

Pasarela para usar transmisores HART desde una red DeviceNet

Gateway to use HART transmitters from a DeviceNet network

Asfur Barandica López
Universidad del Valle, Escuela de
Ingeniería Eléctrica y Electrónica.
asfur.barandica@correounivalle.edu.co

Andrés Ernesto Guevara Escobar
Carvajal Tecnología y Servicios, Área
de Facturación Electrónica.
andresegs@hotmail.com



Resumen

Este artículo demuestra la viabilidad de compatibilizar dos protocolos industriales con características muy diferentes, mediante la implementación de una interfaz hardware-software. HART es un protocolo de muy baja velocidad que usa el paradigma maestro-esclavo, no orientado a la conexión y con un conjunto de comandos definidos, mientras que DeviceNet tiene una velocidad al menos 100 veces superior, que funciona con el modelo productor-consumidor, es orientado a la conexión y no utiliza comandos sino objetos. El desarrollo se soporta en un sistema embebido TINI (Tiny InterNet Interface), programable en Java. La pasarela (*gateway*) actúa como esclavo de una red DeviceNet y como maestro de una red HART. Este novedoso equipo puede ser empleado para incluir equipos HART en procesos de manufactura que requieran pocas variables continuas.

Palabras claves: comunicaciones industriales, DeviceNet, pasarela, protocolo HART, redes industriales.

Abstract

This paper demonstrates the feasibility of harmonizing two industrial protocols with very different characteristics, through the implementation of a hardware-software interface. HART is a very low speed protocol that operates with master-slave paradigm, connectionless oriented and with a defined set of commands, while DeviceNet is at least 100 times faster, works with the producer-consumer model, is connection-oriented and does not use commands but objects. The development is supported on the TINI (Tiny InterNet Interface) embedded system, which is Java programmable. The gateway acts as a slave on a DeviceNet network and as master of a HART network. This innovative equipment can be used to include HART devices in manufacture processes requiring few continuous variables.

Key words: DeviceNet, *gateway*, HART protocol, industrial communications, industrial networks.

Recibido: 16-03-2015
Modificado: 14-04-2015
Aceptado: 20-04-2015

1. Introducción

Existe un sinnúmero de protocolos para comunicación de equipos industriales con características únicas e incompatibles [1], enfocados en cubrir diferentes necesidades como instrumentación de procesos continuos, automatización de sistemas secuenciales, control de movimiento, distribución de inteligencia en sistemas de transporte, entre otros.

En la mayoría de las empresas existen equipos con diferentes protocolos de comunicación que deben hacer parte de un mismo proceso. La integración de la información proveniente de estos equipos puede hacerse en los equipos de supervisión y control como se muestra en la Figura 1(a) o directamente en el piso de planta, como aparece en la Figura 1(b). La primera opción requiere la extensión de todas las redes hasta el cuarto de control y la existencia de interfaces físicas en los PLCs o PCs para cada una de las redes, así como el software de administración de éstas. Con la segunda opción sólo se requiere una interfaz de comunicación en el dispositivo de control, pero debe existir una pasarela (*gateway*) que realice el intercambio de información entre las redes.

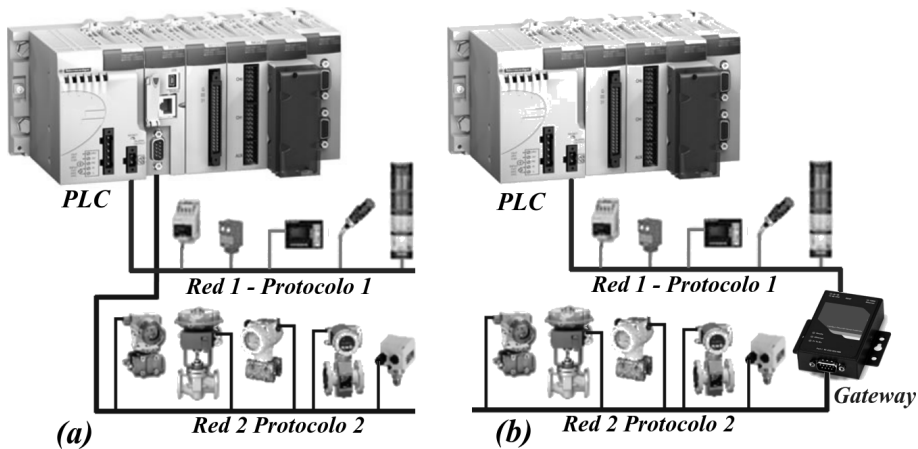


Figura 1. Opciones para coexistencia de varias redes en un proceso.

El diseño de *gateways* ha sido objeto de investigación y desarrollo por muchos años [2] [3] [4] [5] y se han reportado propuestas para insertar transmisores HART en redes con otros protocolos. Hyncica et al. [6] presentan tres opciones: Bluetooth, Modbus sobre Ethernet y Zigbee. Las dos redes inalámbricas no representan una opción que pueda ser utilizada industrialmente, por cuanto no se implementa el *stack* completo de un protocolo de comunicación industrial. Modbus sobre Ethernet, conocido como Modbus/TCP, sí es una alternativa viable en un ambiente industrial. Yanjun y Jun [7], al igual que Liu et al. [8] describen dispositivos *gateway* para insertar equipos HART en redes con el protocolo Profibus-DP, un protocolo de menor complejidad que DeviceNet.

Un estudio realizado en 2008 encontró que los protocolos HART y DeviceNet tienen fuerte presencia en la industria colombiana y particularmente en el Valle del Cauca, pero que sus bondades son poco conocidas y explotadas [9]. Por ello al interior del grupo de investigación

PSI y se definieron proyectos tendientes a formar ingenieros y a desarrollar equipos que aporten a la competitividad industrial mediante el uso adecuado de estas tecnologías. Este artículo presenta la implementación de una interfaz tipo *gateway* entre los protocolos HART y DeviceNet, que permite el uso de transmisores HART a través de una red DeviceNet y de la cual no se encuentran reportes en la literatura. Tampoco se ha encontrado un dispositivo similar en los portafolios de las empresas del sector de la automatización.

2. Los protocolos HART y DeviceNet

2.1. El protocolo de comunicación HART

HART (*Highway Addressable Remote Transducer*) fue introducido a mediados de los 80s por la compañía Rosemount Inc. como un estándar para la comunicación de transmisores de proceso. Representó la primera generación de comunicaciones digitales con dispositivos de campo, guardando compatibilidad con los transmisores convencionales de 4-20 mA. Por ello es quizás el protocolo con mayor número de nodos instalados a nivel mundial. Actualmente el protocolo es administrado por la *HART Communication Foundation* - HCF (www.en.hartcomm.org). Una buena descripción del protocolo, hasta la revisión 6, puede encontrarse en [10].

HART es un protocolo maestro-esclavo, que permite la co-existencia de dos maestros en el enlace. Puede usarse en modo punto a punto, con un único esclavo, o en red, en cuyo caso el número de esclavos puede llegar a 63 (a partir de la revisión 6.0). Utiliza una señal digital para transferir información de diagnóstico, configuración y medición.

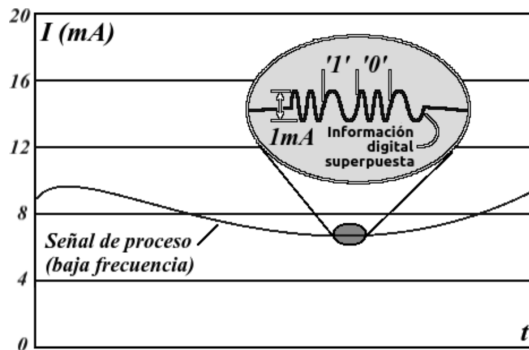


Figura 2. Señal digital HART sobre señal analógica de corriente 4-20 mA.

Aunque HART define varias posibilidades para la capa física, la más difundida es la FSK, basada en el estándar Bell 202, la cual utiliza modulación en frecuencia para la transmisión de señales digitales superpuestas a una señal de corriente analógica de 4-20 mA. Debido a que la modulación FSK posee un valor promedio igual a cero, la señal digital no causa desviaciones de la señal analógica, lo que permite la coexistencia de ambas señales en el mismo par de hilos. El estándar Bell 202 especifica una señal sinusoidal modulada en frecuencia con una amplitud

de 1 mA p-p y una tasa de transmisión de 1200 baudios, con 1200 Hz para representar el '1' binario y 2200 Hz para representar el '0' binario, como se muestra en la Figura 2.

El protocolo HART define dos modos de operación: *POLL* y *BURST*. En modo *POLL* cualquier transacción es originada por el maestro; el esclavo sólo responde cuando recibe un mensaje dirigido a él. La longitud y el retardo típicos de los mensajes permiten idealmente hasta 2,5 actualizaciones de una variable por segundo. Un esclavo configurado en modo de operación *BURST* produce mensajes sucesivamente sin solicitud directa del maestro, lo que permite realizar cerca de 3,5 actualizaciones por segundo, en el mejor de los casos.

La capa de aplicación del protocolo HART define tres grupos de comandos de acuerdo al tipo de servicio que ofrecen:

- Comandos universales: la mayoría son de implementación obligatoria en todos los dispositivos HART. Proveen las funciones básicas para el diagnóstico, la configuración y supervisión.
- Comandos de práctica común: ofrecen funciones comunes a muchos dispositivos de campo que permiten mejorar sus prestaciones. Su implementación es opcional.
- Comandos específicos del dispositivo: proveen funciones especiales únicas, diseñadas por el fabricante para un dispositivo en particular.

2.2. DeviceNet

DeviceNet es un protocolo abierto de comunicación entre equipos de nivel de proceso, para la conformación de redes entre dispositivos industriales simples (actuadores y sensores) y dispositivos de nivel alto (controladores), desarrollado por Allen Bradley en 1993 y administrado actualmente por la *Open DeviceNet Vendor Association* (ODVA – www.odva.org). En [11] puede encontrarse una descripción clara y concisa de la especificación.

La capa física de DeviceNet es definida por el estándar ISO 11898, en la cual el estado lógico '0' es un voltaje diferencial de 3 V y el '1' es un voltaje diferencial nulo. La red admite hasta 64 nodos. La velocidad configurada determina la longitud máxima de la red: 125 kbps – 500 m, 250 kbps – 250 m o 500 kbps – 100 m. El cable de red incluye un par de hilos con 24 VDC para alimentación de los dispositivos que lo requieran.

En la capa de enlace de datos, DeviceNet acoge la especificación CAN 2.0A, que se caracteriza por ofrecer una elevada seguridad en el transporte de datos. DeviceNet permite el intercambio de datos peer-to-peer, en el cual, cualquier dispositivo de la red puede producir y consumir mensajes. Utiliza un modelo basado en conexiones que facilita la comunicación entre aplicaciones.

DeviceNet ofrece al usuario una capa de aplicación orientada a objetos. Los objetos modelan el comportamiento del dispositivo a través de atributos y servicios. De hecho, las conexiones son objetos que se crean y eliminan dinámicamente durante la configuración y operación de la red. Se identifican dos tipos de conexiones: explícitas e I/O. Se establece una conexión

explícita para leer o escribir parámetros de un dispositivo, o para crear una conexión I/O. Se establece una conexión I/O para comunicar datos de proceso por medio de alguno de los siguientes mecanismos, que hacen parte de lo que la especificación denomina el Conjunto de Conexiones Predefinidas *Master-Slave*:

- Sondeo (*poll*): es un mecanismo solicitud/respuesta en el cual un dispositivo produce sus datos sólo después de que le son solicitados.
- Cíclico (*cyclic*): el dispositivo produce sus datos periódicamente, a una rata programada durante su configuración.
- Cambio de estado (*COS*): el dispositivo produce los datos de proceso sólo cuando haya ocurrido un cambio apreciable en ellos; dado que esto puede resultar muy espaciado en el tiempo, se programa también un reporte periódico para indicar que el dispositivo permanece conectado y funcionando, denominado heartbeat.
- Estroboscópico (*strobed*): el controlador envía una única trama mediante la cual indica a todos los dispositivos presentes en la red, programados con el mecanismo *strobed*, que deben producir sus datos.

Es así como durante el proceso de configuración de la red, es necesario asignar a cada dispositivo uno o varios mecanismos de transferencia de datos, teniendo en cuenta los requerimientos específicos de las variables que produce.

Es importante anotar que cualquier conexión creada en DeviceNet tiene un tiempo de vida, de tal forma que la conexión es eliminada si no se utiliza por un lapso de tiempo superior al especificado por el tiempo de vida.

3. Implementación de la pasarela (*gateway*)

3.1. Plataforma Hardware

En la figura 3 pueden identificarse los diferentes bloques que conforman el hardware; la plataforma seleccionada para implementar el Gateway Hart-DeviceNet es el sistema embebido TINI (*Tiny InterNet Interface*) desarrollado por Dallas Semiconductors, que utiliza Java J2SE como lenguaje de programación. La plataforma cuenta con un microprocesador DS80C390 de Dallas Semiconductors, doble puerto serial, un controlador Ethernet 10 Base-T y doble controlador CAN 2.0. Cada controlador CAN posee 15 centros de mensajes con filtros de aceptación configurables de forma independiente. La programación de la TINI se realiza a través del controlador Ethernet [12].

Además de la TINI, se emplea un módem Smar HI311ME con interfaz RS232 para la implementación de la capa física del protocolo HART y un conversor DC/DC que toma los 24 V de la red DeviceNet y genera 5 V para alimentar la TINI y el módem HART.

3.2. División Modular del Software

Para simplificar el diseño del software y aprovechar al máximo las características de la programación orientada a objetos, el sistema ha sido dividido en cuatro módulos (Tabla I), cada uno de los cuales está compuesto por uno o más objetos que en conjunto realizan una tarea específica e independiente de las demás.

Cada módulo del software fue implementado en un hilo de ejecución independiente (*thread*). Los hilos se comunican entre sí mediante espacios de memoria compartida y colas de mensajes. Al interior del bloque TINI, en la Figura 3, se muestra de manera abstracta la interacción entre los módulos software desarrollados.

El módulo *DeviceNet Server stack* es el encargado de implementar el *stack* del protocolo DeviceNet para un dispositivo tipo servidor solo grupo 2. Las principales características implementadas en el servidor DeviceNet son:

- Baud Rate: 125 Kbps.
- MacID configurable por software.
- Perfil: Adaptador de Comunicaciones (Perfil ID 0Chex).
- Soporte de fragmentación de mensajes I/O producidos.
- Soporte de reensamble de mensajes explícitos consumidos.
- No soporta el Administrador de Mensajes Desconectado UCMM
- Soporte de Mensajes Explícitos, Conexión I/O Poll y Conexión I/O Strobed del Conjunto de Conexiones Predefinido Master/Slave.

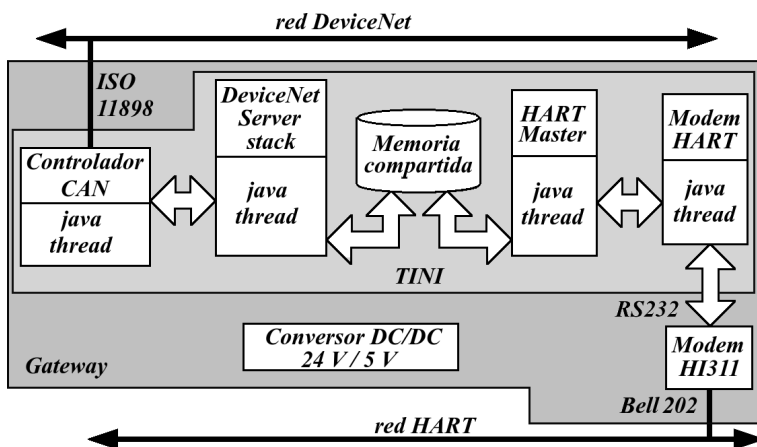


Figura 3. Diagrama de bloques del hardware e Interacción de los módulos software.

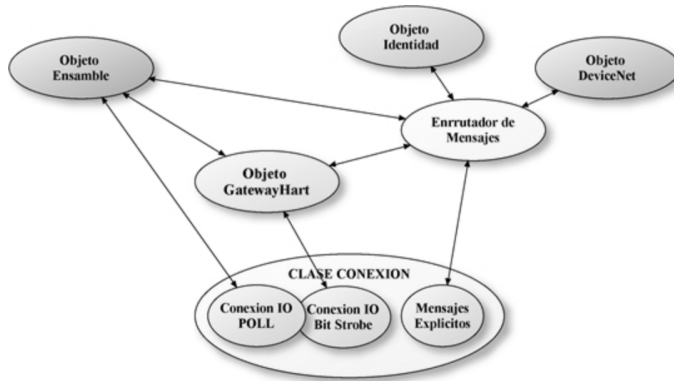


Figura 4. Modelo de objetos para el perfil Adaptador de Comunicaciones.

La ODVA no proporciona una definición para los objetos ensamble y los objetos aplicación necesarios para la operación de un dispositivo bajo el perfil Adaptador de Comunicaciones, ya que ellos dependen del tipo de protocolo a interconectar. La Figura 4 presenta el modelo de objetos implementado siguiendo las especificaciones del perfil y los requerimientos de la aplicación específica al protocolo HART.

El módulo Master Hart implementa una versión reducida del stack del protocolo HART revisión 5.0 para un dispositivo tipo maestro, con soporte de algunos comandos universales y de uso común. Las principales características del maestro HART implementado son:

- Revisión del protocolo: 5.0
- Lectura de las 4 variables dinámicas del proceso.
- Conexiones punto a punto y multipunto (red).
- No soporta el modo BURST, ni operación multimaestro.
- Soporta configuración básica de las funciones de medición: rango, función de transferencia, constante de tiempo.
- Permite administrar la topología de una red multipunto mediante la configuración de la dirección de sondeo (poll address) de los dispositivos esclavos activos.

Tabla I. Módulos software y objetos de la *gateway*

Módulo	Objetos	Descripción
<i>DeviceNet Server stack</i>	<i>Identidad</i>	Implementa la funcionalidad de esclavo del protocolo DeviceNet.
	<i>Enrutador</i>	
	<i>DeviceNet</i>	
	<i>Conexión</i>	
	<i>Ensamble</i>	
<i>Master HART</i>	<i>GatewayHart</i>	Implementa la funcionalidad de un maestro de una red HART.
	<i>HartMaster</i>	
	<i>HartFrame</i>	

Continúa

<i>Interfaces de comunicación I/O</i>	<i>CanBusDriver</i>	Configura y maneja el controlador CAN.
	<i>HartBusDriver</i>	Configura y maneja el puerto serial para conexión a la red HART a través del módem HI311.

Para este módulo se ha seleccionado un subconjunto de comandos universales y de práctica común requeridos para la supervisión de las variables del proceso y la configuración de las características de medición de los dispositivos HART. La Tabla II contiene el listado de los comandos HART implementados.

Tabla II. Comandos HART implementados

Com.	Tipo	Descripción
0	Universal	Lee identificador único del dispositivo, fabricante y revisión de comandos.
2	Universal	Lee valor de la corriente de lazo de la variable primaria.
3	Universal	Obtiene el valor de la corriente de lazo y 4 variables dinámicas.
6	Universal	Escribe dirección de sondeo del dispositivo.
14	Universal	Lee información del sensor: límites, mínimo span y unidades.
15	Universal	Lee información de medición: rangos, función de transferencia, constante de tiempo.
34	Práctica común	Modifica el valor de la constante de tiempo.
35	Práctica común	Modifica rangos superior e inferior de la medición.
47	Práctica común	Modifica la función de transferencia.

El módulo *Interfaces de Comunicación I/O* contiene los drivers necesarios para manejar el controlador CAN y el puerto serie que aloja el módem HART HI311, utilizados para conectar físicamente el sistema TINI con las redes DeviceNet y HART, respectivamente. Debido a su estrecha relación con el hardware, estos módulos serían las únicas secciones del proyecto que deberían modificarse para migrar la aplicación a otra plataforma con soporte de Java J2SE.

3.3. Archivo EDS

La especificación DeviceNet obliga a que cualquier dispositivo que se introduzca en una red disponga de un archivo tipo texto, denominado *Electronic Data Sheet* (EDS), que permite al software de configuración de la red identificarlo y parametrizarlo apropiadamente. Un archivo de este tipo fue creado para el *gateway* HART-DeviceNet, tomando como base la especificación del protocolo y los archivos de los dispositivos ya existentes.

4. Pruebas y Resultados

4.1. Plataforma de Prueba

Las pruebas se realizaron utilizando un PLC Allen Bradley Controllogix 5555, equipado con un módulo Ethernet/IP y una tarjeta scanner DeviceNet 1756-DNB. Esta tarjeta hizo las veces de controlador en la red DeviceNet. Como esclavos de la red se colocaron dos sensores de la misma marca (802DN y 871TM), un PLC Koyo DL-06 equipado con tarjeta D0-DEVNETS y el *gateway* HART-DeviceNet.

Para la conformación de la red HART se utilizaron: transmisores de temperatura Smar TT301 y ABB-TH02, transmisor de nivel Drexelbrook 509-15-X09, transmisor de caudal Siemens SITRANS FM MID Intermag 2 y transmisor de presión Moore 340D.

Las pruebas fueron realizadas con los aplicativos para comunicación, configuración y desarrollo de la firma Rockwell que serán presentados más adelante.

4.2. Operación del *gateway*

La capacidad de supervisión de los transmisores HART a través del *gateway* Hart-DeviceNet fue verificada usando el software RS-Networx para DeviceNet.

La Figura 5 muestra la red mientras está en curso el proceso de identificación de dispositivos. Puede notarse que el *gateway* fue detectado sin errores, al igual que los demás dispositivos presentes en la red. En la misma figura también aparece una foto del banco de pruebas en la que se observa el modem HART, dos sensores DeviceNet, un transmisor HART y el *gateway* con sus componentes internos.

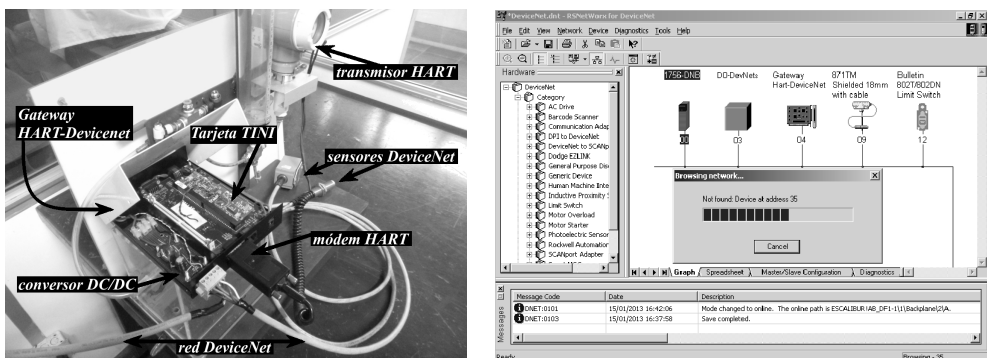


Figura 5. Proceso de identificación de dispositivos en la red DeviceNet y vista parcial del montaje.

La Figura 6 presenta la viñeta de General de la ventana de configuración del *gateway* Hart-DeviceNet, la cual contiene información básica del dispositivo, como el nombre, descripción, dirección e información contenida en el objeto identidad que facilita la identificación automática del dispositivo cuando se realiza la exploración de la red.

El monitoreo y configuración de los parámetros de los dispositivos DeviceNet se realiza a través de la viñeta Parámetros. Los parámetros corresponden a los atributos de instancia definidos por el objeto aplicación (*GatewayHart*) del perfil implementado.

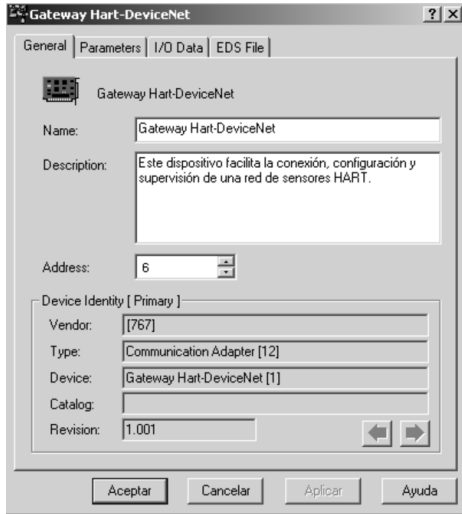


Figura 6. Ventana de propiedades del dispositivo Gateway Hart-DeviceNet.

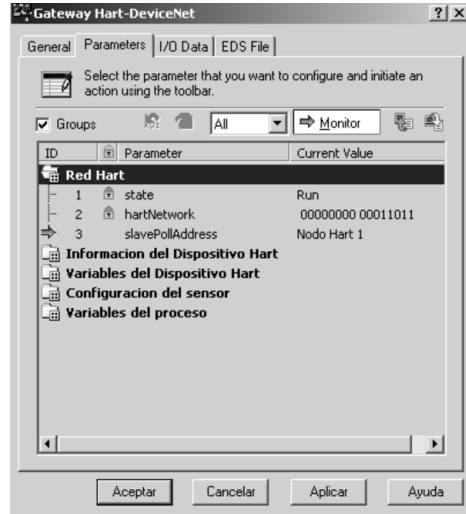


Figura 7. Información básica de la red HART.

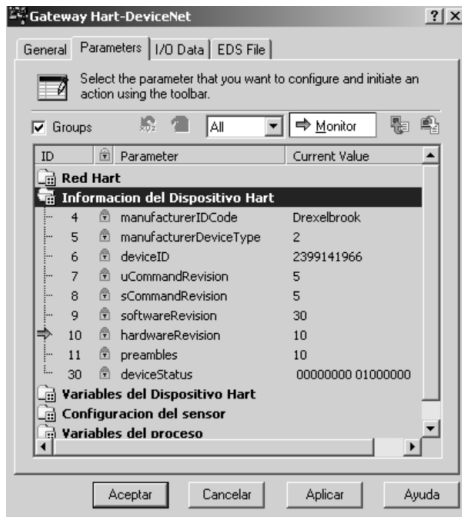


Figura 8. Información de identificación del dispositivo HART seleccionado.

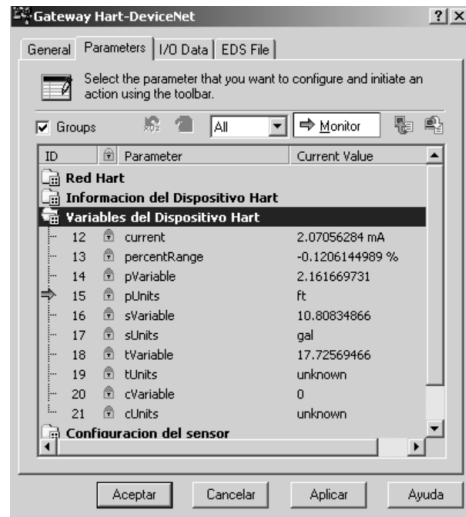


Figura 9. Variables dinámicas del dispositivo HART seleccionado.

Los parámetros han sido agrupados en 5 categorías de acuerdo a su funcionalidad.

- **Red Hart:** (Figura 7) contiene información básica de la topología de la red HART y el estado del *gateway* Hart-DeviceNet. De acuerdo con lo que indica el parámetro

hartNetwork en la Figura 7, la red tendría activos los transmisores con direcciones de sondeo 0, 1, 3 y 4.

- **Información del Dispositivo Hart:** (Figura 8) contiene información con la identificación del dispositivo HART seleccionado con el parámetro *slavePollAddress*. De acuerdo con la Figura 7, la información presentada en la Figura 8 es del nodo con dirección de sondeo 1.
- **Variables del Dispositivo Hart:** (Figura 9) contiene toda la información de medición obtenida del dispositivo HART seleccionado.
- **Configuración del sensor:** (Figura 10) permite la configuración de los parámetros de medición del sensor del dispositivo HART seleccionado.
- **Variables del proceso:** (Figura 11) contiene la variable primaria de todos los dispositivos activos en la red HART.

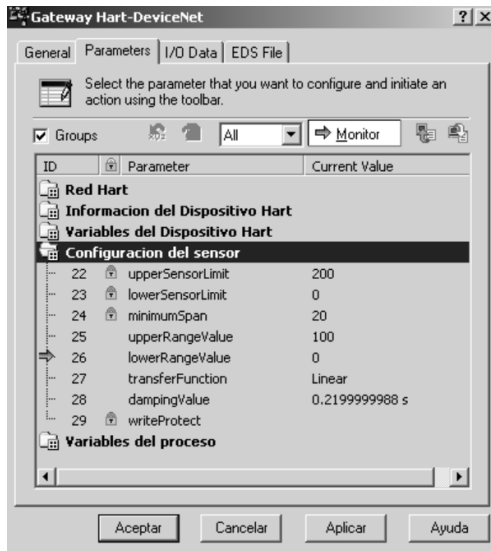


Figura 10. Parámetros de configuración del dispositivo HART seleccionado.

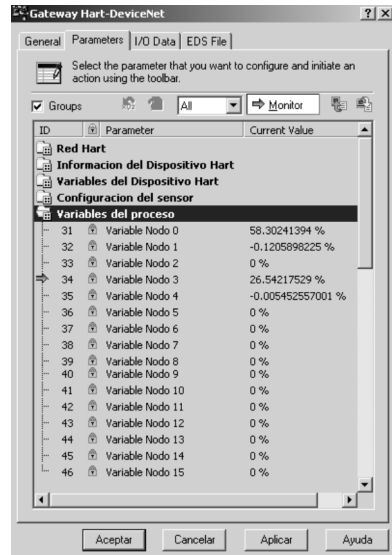


Figura 11. Variable principal de todos los dispositivos activos en la red HART.

4.3. Mensajería Explícita

Se verificó la respuesta del *gateway* HART-DeviceNet a mensajes explícitos mediante la solicitud de ejecución del servicio *StartGatewayHart*. Este servicio se encarga de inicializar el sistema y explorar la red Hart para descubrir los dispositivos activos. Para solicitar el servicio se utilizó la ventana *Class Instance Editor* de RS-Networx. La ventana mostró que el servicio fue exitosamente ejecutado y el parámetro *hartNetwork* fue actualizado con la nueva topología de la red HART.

Los demás servicios de la mensajería explícita fueron también probados exhaustivamente: cambio de dirección de sondeo, cambio de rango, cambio de unidades de ingeniería, etc.

4.4. Operación Continua

Para comprobar la transferencia de datos a través del *gateway* en forma continua, se implementó una estrategia de monitoreo con el programa RS-Logix 5000, para la visualización de las variables principales de cada uno de los dispositivos HART presentes en la red. La Figura 12 muestra la evolución de las diferentes magnitudes en el tiempo. Con 4 transmisores en la red HART, se observó que cada variable se actualizaba en menos de un segundo, algo similar a lo que ocurriría si se tuviera la red HART directamente conectada a un PC. Esta prueba muestra que la implementación de los mecanismos *I/O Poll* e *I/O Strobed* es exitosa y no genera problemas a los demás dispositivos de la red DeviceNet.

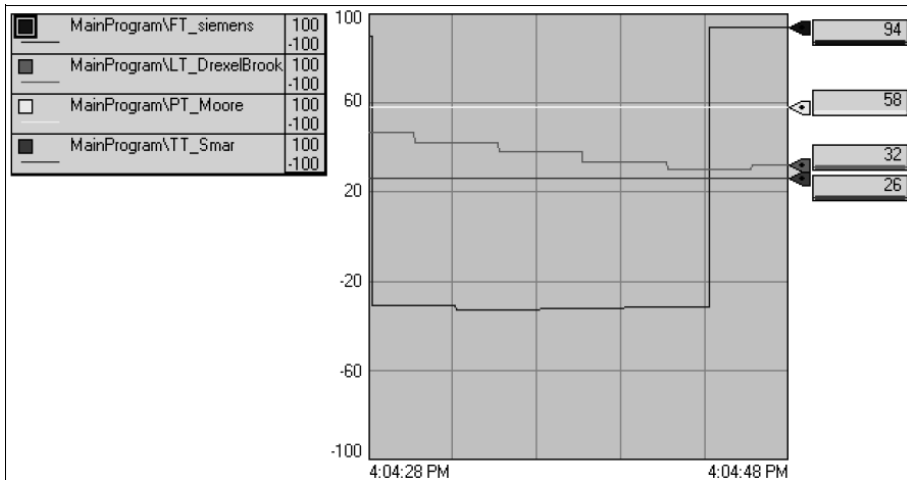


Figura 12. Sistema Scada de la planta PI2 utilizando el Gateway Hart-DeviceNet.

5. Discusión

Aunque la revisión 6.0 del protocolo HART permite 63 dispositivos en una red, es poco probable y nada práctico tener implementaciones con más de 15 dispositivos. Por ello se adoptó el límite definido por la revisión 5.0.

El *gateway* Hart-DeviceNet soporta dos mecanismos de intercambio de datos *I/O*: *Poll* y *Strobed*. La conexión *Poll* produce un arreglo de 64 bytes con la variable principal de todos los dispositivos activos en la red HART en formato IEEE754. La conexión *Strobed* produce un arreglo de tres bytes con el estado del sistema y la topología de la red HART. Sin embargo, el estado del sistema no es actualizado por el maestro HART mientras no se haga una solicitud por mensajería explícita.

La tasa de actualización de las variables tomadas de los transmisores no se ve afectada sustancialmente por la presencia del *gateway* y los distintos elementos de la red DeviceNet, debido a que la velocidad de HART es extremadamente baja. Este hecho fue determinante para la definición de la arquitectura del software, pues un mensaje de solicitud del Scanner DeviceNet debe ser respondido dentro de una ventana de tiempo de algunos milisegundos, mientras que una transacción en la red HART puede tardar más de medio segundo. La ejecución por hilos permite que el bloque *HART Master* actualice en forma permanente las variables de todos los transmisores activos, depositando la información en la memoria compartida. Por su parte, el bloque *DeviceNet Server stack* responde las solicitudes *Poll* y *Strobed* tomando la información existente en la memoria compartida, pudiendo así dar una respuesta inmediata. La Figura 13 muestra los diagramas de secuencia para atención al servicio una conexión I/O *Poll* y a una conexión explícita.

En la Figura 13(a) es claro que los mensajes I/O *Poll* son respondidos de forma inmediata usando la información existente en la memoria compartida (*dataBuffer*). Tal información es refrescada en forma permanente por el hilo Master HART. En la Figura 13(b), la solicitud explícita obtiene una respuesta inmediata con la información almacenada en los atributos de instancia al momento de procesar el servicio; la solicitud también dispara una secuencia de actualización del atributo, que puede ser en la memoria compartida o en el dispositivo HART, de acuerdo con el servicio solicitado.

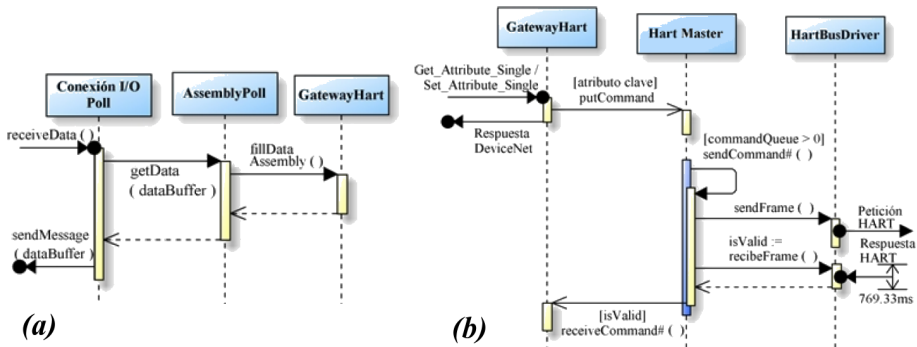


Figura 13. Diagramas de secuencia para atención a las solicitudes I/O y explícitas.

6. Conclusiones

Se ha presentado la implementación de un dispositivo tipo *gateway* que permite la integración de transmisores de proceso con protocolo HART en una red DeviceNet. El dispositivo se soporta en la plataforma TINI de Dallas Semiconductor, la cual mostró muchas ventajas para la implementación, pues dispone de controlador CAN y brinda la facilidad de programación en un lenguaje orientado a objetos, lo cual se adapta perfectamente a la especificación DeviceNet. El único inconveniente encontrado es que el controlador CAN limita la velocidad de transferencia de datos a 125 kbps. Dado que el desarrollo se hizo en Java, puede ser migrado con facilidad a otras plataformas, modificando únicamente los bloques software relacionados con los periféricos de comunicación (controlador CAN y puerto serie).

Para la comunicación con la red HART, el sistema Gateway Hart-DeviceNet implementa el protocolo HART revisión 5 para un dispositivo tipo maestro, con soporte del modo de operación Maestro/Esclavo para conexiones punto a punto y multipunto. El sistema utiliza un subconjunto de comandos universales y de práctica común para la supervisión de las variables del proceso y la configuración de las características de medición de los dispositivos HART. En el lado DeviceNet, el *gateway* implementa mensajería explícita y dos de los mecanismos del Conjunto de Conexiones Predefinidas MasterSlave: *Poll* y *Strobed*.

Las pruebas realizadas permiten afirmar que el *gateway* satisfizo las especificaciones de diseño y ofrece comunicación confiable sobre la red DeviceNet, con una buena tasa de actualización de las variables provenientes de los transmisores y sin afectar los demás dispositivos presentes en la red.

Este proyecto permite ofrecer a la industria nacional un producto que puede mejorar los procesos productivos con costos inferiores a los que tendrían otras soluciones. Con él se ha demostrado la capacidad de diseño de la ingeniería colombiana y se han generado documentos y herramientas útiles para realizar nuevas implementaciones y capacitar personal en el área de las comunicaciones industriales.

Para poder utilizar todo el potencial de los transmisores HART, es necesario acceder a cada transmisor con sus comandos universales. Por ello, se tratará de avanzar en el diseño del *gateway* para que provea soporte multimaestro, de forma tal que se pueda usar una terminal de mano u otro dispositivo de configuración, dado que hacerlo con mensajería explícita a través del *gateway* sería muy dispendioso. También se quieren explorar las posibilidades de migración de la pasarela a plataformas como Arduino y Raspberry- π , puesto que además de un menor costo, tienen mayor disponibilidad y soporte.

Ante el auge del WirelessHART, el desarrollo de una pasarela con la capa física inalámbrica del protocolo HART, definida en la revisión 7, plantearía un reto de mayores proporciones, pues además de que la capa física es considerablemente más compleja, se hace necesario rediseñar la capa de enlace de datos y abordar problemas como la generación y mantenimiento de rutas. Dado que los protocolos ControlNet y Ethernet/IP comparten con DeviceNet las capas superiores del modelo OSI (lo que se denomina el CIP en la terminología de la ODVA), sería interesante también evaluar las posibilidades de reuso del firmware para la construcción de pasarelas HART-ControlNet y HART-Ethernet/IP.

Referencias

- [1] Galloway, B.; Hancke, G.P.; Introduction to Industrial Control Networks, Communications Surveys & Tutorials, IEEE, 15(2), 2013, pp. 860 – 880
- [2] Saleh, K.; Jaragh, M.; Rafiq, O.; A methodology for the synthesis of communication gateways for network interoperability, Computer Standards & Interfaces, 17(2), 1995, pp. 193-207
- [3] Ninagawa, C.; Host-Based Architecture LON Communication Gateway for Building Air-conditioner Monitoring System, Technical Review, Mitsubishi Heavy Industries Ltd., (43) 2 , 2006.
- [4] Yoon, S.S.; Lee, J.C.; Cho S.J.; Jeon, J.W.; Gateway between high-performance Fieldbus and serial communication, International Conference on Control Automation and Systems (ICCAS), 2010, pp. 971–974.

- [5] Wu, H.Z.; Zhang, L.J.; Zhang, C.G.; Realization of Ethernet and RS485 communication gateway based on $\mu C/OS-II$ system, *Applied Mechanics and Materials* (Volume 39), 2010, pp. 416-420.
- [6] Hyncica O., Fiedler P., Bradac Z., Kucera P., Honzik P., Protocol gateways for HART sensors, *IFAC Proceedings Volumes* (IFAC-PapersOnline), v 9, PART 1, 9th IFAC Workshop on Programmable Devices and Embedded Systems, PDES 2009 – Proceedings, 2009, pp. 194-197.
- [7] Yanjun F., Jun X., An approach for interoperation between heterogeneous fieldbus systems, *IEEE Symposium on Emerging Technologies and Factory Automation, ETFA, 2 2 OF 2 VOL, 10th IEEE International Conference on Emerging Technologies and Factory Automation, 2005*, pp. 239-243.
- [8] Liu, J.; Fang, Y. ; Zhang, D.; PROFIBUS-DP and HART Protocol Conversion and the Gateway Development, *2nd IEEE Conference on Industrial Electronics and Applications- ICIEA, 2007*, pp. 15–20.
- [9] Barandica A., Una propuesta para la apropiación de la tecnología HART y su transferencia hacia el sector industrial, tesis de maestría, Escuela de Ing. Eléctrica y Electrónica, Universidad del Valle, Cali, Colombia, 2008.
- [10] Bowden R., HART Field Communications protocol. A Technical Overview. HCF, 2007.
- [11] RTA Automation, DeviceNet™ Unplugged – A View “Under the Hood” for End Users, 2 de abril de 2015, disponible en <http://www.rtaautomation.com/technologies/devicenet/>
- [12] Loomis D., *The TINI Specification and Developer's Guide*, (Primera Edición), 2001.

Asfur Barandica López

Nació en Cali, Colombia. Es Ingeniero Electricista de la Universidad del Valle, Cali, Colombia. Obtuvo su título de Maestría en Ingeniería con énfasis en Electrónica en la misma universidad. Actualmente se desempeña como profesor en el área de Informática Industrial en la Universidad del Valle de Cali, Colombia, y pertenece como investigador al grupo Percepción y Sistemas Inteligentes, donde realiza estudios sobre transductores inteligentes y comunicaciones industriales.

e-mail: asfur.barandica@correounivalle.edu.co

Andrés Ernesto Guevara Escobar

Nació en Cali, Colombia. Es Ingeniero Electrónico de la Universidad del Valle, Cali, Colombia. Actualmente se desempeña como arquitecto de soluciones en el área de facturación electrónica en Carvajal Tecnología y Servicios en Cali, Colombia.

e-mail: andresegs@hotmail.com