

**INSTITUTO TECNOLÓGICO DE COSTA RICA**

**ESCUELA DE INGENIERÍA ELECTRÓNICA**



**COMPAÑÍA NACIONAL DE FUERZA Y LUZ S.A.**

**CNFL S.A.**

**“Optimización de una interface aplicada en un sistema SCADA para comunicar una red que utiliza protocolo ModBus con dispositivos de protección de distribución eléctrica que se manejan con protocolo ASCII”**

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

**Julio César Sanabria Solís**

**Cartago, Junio del 2002**

*Dedico el éxito de este esfuerzo, en primer lugar al Señor Dios,  
de quien nos viene la vida y todos los dones que recibimos,*

***“Amén. Alabanza, gloria, sabiduría  
acción de gracias, honor, poder y fuerza,  
a nuestro Dios por los siglos de los siglos. Amén.”  
Apocalipsis 7, 12***

*A mis padres, quienes con amor abnegado  
me criaron junto a mis cuatro hermanos, enseñándonos que  
lo más importante de la vida está en conocer a Jesucristo, amarle y seguirle.  
Y por su apoyo incondicional, en mis momentos más difíciles, y aún en medio de  
situaciones económicas adversas para nosotros.*

*A mis hermanos y sobrinos, a quienes tanto amo.*

*A mi amada esposa Lizeth, en quien descubro cada día la inmensa riqueza de la  
vocación a la que el Señor me ha llamado, por su buen ejemplo de dedicación,  
disciplina, estudio y trabajo, por su apoyo y comprensión.*

## **Agradecimiento**

Agradezco al Ing. Eric Bogantes por la posibilidad que me brindó de realizar la práctica profesional en esta prestigiosa empresa nacional, por su ayuda y por la confianza depositada en mi persona para llevar a cabo este importante proyecto.

Gracias a todo el personal del Centro de Control, por su espíritu de colaboración y compañerismo; a mis amigos: Gerardo Zúñiga, Jafeth Carrillo, Fernando Quirós, don Luis Umaña y don Mario Vásquez.

A todos los profesores de la Escuela de Ingeniería Electrónica del Instituto Tecnológico de Costa Rica que a lo largo de todos estos años han estado involucrados en el proceso de mi formación profesional, especialmente a los ingenieros: don Carlos Badilla, don Pedro Murillo, don Arnoldo Rojas, don Luis Paulino Méndez, don Eduardo Interiano y don Miguel Hernández. A todos ustedes mi respeto y gratitud.

Por último quiero agradecer de una manera especial a mi apreciado amigo y profesor, el Ing. William Marín, primero por su valiosa amistad y por el tiempo que dedicó a asesorarme en esta práctica profesional.

A todos, muchas gracias.

## 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 de energía eléctrica. En esta red hay unidades de protección de distribución (DPU por sus siglas en inglés), están diseñadas para la protección de fase y tierra contra sobre corrientes en sistemas de distribución de potencia eléctrica. Las DPU permiten además, recierres multidisparo, varias mediciones y funciones de grabación de eventos.

Hasta el momento, el CCE no ha conectado las DPU al sistema SCADA porque los protocolos no son compatibles. El sistema SCADA utiliza un protocolo Modbus RTU, mientras que las DPU se comunican con un protocolo ASCII.

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

Para actualizar la tarjeta que convierte protocolos, primero se debe estudiar todas las características, tanto de hardware como de software de la tarjeta existente.

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

## **Abstract**

In the Energy Control Center (CCE for their initials in Spanish) of the Power and Light National Company, S.A., is made use of a SCADA system for monitoring the electrical energy distribution network. In this network, there are distribution protection units (DPU), those which are designed for phase and ground overcurrent protection in electrical power distribution systems. The DPU allow too multi-shot reclosing, various metering and event recording functions.

Until the moment, the CCE has not connected the DPU to the SCADA system because the communication protocols are not compatible. The SCADA system uses a protocol Modbus RTU, while the DPU made use of protocol ASCII.

In this project a protocols converter card was updated technologically, which allows the DPU to be connected to the monitoring net. This card will be install inside each DPU, making these devices work as any other slave in the SCADA system.

To update technologically the protocols converter card, first they were had to study all the characteristics, hardware and software of the existent card.

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

## ÍNDICE GENERAL

<b>Capítulo 1: Introducción</b>	<b>10</b>
1.1 Descripción de la empresa	10
1.1.1 Descripción general	10
1.1.2 Descripción del Centro de Control de Energía	11
1.2 Definición del problema y su importancia	13
1.3 Objetivo general	18
1.4 Objetivos específicos	19
<b>Capítulo 2: Antecedentes</b>	<b>20</b>
2.1 Estudio del problema a resolver	20
2.1.1 Protocolo de comunicaciones Modbus	23
2.1.1.1 Introducción al Protocolo Modbus	23
2.1.1.2 Transacciones en redes Modbus	24
2.1.1.3 La solicitud (Query)	25
2.1.1.4 La respuesta (Response)	26
2.1.1.5 Modos de transmisión serial	26
2.1.1.6 Entramado del paquete Modbus	29
2.1.1.7 El chequeo de error CRC <sub>1</sub>	30
2.1.1.8 Como se transmiten serialmente los datos	31
2.1.1.9 Respuestas de excepción	32
2.1.2 Unidad de protección de distribución (DPU)	35
2.1.2.1 Descripción general	35
2.1.2.2 Puerto de comunicaciones de la DPU	35
2.1.2.3 Velocidad de transmisión, formato de datos y protocolo	36
2.2 Requerimientos de la empresa	36
2.3 Solución propuesta	38
<b>Capítulo 3: Procedimiento Metodológico</b>	<b>39</b>
<b>Capítulo 4: Descripción del hardware utilizado</b>	<b>43</b>
4.1 Diagrama general de bloques	43
4.2 Diagrama detallado de la TCP	44
4.3 Diagrama esquemático de la TCP	47
<b>Capítulo 5: Descripción del software utilizado</b>	<b>48</b>
5.1 Ambiente de desarrollo del software	48
5.2 Software desarrollado	50

<b>Capítulo 6: Análisis y resultados</b>	<b>52</b>
6.1 Explicación del Diseño	52
6.2 Alcances y Limitaciones	54
6.2.1 Hardware	54
6.2.2 Software	55
<b>Capítulo 7: Conclusiones y recomendaciones</b>	<b>56</b>
7.1 Conclusiones	56
7.2 Recomendaciones	57
<b>Bibliografía</b>	<b>58</b>
<b>Apéndices</b>	<b>59</b>
Apéndice A.1: Glosario	59
Apéndice A.2: Diagrama esquemático de la TCP	62
Apéndice A.3: Comandos de la DPU implementados en la TCP	63
Apéndice A.4: Diagrama de flujo del procedimiento Count_Timeout	64
<b>Anexos</b>	<b>66</b>
Anexo B.1: Distribución de las patillas del PIC16C74A	66
Anexo B.2: Diagrama de bloques interno del PIC16C74A	67
Anexo B.3: Mapa de memoria ROM y stack del PIC16C74A	68
Anexo B.4: Relay commands	69

## ÍNDICE DE FIGURAS

<b>Figura 1.1</b>	<b>Instalación actual de una DPU a la red de monitoreo</b>	<b>16</b>
<b>Figura 1.2</b>	<b>Instalación posible entre la DPU y la red de monitoreo</b>	<b>16</b>
<b>Figura 1.3</b>	<b>Diagrama de bloques de una instalación ideal entre la DPU y el sistema SCADA</b>	<b>17</b>
<b>Figura 2.1</b>	<b>Esquema del Ciclo pregunta-respuesta en la comunicación de sistemas mediante MODBUS</b>	<b>25</b>
<b>Figura 2.2</b>	<b>Entramado de Modbus RTU</b>	<b>29</b>
<b>Figura 2.3</b>	<b>Secuencia de bits para el modo RTU</b>	<b>32</b>
<b>Figura 4.1</b>	<b>Diagrama de bloques general del sistema</b>	<b>43</b>
<b>Figura 4.2</b>	<b>Multiplexión de las líneas Rx y Tx de la TCP</b>	<b>46</b>
<b>Figura 4.3</b>	<b>Diagrama esquemático de la TCP</b>	<b>47</b>
<b>Figura 5.1</b>	<b>Pantalla principal del programa MPLAB</b>	<b>49</b>
<b>Figura 5.2</b>	<b>Vista del programa MODSCAN32</b>	<b>49</b>



## ÍNDICE DE TABLAS

<b>Tabla 2.1</b>	<b>Características básicas de Modbus ASCII</b>	<b>27</b>
<b>Tabla 2.2</b>	<b>Características básicas de Modbus RTU</b>	<b>28</b>

# **CAPÍTULO 1**

## **INTRODUCCIÓN**

---

### **1.1 Descripción de la empresa**

#### **1.1.1 Descripción general**

La Compañía Nacional de Fuerza y Luz, S.A. (CNFL) en su servicio cubre 903 Km<sup>2</sup>, que representan el 2% del territorio nacional, y tiene 400.000 clientes, y se estima que el número de usuarios de sus servicios supera el millón de personas.

Está ubicada en la Gran Área Metropolitana y es responsable del suministro de energía al 40% de los clientes del sistema eléctrico costarricense y comercializa el 46% del total de la electricidad del mercado de distribución nacional.

La empresa inició el 9 de agosto de 1884, cuando el país se ubicó de tercero en el mundo al inaugurar el servicio eléctrico público a partir de fuente renovables.

En 1941 la CNFL adquiere su actual razón social como resultado de la fusión de tres empresas eléctricas: Compañía Nacional de Electricidad, The Costa Rican Electric Light and Traction Company Limited y Compañía Nacional Hidroeléctrica.

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

Desde un punto de vista corporativo, CNFL es actualmente una empresa filial de la Corporación ICE.

El Gerente y Presidente del Consejo de Administración de la CNFL es el Ing. Pablo Cob Saborío. Luego de 60 años de proyección al país, la institución tiene más de 1300 empleados.

### **1.1.2 Descripción del Centro de Control de Energía**

El Centro de Control de Energía (CCE), es el lugar desde el cual se controla y supervisa el funcionamiento integral de todo el sistema eléctrico de la CNFL.

El Centro de Control de Energía (CCE) será el departamento donde se desarrollará 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 Km<sup>2</sup> de la 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 los mismos.

Para lograr sus funciones este departamento cuenta con recursos de hardware y software, tales como el sistema SCADA que le permite realizar el monitoreo desde numerosos Centros de Comunicación Remota, encargados de recopilar la información necesaria.

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 quien es el 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 (Supervisory Control and Data Acquisition) con protocolo de comunicación ModBus. 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 necesitan, un protocolo de comunicación ModBus, 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, el sistema necesita una tarjeta, para poder conectar estos dispositivos a la red de monitoreo. En este informe se le denominará a la tarjeta con el nombre de TCP (tarjeta que convierte protocolos).

En el pasado esta tarjeta fue rediseñada, construida e implementada con algunas características pensando en incorporar una de ellas dentro de cada una de las DPU. Sin embargo no se ha reproducido la tarjeta porque es necesario para esto mejorar sus características como lo son su velocidad, aumentar de ser posible el número de comandos de comunicación, implementar la posibilidad de que todas las DPU puedan recibir y ejecutar simultáneamente un mismo comando dado por la Estación Maestra, lo que se conoce como modo “broadcast”.

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 tales como transformadores, alimentadores, y otros. Además la DPU evita que una falla se propague causando la salida de un sector más grande y perjudicando a más personas.

La DPU se puede acceder por medio de un teclado numérico que viene en la carátula frontal de la unidad. También puede ser accesada remotamente por medio de un puerto serial, y se hace uso del protocolo ASCII.

En el caso de que se quiera operar la unidad, por ejemplo, conocer una corriente o cerrar un circuito, es necesario que un operador vaya hasta el lugar donde se encuentra la DPU e ingrese los comandos en el teclado alfanumérico.

Con la implementación de la TCP se evitaría este inconveniente porque el sistema SCADA permitiría que los operadores del Despacho de Carga operen el dispositivo a distancia y conozcan su estatus.

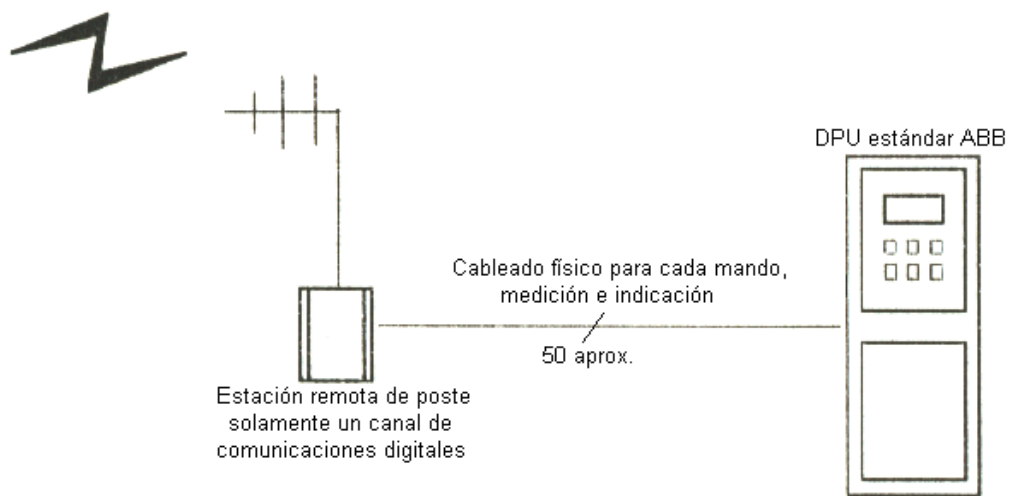
Si alguna de las protecciones de la DPU debe operar, dicha información debe ir al Centro de Control de Energía, de modo que se pueda saber de la falla en el Despacho de Carga. Además el Centro de Control requiere una gran variedad de datos del dispositivo, tales como corrientes, demandas pico, horas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

Aunque la única TCP, con la cual cuenta la CNFL S.A., permite a una DPU comunicarse con el sistema SCADA, ésta cuenta con una gran limitante tecnológica y es que trabaja a una velocidad de 1200 bits por segundo. Tratándose de un sistema que adquiere datos, es necesario aumentar la velocidad lo máximo posible, esto es a 9600 bits por segundo.

También es necesario la implementación del modo “broadcast” porque de esta manera será posible sincronizar los relojes internos de todas las DPU. De lo contrario podría suceder por ejemplo que un evento producido en determinado instante sea registrado por algunas DPU en horas diferentes.

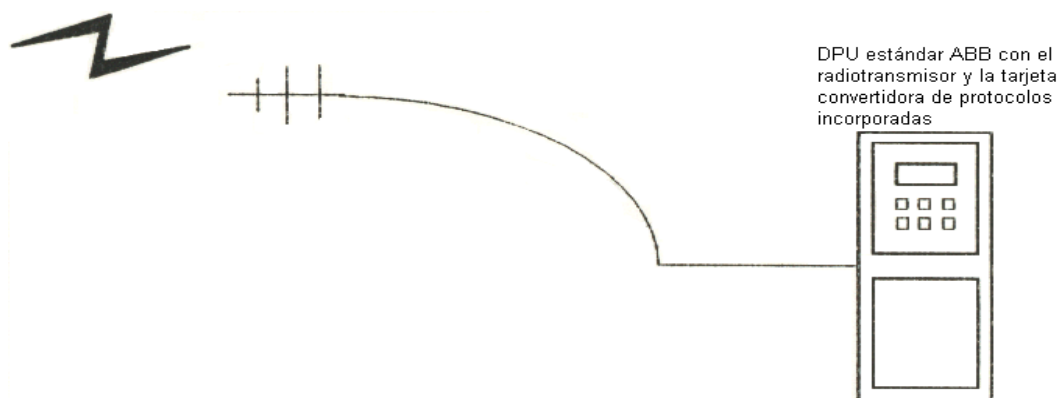
Actualmente para operar la DPU, se necesita hacer una instalación que aparte de costosa es poco eficiente ya que se tiene que utilizar una estación remota de poste, la cual tiene solo un canal de comunicaciones digitales. Por este motivo no es posible interrogar por medio de comunicación digital a través 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 1.1 muestra dicha instalación.



**Figura 1.1** Instalación actual de una DPU a la red de monitoreo

Para mejorar la instalación que se muestra en la figura 1.1, 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 TCP y un radio transmisor. La figura 1.2 muestra como se realizaría la comunicación entre la DPU y la red de monitoreo del CCE.



**Figura 1.2** Instalación posible entre la DPU y la red de monitoreo



La figura 1.3 es un diagrama de bloques que muestra con más detalle la instalación que se muestra en la figura 1.2. 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 al Despacho de Carga del Centro de Control de Energía donde se encuentra el sistema SCADA.



**Figura 1.3** 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 las figuras 1.2 y 1.3 en forma de diagrama de bloques, se obtendría el máximo provecho de las características de la unidad de protección de distribución (DPU por sus siglas en inglés), sin embargo como la TCP no cuenta con las herramientas necesarias para lograr una comunicación eficaz, para que el Despacho de Carga pueda utilizar correctamente a la DPU, la instalación que se muestra en la figura 2 actualmente no se puede llevar a cabo.

### **1.3 Objetivo general**

Al finalizar este proyecto se alcanzó el objetivo general:

Mejorar el monitoreo de la red de distribución de energía de la CNFL S.A. con base en la actualización tecnológica de la tarjeta que convierte del protocolo ASCII al ModBus y viceversa.

#### 1.4 Objetivos específicos

Al concluir el proyecto, se llevaron a cabo los siguientes objetivos específicos:

1. Estudiar la Unidad de Protección de Distribución.
2. Recopilar información de los protocolos y del sistema SCADA, involucrados en el proyecto.
3. Estudiar el Microcontrolador PIC 16C74A.
4. Examinar el funcionamiento de la TCP.
5. Determinar las nuevas funciones de la DPU que se necesitan implementar en la TCP.
6. Estudiar si es posible que una sola tarjeta TCP comunique a todas las DPU con la Estación Maestra.
7. Crear en el firmware una función que utilice el programa hyperterminal de Windows, de modo que permita al sistema la comunicación entre la DPU y la Estación Maestra.
8. Implementar el *firmware* que permita al sistema funcionar en el modo "broadcast".
9. Implementar en el firmware la función que permita sincronizar simultáneamente el reloj interno de todas las DPU.
10. Reprogramar el firmware de la tarjeta TCP para que la DPU trabaje a su máxima velocidad.
11. Analizar los resultados obtenidos en las pruebas del sistema des-arrrollado.
12. Confeccionar la documentación final del proyecto.
13. Realizar la transferencia de tecnología en el Centro de Control de Energía.

## CAPÍTULO 2

### ANTECEDENTES

---

#### 2.1 Estudio del problema a resolver

La tarjeta que convierte protocolos (TCP) permite conectar la DPU al sistema SCADA.

Se desea dotar a la tarjeta con la posibilidad de trabajar a la mayor velocidad posible con que puede operar la DPU, o sea 9600 bits por segundo.

Además ha sido necesario trabajar en la implementación de dos nuevas funciones. Una es la función que actualiza la fecha (día, mes, año) y la hora con minutos y segundos. Debido a que el fin de esta función es sincronizar los relojes de todas las DPU entonces se hizo necesario cambiar la programación de manera que todos los microcontroladores de las TCP ejecuten simultáneamente la instrucción de sincronismo; así, todos deberán reconocer como suya la misma dirección y función. A esto se le conoce como "broadcast".

La otra función es desplegar una tabla de chequeo de estado donde se pueden ver activados o desactivados la ganancia analógica, el convertidor A/D circuito de disparo, si la fuente es de +5 o de +/- 15 v, control de voltaje, memoria RAM y memoria ROM.

Se requiere además de la posibilidad de transmitir información a través de la TCP utilizando el Hyperterminal de Windows. Esto brinda la posibilidad de transmitir información de una manera más rápida porque el sistema no invertiría tiempo en verificar los mensajes de comunicación entre el Sistema Maestro y la DPU.

Esta función permitiría además utilizar una computadora personal desde cualquier punto del Valle Central para consultar cualquiera de las DPU, previa orden dada por el sistema SCADA. El microcontrolador cumpliría con la función de identificar la dirección y la función, de manera que se convierte en un interruptor inteligente que conecta directamente las líneas de comunicación entre la Estación Maestra y la DPU cuando se le solicita; además, espera la orden para desconectarse y permitir al microcontrolador seguir trabajando en su modo normal.

Tanto la orden para entrar al modo hyperterminal, como la orden para salir al modo normal, serán dadas desde la Estación Maestra, con el uso de un protocolo ModBus. Si se establece el modo hyperterminal, la comunicación será más insegura, ya que no habría un código protector de errores tan eficiente como el de chequeo de redundancia cíclica; sin embargo, será posible saber que la comunicación está bien cuando en la Estación Maestra se desplieguen las respuestas de las funciones propias de la DPU.

El componente electrónico principal de la TCP es el 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 varias etapas, las 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 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 uno de ellos; esto implica programar nuevos procedimientos.

La etapa de enviar la respuesta a la red de monitoreo está estrechamente unida a la conversión de protocolos, donde, por cada procedimiento programado para ejecutar un comando, se debe programar un procedimiento de respuesta.

La TCP tenía programadas 16 funciones , con la implementación de las nuevas características ya descritas, la tarjeta permitirá la utilización de todas las funciones que posee la DPU.

Ahora que se tiene un panorama más amplio del problema que se da con la TCP, se presentarán detalles del protocolo Modbus RTU, aplicado en el sistema SCADA y del protocolo ASCII de la DPU, también se verán las características de la DPU y de cómo se puede comunicar serialmente.

## **2.1.1 Protocolo de comunicaciones 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 estos controladores 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)<sup>1</sup>, 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.

<sup>1</sup>Original Equipment Manufacturer (Fabricante de Equipo Original)

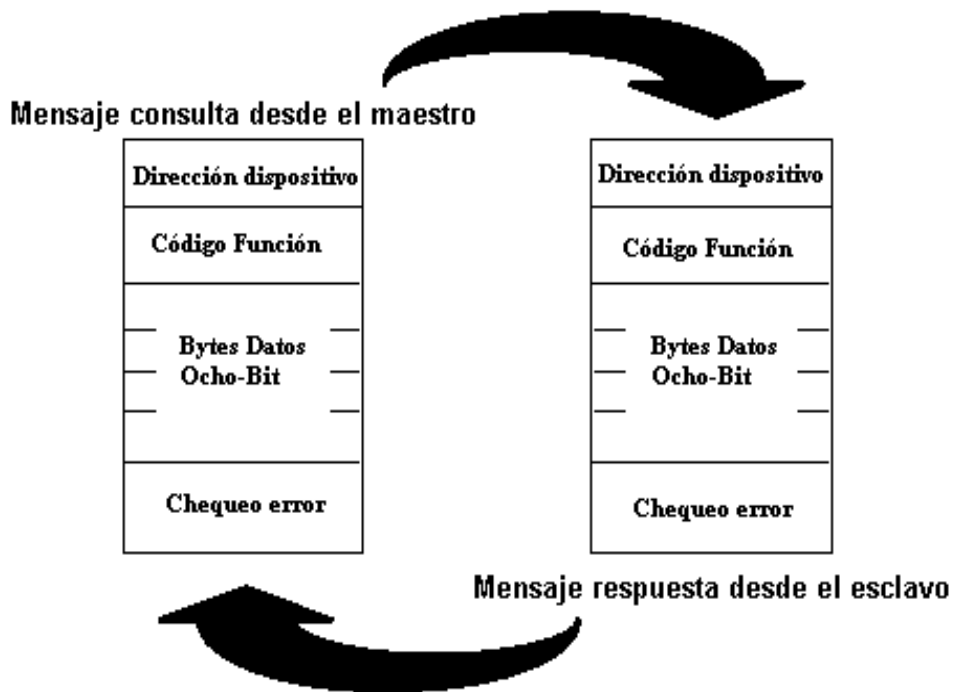
### **2.1.1.2 Transacciones en redes Modbus**

Los puertos Standard Modbus usan una interfaz serial RS-232 que define las distribuciones de pines de los conectores, el cableado, nivel de señales, velocidad de transmisión 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 supliendo 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 del Ciclo pregunta-respuesta en la comunicación de sistemas mediante MODBUS

### 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 velocidad de transmisión , 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 un carácter 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

<b>Sistema de Código</b>
Hexadecimal, Caracteres ASCII 0 ... 9, A ... F
Un carácter hexadecimal en cada carácter ASCII
<b>Bits por Byte</b>
1 start bit
7 bits 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 velocidad de transmisión que en Modbus ASCII, es decir, que es más denso.

**Tabla 2.2** Características básicas de Modbus RTU

<b>Sistema de Código</b>
Hexadecimal, Caracteres ASCII 0 ... 9, A ... F
Dos caracteres hexadecimales en cada carácter ASCII
<b>Bits por Byte</b>
1 start bit
8 bits 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 Entramado 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 velocidad de transmisión 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 CODE	DATOS	CHEQUEO	FIN
T1_T4	8 BIT	8 BIT	N X 8 BIT	16 BIT	T1_T4

**Figura 2.2** Entramado de Modbus RTU

### 2.1.1.7 El chequeo de error CRC<sub>1</sub>

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.

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.

<sup>1</sup> Siglas en inglés de Cyclical Redundancy Check

**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 00100100 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

CON CHEQUEO DE PARIDAD

START	1	2	3	4	5	6	7	8	PAR	STOP
-------	---	---	---	---	---	---	---	---	-----	------

SIN CHEQUEO DE PARIDAD

START	1	2	3	4	5	6	7	8	STOP	STOP
-------	---	---	---	---	---	---	---	---	------	------

**Figura 2.3** Secuencia de bits para el modo RTU

#### 2.1.1.9 Respuestas de excepción

Existen cuatro posibles situaciones que se pueden presentar durante el proceso de comunicación entre un dispositivo maestro y otro esclavo. Estas son:

- El esclavo recibe el mensaje sin errores y ejecuta la orden sin problema, en cuyo caso su respuesta de retorno es normal.
- El esclavo no recibe el mensaje por un error de comunicación, y por consiguiente no retorna ninguna respuesta. El dispositivo maestro detectará el error al acabarse el tiempo de espera (timeout) de la respuesta.



- El esclavo recibe el mensaje, pero detecta un error de integridad en los datos (paridad o CRC), lo que provoca que no retorne un mensaje de respuesta. El maestro se enterará del error al acabarse el tiempo de espera de la respuesta.
- El esclavo recibe el mensaje sin errores pero no puede ejecutar la orden. En este caso el esclavo retorna un mensaje de excepción informándole al maestro la naturaleza del error.

Una respuesta normal del esclavo incluirá su dirección, un campo para la función y un campo para datos. La función retornada es la misma que la función ejecutada y los datos corresponden a cualquier información solicitada.

Un mensaje de excepción incluye los mismos campos de una respuesta normal. La función retornada es la misma que recibió en el mensaje, pero con el valor "1" en el MSB<sup>1</sup>.

Esta diferencia permite detectar que hubo un error, e indica que en el campo de datos se encuentra un código de excepción (código de error). Existen ocho códigos de excepción diferentes:

**01 Función ilegal:** El código de función recibido no es reconocido por el esclavo.

**02 Dirección de datos ilegal:** la dirección para los datos es inválida para el esclavo.

---

<sup>1</sup> Todas las funciones tienen valores inferiores a 80H, por lo que el MSB siempre es cero.

- 03 Valor de dato ilegal:** el valor del dato no es permitido por el esclavo.
- 04 Fallo en el dispositivo esclavo:** un error incorregible ocurrió cuando el esclavo intentó ejecutar la acción pedida.
- 05 Reconocimiento:** el esclavo aceptó la consulta y está procesándola, pero la acción tomará mucho tiempo en completarse.
- 06 Dispositivo esclavo ocupado:** el esclavo está inmerso en el procesamiento de un comando de larga duración en su ejecución.
- 07 Reconocimiento negativo:** el esclavo no pudo ejecutar las funciones 13 o 14.
- 08 Error de paridad en la memoria:** error de paridad en la lectura de los datos en la memoria extendida.

## **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 posibilidad mencionada en el punto g, se utiliza para que la DPU se comunique con la red de monitoreo. Con la TPC como sistema traductor de protocolos se logra que la DPU reciba comandos del sistema SCADA a través del puerto RS-232C, así como que la DPU sea capaz de enviar la información solicitada a la red de monitoreo.

### **2.1.2.2 Puerto de comunicaciones de la DPU**

El puerto de comunicaciones se ubica en el panel trasero de la unidad. 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 Velocidad de transmisión, formato de datos y protocolo**

La DPU soporta una velocidad de transmisión que va desde 50 hasta 9600 bits por segundo, 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 monitoreo por incompatibilidad de protocolos.

Por esta razón el Centro de Control de Energía requiere de un sistema capaz de incorporar estas unidades de protección a la red de monitoreo.

Para utilizar las DPU en el sistema SCADA, el CCE cuenta con una tarjeta que convierte protocolos, a la cual se le rediseño el firmware de manera que satisfaga las necesidades de comunicación entre el sistema SCADA y las DPU.

Se discutió con el ingeniero a cargo, sobre las diferentes posibilidades en las que se podía trabajar para alcanzar la optimización del sistema y se definió que el proyecto final brindará las siguientes funciones:

- a.** La DPU, por medio de la TCP, ejecutará todos los comandos necesarios, tomando en cuenta que estos comandos serán enviados desde la red de monitoreo.
- b.** Permitirá que las respuestas de la DPU, cuando las haya, se envíen a la red de monitoreo en protocolo Modbus RTU.
- c.** Posibilitará la utilización del modo “broadcast” en lo que respecta a sincronía de relojes.
- d.** Posibilitará el uso del modo hyperterminal, en el que luego de haber sido autorizado desde el sistema SCADA, las DPU pueden ser gobernadas desde cualquier computadora personal utilizando el programa hyperterminal de Windows y un radiotransmisor.
- e.** El sistema podrá operar si se desea a la máxima velocidad posible, 9600 bits por segundo.

### **2.3 Solución propuesta**

La solución propuesta es por software y hardware.

La parte de la solución que tiene que ver con la programación se verá en detalle en el Capítulo 5, cuando se hable sobre el software desarrollado.

En cuanto al objetivo de lograr la comunicación que usa el programa hyperterminal de Windows, se consideró que por sencillez y velocidad es una mejor solución hacerlo por hardware, puesto que por software existe la complicación de la multiplexión de las patillas Tx y Rx del PIC y con esto la pérdida innecesaria de tiempo .

Se utilizaron un par de relés para conectar directamente las líneas Tx y Rx de la DPU con las correspondientes líneas que vienen del Sistema Maestro. Estos relés son habilitados y deshabilitados por el microcontrolador, de tal forma que el Sistema Maestro da la señal para entrar o salir del modo hyperterminal de Windows.

El objetivo de aumentar la velocidad hasta 9600 bits por segundo no obligó a variar el hardware, pero sí se cambió el cristal de 12 MHz por uno de 18,4178 MHz con el cual se obtuvo un porcentaje de error en la velocidad de transmisión de 0,08%, es decir una velocidad de 9593 bits por segundo.

## **CAPÍTULO 3**

### **PROCEDIMIENTO METODOLÓGICO**

---

#### **Metodología**

1. Estudiar la Unidad de Protección de Distribución.

Esta parte se llevó a cabo utilizando el manual Circuit-Shield Distribution Protection Unit (DPU) de ABB (Asea Brown Boveri).

Para llevar a cabo esta etapa se analizó con el Jefe del Centro de Control de Energía cuáles son los nuevos comandos de la DPU que era necesario implementar.

2. Recopilar y estudiar la información de los protocolos y del sistema SCADA, involucrados en el proyecto.

Esta etapa se llevó a cabo con ayuda de los manuales presentes en el Centro de Control de Energía así como con información obtenida de Internet. También se observó la red de monitoreo funcionando.

3. Estudiar el Microcontrolador PIC 16C74A.

Se estudió el conjunto de instrucciones y el ensamblador del Microcontrolador PIC 16C74A . Esta etapa se llevó a cabo con ayuda de la información presente en el CCE, además de los catálogos de los microprocesadores de Microchip y con información que Microchip provee a los interesados

4. Examinar el funcionamiento de la tarjeta TCP.

Para alcanzar este objetivo se estudió no solamente el hardware de la tarjeta sino que también se hicieron pruebas con diferentes procedimientos hechos en el firmware del PIC.

5. Estudiar la posibilidad de que una única TCP comunique a todas las DPU con la Estación Maestra.

Se concluyó que es necesario que cada DPU tenga su propia TCP, por razones de seguridad, porque si una sola tarjeta controla a todas las DPU si por alguna razón se daña se perdería la comunicación con todas las DPU.

6. Crear en el firmware una función que utilice el programa hyperterminal de Windows, de modo que permita al sistema la comunicación entre la DPU y la Estación Maestra. Para esto se hizo un procedimiento en el cual el sistema espera el byte de la función adecuada y en cuanto lo recibe pone en alto la patilla RC5 del PIC, esta señal activa los relés que conectan las líneas Rx y Tx adecuadas.



7. Implementar el modo “broadcast” donde varias DPU actúan frente a un mismo comando. El algoritmo y el código en ensamblador para esta función fueron diseñados de manera que la dirección cero sea asumida por todas las TCP como propia. De esta manera, una vez que el programa de cada microcontrolador reciba esta dirección, se saltará a la rutina de sincronización de relojes.

8. Implementar en el firmware la función que permite sincronizar simultáneamente el reloj interno de todas las DPU.

En esta función el usuario envía el valor del año, mes, día, hora, minutos y segundos, los cuales se guardan en una tabla, en la memoria del PIC. Luego el PIC envía a la DPU la función y se entabla una comunicación en la que la DPU pregunta por los nuevos valores y el PIC le responde.

9. Mejorar, el hardware de la TCP, de manera que ésta trabaje a mayor velocidad que la actual.

En este objetivo la solución consistió en la reprogramación del valor de un registro del PIC calculado según la velocidad requerida de 9600 bits por segundo y tomando en cuenta la frecuencia del cristal utilizado. Los cálculos fueron realizados según una fórmula dada en el manual del fabricante. Muy importante fue el diseño de las rutinas de espera que fueron afectadas por la nueva velocidad de comunicación y que tienen que ver con la espera de cuatro tiempos de carácter.

10. Etapa de análisis de los resultados obtenidos en las pruebas del sistema desarrollado.

En esta etapa se analizaron los resultados obtenidos con el fin de evaluar el cumplimiento de los objetivos.

11. Preparar la documentación final del proyecto.

12. Presentación del proyecto a la empresa y las capacitaciones necesarias para el buen uso del mismo.

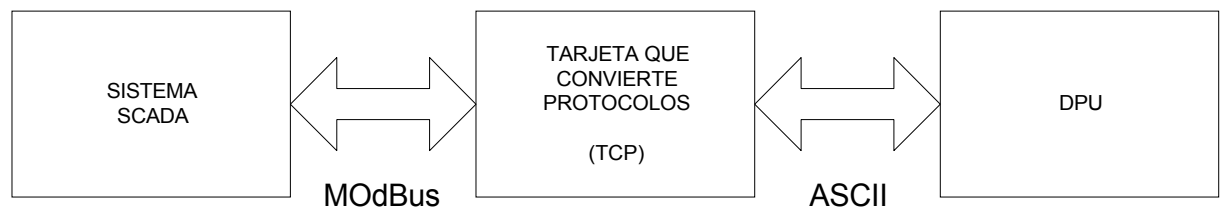
## CAPÍTULO 4

### DESCRIPCIÓN DEL HARDWARE UTILIZADO

---

#### 4.1 Diagrama general de bloques

En el diagrama se representa el funcionamiento del sistema. La señal ModBus entra a la tarjeta que convierte protocolos (TCP) y sale convertida en una señal ASCII hacia la DPU.



**Figura 4.1** Diagrama de bloques general del sistema

## 4.2 Diagrama detallado de la TCP

En la figura 4.2 se presentan los tres 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 sistema de relés es activado o desactivado por la patilla SH del PIC.

Cuando los relés son activados, unen las línea Tx del maestro con la línea Rx de la DPU y la línea Rx del maestro con la línea Tx de la DPU, además simultáneamente se desconectan las líneas Rx0 y Rx1 que vienen del decodificador para no producir conflictos en el decodificador.

El microcontrolador que se utilizó, se seleccionó por tener una USART interna; la cual hace posible la comunicación serie.

A continuación se explican en detalle cada una de las conexiones que son usadas 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 18,4178 MHz 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.

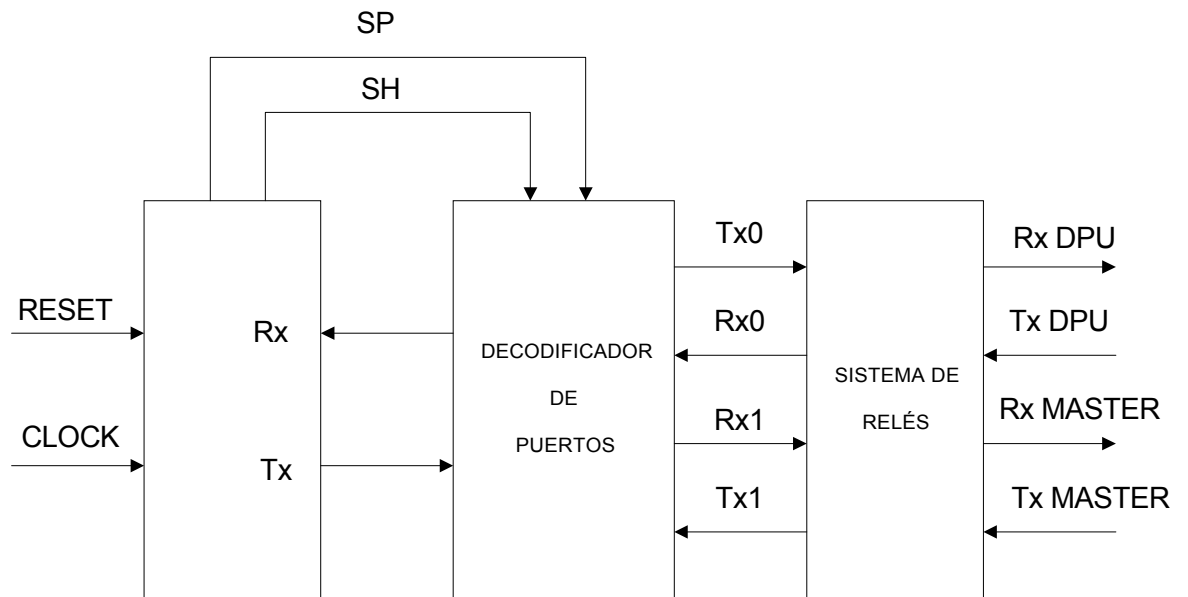
El bloque de decodificador de puertos es un arreglo de componentes TTL, entre los que se tienen un multiplexor (74LS153), un decodificador (74LS139) y un driver de comunicación serie RS-233 (MAX233). Este bloque selecciona uno de los dos puertos con los que cuenta la TCP, respondiendo a la programación 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:** pin de salida que selecciona el puerto que se utilizará.

**SH:** pin de salida que activa los relés para el modo hyperterminal.

**RX0:** pin de entrada de datos del puerto serie al que se conecta la DPU luego de pasar por el sistema de relés.



**Figura 4.2** Multiplexión de las líneas Rx y Tx de la TCP

**RX1:** pin de entrada de datos del puerto serie al que se conecta la red de monitoreo luego de pasar por el sistema de relés.

**TX0:** pin de salida de datos del puerto serie al que se conecta la DPU luego de pasar por el sistema de relés.

**TX1:** pin de salida de datos del puerto serie al que se conecta la red de monitoreo luego de pasar por el sistema de relés.

**RX DPU:** pin de salida de datos del puerto serie al que se conecta la línea de entrada Rx de la DPU.

**RX Maestro:** pin de salida de datos del puerto serie al que se conecta la línea de entrada Rx del sistema de monitoreo.

**TX DPU:** pin de entrada de datos del puerto serie al que se conecta la línea Tx de la DPU.

**TX Maestro:** pin de entrada de datos del puerto serie al que se conecta la línea Tx del sistema de monitoreo.

### 4.3 Diagrama esquemático de la TCP

En la figura 4.3 se muestra el circuito eléctrico de la TCP. Este circuito tiene pocos componentes porque la mayor parte del trabajo se realiza por software.

Una ampliación del diagrama se muestra en el apéndice A.2.

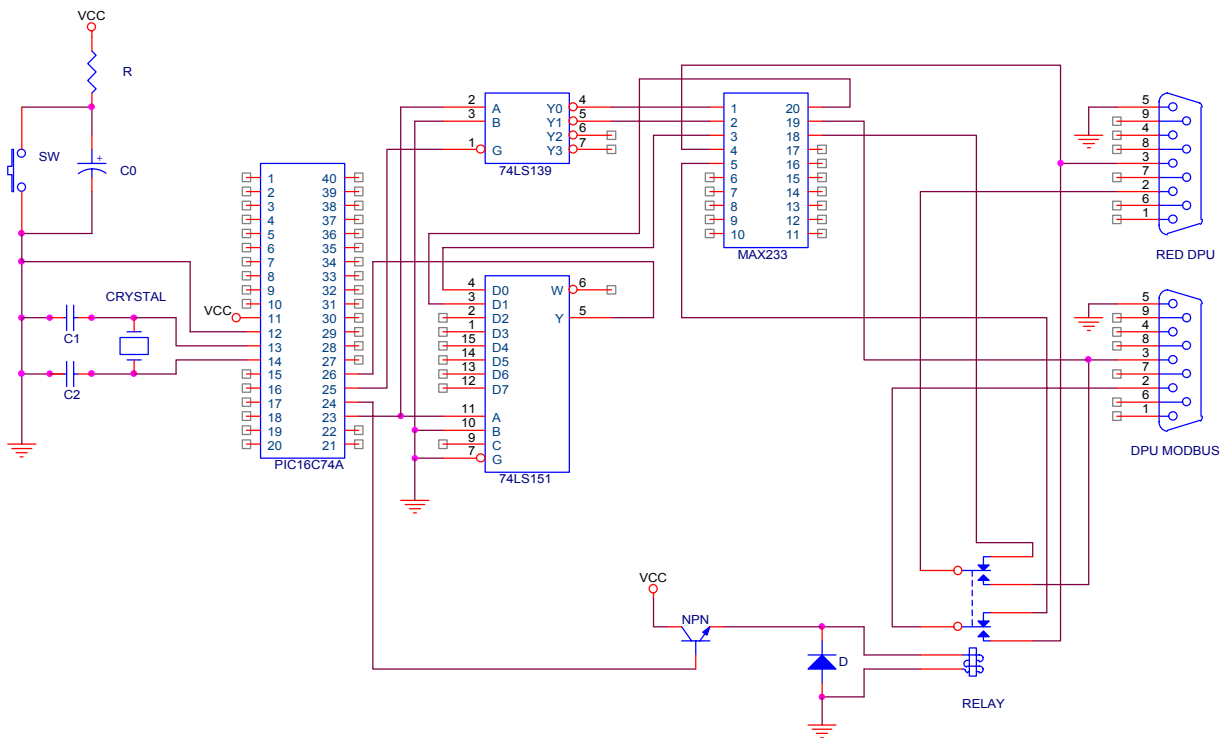


Figura 4.3 Diagrama esquemático de la TCP

## CAPÍTULO 5

### DESCRIPCIÓN DEL SOFTWARE UTILIZADO

---

#### 5.1 Ambiente de desarrollo del software

El ambiente de desarrollo que se utilizó para realizar el firmware del sistema es el MPLAB-IDE, (IDE significa Ambiente Integrado de Desarrollo), del cual se descargó gratuitamente una actualización de Internet en la página web de Microchip <http://www.microchip.com> . El MPLAB-IDE tiene un simulador integrado dentro de sus herramientas, con el cual se puede hacer simulaciones de software. La figura 5.1 muestra una imagen del programa MPLAB.

Para programar el PIC se utilizó el software propio del programador, el EPIC PROGRAMMER VER 1.41.

El diagrama del hardware se realizó por medio de ORCAD EXPRESS 7.0.

En la etapa de pruebas se utilizó un software especial, que se consiguió de forma gratuita en Internet, este es el MODSCAN 32, con el cual se puede simular la operación de un maestro Modbus. La dirección es <http://www.wintech.com/html/modscan32.htm>. La figura 5.2 muestra una imagen del MODSCAN32.

Para realizar las primeras pruebas con la DPU se utilizó el HIPERTEMINAL de Microsoft Windows.



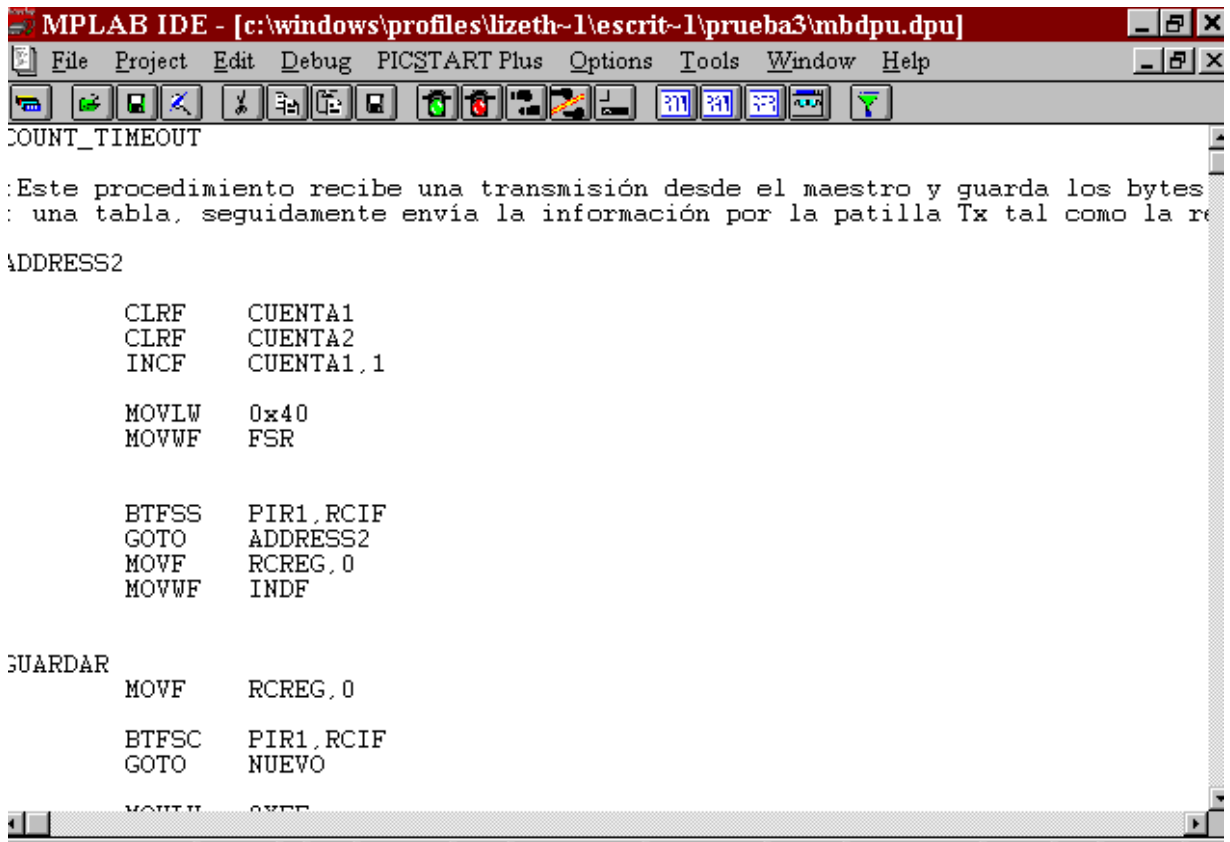


Figura 5.1 Pantalla principal del programa MPLAB

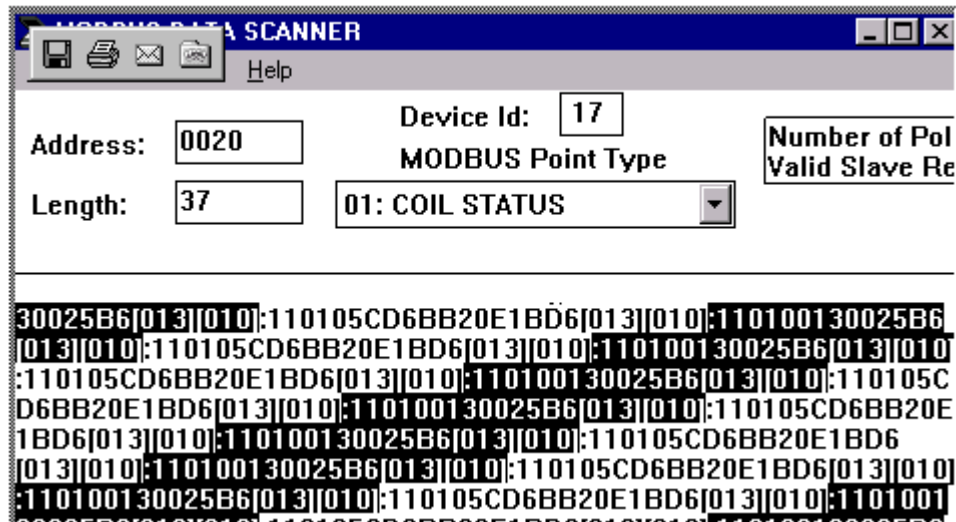


Figura 5.2 Vista del programa MODSCAN32

## 5.2 Software desarrollado

Lo primero que se hizo fue aumentar la velocidad de comunicación del sistema desde 1200 a 9600 bits por segundo. Para esto se grabó en el registro SPBRG del microcontrolador el número calculado según la fórmula del fabricante dada en el manual del PIC. Fue necesario programar el PIC en el modo HS, el cual es el que permite que el sistema trabaje con cristales que tienen alta frecuencia.

Cada TCP tiene en su microcontrolador una dirección preestablecida, pero además todas las TCP admiten la dirección cero. Cuando el sistema es energizado, empieza a revisar el puerto 1, correspondiente a la línea ModBus, esperando que haya un silencio por un tiempo de 4 caracteres a la velocidad de transmisión actual. Una vez que esto sucede, el sistema espera el primer carácter que llegue para probar si corresponde a la dirección del microcontrolador o a la dirección cero.

Si la dirección es cero, todas las TCP admiten como propia esta dirección, y se estará trabajando en el modo "broadcast". Una vez recibida la dirección cero, el PIC recibe el siguiente byte que será la función cero, dicha función hará que todos los microcontroladores ejecuten el procedimiento de sincronizar el reloj interno. El usuario deberá enviar no solo la dirección y el número de función, si no que además los valores del año, mes, día, hora, minutos y segundos para que el sistema actualice esta información en todas las DPU.

Si en un futuro se quisiera implementar otra función en el modo broadcast, bastará que se use la dirección cero y seguidamente el número de función asignado, claro está que también debe existir el procedimiento que se ejecute con esa función.

Si la dirección es la de cierta TCP, y el número de la función es 11, el programa llamará al procedimiento que consulta la tabla No 5 de la respectiva DPU y desplegará en el monitor del Sistema Maestro los 8 dígitos binarios correspondientes a los valores de esta tabla.

Si la dirección es la de determinada TCP, y el número de función es 12, se estará en presencia de la función que hace que la TCP trabaje en modo de hyperterminal de Windows. Sucederá entonces que el PIC activará el sistema de relés permitiendo que toda la información pase directamente del Sistema Maestro a la DPU y viceversa, sin que exista ningún tipo de traducción de protocolos. Dicho procedimiento, hace que el microcontrolador continúe controlando el decodificador de puerto serie, y también permite que aún en este modo, el microcontrolador esté en espera de la función 13. Esta función 13 hace que el PIC desactive el sistema de relés haciendo que la TCP trabaje de nuevo en el modo normal.

Un resumen de las nuevas funciones implementadas se puede ver en el apéndice A.3.

## CAPÍTULO 6

### ANÁLISIS Y RESULTADOS

---

#### 6.1 Explicación del Diseño

Una vez el sistema es alimentado, revisa el puerto que corresponde a la línea Modbus RTU. Espera a que la línea se mantenga en silencio por un tiempo de 4 caracteres a la tasa de 9600 bits por segundo, luego el sistema espera al primer byte 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 velocidad de transmisión actual.

Los caracteres recibidos, se guardan en una tabla creada en la RAM del PIC. Esta tabla se crea cada vez que llega un comando nuevo. El primer byte almacenado en la tabla corresponde a la dirección asignada a la DPU, el segundo byte es el código de la función; los siguientes bytes almacenados corresponden a datos necesarios para que la función se ejecute; por último están los dos bytes que corresponden al código de chequeo de error CRC .

Al procedimiento que realiza el algoritmo explicado hasta aquí se le ha denominado Count\_Timeout, su diagrama de flujo se puede observar en el Apéndice A.4.

Cuando la comunicación ha terminado, se procede a realizar con el mensaje el procedimiento que se necesite, según sea la función recibida en el segundo byte de de la tabla, y se traduce el comando a una orden en protocolo ASCII para enviarla a la DPU.

Después que 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 la 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.

La dirección del esclavo es muy importante 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.

Si se trata de la dirección cero todas las TCP la aceptarán como su propia dirección. El valor de la dirección propia de cada esclavo sólo se puede cambiar modificando el firmware de la TCP. Para esto es necesario volver a compilar el proyecto modbus.pjt y por último se debe volver a grabar el PIC16C74A con el archivo mdbus.hex.

## **6.2 Alcances y Limitaciones**

### **6.2.1 Hardware**

El PIC está limitado a una capacidad de memoria de 4 Kbytes en ROM, lo cual podría ser una limitación importante en el caso de que en el futuro se quisieran desarrollar funciones más complejas.

El Pic cuenta con 33 pines de entrada - salida de los sólo se están usando 4 en forma definitiva. Se pueden implementar algunos tipos de indicadores usando diodos led que indiquen el buen funcionamiento del sistema, o bien, que indiquen cuándo se da un error. 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**

Es importante hacer notar que al implementar el modo de funcionamiento en hyperterminal de Windows, (función 12) se tiene el acceso a todos los comandos con los que cuenta la DPU, de manera inmediata, sin que existan retardos de tiempo en traducciones de protocolos.

Por todas estas razones es que la función 12 es la más importante de las funciones que se hayan programado en el PIC puesto que brinda todos los comandos. La DPU cuenta además con sus propias indicaciones de error en caso de que se introduzcan datos no válidos.

## **CAPÍTULO 7**

### **CONCLUSIONES Y RECOMENDACIONES**

---

#### **7.1 Conclusiones**

- a.** La DPU es un dispositivo de protección que es utilizado en la red de distribución de la CNFL, y puede ser interrogado, tanto manualmente, como por un puerto de comunicaciones serie.
  
- b.** La DPU utiliza el protocolo de comunicación ASCII.
  
- c.** El sistema SCADA de la CNFL utiliza un protocolo de comunicación Modbus RTU para comunicarse con los esclavos.
  
- d.** El microcontrolador PIC16C74A tiene características de comunicación serial eficientes, varios puertos con pines de entrada-salida y un convertidor de analógico a digital.
  
- e.** 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.



- f. La tarjeta que convierte protocolos mejoró sus características tanto de software como de hardware según los objetivos planteados.
- g. El modo “broadcast” es utilizado para sincronizar los relojes de todas las DPU.
- h. El modo hyperterminal de Windows permite que desde una computadora personal sean utilizadas cualquiera de las funciones que brinda la DPU, para ésto conecta directamente las líneas de comunicación de la DPU con las que vienen del Sistema Maestro.
- i. La velocidad de comunicación del sistema fue aumentada hasta 9600 bits por segundo.

## **7.2 Recomendaciones**

- a. Se comprobó que el procedimiento de generación del código de chequeo de error, es incorrecto, por lo que se debe corregir para ser aplicado en todas las respuestas Modbus .
- b. La cantidad de funciones que se han creado hasta el presente, no caben todas en el espacio de 4 Kbytes de memoria ROM con que cuenta el PIC16C74A, por lo tanto se deberá utilizar un PIC con mayor capacidad de memoria.

## BIBLIOGRAFÍA

---

1. ABB. Buyer's Guide 98/99 Protection, monitoring and Control descriptive Literature. ABB. Coral Springs, USA. 1999
2. 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>
3. 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.
4. FieldNet AS. 2001. Modbus Protocol homepage (en línea). Consultado 14 feb. 2001. Disponible en [http://www.fieldnet.no/fbacad/chapter\\_1.htm](http://www.fieldnet.no/fbacad/chapter_1.htm)
5. Rojas Chinchilla, E. 2001. 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. Tesis Bach. Ing. Elec. Cartago, Costa Rica, Instituto Tecnológico de Costa Rica.10,20 p.
6. WinTECH Software, US. 1997. ModScan32 Application Description homepage (en línea). Consultado 16 feb. 2001. Disponible en <http://www.win-tech.com/html/modscan32.htm>

# APÉNDICES

---

## Apéndice A.1: Glosario

**Alimentador:** Circuito que alimenta a una subestación.

**Amperímetro:** Aparato que sirve para medir una corriente.

**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 de un mensaje se envía un carácter ASCII. Utiliza un chequeo de redundancia longitudinal para determinar errores en la transmisión.

**Modbus RTU:** Protocolo de comunicación donde en cada byte de un mensaje se envían 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.

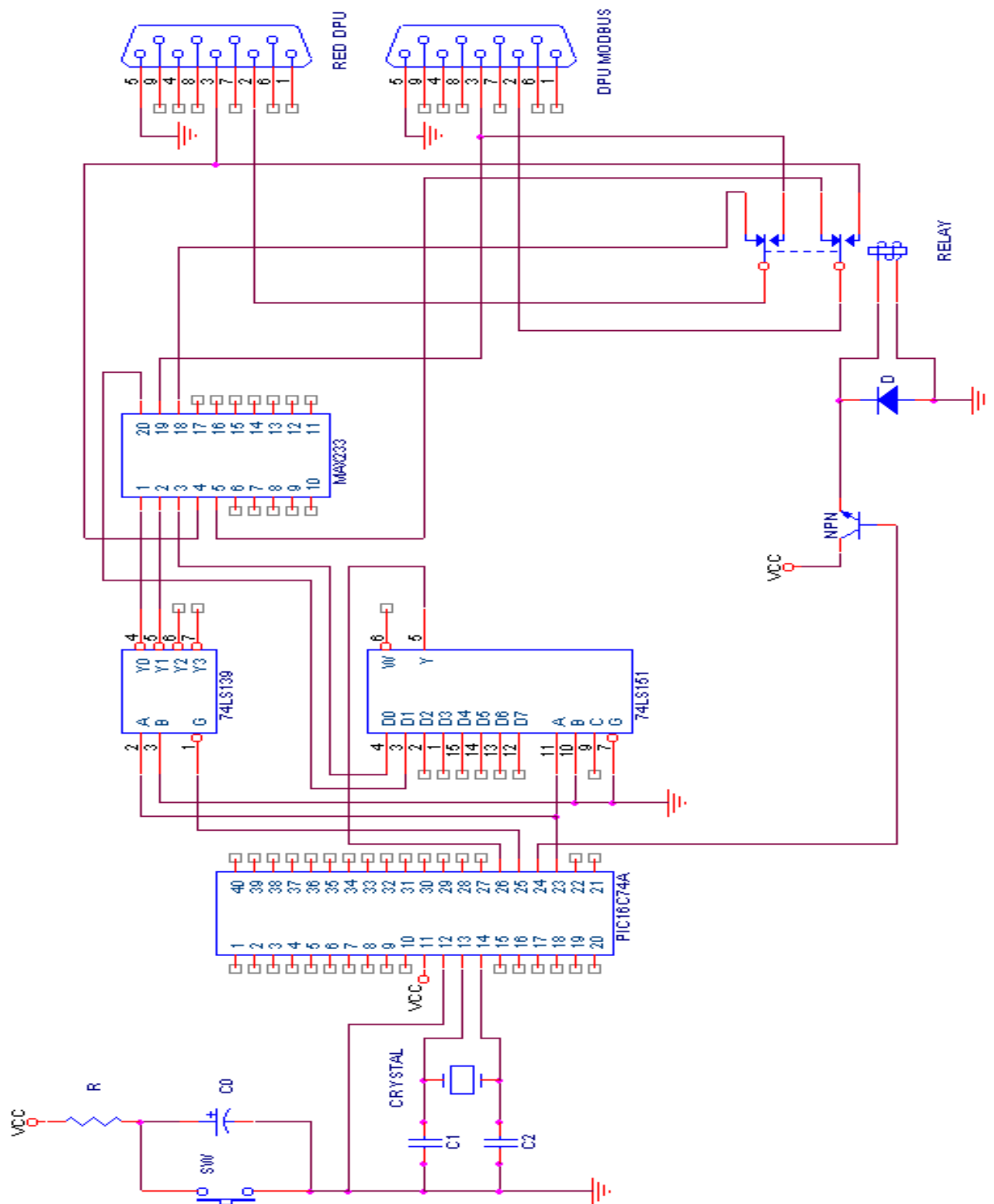
**SCADA:** Sistema de supervisión, control y adquisición de datos.

**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.

## Apéndice A.2: Diagrama esquemático de la TCP

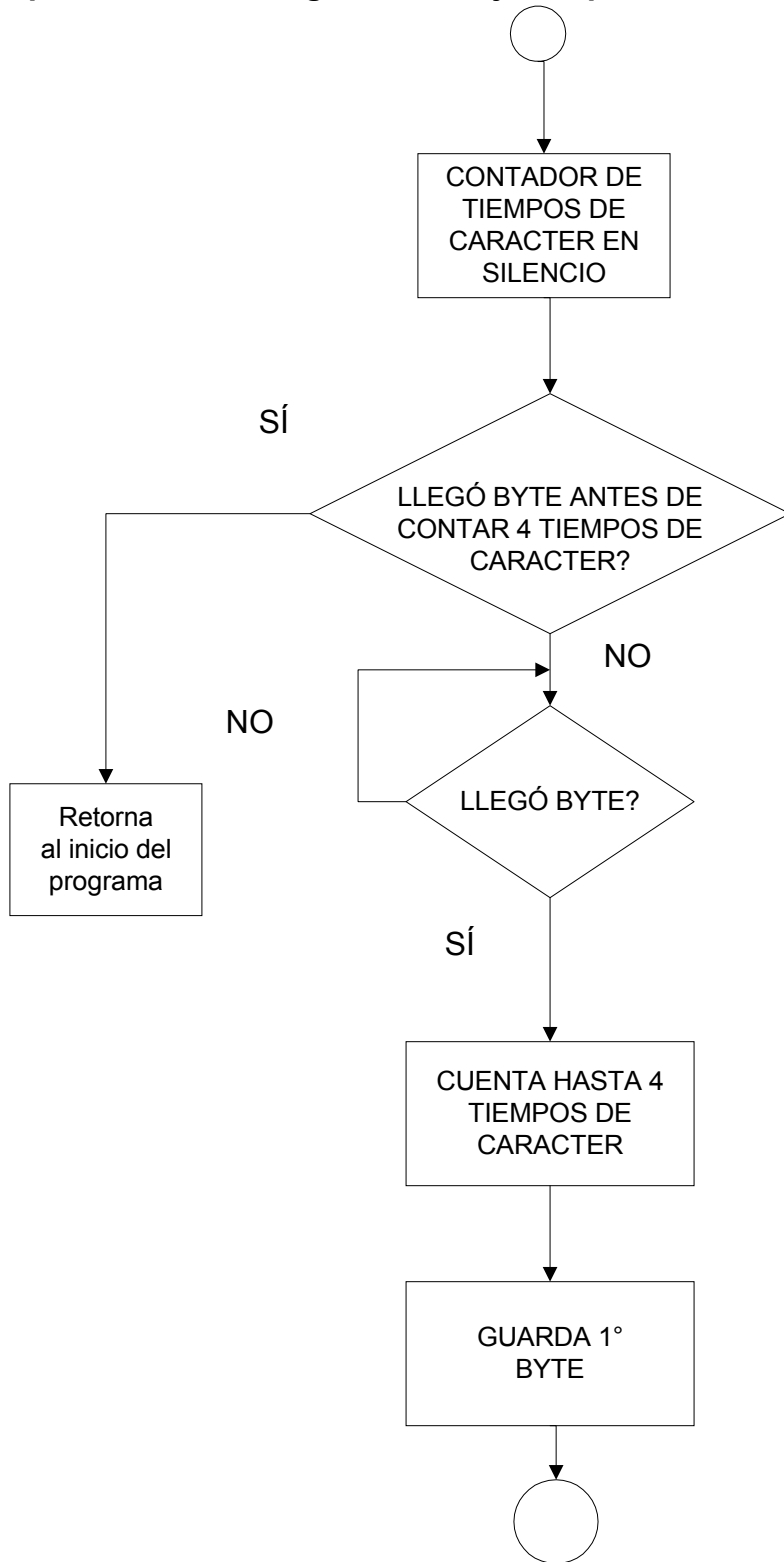


### Apéndice A.3: Comandos de la DPU implementados en la TCP

Comando	Función (HEX)
Modo broadcast	00
Tabla 51	11
Modo hyperterminal	12
Regresar a modo normal	13

**Nota :** En el Anexo B.4 se puede observar todas las funciones de la DPU a las que se tiene acceso por medio del modo hyperterminal, función 12.

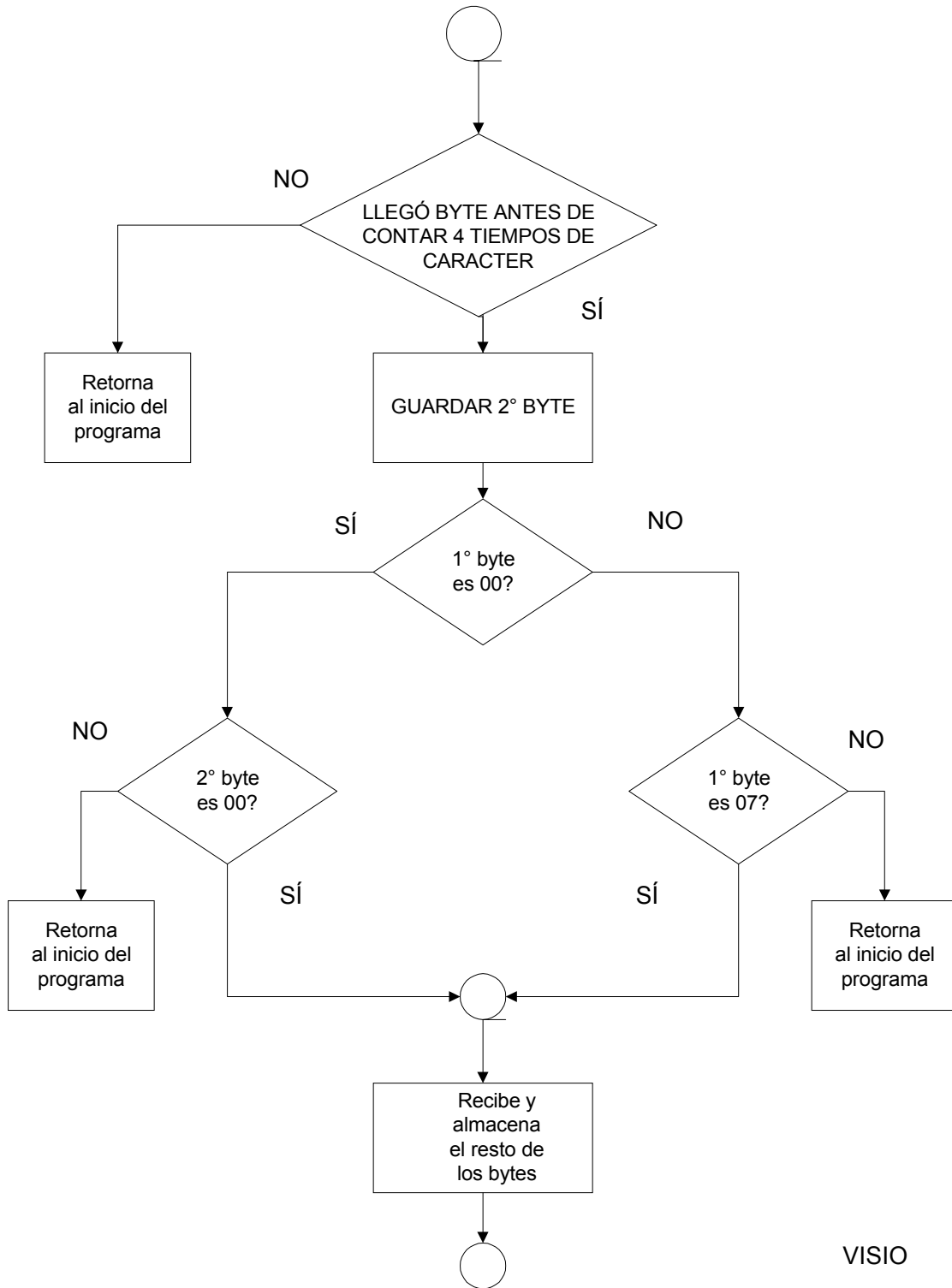
**Apéndice A.4: Diagrama de flujo del procedimiento Count\_Timeout**



VISIO 4.0



### Diagrama de flujo del procedimiento Count\_Timeout (Continuación)

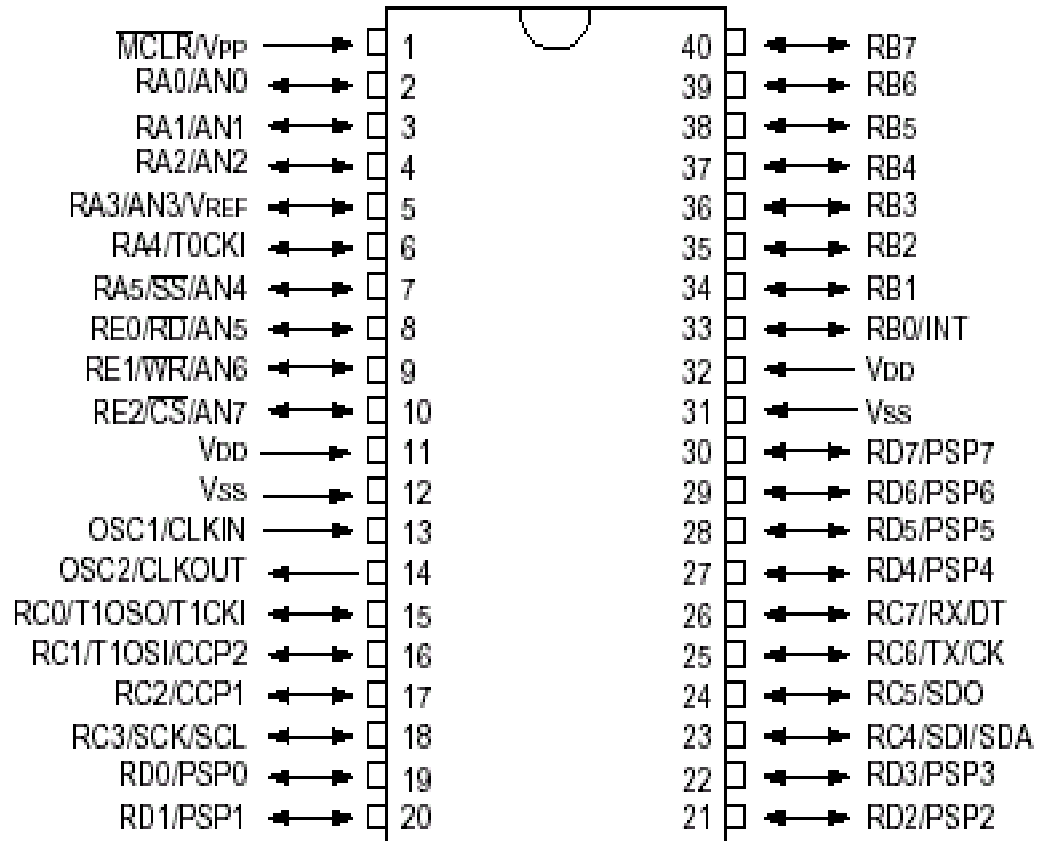


VISIO

# ANEXOS

---

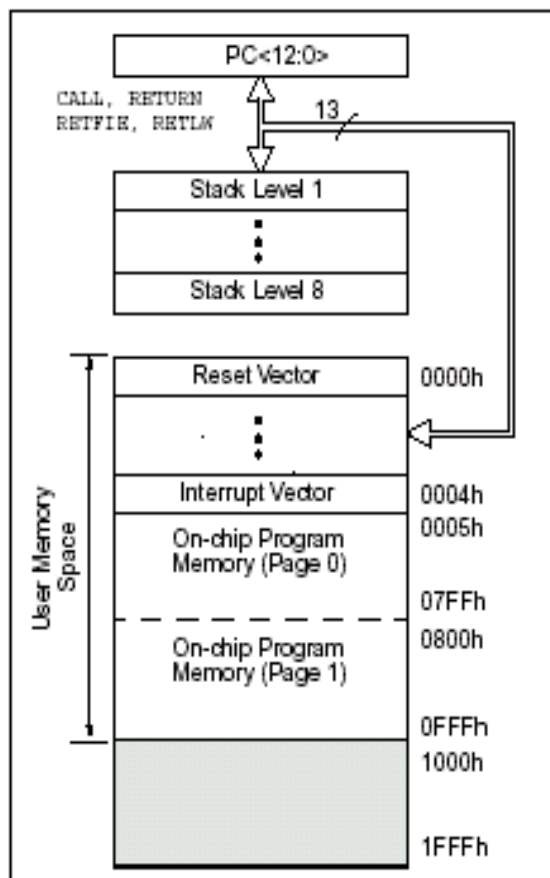
## Anexo B.1: Distribución de las patillas del PIC16C74A



Pegado con Paint



### Anexo B.3: Mapa de memoria ROM y stack del PIC16C74A



Pegado con Paint

## **Anexo B.4: Relay commands**

<b><u>Command Function</u></b>	<b><u>Serial port</u></b>
Display relay settings	5678
Display feeder CT ratios	9A
Display relay ID	89AB
Display/Edit relay settings	FF
Display/Edit baud rate	5B
Display/Edit clock	4567
Display/Edit demand meter time constant	5D
Change editing password	A1
Reset front panel targets	A2 or 3
Change trip/close password	A3
Display /Edit target mode	A4
Display/Edit A5 input mode	A5
Display/Edit 79 cutout Time	A6
Display/Edit cold-load pickup time	A7
Display/Edit phase inst. Vote mode	A8
Display/Edit lockout on 51N trip	A9
Display/Edit TCC output contact mode	A10
Display/Edit instantaneous cutout input mode	A11
Display/Edit failure-to-close time interval	A12
Display/Edit zone sequence coordination mode	B1
Display/Edit alternate minimum trip mode	B2
Display self-check status	51
Display contact input status	52
Display circuit breaker status	B0
Display trip "n" of event "m"	Dbmn

#### **Anexo B.4: Relay commands (continuación)**

<b><u>Command Function</u></b>	<b><u>Serial port</u></b>
Display details of last power failure	30
Display/Edit breaker operations counter	70
Display/Edit overcurrent-trip counter	71
Display current, phase A or B or C	D1,D2,D3
Display phase current	D123
Display neutral current	D0
Display all currents	33
Display demand current, phase A or B or C	DD1, DD2, DD3
Display demand current, neutral	DD0
Display peak demand current A	41
Display peak demand current B	42
Display peak demand current C	43
Display peak demand current N	40
Display peak demand times	45
Display accumulated KSI, phase A	81
Display accumulated KSI, phase B	82
Display accumulated KSI, phase C	83
Clear all peak demand currents	333
Clear peak demand current A	041
Clear peak demand current B	042
Clear peak demand current C	043
Clear KSI, phase A	081
Clear KSI, phase B	082
Clear KSI, phase C	083
Trip the circuit breaker	8747
Close the circuit breaker	7478
Test target lights or printer	8