



**Universidad Autónoma de
Querétaro**

Facultad de Ingeniería

**Sistema de Activación Remota
de Alarmas**

TESIS

**Que para obtener el grado de
Ingeniero en Automatización**

**presenta
Fausto Arvizu de León**

Querétaro, Qro. Junio, 2016



Universidad Autónoma de Querétaro
Facultad de Ingeniería
Ingeniería en Automatización

Sistema de Activación Remota de Alarmas

TESIS

Que como parte de los requisitos para obtener el grado de
Ingeniero en Automatización

Presenta:

Fausto Arvizu de León

Dirigido por:

M. en C. Miguel Angel Martínez Prado

SINODALES

M. en C. Miguel Angel Martínez Prado
Presidente

Firma

Dr. Juvenal Rodríguez Reséndiz
Secretario

Firma

Dr. Roberto Augusto Gómez Loenzo
Vocal

Firma

Dr. Gonzalo Macias Bobadilla
Suplente

Firma

M. en C. Diana Carolina Toledo Pérez
Suplente

Firma

Dr. Aurelio Domínguez González
Director de la Facultad de Ingeniería

Dr. Juvenal Rodríguez Reséndiz
Coordinador de Ingeniería en Automatización.

Centro Universitario
Querétaro, Qro.
Junio de 2016
México

Resumen

La tendencia actual en el campo de la automatización es el uso de redes para la centralización de información y descentralización del manejo de hardware, por ello se han creado diversos protocolos de comunicación estandarizados para cumplir con diversos propósitos, la creación de dichos protocolos ha corrido a cargo de las empresas dedicadas del desarrollo de equipo de automatización.

Para el desarrollo del presente trabajo se ha tomado en cuenta esta necesidad y llevado al campo de la investigación para diseñar, desarrollar y comprobar la fiabilidad de la implementación de un sistema capaz de contener las características de los principales protocolos de comunicación industrial, que tiene como objetivo principal que el desarrollo de una red de arquitectura abierta dote a la industria del crecimiento de un sistema centralizado de alarmas, implementando un protocolo de comunicación, que sea escalable, flexible y de fácil adaptabilidad a sus necesidades.

(Palabras clave: Red, centralizar, arquitectura abierta, sistema escalable.)

Summary

The current trend in the automation field is the use of networks for information centralization and management hardware decentralization , therefore have created various standardized communication protocols to meet various purposes , the creation of these protocols has been carried out charge of development companies dedicated automation equipment .

For the development of this work is taken into account this need and taken to the field of research to design, develop and test the reliability of the implementation of a system capable of containing the characteristics of the main industrial communication protocols , whose main objective the development of a network of open architecture dowry industry growth of a centralized alarm system , implementing a communication protocol that is scalable , flexible and easy adaptability to your needs.

(Keywords: Network,Centralize, Open Architecture, Growth System)

Prefacio

El objeto de esta obra científica, es el de aportar un grano de arena a la comunidad académica y público en general. Dejando una evidencia del método científico aplicado a la rama de la tecnología automática. Esta obra guiará al lector por un camino de aprendizaje continuo y le brindará la oportunidad de reproducir el experimento que lleva como propósito este libro; y además, se deja la ventana abierta a la realización de una mejora al experimento. Todo esto, con el objetivo de brindar un beneficio a la sociedad en general.

La tecnología es un concepto que está en constante movimiento. El ser humano por naturaleza, busca satisfacer el hambre del aprendizaje, conocer más allá de sus alcances y explotar su capacidad creativa en obras que incrementarán la calidad de vida de sus semejantes. Debido a ello, el interés de adentrarme al mundo tecnológico es muy grande. Fascinado por las máquinas trabajando en total sincronía con un propósito en común, comienza la necesidad de satisfacer mi necesidad de aprender todo aquello que hay detrás de los sistemas automáticos. Esto me guía a comenzar mi formación universitaria, y una vez alcanzados todos los conocimientos necesarios para entender y ser capaz de analizar todos estos sistemas automáticos; me doy a la tarea de crear una innovación tecnológica. Además, esta obra deja como muestra mi crecimiento académico a lo largo de mi formación como alumno universitario. Queda de más agradecer a todos los profesores que dieron su tiempo para realizar la gran labor de la enseñanza, compartir sus experiencias y todos sus conocimientos y a los compañeros que me acompañaron en este largo y difícil camino, apoyándonos unos a otros y aprendiendo de cada uno de nosotros.

Agradecimientos

*A mi profesor, mentor y amigo,
Pero principalmente a mi madre que con su esfuerzo, sacrificio
y dedicación hizo posible este logro.*

Índice general

Resumen	IV
Summary	V
Prefacio	VI
Agradecimientos	VII
Indice de Figuras	X
Indice de Cuadros	XII
1. Introducción	1
1.1. Descripción del Problema	1
1.2. Justificación	2
1.3. Hipótesis	4
1.4. Objetivos	4
1.4.1. Objetivo General	4
1.4.2. Objetivos particulares	5
2. Fundamentación teórica	6
2.1. Redes Industriales de Comunicación	6
2.1.1. Sistemas de transporte de señal	8
Sistemas de transmisión de señal	9
Modos de transmisión de señal	12
Conceptos básicos de redes	13
Protocolos de comunicación	14
Topología de una red	15
Formas de comunicación	18

<i>Modos de diálogo</i>	19
2.1.2. Buses de campo	20
Buses de alta velocidad y baja funcionalidad.	21
Buses de alta velocidad y funcionalidad media.	21
Buses de altas prestaciones	21
3. Metodología	23
3.1. Estructura del Sistema	24
3.2. Comunicación	25
3.2.1. Comunicación PC - Maestro	25
3.2.2. Comunicación Maestro - Esclavo - Esclavo	32
3.3. Diseño de Hardware	33
3.3.1. Esclavo	34
3.3.2. Maestro	40
4. Resultados	45
4.0.3. Verificación de funcionalidad	49
4.0.4. Verificación de integridad de datos.	50
5. Conclusiones	55
Bibliografía	57
A. Código de firmware de microcontrolador	60
B. Herramientas para fabricación de PCB	70
C. Diagramas esquemáticos	75
D. Código de interfaz de experimentación	80

Índice de figuras

2.1. Elementos de bucle de corriente.	11
2.2. Codificación Manchester.	14
2.3. Componentes de un enlace de Datos.	15
2.4. Topología tipo estrella.	16
2.5. Topología tipo anillo.	17
2.6. Topología tipo bus.	17
2.7. Topología tipo árbol.	18
3.1. Diseño conceptual de sistema	24
3.2. Trama de comunicacion PC-Maestro	26
3.3. Cadena enviada para la función configurar sistema.	29
3.4. Cadenas enviadas para función cantidad componentes.	29
3.5. cadena enviada para función modificar una variable	30
3.6. Función “Consultar variable”.	30
3.7. Función modificar las variables	31
3.8. Función Consultar variable	32
3.9. Diagrama de Bloques TPIC6B595.[14, pág.11]	35
3.10. Diagrama de bloques comunciación en cascada de TPIC6B595.	37
3.11. Fragmento de Diagrama esquemático componente esclavo.	38
3.12. Circuito PCB del componente esclavo	40
3.13. Circuito esquemático de la tarjeta maestra.	41
3.14. Diagrama de flujo de <i>firmware</i>	42
3.15. Diseño PCB de componentes maestro.	44
4.1. Prototipo componente maestro	46
4.2. Prototipo componente esclavo	47
4.3. Interfaz de Prueba	48
4.4. Interfaz de Prueba	49

ÍNDICE DE FIGURAS

XI

4.5. Evidencias de funcionamiento del primer experimento	50
4.6. Señales de salida del componente maestro	52
4.7. Señales de entrada del ultimo componente esclavo	54
B.1. Diseño PCB de componente maestro	71
B.2. Positivo para aplicacion de pintura antisoldante (maestro	72
B.3. Diseño PCB de componente esclavo	73
B.4. Positivo para aplicacion de pintura antisoldante (esclavo)	74
C.1. Diagrama esquemático de procesamiento y transmisión periférica de componente maestro	76
C.2. Diagrama esquemático de potencia componente esclavo	77
C.3. Diagrama esquemático de transmisión periférica esclavo	78
C.4. Diagrama esquemático de potencia componente esclavo	79

Índice de cuadros

1.1. Estratificación de micro, pequeña y mediana empresa.	2
2.1. Resumen de niveles estándares de comunicación.	11
3.1. Tabla de Funciones.	28
4.1. Pruebas de integridad y eficiencia	53

Capítulo 1

Introducción

1.1. Descripción del Problema

En el sector industrial debido a su elevado grado de siniestralidad, control de calidad y alto grado de pérdida monetaria por fallas en la producción; es necesario la introducción de sistemas informativos y de acción inmediata, los cuales se encargan de informar el estado del proceso; así como de actuar oportunamente en caso de algún siniestro en la línea de producción para salvaguardar tanto al operador como el equipo.

Actualmente en las líneas industriales de producción se logra observar gran cantidad de torretas industriales en toda su extensión, sin embargo la activación de dichas alarmas informativas son activadas localmente y en la mayoría de los casos de forma manual o por medio de un controlador lógico programable (PLC por sus siglas en inglés). También existe una tendencia enfocada a contar con sistemas centralizados y enlazados entre sí, además de tener la posibilidad de adquirir información en cualquier momento y sin la necesidad de moverse del lugar de

trabajo. Es por esto que es necesario, no solamente contar con la información de variables del proceso, historial de producción, SCRAP, entre otras; sino que también es necesario conocer el historial de activación de alarmas, estatus del equipo, parar producción y/o activar una alarma de forma remota.

1.2. Justificación

Según la LEY PARA EL DESARROLLO DE LA COMPETITIVIDAD DE LA MICRO, PEQUEÑA Y MEDIANA EMPRESA (MiPyME) publicada en el año 2002, en su artículo 3° fracción III; una MiPyME debe cumplir diversos parámetros, los cuales se pueden observar dentro de la tabla 1.1 para su clasificación según su número de trabajadores y ventas anuales[11].

Tamaño	Sector	Trabajadores	Ventas por año(mdp)	Tope Max. Combinado*
Micro	Todas	(0 - 10]	(\$0 - \$4]	4.6
	Comercio	(11 - 30)		93
Pequeña	Industria y Servicio	(11 - 50)	(\$4 - \$100]	95
	Comercio	(31 - 100)		235
Mediana	Servicio	(51-100)	(\$100 - \$250]	250
	Industria	(51-250)		250

* Tope Max. Combinado = (Trabajadores)(10 %) + (Ventas Anuales)(90 %)

Tabla 1.1: Estratificación de micro, pequeña y mediana empresa.[13]

Los datos proporcionados por el Sistema de Información Empresarial Mexicano (SIEM) indican que simplemente en nuestro estado existen 471 empresas

catalogadas como empresas MIPYMES en el Sector Industrial (con un número máximo de 250 empleados) registradas en dicho sistema, sin ser la totalidad en nuestra entidad [13].

Entendiéndose que para las empresas MIPYMES, al contar con bajos recursos para adquirir personal especializado y equipo de punta para la producción, les es prácticamente imposible presentar una alta competitividad en los grandes mercados nacionales e internacionales; sin embargo de acuerdo con los datos del último censo de economía realizado por el INEGI y la Secretaría de Economía de nuestro país, las PYMES son un factor esencial en el crecimiento económico y en generación de nuevos empleos formales.

Gracias a datos arrojados por el censo de economía, las PYMES constituyen el 99.8 % de las unidades económicas del país y generan el 52 % del Producto Interno Bruto (PIB) además de representar el 71.4 % de la generación de empleos, sobrepasando a las empresas de mayor envergadura [12].

Conociendo la importancia de las PYMES en el desarrollo económico del país es necesario dotarlas de herramientas flexibles, las cuales puedan ser adaptadas a las necesidades de crecimiento de cada una de ellas.

La creación de un sistema centralizado de alarmas industriales el cual se pueda adaptar al crecimiento de forma continua, ayuda al fortalecimiento del desarrollo empresarial de la pequeña y mediana industria debido a la automatización y centralización de la información dentro de su planta de producción: ayudando a la reducción de tiempos en la ejecución de órdenes de trabajo, difusión de información importante en las líneas de producción, así como de la digitalización y

centralización de información de paros de producción.

No se debe dejar de lado la sencillez con que se adapta al crecimiento del sistema, el cual dota al propietario de una fácil extensión de su red, además de una fácil implementación dentro de sus sistemas informáticos sin la necesidad del uso de protocolos de uso comercial, los cuales no son de código abierto y no se puede hacer modificaciones de acuerdo con las necesidades del usuario.

1.3. Hipótesis

Es factible la fabricación de un sistema cableado centralizado de activación remota el cual sea de fácil expansión para un número variable de elementos, siendo comunicados entre sí de forma síncrona, una longitud no menor de 10m entre cada elemento, una velocidad de transmisión de 100kbps y con un porcentaje de error del 5 % con base a la integridad de la información enviada.

1.4. Objetivos

1.4.1. Objetivo General

Desarrollar un sistema centralizado para la activación de alarmas industriales flexible de arquitectura abierta y de fácil ampliación a un número de elementos variables comunicados de forma síncrona localizados a una distancia no menor de 10 m entre cada elemento.

1.4.2. Objetivos particulares

- Diseñar el hardware y un protocolo de comunicación, de arquitectura abierta y de fácil comprensión y adaptabilidad al crecimiento, necesarios para la implementación de un sistema de activación remota de alarmas industriales.
- Fabricar un prototipo funcional capaz de controlar al menos 24 salidas digitales, siendo factible su posible ampliación a n-ésimas salidas digitales.
- Comprobar el rendimiento del prototipo, mediante pruebas de funcionamiento continuo a una velocidad de transmisión de 100 kbps y una longitud de 10 m entre cada elemento.

Capítulo 2

Fundamentación teórica

2.1. Redes Industriales de Comunicación

Según Galloway y Hanvke [1] “ Industrial networks are employed in many industrial domains including manufacturing, electricity generation, food and beverage processing, transportation, water distribution, waste water disposal and chemical refinement including oil and gas. In almost every situation that requires machinery to be monitored and controlled an industrial control network will be installed in some form ”. Conociendo esto, podemos darle la importancia pertinente en el área de la automatización al uso de las redes industriales; por lo cual es vital conocer el manejo y la construcción de las diversas redes industriales así como su estructura. Hay diversos motivos por los cuales es importante incorporar una red industrial de comunicación dentro de la empresa entre los cuales se pueden mencionar algunos, tales como:

- La reducción de costes de producción.
- El mejoramiento de la calidad.
- La mejora de la productividad.
- La reducción de almacenaje.
- La mejora de la efectividad de procesos.
- La reducción de costos de mantenimiento.

Para alcanzar dichos objetivos, es de suma importancia que los sistemas de comunicación permitan que la planta de producción y la gestión e ingeniería de la empresa se enlacen, además de lograr la integración de bases de datos de la empresa [2]. El control de un proceso se puede clasificar dependiendo de la complejidad del sistema o de los componentes que intervienen en la red de comunicación; en esta clasificación se encuentran los sistemas centralizados y los sistemas distribuidos, que como lo indican sus nombres, existe una sola diferencia entre ellos, la cual radica en donde y por quien se lleva a cabo el control, es decir, en los sistemas centralizados se realiza el control por un solo sistema, y en los sistemas distribuidos el control se realiza a través de diferentes sistemas conectados en una red. Para seleccionar que tipo de control se debía implementar en el sistema, es importante conocer las capacidades, alcances y/o limitaciones de la aplicación, así como de su complejidad. En los sistemas centralizados es indispensable el uso de *buses* de comunicación, los cuales permitirán la comunicación entre controlado-

res secundarios y primarios. Para ello existen en la actualidad diversos métodos de comunicación que son usados ampliamente dentro del ramo.

2.1.1. Sistemas de transporte de señal

Es importante conocer que para lograr un intercambio de información entre dos equipos o más es necesario un medio de transporte para dicha información donde podremos encontrar diversas formas y medios para conseguir lo anteriormente mencionado[3]. Entre los principales medios de transporte de información podemos encontrar:

Cable eléctrico, fácil de comprender y uno de los métodos más usado, el cual consta de un hilo metálico aislado; este se puede dividir en dos grandes tipos:

- Par.
- Coaxial.

Fibra óptica, este consiste en un núcleo de material transparente, típicamente plástico, el cual guía señales luminosas por su interior. A pesar de ser más caro que el cable ha sido el sustituto ideal en ambientes con interferencias eléctricas, pues es completamente inmune a estas.

Radiofrecuencia, basada en señales de radio por debajo de 1GHz, permite comunicar de forma ininterrumpida a ubicaciones distantes entre varios kilómetros.

Sistemas de transmisión de señal

Ya que se ha conocido el medio de transporte de señales sería importante conocer qué y cómo se deberían enviar dichas señales de información, es decir, una vez que se define el medio, nos preguntamos ¿qué se enviara por este medio?, esto se puede realizar de dos formas principales: a partir de tensiones o de un bucle de corriente.

Niveles de voltaje. La recomendación en entornos industriales es que las conexiones se realicen mediante interfaces serie que están normalizadas por la Asociación de Industrias Electrónicas de los Estados Unidos (EIA). Estos estándares solo determinan las características del soporte y características eléctricas de la señal[3].

Entre los estándares recomendados (Recommended Standard, RS) más conocidos se encuentran TTL, RS-232, RS-422 A, y la misma evolución de este el ultimo el RS-485.

RS-232. Estándar de comunicación punto a punto en el cual los niveles binarios de la señal son indicados por niveles de tensión positivo y negativo ($\pm 10V$) con respecto al punto potencial común. Debido a su estructura en la cual no dispone de un estado de alta impedancia, provoca que no sea adecuado para la conexión en paralelo [5].

RS-422. A diferencia del estándar RS-232 el 422 se basa en comunicación por medio de dos hilos sin tener un punto de masa común, es decir, la transmisión de unos y ceros binarios es enviada de forma diferencial (balanceada); gracias a esto la información tiene una gran inmunidad al ruido, además el estándar, al contemplar un estado de alta impedancia dentro de su estructura permite la conexión en paralelo entre transmisión y varios receptores a diferencia del estándar 232. La norma RS-422 especifica una velocidad máxima de 10 Mbps y longitudes de hasta 1200 m. Sin embargo esto no significa que la transmisión a 1200 m se llevara a cabo a la más alta velocidad. Es decir, a una velocidad de 10 Mbps la distancia aproximada será 15 m, y a una distancia de 1200 m la velocidad máxima de transmisión será de 90kbps.

RS-485. Al ser una evolución del estándar 422, amplió la capacidad de elementos conectados en el tramo de 10 a 32 elementos.

TTL. Basada en la transmisión digital de niveles eléctricos de 5 V (TTL), con esta tecnología es posible la conexión en paralelo de nodos a un mismo cable, ya que permite un estado de alta impedancia en la conexión del dispositivo[6].

En la tabla 2.1 se pueden observar las diversas características físicas y de trans-

misión de los estándares, observándose claramente sus diferencias.

Estándar	Señal	Elementos	Distancia(m)	Velocidad(Kbit/s)
RS-232	Asimétrica	1	15	20
RS-422	Simétrica	10	1200	10000
RS-485	Simétrica	32	50	10000

Tabla 2.1: Resumen de niveles estándares de comunicación.[3, pág.260]

Bucle de Corriente. En esta tecnología, los distintos niveles son indicados mediante la corriente, es decir, los datos son enviados e interpretados por medio de la corriente en una línea de transmisión (4 mA a 20 mA).

Algunas de las principales ventajas de esta tecnología son la robustez frente a interferencias eléctricas, a diferencia de los sistemas basados en niveles de tensión; el envío de señales a larga distancia y el autodiagnóstico del valor de offset debido que, al no presentarse un nivel de 4 mA el bucle podría encontrarse abierto.

En la figura 2.1 se muestran los diferentes elementos utilizados para realizar el bucle de corriente de 4-20 mA, que puede contener por lo menos 4 elementos principales.

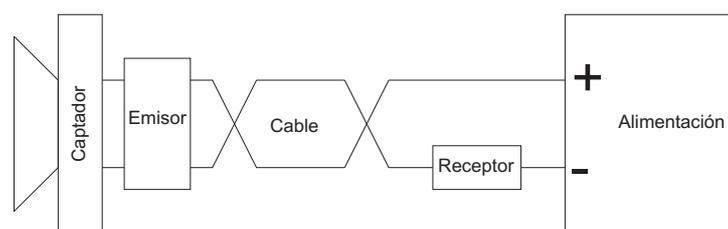


Figura 2.1: Elementos de bucle de corriente.[3, pág.260]

Modos de transmisión de señal

Existen diferentes métodos de transmisión de señales eléctricas, las cuales contienen la información que se requiere transmitir; se comportan como circuitos eléctricos comunes, ya que estos modos de transmisión de datos pueden ser en el paralelo y serie.

Paralelo. Dentro de las ventajas más destacadas del modo de transmisión en paralelo, se tiene la capacidad de realizar la transmisión de información a una gran velocidad, a pesar de ello uno, de los inconvenientes derivados son la distancia máxima de transmisión tolerada.

Serie. Siempre que un sistema clásico de transmisión de señal basado en niveles de tensión el sistema de transmisión genera cambios de niveles de tensión dentro de dos valores preestablecidos, estos deben poder ser interpretados por el receptor para con esto poder ser traducidos a bits. Es por lo anterior que el método exige que tanto el transmisor como el receptor estén sincronizados entre sí.

Existen dos métodos de sincronización para lograr la sincronización de receptor y emisor:

- *Asíncrono.* La sincronización entre emisor y receptor es gracias al trabajo con la misma velocidad y número de bits, la existencia de un bit al inicio del

mensaje que pondrá en aviso al receptor cuando comience el muestreo de la señal de presente en el medio, requiriendo de periodos de reloj constantes en el tiempo.

- *Síncrono.* Existe una señal adicional de reloj la cual indica al receptor los instantes de muestreo de la señal. Una de las principales ventajas se encuentra la no dependencia a la constancia del periodo de tiempo del reloj[3].

Conceptos básicos de redes

Para introducirse en el terreno de las redes computacionales o industriales se deben de conocer conceptos básicos para entender y/o implementar una red industrial.

Nodos y terminales. A todo aquel elemento que influye de manera activa dentro del proceso de transferencia de la información dentro de una red se le nombra nodo.

Se conoce como estación terminal (ETD) a todos aquellos dispositivos que estén conectados a cualquiera de los nodos dentro de la red.

Velocidad de movimiento de la información. La cantidad de información que puede ser transmitida en cada unidad de tiempo es representada por medio de la velocidad de movimiento de la información, es decir, al promedio de bits, caracteres

o bloques de información despreciando los bits de sincronismo, bits de start, stop, etc. transmitida por unidad de tiempo.

Codificación de señales. A grandes rasgos la codificación de los sistemas de comunicación es reducida a representación de la información en pulsos eléctricos rectangulares, tomando en cuenta un código determinado[7].

Dentro de los modelos de codificación síncrona con mayor extensión se encuentra se encuentra la codificación ASCII, la cual cuenta con bits de inicio, parada y paridad. En la figura 2.2 es fácil poder observar diferencias entre dos tipos de codificaciones, la codificación directa y la codificación tipo Manchester, en la cual cada bit es subdividido en dos intervalos, consiguiendo con esto determinar el nivel lógico del bit mediante el flanco de subida del primer y segundo subintervalo; logrando con esto la sincronización entre el transmisor y receptor [6].

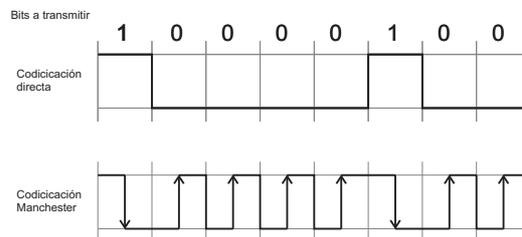


Figura 2.2: Codificación Manchester.[3, pág.263]

Protocolos de comunicación

Una vez conocido el soporte físico y las características de las señales que portaran por el cual será transmitida la información, se debe determinar la forma

en la cual será transmitida dicha información.

El protocolo de comunicación engloba las reglas y las convenciones, las cuales tendrán que cumplir los equipos que deseen intercambiar información para que esto se pueda llevar a cabo. Los enlaces de comunicación deben estructurarse con elementos definidos como lo son el canal, el Equipo de Terminal de Datos (ETD) y el Equipo de Comunicación de Datos (ECD). En la figura 2.3 se puede observar la estructura y las ubicaciones de dichos elementos, los cuales cumplen una tarea específica cada uno de ellos [6].

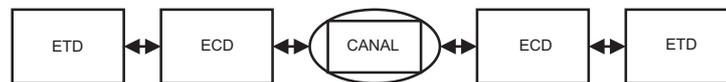


Figura 2.3: Componentes de un enlace de Datos.[3, pág.263]

El objetivo de cualquier protocolo es el lograr y mantener el intercambio de información entre dos ETD, permitiendo que este intercambio de información pueda fluir sin fallos.

La estandarización es un punto de conflicto técnico y comercial entre los fabricantes y desarrolladores, debido a que investigan y desarrollan sus productos para cumplir y cubrir necesidades específicas, por lo cual busca la estandarización de sus desarrollos. Es gracias a esto que existen diversos protocolos y equipo en el mercado.

Topología de una red

Es llamada topología a aquella forma en la cual se encuentran conectadas estaciones de trabajo.

La topología tiene un lugar importante en el diseño de la red debido a que determina en qué lugar deben colocarse estaciones maestras, además de dar una idea de cómo será distribuida la red y además de las longitudes del medio de transmisión. Entre las topologías básicas se pueden encontrar diversos tipos, los cuales variaran tanto en conexión física, interconexión entre sí y algunos otros aspectos.

Tipo estrella. En la topología tipo estrella los equipos de comunicación han sido colocados por una línea punto a punto con el controlador central, la cantidad de las estaciones está limitada por el número de conexiones punto a punto con las que cuente el nodo central, cabe mencionar que la comunicación entre las estaciones y el resto de la red depende del funcionamiento del nodo central y que si éste llegara a fallar el resto de las estaciones no tendría comunicación con los demás elementos de la red; sin embargo, si una de las terminales no funciona, esto no afectará la comunicación de las estaciones sobrantes conectadas al nodo central. En la figura 2.4 se puede observar la estructura básica de la topología tipo estrella apreciando sus componentes como son el nodo central y las estaciones[7].

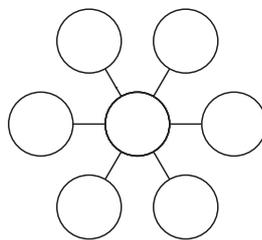


Figura 2.4: Topología tipo estrella. Elaboración propia.

Tipo Anillo. El medio de transmisión de la información forma un circuito cerrado, la información basada en la transmisión punto a punto con entre estaciones y el modo de transmisión se organiza por turnos mediante permisos de transmisión de una a otra estación. En la figura 2.5 se observa la conexión entre estaciones dentro de una red con topología tipo anillo, pudiéndose apreciar la estructura básica de este tipo de red [10].

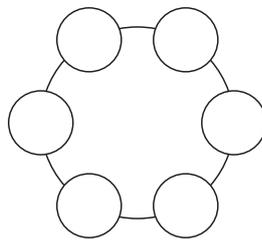


Figura 2.5: Topología tipo anillo. Elaboración propia.

Tipo Bus. En la figura 2.6, observaremos que la topología tipo bus consiste en un cable central, el cual es denominado troncal, del cual se realizan derivaciones (taps), por las cuales serán conectadas las estaciones (drops), las cuales son de pequeñas longitudes, este es el principio básico de topología tipo bus. Es considerado de fácil instalación. El envío de información se realiza de forma simultánea la cual es identificada por dirección y en la cual solo se enviará en el momento que se requiera a la estación o desde la estación requerida [8].

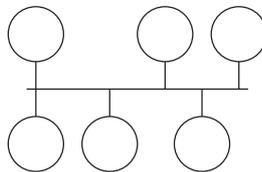


Figura 2.6: Topología tipo bus. Elaboración propia.

Tipo árbol. En la figura 2.7 es fácil apreciar la estructura básica de la topología tipo árbol o jerárquica la cual está conformada de forma jerárquica, donde existen subniveles que pueden ser conectados de formas diversas ya utilizando topologías como los son las de bus y/o anillo. Por lo cual se puede decir que la topología es una combinación de las topologías anteriormente descritas [10].

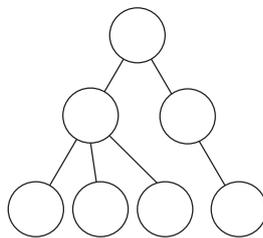


Figura 2.7: Topología tipo árbol.Elaboración propia.

Formas de comunicación

Existen diversas formas de comunicar los sistemas dentro de la red viéndolas desde el punto de vista de la frecuencia, en la cual ocurre el intercambio de información entre cada uno de los equipos, lo cual puede ser de dos maneras distintas:

- *Cíclica (periódica).* En el momento de transmitir información se busca que el tiempo en cual se realiza el intercambio de datos sea el más corto posible, es por ello que en el intercambio de pequeños paquetes de datos de prioridad mayor debe ser de forma cíclica, es decir periódicamente, lo cual se realiza de forma automática.

- *Acíclica (aperiódica)*. Al buscar el intercambio de grandes paquetes de información no crítica y de poca prioridad, se busca que el sistema no esté ocupado por el intercambio de este tipo de información, por lo cual la información de poca prioridad es enviada de tal forma que no interrumpa el envío de datos críticos hecho por la comunicación cíclica.

Modos de diálogo

Según el tipo de enlace físico por el cual se lleva a cabo la comunicación entre los equipos terminal de datos aparecen diferentes posibilidades de comunicación de diálogo entre ellos:

- *Simplex*. El modo en cual solo existe una comunicación entre un emisor y un receptor de datos en un solo sentido, es decir, del primero al segundo.
- *Half Dúplex*. El diálogo entre emisor y transmisor es de forma bidireccional, es decir, en sentidos emisor-receptor y receptor-emisor. Sin embargo, este diálogo no es de forma simultánea, sino que debe de ser turnada.
- *Dúplex*. El modo de diálogo dúplex o también llamado full dúplex, al igual que el half dúplex existe un intercambio de información en los dos sentidos, pero con la diferencia de que en el full dúplex ocurre de manera simultánea al contar con dos canales de F[6].

2.1.2. Buses de campo

Los sistemas de cableado tradicionales como lo son el cableado clásico o mediante bases de pre-cableado, presentan gran cantidad de trabajo entre sensores y actuadores dentro del proceso, la cual crece directamente con la complejidad del sistema; por lo cual los extremos deben ser numerados así como haciendo necesario el uso de mangueras para evitar confusiones en el cableado. Aunado a esto, la gran cantidad de cables dentro del sistema pueden inducir ruidos, lo cual puede provocar fallos en el funcionamiento, siendo afectadas mayormente señales analógicas. Debido a esto, en las décadas de los 80's y 90's se desarrollaron métodos los cuales ayudaron a mejorar la comunicación entre sensores, actuadores y la unidad de control; basado en un cable de comunicación serial, los buses de campo lograrían un diálogo entre los elementos del sistema.

En grandes rasgos los buses de campo son un sistema de comunicación entre los elementos de control industrial y los dispositivos de campo. El uso de buses de campo llego a reducir costos de cableado, configuración y mantenimiento en aplicaciones con cableados tradicionales debidos, un ejemplo claro de las grandes ventajas que trae consigo el uso de buses de campo se puede observar en un proceso automatizado en el cual usen dispositivos inteligentes, en los cuales usando un método de cableado tradicional gastaría muchas entradas/salidas de los autómatas programables [4].

Los fabricantes de equipos de campo han desarrollado distintos buses de campo los cuales se han agrupado en grandes grupos de acuerdo con sus características y ventajas.

Buses de alta velocidad y baja funcionalidad.

Son diseñados para controlar e integrar dispositivos de funcionamiento simples como los sensores de fin de carrera, relevadores y actuadores simples. Todo esto funcionando en tiempo real dentro de una planta.

Algunos de los ejemplos más significativos de este tipo de buses son el CAN, SDS y el AS-I, este último desarrollado por Siemens para la integración de sensores y actuadores.

Buses de alta velocidad y funcionalidad media.

Basados en el envío de paquetes de datos de tamaño medio, dentro de los cuales es posible enviar información con aspectos de configuración, calibración y/o programación. Estos buses son capaces de controlar dispositivos de campo complejos de forma eficiente [9].

DeviceNet es uno de los buses de esta clase, el cual utiliza como base el bus CAN e incorpora una capa orientada a objetos para ampliar su funcionalidad es utilizado ampliamente.

Buses de altas prestaciones

Capaces de soportar comunicaciones de cualquiera de los niveles de la producción dentro de la pirámide automatización, presentan problemas debido a que son basados en buses de alta velocidad, y se deben sobrecargar para alcanzar las características funcionales y de seguridad que se les exigen en los demás nive-

les. Algunos de los buses que cumplen con estas características son el **Profibus**, **WorlFIP** y el **Fielbus Foundation**[9].

Capítulo 3

Metodología

Para ir por el camino correcto en el diseño de un sistema es importante tener cuantificados y cualificados sus alcances, por este motivo definiremos el sistema a diseñar.

Existe la necesidad de un sistema centralizado de alarmas el cual sea capaz de controlar dispositivos para la transmisión de información de forma sonora y visual a los operadores de líneas de producción de forma remota desde un ordenador central. Este sistema es capaz de controlar por lo menos de 2040 salidas digitales. Es importante mencionar que para validar su funcionamiento se realizaron pruebas con un número menor de salidas, la cual se mencionará según se vaya requiriendo dentro del desarrollo del diseño y en secciones posteriores de este documento. Es importante mencionar que uno de los objetivos de este trabajo es crear un sistema de fácil adaptación a las necesidades del usuario, es por ello, que el diseño se plantea con una construcción modular.

3.1. Estructura del Sistema

Basado en la definición de los alcances y estructura modular del diseño se plantearon dos componentes principales, los cuales se enlistan a continuación:

- *Maestro*. Es encargado de la comunicación con el ordenador central, transmisión de información a los dispositivos esclavos, por ello debe ser el primer eslabón dentro de la red.
- *Esclavo*. Dispositivos que recibe y ejecuta la información del dispositivo maestro, pueden encontrarse uno o más dentro de la estructura del sistema.

Estos dos componentes cuentan con 8 puertos de salidas digitales cada uno y deben estar comunicados entre sí por un medio de transmisión cableado a través de un bus de comunicación con una topología tipo anillo, una comunicación unidireccional y sin “FeedBack” (retroalimentación). La comunicación iniciará desde el dispositivo maestro y continuando hacia los dispositivos esclavos consiguientes a través de toda la red, gracias a esto se puede generar un diagrama de bloques de la estructura principal de la red lo cual se aprecia en la figura 3.1.

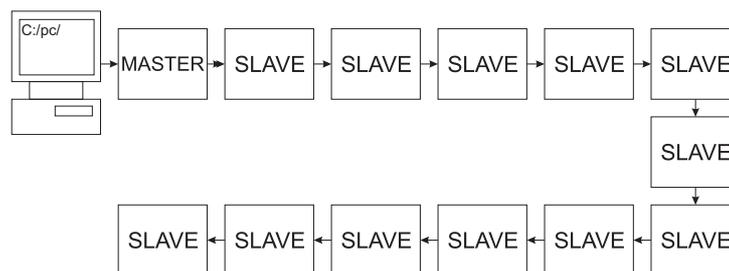


Figura 3.1: Diseño conceptual de sistema

3.2. Comunicación

La forma en la cual se efectúa la comunicación dentro de la red consta de dos fases, por un lado se encuentra la comunicación que existe entre el ordenador central y por el otro la comunicación entre maestro y esclavos.

3.2.1. Comunicación PC - Maestro

Al estar conectado con el componente maestro con el ordenador central es importante el uso de protocolos de comunicación actuales y compatibles con el equipo, es por eso que para la comunicación entre el ordenador y componente maestro se eligió el uso del protocolo USB en su modo CDC (Communication Devices Class), el cual simula un puerto COM dentro el ordenador para la comunicación por el estándar RS-232.

Como se puede constatar en la figura 3.1 al tener un gran número de elementos a controlar es importante poder asignar a cada uno de los componentes que tarea deben realizar, es por ello que el componente maestro es el encargado de recibir la información del ordenador central para asignar las diferentes tareas a los demás; éste debe de ser capaz de modificar el estatus de cada dispositivo controlado de forma individual, grupal o en la totalidad de elementos pertenecientes en la red, esto es posible encapsulado en una trama constituida por identificadores y funciones.

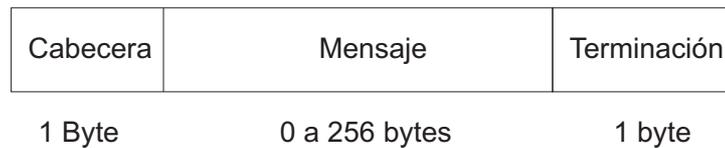


Figura 3.2: Trama de comunicacion PC-Maestro

En la figura 3.2 se plasma la estructura básica de trama de información enviada al componente maestro desde el ordenador central, se puede observar que es de una longitud variable de bytes; así mismo los componentes y sus atributos que la comprenden son enlistados a continuación:

- *Cabecera*. Existen diversas funciones las cuales pueden ser ejecutadas por los componentes del sistema. Estas pueden ser “configurar sistema”, “activar”, “desactivar” y “monitorear estado” de las salidas, las cuales están explícitas dentro de la cabecera.
- *Mensaje*. Es la información enviada desde el ordenador central, procesada y retransmitida a cada uno de los componentes de la red, ya sea el maestro o los esclavos. Dentro del mensaje debe ir explícito, dependiendo de la cabecera, diferentes parámetros para la ejecución de la función como lo son los “Índices” y “Variables”.

 - *Índice*. Los componentes dentro de la red son numerados dependiendo de su posición dentro de la red, empezando por el componente maestro, el cual es el número “1”, continuando con el componente esclavo inmediato siendo el número “2” y así sucesivamente, a este número se le conoce como índice.

- *Variable.* Son aquellos valores los cuales son aplicados a los elementos indicados por el índice, este valor es representado de forma binaria en las 8 salidas digitales de cada uno de los elementos.
- *Terminación.* Debido a que la longitud de la trama de información que es enviada al componente maestro es variable, la trama debe contar con un carácter el cual le indique al maestro que el envío de instrucciones ha concluido.

Como se mencionó dentro de la cabecera, están explícitas las funciones a realizar por cada uno de los componentes pertenecientes a la red, cada una de las funciones tiene un valor numérico, el cual es interpretado y procesado por el componente maestro para la ejecución en cada uno de los componentes. Pueden existir dos tipos de funciones, con y sin retorno, en la tabla 3.1 se han agrupado las funciones existentes así como sus características.

Tipo	No. (HEX)	Función	Parámetros	Retorno
Con Retorno	01	Configuración	#Componentes desados	#Componentes Configurados
Con Retorno	02	Consultar Variables		Valor de Variables

Continúa en la siguiente página

Tipo	No. (HEX)	Función	Parámetros	Retorno
Sin Retorno	03	Modificar una Variable	Índice, Valor de Variable desados	
Sin Retorno	04	Modificar las Variables	Valores Deseados	
Sin Retorno	05	Consultar una Variable	Índice	
Con Retorno	06	Cantidad de Componen- tes		#Componentes instalados

Tabla 3.1: Tabla de Funciones.

Configurar Sistema. Esta función es la encargada de efectuar el cambio del

número de componentes dentro de la red, el cual se conservará configurado aunque la alimentación sea retirada y sólo será modificado si se vuelve a ejecutar esta función. En la figura 3.3 se muestra la trama enviada del ordenador central al componente “Maestro”.

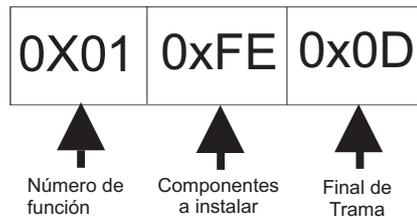


Figura 3.3: Cadena enviada para la función configurar sistema.

Cantidad de Componentes. El único objetivo de esta función es el obtener el número de componentes instalados en la red, es por ello que el componente maestro responde al ordenador central con una trama de 2 solo byte donde está indicado dicho valor antepuesto al número 1 que indica la respuesta. Para ejecutar esta función simplemente es requerido enviar al componente maestro la trama básica donde indique el número de función y el byte de terminación de trama, lo anterior mencionado es ilustrado en las figuras 3.4a y 3.4b.

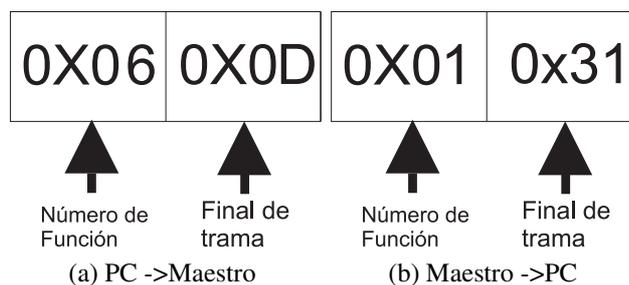


Figura 3.4: Cadenas enviadas para función cantidad componentes.

Modificar una Variable. Función sin retorno que para ser ejecutada deben ser enviados el índice y el valor de la variable; este valor será mostrado en las salidas digitales del componente correspondiente. La trama enviada al componente maestro se ilustra en la figura 3.5.

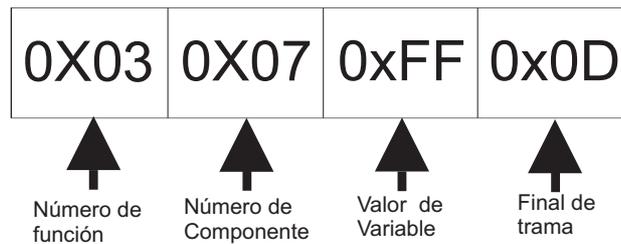


Figura 3.5: cadena enviada para función modificar una variable

Consultar Variable. Utilizada para conocer el valor de una variable en específico. Esta función con tan solo enviar el número del componente después del código de función retornará al ordenador central el valor de la variables del componente indicado. Para ilustrar el funcionamiento de la función se pueden ver las figuras 3.6a y 3.6b, que corresponden a la trama enviada al maestro y la respuesta de éste respectivamente.

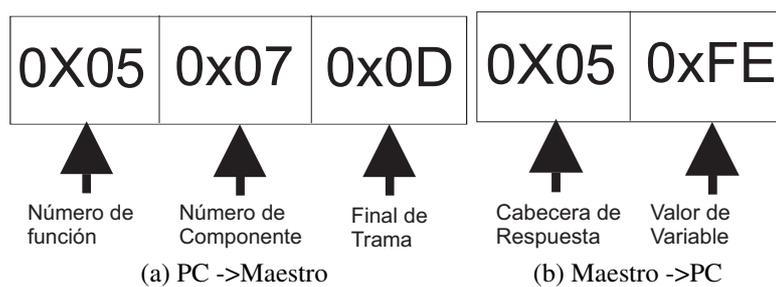


Figura 3.6: Función “Consultar variable”.

Modificar las Variables. Mediante esta función sin retorno, es posible modificar los valores de las variables de todos los componentes pertenecientes a la red, simplemente enviando después de la función los valores en el orden de la localización física de cada uno de los componentes. En la figura 3.7 es evidente que la longitud de la trama enviada al componente maestro es igual al número de componentes de la red más 2 bytes, que corresponden a la cabecera y a la terminación de trama.

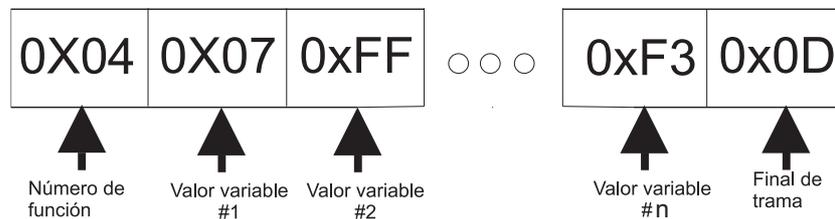


Figura 3.7: Función modificar las variables

Consultar Variables. Función con retorno que al ser ejecutada por el componente maestro responderá al ordenador central con el valor actual de todas las variables de los componentes de la red. Es importante recordar que la longitud de la trama recibida por el ordenador será directamente proporcional al número de componentes dados de alta en la red y que el orden de los bytes dentro de la trama recibida refleja el orden en el cual se encuentran físicamente, es decir, el primer byte corresponde al valor de las salidas del componente maestro, el segundo byte al primer esclavo y así sucesivamente. Podemos observar un ejemplo de la solicitud del ordenador y la respuesta del componente maestro en las figuras 3.8a y 3.4b respectivamente.

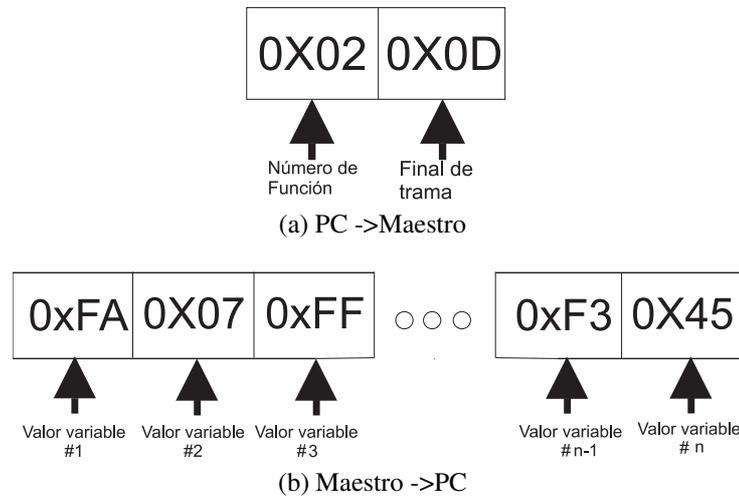


Figura 3.8: Función Consultar variable

3.2.2. Comunicación Maestro - Esclavo - Esclavo

Al haber concluido con la descripción de la comunicación entre el componente maestro y el ordenador central, ahora es el turno de describir la manera en la que se lleva a cabo la comunicación entre los demás componentes dentro del sistema, es decir, cómo es que es llevada la información recibida por el maestro proveniente del ordenador central a todos los componentes de la red y como es ejecutada. A lo largo de este apartado se describen los aspectos de la comunicación y el medio de difusión de la información entre los componentes esclavos y maestro.

A diferencia de la comunicación entre el ordenador central y el componente maestro, la transmisión de información entre los esclavos se hace de manera síncrona a través un cable de 5 hilos, es decir, existe una señal de reloj maestro que es el encargado de lograr la sincronización de los componentes para la ejecución

de la información enviada desde el componente maestro.

A pesar de esto, la comunicación entre los esclavos y el maestro no está desligada de la primera fase de comunicación, la comunicación PC<->maestro, esto debido a que como se estudió con anterioridad, existe una correspondencia entre las órdenes que llegan desde el ordenador central al maestro (funciones) y lo ejecutado por los esclavos, siendo algunas de estas decodificadas y retransmitidas a la totalidad de componentes.

La decodificación de la información está basada en la interpretación del índice y los valores de las variables de cada componente, consistiendo en representar en una cadena de bits de longitud $8*n(\text{bits})$ los valores de las variables; donde cada variable es representada de manera binaria y concatenada a la anterior en orden ascendente dependiendo de su ubicación dentro de la red; siendo n la cantidad de componentes instalados y 8 representa la cantidad de salidas digitales por cada componente. Una vez concluida la decodificación de la información, la cadena de bits obtenida por la decodificación debe de ser enviada a los esclavos de manera serial por una de las señales transmitidas por el bus de datos. Las señales sobrantes son utilizadas por los componentes esclavos para la sincronización y la habilitación.

3.3. Diseño de Hardware

Basado en la minimización de espacio, la adaptabilidad al entorno y la fácil ampliabilidad del sistema según los requerimientos del usuario, fue indispensable

el uso de tecnología que contribuyera al logro de estos propósitos.

3.3.1. Esclavo

La comunicación síncrona en la etapa comunicación Maestro ->Esclavo ->Esclavo, es debida al uso del elemento central del componente esclavo, el TPIC6B595 de Texas Instruments© el cual es un registro de desplazamiento de 8 bits con entrada serie y salidas en paralelo, esto nos indica que los datos recibidos de forma serial en la entrada serán representados en base 2 en las salidas digitales del encapsulado.

Para la transmisión de información através de los esclavos es indispensable conocer la construcción y el funcionamiento del dispositivo central. El TPIC6B595, en su construcción contiene un registro de desplazamiento serie - paralelo de 8 bits de tipo D mostrando en el diagrama de bloques de la figura 3.9.

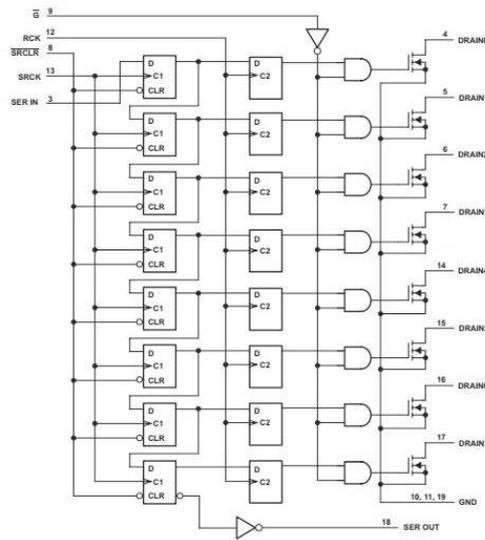


Figura 3.9: Diagrama de Bloques TPIC6B595.[14, pág.11]

Para efectuar el desplazamiento y actualización del registro de datos obtenido a través de la entrada serial (SERIN), son utilizados dos señales de reloj, el “Reloj de registro de desplazamiento” (SRCK) y el “reloj de registro” (RCK), los cuales, como se observa en la figura 3.9 son señales de reloj de sincronía de cada uno de los registros.

Tanto el SRCK como el RCK son entradas sensibles a los flancos de subida, sin embargo cada uno de ellos tienen funciones distintas. El SRCK es el responsable de que cada una de las salidas de las unidades de memoria individuales (flip-flops) del registro de desplazamiento tomen el valor de su entrada cada flanco de subida, provocando el desplazamiento de los bits por cada una de las flip-flops, es decir, es el responsable del desplazamiento bit a bit de la información recibida por el puerto SERIN cada flanco positivo, a diferencia del RCK que se encarga de la

actualización del registro de salida paralelo por las salidas DRAIN cada flanco de subida.

Es importante mencionar que aunque sean señales de reloj independientes siempre SRCK debe ser de mayor frecuencia debido a que de esto depende la coincidencia de la información recibida y la mostrada en las salidas DRAIN.

Como se observa en la fig 3.9 que el TPIC6B595 cuenta con 1 salida y 2 entradas digitales restantes las cuales son:

- SEROUT. Por medio de esta salida digital se continúa el desplazamiento de los bits de la cadena recibida por SERIN, es decir, como se puede observar está directamente conectada a la salida del último flip-flop del registro de desplazamiento; gracias a esta salida existe la posibilidad de continuar con la transmisión de información conectada en cascada mediante varios TPIC6B595, simplemente compartiendo las señales de reloj y conectando el SEROUT del registro de desplazamiento anterior con el SERIN siguiente registro.
- G. Se trata de una entrada para la habilitación de las salidas del registro paralelo, simplemente aplicando un nivel bajo en esta entrada, los valores del registro serán transparentes en las terminales DRAIN, esto debido a que como se observa en el diagrama de bloques en la figura 3.9 es un operador común en las compuertas AND conectadas a las salidas de cada flip-flop del registro paralelo.
- SRCLR. Siendo terminal común de “Reset” de los flip-flop’s del registro de

desplazamiento, simplemente poniendo en alto esta entrada, los valores de todos los flip-flop en el siguiente ciclo de reloj de SRCK serán puestos en '0', borrando dicha información que en ellos existiera[14].

En la sección 3.1 de este documento se planteó el uso de una comunicación con una topología de anillo abierto y sin *feedback*, lo que quiere decir que la información pasará por cada elemento y siendo retransmitido a los siguientes elementos, como se pudo observar gracias al uso del registro de desplazamiento serie - paralelo de 8 bits es posible la implementación de esta topología, ayude a reducir la complejidad de la electrónica requerida, siendo apta para los alcances planteados. En la figura 3.10, se muestra un diagrama de bloques donde se encuentra la comunicación entre los componentes esclavos, haciendo énfasis en el elemento central de los componentes, el TPIC6B595.

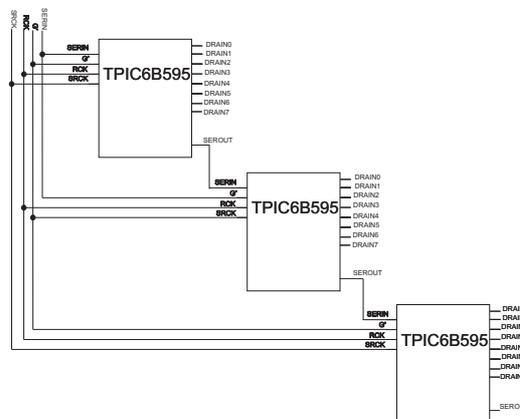
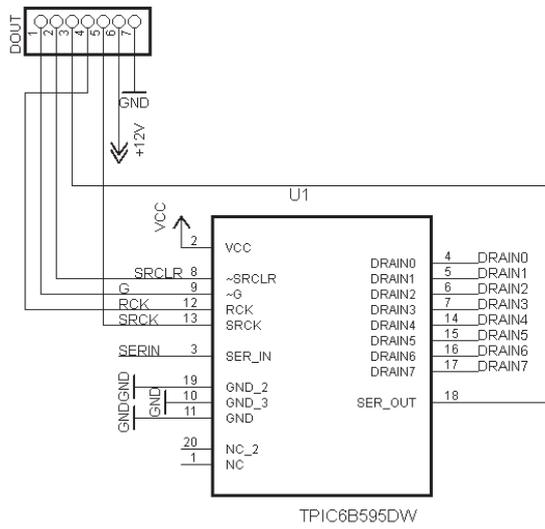


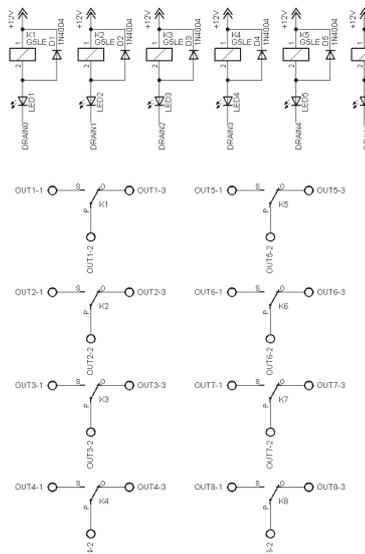
Figura 3.10: Diagrama de bloques de comunicación en cascada de TPIC6B595.

En el diagrama esquemático de la figura 3.11 se pueden observar dos zonas principales que comprenden la tarjeta esclava que serán llamadas de ahora en

adelante etapa de transmisión periférica(a) y etapa de potencia(b).



(a) Etapa de transmisión periférica.



(b) Etapa de potencia.

Figura 3.11: Fragmento de Diagrama esquemático componente esclavo.

- Etapa de potencia. En el circuito que se presenta en la figura 3.11 b se ilustra la etapa de potencia que al igual que la etapa de transmisión periférica está presente en los componentes esclavo y maestro, dicha etapa es la encargada de realizar el aislamiento por medio de relevadores el cierre de circuito entre dos contactos, los cuales serán los encargados de externar las señales digitales enviadas por el ordenador, procesadas por el maestro y externadas por cada uno de los componentes de la red.
- Etapa de transmisión periférica. En esta etapa ilustrada en la figura 3.11 a se encuentra el elemento de registro serial dentro de la red, siendo el encargado de recibir las señales provenientes del componente maestro que son usadas, como se comentó anteriormente (sección 3.3.1).

Concluido el diseño esquemático del hardware del componente maestro se procedió al diseño de circuito impreso haciendo uso de componentes de montaje superficial para la minimización de los componentes, en donde las dos etapas que componen el esclavo son integradas dentro de una sola tarjeta mostrada en la figura 3.12.

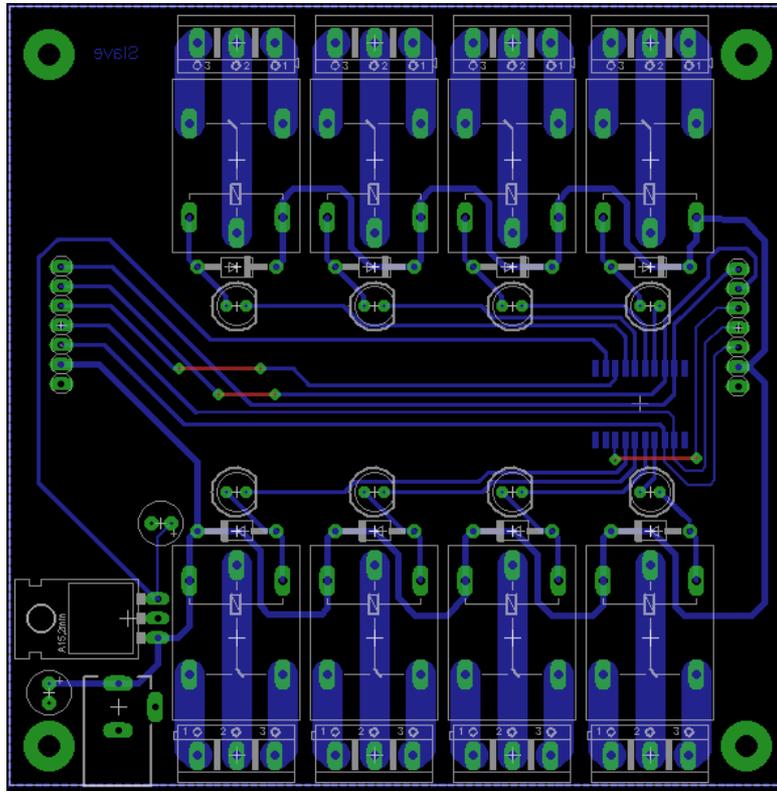


Figura 3.12: Circuito PCB del componente esclavo

3.3.2. Maestro

En la sección 3.2.1 se habló sin profundizar de la comunicación entre el maestro y el ordenador central, por tal motivo se abordará de una forma más descriptiva, centrándonos en el diseño electrónico y el desarrollo *firmware*. Esto nos lleva al diseño del circuito electrónico del componente maestro, el cual se basó en la programación de un microcontrolador de 8 bits, PIC de la familia 18Fxxxx, con la capacidad de establecer una comunicación por medio del protocolo CDC del USB de modo de con el ordenador central. Con esto se hizo posible la transmisión de

información entre ellos, para ser retransmitida a los demás componentes dentro de la red.[15]

Como se puede observar en diagrama esquemático presentado en la figura 3.13, el cual puede ser dividido en tres etapas principales, las cuales son las mismas dos etapas que en el componente esclavo, pero adicionándole una extra, la etapa de procesamiento.

- Etapa de procesamiento. Comprendida por el microcontrolador PIC 18F4550 y un puerto USB como se observa en B, es la encargada de realizar la recepción, procesamiento, y retransmisión de la información, gracias a la descripción del *firmware* dentro del microcontrolador (ver apéndice A). El diagrama electrónico del componente maestro se presenta en la figura 3.13.

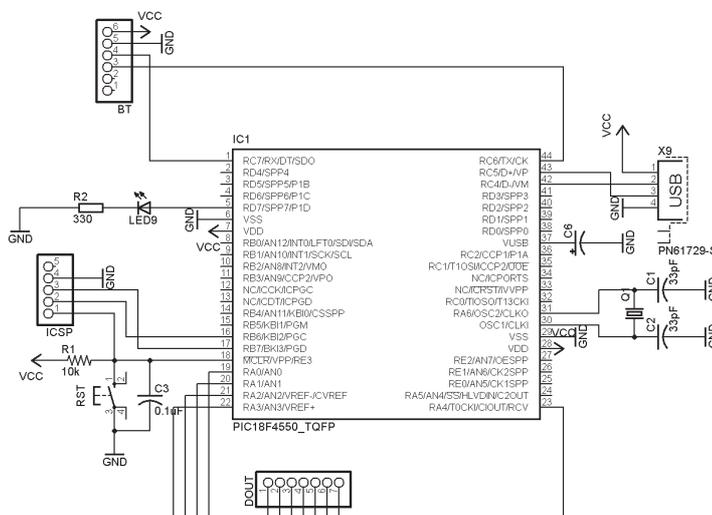


Figura 3.13: Circuito esquemático de la tarjeta maestra.

Para comprender el *firmware* desarrollado y almacenado dentro de la memoria

del microcontrolador (véase A), es esencial conocer las tareas básicas que realizará, así como el proceso que se realiza en la lectura, y la ejecución de las tareas por la etapa de procesamiento; por este motivo en la figura 3.14, de manera general se puede observar el proceso que se realiza dentro del microcontrolador.

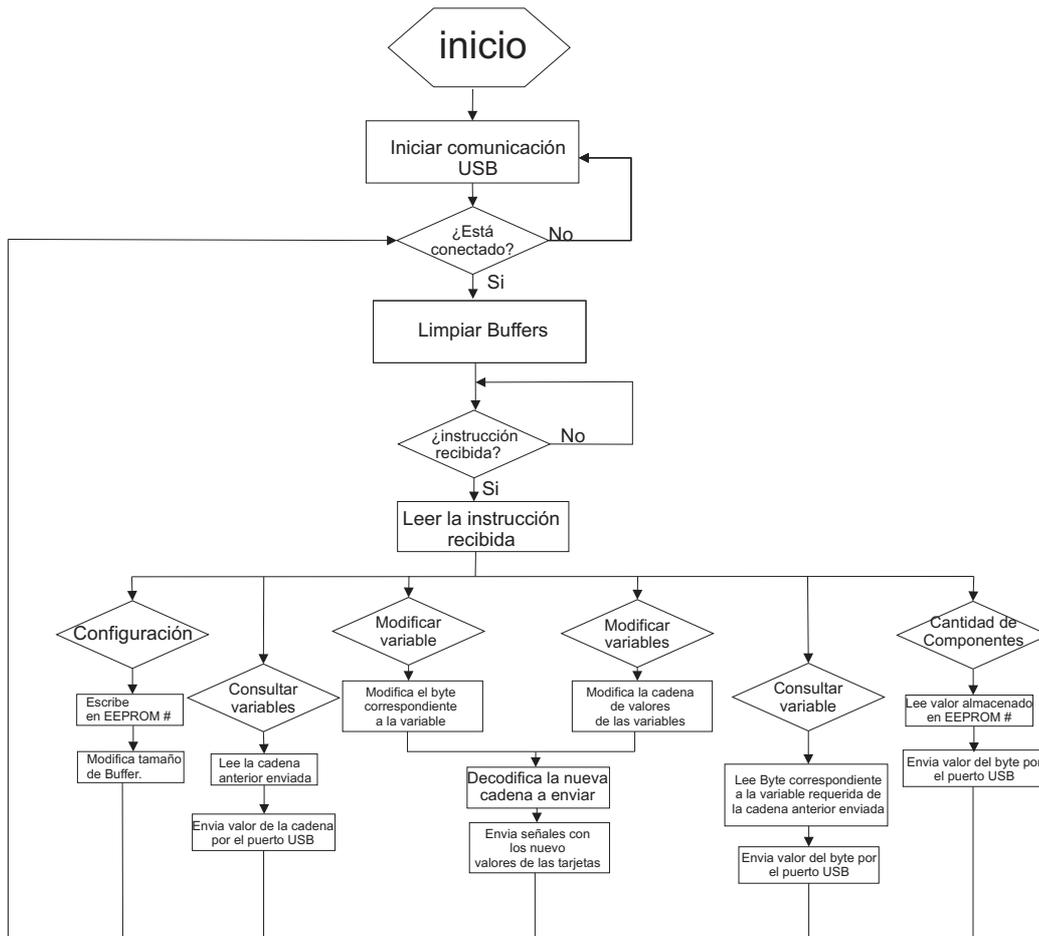


Figura 3.14: Diagrama de flujo de *firmware*.

Dentro de la ejecución del procesamiento hay tres acciones primordiales para el funcionamiento de la red como lo son:

- La comunicación USB.
- La escritura en la memoria EEPROM.
- La decodificación de la cadena recibida y envío de señales.

Las dos primeras acciones mencionadas hacen uso de tecnología embebida dentro del microcontrolador de la familia 18Fxxxx, y son responsables la interacción del componente maestro y el ordenador central, así como de la facilidad de configuración y puesta en marcha de la red.

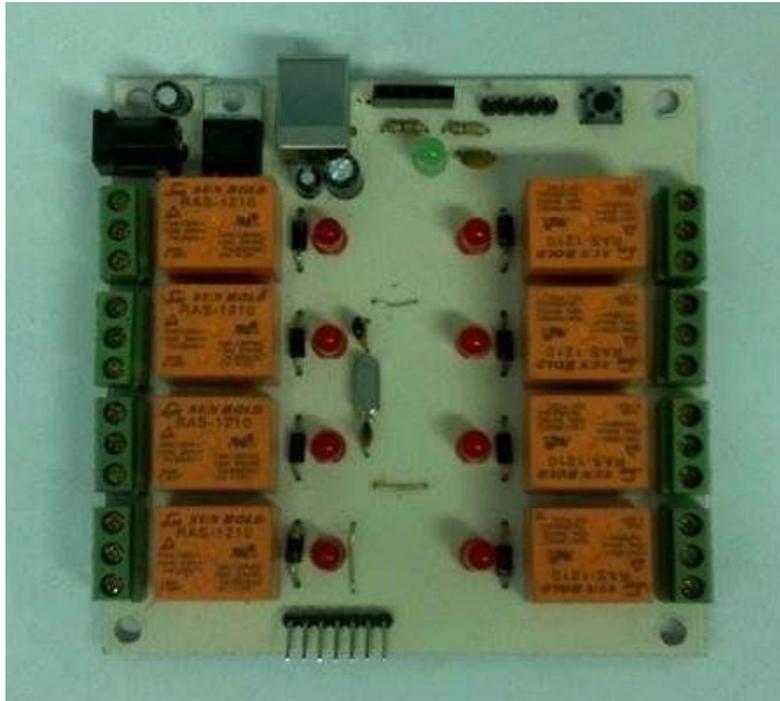
La tercera acción y aún más importante que las dos primera es la responsable de la interacción con los componentes esclavos que, como se mencionó en la sección 3.3.1 de éste documento, su funcionamiento está basado en la transmisión de los datos en el uso de registros seriales de desplazamiento. Siendo aquí donde se generan las señales que serán enviadas por el cableado que comunica los componentes.

Al igual que en el componente esclavo fue necesaria la fabricación de una tarjeta de circuito impreso para implementar el componente maestro, haciendo uso de componentes de montaje superficial, donde dentro de una sola tarjeta se encuentran las tres etapas que integran el componente maestro, el circuito impreso se puede observar en la figura 3.15.

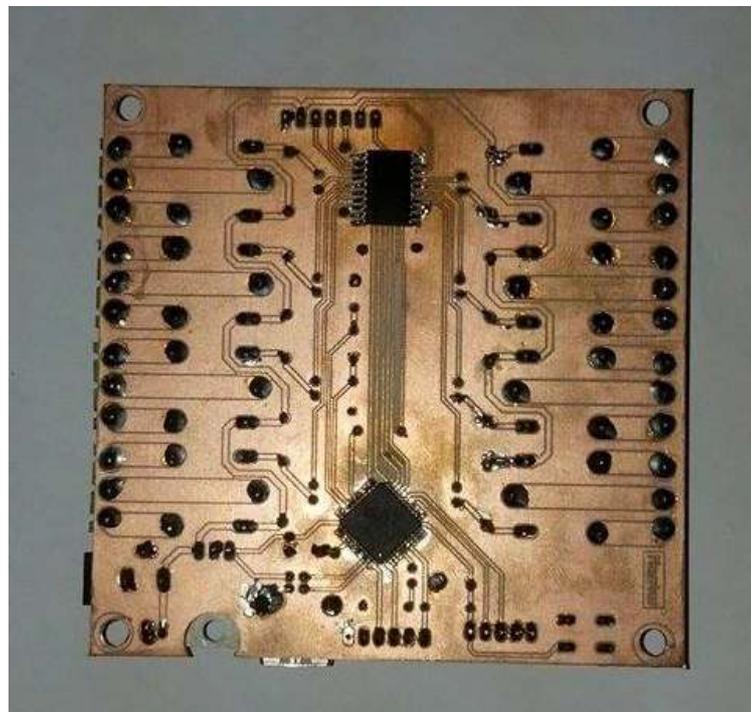
Capítulo 4

Resultados

Una vez concluido el diseño de los prototipos de los componentes esclavos y maestro se procedió a la fabricación de dos componentes esclavos y un componente maestro, figuras 4.1 y 4.2, para realizar la validación del funcionamiento del sistema se montó una red de tres elementos principalmente, además de realizar pruebas sistematizadas registrando los datos obtenidos.

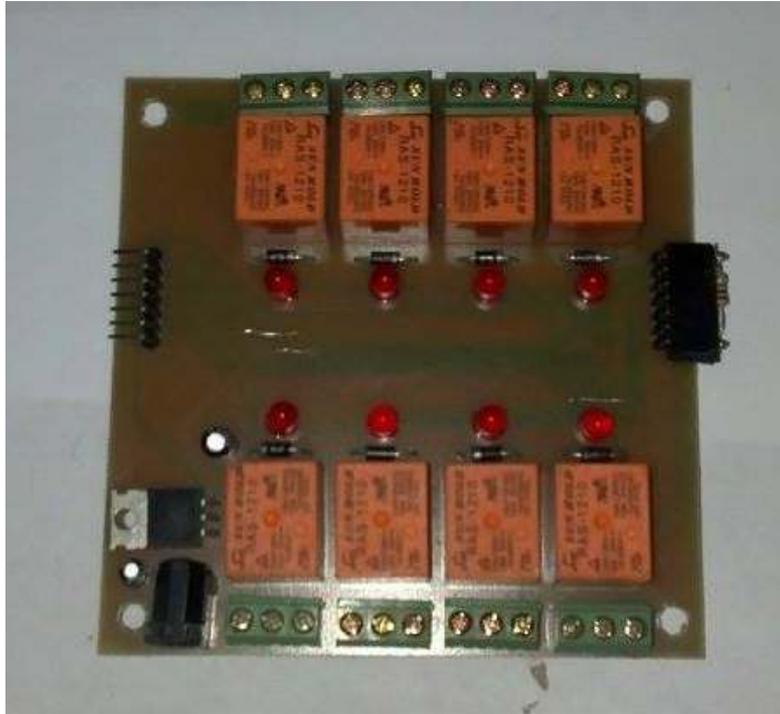


(a) Vista superior

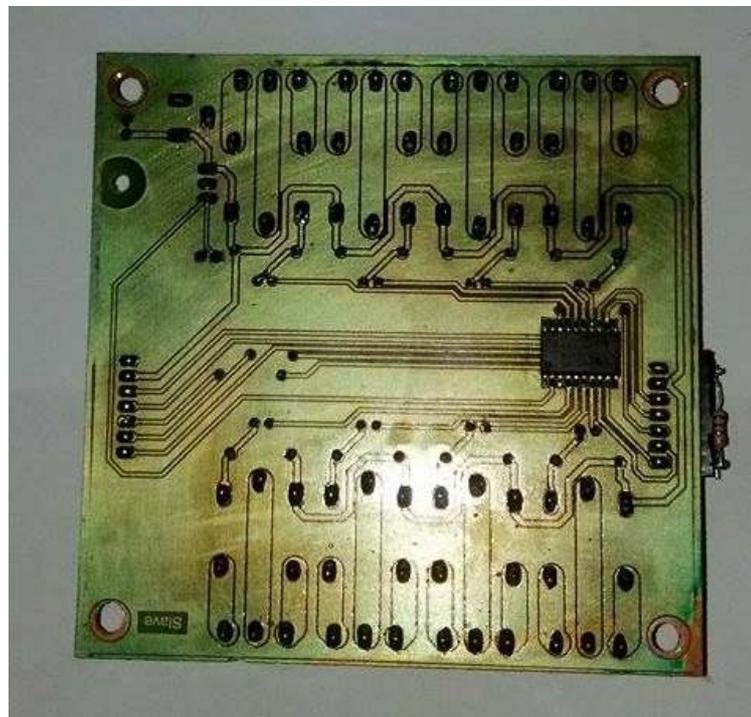


(b) Vista inferior

Figura 4.1: Prototipo componente maestro



(a) Vista superior



(b) Vista inferior

Figura 4.2: Prototipo componente esclavo

Para la realización de las pruebas se desarrolló un interfaz haciendo uso de programación en C# (ver apéndice D) utilizando el entorno de desarrollo Microsoft VisualStudio 2010. Por medio de esta aplicación se realizó el envío y recepción de información entre el ordenador central y el componente maestro de la red. En la figuras 4.3 y 4.4 se pueden observar la diversas funciones de la interfaz las que concuerdan con la las funciones de que pueden ser ejecutadas por el sistema.



(a) Conexión via USB



(b) Configuración de Módulos

Figura 4.3: Interfaz de Prueba



(a) Modificación de una Variable



(b) Modificación de Variables

Figura 4.4: Interfaz de Prueba

4.0.3. Verificación de funcionalidad

Siendo una prueba puramente cualitativa, el principal objetivo de la realización del experimento fue la validación del funcionamiento de una red de 3 elementos a una corta distancia (considerándose corta una distancia 10 cm de cable multiconductor de calibre AWG 22 entre cada uno de los componentes). Para validar el experimento como satisfactorio fue necesaria la correcta conexión del componente maestro al ordenador central, y la transmisión de información entre los componentes, es decir, la activación de las salidas digitales. Dichas pruebas se ilustran en la figura 4.5 donde se puede observar dicho funcionamiento.

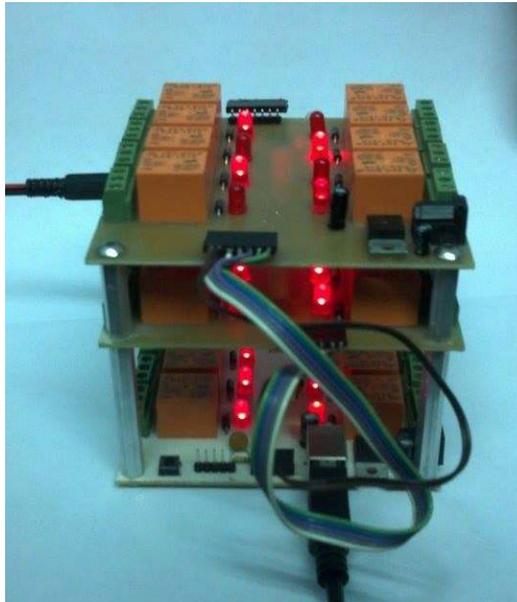


Figura 4.5: Evidencias de funcionamiento del primer experimento

4.0.4. Verificación de integridad de datos.

El experimento descrito a continuación fue realizado con el objetivo de validar y conocer la eficiencia de la transmisión y procesamiento de información enviada a través de la red, para la realización del experimento se consideraron diversas variables, las cuales se consideraron podrían influir en el desempeño del sistema como los son:

- Longitud de cable.
- Velocidad de transmisión.
- Número de componentes instalados.

El experimento consiste en el envío de 100 tramas de información para la activación de las varias salidas digitales en 3 componentes instalados dentro de la red (un componente maestro y dos esclavos), conectados entre sí por un cable multi-conductor de calibre AWG22 a una distancia de 10, 15 y 20 metros entre cada uno de los componentes y con una velocidad de transmisión de 25, 50, 100 Kbits/s. Al momento de realizar el experimento se realizó el monitoreo de las señales de comunicación, apoyándose en el uso de un osciloscopio digital, realizando mediciones en dos principales puntos, la salida del componente maestro y en la entrada del ultimo componente dentro de la red, con la finalidad de observar el comportamiento de las señales de comunicación, en la figura 4.6 a una velocidad de 100 kbps se presenta la captura de pantalla del osciloscopio de una muestra de las señales obtenidas en el primer punto de monitoreo. Donde las señales mostradas son G(señal 1) , SRCK(señal 2) y SEROUT(señal 3).

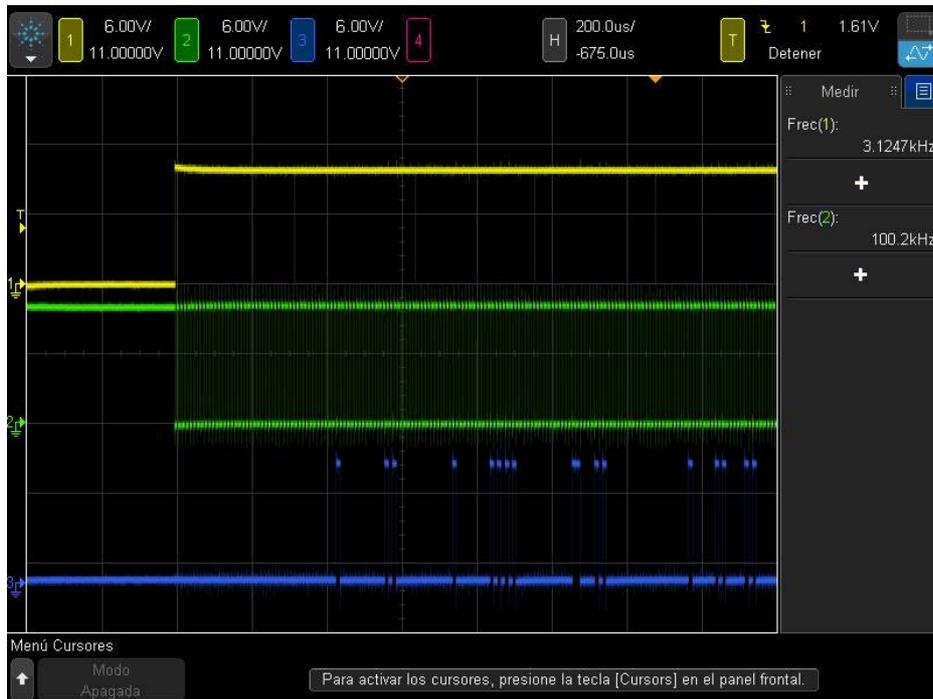


Figura 4.6: Señales de salida del componente maestro

Los datos arrojados por la experimentación han sido plasmados en la tabla 4.1 donde se puede observar el comportamiento del sistema. Como se puede observar la eficiencia del sistema va decayendo proporcionalmente al aumento de velocidad de transmisión y longitud entre los componentes.

longitud (m)	velocidad (kbps)	%eficiencia
10	25	100
	50	99
	100	99
15	25	98
	50	97
	100	89
20	25	80
	50	80
	100	75

Tabla 4.1: Pruebas de integridad y eficiencia

Al indagar y observar el comportamiento de los componentes esclavos instalados dentro de la red obtenidas en el experimento con una velocidad de transmisión de 100 kbps y una longitud de 20 m, se observó una caída de voltaje, además de presentar ruido eléctrico en cada una de las señales de entrada del último componente esclavo. Estos fenómenos se pueden observar en la figura 4.7, donde las señales con mayor afectación son las señales de G y SRCK que cumplen un papel fundamental dentro del funcionamiento del sistema, al ser las encargadas de la habilitación de transmisión y de la sincronía de la comunicación respectivamente.



Figura 4.7: Señales de entrada del ultimo componente esclavo

Capítulo 5

Conclusiones

Una vez concluido con el desarrollo y la experimentación de los prototipos para la implementación de una red centralizada de activación de alarmas remotas, basado en una comunicación síncrona entre sus elementos, se logró observar el buen comportamiento del sistema a una distancia no mayor a 15 m y una velocidad de transmisión de 50kbps, obteniendo un porcentaje de error no mayor del 3 % .

Sin embargo las pruebas realizadas a la red, mostraron que el uso de mayores velocidades y longitudes entre cada uno de los componentes del sistema provoca la susceptibilidad del sistema a un aumento del error que crece proporcionalmente con la longitud de la red. Los efectos inducidos por las caídas de voltaje en las señales de transmisión y el ruido eléctrico son debidos al uso de señales no diferenciadas.

El uso de señales diferenciadas aporta una mayor robustez a los efectos presentados debido al aumento de longitud de los medio de transmisión, con esto se

puede concluir que la implementación de un estándar de comunicación como lo es el RS-422 o RS-485 como intermediario entre los componentes de la red aportaría la disminución del error en la transmisión de información a largas distancias. Debido a que el sistema está basado en una arquitectura abierta nos permite la adaptación de componentes extras dentro de la red como lo puede ser un componente que permita la incursión al sistema de los estándares antes mencionados.

Bibliografía

- [1] Galloway, B., Bancke G. P.(2014). *Introduction to Industrial Control Networks. IEEE COMMUNICATIONS SURVEYS & TUTORIALS, 15(2)pp. 860 -880.*
- [2] Guerrero, V., Yuste R. L., Martinez L. (2009). *Comunicaciones industriales. México. Alfaomega*
- [3] Rodríguez, A. (2007). *Sistemas Scada. México. Alfaomega*
- [4] Piedrafita, R. (2010). *Ingeniería de la automatización industrial. México. Alfaomega*
- [5] EIA standard RS-232-C: Interface between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange. *Washington. Electronic Industries Association. Engineering Dept.*
- [6] Tomasy, W. (2003). *Sistemas de comunicaciones electrónicas. México. Pearson Educación.*

- [7] Lazo, J. A. y López C. E. (1999). *Análisis en automatización industrial. Tesina. Mexico, Universidad Autónoma de Querétaro.*
- [8] Bicsi, B., (2002). *Network Design Basics for Cabling Professionals. McGraw-Hill Professional.*
- [9] Correa, L. y Rodríguez, J (2009). *Descripción del bus de campo DeviceNet . Tesina. Mexico, Universidad Autónoma de Querétaro.*
- [10] Perez, G. (2004). *Red de comunicación Profibus. Tesina. Mexico, Universidad Autónoma de Querétaro.*
- [11] Ley para el desarrollo de la competitividad de la micro,pequeña y mediana empresa. *Diario Oficial de la Federación.Mexico, DF. 21 de enero de 2015.*
- [12] INEGI. (2014). *Censos Económicos 2014.14 de enero de 2015, de INEGI Sitio web: <http://www.inegi.org.mx/est/contenidos/proyectos/ce/ce2014/>*
- [13] SIEM (2015). *Registro de empresas ante SIEM.2 de septiembre de 2015, de SIEM Sitio web: <http://www.siem.gob.mx/siem/estadisticas/estadotamanoPublico.asp?tam=0&p=1>*

- [14] Texas Instruments (2015). *Hoja de Datos TPIC6C595 Power Logic 8-Bit Shift Register*. 9 de junio de 2016 de sitio web <http://www.ti.com.cn/cn/lit/ds/symlink/tpic6c595.pdf>
- [15] Microchip Technology Inc (2006). *PIC18F2455/2550/4455/4550 DataSheet*. 9 de junio de 2016 de sitio web <http://ww1.microchip.com/downloads/en/devicedoc/39632c.pdf>
- [16] Fairchild Semiconductor Corporation (2014). *LM78XX/LM78XXA 3-terminal 1 A positive Voltage Regulator*. 9 de junio de 2016 de sitio web <https://www.fairchildsemi.com/datasheets/LM/LM7805.pdf>

Apéndice A

Código de firmware de microcontrolador

```
#include <18f4550.h>

#fuses
    HSPLL, NOWDT, NOPROTECT, NOLVP, NODEBUG, USBDIV, PLL5, CPUDIV1, VREGEN

#use delay(clock=48M)

#use rs232
    (BAUD=9600, XMIT=PIN_C6, RCV=PIN_C7, BITS=8, PARITY=N)

//definicion de señales de control de registro serial

#define G    PIN_A2
#define SRCLR PIN_A1
#define SERIN PIN_A0
#define RCK PIN_A3
```

```
#define SRCK PIN_A4
#define targetmax 255
char a[targetmax],str[targetmax];
int i=0;
#include <usb_cdc.h>
#include <usb.h>

void configurar(char);
void leerConf(void);
void correr(void);
void modificarByte(char, char);
void modificar( void );
void clrBuf(void);
void clrStr( void );
void leerValor(void);

//recepcion de datos por puerto UART
#int_rda
void RX()
{
    a[i++]=getc();
}
```

```
}

void main()
{
    a[0]=0;
    usb_cdc_init();
    usb_init();
    enable_interrupts(int_rda);
    clrStr();// limpieza de buffer de datos
    clrBuf();// limpieza de buffer de recepcion de datos
do
{

    usb_task();
    if(usb_enumerated())
    {
        output_toggle(PIN_D7);// led indicador de conexion
        usb
        delay_ms(100);

        // recepcion de datos via USB
    }
}
```

```
while(usb_cdc_kbhit())
{
    a[i++]=usb_cdc_getc();
}
}
else
{
    output_high(PIN_D7);
}
switch (a[0])
{
    case 0x00:
        clrBuf();
        break ;

    case 0x01: //configuracion de modulos
        if(a[2]==0x0D)
        {
            configurar(a[1]);
            clrStr();
            correr();
            clrBuf();
        }
        break;
```

```
case 0x02 :
    correr();
    clrBuf();
break ;

case 0x03 ://modificar byte
    if (a[3]==0x0D)
    {
        modificarByte(a[1],a[2]);// modificador de
            valor de byte (# de modulo, valor)
        correr();
        clrBuf();
    }
break;

case 0x04 : //modificar trama
    if(a[i-1]==0x0D)
    {
        modificar();//modificador de trama
        correr();// funcion de enviar datos a los
            registros
        clrBuf();
    }
}
```

```

break;

case 0x05 ://leer valores
    if (a[2]==0x0D)
    {
        int index = a[1];
        printf("%c%c",0x05,str[index-1]);// envia
            valor apor el puerto uart de el valor en la
            tarjeta
        printf(usb_cdc_putc,"%c%c",0x05,str[index-1]);
        clrBuf();
    }
break ;

case 0x06://LEER CONFIGURACION
    if(a[1]==0x0D)
    {
        leerConf();
        clrBuf();
    }
break;

default :

```

```
        //clrBuf();

        break ;

    }

}while(1);

}

void configurar(char modules)
{
    write_eeprom(0x00,modules);//escribe en la eeprom
    interna el valor de modulos confiurados
}

void leerConf()// lee el valor de modulos configurados y
    responde con el valor
{
    char b = read_eeprom(0x00);
    printf("%c%c",0x01,b);
    printf(usb_cdc_putc, "%c%c",0x01,b);
}

void correr()
{
    int i,j,bit,mod=read_eeprom(0x00);
    j=mod;
    output_low(G);// desabilita muestra de valor salida
```

```

output_low(SRCLR); // limpia flipflop de entrada
output_high(SRCLR); // permite entrada de datos a registro
do
{
    j--;
    for (i = 0; i < 8; i++)
    {
        bit = str[j] & (0x80 >> i); // valor almacenado en
            el arreglo comparacion bit a bit
        if (bit)
            output_high(SERIN); // si el valor es uno manda 1
                al buffer de registros
        else
            output_low(SERIN); // si es 0 envia 0 al buffer de
                registros
        output_high(SRCK); // reloj de sincronia de primer
            registro
        output_low(SRCK); // reloj de sincronia de primer
            registro
    }

}while(j!=0);

output_high(RCK); //reloj de sincronia 2dos flipflops
output_low(RCK); // reloj de sinrocnia 2dos flipflops

```

```
}  
  
void modificarByte(char module,char value)  
{  
    str[module-1]=a[2];  
}  
  
void modificar()  
{  
    clrStr();  
    int h;  
    int tar=read_eeeprom(0x00);//lee el valor de las tarjetas  
        confiuradas  
    for(h=0;h<tar;h++)  
        str[h]=a[h+1];// copia el valor de el buffer de  
            recepcion al buffer de datos  
}  
  
  
void clrBuf()  
{  
    int h;  
    for(h=0;h<targetmax;h++)  
        a[h]=0x00;  
    i=0;  
}
```

```
void clrStr()  
{  
    int h;  
    for (h=0;h<targetmax;h++)  
        str[h]=0;  
}
```

Apéndice B

Herramientas para fabricación de PCB

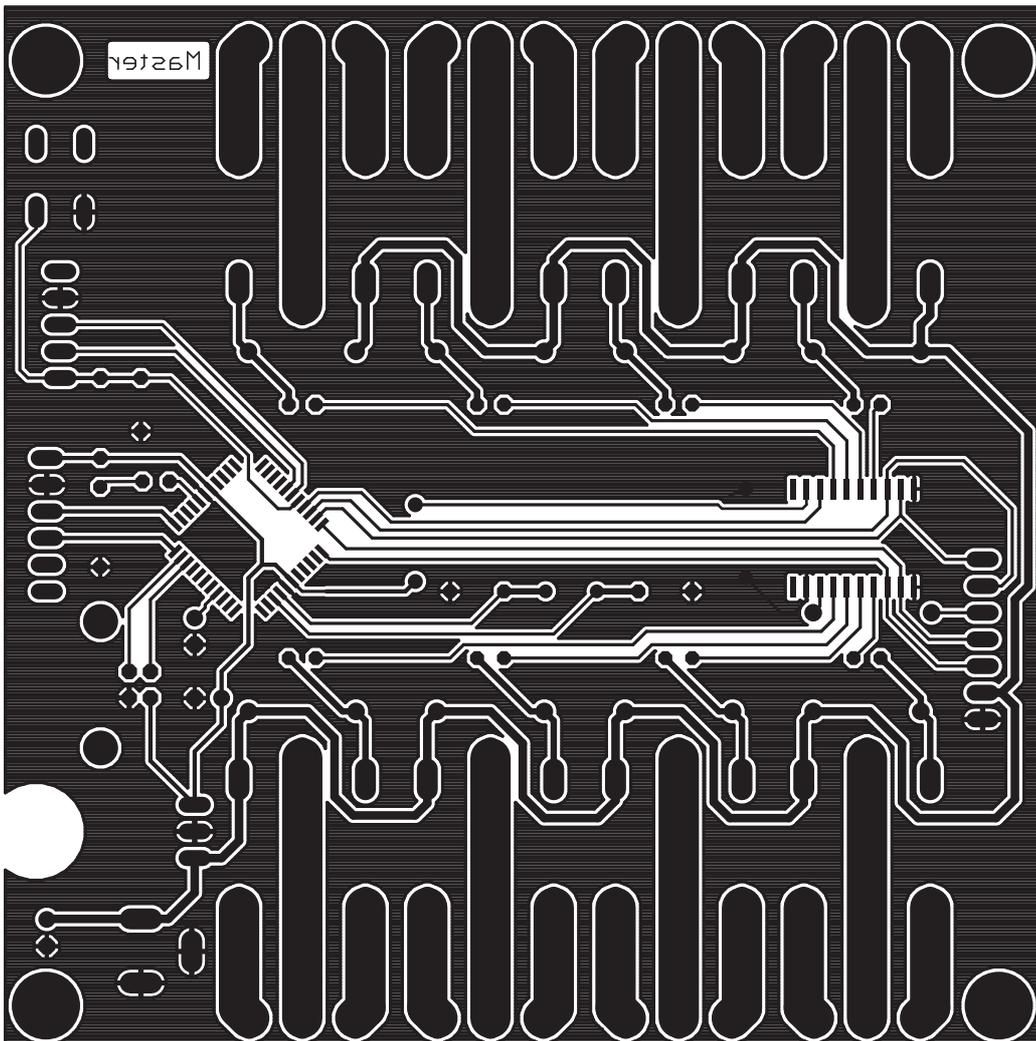


Figura B.1: Diseño PCB de componente maestro

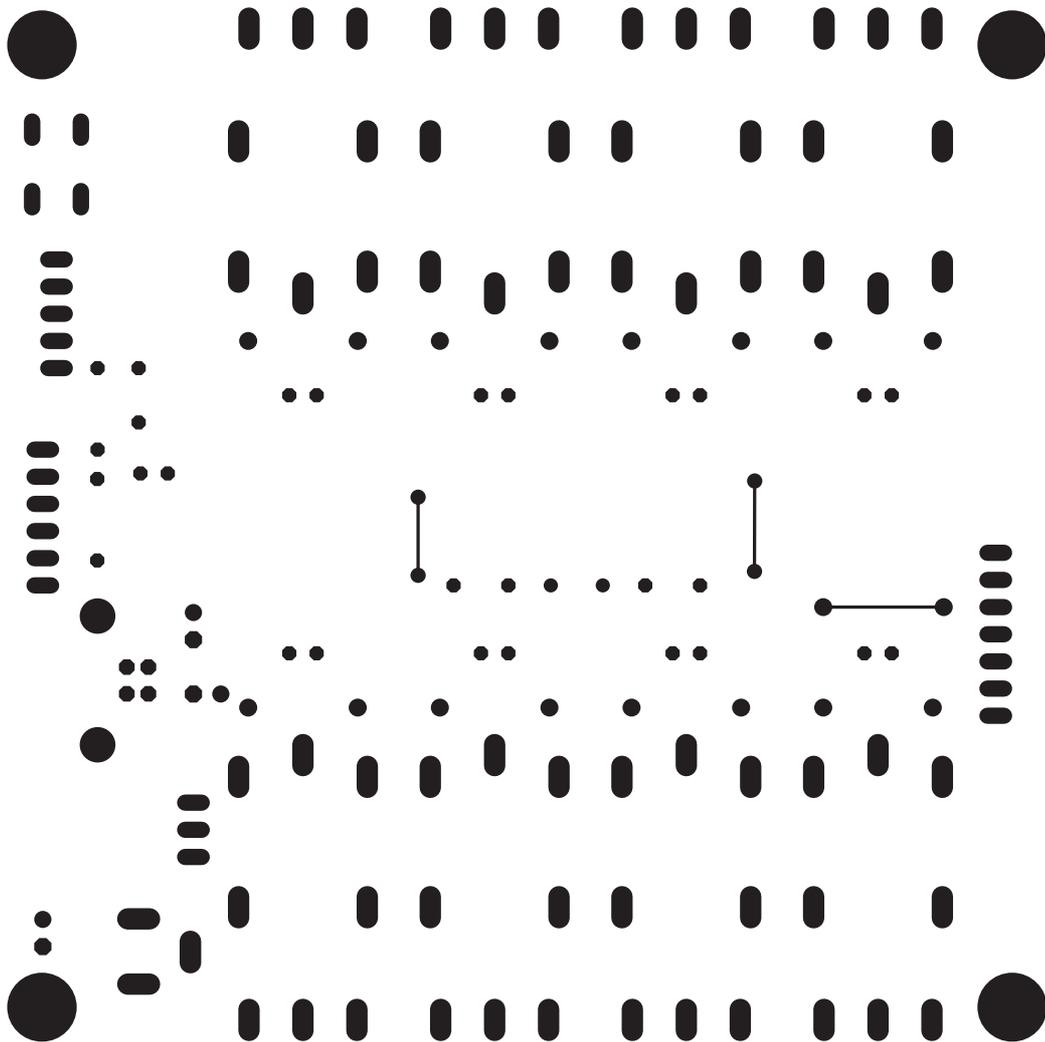


Figura B.2: Positivo para aplicacion de pintura antisoldante (maestro

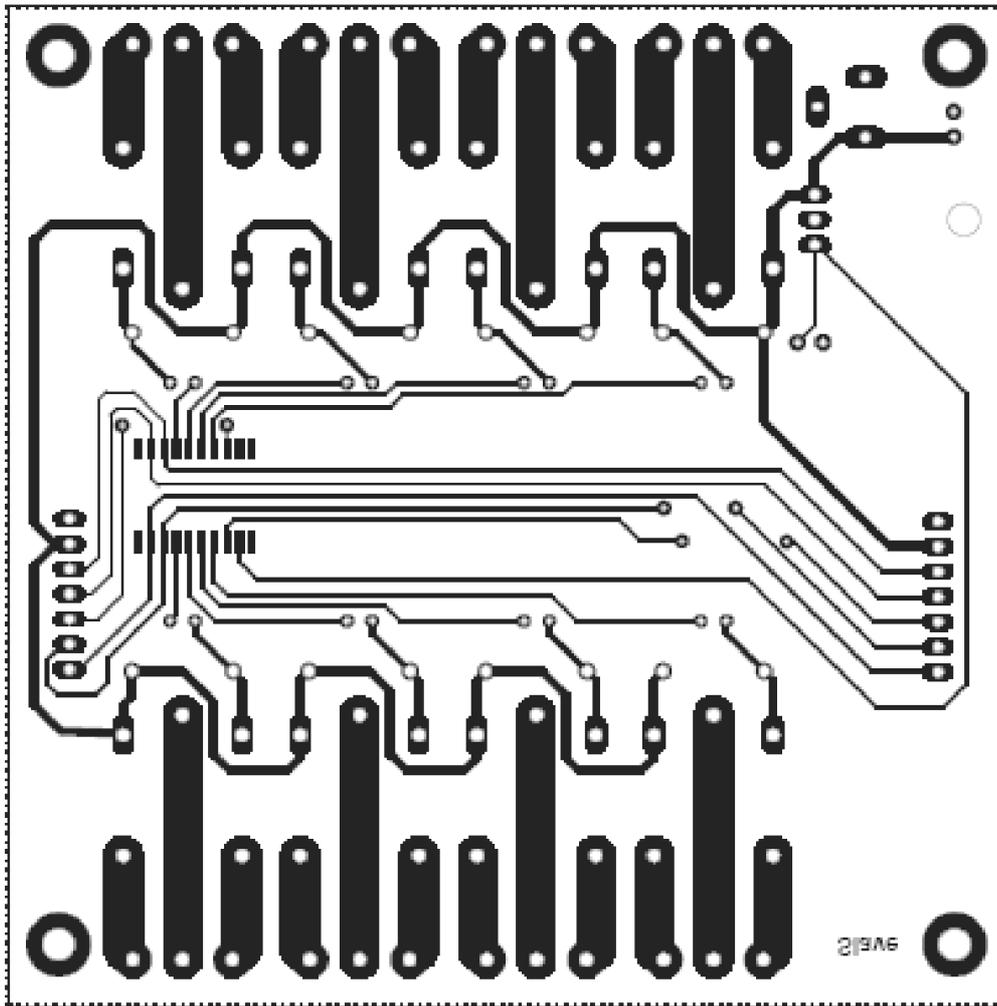


Figura B.3: Diseño PCB de componente esclavo

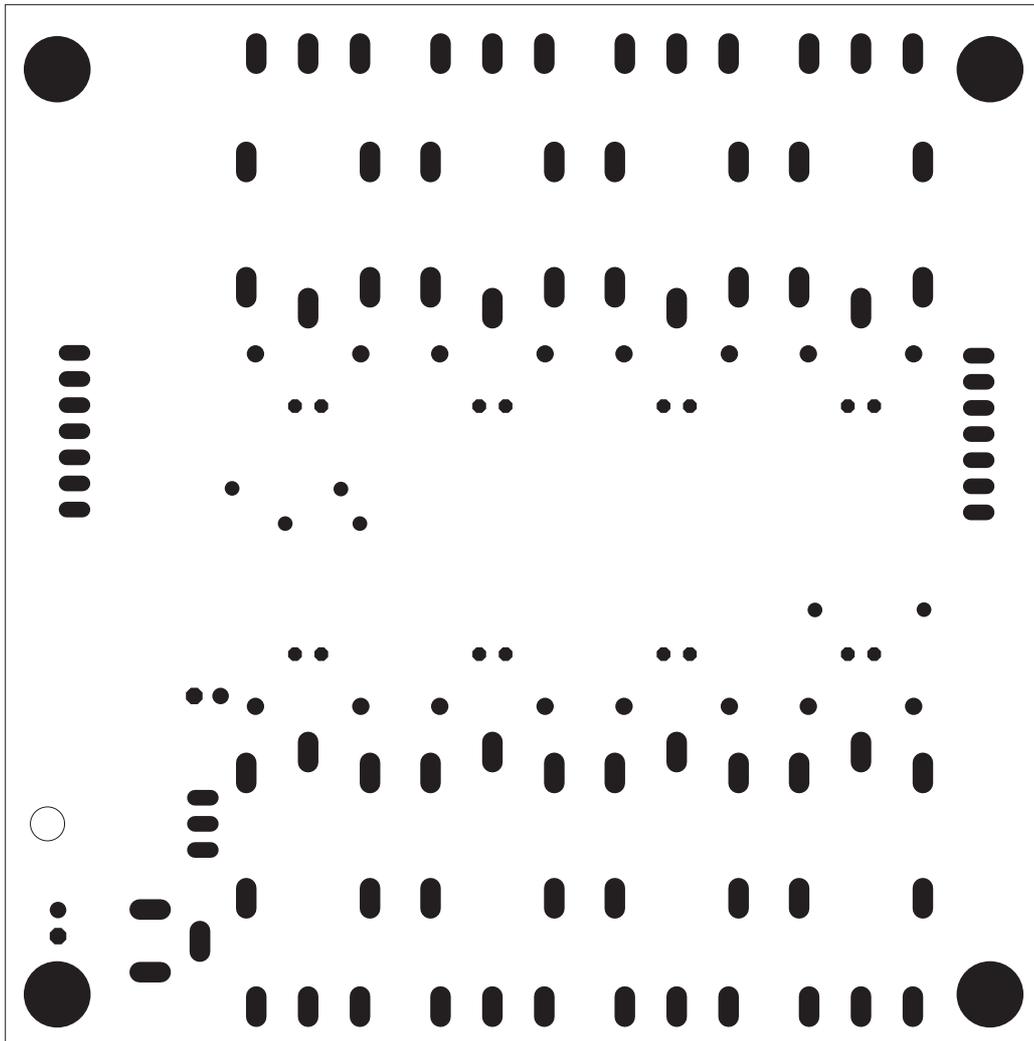


Figura B.4: Positivo para aplicacion de pintura antisoldante (esclavo)

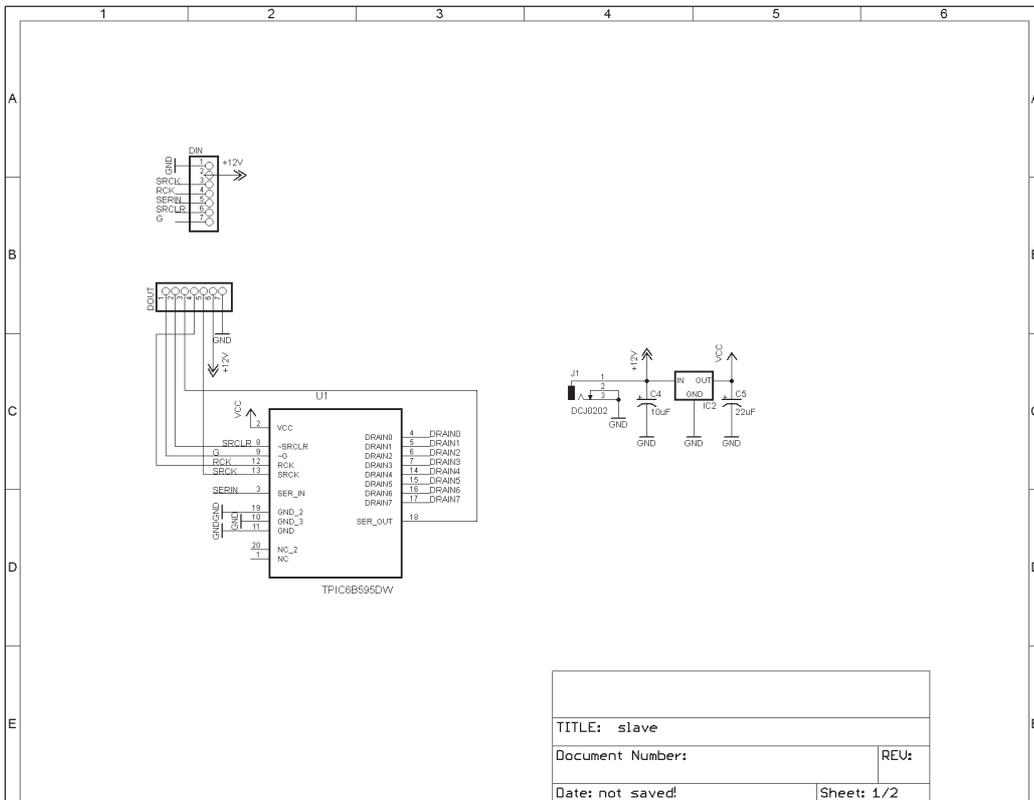


Figura C.2: Diagrama esquemático de potencia componente esclavo

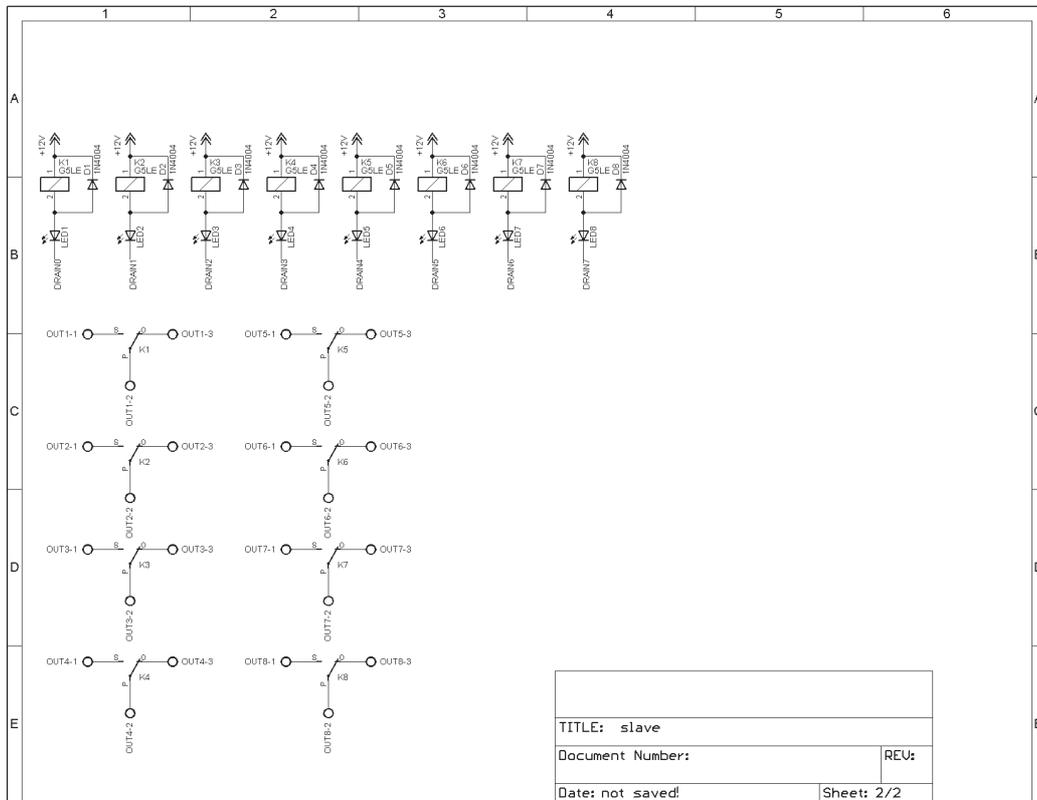


Figura C.3: Diagrama esquemático de transmisión periférica esclavo

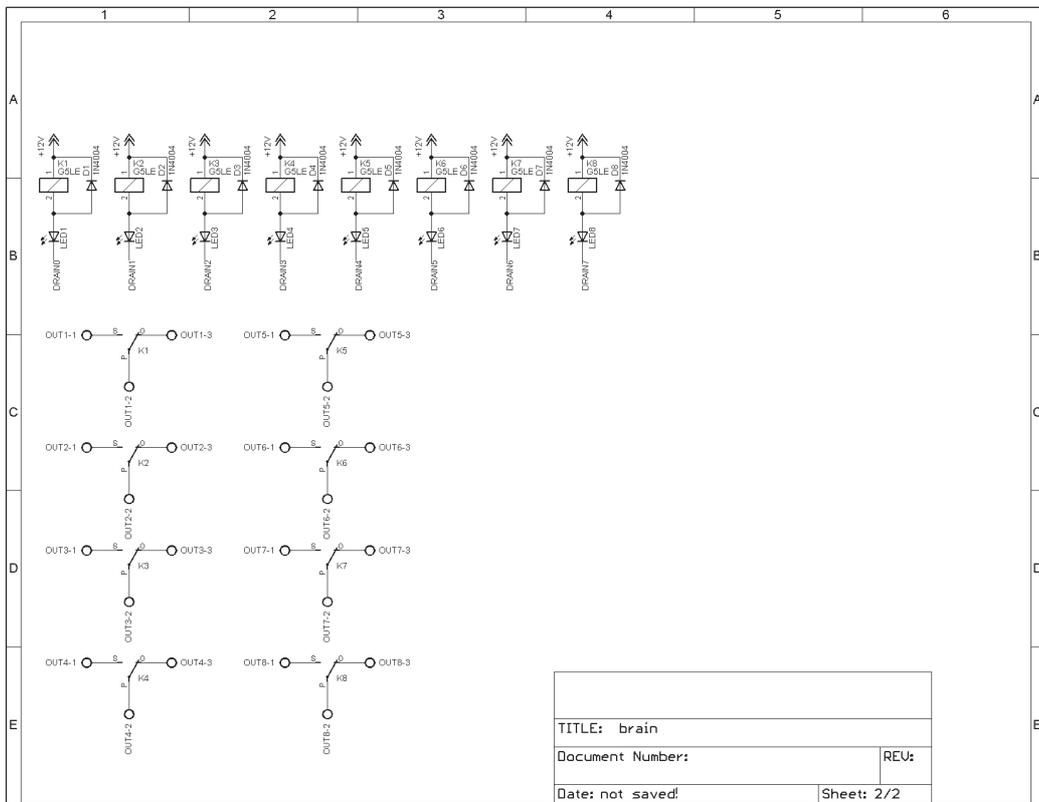


Figura C.4: Diagrama esquemático de potencia componente esclavo

Apéndice D

Código de interfaz de experimentación

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using Dotnetrix;  
using System.Windows.Forms;  
using System.Collections;
```

```
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {

        byte[] Rx = new byte[4];
        byte[] Tx = new byte[4];
        Object[] NoModules = new Object[256];
        String[] Puertos =
            System.IO.Ports.SerialPort.GetPortNames();

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {

            this.puerto.DataSource = Puertos;
            for (int i = 0; i <= 255; i++)
            {
                NoModules[i] = i + 1;
            }
        }
    }
}
```

```
    }  
    modules.Items.AddRange(NoModules);  
}  
  
private void Conect_Click(object sender, EventArgs e)  
{  
  
    serialPort1.PortName = puerto.Text;  
    try  
    {  
        serialPort1.Open();  
  
    }  
    catch  
    {  
        MessageBox.Show("Puerto no válido");  
        return;  
    }  
    puerto.Enabled = false;  
    tabControll.Enabled = true;  
}
```

```
private void Disconnect_Click(object sender,
    EventArgs e)
{
    serialPort1.Close();
    puerto.Enabled = true;
    tabControl1.Enabled = false;

}

private void DisplayText(object sender, EventArgs e)
{

    label4.Text = Convert.ToString(Rx[1]);

}

private void ModificarTarjetas(object sender,
    EventArgs e)
{
    modSel.Items.Clear();

    for (int i = 1; i <= (int)Rx[1]; i++)
    {
        modSel.Items.Add(i);
    }
}
```

```
    }  
    private void ModValueModule(object sender, EventArgs  
        e)  
    {  
        int g;  
  
        BitArray bits = new BitArray (new byte[] {Rx[1]});  
        for ( g=0; g < checkedListBox1.Items.Count; g++)  
        {  
  
            if(bits[g] == true)  
                checkedListBox1.SetItemCheckState(g,  
                    CheckState.Checked);  
            else  
                checkedListBox1.SetItemCheckState(g,  
                    CheckState.Unchecked);  
  
        }  
    }  
  
    private void serialPort1_DataReceived(object sender,  
        System.IO.Ports.SerialDataReceivedEventArgs e)  
    {
```

```
int bytes = serialPort1.BytesToRead;

byte[] buffer = new byte[bytes];

serialPort1.Read(buffer, 0, bytes);
Rx = buffer;
this.Invoke(new EventHandler(DisplayText));

if (Rx[0] == 0x01)
    this.Invoke(new
        EventHandler(ModificarTarjetas));
else if (Rx[0]==0x05)
    this.Invoke(new EventHandler(ModValueModule));
}

private void configurar_Click(object sender,
    EventArgs e)
{
    byte mod = byte.Parse(modules.Text);
    Tx[0] = 0x01;
    Tx[1] = mod;
    Tx[2] = 0x0D;
```

```
        serialPort1.Write(Tx, 0, 3);
    }

private void ReadValue_Click(object sender,
    EventArgs e)
{
    byte mod = byte.Parse(modSel.Text);
    Tx[0] = 0x05;
    Tx[1] = mod;
    Tx[2] = 0x0D;
    serialPort1.Write(Tx, 0, 4);
}

private void sendByte_Click(object sender, EventArgs
    e)
{
    byte valueByte=0;
    BitArray bits = new BitArray(new byte[8]);
    for (int g = 0; g < checkedListBox1.Items.Count;
        g++)
    {
        if (checkedListBox1.GetItemCheckState(g) ==
            CheckState.Checked)
```

```
        bits[g] = true;
    else
        bits[g] = false;
    }
    for (int g = 0; g < bits.Count; g++)
    {
        if (bits[g] == true)
            valueByte += (byte) (Math.Pow(2, g)) ;
    }

    byte mod = byte.Parse(modSel.Text);
    //byte var;
    Tx[0] = 0x03;
    Tx[1] = mod;
    Tx[2] = valueByte;
    Tx[3] = 0x0D;
    serialPort1.Write(Tx, 0, 4);
    }
}
}
```
