

INSTITUTO TECNOLÓGICO DE COSTA RICA

ESCUELA DE INGENIERÍA ELECTRÓNICA



COMPAÑÍA NACIONAL DE FUERA Y LUZ S.A.

CNFL S.A.

“Actualización tecnológica de una tarjeta convertidora de protocolos aplicada en un sistema SCADA para enviar comandos y recibir datos desde una red con protocolo Modbus a una DPU con protocolo ASCII.”

**Informe de Proyecto de Graduación para optar por el Grado de Bachiller en
Ingeniería Electrónica**

Eduardo Rojas Chinchilla

CARTAGO, Junio del 2001

Dedico este trabajo a mi querida esposa y a mis padres, quienes han estado junto a mi en los momentos más difíciles de mis estudios, para ellos, con todo mi corazón.

También quiero dedicar este trabajo a mi hermano, a mi primo y a mi abuelo, quienes desde el cielo me han dado la fortaleza necesaria para seguir adelante.

Por último, le dedico este esfuerzo a los seres más importante en mi vida: a la Santísima Trinidad, a la Virgen María y a Don Bosco.

Agradecimiento

A lo largo de este proyecto, he recibido la invaluable ayuda de muchas personas, que no sólo me han aportado su conocimiento, sino que también me han brindado su amistad.

Quiero agradecer al Ing. Eric Bogantes, por haberme dado la oportunidad de realizar mi proyecto de graduación en el Centro de Control de Energía, por la confianza que depositó en mí y por toda la ayuda que me brindó.

Al Ing. Alberto Coto le agradezco primero que nada su amistad, y luego, todo el conocimiento sobre protecciones y redes de distribución que muy generosamente compartió conmigo. Al resto de mis compañeros: Tatiana Mora, Jafeth Carrillo, Gerardo Zúñiga, Lic. Carlos Carrillo, Ing. Franco Mora, Ing. Jaime Hidalgo y al Ing. José Salazar, quienes con su amistad, me hicieron sentir como un miembro más de la empresa, a ellos les debo mucho y estaré siempre muy agradecido.

También, quiero agradecer a Xiomara y a Adán, quienes no solo me ayudaron a incorporarme a la CNFL como practicante, sino que también me han brindado una amistad sin igual.

Y por último, quiero agradecer a todo el personal de la CNFL, ya que nunca me cerraron las puertas y me brindaron las herramientas necesarias para llevar a cabo mi proyecto.

Resumen

En el Centro de Control de Energía (CCE) de la Compañía Nacional de Fuerza y Luz, S.A., se hace uso de un sistema SCADA para monitorear la red de distribución. En esta red de distribución, hay unidades de protección de distribución (DPU por sus siglas en inglés), las cuales, aparte de brindar información útil, son dispositivos, que bajo ciertas fallas de la red de distribución, abren y cierran interruptores, con el fin de evitar daños a equipos de alto costo económico.

Hasta el momento, el CCE no ha conectado las DPUs al sistema SCADA, a pesar de que son dispositivos muy útiles. La razón de esto, es que el protocolo de comunicación que utilizan no es compatible con el utilizado por el sistema SCADA, el cual utiliza un protocolo Modbus RTU, mientras que las DPUs se comunican con un protocolo Modbus ASCII.

En este proyecto se actualizó tecnológicamente una tarjeta convertidora de protocolos, la cual le permite a las DPUs conectarse a la red de monitoreo. Esta tarjeta se instala dentro de cada DPU, haciendo que estos dispositivos trabajen como cualquier otro esclavo en el sistema SCADA.

Para actualizar la tarjeta convertidora de protocolos, primero se tuvieron que estudiar todas las características, tanto de hardware como de software, de la tarjeta existente, examinar sus puntos débiles y buscarles una mejor solución. El resultado fue una tarjeta más eficiente que la existente, además de ser mucho más pequeña, dando facilidades para ser instalada dentro de la DPU.

Palabras clave: Protocolos; RTU; sistema SCADA; ASCII; DPU; red de monitoreo.

Abstract

In the Energy Control Center (CCE for their initials in Spanish) of the Force and Light National Company, S.A., is made use of a SCADA system for viewing the distribution net. In this distribution net, there are distribution protection units (DPUs), those which, apart from offering useful information, they are devices that under certain errors of the distribution net, trip and close switches, with the purpose of avoiding damages to devices of high economic cost.

Until the moment, the CCE has not connected the DPUs to the SCADA system, although they are very useful devices. The reason for the one that the DPUs are not connected to the SCADA system is because the communication protocol that they use is not compatible with the one used by the SCADA system, which uses a protocol Modbus RTU, while the DPUs use a protocol Modbus ASCII.

In this project a protocols changer card was modernized technologically, which allows the DPUs to be connected to the viewing net. This card settles inside each DPU, making that these devices work as any other slave in the SCADA system.

To modernize technologically the protocols changer card, first they were had to study all the characteristics, so much of hardware as of software, of the existent card, to examine the weak points and to look for them a better solution. The result was a more efficient card that the existent one, besides being much smaller, giving facilities to be installed inside the DPU.

Keywords: Protocols; RTU; SCADA system; ASCII; DPU; viewing net.

INDICE GENERAL

<i>CAPITULO 1 INTRODUCCION</i>	10
1.1. Descripción de la empresa	10
1.2. Definición del problema y su importancia	12
1.3. Objetivo General	17
1.4. Objetivos Específicos	18
<i>CAPITULO 2 ANTECEDENTES</i>	20
2.1. Estudio del problema a resolver	20
2.1.1. Protocolo Modbus	22
2.1.2. Unidad de protección de distribución (DPU)	30
2.2. Requerimientos de la empresa	33
2.3. Solución propuesta	34
<i>CAPITULO 3 PROCEDIMIENTO METODOLOGICO</i>	36
3. Metodología	36
3.1. Etapas correspondientes a los objetivos de documentación	36
3.2. Etapa de análisis de los comandos de la DPU	37
3.3. Etapa de estudio de la tarjeta convertidora de protocolos	37
3.4. Etapa de diseño del “firmware”	38
3.5. Etapa de diseño de hardware	39
3.6. Etapa de pruebas y análisis de resultados	40
3.7. Etapa de implementación del prototipo	40
3.8. Etapa de elaboración de la documentación	41
<i>CAPITULO 4 DESCRIPCION DEL HARDWARE UTILIZADO</i>	42
4.1. Diagrama general de bloques	42
4.2. Diagrama detallado de la tarjeta convertidora de protocolos	43
4.3. Diagrama esquemático de la tarjeta convertidora de protocolos	45
<i>CAPITULO 5 DESCRIPCION DEL SOFTWARE UTILIZADO</i>	47
5.1. Ambiente de desarrollo del software	47
5.2. Software desarrollado	48

Continua pag. siguiente...

CAPITULO 6 ANALISIS Y RESULTADOS	53
6.1. Explicación del Diseño	53
6.2. Alcances y Limitaciones	57
6.2.1. Hardware	57
6.2.2. Software	58
CAPITULO 7 CONCLUSIONES Y RECOMENDACIONES	59
7.1. Conclusiones	59
7.2. Recomendaciones	61
BIBLIOGRAFIA	62
APENDICE: Glosario	63
ANEXOS	65
Anexo 1: Características del MCU PIC16C74A	65
Anexo 2: Características eléctricas y ambientales del MCU	66
Anexo 3: Distribución de pines del PIC16C74A	67
Anexo 4: Características del decoder/demux 74LS139	68
Anexo 5: Datos generales del MUX 74LS153	69
Anexo 6: Características del MAX233	70

INDICE DE FIGURAS

Figura 1.1	Panel frontal de la DPU	13
Figura 1.2	Instalación actual de una DPU a la red de monitoreo	15
Figura 1.3	Instalación ideal entre la DPU y la red de monitoreo	16
Figura 1.4	Diagrama de bloques de una instalación ideal entre la DPU y el sistema SCADA	16
Figura 2.1	Esquema de la técnica Master – Slave	23
Figura 2.2	Framing de Modbus RTU	27
Figura 2.3	Secuencia de bits para el modo RTU	29
Figura 2.4	Panel posterior de la DPU	31
Figura 2.5	Diagrama de bloques general de la solución	34
Figura 4.1	Diagrama de bloques general del sistema	42
Figura 4.2	Diagrama de bloques	44
Figura 4.3	Diagrama esquemático	45
Figura 4.4	Circuito impreso de la tarjeta convertidora de protocolos	46
Figura 6.1	Diagrama de flujo General	55

INDICE DE TABLAS

Tabla 2.1	Características básicas de Modbus ASCII	25
Tabla 2.2	Características básicas de Modbus RTU	26
Tabla 5.1	Comandos de la DPU implementados en la tarjeta convertidora de protocolos	49
Tabla 5.2	Significado de los códigos de error	51

CAPITULO 1

INTRODUCCION

1.1. Descripción de la empresa

La CNFL S.A. se dedica a satisfacer las necesidades de energía en forma oportuna, eficiente y eficaz, en el área bajo su servicio. Además desarrolla proyectos de inversión, abasteciéndose del equipo necesario para afrontar el crecimiento de la demanda de servicio eléctrico.

La misión empresarial de la CNFL S.A. es: “contribuir al desarrollo económico y social del país mediante el suministro adecuado de la energía eléctrica en el área servida por la empresa, en busca tanto de la excelencia en la utilización de los recursos, como en la calidad del servicio al cliente”.

Esta empresa recoge una tradición de más de 100 años de actividad eléctrica en Costa Rica. El 9 de agosto de 1884 fue un día memorable para los josefinos de la época. Con la puesta en operación de la primera planta hidroeléctrica del país, situada en Barrio Aranjuez, se encendieron 25 luminarias en las principales vías públicas de la ciudad de San José. De esta forma, nuestra capital se constituyó en la tercera ciudad del mundo y la primera en Latinoamérica, en ser iluminada gracias a la energía eléctrica.

Su capital mayoritario, 98% de las acciones, fue comprado por el Instituto Costarricense de Electricidad, el ICE, el 30 de abril de 1968 a la Transnacional norteamericana Electric Bond and Share Co con lo cual se finiquitó su

nacionalización. Actualmente la empresa, tras 57 años de operación cuenta con 1300 empleados y 342 000 clientes (los usuarios se estima que superan el millón).

Descripción del Centro de Control de Energía

El Centro de Control de Energía (CCE) es el departamento donde se desarrolló el proyecto. Está dedicado al monitoreo y control de la red de distribución de energía eléctrica, la cual cuenta con un tendido de redes que cubren 934 Km² del gran Área Metropolitana (Alajuela, Heredia, Cartago y San José).

Bajo la dirección del CCE se encuentra el Despacho de Carga, en el cual se realiza el monitoreo de la red de distribución por medio de un sistema SCADA (Supervisory Control and Data Acquisition), es decir, un sistema de supervisión, control y adquisición de datos.

Este departamento se encarga de realizar proyectos que tienen que ver con la mencionada red, su mantenimiento y extensión. El CCE se encarga de agregar dispositivos adicionales a la red y de llevar la información desde los lugares donde dicha información es recopilada, hasta el Despacho de Carga, donde, por medio de pantallas y monitores relacionados con el sistema SCADA, los operadores observan el estado de los diferentes circuitos y realizan las operaciones de cierre y apertura de circuitos.

El CCE se ubica en La Uruca, contiguo a Automatra, donde laboran alrededor de 36 personas, entre ellas varios ingenieros, entre los cuales se encuentra el Ingeniero Eric Bogantes Cabezas, actual Jefe del Centro de Control de Energía.

1.2. Definición del problema y su importancia

En el Centro de Control de Energía (CCE) de la CNFL S.A. se utiliza un sistema SCADA con protocolo de comunicación Modbus RTU. Con este sistema se lleva a cabo el monitoreo de la red de distribución de energía eléctrica.

El SCADA está conectado a una serie de dispositivos, de los que toma la información necesaria para llevar a cabo el monitoreo de dicha red. Estos dispositivos por ende, deberían tener un protocolo de comunicación Modbus RTU, lo cual no es así en su totalidad ya que hay dispositivos como la DPU (Distribution Protection Unit) que no cuentan con dicho protocolo. Por este motivo, se hace necesaria una tarjeta convertidora de protocolos, para poder conectar estos dispositivos a la red de monitoreo.

La ventaja de conectar la DPU al sistema SCADA consiste en que es un aparato de vital importancia tanto en lo que respecta a seguridad como para proveer información al CCE.

En lo que concierne a seguridad, la DPU se encarga de sensar fallas en la red de distribución de energía y abrir circuitos en caso de que se dé alguna falla. La DPU es un instrumento muy importante ya que protege equipos de alto costo económico tales como transformadores, alimentadores, etc. Además la DPU evita que una falla se propague causando la salida de un sector más grande y perjudicando a más personas.

La DPU puede ser accesada manualmente por medio de un teclado alfanumérico, el cual se encuentra en la carátula frontal de la unidad, ver figura 1.1. También puede ser accesada remotamente por medio de un puerto serial, haciendo uso del protocolo Modbus ASCII.



Figura 1.1 Panel frontal de la DPU

En el caso de que se quiera operar la unidad, por ejemplo: conocer una corriente o cerrar un circuito, un operador debería ir hasta el lugar donde se encuentra la DPU e ingresar los comandos en el teclado alfanumérico. Con el sistema SCADA se evita ese inconveniente, pues permite que los operadores del Despacho de Carga operen el dispositivo a distancia y conozcan su estatus.

Si alguna de las protecciones de la DPU debe operar, dicha información debe ir al Centro de Control de Energía, de modo que se pueda saber de la falla en el Despacho de Carga. Además el Centro de Control requiere una gran variedad de datos del dispositivo, tales como corrientes, demandas pico, horas de eventos tales como la operación de un interruptor, etc.

Haciendo uso del puerto serie de la DPU y una adecuada tarjeta convertidora de protocolos, se puede conectar esta unidad al sistema SCADA, así desde la Estación Maestra se podrá tener control de la información y los comandos de la DPU. De este modo el monitoreo de la red de distribución de energía eléctrica será más eficiente, evitando de este modo, tener que ir hasta donde se encuentra la unidad para poder extraer la información o para operar dicho dispositivo.

Aunque la tarjeta convertidora de protocolos, con la cual cuenta la CNFL S.A., permite a la DPU conectarse al sistema SCADA, esta tiene una limitante tecnológica, la cual consiste en no tener las herramientas necesarias para poder leer los datos de la DPU o bien mandarle comandos desde la estación maestra.

Se puede ver que no se tienen ventajas al utilizar la tarjeta convertidora de protocolos, ya que siempre se tiene que ir donde se encuentra la DPU para obtener la información, o bien, para programarle los comandos que debe realizar. Debido a esta deficiencia tecnológica, la tarjeta convertidora de protocolos no se puede aplicar al campo.

Para obtener la mayor eficiencia de la DPU, es esencial actualizar tecnológicamente la tarjeta convertidora de protocolos, o bien adquirir una que cuente con las herramientas necesarias.

Actualmente para operar la DPU, se necesita hacer una instalación que aparte de costosa es poco eficiente ya que se tiene que utilizar una estación remota de poste, la cual tiene solo un canal de comunicaciones digitales. Por este motivo no es posible interrogar por medio de comunicación digital a través de dicha estación la información almacenada en la DPU.

También se tiene que realizar un cableado entre la estación remota de poste y la DPU el cual consta de muchos cables. Además con la forma actual de operar no se tiene acceso a todos los comandos e información de la DPU. La figura 1.2 muestra dicha instalación.

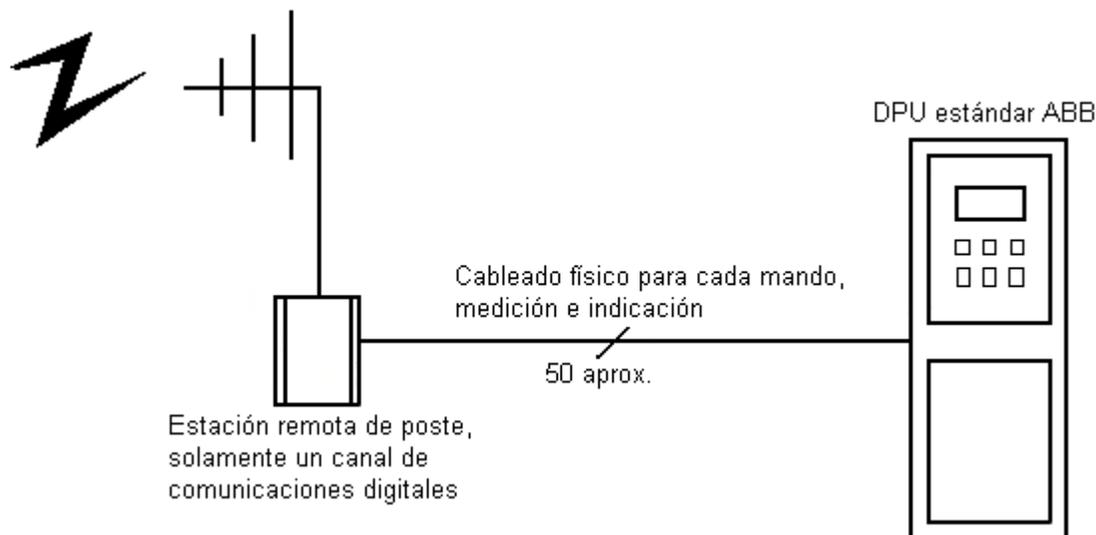


Figura 1.2 Instalación actual de una DPU a la red de monitoreo

Para mejorar la instalación que se muestra en la figura 1.2, se tiene que eliminar la estación remota (la cual tiene un costo de aproximadamente \$3000) y por ende, el cableado entre esta y la DPU. Para poder realizar dicha tarea, se tiene que utilizar la tarjeta convertidora de protocolos y un radio transmisor, los cuales se deben instalar directamente a la DPU, la figura 1.3 muestra como se realizaría la comunicación entre la DPU y la red de monitoreo del CCE.

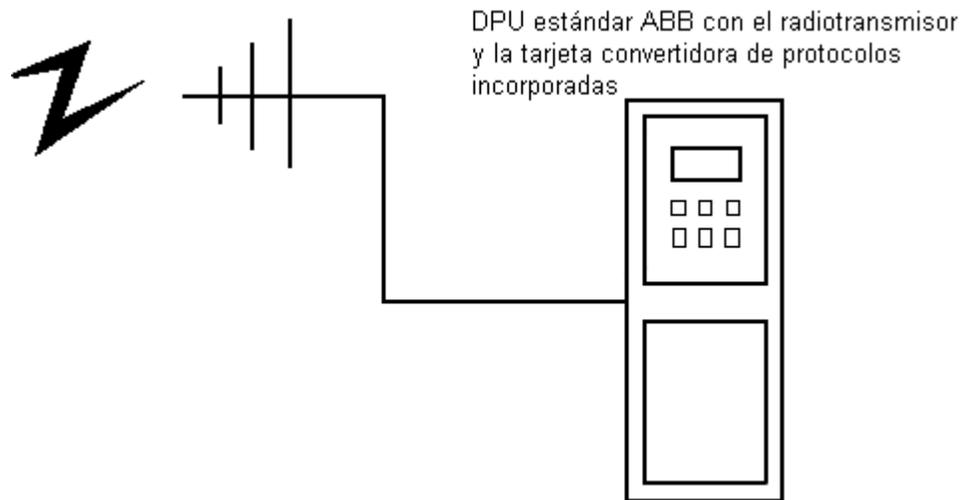


Figura 1.3 Instalación ideal entre la DPU y la red de monitoreo

La figura 1.4 es un diagrama de bloques que muestra con más detalle la instalación que se muestra en la figura 1.3. En este diagrama se evidencia la comunicación bidireccional entre el sistema SCADA y la DPU, también se muestra la RTU (Radio Terminal Unit) que es la unidad encargada de hacer la transmisión radial desde las subestaciones de distribución de energía, donde están colocadas las DPU's, al Despacho de Carga del Centro de Control de Energía, donde se encuentra el sistema SCADA.

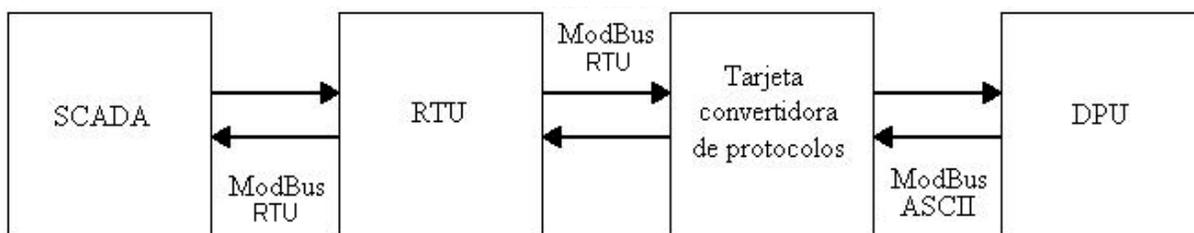


Figura 1.4 Diagrama de bloques de una instalación ideal entre la DPU y el sistema SCADA

Como se mencionó anteriormente con una instalación como la que se muestra en la figura 1.3 y en la figura 1.4 en forma de diagrama de bloques, se obtendría el máximo provecho de las características de la DPU.

1.3. Objetivo General

Se alcanzó, en el desarrollo de este proyecto, el siguiente objetivo general:

“Actualizar tecnológicamente la tarjeta convertidora de protocolos con el fin de hacer un sistema bidireccional capaz de permitirle al sistema SCADA enviar comandos o bien recibir información hacia y desde la DPU respectivamente para optimizar el monitoreo de la red de distribución de energía de la CNFL S.A.”

1.4. Objetivos Específicos

Se alcanzaron, en el desarrollo de este proyecto, los siguientes objetivos específicos:

- a.** Recopilar información de los diferentes protocolos involucrados en el proyecto.
- b.** Recopilar información del sistema SCADA utilizado en el CCE.
- c.** Investigar el funcionamiento de la Unidad de Protección de Distribución.
- d.** Determinar cuales comandos debe recibir la DPU de la red de monitoreo.
- e.** Determinar la información que la Estación Maestra necesita de la DPU.
- f.** Analizar las características de hardware del microcontrolador (PIC16C74A).
- g.** Investigar el set de instrucciones del microcontrolador (PIC16C74A).
- h.** Examinar el funcionamiento de la tarjeta convertidora de protocolos.
- i.** Probar la tarjeta convertidora de protocolos.
- j.** Analizar por completo el "firmware" de la tarjeta convertidora de protocolos.
- k.** Diseñar el "firmware" para que la DPU pueda recibir comandos de la Estación Maestra.
- l.** Diseñar el "firmware" para que la Estación Maestra pueda recibir información de la DPU.
- m.** Rediseñar el hardware de la tarjeta convertidora de protocolos.
- n.** Analizar si se puede optimizar el hardware de la tarjeta.

- o.** Planificar las pruebas a las nuevas características diseñadas.
- p.** Probar las nuevas características diseñadas.
- q.** Analizar los resultados de las pruebas.
- r.** Implementar un prototipo del sistema.
- s.** Confeccionar la documentación final del proyecto.
- t.** Realizar la transferencia de tecnología en el Centro de Control de Energía.

CAPITULO 2

ANTECEDENTES

2.1. Estudio del problema a resolver

Como se mencionó en el capítulo 1, el CCE cuenta con una tarjeta convertidora de protocolos, la cual le permite conectar la DPU al sistema SCADA, pero esta tarjeta no tiene las herramientas necesarias para poder operar a la DPU eficientemente, por lo que se tiene que seguir operando la DPU manualmente.

La tarjeta convertidora de protocolos, está compuesta principalmente por un microcontrolador (PIC16C74A). El firmware de este PIC, es el encargado de hacer la tarea de conversión de protocolos, esta conversión se lleva a cabo en varios pasos, los cuales se muestran a continuación:

- Detectar y comparar la dirección del dispositivo en el mensaje.
- Chequear el CRC (chequeo de redundancia cíclica) del mensaje.
- Decodificar el comando que se desea ejecutar.
- Ejecutar el comando.
- Enviar la respuesta de la DPU a la red de monitoreo.

La tarjeta convertidora de protocolos, con la cual cuenta el CCE, sólo realiza los dos primeros pasos en su totalidad, pero los demás pasos no los ejecuta correctamente, por lo que la ejecución de comandos no es nada eficiente, de hecho, de los 16 comandos más importantes que debería ejecutar la DPU, esta tarjeta puede operar a sólo dos de ellos.

La principal causa por la que la tarjeta convertidora de protocolos no opera correctamente, es que en la etapa de ejecución de comandos, se deben programar todos los comandos que la DPU debe ejecutar, sin embargo sólo se programaron dos de ellos. La programación de esta etapa es una de las más importantes, ya que es en ésta, junto con la última etapa (enviar la respuesta a la red de monitoreo), donde se hace la conversión de protocolos.

La conversión de protocolos en el paso de ejecución de comandos, es específica para cada comando de la DPU, por lo que debe haber un procedimiento especial para cada comando, lo cual no se da en esta tarjeta, donde sólo existen dos procedimientos programados, de ahí que únicamente se puedan ejecutar dos comandos.

La etapa de enviar la respuesta a la red de monitoreo está estrechamente unida a la etapa anterior, donde, por cada procedimiento programado para ejecutar los comandos, se debe programar un procedimiento de respuesta en el último de los pasos de conversión de protocolos, por lo que en esta etapa, también sólo hay dos procedimientos programados.

Ahora que se tiene un conocimiento más amplio del problema que se da en la tarjeta convertidora de protocolos, se presentan a continuación detalles del protocolo Modbus RTU, aplicado en el sistema SCADA y del protocolo Modbus ASCII aplicado en la DPU, también se verán las características de la DPU y de cómo se puede comunicar serialmente.

2.1.1. Protocolo Modbus

2.1.1.1. Introducción al Protocolo Modbus

Los controladores programables Modicon pueden comunicarse entre sí y con otros dispositivos sobre una gran variedad de redes. Las redes que soportan incluyen redes industriales Modbus Modicon y Modbus Plus y redes standard como MAP y Ethernet. Estas redes son accesibles por puertos construidos en los controladores o por adaptadores de red, módulos opcionales y puertas, disponibles en el mercado.

Para fabricantes de equipo original (OEM)¹, los programas de Modicon ModConnect, están disponibles para integrar redes como Modbus en diseño de productos propietarios.

El protocolo Modbus define una estructura de mensaje, que los dispositivos reconocerán y usarán sin importar el tipo de red sobre la cual se estén comunicando. Esta estructura describe el proceso con el cual el dispositivo accesa a otros, cómo responde ante otros dispositivos y cómo los errores se chequean y reportan. Establece un formato común del mensaje y el contenido de los campos y provee una estructura que determina cómo cada dispositivo conocerá su dirección, determinará la acción a tomar y extraerá datos u otra información contenida en el mensaje. Si se requiere una respuesta, ésta se construirá en protocolo Modbus.

¹ Original Equipment Manufacturer (Fabricante de Equipo Original)

2.1.1.2. Transacciones en redes Modbus

Los puertos Standard Modbus usan una interfaz serial RS-232 que define las distribuciones de pines de los conectores, el cableado, nivel de señales, tasa de baudios y chequeo de paridad. Los dispositivos pueden ser puestos en red por medio de modems. Estos usan una técnica maestro - esclavo en la cual sólo un dispositivo (el maestro) puede iniciar transacciones. Los otros dispositivos (esclavos) responden sufriendo los datos requeridos o ejecutando las acciones solicitadas.

El maestro puede direccionar esclavos individuales o iniciar un mensaje transmitido a todos ellos lo cual se denomina modo "broadcast" o transmisión pública. Las respuestas de los esclavos a los direccionamientos individuales se dan de una manera específica. Para los mensajes en modo "broadcast" no hay respuestas por parte de los esclavos.

El protocolo establece el formato para la solicitud del maestro poniendo en ella la dirección del esclavo, el código de la función, cualquier dato a ser enviado y un campo de chequeo de error. La respuesta del esclavo está compuesta en Modbus y contiene campos que confirman la función solicitada, los datos y el chequeo de error. Si ocurre un error recibiendo un mensaje, el esclavo construye un mensaje de error que se envía como respuesta. Esto se aprecia en la figura 2.1.



Figura 2.1 Esquema de la técnica Master – Slave

2.1.1.3. La solicitud (Query)

El código de función en la solicitud le dice al dispositivo direccionado, qué tipo de acción tomar. Los bits de datos contienen la información adicional que el esclavo necesite. El campo de error hace fácil revisar la integridad de los datos.

2.1.1.4. La respuesta (Response)

Si el esclavo brinda una respuesta normal, el código de función de las respuestas es un eco del código de operación de la solicitud.

Los bytes de datos contienen los datos recolectados por el esclavo, como valores de registros o estados. Si un error ocurre, la respuesta cambia para reflejar que ha ocurrido un error. En caso de error, el código de operación de la respuesta tiene el MSB (bit más significativo) como 1 en vez de cero como es normal, de modo que el código con error es 80 hex mayor que el código sin error. El chequeo de error le permite al maestro confirmar que el contenido del mensaje fue válido.

2.1.1.5. Modos de transmisión serial

Los controladores pueden configurarse para trabajar en dos modos de transmisión dentro del protocolo Modbus, ASCII o RTU (Radio Terminal Unit). Los usuarios configuran la tasa de baudios, la paridad etc., y todos en la red deben tener los mismos parámetros.

El modo que se ha utilizado al realizar el proyecto y que es el que toda la red comparte, es Modbus RTU.

a) Modo ASCII

Cuando los dispositivos se comunican en Modbus ASCII (American Standard Code for Information Interchange), cada byte en un mensaje se envía como dos caracteres ASCII. La principal ventaja de este modo es que permite intervalos de hasta 1 segundo sin que haya time out (vencimiento de tiempo).

Tabla 2.1 Características básicas de Modbus ASCII

Sistema de Código
Hexadecimal, Caracteres ASCII 0 ... 9, A ... F
Un carácter hexadecimal en cada carácter ASCII
Bits por Byte
1 start bit
7 de datos, LSB se envía de primero
1 bit para even / odd parity – ningún bit si no hay paridad
1 stop bit si hay paridad – 2 bits si no hay paridad
Campo de Chequeo de Error
“Longitudinal Redundancy Check” (LRC)

b) Modo RTU

Cuando se usa el modo RTU, cada byte contiene dos caracteres hexadecimales, de modo que se tiene mejor rendimiento de entrada - salida a la misma tasa de baudios que en Modbus ASCII, es decir, que es más denso.

Tabla 2.2 Características básicas de Modbus RTU

Sistema de Código
Hexadecimal, Caracteres ASCII 0 ... 9, A ... F
Dos caracteres hexadecimales en cada carácter ASCII
Bits por Byte
1 start bit
8 de datos, LSB se envía de primero
1 bit para even / odd parity – ningún bit si no hay paridad
1 stop bit si hay paridad – 2 bits si no hay paridad
Campo de Chequeo de Error
“Cyclical Redundancy Check” (CRC)

2.1.1.6. Enmarcado del paquete Modbus

En el modo RTU, un mensaje empieza después de un silencio de al menos 3.5 caracteres, lo cual se implementa como 4 tiempos de carácter a la tasa de baudios empleada. En este modo los dispositivos en red monitorean el bus continuamente inclusive en los intervalos de silencio. Un nuevo mensaje como ya se dijo, empieza después de este intervalo de 3.5 caracteres.

El mensaje o “frame” debe ser enviado como un flujo continuo de datos, si un intervalo de 3.5 caracteres o más, se produce antes de que el “frame” se complete, se elimina el mensaje recibido y se espera que el nuevo carácter que se reciba sea la dirección de un nuevo mensaje.

Si un carácter llega antes de un intervalo de 3.5 caracteres después de que se terminó de recibir el frame, se toma como una continuación del mensaje anterior. Esto dará un error al momento de chequear el CRC. Un mensaje típico en RTU se ve a continuación en la Figura 2.2. De este punto en adelante el análisis se realiza en torno al modo RTU.

INICIO	DIR	OP COD	DATOS	CHEQ	FIN
T1_T4	8 BIT	8 BIT	nX8 BIT	16 BIT	T1_T4

Figura 2.2 Framing de Modbus RTU

2.1.1.7. El chequeo de error CRC¹

Este campo de 16 bits es el resultado del cálculo de un Chequeo de Redundancia Cíclica. Este se usa para rellenar los últimos dos bytes de un mensaje, pero se rellena con el byte menos significativo en primer lugar.

¹ Siglas en ingles de Cyclical Redundancy Check

El chequeo de error CRC, se aplica sin importar la paridad que se use, o si no se está usando alguna. El CRC se calcula en el dispositivo que transmite y es recalculado y comparado en el dispositivo que recibe. Este valor está compuesto por dos bytes que se calculan de la siguiente forma:

Paso 1 Se carga un registro de 16 bits (llamado CRC) con unos.

Paso 2 Se hace XOR del primer byte del mensaje almacenado en un registro temporal con el byte menos significativo del registro CRC, poniendo el resultado en un registro de trabajo.

Paso 3 Se desplaza el registro CRC hacia la derecha, o sea hacia el LSB y se llena con un cero el MSB.

Paso 4 Se hace una AND entre el registro de trabajo y el valor 01 hex.

Paso 5 Si el resultado de la operación AND es 0 hex, se hace una XOR entre el registro CRC y el valor A001 hex, y el resultado se guarda en el registro CRC. Si el resultado de la operación AND es 1 hex se hace el paso 6.

Paso 6 Se desplaza a la derecha el registro temporal el cual contiene el dato del mensaje.

Paso 7 Se repiten los pasos del 2 al 6 hasta que el registro temporal se haya desplazado 8 veces a la derecha.

Paso 8 Se repiten los pasos del 2 al 7 para el próximo byte y todos los subsecuentes bytes del campo de datos.

Resultado El contenido del registro es el valor CRC.

Paso 9 Cuando se ponen el CRC en el mensaje se debe hacer con el byte menos significativo de primero, por ejemplo si el valor CRC es 1241 hex (0001 0010 0100 0001), se envía 4112 hex.

2.1.1.8. Como se transmiten serialmente los datos

En todo mensaje, cada byte se envía con el LSB (bit menos significativo) de primero y el MSB (bit más significativo) de último, a continuación puede verse esta secuencia en la figura 2.3

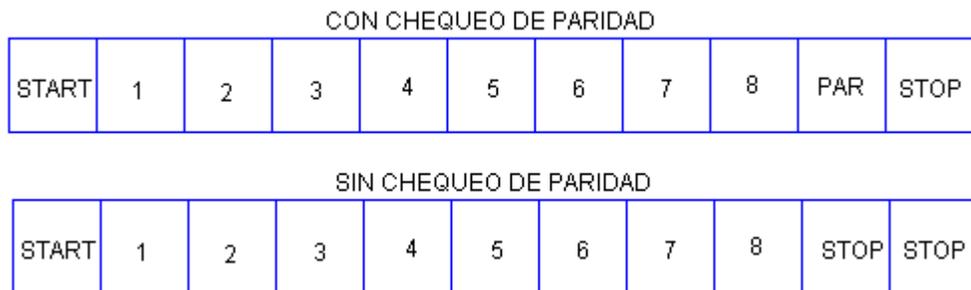


Figura 2.3 Secuencia de bits para el modo RTU

2.1.2. Unidad de protección de distribución (DPU)

2.1.2.1. Descripción general

La unidad de protección de distribución DPU fabricada por Asea Brown Boveri o ABB, está capacitada para dar protección trifásica de un alimentador.

El microprocesador interno brinda las siguientes funciones:

- a.** Protección trifásica contra sobre corrientes (en un intervalo o instantánea).
- b.** Protección de tierra contra sobre corrientes (en un intervalo o instantánea).
- c.** Recierre de multidisparo.
- d.** Amperímetro, amperímetros de picos y amperímetro de picos de demanda.
- e.** Grabación de eventos.
- f.** Auto chequeo continuo.
- g.** Puerto de comunicaciones RS-232C, para conexión con terminal remota.

La función definida por el punto g, fue la que se utilizó para lograr que la DPU se comunicara con la red de monitoreo. Con la tarjeta convertidora de protocolos como puente se logró que la DPU recibiera comandos del sistema SCADA a través del puerto RS-232C, de igual manera, la DPU fue capaz de enviar la información solicitada a la red de monitoreo.

2.1.2.2. Puerto de comunicaciones de la DPU

Tal y como se muestra en la figura 2.4, este puerto se ubica en el panel trasero de la unidad.

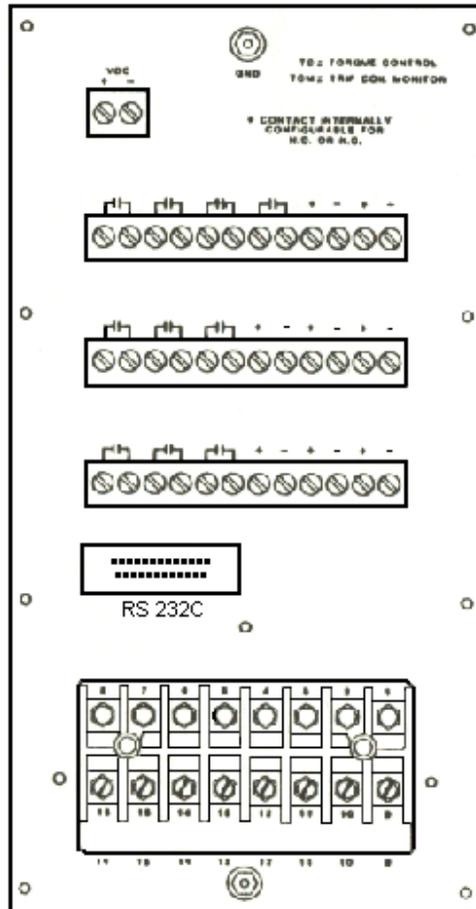


Figura 2.4 Panel posterior de la DPU

La disposición de este puerto es RS-232C, la cual establece el cableado que se debe implementar para comunicar la unidad con una terminal de datos. El conector es macho dado que el equipo es DTE (Data Terminal Equipment).

2.1.2.3. Tasa de Baudios, formato de datos y protocolo

La DPU soporta una tasa de baudios que va desde 50 hasta 9600 baudios pero este parámetro sólo se edita desde el teclado del panel frontal. La transmisión se realiza por medio de un formato de datos de 8 bits, sin paridad y dos bits de paro.

Los comandos que se envían a la DPU deben terminar con un 0Dh y un 0Ah, es decir un retorno de carro y una alimentación de línea respectivamente.

Cuando la DPU envía información a la terminal remota, termina sus respuestas con estos mismos caracteres. Antes de responder un comando, la DPU envía a la terminal un eco de cada carácter que se le transmite.

El dispositivo posee un set de instrucciones más dirigido al usuario final, por lo cual los caracteres inútiles para el protocolo Modbus deben eliminarse de las respuestas y los datos valiosos, convertirse a un formato útil para el almacenamiento, ya que la unidad devuelve los valores como caracteres ASCII con los cuales no se pueden realizar cálculos directos y además ocupan más espacio.

2.2. Requerimientos de la empresa

El Centro de Control de Energía, cuenta con dispositivos de protección llamados DPU, los cuales no se pueden integrar a la red de monitorio por incompatibilidad de protocolos, por ende, la empresa requiere de un dispositivo capaz de acoplar estas unidades de protección a la red de monitoreo.

Para utilizar las DPUs en el sistema SCADA, el Centro de Control de Energía cuenta con una tarjeta convertidora de protocolos, la cual se tuvo que actualizar para lograr que su rendimiento fuera el óptimo, de aquí que la empresa abordara este proyecto con varias ideas en mente, las cuales se definieron claramente con el ingeniero a cargo.

El proyecto final brinda las siguientes funciones:

- a.** Permite que la DPU, por medio de la tarjeta convertidora de protocolos, ejecute los comandos más utilizados, tomando en cuenta que estos comandos serán enviados desde la red de monitoreo.
- b.** Permite que las respuestas de la DPU, cuando las hay, se envíen a la red de monitoreo en protocolo Modbus RTU.
- c.** Se redujo el tamaño del circuito impreso de la tarjeta convertidora de protocolos, por lo que ésta puede ser instalada dentro de la DPU.
- d.** Permite que la tarjeta convertidora de protocolos, se pueda alimentar con el voltaje que le suministra la DPU, el cual es de 24 V en corriente directa.

2.3. Solución propuesta

Se trabajó con la tarjeta convertidora de protocolos, la cual es un dispositivo bidireccional de comunicaciones, la cual funciona como puente para acoplar un dispositivo que utiliza protocolo Modbus RTU con otro que utiliza protocolo Modbus ASCII.

Como se muestra en la figura 2.5, la red de monitoreo utiliza un protocolo de comunicaciones Modbus RTU, mientras que la DPU hace uso del protocolo Modbus ASCII, de esta manera, la tarjeta convertidora de protocolos hace posible que la DPU se pueda comunicar con la red de monitoreo.

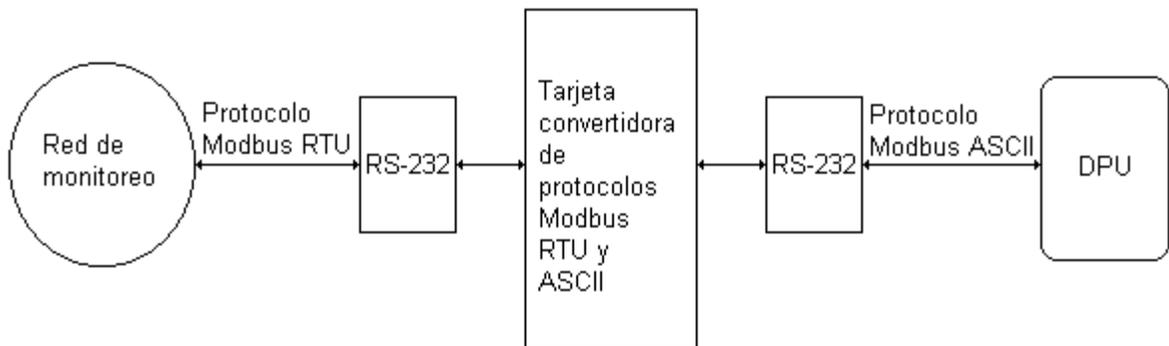


Figura 2.5 Diagrama de bloques general de la solución

Debido a que la tarjeta convertidora de protocolos contaba con un firmware muy escueto, se agregó a dicho firmware más características que le permiten a la tarjeta convertidora de protocolos, decodificar los comandos de la DPU y poder ejecutarlos.

La deficiencia con la cual contaba la tarjeta convertidora de protocolos era que el firmware del microcontrolador no tenía procedimientos que decodificaran los comandos de la DPU.

Debido a que la DPU tiene demasiados comandos, de los cuales la mayoría no se utilizan, se determinaron cuáles eran los comandos de mayor importancia para que la DPU cumpliera con todas las funciones requeridas por el Centro de Control de Energía.

Los comandos implementados en la tarjeta convertidora de protocolos, se diseñaron para que su tiempo de respuesta fuese el mínimo posible, ya que la tarjeta debe ser lo más transparente posible para la red de monitoreo, para que de esta manera, la DPU pueda trabajar como un dispositivo que está directamente conectado a la red.

No sólo se trabajó con la decodificación de los comandos, sino que se tuvo que lograr que las respuestas de la DPU, que son en protocolo ASCII, fuesen vistas por la red de monitoreo en protocolo RTU.

CAPITULO 3

PROCEDIMIENTO METODOLOGICO

3. Metodología

Para alcanzar los objetivos que se mencionaron anteriormente, se siguieron los siguientes pasos metodológicos.

3.1. Etapas correspondientes a los objetivos de documentación

Durante estas etapas se recopiló una documentación adecuada sobre los protocolos que se enlazaron, los cuales son Modbus RTU y Modbus ASCII. También se buscó toda la información con respecto al PIC16C74A, tanto en lo concerniente a la arquitectura como en el set de instrucciones. Esto se realizó sobre la base de los manuales que la empresa facilita, además del acceso a CD ROM de información y a Internet que brinda también la compañía.

En cuanto a la información necesaria para utilizar la DPU, se consultó el manual de usuario de la DPU de ABB, de donde se tomaron todos los datos de los comandos que se implementaron en este proyecto.

En el proceso de información y documentación se determinaron los elementos que intervienen en la conversión de protocolos, de este modo, se definió el “firmware” (microprogramación) requerido. En estas etapas todo salió acorde a lo esperado y no se tuvieron atrasos según lo programado.

3.2. Etapa de análisis de los comandos de la DPU

En esta etapa se definió, con base en reuniones con el Ing. Eric Bogantes, cuáles eran los comandos más importantes para el buen funcionamiento de la DPU. No se seleccionaron todos los comandos de la DPU ya que muchos de ellos son innecesarios para que la DPU tenga una excelente eficiencia.

Con el conocimiento adquirido sobre el funcionamiento de la DPU, llevar a cabo estas etapas no produjo ningún atraso según lo establecido.

3.3. Etapa de estudio de la tarjeta convertidora de protocolos

En esta fase se llevaron a cabo varias pruebas a la tarjeta convertidora de protocolos para conocer y probar su funcionamiento. También se estudió el firmware del microcontrolador de la tarjeta para comprender a fondo todos los detalles de la tarjeta convertidora de protocolos.

En este punto se tuvieron muchos inconvenientes, por lo que se atrasó el proyecto en por lo menos tres semanas. Estos inconvenientes radicaron en un circuito impreso defectuoso y un firmware mal documentado.

Con respecto al circuito impreso de la tarjeta convertidora de protocolos, con la que contaba el CCE, se descubrió que era de mala calidad, ya que tenía varios cortos en sus pistas, los cuales se tuvieron que detectar y eliminar.

Otro problema con el que se tuvo que lidiar, fue causado por la mala documentación e información que se tenía de la tarjeta convertidora de protocolos. A la hora de estudiar el firmware del microcontrolador de la tarjeta, resultó que no era el firmware correcto, por lo que surgieron muchas incongruencias. Esto generó una pérdida de tiempo ya que se invirtió como semana y media estudiando el firmware equivocado.

El último inconveniente que se presentó en el proyecto, se dio a la hora de buscar el firmware correcto del microcontrolador. Se tenía un disco compacto con la información y documentación de la tarjeta convertidora de protocolos, sin embargo todo estaba desordenado y habían como 50 versiones de lo que podía ser el firmware correcto.

Se contactó al Ing. Alberto Fallas, quien fue el diseñador de la tarjeta convertidora de protocolos en su primera fase, para determinar cual era el firmware que se debía estudiar. Cuando se tuvo el firmware correcto, se analizó y estudió para conocer su funcionamiento y de esa manera actualizarlo tecnológicamente.

3.4. Etapa de diseño del “firmware”

En esta fase, se realizó el trabajo de diseño de algoritmos necesarios y se implementaron en el microcontrolador con el uso del lenguaje ensamblador MPLAB. Aquí se hicieron las pruebas tradicionales de programación: verificación y depuración (debuging) con el fin de obtener un firmware rígido y eficiente.

Esta etapa consistió en el desarrollo de firmware experimental, para probar los comandos de la DPU, luego de haber probado los comandos, se diseñó y programó el firmware necesario para que la tarjeta convertidora de protocolo realizara operaciones de decodificación de comandos y conversión de protocolos de ASCII a RTU y viceversa.

Entre mayor sea el número de comandos que decodifique la tarjeta convertidora de protocolos mejor será el rendimiento de la DPU, sin embargo fue innecesario decodificar todos los comandos ya que no todos se utilizan, por lo que se diseñó un decodificador de los comandos más utilizados de la DPU.

3.5. Etapa de diseño de hardware

En esta fase, la cual se realizó en paralelo con la fase anterior, se determinó la necesidad de implementar nuevo hardware a la tarjeta convertidora de protocolos, o bien, si el hardware existente se podía optimizar.

Debido a la necesidad de instalar la tarjeta convertidora de protocolos dentro de la DPU, se necesitó que la alimentación de corriente de esta tarjeta se acoplara al voltaje que proporciona la DPU, el cual es de 24 V. Debido a que en la tarjeta convertidora de protocolos hay componentes TTL, se implementó un regulador de voltaje para bajar el voltaje de la DPU (24 V) a un nivel TTL (5 V).

3.6. Etapa de pruebas y análisis de resultados

En esta fase, una vez terminada la programación del firmware, se planificaron pruebas para comprobar el funcionamiento de la tarjeta convertidora de protocolos actualizada tecnológicamente. Dichas pruebas se llevaron a cabo y se analizaron los resultados para comprobar el funcionamiento del sistema.

Las pruebas que se llevaron a cabo junto con el análisis de los resultados de dichas pruebas, comprobaron que el nuevo diseño implementado funciona según lo esperado.

3.7. Etapa de implementación del prototipo

En esta fase, después de haber rediseñado el hardware del sistema, y haber comprobado que el firmware diseñado funciona correctamente, se procedió a implementar un circuito impreso.

Este nuevo circuito impreso tiene mejoras que le permite a la tarjeta convertidora de protocolos ser instalada dentro de la DPU, ya que se logró disminuir el tamaño del circuito impreso, además de poder ser alimentada con 24 V.

3.8. Etapa de elaboración de la documentación

En esta fase, se desarrolló toda la documentación formal del proyecto, tal como la elaboración del informe final del proyecto. También se planeó la exposición para la presentación del proyecto, en la cual se debe dar la transferencia de tecnología, para que se le pueda dar un buen uso a la tarjeta convertidora de protocolos.

CAPITULO 4

DESCRIPCION DEL HARDWARE UTILIZADO

4.1. Diagrama general de bloques

De la figura 4.1, se puede ver que en sí, la tarjeta convertidora de protocolos consta de dos módulos, el primero de ellos es el módulo de conversión y decodificación. Este módulo está formado por el microcontrolador PIC16C74A, cuyo firmware tiene procedimientos que se encargan de convertir los protocolos de ASCII a RTU y viceversa. También tiene procedimientos de decodificación de los comandos de la DPU.

El segundo módulo es el del puerto serie, el cual está formado por dos puertos de comunicación serial, uno para conectar la tarjeta a la red de monitoreo y el otro para conectarla a la DPU, más adelante se explicarán con más detalle cada módulo.

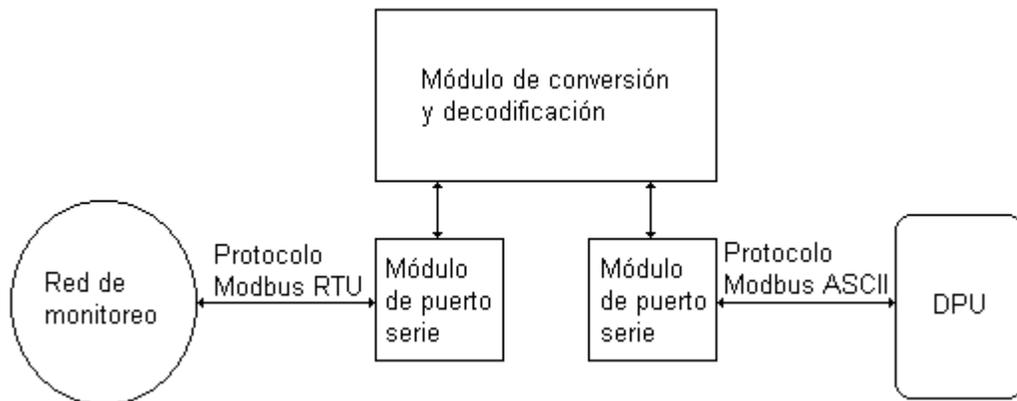


Figura 4.1 Diagrama de bloques general del sistema

4.2. Diagrama detallado de la tarjeta convertidora de protocolos

En la figura 4.2 se presentan los dos módulos que conforman al sistema. El módulo de conversión y decodificación está representado por el CPU (microcontrolador PIC16C74A), mientras que los módulos de puerto serie están representados en la figura como el decodificador de puertos serie, el cual es un arreglo de componentes TTL, entre los cuales se tienen un 74LS139, un 74LS153 y un MAX233.

El microcontrolador utilizado fue seleccionado por tener una USART interna, la cual hace posible la comunicación serie, entre otras características adicionales. A continuación se explican en detalle cada una de las conexiones utilizadas por el microcontrolador:

RESET: Esta señal está controlada por un botón pulsador y produce el reinicio del sistema.

CLOCK: es una señal que proviene de un cristal de cuarzo, el cual oscila a 12Mhz y por medio de la cual se temporiza todo el proceso interno del MCU.

RX: es el pin de entrada de datos a la USART, por la cual entran al microcontrolador todos los datos de ambos puertos (puerto 0 y puerto 1).

TX: es el pin de salida de datos de la USART, por la cual salen del microcontrolador todos los datos de ambos puertos (puerto 0 y puerto 1).

SP: es un pin de salida que controla el decodificador de puertos, por medio de este pin se selecciona cuál de los puertos se desea utilizar, si esta salida tiene un nivel TTL de 5 V, se selecciona el puerto serie que se conecta a la red de monitoreo, mientras que si la salida tiene un nivel TTL de 0 V, se selecciona el puerto que se comunica con la DPU.

Como se mencionó anteriormente, el bloque de decodificador de puertos es un arreglo de componentes TTL, entre los que se tienen un multiplexor (74LS153), un decodificador (74LS139) y un driver de comunicación serie RS-233 (MAX233). Este bloque selecciona uno de los dos puertos con los que cuenta la tarjeta convertidora de protocolos, respondiendo a la programación del microcontrolador.

Debido a que el PIC16C74A tiene una única USART, el bloque de decodificador de puertos serie es esencial para seleccionar uno de los dos puertos serie disponibles, sacando ventaja de que nunca se tienen que utilizar ambos puertos a la vez. Las conexiones de este bloque se explican a continuación:

SP: es un pin de entrada que selecciona el puerto que se utilizará.

RX0: es el pin de entrada de datos del puerto serie al que se conecta la DPU.

RX1: es el pin de entrada de datos del puerto serie al que se conecta la red de monitoreo.

TX0: es el pin de salida de datos del puerto serie al que se conecta la DPU.

TX1: es el pin de salida de datos del puerto serie al que se conecta la red de monitoreo.

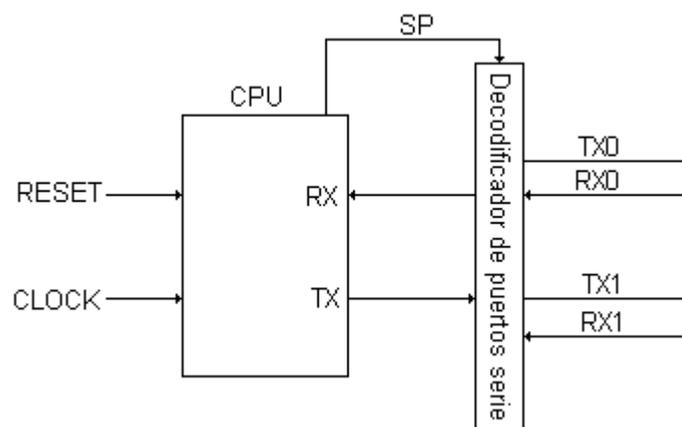


Figura 4.2 Diagrama de bloques

En la figura 4.4 se muestra el circuito impreso de la tarjeta convertidora de protocolos, ésta se diseñó de manera que su tamaño fuera el menor posible, por lo que se implementó un impreso con pistas en ambos lados de la tarjeta. En la figura, las pistas de color azul son las de la cara inferior y las de color rojo son las de la cara superior.

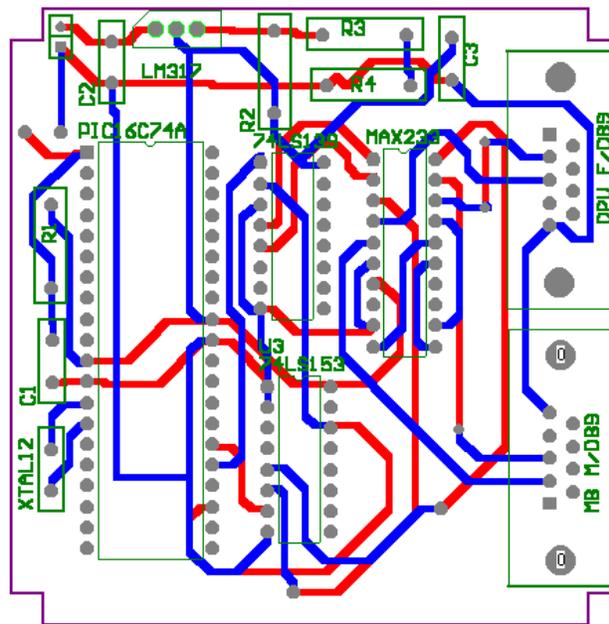


Figura 4.4 Circuito impreso de la tarjeta convertidora de protocolos

CAPITULO 5

DESCRIPCION DEL SOFTWARE UTILIZADO

5.1. Ambiente de desarrollo del software

El ambiente de desarrollo en que se trabajó al implementar el “software” del sistema fue el MPLAB-IDE, (IDE significa Ambiente Integrado de Desarrollo), el cual se puede bajar gratuitamente en Internet en la página web de Microchip. Las simulaciones de “software” se realizaron con el mismo MPLAB-IDE, el cual tiene un simulador integrado dentro de sus herramientas. Para programar el PIC se utilizó el “software” propio del programador EPIC.

Los diagramas del “hardware” se realizaron por medio de ORCAD EXPRESS 7.0. El circuito impreso se desarrolló con Protel 99. La etapa de pruebas necesitó de un “software” especial, que sin embargo, se consiguió de forma gratuita, este es el MODSCAN 32, con el cual se pudo simular fielmente la operación de un maestro Modbus, de modo que el módulo desarrollado se pudo probar adecuadamente. Para realizar las pruebas más tempranas sobre la DPU, es decir, para entender su funcionamiento, se utilizó la HIPERTERMINAL de Microsoft Windows.

5.2. Software desarrollado

La implementación del software que se desarrolló, permite a la DPU ejecutar sus comandos. Un comando totalmente terminado es aquel que se envía desde la red de monitoreo, la tarjeta convertidora de protocolos lo detecta, lo decodifica, convierte el protocolo de RTU a ASCII y lo envía a la DPU para su ejecución, luego recibe la respuesta de la DPU, convierte el protocolo de ASCII a RTU y manda la respuesta a la red de monitoreo.

La segunda acción (decodificación del comando) es de suma importancia, ya que con este procedimiento, la tarjeta convertidora de protocolos determina cual es el comando que se desea ejecutar, o bien, determinar si el comando no existe. La parte de detectar el comando desde la red de monitoreo es lo que hacía la tarjeta convertidora de protocolos en su fase inicial.

En las etapas de pruebas, para conocer el funcionamiento de los comandos de la DPU y probar el hardware existente, se implementó un firmware para que la tarjeta convertidora de protocolos ejecutara una serie de comandos de la DPU y la información que dichos comandos proporcionan, visualizarla en la PC, haciendo uso del programa Hyper Terminal de Microsoft Windows.

Se diferencian dos tipos de comandos, los que hacen una acción en la DPU y los que mandan información de la DPU a la red de monitoreo como respuesta del comando.

Los comandos que realizan una acción en la DPU son: Trip the circuit breaker (8747), Close the circuit breaker (7478) y Clear all peak demand current (333). Los otros comandos nutren de información a la red de monitoreo y estos son: Display circuit breaker status (B0), display/edit alternate minimum trip mode (B2), display/edit lockout on 51N trip (A9), display/edit breaker operation counter (70), display/edit overcurrent-trip counter (71) y 8 comandos más que despliegan diferentes tipos de corrientes.

En la tabla 5.1 se tienen los comandos implementados, sus respectivos valores de función, con el cual puede ser activado el comando, y los diferentes códigos de error que se pueden generar.

Tabla 5.1 Comandos de la DPU implementados en la tarjeta convertidora de protocolos

Comando	Función (HEX)	Código de error (HEX)
A9	01	02,04
B2	02	02,04
B0	03	03
70	04	04
71	05	04
D1	06	03
D2	07	03
D3	08	03
D0	09	03
41	0A	03
42	0B	03
43	0C	03
40	0D	03
333	0E	03
8747	0F	01,03
7478	10	01,03
No implementado	Si es igual a 00 o mayor a 10	00

Un ejemplo de ejecución de un comando se presenta a continuación.

Se desea ejecutar el comando 70, este es un comando de desplegar / editar, lo que significa que si se desea, sólo se puede ver el valor del contador de operaciones, o bien, se puede editar este valor. Para el caso de desplegar el valor, el formato del mensaje que debe enviar la red de monitoreo a la DPU es como se muestra a continuación:

Dirección del esclavo	Función	CRC (LSB)	CRC (MSB)
15 hex	04 hex	0E hex	E3 hex

Si el comando es correcto, y la DPU no detectó ningún error, la respuesta puede ser como se muestra a continuación:

Dirección del esclavo	Función	Datos	Datos	CRC (LSB)	CRC (MSB)
15 hex	04 hex	05 hex	03 hex	06 hex	B8 hex

En la respuesta generada por la DPU para la red de monitoreo, debe aparecer la identificación del esclavo, el valor de la función, los datos del conteo (información) y el código de error. En este caso el valor del conteo se lee como 53, o sea, se han dado 53 operaciones del interruptor.

Si un comando presenta un error, se puede identificar fácilmente, ya que el valor de la función tendrá un valor de 80 hex mayor a la función solicitada, o sea, si se quiere ejecutar la función 05 hex y se da un error, en la respuesta, en el campo de la función tendremos un 85 hex, lo cual quiere decir que hubo un error.

Para identificar qué tipo de error se está dando, se hace uso de la tabla 5.2, donde comparando el valor del campo de datos de la respuesta con esta tabla, se sabrá cual fue la causa que generó el error.

Tabla 5.2 Significado de los códigos de error

Código de error (HEX)	Causa del error
00	El comando no existe.
01	Si aplicó el comando 8747, significa que el interruptor ya está abierto. Si aplicó el comando 4787, significa que el interruptor ya está cerrado.
02	El valor que desea editar es incorrecto, el valor debe ser 0 o 1.
03	El comando no debe tener valores en el campo de datos.
04	El valor del campo de datos está fuera de rango.

Un ejemplo de una respuesta donde se generó un error para el comando D1 se muestra a continuación:

Dirección del esclavo	Función	Datos	CRC (LSB)	CRC (MSB)
15 hex	86 hex	03 hex	42 hex	65 hex

De esta respuesta podemos notar que el valor de la función es 80 hex veces mayor al valor de la función del comando D1, lo cual nos indica que se generó un error. El valor del campo de datos nos dice qué tipo de error se tuvo, 03 en este caso, por lo que inspeccionando la tabla 5.2, nos damos cuenta que el error se debió a que al mandar el comando desde la red de monitoreo, se introdujeron valores en el campo de datos, lo cual no es permitido.

Con respecto al valor de la dirección del esclavo, se puede decir que es de vital importancia, ya que cuando la red de monitoreo envía mensajes, sólo el esclavo que tenga la dirección especificada por el campo de la dirección de esclavo, recibirá el mensaje. Este valor sólo se puede cambiar modificando el firmware de la tarjeta convertidora de protocolos. Para modificar esta dirección se tiene que cambiar el valor especificado en la línea 29 del archivo `mdbus.dpu`, volver a compilar el proyecto `mdbus.pjt` y por último se debe volver a grabar el PIC16C74A con el archivo `mdbus.hex`.

CAPITULO 6

ANALISIS Y RESULTADOS

6.1. Explicación del Diseño

Cuando el sistema es energizado, empieza a revisar el puerto1, correspondiente a la línea Modbus RTU, esperando a que la línea se mantenga en silencio por un tiempo de 4 caracteres a la tasa de baudios actual. Una vez que el bus de la red alcanza un estado silencioso por ese tiempo, el sistema espera el primer carácter que llegue para chequearlo. Si este carácter no coincide con una dirección de 8 bits que el sistema tiene preestablecida, se espera a que la línea quede en silencio nuevamente, pero si es la dirección del sistema, éste se encarga de recibir de ese momento en adelante todos los caracteres que lleguen, como un mensaje Modbus RTU, hasta que la línea se vuelva a quedar en silencio por 4 caracteres a la tasa de baudios actual.

Cada carácter que se recibe por este procedimiento, se almacena en la RAM del MCU por medio de una tabla que se crea cada vez que llega un comando nuevo. Cuando el mensaje está terminado, se procede a realizar con el mensaje el procedimiento que se necesite y a traducir el comando a una orden ASCII para enviarla a la DPU, además de calcular el CRC para revisar si es válido el mensaje.

Cuando se ha decodificado el comando Modbus, se cambia el puerto de atención y se le envía el comando ASCII a la DPU y se escucha su respuesta, la cual es en ASCII y está determinada por caracteres de inicio y de final.

Los datos ASCII se almacenan igualmente en una tabla hasta que llegan los caracteres que indican el fin de transmisión. Los datos que se almacenan en la tabla de la RAM se procesan para fabricar la respuesta. Fabricar la respuesta Modbus conlleva eliminar los datos no importantes, convertir los datos importantes a un formato adecuado (binario), calcular el CRC y enviar estos datos en el orden que debe tener un mensaje Modbus.

El sistema debe permitir que el maestro Modbus espere la respuesta mientras el sistema se demora en decodificar el comando y revisarlo, enviar el comando a la DPU, leer su respuesta y construir la respuesta Modbus.

Se desarrolló un algoritmo relacionado con el problema que se debe resolver. Esta solución se orienta a la recepción de un mensaje Modbus. En la figura 6.1, se observa el diagrama de flujo de la solución de “software” propuesta. En ella se observan los procedimientos necesarios para que se cumpla con las especificaciones de Modicon para la comunicación maestro esclavo por medio de una red con topología de anillo.

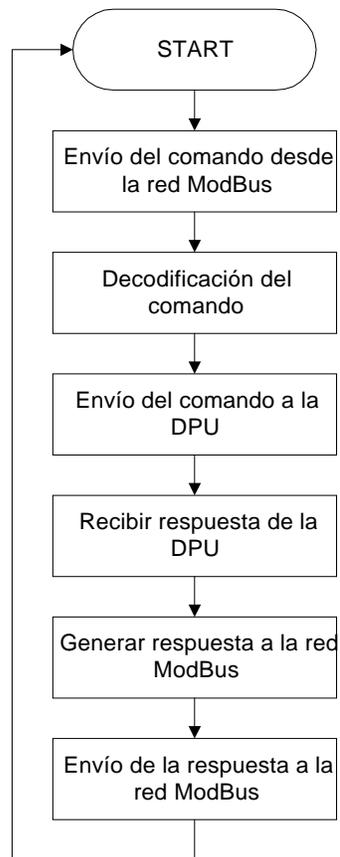


Figura 6.1 Diagrama de flujo General

En este diagrama se pueden apreciar los siguientes procedimientos principales:

Envío del comando desde la red Modbus: Este procedimiento es en sí lo que se tenía de la tarjeta convertidora de protocolos, cuya función es detectar un mensaje de la red Modbus para la DPU, recibir el mensaje, chequear que se haya recibido sin errores y generar una respuesta de que el mensaje se recibió con o sin errores.

Decodificación del comando: Este bloque es el que se encarga de identificar cuál es el comando de la DPU que se quiere utilizar. Una vez identificado el comando procede a la conversión del protocolo de Modbus RTU a Modbus ASCII.

Envío del comando a la DPU: Este bloque se encarga de mandar el comando a la DPU, recibir las respuestas de la DPU, ya que por cada carácter que se envía a la DPU, ésta responde con el mismo carácter.

Recibir respuesta de la DPU: Todos los comandos de la DPU generan respuestas para la red de monitoreo, después de haber realizado alguna acción. Este procedimiento genera una tabla en la memoria del microcontrolador con todos los datos que genera la DPU.

Generar respuesta a la red Modbus: Este procedimiento se encarga de tomar los datos que se almacenaron en la pila debido a la respuesta de la DPU, tomar los datos importantes y generar un paquete de respuesta para la red de monitoreo, aquí se debe calcular el CRC para adjuntarlo al mensaje de respuesta.

Envío de la respuesta a la red Modbus: Este procedimiento toma el mensaje de respuesta generado y lo envía a la red de monitoreo como un dato Modbus RTU y se va al inicio donde se espera otro comando.

6.2. Alcances y Limitaciones

6.2.1. Hardware

El hardware del sistema se modificó para que el PCB (Printer Circuit Board) se pueda instalar dentro de la DPU. Con el actual circuito se puede alimentar a la tarjeta convertidora de protocolos con los 24 V que suministra la DPU sin que se dé ningún problema de sobrevoltaje. El hardware actual tiene los dos puertos RS-232C requeridos y los “drivers” para manejarlos.

Podría surgir en el futuro la necesidad de aumentar la memoria RAM para ajustar el sistema a comandos complejos y podría hacerse fácilmente ya que existen en el microcontrolador seleccionado 33 pines de entrada - salida con los cuales se puede manejar un chip externo, de los cuales en este momento sólo se están usando 3 en forma definitiva. Se pueden implementar algunos tipos de indicadores usando led's que indiquen el buen funcionamiento del sistema, o bien, que indiquen cuándo se da un error, esto gracias a la amplia cantidad de puertos con los que cuenta el PIC16C74A. Si en un futuro se quisiera obtener algún tipo de medición que la DPU no permite, el PIC está listo con varias entradas analógicas para ser digitalizadas y convertidas en datos.

6.2.2. Software

El “firmware” está terminado para todos los comandos de la DPU que se habían planificado implementar en el proyecto. Con las nuevas características del firmware, llevadas a cabo en esta actualización tecnológica, realizada a la tarjeta convertidora de protocolos, la DPU es capaz de ejecutar 16 de los comandos más importantes que ella es capaz de realizar, vía puerto serie, con lo cual ahora se podrán colocar las DPUs con las que cuenta el CCE al sistema SCADA y de esta manera tener un monitoreo más eficiente.

Uno de los apartados en que hay que hacer hincapié es en la velocidad máxima a la cual se puede llevar el sistema. En torno a este aspecto, hay que recalcar que el protocolo Modbus RTU posee un “time out” que se debe respetar y esto es un problema ya que si se usa la misma tasa de baudios para ambos puertos, mientras la DPU contesta el comando que se le envía, el maestro Modbus da un “time out”. Por esta razón el puerto de la DPU debe funcionar a una velocidad mayor que el puerto Modbus RTU, de forma que la DPU conteste los comandos y estos sean procesados antes de que el maestro Modbus note el “time out”. Se debe estudiar la posibilidad de programar el “time out” Modbus RTU a un tiempo mayor de modo que no exista conflicto en ese sentido.

CAPITULO 7

CONCLUSIONES Y RECOMENDACIONES

7.1. Conclusiones

- a. La DPU se comunica serialmente con un protocolo Modbus ASCII, por su parte, la red de monitoreo del CCE utiliza un protocolo de comunicación Modbus RTU.
- b. El sistema SCADA del CCE monitorea la red de distribución haciendo uso de equipo de protección, RTUs, entre otros dispositivos, y además, utiliza un protocolo de comunicación Modbus RTU.
- c. Se comprobó que la DPU es un dispositivo de protección utilizado en la red de distribución de la CNFL, que puede ser interrogado, tanto manualmente, como por un puerto serie de comunicaciones con protocolo ASCII.
- d. Se concluye que la DPU no necesita de todos sus comandos para funcionar óptimamente, por lo que se tomaron en cuenta en el proyecto sólo los principales comandos de este dispositivo de protección.
- e. Se estableció que el microcontrolador PIC16C74A presenta características de comunicación serial eficientes, tiene varios puertos con pines de entrada-salida, tiene un convertidor de analógico a digital, y tiene suficiente memoria de programa.
- f. Se comprobó que el set de instrucciones del PIC16C74A tiene 35 instrucciones divididas en tres categorías: operaciones de bytes, operaciones de bits y operaciones con literales donde también se incluyen las operaciones de control.

- g.** Se determinó que el hardware presente en el CCE de la tarjeta convertidora de protocolos, no funcionaba correctamente, debido a varios cortos en las pistas del circuito impreso, además, no se podía instalar dentro de la DPU.
- h.** Se concluyó que el firmware que tenía la tarjeta convertidora de protocolos sólo detectaba los comandos, pero no los ejecutaba, debido a la falta de un procedimiento de decodificación y de procedimientos que ejecutaran los comandos.
- i.** Se diseñó un firmware para que la tarjeta convertidora de protocolos decodificara los comandos, los ejecutara y para que pudiese realizar la conversión de protocolos.
- j.** Se determinó que el hardware del proyecto necesitaba un regulador de voltaje, capaz de bajar los 24 V que da la DPU a 5 V, para alimentar los componentes de la tarjeta convertidora de protocolos.
- k.** Se comprobó que las nuevas características diseñadas al firmware de la tarjeta convertidora de protocolos funcionan perfectamente, esto después de haber realizado varias pruebas y haber analizado los resultados de dichas pruebas.

7.2.Recomendaciones

Las siguientes son recomendaciones que serían útiles en caso de que se deseara mejorar el desempeño del sistema más allá de los límites de este proyecto:

- a.** Implementar un modo “broadcast” donde varias DPU actúen frente a un mismo comando.
- b.** Dar soporte a una velocidad mayor de trabajo.
- c.** Poder implementar vía software, un diagnóstico de la DPU.

BIBLIOGRAFIA

1. Texas Instruments, US. 1997. Designer's guide and Databook. 1 disco compacto, 8mm.
2. Texas Instruments, US. 1997. Logic Selector Guide and Databook. 1 disco compacto, 8mm.
3. ABB. Buyer's Guide 98/99 Protection, monitoring and Control descriptive Literature. ABB. Coral Springs, USA. 1999
4. Microchip, US. 2000. Microchip technical library CD-ROM, second edition 2000. 2 discos compactos, 8mm.
5. Microchip, US. 2000. PICmicro Microcontrollers (MCUs), PIC16C7X Family, PIC16C74A datasheet and MPLAB® version 5.11.02 homepage (en línea). Consultado 20 feb. 2001. Disponible en <http://www.microchip.com>
6. Fallas Chinchilla, A. 2000. Interfaz de Comunicación entre dispositivos Modbus® y dispositivos ASCII para Red de Monitoreo del Centro de Control de Energía de la CNFL S.A. Tesis Bach. Ing. Elec. Cartago, Costa Rica, Instituto Tecnológico de Costa Rica.10,20 p.
7. FieldNet AS. 2001. Modbus Protocol homepage (en línea). Consultado 14 feb. 2001. Disponible en http://www.fieldnet.no/fbacad/chapter_1.htm
8. WinTECH Software, US. 1997. ModScan32 Application Description homepage (en línea). Consultado 16 feb. 2001. Disponible en <http://www.win-tech.com/html/modscan32.htm>

APENDICE: Glosario

Alimentador: Circuito que alimenta a una subestación.

Amperímetro: Aparato que sirve para medir los amperios de una corriente.

Baudios: Medida utilizada para la velocidad de una transmisión digital, la cual se da en bits por segundo.

CPU: Unidad de procesamiento central.

CRC: Chequeo de redundancia cíclica, es un método para determinar errores en transmisiones digitales de datos, este genera un valor de 16 bits el cual es añadido al mensaje en el campo de chequeo de error, donde se envían primero el byte menos significativo y luego el más significativo.

DPU: Unidad de protección de distribución, es un dispositivo utilizado para protección de circuitos de distribución de energía eléctrica.

Estación remota de poste: Es una RTU colocada en un punto específico, la cual se encarga de una aplicación específica.

Firmware: Microprograma de un procesador o microcontrolador.

Frame: Estructura de un mensaje en una transmisión digital.

Modbus ASCII: Protocolo de comunicación donde cada byte en un mensaje se envía como dos caracteres ASCII. Utiliza un chequeo de redundancia longitudinal para determinar errores en la transmisión.

Modbus RTU: Protocolo de comunicación donde cada byte en un mensaje se envía como dos caracteres hexadecimales. Utiliza un chequeo de redundancia cíclica para determinar errores en la transmisión.

Protocolo: Set de reglas que hacen la comunicación en las redes más eficiente. Determina el formato y transmisión de los datos.

LRC: Chequeo de redundancia longitudinal, es un método para determinar errores en transmisiones digitales de datos, este genera un valor de 8 bits el cual es añadido al mensaje en el campo de chequeo de error.

LSB: Bit menos significativo, es aquel que está más a la derecha y que tiene el menor valor posicional.

MSB: Bit más significativo, es aquel que está más a la izquierda y que tiene el mayor valor posicional.

PCB: Tarjeta de circuito impreso, está formada por pistas conductoras de corriente, las cuales conectan varios tipos de dispositivos (chips, resistencias, capacitores, interruptores, bases para montar chips, etc.) entre sí, también tiene agujeros para colocar y soldar los dispositivos.

RAM: Memoria de acceso aleatorio, es una memoria semiconductor volátil, en la cual se garantiza que los tiempos de acceso son iguales sin importar la ubicación de la información.

RTU: Unidad terminal de radio, es un dispositivo de comunicación de datos, puede ser unidireccional o bidireccional, funciona como interfaz para ejecutar comandos o recibir datos en una red de distribución eléctrica.

SCADA: Sistema de supervisión, control y adquisición de datos, es una red que se encarga de monitorear la red de distribución eléctrica de la CNFL.

Transformador: Máquina eléctrica que se utiliza para transformar un voltaje de un valor a otro.

TTL: Lógica de transistor transistor, son dispositivos electrónicos formados por transistores que utilizan niveles de voltaje binarios para su operación, donde 0V establece un nivel lógico de cero y 5V establece un nivel lógico de uno.

USART: Transmisor y receptor asíncrono sincrónico universal, es un chip programable que se encarga de controlar la transferencia y recepción de datos asíncronos o sincrónicos por el puerto serie.

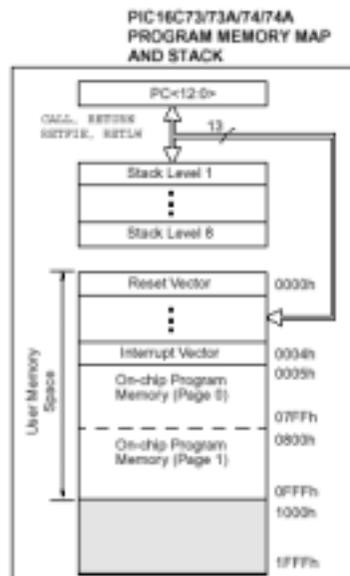
ANEXOS

Anexo 1: Características del MCU PIC16C74A

		PIC16C73A	PIC16C74A	PIC16C76	PIC16C77
Clock	Maximum Frequency of Operation (MHz)	20	20	20	20
	Memory				
Memory	EPROM Program Memory (x14 words)	4K	4K	8K	8K
	Data Memory (bytes)	192	192	368	368
Peripherals	Timer Module(s)	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2	TMR0, TMR1, TMR2
	Capture/Compare/PWM Module(s)	2	2	2	2
Features	Serial Port(s) (SPI [®] /C, USART)	SPI [®] /C, USART	SPI [®] /C, USART	SPI [®] /C, USART	SPI [®] /C, USART
	Parallel Slave Port	—	Yes	—	Yes
	A/D Converter (8-bit) Channels	5	8	5	8
	Interrupt Sources	11	12	11	12
Features	I/O Pins	22	33	22	33
	Voltage Range (Volts)	2.5-6.0	2.5-6.0	2.5-6.0	2.5-6.0
	In-Circuit Serial Programming	Yes	Yes	Yes	Yes
	Brown-out Reset	Yes	Yes	Yes	Yes
	Packages	28-pin SDIP, SOIC	40-pin DIP, 44-pin PLCC, MQFP, TQFP	28-pin SDIP, SOIC	40-pin DIP, 44-pin PLCC, MQFP, TQFP

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability. All PIC16C70X Family devices use serial programming with clock pin RB6 and data pin RB7.

Note 1: Please contact your local Microchip sales office for availability of these devices.



Anexo 2: Características eléctricas y ambientales del MCU

Absolute Maximum Ratings †

Ambient temperature under bias.....	-55 to +125°C
Storage temperature	-65°C to +150°C
Voltage on any pin with respect to Vss (except VDD, $\overline{\text{MCLR}}$, and RA4).....	-0.3V to (VDD + 0.3V)
Voltage on VDD with respect to Vss	-0.3 to +7.5V
Voltage on $\overline{\text{MCLR}}$ with respect to Vss (Note 2)	0 to +14V
Voltage on RA4 with respect to Vss	0 to +14V
Total power dissipation (Note 1).....	1.0W
Maximum current out of Vss pin	300 mA
Maximum current into VDD pin	250 mA
Input clamp current, I _{IK} (V _I < 0 or V _I > VDD)	±20 mA
Output clamp current, I _{OK} (V _O < 0 or V _O > VDD).....	±20 mA
Maximum output current sunk by any I/O pin.....	25 mA
Maximum output current sourced by any I/O pin	25 mA
Maximum current sunk by PORTA, PORTB, and PORTE (combined) (Note 3).....	200 mA
Maximum current sourced by PORTA, PORTB, and PORTE (combined) (Note 3)	200 mA
Maximum current sunk by PORTC and PORTD (combined) (Note 3).....	200 mA
Maximum current sourced by PORTC and PORTD (combined) (Note 3).....	200 mA

Note 1: Power dissipation is calculated as follows: $P_{dis} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$

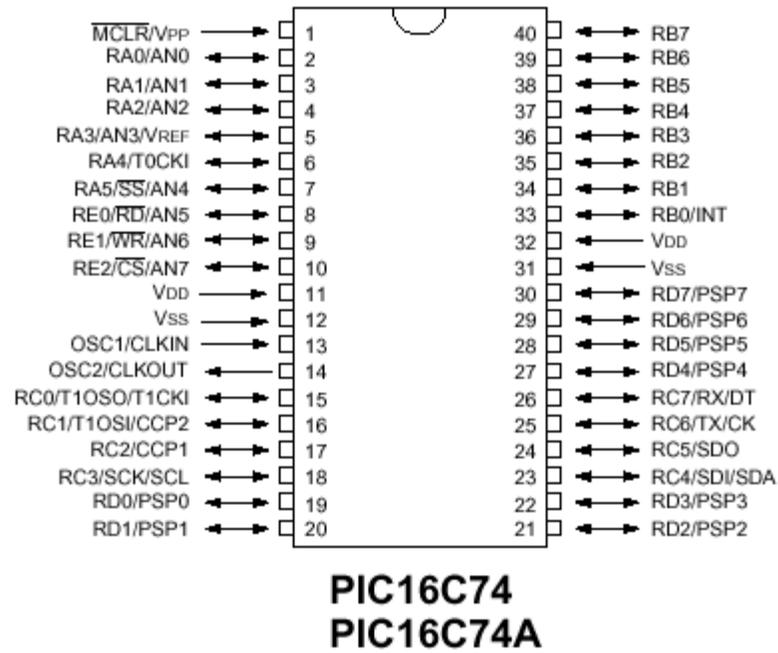
Note 2: Voltage spikes below Vss at the $\overline{\text{MCLR}}$ pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the $\overline{\text{MCLR}}$ pin rather than pulling this pin directly to Vss.

Note 3: PORTD and PORTE are not implemented on the PIC16C73.

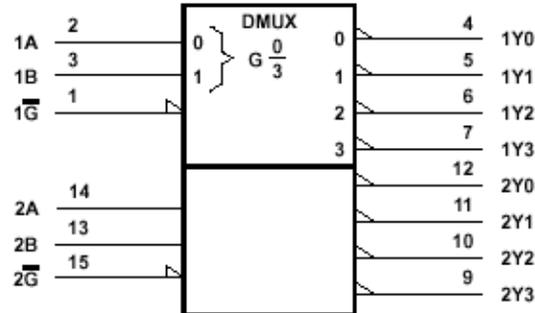
† NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

Anexo 3: Distribución de pines del PIC16C74A

PDIP, Windowed Cerdip



Anexo 4: Características del decoder/demux 74LS139



FUNCTION TABLE

INPUTS			OUTPUTS			
ENABLE \overline{G}	SELECT		Y0	Y1	Y2	Y3
	B	A				
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage, V_{CC}	7 V
Input voltage, V_I	7 V
Operating free-air temperature range, T_A : SN54ALS139	-55°C to 125°C
SN74ALS139	0°C to 70°C
Storage temperature range	-65°C to 150°C

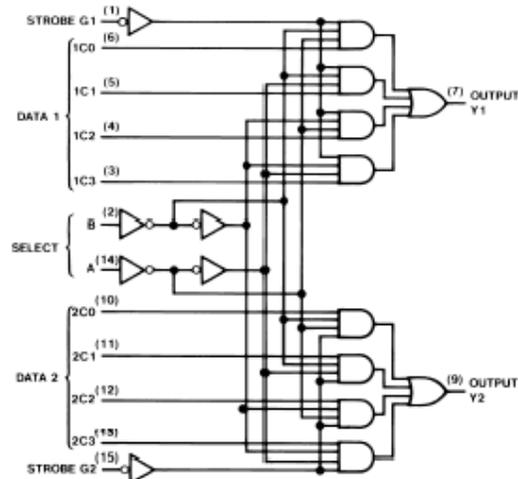
† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

recommended operating conditions

	SN54ALS139			SN74ALS139			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC} Supply voltage	4.5	5	5.5	4.5	5	5.5	V
V_{IH} High-level input voltage	2			2			V
V_{IL} Low-level input voltage			0.7			0.8	V
I_{OH} High-level output current			-0.4			-0.4	mA
I_{OL} Low-level output current			4			8	mA
T_A Operating free-air temperature	-55		125	0		70	°C

Anexo 5: Datos generales del MUX 74LS153

Logic Diagram



Function Table

Select Inputs		Data Inputs				Strobe	Output
B	A	C0	C1	C2	C3	G	Y
X	X	X	X	X	X	H	L
L	L	L	X	X	X	L	L
L	L	H	X	X	X	L	H
L	H	X	L	X	X	L	L
L	H	X	H	X	X	L	H
H	L	X	X	L	X	L	L
H	L	X	X	H	X	L	H
H	H	X	X	X	L	L	L
H	H	X	X	X	H	L	H

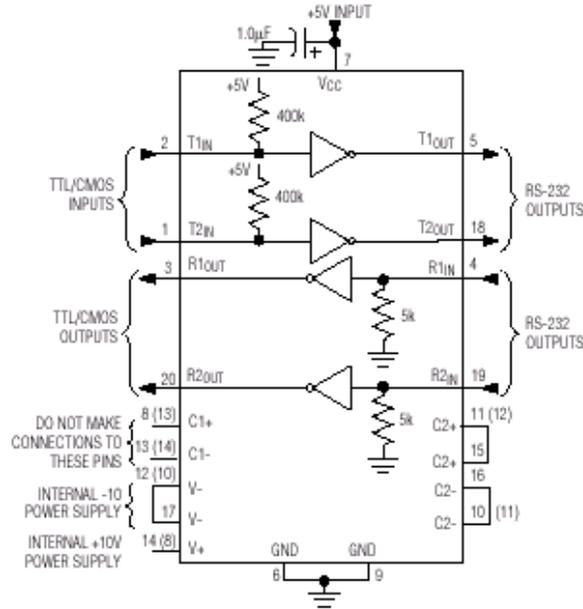
Select inputs A and B are common to both sections.
H = High Level, L = Low Level, X = Don't Care

Recommended Operating Conditions

Symbol	Parameter	DM54LS153			DM74LS153			Units
		Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High Level Input Voltage	2			2			V
V_{IL}	Low Level Input Voltage			0.7			0.8	V
I_{OH}	High Level Output Current			-0.4			-0.4	mA
I_{OL}	Low Level Output Current			4			8	mA
T_A	Free Air Operating Temperature	-55		125	0		70	°C

Note 1: The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.

Anexo 6: Características del MAX233



ABSOLUTE MAXIMUM RATINGS—MAX220/222/232A/233A/242/243

Supply Voltage (V_{CC}).....	-0.3V to +6V	20-Pin Plastic DIP (derate 8.00mW/°C above +70°C) ..	440mW
Input Voltages		16-Pin Narrow SO (derate 8.70mW/°C above +70°C) ..	696mW
T_{IN}	-0.3V to ($V_{CC} - 0.3V$)	16-Pin Wide SO (derate 9.52mW/°C above +70°C).....	762mW
R_{IN} (Except MAX220)	$\pm 30V$	18-Pin Wide SO (derate 9.52mW/°C above +70°C).....	762mW
R_{IN} (MAX220).....	$\pm 25V$	20-Pin Wide SO (derate 10.00mW/°C above +70°C).....	800mW
T_{OUT} (Except MAX220) (Note 1)	$\pm 15V$	20-Pin SSOP (derate 8.00mW/°C above +70°C)	640mW
T_{OUT} (MAX220).....	$\pm 13.2V$	16-Pin CERDIP (derate 10.00mW/°C above +70°C).....	800mW
Output Voltages		18-Pin CERDIP (derate 10.53mW/°C above +70°C).....	842mW
T_{OUT}	$\pm 15V$	Operating Temperature Ranges	
R_{OUT}	-0.3V to ($V_{CC} + 0.3V$)	MAX2_ AC_ , MAX2_ C_	0°C to +70°C
Driver/Receiver Output Short Circuited to GND.....	Continuous	MAX2_ AE_ , MAX2_ E_	-40°C to +85°C
Continuous Power Dissipation ($T_A = +70^\circ C$)		MAX2_ AM_ , MAX2_ M_	-55°C to +125°C
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C).....	842mW	Storage Temperature Range	-65°C to +160°C
18-Pin Plastic DIP (derate 11.11mW/°C above +70°C).....	889mW	Lead Temperature (soldering, 10sec)	+300°C

Note 1: Input voltage measured with T_{OUT} in high-impedance state, \overline{SHDN} or $V_{CC} = 0V$.

Note 2: For the MAX220, V_+ and V_- can have a maximum magnitude of 7V, but their absolute difference cannot exceed 13V.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.