumentación virtual. Fundamentos de programación gráfica con labview.* Recuperado el 29 de Septiembre del 2015, de https://www.editorialdigitaltec.com/materialadicional/ID355_PinedaOlivares_Instrumentacionvirtual.cap1.pdf.
- [21] National Instruments (2014, 15 de Octubre). *Introduction to Modbus - National Instruments* Recuperado el 06 de Noviembre del 2015, de <http://www.ni.com/white-paper/7675/es/>.
- [22] ABACOM Technologies (2009, 01 de Noviembre). *WIZ-SML-IA Transceiver Modules* Recuperado el 06 de Noviembre del 2015, de http://www.abacom-tech.com/assets/data_sheets/WIZ-SML-IAuser.pdf
- [23] Rivas, B. (2012, 04 de Mayo). *Transmisión de datos por RF* Recuperado el 06 de Noviembre del 2015, de

<http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/701/A7.pdf?sequence=7>

[24] SUNROM technologies. (2014, 03 de Noviembre). RF Serial Data Link UART, 2.4 Ghz [1124] Sunrom Technologies Recuperado el 06 de Noviembre del 2015, de <http://www.sunrom.com/p/rf-serial-data-link-uart-2-4-ghz>

Apéndice A. Diagramas de Casos.

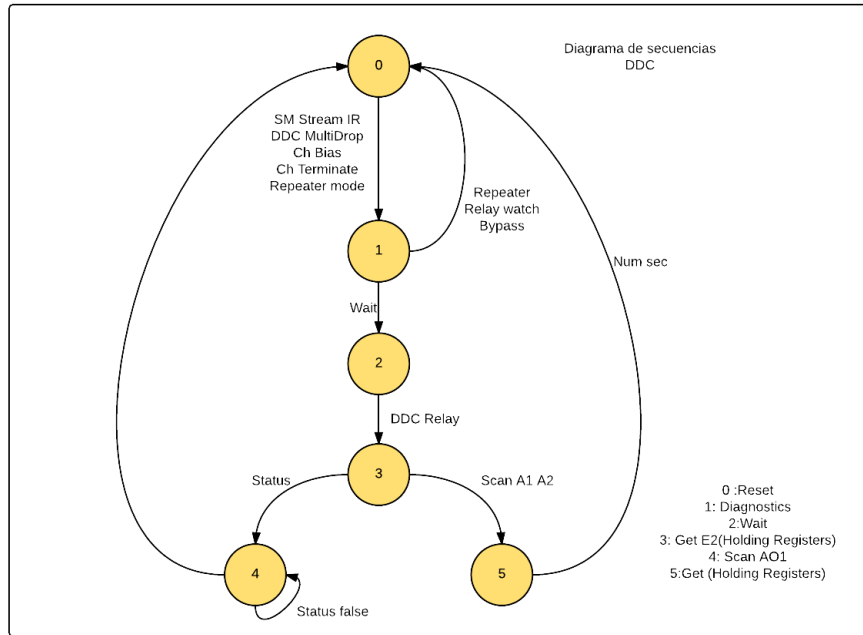


Figura A. 1. Diagrama general de secuencias DDC.

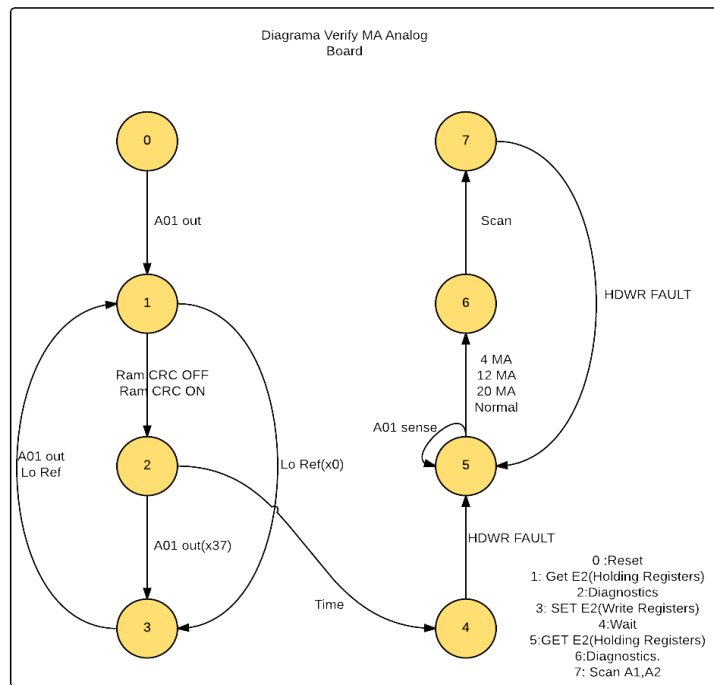


Figura A. 2. Diagrama Verify MA, Analog Board.

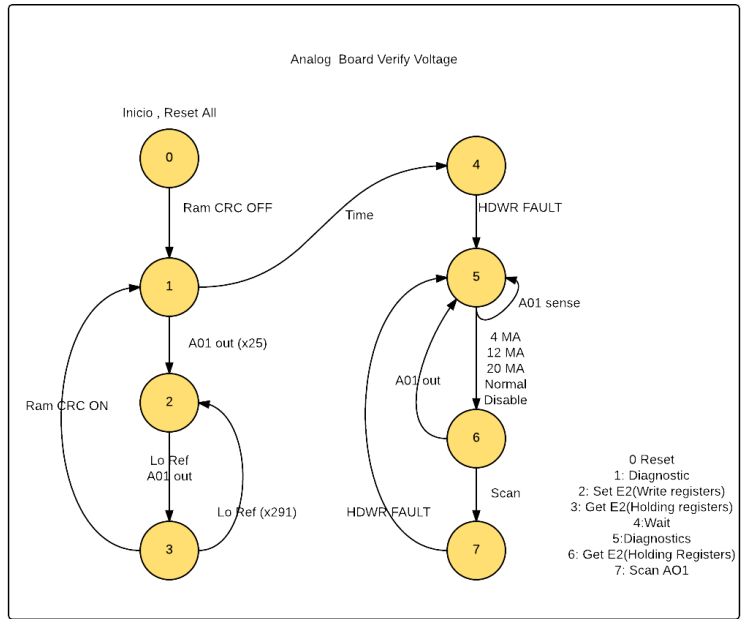


Figura A. 3. Verify Voltage, Analog Board, Secuencia general.

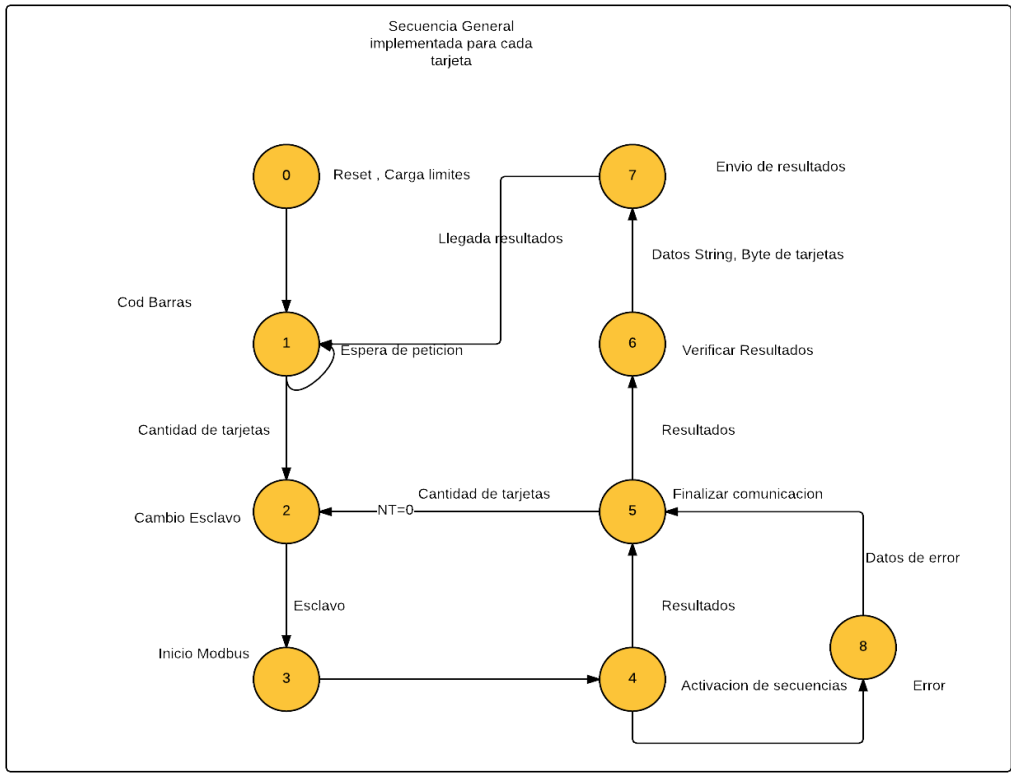
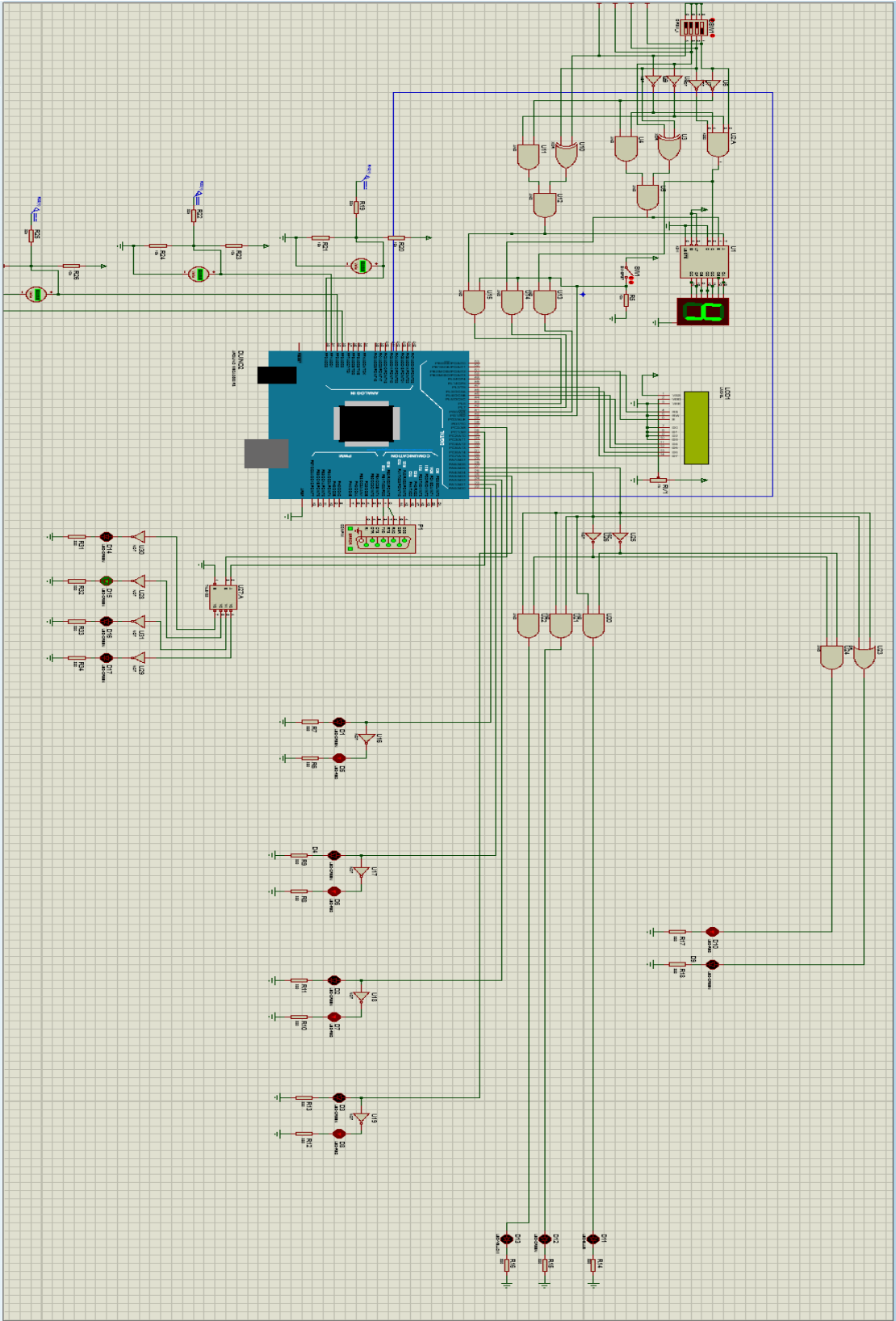


Figura A. 4. Secuencia General para cada tarjeta.

Apéndice B. Diagrama del Circuito.



Apéndice C. Programación en Labview.

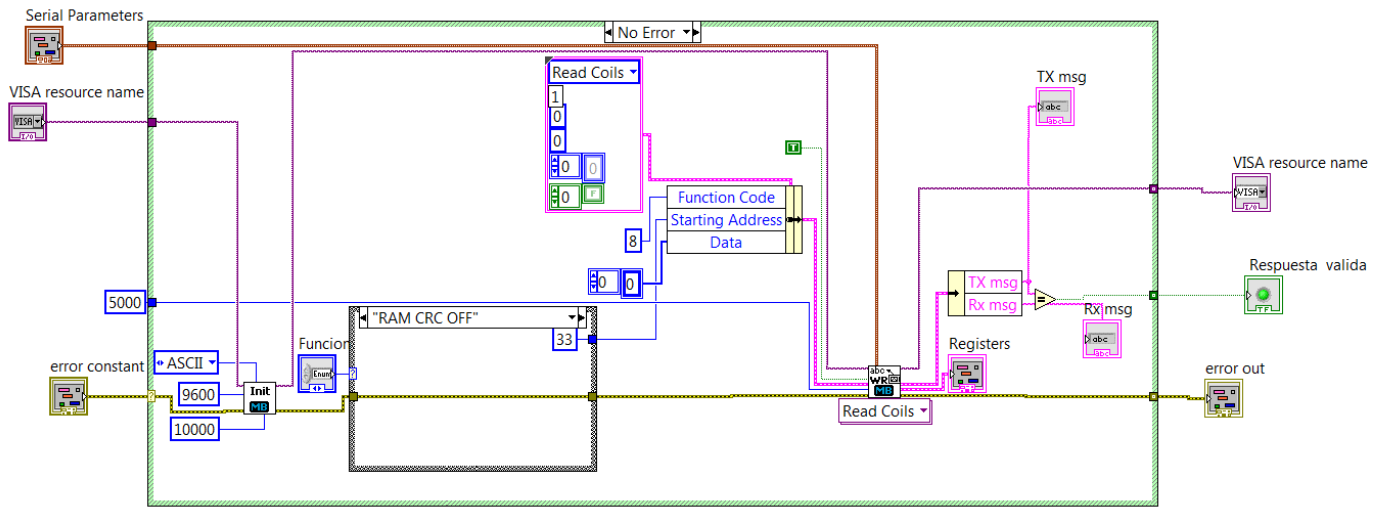


Figura C. 1 Diagnostics.

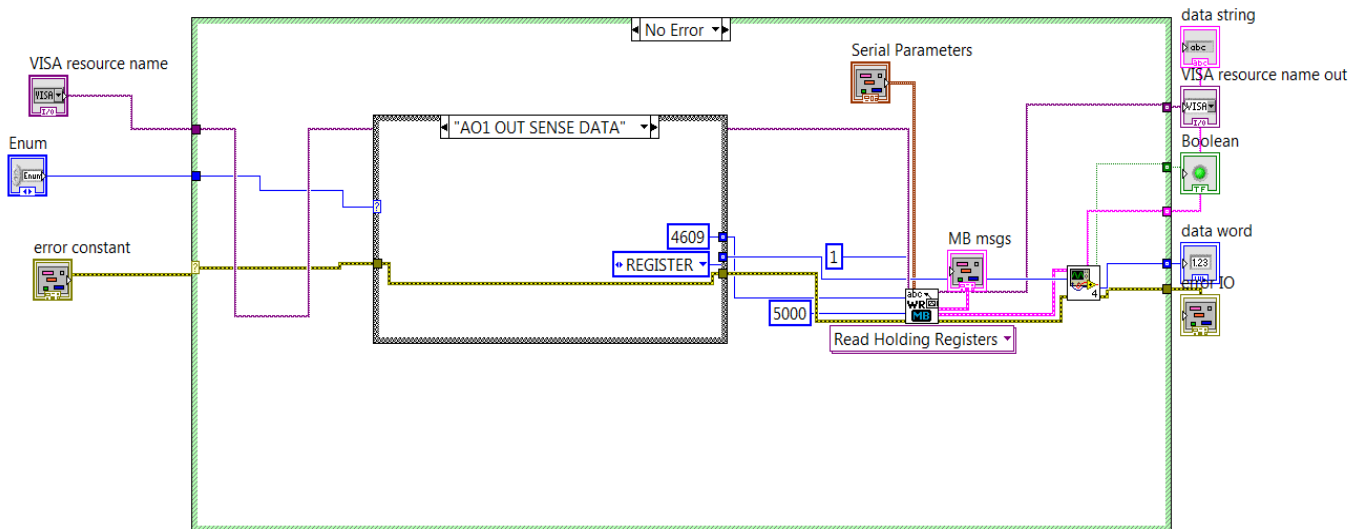


Figura C. 2 GET E2, Holding Registers.

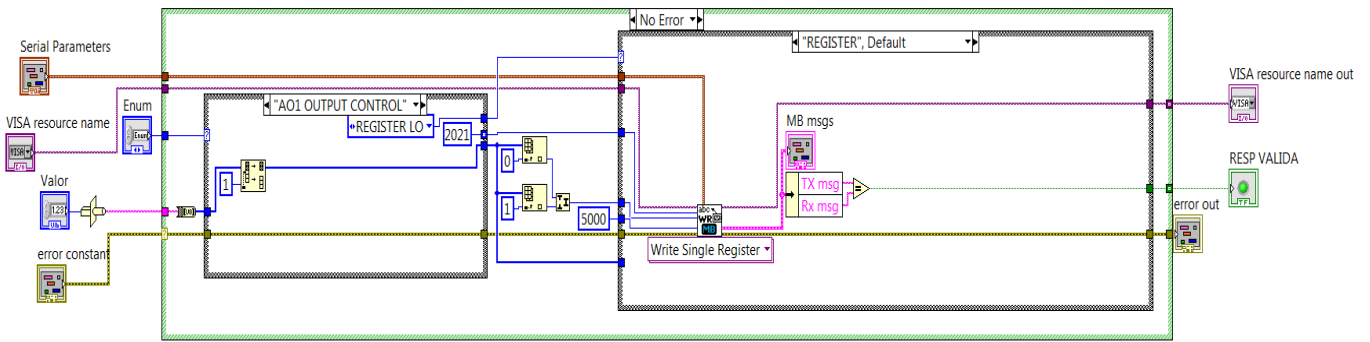


Figura C. 3 Set E2, Write Registers.

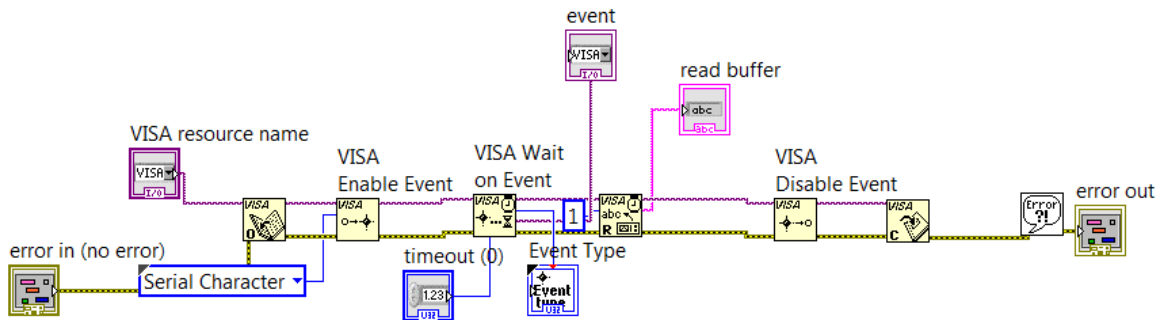


Figura C. 4 Espera de Petición 1.

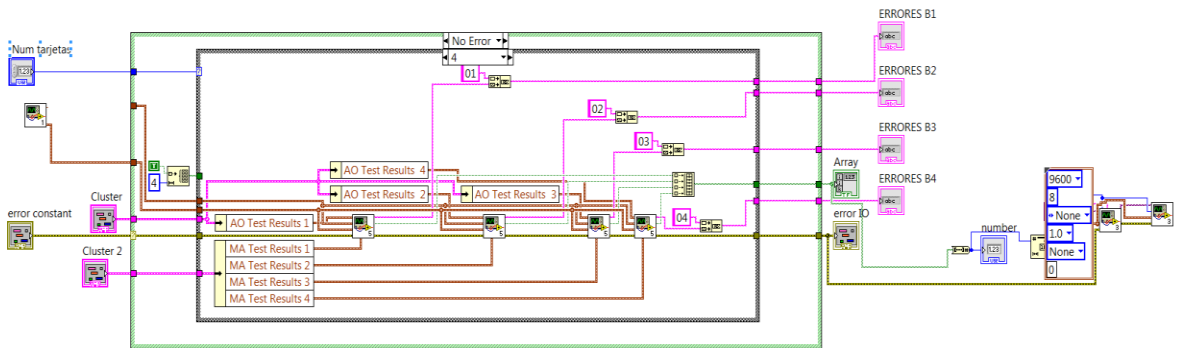


Figura C. 5. Verificar límites.

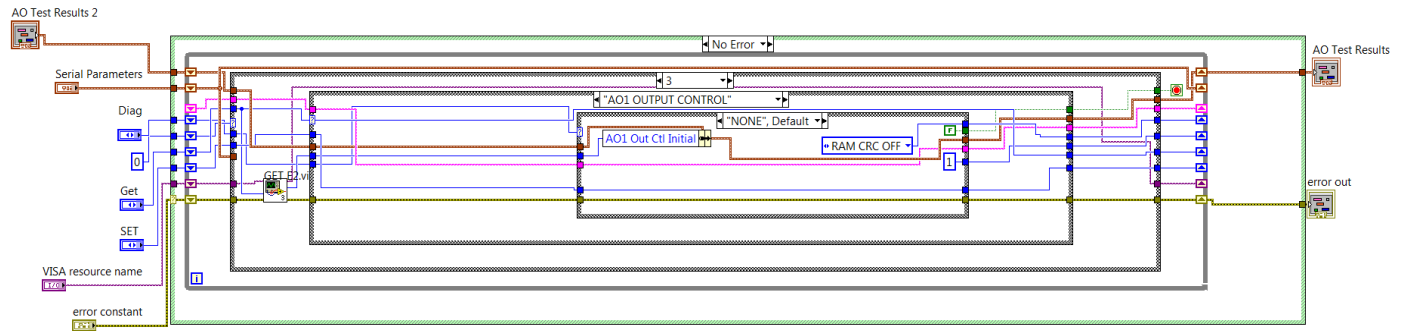


Figura C. 6. Verify Voltage secuencia general

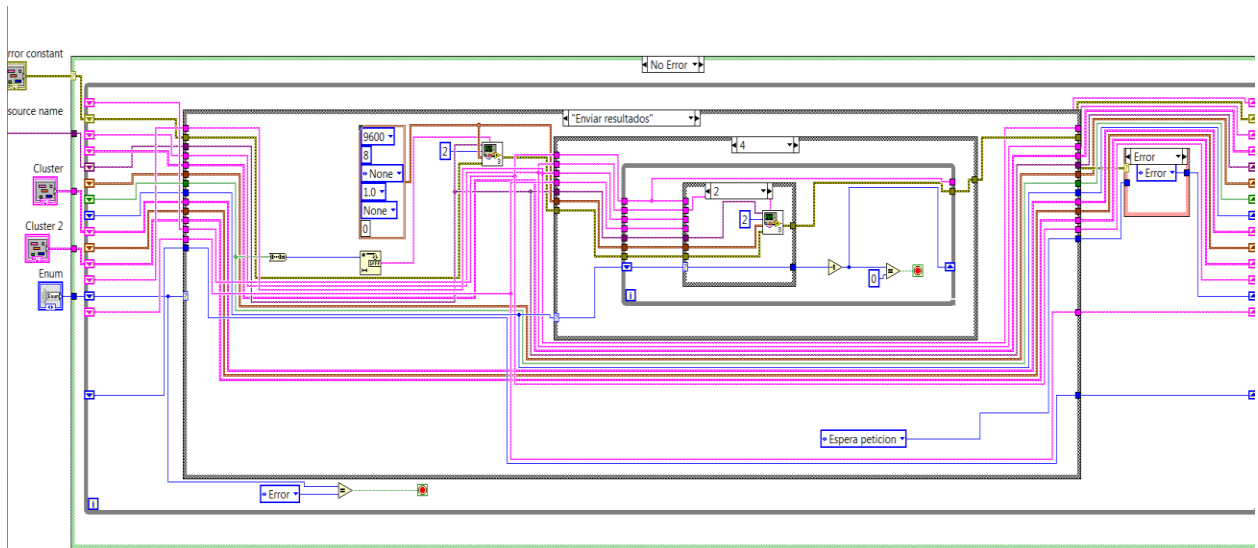


Figura C. 7. Programa general.