

ESTUDIO COMPARATIVO DEL DESEMPEÑO DEL PROTOCOLO IP Y EL PROFIBUS DE TIPO PA, CON BASE EN EL DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE COMUNICACIÓN ENTRE ELLOS MEDIANTE UNA APLICACIÓN GENERAL USANDO SISTEMAS PROGRAMABLES DE DESARROLLO

COMPARATIVE PERFORMANCE ASSESSMENT OF IP AND TYPE-PA PROFIBUS BASED ON A COMMUNICATIONS SYSTEM (DESIGN AND IMPLEMENTATION) THROUGH A GENERAL APPLICATIONS THAT USES DESIGN PROGRAMABLE SYSTEMS

ABSTRACT

This paper presents the development of an application to interconnect protocols, namely protocols Profibus and IPv4. This work attempts to implement a type-PA field bus without intrinsic security that can be monitored from the Internet through a web service called ThingSpeak, which serves as the basis of this study regarding the aforementioned protocols. Limitations of industrial systems are considered as well as their trends, which suggest these two protocols are likely to merge with Ethernet, creating a new generation of field-bus standards.

Keywords: Arduino, Ethernet, IPv4, Profibus

RESUMEN

El artículo presenta el desarrollo de una aplicación de interconexión entre los protocolos Profibus e IPv4, pretendiendo implementar un bus de campo tipo PA sin seguridad intrínseca y que sea monitoreado desde Internet mediante un servicio web llamado ThingSpeak, el cual sirve como base para la elaboración del estudio concerniente a los protocolos enunciados al inicio, apreciando las limitaciones de los sistemas industriales y sus tendencias hacia el futuro, donde llegarán a fusionarse con el protocolo Ethernet, dando origen a nueva generación de la estandarización de buses de campo.

Palabras clave: Arduino, Ethernet, IPv4, Profibus.

Cristian Camilo Caballero

Ing. electrónico egresado de la Universidad Distrital "Francisco José de Caldas". Actualmente se desempeña como Ingeniero de soporte.
k4ballero@gmail.com

Rodolfo Romero

Ing. electrónico egresado de la Universidad Distrital "Francisco José de Caldas". Actualmente se desempeña como Ingeniero de soporte.
rrromero71@gmail.com

Danilo Alfonso López Sarmiento

Ing. electrónico, MSc. en Teleinformática, docente de planta de la Universidad Distrital "Francisco José de Caldas", Coordinador de la Maestría en Ciencias de la Información y las Comunicaciones de la misma Universidad.
dalopez@udistrital.edu.co

Tipo: Artículo de investigación

Fecha de Recepción: Agosto 25 de 2011
Fecha de Aceptación: Septiembre 28 de 2011

1. INTRODUCCIÓN

Al tratar los sistemas electrónicos industriales, se hace notoria la participación de los buses de campo y los desarrollos tecnológicos con respecto a ellos, por tanto siguiendo la actual tendencia de mejoramiento e integración de tecnologías, se ve inminente la necesidad de incluir la conexión a Internet de dichos elementos, principalmente para supervisión remota desde casi cualquier lugar del planeta, por tanto se busca realizar un prototipo que cumpla con las especificaciones del protocolo Profibus al implementar una planta de tipo PA sin seguridad intrínseca, que tenga conectividad a la red mediante TCP, usando elementos arduino, además de conversores entre SPI y RS-485.

El paper resume la situación actual de los buses de campo, los trabajos que se han adelantado en cuanto a la comunicación de estos últimos con la web, el sistema de desarrollo que nos ofrece Arduino, la metodología, el diseño, la integración y las pruebas realizadas para lograr la comunicación entre el protocolo Profibus PA y el estándar Ethernet shield.

2. MARCO TEÓRICO

Al tratar la electrónica a nivel industrial se llevan a cabo interacciones con algunas áreas de la física, como la neumática o la hidráulica, por tanto diferentes grupos de empresas e instituciones han dedicado tiempo y esfuerzo para la integración de algunos dispositivos y sensores a sus sistemas de comunicación, es así como se dio origen a diversos estándares y protocolos, para muchos tipos de aplicación, obteniendo modelos tales como CAN, BitBus, Modbus, WorldFIP, Hart y Profibus, que se usan actualmente alrededor del mundo (algunos son de libre utilización), permitiendo adelantar nuevos avances en casi cualquier lugar del planeta.

En cuanto a Profibus, este es un bus de campo estándar que acoge un amplio rango de aplicaciones en fabricación, procesamiento y automatización, rigiéndose por la norma EN 50-170. Su principal finalidad comunicar componentes de distintos fabricantes sin necesidad de ajustes

especiales en sus interfaces, por tanto puede ser usado para transmisión crítica en el tiempo a alta velocidad y para tareas de comunicación extensas y complejas [1].

Su versatilidad viene dada por estas tres versiones mutuamente compatibles:

- PA: Diseñada para automatización de procesos, permitiendo la conexión de sensores y actuadores a una línea de bus común, inclusive en áreas de alto peligro, adicionalmente posibilita la comunicación de datos y energía mediante el uso de 2 tecnologías, la primera para áreas con peligros de explosión estipulada en la norma IEC 1158-2 [23] y otra sin seguridad intrínseca que emplea RS-485.
- DP: Optimizado para alta velocidad, usa conexiones sencillas y de bajo costo, fue diseñada especialmente para la comunicación entre los sistemas de control de automatismos con entradas y salidas distribuidas.
- FMS: Esta ideada como solución general para tareas de comunicación a nivel de célula, soporta gran rango de aplicaciones, posibilitando su uso en tareas de comunicación compleja y extensa como las de Internet.

El diseño de Profibus define toda una red de comunicación industrial, desde el nivel de campo hasta el de planta. En la Fig. 1 se muestra la familia de este bus de campo aplicada en toda la pirámide CIM (Manufactura integrada por computador).

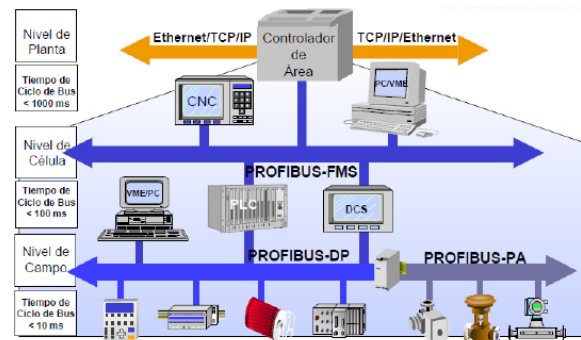


Fig. 1. Bus de campo Profibus aplicado en la pirámide CIM. [1]

Cabe anotar que Profibus PA se encuentra en el nivel de campo donde se tienen transductores, actuadores y sensores, los que requieren un determinismo absoluto con tiempos de ciclo de bus menores a 10 ms y con una operatividad superior al 99%.

La estructura de comunicación maneja tres formatos de datagramas, uno con longitud fija sin datos, otro con longitud fija con campo de datos de 11 bytes y uno con longitud variable con campo de datos que va desde 4 bytes hasta 240 bytes, como lo muestra la Fig. 2.

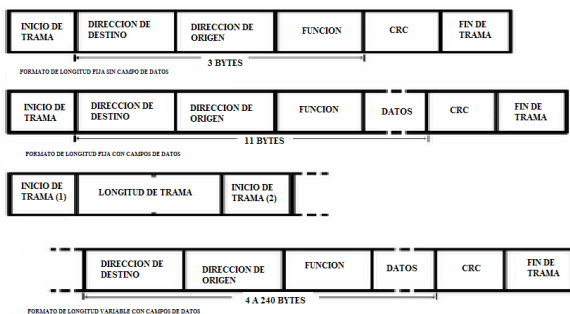


Fig. 2. Formatos de datagramas Profibus: con longitud del campo de información fija y sin datos (parte superior), con longitud del campo de información y datos fijo (parte intermedia), y con longitud del campo de información variable (parte inferior). [3]

En la actualidad Profibus es empleado por el grupo de los 20 países más industrializados del planeta, los cuales ofrecen un soporte para el resto del mundo [2], rigiéndose bajo la normatividad de Organización PROFIBUS International (PI), que cuenta con más de 750 miembros. Por otro lado al hablar de sistemas de desarrollo programable, se encuentra Arduino como una plataforma electrónica para el desarrollo de prototipos haciendo uso de un código abierto basado en hardware y software caracterizados tanto por su flexibilidad así como su facilidad de uso. En cuanto a su estructura se puede afirmar que está basado en una placa con entradas y salidas de tipo analógico-digital, que son programadas en un entorno de desarrollo que se implementa a partir del lenguaje procesing [4].

Dependiendo del dispositivo, se utiliza algún miembro de la familia de procesadores ATmega

[24] del fabricante Atmel, que son chips sencillos y económicos, permitiendo el desarrollo de múltiples diseños.

Un Arduino puede percibir el entorno mediante una variedad de sensores y de igual forma lo modifica controlando actuadores como luces, motores, controles, entre otros. Adicionalmente los proyectos de la plataforma pueden ser autónomos o comunicarse con algún software que se ejecute en un ordenador.

Entre las ventajas presentadas por Arduino se destaca la facilidad de adaptación del código de programación, siempre y cuando el dispositivo en el cual se va a implementar tenga las características necesarias. Actualmente es común encontrar en el comercio gran variedad de placas como la Duemilanove, Uno, Pro 328, Nano, Bluetooth, LilyPad, Mini, Mega 2560, entre otras [5].

3. ESTADO DEL ARTE

Los avances para conectar diversos buses de campo a Internet se están dando lentamente; en el caso particular de Profibus se tienen desarrollos como los que se presentan a continuación:

Evaluation of Delays Induced by Profibus PA Networks [6]: Habla sobre las soluciones para la gestión de objetos activos y su importancia en los sistemas modernos de control, siendo especialmente necesarias en el área de automatización de procesos. Los sistemas utilizados aquí por lo general consisten en redes basadas en PLC, que están optimizadas sobre los requisitos especiales temporales y topológicas del proceso. Funciones de gestión de activos han de introducirse a los sistemas, sin ningún tipo de influencia en el equipo existente. El documento muestra una solución orientada a objetos basada un dispositivo Profibus-PA. Evaluando de forma automática una estructura de un sistema y mapeando los parámetros relevantes para los perfiles de los dispositivos proxy, diferentes módulos de aplicación para las funciones de gestión de activos se pueden implementar actuando sobre estos objetos.

Integration of TCP/IP and PROFIBUS Protocols [7]: Presenta la arquitectura de una solución propuesta para incorporar un soporte para aplicaciones basadas en el protocolo TCP/IP en redes Profibus. La solución provista permite garantizar los requisitos de tiempo del control de tráfico y soportar a QoS (quality of service) en las aplicaciones multimedia, siendo implementada dentro de la interfaz de Windows NT NDIS.

Dentro de las consideraciones de este artículo indican que la integración entre Profibus y TCP/IP debe ser correctamente especificada, para proveer no sólo el adecuado QoS para las aplicaciones que soporta TCP/IP sino también garantizar los requisitos de tiempo del control de tráfico de Profibus. Además deben considerarse los siguientes detalles: el ser una solución que adapte el paradigma de maestro/esclavo del bus de campo para la naturaleza simétrica de las redes IP, así como la integración debe ser transparente desde el punto de vista de la aplicación.

Real-time property analysis of single master DP/PA hybrid network control system [8]: Se trata un sistema Profibus PA que es construida a partir de una red Profibus DP, diferenciando su medio de envío, la velocidad de transmisión, el formato de trama de datos. Se tomó la construcción típica de una red con un sistema de control con un maestro híbrido DP/PA como objeto de investigación, para hacer una propuesta sobre un método de mejoramiento del control de rendimiento en la red y crear condiciones favorables para su posterior análisis.

4. RESULTADOS

4.1. Resultados de implementación

Teniendo en cuenta que el objetivo principal de la implementación física es generar un sistema que permita realizar la conectividad de un dispositivo tipo PA sin seguridad intrínseca a Internet, se obtiene como resultado el esquema que se aprecia en la Fig.3, donde se evidencian cinco bloques relevantes, los cuales son la planta, el maestro, el bus, el esclavo y la placa Ether-

net (que se conectará al router para acceder a la red mundial).

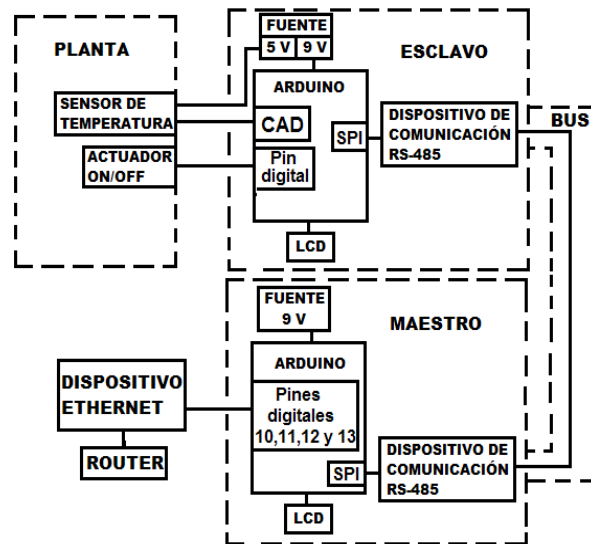


Fig. 3. Diagrama de bloques del prototipo implementado.

A partir de este punto se presentan una serie de pruebas realizadas con fin de construir y validar los avances para obtener el prototipo Profibus, por lo tanto es recomendable tener presente la representación previa para comprender más fácilmente como se llevó la secuencia de inclusión de los bloques allí mostrados.

Vale hacer especial énfasis en mencionar que al usar los dispositivos LCD, estos son empleados como mecanismos de visualización local de la variable (medida y/o transmitida) en cada uno de las terminales del bus. Por parte del conversor analógico digital (a partir de ahora CAD), él se encarga de capturar el valor de la variable evaluada, mientras que un par de pines (0 y 1) permiten la comunicación SPI con otros elementos (como los MAX485), en este caso haciendo posible la configuración de maestro y esclavo (donde se utilizan un par de Arduinos UNO), como en el presente desarrollo. Por último la tarjeta Arduino Ethernet Shield (que se inserta directamente sobre el maestro) permite la conectividad a la web mediante un cable de red que la comunica directamente a un router. Por tanto una vez dadas estas nociones se tiene los siguientes resultados parciales:

4.1.1. Visualización serial

Al emplear el Arduino UNO tras ser programado, se utiliza la visualización serial producida por el dispositivo y que mediante el cable tipo USB (que lo energiza y que posteriormente es reemplazado por fuentes de voltaje) realiza una comunicación con el ordenador mediante este puerto, pudiendo tener un monitoreo del comportamiento de la placa (Fig. 4), aprovechando una herramienta gratuita entregada por el fabricante del software (el monitor serial), que trabaja velocidades de transmisión como 9600 y 19200 bps, reconocidas por el computador.

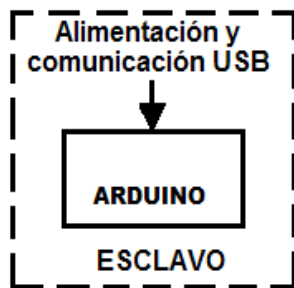


Fig. 4. Esquema del sistema para la visualización serial.

Para la programación de la tarjeta Arduino es indispensable descargar el software del mismo nombre (que en este caso es la versión 0022), proceder a conectar la placa e instalar el controlador del dispositivo para el sistema operativo empleado (siguiendo las instrucciones del fabricante), luego abrir el programa y siguiendo en la zona de configuración de código, inicializar el puerto serial con la instrucción "Serial.begin(9600)" donde ese número representa la tasa de transmisión deseada, la que en este caso es reconocida por el ordenador. Más adelante se establece la parte cíclica, donde se incluye "Serial.println(40)" y "delay(500)", la primera se encargará de imprimir ese valor en el puerto serial y dar un salto de línea, mientras que la segunda genera una pausa de quinientos milisegundos, para luego reiniciar el bucle, obteniendo un resultado como el de la Fig. 5. Finalmente se programa el hardware y se habilita la herramienta de visualización, pudiendo sintetizar el proceso en el diagrama de flujo mostrado a continuación.

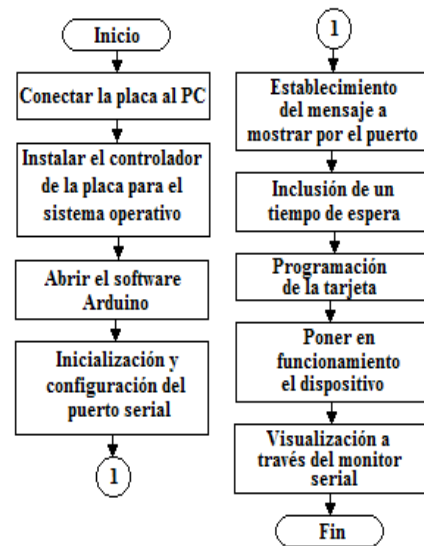


Fig. 5. Diagrama de flujo de la visualización serial con la placa UNO.

4.1.2. Lectura del CAD y visualización con LCD

Se capta el valor de un potenciómetro (que emula temporalmente al sensor por su alto consumo de corriente), mostrándolo en una pantalla matricial LCD [20] de 16x2 (Fig. 6) para supervisarlos sin necesidad de un ordenador, lo que es fundamental en las pruebas futuras, debido a que el protocolo Profibus tiene una tasa de transmisión de 31.25 kbps y no es posible que el monitor serial funcione en dicha velocidad.

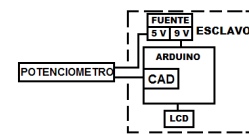


Fig. 6. Diagrama del dispositivo empleando el CAD y la visualización con LCD.

4.1.3. Comunicación serial SPI maestro-esclavo con visualización LCD

Utilizando la comunicación serial para conformar un sistema maestro-esclavo, el segundo captura el dato del potenciómetro (mediante su respectivo CAD) enviándolo a la otra placa, comprobando su estado con visualizaciones con LCD en cada terminal. Es importante aclarar que los pines 0 y 1 del Arduino UNO corresponden a la recepción y transmisión serial

respectivamente (como se muestra en la Fig. 7, con imágenes basadas en un simulador llamado VirtualBreadBoard o VBB [9]). Aclarando que para la comunicación del maestro al esclavo se representa la conexión con la línea de color azul, mientras que para la operación contraria se da con la roja. En la Fig. 8 se muestra el diagrama a bloques del montaje mencionado.

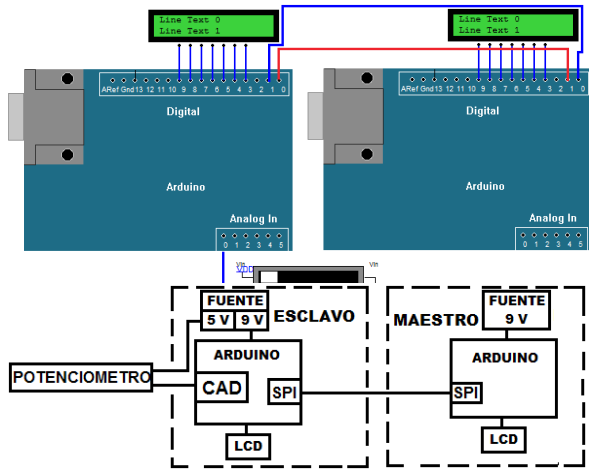


Fig. 8. Diagrama de bloques del sistema maestro esclavo comunicados por SPI.

También se debe anotar que los desarrollos parciales manejaron una secuencia donde se transmitió un dato, luego modificando el código se pasa a manejar una cadena de estos y finalmente tras adecuaciones posteriores de tiempos, se da una comunicación bidireccional, para finalmente modificar la velocidad de especificada por el protocolo Profibus.

4.1.4. Comunicación serial RS-485 maestro-esclavo con visualización LCD

Debido a que los dispositivos Arduino manejan comunicación SPI, se emplea una conversión desde ésta a RS-485 y viceversa, cumpliendo el protocolo Profibus PA sin seguridad intrínseca en sus especificaciones de la capa física (ver [10] y [11]) donde, según las especificaciones de la norma TIA/EIA-485, se debe usar un par trenzado-apantallado de entre 20 y 24 AWG dependiendo de las especificaciones de corriente e impedancia que presente la línea (típicamente de 100 a 120 ohmios), teniendo que colocar una resistencia de terminación al final del bus

para evitar las reflexiones de señal. El integrado de la Fig. 9, MAX485 de MAXIM (ver [12]), es el encargado de hacer la conversión bidireccional previamente enunciada, alternando su sentido de operación por una señal digital en sus pines 2 y 3, para un tipo encapsulado DIP/SO.

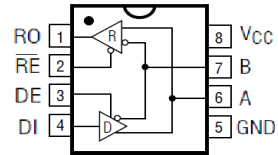


Fig. 9. Configuración interna del MAX485 [12].

De manera similar a los desarrollos con SPI se llega a la transmisión de una cadena de datos de manera bidireccional, todos manejando la visualización LCD y variando las tasas de transmisión a la velocidad establecida por Profibus de 31.25 kbps, obteniendo un esquema de la conectividad física de los diferentes módulos utilizados (fig. 10).



Fig. 10. Diagrama de bloques del sistema maestro esclavo utilizando RS-485.

4.1.5. Lectura CAD, conexión y visualización Ethernet

Mediante la placa Arduino Ethernet Shield SD [25], se da una expansión de la funcionalidad de la placa base donde se instala, permitiendo una conexión a la red, al emplear los pines 10 al 13 para comunicarse entre ellas.

En el esquema de la Fig. 11 se muestra un potenciómetro al que se le captura su valor de voltaje, generando un dato, el cual es enviado posteriormente por la placa Ethernet a la red.

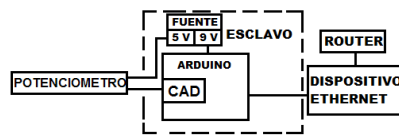


Fig. 11. Diagrama de bloques del sistema para visualización con la placa Ethernet.

Para esto se programa la tarjeta UNO de manera independiente con un procedimiento similar al descrito en el diagrama de la figura 5, sin embargo al conformar el código fuente, se debe realizar la inicialización adicional del módulo Ethernet mediante la instrucción "Ethernet.begin(mac, ip, gateway, subnet)", cuyos valores varían según la red donde se haga la conexión, luego dependiendo del tipo de aplicación se puede emplear como servidor o cliente, esta última es la que se emplea para los requerimientos del proyecto. Una vez programada la tarjeta, se le inserta el módulo de red y se hace la conexión tanto a la alimentación de energía y el router, para luego visualizar la información deseada a través de un navegador web. El anterior procedimiento descrito se plasma en la Fig. 12.

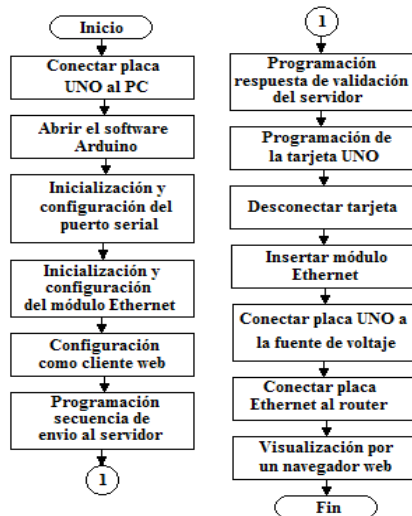


Fig. 12. Diagrama de flujo de la visualización de la placa Ethernet.

4.1.6. Lectura CAD, conexión y visualización ThingSpeak y LCD

Al integrar el módulo Ethernet Shield SD se tiene la conexión a Internet para realizar la visualización a través de una página web, incluyendo con esto la última parte del desarrollo al utilizar el servicio ThingSpeak [13], el cual recibe la información enviada por la placa, almacenándola en una cuenta y retornando una gráfica histórica al usuario (como en la figura 22). Cabe resaltar que la conexión se da con la página en una

configuración de cliente web, direccionando los datos hacia la cuenta a través de una clave única llamada API KEY, y mediante el comando "client.print("X-THINGSPEAKAPIKEY: "Insertar aquí la API KEY"\n")" que hace la identificación para acceder a modificar la base de datos. Además se emplea un hardware similar al de la Fig. 11, diferenciándose únicamente por incluir una LCD, que se emplea en la verificación de la información local del Arduino.

4.1.7. Comunicación serial SPI maestro-esclavo con visualización ThingSpeak y LCD

Se hace la fusión de la tarjeta Ethernet al sistema maestro-esclavo, realizando una comunicación por SPI, similar a los desarrollos previos, y al adaptar el código para la utilización de dicha placa, se coordinan los tiempos manejados en ambas aplicaciones, obteniendo una implementación física como la presentada en la Fig. 13. Cabe aclarar que su esquema por bloques es muy similar a la Fig. 7, sin embargo se añade la conexión al dispositivo Ethernet y esto se lleva a cabo sobre el maestro.

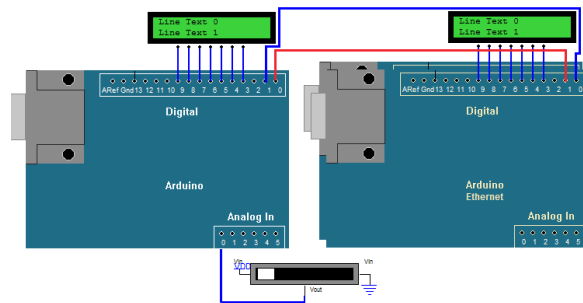


Fig. 13. Dispositivo Arduino con placa Ethernet en el maestro y comunicación SPI.

4.1.8. Comunicación serial RS-485 maestro esclavo con visualización ThingSpeak y LCD

Se emplean los dispositivos para la comunicación RS-485 al hardware implementado previamente (Fig. 14), haciendo los respectivos ajustes de código para operar los MAX485, y conservar la comunicación bidireccional, llegando a un modelo como el expuesto en la Fig. 3, sin embargo este no cuenta con el actuador, ni el sensor.

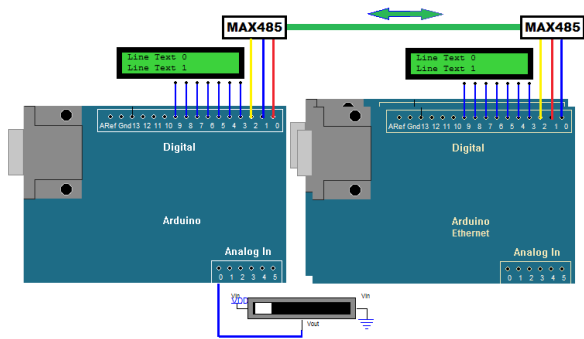


Fig. 14. Dispositivo Arduino con placa Ethernet en el maestro y comunicación RS-485.

4.1.9. Comunicación serial RS-485 maestro esclavo con visualización ThingSpeak y LCD utilizando la planta térmica

Se realiza la construcción de una planta térmica (Fig. 15) y se usó el LM35 [18], que posee un comportamiento lineal en la escala que se desea manejar (de 10 a 60 °C); además se incluyó un actuador (bombillo de tungsteno), encargado de elevar la temperatura interna del horno.



Fig. 15. Vista superior de la planta.

4.1.10. Prototipo final

El hardware emplea un par de placas Arduino UNO, para el procesamiento de la información y la comunicación SPI (obteniendo sistemas de tipo maestro-esclavo), además de igual cantidad de MAX485, que permite hacer la conversión entre ese estándar y el RS-485 (Fig. 16). Adicionalmente se hace uso de otro módulo de la familia Arduino, conocido como "Ethernet Shield SD", que se ensambla directamente sobre la placa UNO manualmente al manejar la misma distribución de pines. Por otro lado se

tiene un control ON/OFF para manejar un actuador (bombillo) encargado de mantener la temperatura de referencia (Fig. 17 y 18).

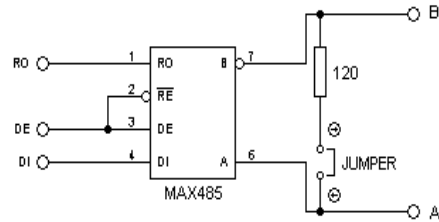


Fig. 16. Convertor usando el MAX485.

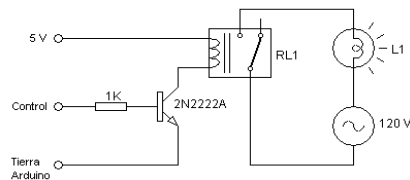


Fig. 17. Control del actuador.

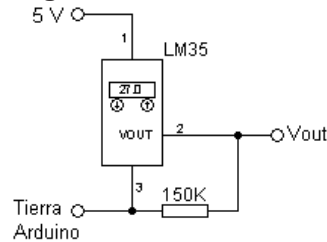


Fig. 18. Circuito de conexión del sensor.

Por último para obtener el prototipo definitivo (Fig. 19 y 20) acorde a la Fig. 3, se hace uso de un servicio web (ThingSpeak) para supervisar los datos del sensor en intervalos de tiempo, generando una gráfica, donde se evidencian las perturbaciones inducidas en la planta, y apreciadas de manera detallada como administrador de la cuenta. Sin embargo al activarse el acceso a visitantes, este solo aprecia una línea de tendencia sin mayores detalles del comportamiento del sensor (para este caso), así como se ve en la Fig. 21 y 22.

En la figura 20 se aprecia el desarrollo final (para diferenciar los componentes se incluye una nomenclatura numérica): 1) la planta; 2) sensor; 3) Arduino maestro; 4) Arduino esclavo; 5) LCD maestro; 6) LCD esclavo; 7) Convertor RS-485 maestro; 8) Convertor RS-485 esclavo; 9) Circuito actuador; 10) Par trenzado

del bus.

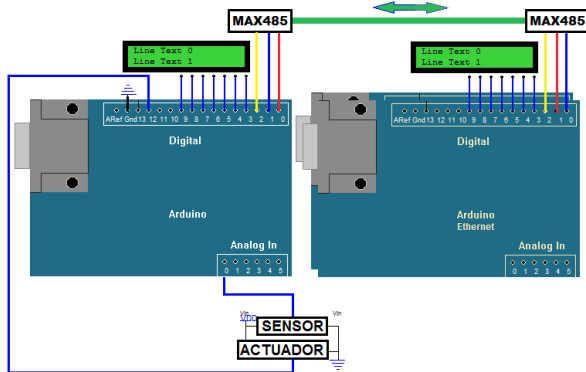


Fig. 19. Esquema del prototipo final.

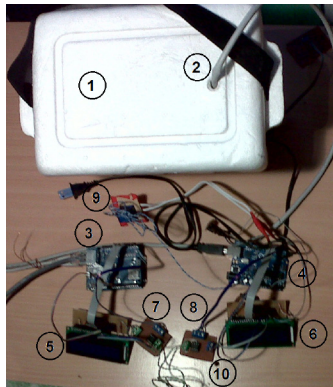


Fig. 20. Vista superior del hardware y la planta.

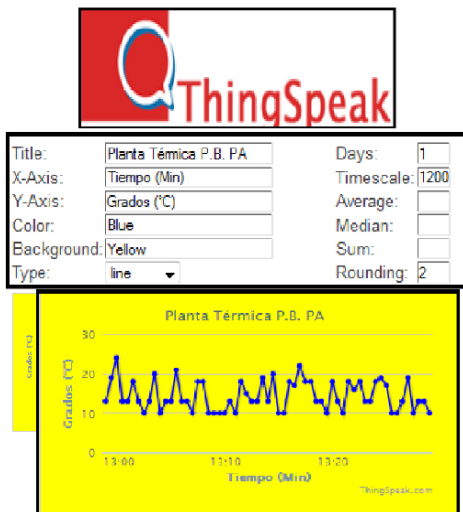


Fig. 21. Visualización de la cuenta de ThingSpeak como administrador.



Fig. 22. Visualización de la cuenta de ThingSpeak como visitante.

En la Fig. 23 se aprecia el diagrama de flujo donde se plasma el funcionamiento del código del esclavo de manera general, ya que siguiendo la estructura exigida por el software, este debe iniciar definiendo las bibliotecas a utilizar (en este caso la relacionada a la pantalla LCD), luego se concretan los pines de conexión para la pantalla matricial y posteriormente se establecen las variables a emplear. Ahora se entra en la zona de configuración en la cual se definen las características del puerto serial, así como los pines digitales (de entrada o salida) y por último la asignación de pines a la LCD. En esta instancia se entra a la parte cíclica del código, capturando el dato del CAD, para visualizarlo en la LCD y esperar la petición del maestro durante un determinado tiempo, donde se da respuesta o se reinicia el bucle.

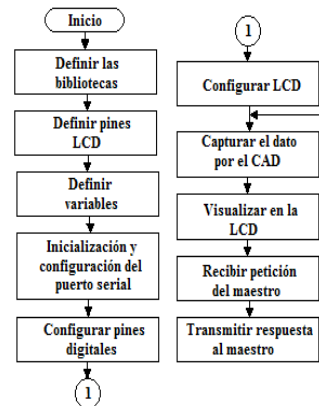


Fig. 23. Diagrama de flujo del código del esclavo. Para el maestro (Fig. 24) se tiene un diagrama de flujo similar al del esclavo, aunque se incluye el uso de la biblioteca Ethernet para emplear el módulo del mismo nombre. En la parte de configuración se adiciona la adecuación del sistema al punto de acceso a la red como se vio en una de las pruebas previas. Finalmente en la sección cíclica se inicia realizando la petición del dato al maestro, el cual una vez le da respuesta, se visualiza el valor de manera local mediante una LCD, para luego realizar una interacción como cliente web con el servidor de ThingSpeak, en donde al hacer los debidos pasos se establece una comunicación que permitirá observar la información a través de un navegador web, culminando el ciclo.

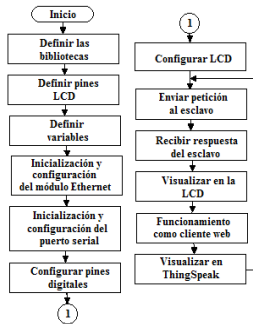


Fig. 24. Diagrama de flujo del código del maestro

4.2. Pruebas de validación

En la realización de las pruebas y el prototipo, se empleó la placa Arduino UNO cuyo microcontrolador es un ATmega328 [14] que permite una comunicación serial SPI, donde es necesario establecer una tasa de transferencia deseada (en baudios) [15], ubicándola inicialmente en 9600 bps (reconocida por el ordenador), para transmitir un byte constante de manera cíclica, obteniendo la Fig. 23, y luego aumentar la tasa a 31250 bps (establecida por la especificación Profibus) repitiendo el proceso, llegando a lo observado en la Fig. 24.



Fig. 25. Captura de un byte transmitido por el Arduino UNO a 9600 bps, con 5 V/ div vs. 200 µs/div.



Fig. 26. Captura de un byte transmitido por el Arduino UNO a 31250 bps, con 5 V/div vs. 500 µs/div.

Al comprobar que los Arduinos soportan las velocidades necesarias para los desarrollos, se procedió a manejar la visualización del monitor serial [19] a 9600 bps (Fig. 27), para supervisar el funcionamiento de los dispositivos en algunas pruebas, especialmente al conformar los sistemas maestro (puerto com10) esclavo (puerto com9), nombrados previamente.

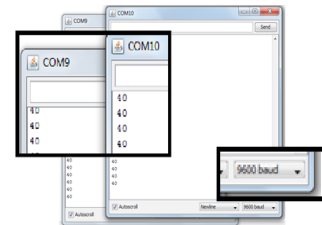


Fig. 27. Visualización de monitores seriales para una configuración de Arduinos UNO como maestro-esclavo.

Cuando se modifica la tasa de transferencia del puerto serial a 31250 bps, la velocidad no es reconocida por el computador, por lo que se hace indispensable tener otra manera de verificar la conexión adecuada del sistema, siendo este punto donde toman importancia las visualizaciones con LCD, pudiendo apreciar mediante ellas la transmisión adecuada del dato; un ejemplo es el mostrado en la Fig. 26 donde se tiene un potenciómetro cuyo valor es visualizado por una pantalla matricial de 16x2 por medio del Arduino UNO.



Fig. 28. Visualización con LCD de un dato capturado por el CAD.

En cuanto al dispositivo conversor entre SPI y RS-485 es posible observar mediante un osciloscopio las señales diferenciales transmitidas por el bus (Fig. 29), encontrando en la parte superior la señal A y en la inferior la B, que cumple la especificación para su transmisión por un par trenzado y apantallado, al emplear el MAX485 (Fig. 9).

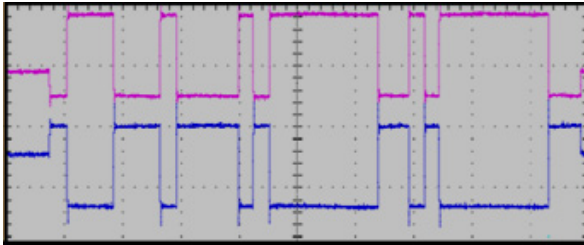


Fig. 29. Señales diferenciales transmitidas por el bus con el MAX485 a 9600 bps, con 2 V/div vs, 500 μ s/div.

Para validar el envío de los paquetes de datos a la salida del maestro se usó el software WIRESHARK, conectando el equipo con el analizador, a la salida de la tarjeta Ethernet Shield de arduino y realizando capturas de las trama enviadas por esta a la página de ThingSpeak durante 305 segundos, se obtiene la Fig. 30, donde se aprecia claramente el envío de la petición cada 30 segundos como está estipulado en el código fuente del maestro, ya que idealmente es el intervalo mínimo de comunicación que ofrece el servidor de manera gratuito. En el eje Y se grafica la cantidad de paquetes enviados por intervalo.

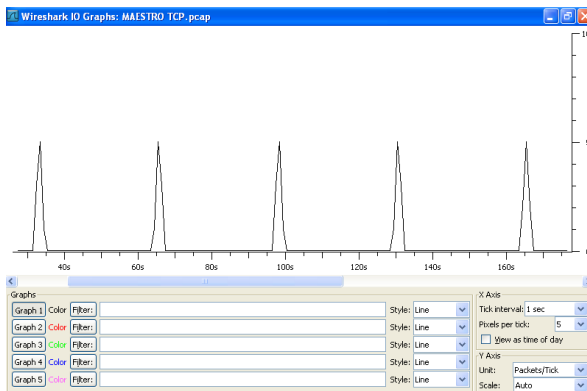


Fig. 30. Gráfica del envío de la trama contra el tiempo usando Wireshark.

En la tabla 1 se muestran estadísticas de la captura graficada en la Fig. 30 donde se observa que en los 305 segundos se capturaron 103 paquetes y en cada petición se enviaron 10 paquetes por lo que se puede inferir que se enviaron 10 peticiones, una cada 30 segundos.

Tabla 1. Estadísticas de captura en la salida de la Ethernet shield.

| Estadísticas de captura wireshark | |
|--|--------------|
| Filtro de captura | Ethernet |
| Paquetes capturados | 103 |
| Tiempo entre el primer y el último paquete | 305,651 seg |
| Promedio longitud de paquete | 73,981 bytes |
| Bytes | 7620 |
| Promedio de bytes por segundo | 24,93 |
| Paquetes por petición | 10 |

El envío de datos por parte del maestro y la tarjeta ethernet shield se realiza mediante peticiones HTTP 1.1 del tipo POST colocadas dentro de su programación, por esta razón el servidor puede tomar y procesar los datos que por su funcionalidad le sean permitidos. Al hacer un promedio de los datos recibidos por el servicio web y las provenientes de mantener la planta en funcionamiento durante 10 horas continuas, se obtiene la Fig. 31, donde se hace una estimación de 15.4 datos subidos por hora, es decir que alrededor de cada 3 minutos con 54 segundos es almacenado allí un nuevo valor, lo que puede llegar a reducirse, según el proveedor del servicio, a unos 5 segundos al pagar por una cuenta con mayores privilegios.



Fig. 31. Gráfica del registro histórico de la cantidad de datos almacenados por hora.

5. ESTUDIO COMPARATIVO ENTRE LA RED PROFIBUS PA E IPV4

Para llegar a este punto fue necesario implementar la conexión entre un sensor ubicado en el extremo de una red tipo Profibus con perfil PA, hasta una visualización web de los datos, tras ser almacenados en un servidor, donde se grafica cada valor contra el tiempo y produce una base de datos, que puede servir de gran manera para tomar decisiones sobre la variable censada. Cabe anotar que dicha captura se dio por medio de unas placas Arduino UNO en configuración maestro-esclavo interconectadas por un bus de tipo RS-485, obteniendo un montaje empleado como base para conocer a fondo las características de cada uno los elementos allí presentes, así como su trabajo en conjunto y posteriormente realizar una comparación como la que se presenta en la tabla 2.

Tabla 2. Comparación profibus PA e IPV4

| Información básica | Profibus tipo PA | IP |
|------------------------------------|--|--|
| Desarrollo de la tecnología | SIEMENS | Agencia de proyectos de investigación del pentágono ARPANET |
| Año de introducción | 1994-1995 | 1969 |
| Estándar | E N 5 0 1 7 0 / DIN1945 parte 3 y 4/IEC 1158-2 | IPV4 TCP/IP |
| Características | ASICs de Siemens y Profichip productos de más de 300 vendedores | Multitud de chips y vendedores |
| Características Físicas | | |
| Topología de red | Línea, estrella, anillo o árbol | Estrella, Daisy-Chain |
| Soporte físico | Par trenzado o fibra | Coaxial, par trenzado, fibra, inalámbrico |
| Max. número de nodos, dispositivos | 127 nodos (124 esclavos - 4 segmentos, 3 repetidores)+3 maestros | Según la topología de red, pero en teoría hasta 4.294.967.296 que son el número de direcciones |
| Max. distancia | 1900 m entre segmentos A 31,25 Mbps | Por su esquema no hay limite |
| Mecanismo de transporte | | |

| Método de comunicación | maestro/esclavo, peer to peer | Peer to peer |
|--|--|--|
| Propiedades de transporte | DP: 9,6, 19,2, 93.75, 187,5, 500 Kbps, 1,5,3, 6,12 Mbps PA:31.25 Kbps | Condicionada por el ancho de banda disponible |
| Tamaño del dato de Transferencia | 0-244 bytes | Máximo 65536 bytes, Pero varía según el tipo de red |
| Método de prioridad | Paso de testigo | CSMA/CD y otras según la red |
| Chequeo de error | CRC | CRC |
| Tiempo de envío de tramas (para este caso) | 1 trama cada 0.5 segundos (programados en los arduinos) | 1 trama cada 3 segundos (programadas en la tarjeta Ethernet petición HTTP) |
| Diagnóstico | Estación, módulo y diagnóstico de canal | |

Se puede decir que el desarrollo de estas tecnologías ha sido iniciativa de grandes empresas y/o países, enfocados en la investigación de la parte electrónica y de comunicaciones, quienes concibieron diversos estándares para diferentes aplicaciones. IP comenzó como un proyecto de investigación del Departamento de Defensa de los Estados Unidos hacia el año 1969, mientras que Profibus nació por iniciativa de Siemens para poder unificar los buses de campo que estaban en auge la última década del siglo XX. El perfil PA nació un poco después que DP y FMS, con el fin principal de llegar a los actuadores y sensores industriales, que se encontraran en los medios complejos y hostiles.

Las dos redes se encuentran totalmente normalizadas dentro sus respectivos estándares que son aplicados a nivel mundial, en consecuencia tienen gran variedad de dispositivos. Por el lado del bus de campo existen algunos como los Asic [21] (chips de Siemens) y Profichip [22], que facilitan la implementación de nodos esclavos con sus transductores o actuadores manejando una limitada capacidad de procesamiento, mientras que IP cuenta con una normalización extensa de protocolos, de los cuales el más representativo es el conocido como TCP/IP.

La topología de red en el modelo PA puede ser de línea, estrella, anillo o árbol, con conectivi-

dad física por medio de par trenzado o fibra óptica, además de un máximo de 127 nodos divididos en 4 segmentos, donde debe haber un repetidor y un maestro (como mínimo) por cada segmento, además debido al manejo de corriente se limita a 32 nodos dependientes de la misma fuente.

Por el lado del protocolo de Internet se usa casi todo tipo de topología de red como estrella o Daisy-Chain [30] (que consiste en una sucesión de enlaces). Su enlace físico se puede dar con cable coaxial, par trenzado y fibra óptica, manejando hasta 4.294.967.296 nodos, que en teoría es la cantidad máxima de direcciones disponibles en IPv4.

Es de resaltar que el direccionamiento IP puede ser de dos tipos: estático y dinámico, lo que depende del tipo asignación de la dirección al dispositivo, bien sea de manera fija o que se le asigne una nueva cada vez que este se conecte a la red, mientras que en la red Profibus se tiene un sistema fijo para cada nodo de la red que se programa y se comunica a todos los dispositivos para poder hacer un reconocimiento inmediato.

En cuanto a los posibles mecanismos de transporte el método de comunicación de PA es maestro-esclavo, aunque en el caso de haber más de un maestro se usa entre ellos peer to peer (que también es un modelo muy usado en la comunicación IP), a una velocidad de transmisión definida de 31.25 kbps.

Las dos redes tienen un sistema de corrección de errores (CRC) para determinar si la trama evaluada es correcta o se debe pedir el reenvío por corrupción de los datos. Adicionalmente en Profibus PA se puede tener una configuración específica para hacer un diagnóstico del canal, de la estación y el módulo, desde el maestro al inicializar la red.

El tamaño de los datos de transferencia en Profibus es de 0 a 244 bytes y se priorizan los mensajes con el paso de testigo entre los maestros, mientras que para Internet la longitud puede ser hasta de 65536 bytes.

En cuanto a velocidad la especificación PA tiene una tasa definida en el protocolo de 31,25 kbps (que es resultado de tomar un oscilador de cristal de 1 MHz, dividiendo la frecuencia por 32), la cual resulta de tomar como referencia la tecnología ASI [16], que era bastante usada en el momento de la consolidación del estándar, además uno de los principales objetivos de Profibus es tener un bus de bajo nivel para los instrumentos de campo, donde la capacidad de procesamiento no es muy alta y se requieren velocidades de transmisión bajas, para lo cual esa tasa de transferencia fue una buena solución.

Para el protocolo de Internet la velocidad depende del ancho de banda que se esté usando, por tanto es importante resaltar que dicha tasa de transferencia de la red varía dependiendo de factores como la hora del día o la cantidad de usuarios activos. Adicionalmente se debe considerar que al hacer una petición de una página web a algún servidor, tanto la solicitud como su respuesta viajan por la red atravesando diferentes enlaces hasta encontrar la ruta menos saturada, gastando algunas milésimas de segundo, haciendo que el resultado se altere según el tráfico que existió en ese momento.

6. CONCLUSIONES

Se ha diseñado y desarrollado un sistema Profibus PA que permite manejar una velocidad de transmisión de hasta 31.25 kbps (como lo sugiere el estándar).

En cuanto a la modularidad del sistema diseñado con Arduinos para este bus de campo, permite que los dispositivos configurados con el esquema desarrollado tengan una adaptación sencilla, cumpliendo uno de los objetivos principales de la generación de dicho estándar hace más de dos décadas. Desafortunadamente esta familia de las placas de desarrollo no traen integrada la transmisión de información según el estándar RS-485 (que es el estándar del medio físico Profibus), sin embargo este aspecto es solventable con relativa sencillez al emplear el circuito integrado MAX485, que se encarga de hacer conversión desde o hacia SPI, permiti-

tiendo manejar diversos esquemas de buses de campo.

Se debe anotar que en este sistema se puede implementar en diversas aplicaciones siempre y cuando el tiempo de respuesta en el monitoreo no sea crítico, ya que los tiempos de visualización en los servicios web probados están por el orden de minutos, algo que puede ser una seria limitante al momento de tomar decisiones en plantas o sistemas que necesiten rápidas

modificaciones.

Para culminar solo resta decir que a pesar de ser una aplicación académica, quedan las puertas abiertas para futuras innovaciones que pudiesen tomar el presente trabajo como una de sus bases, ya que las aplicaciones futuras serán muy numerosas, debido a que día a día surgen nuevas necesidades que dependen de los buses de campo para ser debidamente solucionadas.

Referencias Bibliográficas

- [1] PROFIBUS, información general coherente y orientada a la aplicación. [En línea], Consultado en febrero 2 de 2011, disponible en <http://www.profibus.com>.
- [2] PROFIBUS (PA/DP/FMS), Universidad olitécnica de Cartagena, Departamento de Tecnología Electrónica, asignatura de Comunicaciones industriales, 2010.
- [3] F. López, J. Profibus, Universidad de Vigo. Departamento de ingeniería de Sistemas y Automática, 2010
- [4] Arduino Company. [En línea], consultado en Junio 2 de 2011, disponible en <http://www.arduino.cc>.
- [5] Arduino Company en español. [En línea], consultado en Junio 2 de 2011. <http://www.arduino.cc/es/>
- [6] [Q. Li, J. Jiang, D. Rankin, Evaluation of delays induced by profibus PA networks, Department of Electr. & Comput. Eng., University of Western Ontario, London, Canada. 2009.
- [7] N. Pereira, Grupo de investigación del Instituto Politécnico de Porto, Portugal; Filipe Pacheco Paulo, Instituto de Sistemas Industriales, Universidad de Patras, Grecia; Luis Miguel Pinho, Instituto de Automatización y Comunicación, Magdeburg, Alemania. 2002.
- [8] C. Zheng-shi; S. Jun-zheng; S. Jianfang, Real-time property analysis of single master DP/PA hybrid network control system. Coll. of Comput. & Electron. Inf., Maoming Univ., Maoming, China, 2010.
- [9] C. James, Virtual breadboard. [En línea], consultado en Julio 12 de 2011, disponible en <http://www.virtualbreadboard.net/>.
- [10] A. Jan, Programming and circuits for SPI and RS-485. serial port complete, Lakeview Research, USA, 2003.
- [11] [En línea], consultado en Mayo 3 de 2011, disponible en: www.neoteo.com/rs485-domotica-al-alcance-de-tu-mano-15810
- [12] Low-Power, Slew-Rate-Limited RS-485/RS-422 transceivers. Maxim integrated products. USA, 2003
- [13] IoBridge. ThingSpeak company. [En línea], consultado Agosto 13 de 2011, disponible en: <https://www.thingSpeak.com/>
- [14] ATmega328P. ATmel corporation. [En línea], consultado en 24 de Marzo de 2011, disponible en: http://www.atmel.com/dyn/resources/prod_documents/doc8161.pdf.
- [15] Arduino Company en español. Comunicación vía puerto serie. [En línea], consultado en Agosto 10 de 2011, disponible en: http://arduino.cc/es_old/Ejercicios/PuertoSerie

- [16] Vicent Guerrero, Bus AS-I: Configuración y programación de una red AS-i. [En línea], consultado en Enero 21 de 2011, disponible en http://instrumentacionycontrol.net/Descargas/Descargas/SIEMENS/Comunicaciones/IyC-net_Siemens_01_ASi_S7_300.pdf
- [17] A. Quiroga, J. Ignacio. Clase número 79 del curso de automatización y control. [En línea], consultada en Mayo 18 de 2011, disponible en: <http://www.youtube.com/watch?v=DsPGXfBXfZY>
- [18] LM35 Precision centigrade temperature sensors. National semiconductor Corporation. 2000.
- [19] Arduino Company. Arduino Development Environment. [En línea], consultado Marzo 19 de 2011, disponible en: <http://www.arduino.cc/en/Guide/Environment>.
- [20] Datasheet Led. [En línea], consultado en Marzo 20 de 2011, disponible en: <http://es.scribd.com/doc/18946526/Datasheet-Lcd-16x2>
- [21] Procentec. SPC4 Protocol ASIC. [En línea], consultado en Julio 27 de 2011, disponible en: <http://www.procentec.com/products/spc4-profibus-asic.htm>
- [22] Profichip Gmb. PROFIBUS-DP slave Controller with serial Interface. [En línea], consultado en Abril 21 de 2011, disponible en: <http://www.profichip.com/products/overviewasics/>
- [23] IEC 1158-2, Tecnología de transmisión, [En línea], consultado en marzo 25 de 2011, disponible en: <http://www.control.aau.dk/~henrik/undervisning/netpro/033iec1158.html>.
- [24] ATMEL, Atmel AVR 8 - y 32 bits, [En línea], consultado marzo 21 de 2011, disponible en: http://www.atmel.com/dyn/products/devices.asp?family_id=607
- [25] Arduino, Ethernet Shield. [En línea], consultado marzo 3 de 2011, disponible en: <http://arduino.cc/es/Main/ArduinoEthernetShield>
- [26] D. Chain. [En línea], consultado en abril 20 de 2011, disponible en: http://es.wikipedia.org/wiki/Daisy_chain