PROCESADOR DE COMUNICACIONES MODBUS

JULIO AUGUSTO GELVEZ FIGUEREDO
Ingeniero Electricista, MPE.
Profesor Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones
Universidad Industrial de Santander
jagelvez@uis.edu.co

JORGE ELIÉCER DUQUE PARDO
Ingeniero Electricista
Profesor Universidad Tecnológica de Bolivar
jduque@unitecnologica.edu.co

Fecha Recepción: 24 de agosto de 2006
Fecha Aceptación: 21 de noviembre 2006

RESUMEN

El uso de los buses de campo se ha constituido en un elemento fundamental en el desarrollo de la automatización de procesos, facilitando el intercambio de datos entre controladores, la distribución de los procesos y la supervisión de los mismos. Este artículo presenta los requerimientos y la metodología necesaria para la implementación del protocolo Modbus, en un microcontrolador de la familia 68 de Motorola. Para evaluar la metodología se llevó a cabo la implementación del protocolo MODBUS en el microcontrolador HCS12 de 16 bits, en el cual se programó una pequeña aplicación y se conecto a una red en donde el maestro es un programa en LabView corriendo en un PC y se verificó el correcto funcionamiento mediante un sniffer (supervisor) Modbus programado en LabView y ejecutándose en otro PC.

PALABRAS CLAVES: *Redes Industriales, buses de campo, Modbus, protocolos de comunicación, automatización industrial*

ABSTRACT

Field-buses constitute a fundamental element in the development of processes automation. They facilitate the exchange of data among programmable logical controllers, distributed processes and the supervision system. In this article the requirements and the necessary methodology for the implementation on a microcontroller of the Modbus protocol are presented. In order to evaluate this methodology, the protocol was implemented in the 16 bits HCS12 microcontroller. A simple application was programmed and the microcontroller was connected to a net where the master is a LabView program running in a PC. The correct performance was verified by means of a Modbus Sniffer programmed in LabView.

KEYWORDS: *Industrial Networks, field buses, Modbus, communication protocols, Industrial Automation*

INTRODUCCIÓN

En la industria, durante muchos años se ha empleado la tecnología analógica de señales de 4 a 20 mA, para la comunicación entre los dispositivos de campo (sensores y actuadores) y sus correspondientes sistemas de control del proceso.

Esta tecnología está bastante extendida, pero tiene inconvenientes en los sistemas de control distribuido debido a factores tales como: elevado número de conductores para llevar las señales, susceptibilidad a interferencias electromagnéticas y la necesidad de utilizar barreras en zonas de seguridad intrínseca.

Por el contrario, la tecnología de *bus de campo*, utiliza señales digitales para transmitir los datos entre los dispositivos de campo y sus respectivos sistemas de control, lo cual le confiere mayor inmunidad a las interferencias y la posibilidad de utilizar menos conductores.

"Un bus de campo es un sistema de comunicación digital, serial y multipunto para comunicación de bajo nivel destinado a equipos de control de procesos industriales y dispositivos de instrumentación tales como actuadores, sensores y controladores locales".[1]

Debido a que la transmisión de los datos en un bus de campo se realiza en forma serial, normalmente se utilizan uno o dos pares de conductores, lo cual reduce grandemente el cableado de los dispositivos de control. Por otra parte esta característica permite el uso de fibra óptica o de medios inalámbricos para el transporte de datos, siendo bastante útil cuando se quiere aumentar la seguridad contra interferencias electromagnéticas.

Además, cuando se utilizan instrumentos de campo digitales se dispone de una cantidad mucho mayor de datos; los transmisores "inteligentes", pueden entregar información referente al estado y la configuración del dispositivo.

Entre los dispositivos que pueden conectarse a buses de campo están los instrumentos de medición de flujo, presión, temperatura y nivel, como también equipos con puertos de comunicación tales como los analizadores, RTUs *(Remote Terminal Unit)*, controladores lógicos programables, controladores de procesos, estaciones remotas de Entrada/ Salida, unidades de almacenamiento de datos, controladores de motores e interfaces hombre-máquina.

Fabricantes de equipos para la automatización implementan buses de campo que responden a sus necesidades particulares. Algunos son de carácter propietario y otros son de uso tan extenso que pueden considerarse estándares de facto, como es el caso de Modbus.

Protocolo Modbus

Modbus es un protocolo de comunicación industrial que apareció en 1979 como un bus para transmitir y recibir datos de control entre controladores e instrumentos de campo en forma serial, mediante una topología de maestro/ esclavo.

Las especificaciones del protocolo Modbus se encuentran disponibles al público y está reconocido por la IEC como una especificación públicamente disponible (*Public Available Specification*) bajo la designación IEC PAS 6203. Actualmente es soportado por la organización independiente Modbus-IDA.

La designación Modbus no corresponde propiamente a un estándar de red que incluye todos los aspectos desde el nivel físico hasta el de aplicación, sino a un protocolo de mensajes, posicionado en la capa de aplicación o nivel 7 del modelo OSI (Open System Interconnectio), tal como se muestra en la Figura 1

Modbus es un protocolo de comunicación maestro/esclavo entre dispositivos conectados sobre una red tipo bus, con transmisión serial asíncrona sobre uno de los siguientes medios: par trenzado, fibra óptica y radio.

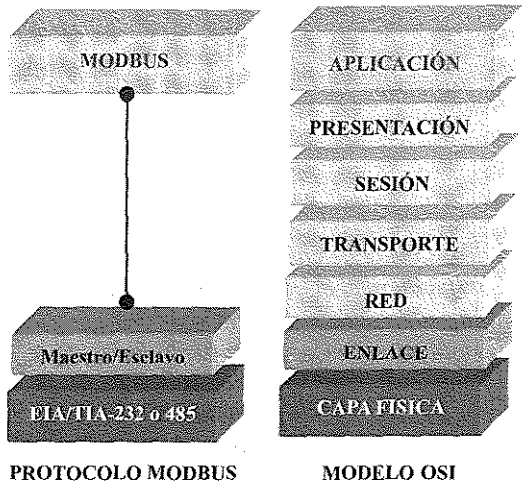

Figura 1. Capas del protocolo Modbus

En este artículo se presenta la implementación del protocolo Modbus para transmisión serial, el cual se encuentra completamente especificado en el documento PI–MBUS–300 Rev. J. [3]

Las principales características de este protocolo se presentan a continuación:

ESTRUCTURA DE RED

Medio Físico

El medio físico de conexión puede ser un bus semidúplex (*half duplex*) (RS-485 o fibra óptica) o dúplex (*full duplex*) (RS-422, BC 0-20mA o fibra óptica).

La comunicación es asíncrona y las velocidades de transmisión previstas van hasta los 19.200 baudios. La máxima distancia entre estaciones depende del nivel físico, pudiendo alcanzar hasta 1200 m sin repetidores.

Acceso al Medio

La estructura lógica es del tipo maestro-esclavo, con acceso al medio controlado por el maestro. El número máximo de estaciones previsto es de 63 esclavos más una estación maestra.

[1]IEC (Internacional Electrotechnical Commision) e ISA (Instrument Society of America)

Los intercambios de mensajes pueden ser de dos tipos:

- **Intercambios punto a punto.** Consta siempre de dos mensajes: una demanda del maestro y una respuesta del esclavo, el cual puede ser simplemente un reconocimiento (*acknowledge*).

- **Mensajes difundidos.** Estos consisten en una comunicación unidireccional del maestro a todos los esclavos. Este tipo de mensajes no tiene respuesta por parte de los esclavos y se suelen emplear para enviar datos comunes de configuración.

DESCRIPCIÓN DEL PROTOCOLO

La codificación de datos dentro de la trama puede hacerse en modo ASCII o puramente binario, según el estándar RTU (*Remote Transmission Unit*).

En cualquiera de los dos casos, cada mensaje obedece a una trama que contiene cuatro campos principales: Dirección, función, datos y chequeo de errores.

En los dos modos (RTU ó ASCII), un mensaje Modbus es enviado por el Maestro en un formato (*frame*) que tiene unos campos conocidos como de comienzo y terminación. Estos permiten a los dispositivos de recepción reconocer el comienzo del mensaje, leer el campo de dirección, determinando a que unidad va dirigida; además, facilita conocer cuando está completo el mensaje. Se pueden detectar mensajes parciales y generar códigos de error como resultado.

Formato ASCII

En modo ASCII, los mensajes comienzan con "dos puntos" (":" o carácter ASCII 3AH), y terminan con el par de caracteres "retorno de carro – salto de línea" (CR-LF) (ASCII 0DH y 0AH).

Los caracteres permitidos en la transmisión para todos los demás campos son 0-9, A-F (Hexadecimal).

En la Fig. 2 se muestra un formato de mensaje ASCII.

Inicio	Dirección	Función	datos	Chequeo LRC	Final
1 carácter :	2 caracteres	2 caracteres	N caracteres	2 caracteres	2 caracteres CR -LF

Figura 2. Formato de mensaje ASCII

Las unidades conectadas vigilan la red continuamente para detectar el carácter (:). Cuando se recibe, cada dispositivo decodifica el siguiente campo (el campo de dirección) para averiguar si corresponde a su dirección.

Se permiten intervalos de hasta un segundo entre caracteres dentro del mensaje. Si transcurre un tiempo mayor, el dispositivo receptor supone que ha ocurrido un error.

Formato RTU

En modo RTU, empiezan los mensajes con un intervalo de silencio de al menos 3,5 veces un carácter. Esto se realiza esperando un tiempo múltiplo de la velocidad en baudios que se está utilizando en la red. Luego se transmite el primer campo, dirección del dispositivo.

Los caracteres permitidos para todos los campo son 0-9, A-F hexadecimal. Los dispositivos conectados vigilan el bus de red continuamente, incluso en los intervalos de silencio. Cuando se recibe el primer campo (el campo de dirección), cada unidad lo decodifica para averiguar si es el dispositivo direccionado.

Después del último carácter transmitido se intercala un intervalo de tiempo equivalente, al menos, a 3.5 veces el tiempo de un carácter (TC) para marcar el fin del mensaje. Después de este intervalo puede comenzar un nuevo mensaje.

El formato de mensaje completo tiene que transmitirse conjuntamente. Si se produce un intervalo de más de 1.5 veces un carácter antes de la terminación del formato el dispositivo receptor asume el mensaje como incompleto y supone que el byte n, siguiente será el campo de dirección de un nuevo mensaje.

Igualmente, si un nuevo mensaje comienza antes de 3.5 veces el tiempo de un carácter el segundo mensaje se considerará como continuación del anterior. Esto provocará un error, ya que el valor del campo CRC final no será válido por los dos mensajes combinados. A continuación se muestra el formato del mensaje.

Inicio	Dirección	Función	datos	Chequeo CRC	Final
3.5 T.C.	8 bits	8 bits	N*8 bits	1 Byte	3.5 T.C.

Figura 3. Formato de mensaje RTU

Campo de verificación de errores

- **ASCII**

En modo ASCII la verificación de errores en el mensaje completo (excluyendo los caracteres de inicio y final) se realiza mediante el chequeo de redundancia longitudinal (LRC).

- **RTU**

En modo RTU el campo de verificación de errores contiene dos bytes, resultado del cálculo del chequeo de redundancia cíclica (CRC)

Implementación De Modbus en Microcontroladores

En general, hay varias maneras de implementar el protocolo Modbus en un microcontrolador. La decisión acerca cual método de implementación será llevado a cabo, dependerá esencialmente de las especificaciones requeridas del prototipo. Para los cual existen diferentes metodologías:

Implementación compacta: El protocolo de comunicación y las tareas de aplicación se implementan en el mismo microcontrolador. En este caso los programas de aplicación pueden ser interrumpidos por las funciones de comunicación.

Implementación con procesador de comunicación separado: En esta variante se usa un microcontrolador separado para la ejecución del protocolo Modbus. Esta implementación es la más adecuada para dispositivos complejos. En este caso es posible cumplir con todas las funciones del protocolo sin restricciones en el programa de aplicación, debido a la separación de las funciones de comunicación y la tarea de aplicación.

Para la implementación del procesador de comunicaciones Modbus se utilizó un microcontrolador de 16 bits de la familia *HCS12* de Motorola. (Fig. 4) y en el mismo, aprovechando la capacidad de memoria, se programan las tareas de aplicación consistentes en el control de presión de un proceso; obteniendo finalmente un controlador compacto con protocolo de comunicación Modbus, el cual puede ser supervisado y controlado por un sistema SCADA.

Las principales características del microcontrolador HCS12 y su utilización en el desarrollo del procesador de comunicaciones son:

Pines de Entrada/salida de propósito general (*GPIO*)
Estos pines se utilizan para leer estados lógicos en un pin de entrada del HCS12 o para escribir niveles lógicos en pines de salida. Cada pin tiene asociado un bit en el registro de dirección de datos (DDR), el cual es utilizado para configurar el pin GPIO como entrada o como salida.

Timer-Counter: El HCS12 posee tres temporizadores-contadores de cuatro canales cada uno. Uno de éstos *timers* se emplea para generar los intervalos de tiempo 1.5 T.C. y 3.5 T.C. necesarios en la implementación del modo RTU, así como el *time out* en el modo ASCII.

Interfaz serial (SCI): El HCS12 posee tres interfaces de comunicación serial asíncrona (SCI). Una de las cuales se utiliza para la comunicación serial multipunto a través de RS-485 y la otra para comunicación serial punto a punto a través de RS-232.

Convertidor Análogo-Digital (ADC): El convertidor ADC de 10 bits que posee el HCS12 se utilizó para captar la señal analógica proveniente del sensor de presión.

En la figura 4 se muestra un diagrama de bloques, en donde se presentan las principales funciones del microcontrolador HCS12.

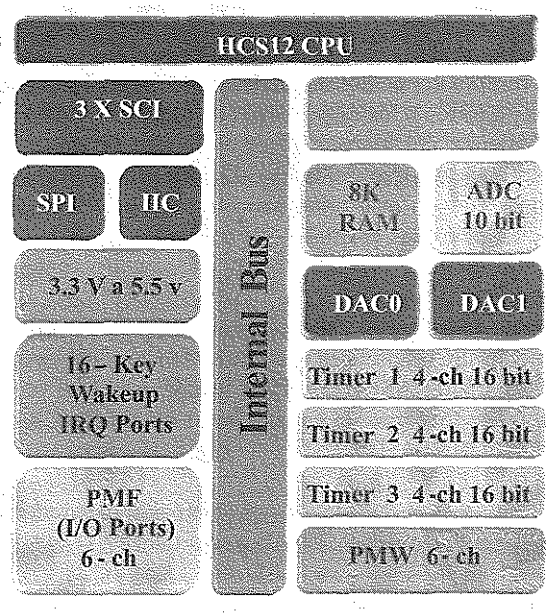

Figura 4. Diagrama de bloques del HCS12

Descripción Del Sistema Implementado

La Fig. 5 muestra el diagrama de bloques del sistema utilizado para verificar el correcto funcionamiento del procesador Modbus implementado; obsérvese que está

conectado a una red en la cual se encuentra el maestro, dos PLC´s y el controlador de presión.

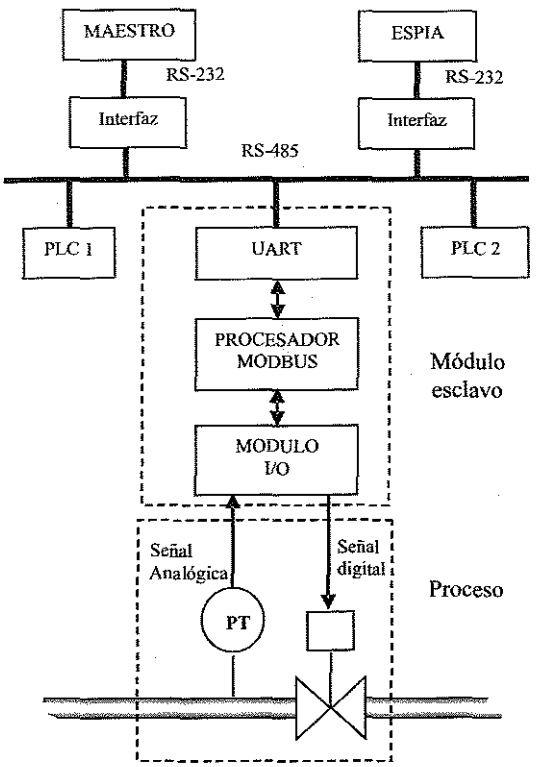

Figura 5. Sistema implementado

A continuación se presenta una breve descripción de cada parte del sistema.

MAESTRO MODBUS

Las funciones que realiza el maestro se implementaron en un programa en Labview, ejecutándose en un PC, el cual puede comunicarse con los demás dispositivos en los dos modos de transmisión ASCII o RTU. [1] mediante una interfaz RS-485

El protocolo de Modbus establece el formato para la consulta del maestro enviando hacia el dispositivo la dirección, el código de función, información adicional, y un campo de comprobación de error. El mensaje respuesta del esclavo se construye utilizando el formato de protocolo Modbus.

SUPERVISOR DE RED O (ESPIA).

Esta es una aplicación software realizada en Labview utilizada para capturar todas las tramas que viajan a través del bus.

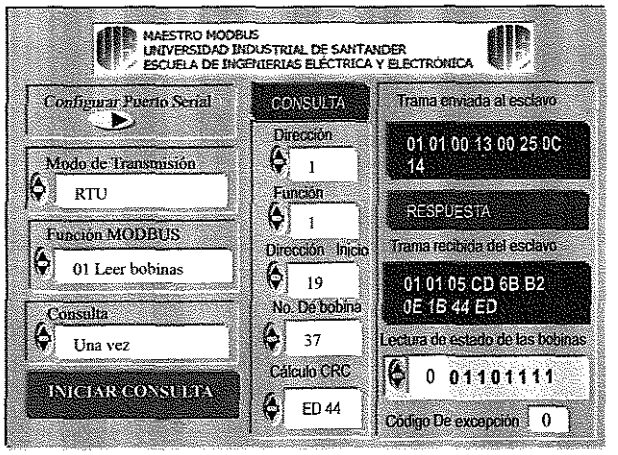

Figura 6. Maestro Modbus

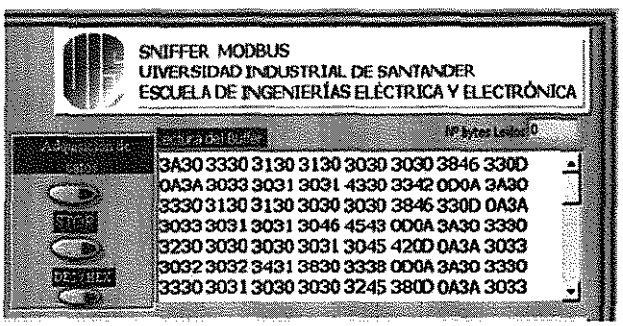

Figura 7. Supervisor (espía) de red

El espía esta instalado en un PC que lee las tramas vía puerto serial. Este software se usa para depurar la aplicación y verificar las tramas enviadas por el procesador Modbus.

CONVERSOR RS-232 A RS-485

En el sistema implementado el computador actúa como maestro ó como espía de la red Modbus. Para ésto se deben acoplar las señales eléctricas que envía y recibe el computador por su puerto serial (COM1) que utiliza el estándar RS-232 con las señales eléctricas de la red Modbus, la cual utiliza para su comunicación el estándar RS-485.

Para la implementación del conversor se utilizaron los siguientes circuitos integrados: MAX232 de *Texas Instruments*, ADM485 y ADP3367 de *Analog Devices* [8]. En la Figura 8 se muestra un esquema del conversor, sin tener en cuenta la alimentación de los circuitos.

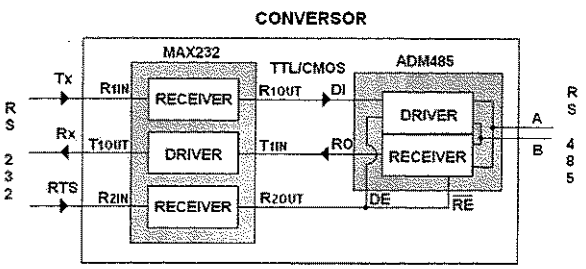

Figura 8. Conversor RS-232 a RS-485

MÓDULO I/O

El módulo I/O provee el acondicionamiento de las señales que salen y entran al microcontrolador. Entre sus funciones se cuenta: amplificación de la señal proveniente del sensor de presión y el control de la señal que va hacia la válvula solenoide.

SENSOR DE PRESIÓN

Se utilizó un sensor MPX2202, el cual tiene una salida lineal que cambia a razón de mV/kPa y opera desde 0 hasta 200KPa. Como la salida del sensor es analógica, se utiliza el ADC de 10 bits disponible en el microcontrolador HCS12.

PROCESADOR MODBUS

En la Fig. 9 se muestra el diagrama de bloques del procesador Modbus implementado en el microcontrolador. Este consta de módulo de funciones, módulo RTU, Módulo ASCII, Módulo de aplicación, módulo de configuración y los periféricos necesarios para comunicación (SCI), temporización (Timer), manejo de puertos (GPIO) y conversión analógica-digital (ADC).

MÓDULO RTU

Este módulo es el encargado de procesar el mensaje Modbus RTU proveniente de la interfaz SCI del microcontrolador .También realiza el chequeo CRC y genera las acciones adecuadas en caso de error.

MÓDULO ASCII

Este módulo es el encargado de procesar la trama ASCII proveniente de la interfaz SCI del microcontrolador, realiza el chequeo LRC y genera las acciones adecuadas en caso de error.

MÓDULO DE FUNCIONES MODBUS

En este módulo se programaron todas las funciones correspondientes al protocolo Modbus. Esta funciones incluyen: lectura y escritura de bits y registros de 16 bits, así como funciones de diagnóstico.

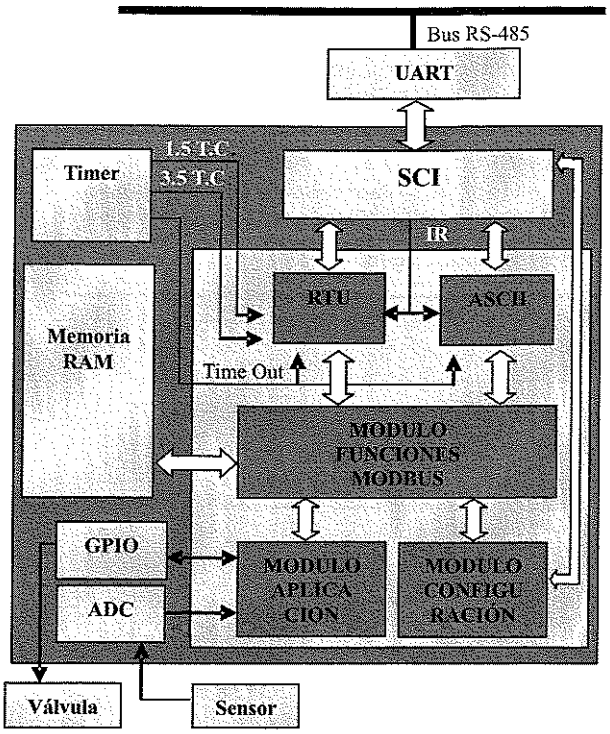

Figura 9. Diagrama de bloques del procesador

MÓDULO DE APLICACIÓN

El módulo de aplicación es el encargado de realizar la interfaz entre las variables físicas y los datos que son enviados o recibidos en formato Modbus a través de la red. En el trabajo se utilizó la función Modbus de lectura de registros para adquirir los datos de presión provenientes del sensor y la función Modbus de escritura de bits para activar la válvula solenoide.

MÓDULO DE CONFIGURACIÓN

Permite la configuración de los siguientes parámetros del procesador Modbus: Modo de transmisión, velocidad de transmisión, chequeo de paridad, time out y dirección del esclavo.

CONCLUSIONES

El protocolo Modbus es un formato de transmisión en serie de datos utilizado extensamente en las comunicaciones con PLC's pero fácilmente adaptable a otros tipos de instrumentación gracias a su particular estructura de mensaje (no opera con variables concretas sino con direcciones de memoria), lo cual permite que un instrumento se conecte en sistemas ya existentes sin necesidad de crear programas de comunicaciones específicos.

La arquitectura abierta del protocolo Modbus permitió su implementación en diferentes plataformas de software y hardware; el esclavo (procesador Modbus) se desarrolló en un microcontrolador de 16 bit y el maestro en LabView; con el único propósito de facilitar la comprensión del protocolo y disponer de una herramienta didáctica para la enseñanza de uno de los protocolos de mayor utilización en la industria Colombiana.

Las principales características del procesador de comunicaciones (esclavo Modbus) desarrollado son:

- Capacidad de manejo de dos puertos de comunicación: RS232 y RS485
- Configuración de modos de transmisión ASCII y RTU.
- Implementación de 15 funciones del protocolo Modbus para el acceso y manipulación de datos, tanto a nivel de bits como a nivel de registros de 16 bits y funciones de diagnóstico y configuración
- Capacidad para manejar hasta 2024 bits y 250 registros de 16 bits
- Posibilidad de configuración de los parámetros de comunicación (paridad, modo y velocidad de transmisión)

Para la verificación de la correcta ejecución de las funciones solicitadas por el maestro Modbus se emplearon las siguientes herramientas de software:

- Comdebug. Esta herramienta es un programa que permite la comunicación serial con diferentes dispositivos. Su principal característica es que calcula el CRC en forma automática, lo adiciona a la trama a enviar y lo muestra en pantalla.

- Maestro Modbus. Programa desarrollado en LabView, el cual permitió verificar que el esclavo, desarrollado en el microcontrolador, respondía a las peticiones (funciones) programadas en el maestro (ver figura 6).

- Sniffer Modbus. Este programa (ver figura 7) facilitó verificar la integridad de los datos enviados y recibidos por la aplicación desarrollada (esclavo Modbus. Ver figura 5) para poder validar el funcionamiento del procesador de comunicaciones en una red industrial, interactuando con PLC's comerciales (Koyo, Trilogic y Telemecanique)

Reconocimientos

Los autores reconocen el aporte de los Ingenieros Pedro Ardila y Sinle M. Carreño quienes desarrollaron los programas en LabView para el monitoreo de la red y la implementación del maestro Modbus [1].

Bibliografía

[1] ARDILA, Pedro, CARREÑO, Sinle M. Modbus. Monitoreo de la red empleando LabView. Tesis de grado. Universidad Industrial de Santander. 2005.

[2] CUNHA, J. M. Protótipo de rede industrial utilizando o padrão serial RS485 e protocolo Modbus. I Congresso Brasileiro de Computação – CBComp 2001.

[3] MODICON, Inc. Modbus Protocol Reference Guide PI–MBUS–300 Rev. J. Massachusetts. 1996.

[4] MODBUS.ORG, MODBUS over Serial Line Specification & Implementation guide V1.0 Enero 2004

[5] MODBUS.ORG, MODBUS application protocol specification V1.1. Agosto de 2004

[6] RAMÍREZ L., ACEVEDO C., "Wireless System for Electrical Networks Testing Based on MODBUS Protocol". Proceedings of the 14th International Conference on Electronics, Communications and Computers (CONIELECOMP'04) 2004

[7] RUBIO A., FUENTES, KAHORAHO E., PEREZ A., Performance evaluation of four field buses. Emerging Technologies and Factory Automation, 1999. Proceedings. ETFA '99. 1999 7th IEEE Int.

[8] TEXAS INSTRUMENTS. Interface Circuits for TIA/EIA485 (RS-485), Texas Instruments Design Notes slla036A, 2000.

[9] XIE J., DONG Q., Fieldbus network implementation based on RS-485. Intelligent Control and Automation, 2002. Proceedings of the 4th World Congress on Intelligent Control and Automation.