Implementación de una Red MODBUS/TCP

Andrés F. Ruiz Olaya*
Asfur Barandica López**
Fabio G. Guerrero Moreno***

RESUMEN

En este artículo se describe el desarrollo de un sistema que permite la supervisión, mando y adquisición de datos de diversos controladores de procesos desde cualquier computador conectado a Internet, pasando por diversos medios físicos (Ethernet, RS232 y RS485) y protocolos de comunicación (Modbus/TCP, Modbus y protocolos ASCII). Se presentan los conceptos básicos del protocolo Modbus/TCP. Esta aplicación particular soluciona un problema del laboratorio de Automática de la Universidad del Valle.

Palabras Claves: TCP/IP, comunicaciones industriales, Modbus, redes de comunicación, Ethernet industrial.

* Ingeniero Electrónico - Candidato a Doctor - Instituto de Automática Industrial, en Madrid, España.
E-mail: anferol1@hotmail.com

** Ingeniero Electricista - Profesor - Escuela de Ingeniería Eléctrica y Electrónica - Facultad de Ingeniería - Universidad del Valle - Santiago de Cali, Colombia
E-mail: asfur@univalle.edu.co

*** M.Sc. Profesor - Escuela de Ingeniería Eléctrica y Electrónica - Facultad de Ingeniería - Universidad del Valle - Santiago de Cali, Colombia.
E-mail: fguerrer@univalle.edu.co

Fecha de recepción: Febrero 24 de 2004
Fecha de aprobación: Agosto 23 de 2004

ABSTRACT

This paper describes the development of a system for monitoring, control, and data acquisition of several process controllers from any computer with internet conecction by means of both different physical media (Ethernet, RS232 y RS485) and communication protocols (Modbus/TCP, Modbus, and ASCII protocols). Basic concepts about Modbus/TCP protocol are presented. This particular application solved a problem at the automatic control laboratory at Universidad del Valle.

Key Words: TCP/IP, industrial communications, Modbus, communications networks, industrial Ethernet.

1. INTRODUCCIÓN

En el área de las comunicaciones en entornos industriales la estandarización de protocolos es un tema en permanente discusión, donde intervienen problemas técnicos y comerciales. Cada protocolo está optimizado para diferentes niveles de automatización y en consecuencia responden al interés de diferentes proveedores. Por ejemplo Fieldbus Foundation, Profibus PA y HART, están diseñados para el control de procesos. En cambio DeviceNet y SDC están optimizados para los mercados de los dispositivos detectores, actuadores e interruptores, dispositivos discretos (on-off), donde el tiempo de respuesta y repetibilidad son factores críticos [1].

Cada protocolo tiene un rango de aplicación por fuera del cual disminuye su rendimiento y aumenta la relación costo/beneficio.

La prolongada ausencia de un estándar único para comunicaciones industriales, ha hecho que los múltiples protocolos existentes en esta área hayan perdido terreno ante la incursión de tecnologías de comunicación maduras a nivel de oficina pero emergentes en el nivel de planta, como Ethernet.

La aceptación mundial de Ethernet en los entornos administrativos y de oficina ha generado el interés de expandir su aplicación a la planta. Ethernet se está moviendo rápidamente hacia el mercado de los sistemas de control de procesos y la automatización para la interconexión de sensores y actuadores a nivel de campo, reemplazando de esta forma a los buses de campo en las industrias. Han surgido diversos protocolos para comunicación industrial sobre Ethernet. Sin embargo, no existe una capa de aplicación estándar con un modelo de objetos común. Modbus/TCP es un estándar *de-facto* ampliamente extendido y aceptado; existen otros protocolos para Ethernet a nivel industrial: EtherNet/IP (esencialmente objetos ControlNet y DeviceNet sobre TCP/IP y UDP), ProfiNet (combina el protocolo Profibus, OLE para control de procesos OPC y TCP/IP) y Fieldbus Foundation high-speed Ethernet HSE (coloca el protocolo H1 de Foundation Fieldbus sobre TCP/IP y añade OPC y el lenguaje XML). Es posible que con el aumento de velocidad de Ethernet (Fast Ethernet, Gigabit Ethernet) se pueda usar también en el manejo de aplicaciones críticas de control [2].

Los buses de campo son una forma especial de LAN (Local Area Network) dedicada a aplicaciones de adquisición de datos y comando de elementos finales de control sobre la planta. Típicamente operan sobre cables de par trenzado de bajo costo. A diferencia de Ethernet, donde no se puede garantizar determinismo sobre la llegada de paquetes, los diseñadores optimizan los buses de campo para el intercambio de mensajes cortos de comando y de control con alta confiabilidad y temporización estricta.

En aplicaciones industriales, Ethernet es usado en conjunto con TCP/IP (protocolos usados en Internet), suministrando un mecanismo de transporte de datos entre máquinas confiable y permitiendo interoperabilidad entre diversas plataformas. Usar TCP/IP sobre Ethernet a nivel de campo en la industria permite tener una verdadera integración con la red corporativa y de esta forma se puede ejercer un control estricto sobre la producción [3].

En este artículo se presentan los resultados obtenidos en la implementación de una red de control industrial utilizando Modbus/TCP, un protocolo estándar de instrumentación sobre Ethernet [4], la cual puede ser accedida a través de Internet o la Intranet local usando los protocolos TCP/IP. El protocolo Modbus/TCP se ha difundido ampliamente por ser abierto, lo cual le permite la comunicación con una gran diversidad de elementos industriales y por su sencillez.

En la sección 2 del presente artículo se exponen las principales características de Modbus/TCP, resaltando la manera en que se lleva a cabo su implementación. En la sección 3 se presentan las características de cada uno de los elementos que conforman la red Modbus/TCP implementada y el papel que desempeñan en el sistema completo. En la sección 4 se presenta una descripción breve sobre la forma en que se desarrollaron los programas y aplicaciones que corren sobre los diversos elementos de la red; además se explica la funcionalidad que brindan y la interacción entre los componentes de software. Finalmente, en la sección 5 se exponen los resultados y conclusiones del proyecto y propuestas de trabajo futuro en esta área.

2. PROTOCOLO MODBUS/TCP

Modbus/TCP es un protocolo de comunicación diseñado para permitir a equipos industriales tales como Controladores Lógicos Programables (PLCs), computadores, drivers para motores y otros tipos de dispositivos físicos de entrada/salida comunicarse sobre una red. Modbus/TCP fue introducido por Schneider Automation como una variante de la familia de protocolos MODBUS, ampliamente usada para la supervisión y el control de equipo de automatización. Específicamente el protocolo define el uso de mensajes MODBUS en un entorno intranet o internet usando los protocolos TCP/IP [5].

La especificación Modbus/TCP define un estándar interoperable en el campo de la automatización industrial, el cual es simple de implementar para cualquier dispositivo que soporte sockets TCP/IP. Todas las solicitudes son enviadas vía TCP sobre el puerto registrado 502 y normalmente usando comunicación half-duplex sobre una conexión dada. Es decir, no hay beneficio en enviar solicitudes adicionales sobre una conexión única mientras una respuesta está pendiente.

Modbus/TCP básicamente encapsula una trama MODBUS dentro de una trama TCP en una manera simple como se muestra en la Figura 1.

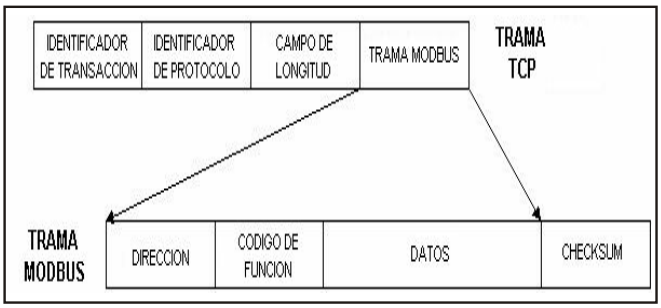

Figura 1. Esquema de encapsulación en Modbus/TCP

A continuación se citan sus principales características:

a. Mecanismo de conexión:
MODBUS es un protocolo maestro/esclavo en el que cada solicitud del maestro es tratada de forma independiente por el esclavo, sin relación con las anteriores. Esto facilita proveer transacciones de datos resistentes a rupturas, requiriendo mínima información de recuperación para mantener una transacción en cualquiera de los dos terminales.

De otro lado, las operaciones de programación esperan una comunicación orientada a la conexión, es decir, las máquinas de origen y de destino deben establecer un canal de comunicaciones antes de transferir datos. Este tipo de operaciones son implementadas de diferentes maneras por las diversas variantes de MODBUS (Modbus RTU, Modbus ASCII, Modbus PLUS).

En Modbus/TCP una conexión se establece inicialmente en la capa de aplicación y esta única conexión puede llevar múltiples transacciones independientes. Además, TCP permite establecer un gran número de conexiones concurrentes, de modo que el cliente (maestro) puede tanto re-usar una conexión previamente establecida como crear una nueva en el momento de requerir una transacción de datos.

En Modbus/TCP se usa el protocolo orientado a la conexión TCP en lugar del protocolo orientado a datagramas UDP. Se busca mantener el control de una transacción individual encerrándola en una conexión que pueda ser identificada, supervisada y cancelada sin requerir acción específica por parte de las aplicaciones cliente y servidor. Esto da al mecanismo una amplia tolerancia a cambios del desempeño de la red y permite que se puedan añadir fácilmente herramientas de seguridad tales como firewalls y proxies.

El monitoreo continuo de datos o "streaming data" no es muy eficiente con el protocolo Modbus/TCP, debido básicamente a la sobrecarga que impone el protocolo de transporte TCP. Esta sobrecarga se debe al servicio de entrega de datos confiable y el acuse de recibo para cada paquete transmitido, incrementándose considerablemente el tráfico en la red cuando se monitorea en todo momento un esclavo Modbus/TCP particular. Sin embargo, esto tiende a ser un problema menor a medida que Ethernet aumenta de velocidad. La solución para la supervisión de datos continuos sobre una red Ethernet es la utilización del protocolo de transporte UDP, pero se pierde confiabilidad.

b. Modelo de datos:
MODBUS basa su modelo de datos sobre una serie de tablas las cuales tienen características distintivas. Las cuatro principales son:

- ☑ Entradas discretas: bit único; suministradas por un sistema I/O (entrada/salida); de sólo lectura.
- ☑ Salidas discretas: bit único; alterable por un programa de aplicación; de lectura-escritura.
- ☑ Registros de entrada: 16 bits suministrados por un sistema I/O; de sólo lectura.
- ☑ Registros de salida: 16 bits, alterables por un programa de aplicación; de lectura-escritura.

La distinción entre entradas y salidas, y entre datos direccionables por bit y direccionables por palabra no implica algún comportamiento de la aplicación. Es aceptable y común considerar las cuatro tablas solapándose una con otra, si ésta es la interpretación más natural sobre la máquina (esclavo MODBUS) en cuestión.

c. Filosofía de longitud:
Todas las solicitudes y respuestas MODBUS están diseñadas en tal forma que el receptor pueda verificar que un mensaje está completo. Para códigos de función donde la solicitud y respuesta tienen longitud fija, basta el código de función. Para códigos de función llevando una cantidad variable de datos en la solicitud ó respuesta, la porción de datos estará precedida por un campo que representa el número de bytes que siguen.

Cuando MODBUS es transportado sobre TCP, se adiciona información de longitud en el prefijo (o encabezado) para permitir al receptor reconocer los límites del mensaje, así el mensaje haya sido dividido en múltiples paquetes para la transmisión. La existencia de reglas de longitud implícitas o explícitas y el uso de un código de chequeo de error CRC-32 (sobre Ethernet), resulta en una probabilidad muy pequeña de corrupción no detectada sobre un mensaje de solicitud o respuesta.

d. Ventajas del protocolo MODBUS/TCP:

- ☑ Es escalable en complejidad. Un dispositivo que tenga un propósito simple necesita implementar sólo uno o dos tipos de mensaje.
- ☑ Es simple para administrar y expandir. No se requiere usar herramientas de configuración complejas cuando se añade una nueva estación a una red Modbus/TCP.
- ☑ No es necesario equipo o software propietario

de algún vendedor. Cualquier sistema de cómputo con una pila de protocolos TCP/IP puede usar Modbus/TCP.

- ☑ Puede ser usado para comunicación con una gran base instalada de dispositivos MODBUS, usando productos de conversión los cuales no requieren configuración.
- ☑ Es de muy alto desempeño, limitado típicamente por las capacidades de comunicación del sistema operativo del computador. Se pueden obtener altas ratas de transmisión sobre una estación única y la red puede ser configurada para lograr tiempos de respuesta garantizados en el rango de milisegundos.

No existe una especificación precisa acerca del tiempo de respuesta requerida para una transacción sobre MODBUS o Modbus/TCP. Esto es debido a que se espera que Modbus/TCP sea usado en la más amplia variedad posible de situaciones de comunicación, desde sistemas I/O esperando temporización en milisegundos, a enlaces de radio de larga distancia con retardos de varios segundos.

En general, los dispositivos tales como PLC´s responderán a solicitudes de entrada en un tiempo que típicamente puede variar entre 20 y 200 ms. Desde la perspectiva del cliente, ese tiempo de respuesta debe ser extendido por los retardos de transporte a través de la red a un tiempo de respuesta razonable. Tales retardos pueden ser de milisegundos en una red de área local, hasta cientos de milisegundos para una conexión de red de área amplia (WAN).

Toda temporización (timeout) usada en un cliente debe ser mayor que el tiempo máximo de respuesta razonable, para así evitar una excesiva congestión en el dispositivo servidor o en la red, lo cual puede causar errores. En la práctica las temporizaciones usadas en aplicaciones de alto desempeño serán probablemente algo dependientes de la topología de la red y el desempeño esperado del servidor. Aplicaciones cliente no críticas en tiempo pueden con frecuencia dejar los valores de temporización al establecido por defecto en TCP, el cual reporta fallo en la comunicación después de varios segundos.

Los clientes pueden cerrar y re-establecer conexiones Modbus/TCP cuando el timeout ha expirado. Sin embargo, al retransmitir una solicitud es aconsejable establecer un timeout más grande para permitir al servidor recuperarse de una posible condición de falla.

3. DESCRIPCIÓN DEL HARDWARE DE LA RED

Los principales elementos de la red implementada (figura 2) son:

- ☑ Tarjeta TINI.
- ☑ Tarjeta CPU08.
- ☑ PLC DL05 de Koyo.
- ☑ Controladores 452 PLUS.

a. Tarjeta TINI:

La Tiny InterNet Interface (TINI) es una plataforma desarrollada por Dallas Semiconductor que suministra un medio simple, flexible y económico para diseñar una extensa variedad de dispositivos hardware capaces de conectarse directamente a redes corporativas y locales. Las características de la plataforma son expuestas al desarrollador de software a través de un conjunto de interfaces de programación de aplicaciones (APIs) en Java, brindando un entorno de programación orientado a objetos y facultando al programador en la creación de aplicaciones utilizando las ventajas que ofrece el lenguaje Java.

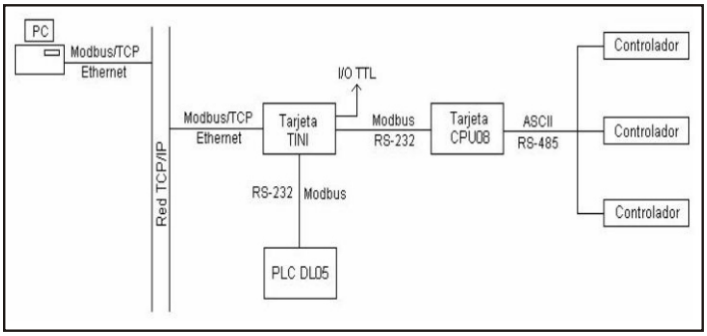

Figura 2. Diagrama en Bloques del Sistema

La plataforma TINI dispone de entradas y salidas que permiten obtener información proveniente de sensores y establecer el estado de actuadores; los datos son transferidos hacia/desde el exterior de la TINI mediante el enlace Ethernet. Esta capacidad de interconectividad de red de la TINI permite a cualquier dispositivo conectado a ella una interacción con sistemas remotos y usuarios a través de aplicaciones de red estándar, tales como browsers Web.

En la tarjeta TINI se ejecutan el servidor Modbus/TCP, el servidor Web y el software para actuar como gateway de otros esclavos, como el PLC DL05 y la tarjeta CPU08.

Las características de la tarjeta TINI relevantes en este proyecto son:

- ☑ 512 kbytes de memoria flash para código del sistema crítico y 512 kbytes de memoria SRAM no-volátil, expandible a 1 Mbyte.
- ☑ Controlador Ethernet 10Base-T.
- ☑ Doble puerto serial, doble controlador CAN y doble interfaz de red 1-Wire.
- ☑ El sistema operativo que reside en memoria flash se puede actualizar bajando las nuevas versiones de la página del fabricante.

La Figura 3 muestra un modelo de uso en el cual la TINI es empleada como un conversor de protocolos (o enlaces) entre un dispositivo embebido y una red Ethernet.

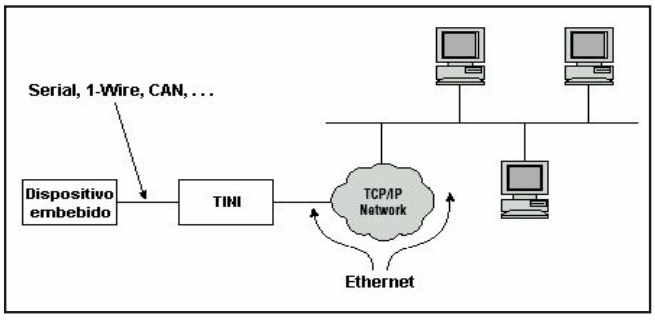

Figura 3. La TINI como un conversor de protocolos

La aplicación Java corriendo sobre la TINI desempeña la tarea de comunicación con el dispositivo en su lenguaje nativo (usando un protocolo de comunicación específico del dispositivo) y presenta los resultados a sistemas remotos alcanzables a través de una red TCP/IP.

b. Sistema embedido CPU08:

La tarjeta CPU08 (Figura 4) es un sistema embebido desarrollado en la Universidad del Valle basado en el microcontrolador AT89C52 de Atmel.

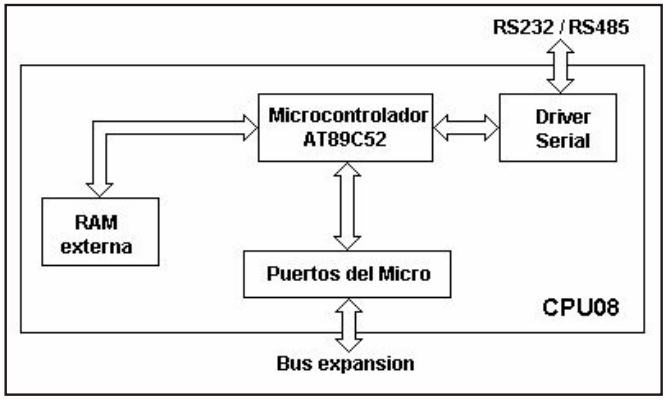

Figura 4. Diagrama general sistema CPU08

En esta tarjeta reside el software que realiza las funciones de esclavo MODBUS y el software que maneja la red RS-485 de controladores a través de un protocolo propietario codificado en ASCII. La CPU08 cuenta con 8 kbytes de memoria de programa interna, 256 bytes de memoria RAM interna, hasta 128 kbytes de memoria externa para programa o datos y un puerto serial RS-232 / RS-485.

La tarjeta CPU08 se programa directamente en el lenguaje propio del microcontrolador AT89C52. Opcionalmente, es posible realizar la programación en lenguaje C y utilizar un compilador cruzado que traslade el código fuente C al lenguaje del microcontrolador. La utilización del lenguaje C para realizar la programación depende de los requerimientos de desempeño y tamaño de código que se necesiten satisfacer.

c. PLC KOYO DL05:

Este dispositivo PLC ofrece características

convenientes para integrarlo en la red mostrada en la Figura 2, ya que proporciona un puerto serial que permite al PLC ser configurado como un maestro o un esclavo MODBUS. En esta aplicación el PLC se comporta como un dispositivo esclavo MODBUS. El diagrama de bloques del PLC Direct DL05 de Koyo se muestra en la Figura 5.

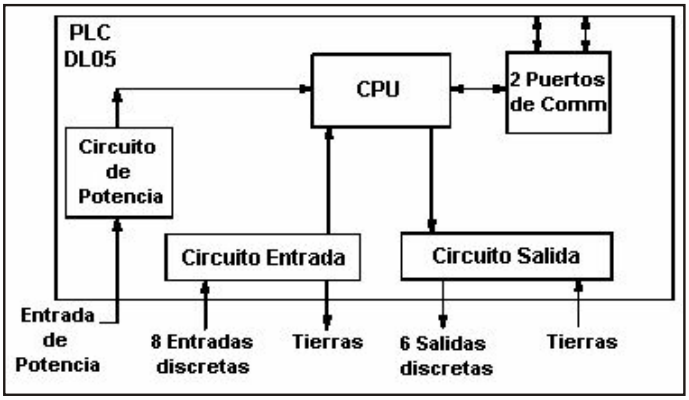

Figura 5. Diagrama en bloques PLC DL05 de Koyo

D. Controladores de lazo 452 PLUS:
Los controladores 452 Plus son dispositivos que permiten realizar las tareas propias para el control de un lazo en un proceso, mediante diferentes estrategias de control, entre ellas PID. La configuración de los parámetros del controlador (setpoint, constantes proporcional, integral y derivativa, etc.) pueden ser establecidos localmente a través de su panel frontal, o remotamente a través de una interfaz serial RS-232, RS-423A, RS-422A o RS-485.

La comunicación remota con el controlador se desarrolla serialmente usando un protocolo asíncrono codificado en ASCII de 7 bits con paridad par, un bit de parada y un bit de inicio, a 9600, 4800, 1200 ó 300 bps. El protocolo de comunicación está definido en dos *capas* o niveles; la primera define el *wrapping* del mensaje (la delimitación de su inicio y su fin), mientras la segunda describe el contenido del mismo.

4. DESCRIPCIÓN DEL SOFTWARE

El software de la red Modbus/TCP (Figura 2) está desarrollado a través de los siguientes módulos claramente identificables:

- ☑ Interfaz gráfica para acceso vía Web.
- ☑ Servidor Modbus/TCP en la tarjeta TINI.
- ☑ Servidor Web en la tarjeta TINI.
- ☑ Esclavo MODBUS en la tarjeta CPU08.
- ☑ Programación de la tarjeta CPU08 como maestro de una red RS-485 de controladores.
- ☑ Configuración del PLC DL05 como un esclavo MODBUS.
- ☑ Configuración de los controladores 452 Plus.

a. Interfaz para acceso web:
La interfaz permite leer y escribir remotamente registros o posiciones discretas sobre los diversos elementos que componen la red Modbus/TCP a través de una red TCP/IP como por ejemplo Internet.

La herramienta se ha desarrollado como un applet de Java de forma que un usuario puede acceder la red Modbus/TCP desde cualquier lugar vía Internet, utilizando solamente un browser o navegador con soporte Java independientemente de la plataforma en que se encuentre. El applet desarrollado posee las siguientes características:

- ☑ Control de acceso de usuarios; sólo personas autorizadas pueden acceder la red Modbus/TCP.
- ☑ Monitoreo de variables (valor medido, entradas discretas, etc.) .
- ☑ Control de variables (setpoint, salidas discretas, etc.).

Todas las operaciones de supervisión y control de variables se realizan utilizando el protocolo Modbus/TCP, por tanto el applet debe codificar las solicitudes según este estándar.

Además de proveer una interfaz gráfica para interacción con el usuario, el applet también debe comunicarse con un servidor Modbus/TCP. Por tanto, como parte integral del applet se ha implementado un cliente Modbus/TCP.

El sistema de acceso solicita un nombre de usuario (login) y una clave (password) para obtener el panel de control que da acceso a la red Modbus/TCP.

El panel de control provee elementos como botones, campos de texto, componentes gráficos, etc., para visualizar el contenido o el estado de los diferentes registros y datos discretos que poseen los esclavos que conforman la red Modbus/TCP. Además, desde el panel es posible cambiar el contenido de determinados registros y valores discretos. En la Figura 6 puede observarse el aspecto del panel de control que se carga en el navegador.

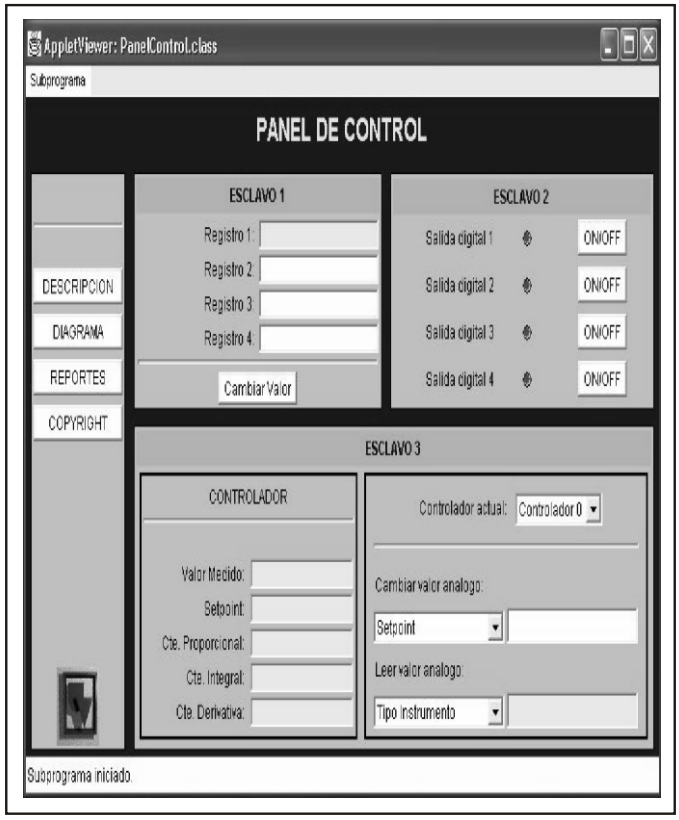

Figura 6. Vista del panel de control

La información que se despliega en el panel de control se actualiza constantemente; esto se realiza a través de un thread independiente que se comunica con los elementos remotos de la red, enviándoles solicitudes Modbus/TCP de lectura.

b. Servidor Modbus/TCP en la TINI:
El protocolo Modbus/TCP está basado en el paradigma cliente/servidor: el proceso servidor acepta una petición desde la red, ejecuta una acción basado en la petición y devuelve el resultado al solicitante; un programa se convierte en cliente cuando envía una petición al servidor y espera una respuesta.

El programa servidor para Modbus/TCP reside en la tarjeta TINI y se desarrolló en Java. La implementación del servidor hace uso de la programación orientada a objetos y multithread para aceptar solicitudes concurrentes. La estructura de clases y parte de la implementación está basada en el proyecto **jmodbus** [6].

El objetivo del proyecto **jmodbus** es proveer librerías Java para permitir a dispositivos basados en este lenguaje comunicarse como maestros o esclavos a través de Modbus/RTU, Modbus/ASCII o Modbus/TCP. El código es abierto y está diseñado para correr sobre dispositivos con poca memoria.

La aplicación desarrollada se comporta como un servidor (o esclavo) Modbus/TCP, pero también actúa como un gateway de forma tal que es capaz de direccionar solicitudes Modbus/TCP, cuyo destino son otros esclavos conectados a los puertos seriales de la TINI, con el protocolo Modbus RTU.

c. Sevidor Web:
El protocolo HTTP (HyperText Transfer Protocol, protocolo de transferencia de hipertexto) que constituye la base de la World Wide Web, está basado en el modelo cliente/servidor y posee el puerto registrado 80 de TCP.

El servidor Web que corre en la tarjeta TINI básicamente implementa la función GET que se define en la especificación del protocolo HTTP.

El servidor permite manejar múltiples conexiones concurrentemente; el programa se mantiene escuchando en el puerto de HTTP y cuando llega una petición GET, se crea un proceso hijo que

maneja esa transacción particular y éste devuelve una página HTML (HyperText Markup Language) al cliente.

La página HTML devuelta por el servidor Web al cliente contiene un applet, el cual implementa la interfaz gráfica de usuario que permite acceso remoto desde Internet. Los applets son un tipo de aplicaciones que Java permite crear, que se mantienen (o residen) en el servidor Web, son transportadas a través de Internet, instaladas automáticamente en la máquina cliente y se ejecutan localmente.

d. Esclavo Modbus en el sistema embebido CPU08:
El programa residente en la CPU08 debe realizar las funciones del protocolo MODBUS *escribir múltiples registros* y *leer múltiples registros*. Para la aplicación a la que está destinada la tarjeta CPU08, sólo se requieren estas dos funciones ya que el programa básicamente redirecciona la solicitud (de lectura o escritura) a una subred de controladores RS-485, los cuales manejan su propia sintaxis de mensajes. La CPU08 actúa como interfaz entre el protocolo MODBUS y el protocolo codificado en ASCII de los controladores 452 Plus.

El programa de la CPU08 efectúa la conversión de protocolos de forma que la solicitud MODBUS de lectura de registros se asocia con el tipo de mensaje RA (leer variable análoga) del controlador 452 Plus; así mismo, la solicitud MODBUS de escritura de registros se asocia con el tipo de mensaje SA (escribir en localización análoga) del controlador 452 Plus.

Dado que la tarjeta CPU08 posee un solo puerto RS-232 ó RS-485 y que se requieren dos puertos de comunicación serial, se implementó un puerto por software utilizando dos pines de propósito general que simulan las señales Rx y Tx de una UART.

El puerto de comunicaciones RS-485 que tiene la tarjeta se utiliza para establecer la comunicación con los controladores 452 Plus. Los datos Intercambiados entre la CPU08 y los controladores son de tipo decimal codificados en ASCII; en la CPU08 se debe cambiar la codificación de los datos de acuerdo al estándar IEEE754 que define el formato para el almacenamiento y la transmisión de datos flotantes, cada uno de los cuales se debe tratar como dos registros holding según el protocolo MODBUS.

e. Maestro de la red de controladores:
El maestro de la red RS-485 de controladores 452 Plus se programa en la tarjeta CPU08. Para la aplicación a la que está destinada la CPU08, existen dos tipos de mensajes distintos para comunicarse con los controladores 452 Plus: un mensaje para leer una variable análoga y mensaje para escribir en una variable análoga.

La tarjeta CPU08 envía un mensaje de solicitud y entonces espera por la respuesta. Sin embargo, es posible que el controlador 452 Plus no se encuentre encendido, o no reciba el mensaje por alguna razón. Por tanto en el programa de la CPU08 se establece un timeout cada vez que se envíe una trama al controlador, para evitar que posiblemente se quede esperando indefinidamente. Si se completa en timeout, la CPU08 arma una respuesta de excepción y la retorna al dispositivo maestro.

f. Configuración del PLC DL05 como esclavo MODBUS:
La comunicación del PLC como dispositivo esclavo puede hacerse a través de cualquiera de los dos puertos de comunicación serial que posee. La comunicación con el PLC se establece con el protocolo MODBUS en modo RTU. Los códigos de función MODBUS soportados por el PLC DL05, determinan si el acceso es de lectura o escritura y si el acceso es a un punto de datos simple o a un grupo de ellos. Para que puedan realizarse operaciones de escritura de registros o de datos discretos sobre el PLC a través del protocolo MODBUS RTU, el PLC debe encontrarse en modo de operación TERM.

g. Configuración de los controladores 452 PLUS:

La interfaz serial RS-485 de los controladores 452 Plus se presenta físicamente a través de 4 pines. Las señales Tx- y Tx+ conforman el canal diferencial de transmisión y las señales Rx- y Rx+ conforman el canal diferencial de recepción. Para realizar la conexión entre la tarjeta CPU08 y la subred RS-485 de controladores 452 Plus, se unen las señales Tx+ y Rx+ de cada uno de los controladores con la línea A del driver RS-485 de la CPU08. Igualmente se unen las señales Rx- y Tx- de cada uno de los controladores con la línea B del driver RS-485 de la CPU08. A nivel lógico para la comunicación con los controladores 452 Plus se emplea el modo CRL, debido a que sólo en este modo se habilita la comunicación con más de un controlador, lo cual resulta apropiado para la implementación de la subred RS-485 dentro de la red Modbus/TCP.

5. CONCLUSIONES

Modbus/TCP permite supervisar controladores o PLCs distribuidos alrededor de la planta haciendo posible, por ejemplo, que un sólo operario pueda atender remotamente diversos procesos simultáneamente desde un mismo puesto de trabajo. Además del monitoreo tradicional de variables es posible cambiar los parámetros operativos individuales de los controladores.

Con la implementación en éste y otros proyectos [7] del protocolo MODBUS y Modbus/TCP en las tarjetas CPU08 y TINI respectivamente, se evidencia la facilidad y flexibilidad del uso de este protocolo. Ésta es quizás la principal razón de su alta difusión en entornos industriales. Se demostró su capacidad de operación en red e interoperabilidad, de forma que desde clientes Modbus/TCP de diferentes fabricantes fue posible leer y escribir registros y datos discretos sobre los diversos elementos que conforman la red.

En la red implementada se integraron los protocolos Modbus/TCP, MODBUS y ASCII y las interfaces Ethernet, RS-232 y RS-485, demostrando que sistemas de control de procesos ya instalados pueden adaptarse para ser supervisados y controlados desde una red TCP/IP (Internet o intranet local) utilizando el estándar Modbus/TCP.

La plataforma TINI provee un entorno conveniente y apropiado para este tipo de aplicaciones por sus facilidades para desarrollar aplicaciones Java.

Como una proyección futura del presente trabajo se busca integrar en la tarjeta TINI el protocolo XML (eXtensible Markup Languaje), un lenguaje de gran crecimiento que ya abarca muchos campos en servicios de Web y el cual ya incursionó en el área industrial para el intercambio de información. Con la tarjeta TINI, básicamente este estándar se utilizaría para que una base de datos con soporte XML se comunique con ella y acceda continuamente a la información proveniente de los procesos. Esta información es almacenada en las tablas internas de la base de datos y de esa forma es posible llevar registros, archivos históricos, tendencias, gráficos, reportes, etc. del comportamiento de las variables que se requieran.

La interfaz gráfica de usuario para la supervisión y control de la red Modbus/TCP desde Internet que se desarrolló como un applet de Java también es posible enfocarla desde un punto de vista diferente: el mecanismo de interacción entre al usuario en la Web y la plataforma TINI se puede realizar a través de la utilización de servlets de Java, los cuales residirían y se ejecutarían sobre la TINI.

6. BIBLIOGRAFÍA

[1]. Distéfano, Mario. *Comunicaciones en entornos industriales*. Disponible en línea en http://fing.uncu.edu.ar/investigacion/institutos/IAEI/Cursos2.htm

[2]. Malizia, Sam. *Moving Ethernet to plant floors*, Disponible en línea en: http://www.isa.org/journals/ic/feature/1,1162,541,00.html

[3]. Disponible en línea en: http://www.modbus.org/modbus_tcp_new.htm

[4]. Schneider Automation *Modbus/TCP protocol specification*. Disponible en línea en: http://www.modicon.com/openmbus.

[5]. Disponible en línea en: http://www.modbus.org

[6]. Disponible en línea en: http://jmodbus.sourceforge.net

[7]. Escobar, A.M.; Sánchez, H.A. *Implementación de una red inalámbrica usando el protocolo Modbus*. Universidad del Valle, Cali, 2001.