

Plataforma autoconfigurable de monitoreo remoto para aplicaciones móviles en WSAN

*Self-Configurable Monitoring Framework for Mobile
Wsan's Applications*

Espinoza-Ruiz Adolfo

Departamento de Ingeniería Eléctrica y Electrónica
Instituto Tecnológico de Sonora
Correo: adolfo.espinoza@itson.edu.mx

Ruiz-Ibarra Erica

Departamento de Ingeniería Eléctrica y Electrónica
Instituto Tecnológico de Sonora
Correo: erica.ruiz@itson.edu.mx

García-Berúmen Armando

Departamento de Ingeniería Eléctrica y Electrónica
Instituto Tecnológico de Sonora
Correo: armando.garcia@itson.edu.mx

López-Chaides Juan Antonio

Departamento de Ingeniería Eléctrica y Electrónica
Instituto Tecnológico de Sonora
Correo: juantonio52@hotmail.com

Cortez-González Joaquín

Departamento de Ingeniería Eléctrica y Electrónica
Instituto Tecnológico de Sonora
Correo: joaquin.cortez@itson.edu.mx

Ruiz-Ibarra Joel

Departamento de Ingeniería Eléctrica y Electrónica
Instituto Tecnológico de Sonora
Correo: joel.ruizi@itson.edu.mx

Información del artículo: recibido: febrero de 2014, reevaluado: marzo de 2014, aceptado: noviembre de 2014

Resumen

En la actualidad, cada vez es más común encontrar redes inalámbricas realizando diversas aplicaciones, esto hace que no exista una alternativa ideal para todos los casos, sino que se desarrollen propuestas adecuadas para cada tipo de contexto. En este trabajo se presenta una propuesta para una red de nodos interconectados por medio de transceptores IEEE 802.15.4, que integran un sistema cuyo propósito es el monitoreo de sensores y actuadores. El objetivo principal es el desarrollo de una plataforma de propósito general para una fácil, flexible, dinámica y automática formación de redes inalámbricas de sensores en arquitectura estrella por medio de microcontroladores, que además de realizar las tareas de comunicación, tienen capacidad de procesamiento y control, con lo que el sistema habilita exitosamente la gestión de redes de monitoreo de sensores.

Descriptores:

- redes inalámbricas de sensores y actores
- monitoreo remoto
- microcontroladores

Abstract

Wireless networks have been evolving toward higher complexity applications. It makes sense not to expect an one-fits-all application framework. These applications require a particular framework profile in order to solve particular issues. In this paper we propose a Wireless Sensor and Actuator Network (WSAN) framework profile, which solves the particular issue of writing and reading memory slots on the sensor node's microcontroller. As simple as it sounds this application opens the framework to a wide set of possibilities, which could be reading a sensed data, or even switching the purpose of the whole node just by changing a pointer value on the remote microcontroller memory. This framework is suitable for a star topology amidst an IEEE802.15.4 standard. It is also able to control, process and communicate successfully among the network nodes.

Keywords:

- wireless sensor and actor networks (WSAN)
- WSN
- micrcontrollers
- monitoring framework

Introducción

Como consecuencia de la reducción de costos de microcontroladores y transreceptores inalámbricos, cada vez es más común aplicar redes de monitoreo y control inalámbrico en sistemas productivos tales como invernaderos, granjas acuícolas u otros sistemas productivos en donde es necesario monitorizar parámetros físicos y realizar acciones de control en función de los valores de dichos parámetros. Las redes utilizadas en todos los casos mencionados tienen las similitudes siguientes:

- a) La lectura de los parámetros del proceso se realiza en puntos distribuidos geográficamente en áreas de algunas decenas o cientos de metros.
- b) Generalmente los sistemas son lentos, por lo que los datos recolectados son de baja transferencia.
- c) Se requiere control distribuido, de tal forma que un nodo puede tomar decisiones de control con base en la lectura de parámetros, pero los valores de referencia se envían a través de la red, desde el nodo coordinador que actúa como *sink* de la red.

Según lo anterior, el objetivo del trabajo fue desarrollar una plataforma para aplicaciones en red de microcontroladores de bajo costo y transreceptores XBee de la serie 1, cuya gestión de funciones y configuración en red sean controladas automáticamente por los microcontroladores, ayudando a orientar los esfuerzos de los usuarios para el desarrollo de aplicaciones específicas que ejecutan los nodos de la red. Para lograr esto, el *software* en los nodos se dividió en dos módulos, uno encargado de la gestión de comunicación y el otro de la aplicación particular que realiza el nodo. Estos módulos son completamente independientes entre sí por lo que al diseñar la aplicación no es necesario considerar la comunicación.

Estado del arte

De las aplicaciones que forman parte de las redes inalámbricas de área personal (WPAN) existen aquellas orientadas a cubrir diversas necesidades en el mercado actual. Al diversificar la movilidad en las redes inalámbricas inevitablemente comienza a segregarse el desarrollo de tecnologías orientado para aplicaciones de monitoreo y control en sensores y actuadores, de aquellas cuyo enfoque es el desarrollo de aplicaciones de multimedios.

Ciertamente, algunas aplicaciones pueden permitirse pagar el costo de agregar sistemas de comunicaciones inalámbricos de gama alta como los teléfonos celulares (Gutiérrez, 2004), sin embargo, muchas otras aplicaciones no requieren una transferencia de datos tan grande ni pueden costear un consumo de energía excesivo.

El grupo de trabajo IEEE 802.15 define tres tipos de clases para las redes WPAN, que se distinguen por transferencia de datos, consumo de energía y calidad de servicio (QoS). Las WPAN de gama alta (IEEE std. 802.15.3) que son útiles para aplicaciones que requieren una alta calidad de servicio. En su gama media, Bluetooth se diseña como un remplazo para medios alámbricos en aplicaciones diseñadas para dispositivos móviles y organizadores personales (PDA) con una calidad de servicio necesaria para aplicaciones de voz o transferencia de archivos, tales como la aplicación propuesta por Cordova *et al.* (2014). La última clase de WPAN, redes inalámbricas de área personal de baja transferencia (LR-WPAN) IEEE 802.15.4, encaja en las necesidades de aplicaciones que requieren una baja transferencia de datos y calidad de servicio (Gutiérrez, 2004; Ramjee y Deneire, 2006). Con un consumo de energía promedio de 211μW (Bougard *et al.*, 2005), el estándar IEEE 802.15.4 ofrece una alternativa a sistemas

diseñados para aplicaciones que requieran un bajo costo y una escasa o nula infraestructura para su diseño e implementación, maximizando su portabilidad y eficiencia en el consumo energético para aplicaciones móviles en redes de sensores. Algunos trabajos como el reportado en Espinosa y Rendón (2012) presentan un sistema para el monitoreo de granjas acuícolas.

Como resultado del reciente interés en tecnologías por redes de sensores y la infinidad de aplicaciones favorecidas por la implementación de estándares de bajo consumo de energía, ideales para aplicaciones de control y monitoreo que comparten características en común para la gestión y formación de redes, se requiere desarrollar una plataforma que permita la implementación de redes robustas que puedan emplearse para diferentes propósitos.

Desarrollo

Este proyecto provee un conjunto de librerías en lenguaje ANSI C, que permiten la comunicación entre dispositivos remotos. Para simplificar la metodología de diseño se requiere que los módulos operen de forma reproducible, de forma, que el mismo código sirva para enlazar N cantidad de dispositivos en la misma red. Esto reduce la complejidad del sistema a tres módulos principales: computadora personal, módulo coordinador y módulo remoto, como se muestra en la figura 1.

La computadora personal se encarga de proveer una interfaz al usuario que permite comunicarse con los dispositivos remotos mediante una conexión USB CDC, que solicita y modifica los parámetros utilizados por las aplicaciones en los microcontroladores remotos; asimismo tiene la capacidad de realizar pruebas para solicitar la potencia de transmisión en -dBm de todos los dispositivos remotos alojados en la red.

El módulo coordinador es el que se encarga de realizar las gestiones de red e interpretar las órdenes enviadas desde la PC. Se compone de un microcontrolador (MCU) principal con un módulo RF. La comunicación entre ambos dispositivos se realiza mediante el modo de operación API de los módulos XBee OEM RF. El MCU principal configura al módulo RF para iniciarse como coordinador y mantiene alojada la infor-

mación de los dispositivos remotos, tales como dirección física y número de dispositivo. Otra de las funciones del MCU coordinador es resolver mensajes de asociación enviados por los dispositivos remotos. Acorde a la lista de dispositivos alojados en su memoria, el MCU responde indicando un número de dispositivo a ser asignado, retenido o modificado por los dispositivos remotos. De esta manera, se evitan conflictos en la red y la interfaz de usuario, ya que puede referenciarlos de manera simple y eficaz, evitando que los identificadores cambien entre ciclos de energía. El MCU coordinador guarda las direcciones físicas de los dispositivos remotos en la memoria volátil; por consiguiente, se implementa una resolución de direcciones, en la cual el MCU coordinador busca a un dispositivo remoto por su número asignado, en caso de que no cuente con su dirección física.

Los nodos finales del sistema o módulos remotos constan de un MCU y un módulo RF configurado como *end device*. Al igual que el módulo coordinador utiliza el modo de operación API para realizar la configuración y establecer comunicación mediante su respectivo módulo RF. El MCU remoto gestiona el modo de operación de su módulo RF e interpreta las órdenes enviadas desde el coordinador. El dispositivo remoto tiene la función de establecer comunicación con su coordinador cada vez que es energizado. De esta manera, realiza una petición de asociación donde el MCU coordinador resuelve un número de dispositivo para su operación. Dicho parámetro se almacena en el módulo RF del dispositivo remoto. Solo el coordinador decide si el dis-

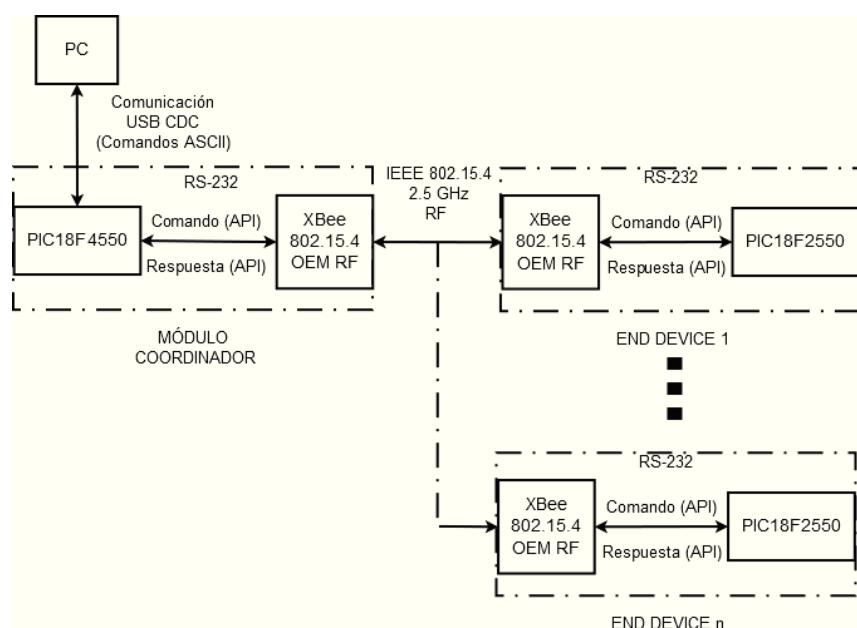


Figura 1. Diagrama de bloques del sistema

sitivo remoto opera con el mismo número o si es necesario asignar uno nuevo. La petición de asociación también tiene la función de comunicar al dispositivo remoto la dirección física del coordinador. Tras iniciarse, el dispositivo remoto realiza un escaneo activo buscando todos los coordinadores a su alrededor. Al finalizar, el módulo RF entrega un reporte al MCU con la información de hasta 5 coordinadores encontrados. Posteriormente, el dispositivo envía solicitudes de asociación a cada uno de ellos mediante un mensaje dirigido al MCU del coordinador. Al recibir una respuesta, el dispositivo remoto opera en los parámetros de dicho coordinador: Canal y PAN ID.

Acceso a módulos remotos mediante pizarra de configuración

Con la finalidad de que la aplicación en la PC pueda actuar sobre las aplicaciones en los módulos remotos, se decidió implementar en estos un vector de configuración, el cual guardó parámetros de configuración de las aplicaciones implementadas sobre los MCU de los dispositivos finales. La PC usando la interface de comunicación inalámbrica, tiene acceso a esta región de la memoria mediante una rutina de servicio a interrupciones del puerto serie en el módulo remoto, la cual atiende los comandos recibidos por la PC en forma totalmente independiente de la aplicación que se ejecuta en el módulo. De esta forma, el vector puede leerse y escribirse por ambas aplicaciones según sea necesario.

Adicionalmente a los datos escritos en ese vector, este proyecto contempla la posibilidad de que la PC solicite el RSS (*Received Signal Strength*) de un determinado dispositivo en cualquier momento. Un sistema calendarizado se encarga de rotar las distintas tareas alojadas por las aplicaciones del módulo remoto.

Comandos de texto

Los comandos de comunicación se envían desde la PC como texto mediante un puerto USB CDC COM e interpretados por el microcontrolador. Los comandos de texto y su sintaxis se describen a continuación:

- *Comando de lectura: "R [dispositivo] [índice]"*. Con este comando la PC, a través del microcontrolador, solicita el valor localizado en la posición de la pizarra indicada por “índice” en el dispositivo remoto señalado por

“dispositivo”. El MCU *end device* determina si se está tratando de leer una localidad válida en su pizarra. Si la orden es válida, el MCU remoto entrega el valor solicitado al MCU coordinador, quien lo envía como texto a la PC; de lo contrario, notifica al MCU coordinador, quien en turno despliega el siguiente mensaje de error a la PC “Se intentó leer un registro no indexado del dispositivo remoto”.

- *Comando de escritura: "W [dispositivo] [índice] [valor]"*. Este comando se encarga de solicitar al dispositivo definido que escriba el *valor* anexado en el comando en la localidad de su pizarra señalada por *índice*. El coordinador notifica el estado de la transmisión a la PC mediante los mensajes “Valor escrito correctamente” y “Se intentó escribir en un registro no indexado del dispositivo remoto”.
- *Potencia de transmisión: "P [dispositivo]"*. Solicita enviar un paquete de prueba al dispositivo remoto señalado para leer el parámetro RSS y luego enviarlo a la PC. El valor que regresa el dispositivo remoto se expresa en –dBm.
- *Reiniciar Red: "KILL"* envía una señal *broadcast* a todos los dispositivos alojados en la red para que los mismos borren el número de dispositivo que les fue asignado y reinicien a su microcontrolador. De esta manera, el MCU coordinador se encargará de resolver una nueva lista de dispositivos y asignarles un número nuevo a todos los miembros de la red.

Interfaces de comunicación serial

Este trabajo emplea dos interfaces de comunicación serial entre dispositivos: la primera, se refiere a la comunicación por medio de USB, configurado como dispositivo de clase de comunicaciones para comunicar al microcontrolador coordinador con la PC; la segunda, enmarca la comunicación por medio de RS-232 entre microcontroladores y los módulos XBee (OEM 802.15.4), mediante la interfaz de aplicación programable (API) provista por Digi Internacional.

La interface de comunicación USB hace uso de librerías genéricas que permiten establecer su funcionamiento mediante drivers genéricos CDC, los cuales funcionan como un reemplazo que permite emular una



Figura 2. Estructura básica para las tramas API (Bougard et al., 2005)

conexión RS-232 con una computadora personal (García, 2008).

El modo de operación API es una alternativa a la operación transparente por default. API se basa en tramas que extienden el nivel en el que un dispositivo anfitrión puede interactuar con las capacidades de red de un módulo XBee. Cuando un módulo RF opera en modo API, todos los datos de la interfaz entre microcontrolador y módulo RF se contienen en tramas que definen operaciones o eventos en el módulo. API provee medios alternativos para configuración y direccionamiento en el microcontrolador encargado del funcionamiento de los módulos. Se pueden enviar tramas al módulo RF que contengan direcciones de destino y carga de datos en lugar de utilizar el modo de comando para modificar las mismas. El módulo responde con tramas al microcontrolador que contienen estado de los paquetes, así como la fuente, RSS y carga de datos para cada paquete de entrada.

El esquema básico para las tramas API se detalla en la figura 2. El primer byte funciona como un delimitador de inicio, que advierte tanto al dispositivo anfitrión como al módulo RF que está por recibirse una trama nueva. El segundo y tercer byte especifican en un entero de 16 bits la longitud del paquete API. Dicho paquete se refiere a la información que será entregada a la aplicación por el módulo XBee o dispositivo anfitrión. Por último, el *checksum* forma parte de un protocolo de comprobación de errores, este Byte se calcula del paquete API sumando todos sus componentes y sustrayendo su resultado de “0xFF” (255 decimal).

En lo que respecta a la estructura del paquete API incluido en la trama, un encabezado principal (denominado identificador de paquete específico API, o API ID) define al módulo o dispositivo anfitrión el propósito del paquete en cuestión. La tabla 1 enlista el valor para cada uno de los identificadores. Para mayor información referente a los modos de operación de los módulos XBee, las tramas API y su funcionamiento consultar MaxStream (2007).

Tabla 1. Diferentes tipos de identificadores API

Valor hexadecimal	API ID
0x8A	Estado del modem
0x08	Comando AT
0x09	Comando AT a cola
0x88	Respuesta comando AT
0x00	Petición de transmisión: formato de 64 bits
0x01	PeticIÓN de transmisión: formato de 16 bits
0x89	Estado de la transmisión
0x80	Recepción de paquete RF: formato
0x81	Recepción de paquete RF: formato 16 bits

Interfaces de comunicación inalámbrica

Esta sección describe el desarrollo de los procesos relacionados con la configuración, operación e intercambio de paquetes entre módulos remotos. Debido a la complejidad que conlleva configurar diferentes dispositivos para operar bajo los mismos parámetros de transmisión, es necesario desarrollar un proceso que se encargue de poner en sintonía los dispositivos que forman parte de la red, tomando en cuenta posibles fallas e interrupciones de energía que se puedan presentar. Para esto se diseña esta interface mediante tres procedimientos principales: inicialización, asociación, y comunicación.

En la inicialización, los microcontroladores proceden a identificar o corregir los parámetros de funcionamiento de su respectivo módulo inalámbrico. Además, el microcontrolador a cargo de la gestión del módulo XBee coordinador, recupera de su memoria EEPROM la lista de dispositivos asignados con anterioridad. Los microcontroladores proceden a verificar si se encuentra habilitado el modo de operación API, el tipo de dispositivo con el cual se desea operar (coordinador o dispositivo final), recupera el número de dispositivo con el cual operan en la red y por último, si existen cambios en los parámetros del módulo XBee, solicita que estos se guarden en la memoria no volátil del módulo RF.

En el proceso de inicialización, el MCU central configura al XBee coordinador, de modo que cualquier equipo pueda intentar asociarse a él. Mientras, el *end device* se configura para operar únicamente con los dispositivos coordinadores que coincidan con sus parámetros de canal y PAN ID. A su vez, el MCU configura a su módulo RF para que realice un proceso llamado escaneo activo al encenderse. Este escaneo selecciona un canal y transmite una baliza de difusión (*broadcast*) a toda dirección física, lógica y de red (PAN ID) que se encuentre en ese canal. Después escucha por balizas de cualquier coordinador operando en ese canal. Una vez que expira el tiempo de espera en ese canal el escaneo activo selecciona otro canal y nuevamente difunde una petición en busca de más coordinadores. Este proceso continúa hasta que se escanean todos los canales o se descubre un total de cinco coordinadores. Este proceso lo ejecuta automáticamente el módulo RF para seleccionar el mejor canal de operación y PAN ID antes de comenzar a funcionar, de modo que el MCU debe asegurarse de leer los parámetros del módulo XBee después del encendido. Sin embargo, este proceso también puede llamarlo el MCU, tal caso forma parte del proceso de asociación por parte de los *end devices* y se explica con detalle más adelante.

La asociación es el establecimiento de la membresía entre los *end devices* y un coordinador. El establecimiento de dicha membresía es útil en escenarios que requieren una unidad coordinadora central para recabar datos de varias unidades remotas, asignar canales de operación e identificadores de red (PAN ID). Una red PAN consiste en un coordinador y uno o más dispositivos finales, cada dispositivo en una PAN tiene un identificador de red de área personal (PAN ID). Los PAN ID deben ser únicos para prevenir fallas de comunicación entre los PAN.

Los módulos XBee 802.15.4 cuentan con procesos de asociación automáticos; sin embargo, no se elige esta opción, debido a que se requiere que la red pueda formarse dinámica y aleatoriamente, por lo que es necesaria la intervención de los MCU en este proceso para prevenir que los *end devices* se asocien con dispositivos que no pertenecen a su misma aplicación.

El MCU coordinador toma un rol importante para la identificación y asociación de dispositivos, debido a que se encarga de asignar un número de operación a todos los dispositivos remotos, para poder resolverlos a una dirección física en todo momento, siempre que la aplicación alojada en la PC lo solicite. Este proceso de asociación se realiza mediante un procedimiento de solicitud, asignación y respuesta, ejecutado por los dispositivos que forman parte de la red, mediante la implementación de un byte que sirve como campo para comandos de red, que se envía entre los microcontroladores a través de los distintos módulos RF. La figura 3 muestra el campo de comandos de red y sus distintos valores pertinentes a la asociación de dispositivos inalámbricos.

El comando “solicitud de asociación”, se envía desde el MCU remoto con destino al MCU coordinador con el propósito de identificar el dispositivo con el cual el *end device* debe asociarse. Seguido al byte de comando, se encuentra un valor que identifica el número de dispositivo deliberado por el *end device* en el proceso de inicialización. El MCU coordinador guarda la dirección del dispositivo y procede a realizar uno de los siguientes pasos:

0x00 = Transmisión Normal	Comando de red
0x08 = Solicitud de asociación	
0x03 = Nuevo número de dispositivo	

Figura 3. Byte comando de red

- Si el número de dispositivo es cero (no asignado), el MCU coordinador asigna un número de dispositivo que no haya asignado previamente contando de forma ascendente.
- Si el número de dispositivo es diferente de cero, el MCU verifica que dicho número haya sido asignado previamente y envía mensaje al *end device* para que conserve su mismo número de dispositivo.
- Si el número de dispositivo es diferente de cero, pero el MCU no había asignado ya ese número. El MCU coordinador asigna un nuevo número de dispositivo al MCU *end device*.

El comando “nuevo número de dispositivo” funciona de manera menos compleja. Dicho comando se envía desde el MCU coordinador e indica al *end device* si debe cambiar o conservar el mismo número de dispositivo que poseía. Seguido del Byte que indica el comando, aparece el número de dispositivo que el coordinador desea asignar al MCU *end device*, cero en caso de que el coordinador desee que el *end device* opere con el mismo número de identificación.

Otra parte importante del proceso de inicialización es el descubrimiento de los nodos presentes. Una vez realizado su proceso de inicialización, el MCU coordinador procede a buscar todos los dispositivos que se encuentren asignados en su lista para recuperar sus direcciones físicas. Siguiendo la secuencia mostrada en el diagrama de la figura 4 se logra que el coordinador tenga siempre conocimiento de las direcciones físicas de los módulos asociados a él. En adición, el coordinador siempre identifica al mismo módulo bajo el mismo número asignado cuando estableció membresía en la red por primera vez, a pesar de que existan interrupciones en la energía, logrando un método de identificación dinámico, automático y sencillo donde la aplicación procederá a referirse a los módulos por el número que le fue asignado, el cual no depende de ningún módulo RF en particular, abriendo la posibilidad de que los módulos puedan ser remplazados en caso de algún desperfecto o descompostura.

Después del proceso de inicialización, el módulo RF *end device* realiza un escaneo activo de la misma manera que el coordinador al iniciarse; sin embargo, difiere, en que esta vez el proceso se encuentra monitorizado por su respectivo microcontrolador. El propósito del escaneo activo en los dispositivos finales es tener acceso a una lista de parámetros de hasta cinco coordinadores encontrados por el escaneo en las cercanías. De esta manera, el micro controlador puede solicitar al módulo remoto intentar asociarse con los diferentes coordinadores mediante el comando mostrado en la figura 3. El

proceso mostrado en la figura 5 se repite una vez por cada coordinador encontrado en el proceso de escaneo activo hasta localizar el microcontrolador adecuado.

Una vez que se recuperan todos los datos de los coordinadores en las cercanías, el MCU *end device* ordena a su módulo RF sintonizarse con el canal de transmi-

sión de cada coordinador encontrado, para realizar una solicitud de asociación de MCU a MCU en modo de *broadcast*. El contenido de este paquete incluye como carga de datos el número de dispositivo deliberado por el proceso de inicialización (sí existe). El MCU coordinador procede a deliberar si el número de dis-

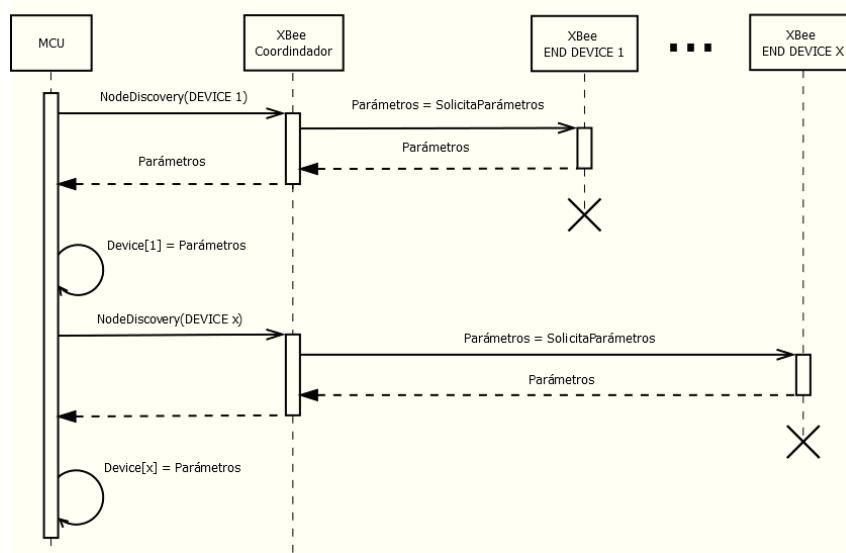


Figura 4. Diagrama de secuencia para el descubrimiento de nodos

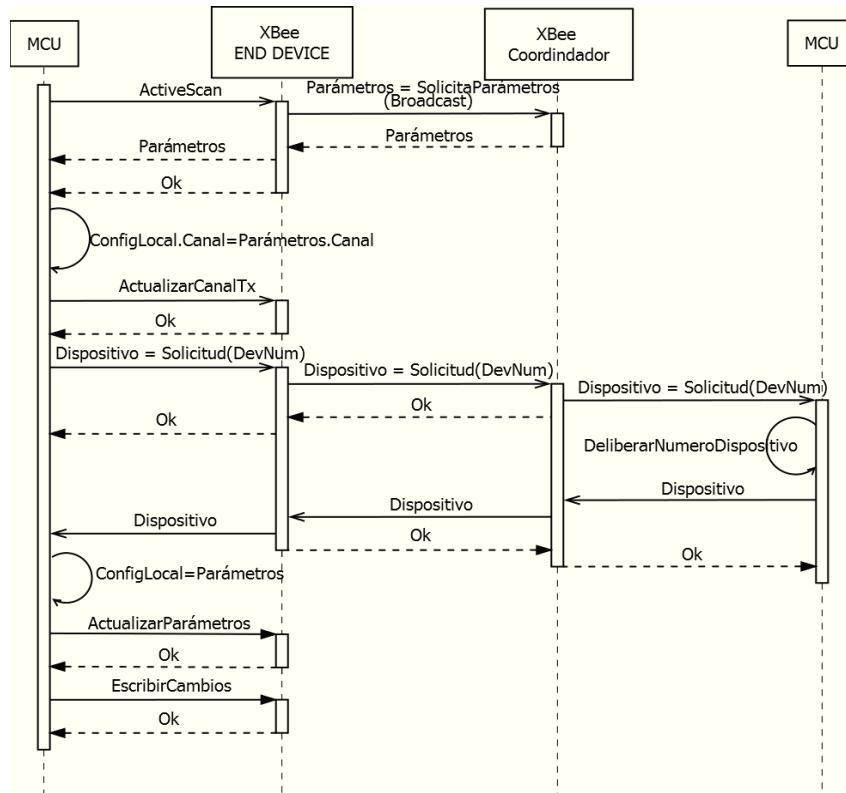


Figura 5. Asociación de un dispositivo *end device*, caso exitoso

positivo no crea conflictos con su tabla de asociación y responde con un paquete del mismo tipo, que le indica al MCU *end device* si debe continuar utilizando el mismo número de dispositivo o si se le asigna otro. Este proceso se repite una vez por cada coordinador encontrado hasta recibir una respuesta de asociación por parte del MCU coordinador. Una vez que se han confirmado los datos, el MCU ordena a su módulo RF actualizarse y respaldar los nuevos parámetros de red en su memoria no volátil.

Como paso final para implementar una red funcional se dispone de esta etapa del diseño, la cual se encarga de gestionar las comunicaciones entre dispositivos remotos mediante el módulo coordinador central. Los procesos asociados con la comunicación entre dispositivos remotos son menos complejos debido a que usan toda la base de pilas, procesos y funciones implementadas hasta este punto.

La transmisión de paquetes entre módulos remotos se lleva a cabo mediante tramas API, donde la solicitud de transmisión para paquetes de 64 bits es la implementada para cumplir con los objetivos planteados en este trabajo.

Pruebas y resultados

Debido a que resulta difícil monitorear todos los eventos y las tramas en los puertos serie de los dispositivos involucrados, se opta por utilizar las tarjetas de desarrollo RS-232 provistas por maxstream/Digi International, para monitorear el cambio de los valores en las variables de configuración de los módulos RF para las pruebas realizadas en este punto. La figura 6 muestra la tarjeta de desarrollo RS-232 utilizada para verificar los valores escritos en los módulos XBee RF 802.15.4.

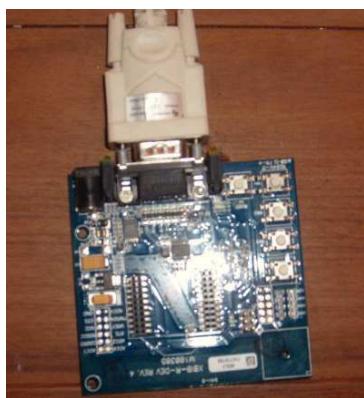


Figura 6. Tarjeta de desarrollo XBIB-R-DEV REV 4

Se utilizó el software X-CTU para interactuar y configurar los módulos RF. Esta herramienta se organiza en las siguientes cuatro pestañas:

- Ajustes de PC: configura los puertos seriales de la PC para interactuar con los módulos de radiofrecuencia.
- Pruebas de rango: permite probar la capacidad de rango de los módulos y monitorear paquetes enviados y recibidos.
- Terminal: permite escribir y leer parámetros en los módulos RF utilizando comandos AT.
- Configuración de modem: escribe y lee los parámetros de los módulos inalámbricos.

En las figuras 7 y 8 se muestra la ventana de la interface de este programa.

Como primer paso a realizar en las pruebas, se toma el módulo RF encargado de ejercer la función de coordinador tras haber restaurado los valores de fábrica con el X-CTU. Mediante la ventana de configuración del modem del software X-CTU, es posible ver los valores de los parámetros principales de configuración dentro del módulo RF. Dichos registros y sus respectivos valores de fábrica se muestran en la figura 7. Como paso siguiente, se conecta el módulo RF al MCU que opera como coordinador y se procede a la leer la misma tabla de parámetros, mediante el software de configuración X-CTU. Los resultados obtenidos, después de permitir al módulo IEEE 802.15.4 interactuar con el MCU coordinador, se muestran en la figura 8. En las figuras 7 y 8 se observa cómo los parámetros SH y SL permanecen inalterados, indicando que se trata del mismo dispositivo inalámbrico. Estos cambios se realizaron automáticamente por el software del módulo coordinador al detectar un transceptor XBee en su puerto serie.

Tras la interacción con el MCU, se observa cómo los cambios en los registros señalados con color rojo, permiten al módulo inalámbrico operar en un modo de direccionamiento de 64 bits (Parámetro MY), se habilita la operación como coordinador (CE) y se escribe un nombre de dispositivo dado por el mismo MCU; el cual es "Coordinador" (NI). De esta manera se prueba que el MCU no solo tiene éxito al configurar su dispositivo, sino al ordenar al módulo XBee guardar los cambios en su memoria no volátil, ya que la implementación de esta prueba requiere desconectar el módulo RF de su sitio habitual para poder leer sus parámetros mediante la tarjeta de desarrollo XBIB-R-DEV REV 4. Dado que ambos MCU emplean los mismos algoritmos y funciones para configurar a sus dispositivos inalámbricos, el MCU *end device* será capaz de establecer comunicación y configurar a su respectivo módulo RF asignado.

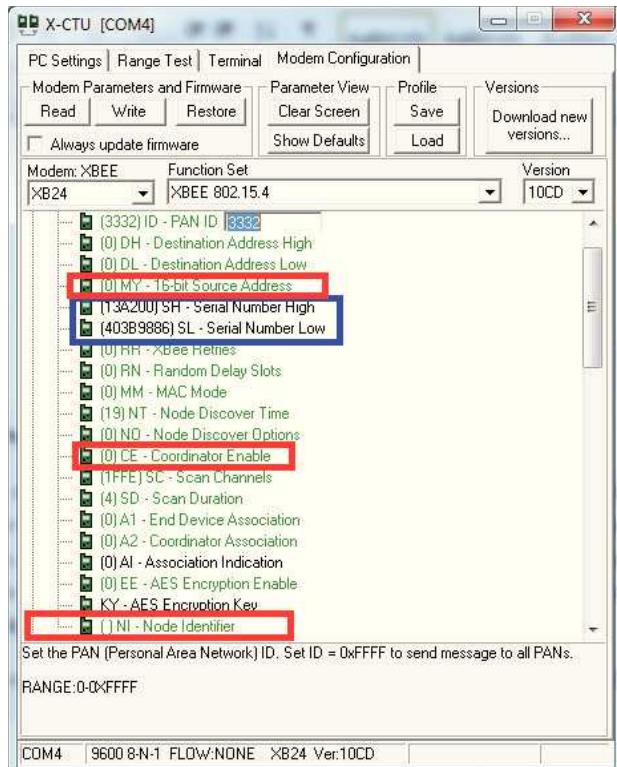


Figura 7. Valores de fábrica para el módulo RF mostrados por X-CTU

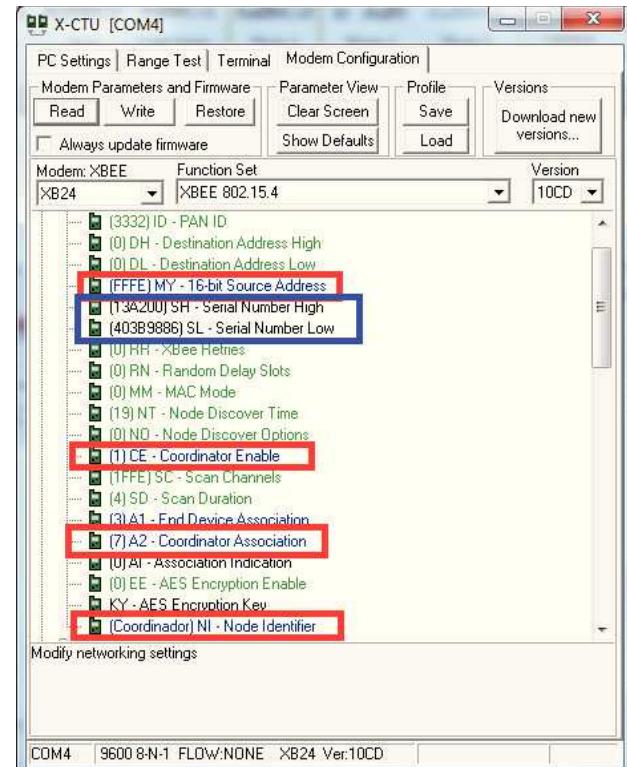


Figura 8. Valores de los registros de configuración de red del módulo RF después de interactuar con el MCU coordinador

El siguiente paso es realizar las pruebas de asociación de los módulos remotos mediante la lectura de parámetros almacenados en el módulo coordinador y en el módulo remoto. Para el caso específico de la asociación de dispositivos, los objetivos de este proyecto plantean que el MCU final sea capaz de localizar y asociarse con los parámetros de red elegidos por el módulo XBee coordinador, de modo que la prueba comienza con la lectura de los parámetros en X-CTU de los dos dispositivos 802.15.4 involucrados en esta red. Cabe señalar, que se decide restaurar el módulo remoto a sus valores de fábrica para apreciar cómo el MCU influye sobre él. Antes de la realización de la prueba, se toman los parámetros del dispositivo RF que se va a conectar con el MCU remoto. En la figura 9 se observa que este módulo está previamente configurado para funcionar como coordinador, lo cual puede significar conflictos en la operación de los dispositivos de no ser reconfigurado correctamente por su MCU asignado.

La figura 10 muestra los parámetros desplegados por el módulo XBee 802.15.4 después de conectarse a su respectivo MCU. En ella se observa cómo todos los

parámetros de red ahora coinciden con los resultados obtenidos en la figura 9. El modo de direccionamiento ha cambiado a 64 bits como era de suponerse y además es posible ver que el parámetro NI ha cambiado su valor a "DEVICE 1", que corresponde al número de dispositivo asignado por el MCU coordinador central. SL y SH no cambian, puesto se trata del mismo dispositivo.

Una vez comprobada la configuración y agregación automática en la red, se decide probar el sistema como un todo. En esta prueba se conecta la PC al módulo coordinador con su transceptor XBee y se coloca un módulo final dentro del alcance del coordinador como se muestra la figura 1. El MCU coordinador puede comunicarse por un dispositivo vía USB mediante comandos de texto, dichos comandos facilitan el desarrollo de una aplicación mediante el uso de la interfaz diseñada en este proyecto. Para poder visualizar, probar y evaluar dichos resultados se utilizó el conocido software *hyperterminal*, el cual permite establecer comunicación con un dispositivo mediante puertos de comunicación (COM).

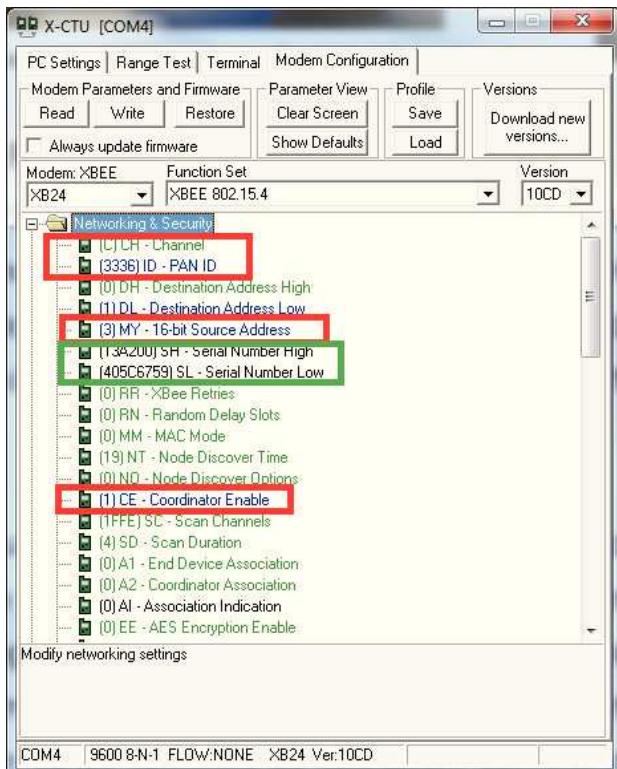


Figura 9. Parámetros de configuración antes de la prueba

Como se describe en la sección de Comandos de texto, se puede enviar tres comandos principales desde la PC al MCU coordinador: leer, escribir y potencia. En la figura 11 se aprecia cómo se usa el comando “escribir” (W) para introducir el número 3 en la primera localidad de la pizarra del dispositivo remoto número uno. Se puede apreciar un mensaje de confirmación enviado por el MCU una vez que este recibe una confirmación exitosa de parte del dispositivo remoto. Como verificación, se envía el comando “lectura” (R) al módulo coordinador para que solicite el contenido de la localidad escrita en el comando anterior como se ve en la línea 3 de la figura 11, se observa en la línea siguiente la respuesta del módulo que muestra que el valor recibido se escribe en el comando anterior.

```
# 1 1 3
Valor Escrito Correctamente <- Respuesta MCU
R 1 1
Dispositivo 1 Indice 1 Valor 3 <- Respuesta MCU
```

Figura 11. Ejemplo de comandos enviados por la PC al MCU coordinador y las respuestas de este después de modificar un registro de un dispositivo remoto. El comando W escribe un valor y posteriormente se lee con el comando R el mismo valor

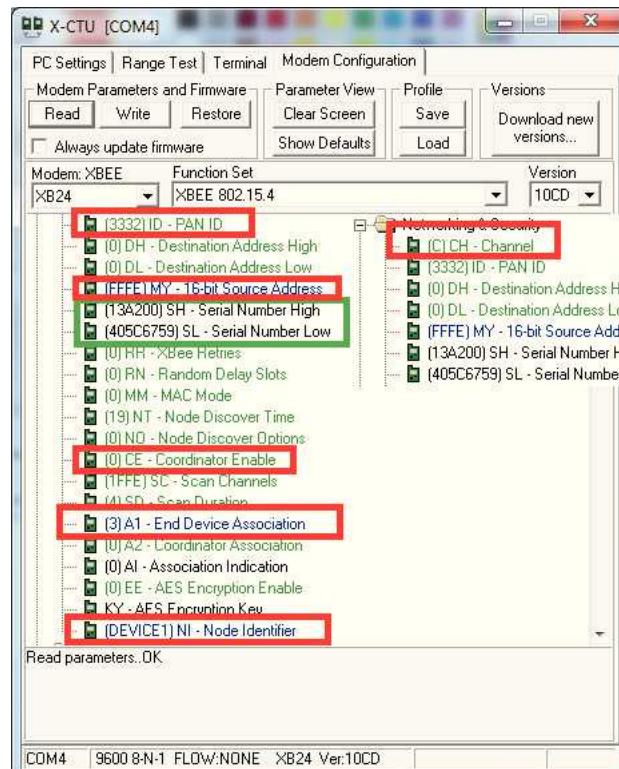


Figura 10. Parámetros leídos por el software X-CTU después de que su microcontrolador lo configura y establece comunicación con su microcontrolador coordinador

Posteriormente se realiza una prueba del comando: “Potencia de transmisión”, usado para conocer la RSS de algún dispositivo remoto solicitado. En este caso, la PC solicita la potencia de transmisión del dispositivo 1 y se tiene como respuesta que el módulo coordinador recibe la señal del dispositivo con una potencia de -66 decibeles (figura 12). La figura 13 muestra la disposición física de los dispositivos al momento de realizarse la prueba.

Con las pruebas anteriores se verificó que el sistema propuesto tiene la capacidad de auto-organizarse como una red de sensores adaptable a cualquier aplicación de monitoreo o control. Esta plataforma se probó en una aplicación de monitoreo de variables para invernaderos tomando un proyecto ya realizado (Olán, 2012) en el que las rutinas de comunicación diseñadas a la medida

```
R 1 1
Dispositivo 1 Indice 1 Valor 3
P 1
Dispositivo 1 Potencia -66 Dbm <- Respuesta MCU
```

Figura 12. Resultado para la prueba de lectura RSS

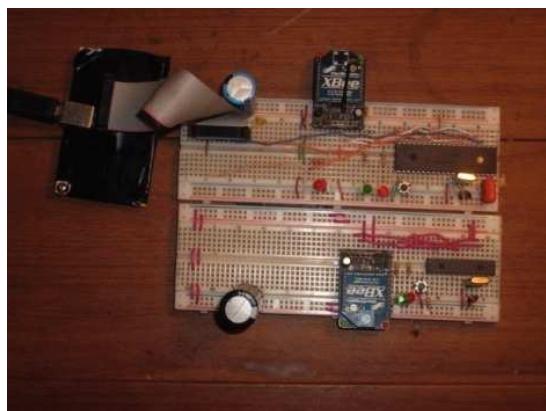


Figura 13. Disposición de los dispositivos en la prueba para la lectura del parámetro RSS

para comunicación con los módulos XBee se sustituyen por rutinas sencillas que solo usan los comandos de texto mostrados en la sección mencionada. Es decir, la aplicación solo se enfoca en las funciones de invernadero y se abstrae de los comandos de red. En los nodos finales se ejecuta una aplicación que simula la lectura

de los parámetros monitoreados y los almacena en la pizarra para que la aplicación en la PC tenga acceso a través del módulo coordinador.

Para esta última prueba, se conectó la PC al módulo de coordinador, el cual se configura automáticamente. Una vez inicializado el coordinador, se acercan los módulos remotos, los cuales se van agregando automáticamente en la lista de nodos finales disponibles para el coordinador. Se agregaron en total 4 módulos remotos, 3 de ellos simulan la lectura de dos parámetros (temperatura y humedad relativa; velocidad del viento y PH; CO₂ y luminosidad); el cuarto módulo simula la variable de radiación PAR. En la figura 14 se observa la interfaz del programa para monitorizar el invernadero y el despliegue gráfico de las variables mostradas. Se configuró la aplicación en la PC para muestrear datos cada 5 minutos y almacenar el historial en un archivo Excel. La aplicación también tiene capacidad para analizar los datos capturados, en la figura 15 se observa la pantalla donde se muestra la tabla de datos capturados y la gráfica del comportamiento de la variable que indica la temperatura.

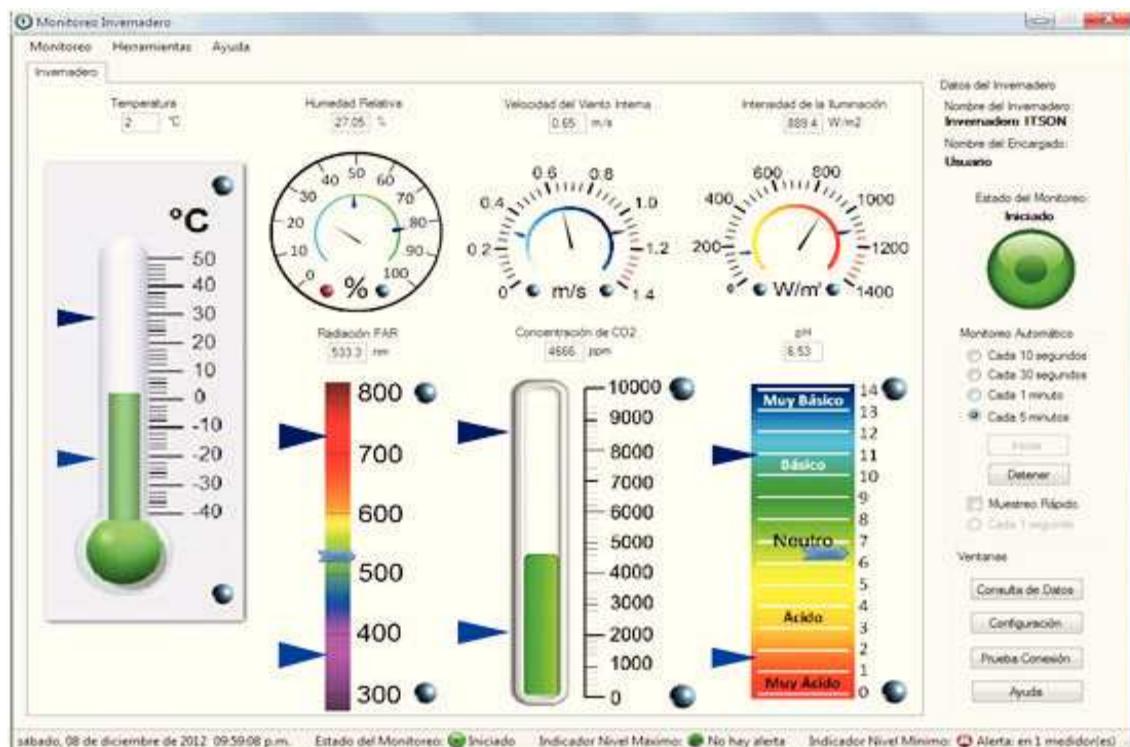


Figura 14. Despliegue gráfico de los valores monitoreados por el control de invernaderos

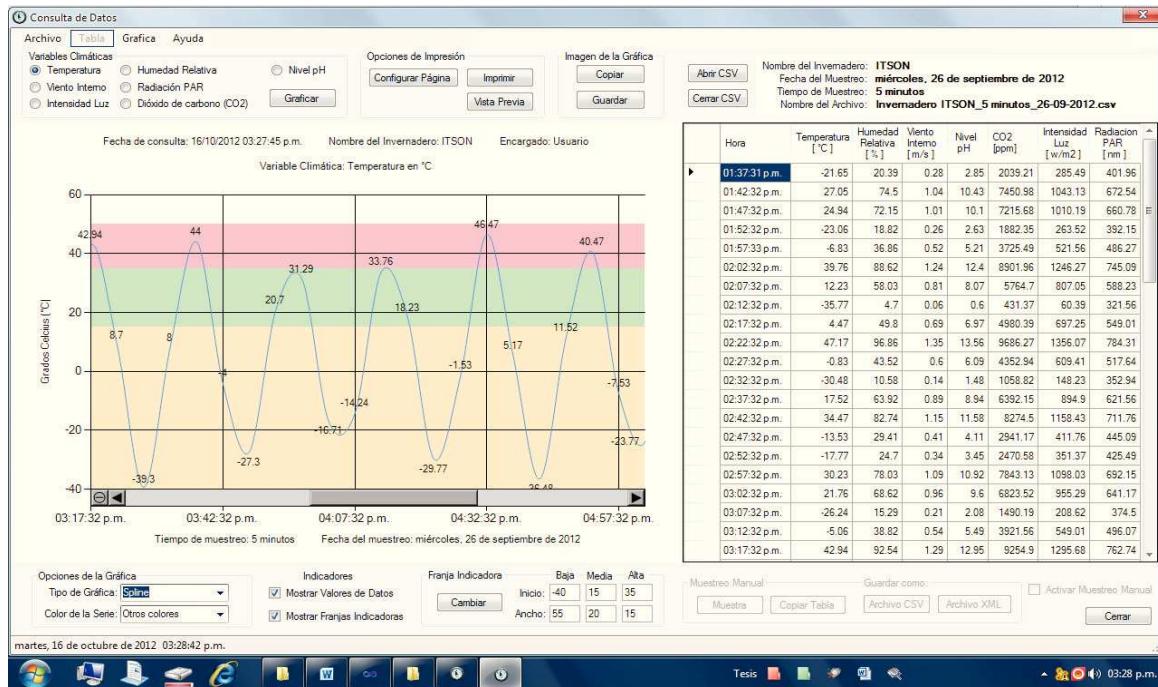


Figura 15. Ventana para visualización de los datos capturados por el control de invernaderos. A la derecha la tabla de los valores de todas los parámetros y a la izquierda el despliegue gráfico alguna de las variables, en este caso temperatura

Conclusiones

El sistema mostrado en este trabajo logra exitosamente cumplir su cometido, que era propiciar una plataforma flexible para la creación de redes inalámbricas de sensores, que puedan utilizar varias aplicaciones con diversos propósitos. La complejidad detrás del desarrollo de dicha interfaz agrega un sin número de retos y posibles mejoras para futuras versiones. Por ejemplo, la gestión de memoria en el MCU principal puede ser más amplia, de modo que permita el ingreso de una cantidad mayor de dispositivos hijos a la red. Asimismo, el sistema de direccionamiento puede implementarse bajo el modo de direccionamiento de 16 bits a fin de obtener direcciones más cortas que puedan ser más fáciles de gestionar por los dispositivos coordinadores. La implementación de dicho sistema de direccionamiento permite que la aplicación sea la que dé nombre a los dispositivos remotos, de manera que sea más fácil que el usuario los identifique. Además, el desarrollo de la plataforma de tramas API de los módulos XBee abre la posibilidad de portar esta plataforma de desarrollo hacia futuras versiones de los módulos provistos por Digi International, ya que la estructura principal de las tramas varía muy poco entre la versión presentada en este artículo y las versiones futuras. El sistema en su versión

temprana permite agregar una cantidad limitada de dispositivos; sin embargo, ofrece una buena opción para la implementación de pequeñas redes de sistemas sensoriales.

Agradecimientos

Este trabajo fue financiado por SEP-CONACYT bajo el proyecto Núm. 183703.

Referencias

- Bougard B., Catthoor F., Daly D.C., Chandrakasan A., Dehaene W. Energy efficiency of the IEEE 802.15.4 standard in dense wireless microsensor networks: modeling and improvement perspectives. Design, Automation and Test in Europe, 2005, Proceedings, I, pp. 196-201.
- Alpi-A. F.T. *Cultivo en invernadero*, Madrid, Mundi-Prensa, 1990.
- ACCI (8 de marzo de 2008). Recuperado en febrero de 2012, Norma Mexicana, NMX-E-255-CNCP-2008 [en línea]. Disponible en: www.amci.org.mx/nmx-e-255-cncp-08.php
- Bastida I.A. Agricultura protegida (ventajas y desventajas en el uso de invernaderos). *TecnoAgro: Avances Tecnológicos y Agrícolas*, edición especial de invernaderos, año 11 (número 69), 2011: 4-9.
- Baudoin W.O. *El cultivo protegido*, Roma, FAO, 2002.

- Bazua A.C. *Programación en C#.NET para automatización*, Cd. Obregón, Sonora, 2010.
- Campaña I.C. Definición técnica y no comercial de invernadero. *TecnoAgro: Avances Tecnologicos y Agrícolas*, edición especial de invernaderos, año 11 (número 69), 2011: 10-13.
- Colombi C. *Invernadero automatizado*, Buenos Aires, Argentina, Instituto Nacional de Educación Tecnológica, 2005.
- García-Breijo E. *Compilador C CCS y simulador PROTEUS para microcontroladores PIC*, Barcelona, Alfaomega grupo editor SA de CV, 2008
- Gobierno del Estado de Sonora 2009. *Plan estatal de desarrollo 2004-2009*. Recuperado en enero de 2012, Gobierno del Estado [en línea]. Disponible en: www.esonora.gob.mx/RAIVD/Dокументos/Plan_Estatal_Dllo_2004-2009.pdf
- Gonzales J.A. 2004. Recuperado en enero de 2012, El lenguaje de programación C#, libros ingenegros.org.
- Gutiérrez J.A. *Low-rate wireless personal area networks, Enabling Wireless Sensors With IEEE 802.15.4.*, IEEE Standards Association, 2004.
- Joyanes-Aguilar L. *C#. Manual de programación*, España, McGraw-Hill, 2002.
- Joyanes-Aguilar L. *Programación en C++, algoritmos, estructuras de datos y objetos*, Madrid, McGraw Hill, 2006.
- López D.P. Estructuras utilizadas en la agricultura protegida. *Revista Fuente*, año 3 (número 8), septiembre 2011.
- Lopez J.C. *Incorporación de tecnología al invernadero mediterráneo*, El ejido, Almeria, Escobar Impresiones, 2001.
- Lovello J.M. *Lenguaje para modelar Objetos*, Oviedo, España, Departamento de Informatica, 1999.
- MaxStream. *XBee/XBee-PRO OEM RF Modules*. M100232, 2007.
- Ramjee P. y Deneire L. *From WPANs to personal networks, technologies and applications*, Artech House, 2006.
- Rivera M.R. *La tecnología del invernadero en el Valle del Yaqui*, Cd. Obregón, Despacho Inbernal, 2007
- Serna A. Control climático en invernadero. *TecnoAgro: Avances Tecnologicos y Agrícolas*, edición especial de invernaderos, año 11 (número 69), 2011: 20-22.
- Torres A.P. Manejo de la alcalinidad en sustratos hidropónicos. *Extensión de Perdue*, 2010: 1-5.

Este artículo se cita:

Citación estilo Chicago

Espinoza-Ruiz, Adolfo, Erica Ruiz-Ibarra, Armando García-Berúmen, Juan Antonio López-Chaides, Joaquín Cortez-González, Joel Ruiz-Ibarra. Plataforma autoconfigurable de monitoreo remoto para aplicaciones móviles en WSAN. *Ingeniería Investigación y Tecnología*, XVI, 03 (2015): 369-382.

Citación estilo ISO 690

Espinoza-Ruiz A., Ruiz-Ibarra E., García-Berumen A., López-Chaides J.A., Cortez-González J., Ruiz-Ibarra J. Plataforma autoconfigurable de monitoreo remoto para aplicaciones móviles en WSAN. *Ingeniería Investigación y Tecnología*, volumen XVI (número 3), julio-septiembre 2015: 369-382.

Semblanzas de los autores

Adolfo Espinoza-Ruiz. Recibió el grado de maestro en ciencias computacionales por ITESM, Campus Estado de México en 2003. Actualmente es profesor del Departamento de Ingeniería Eléctrica y Electrónica en el Instituto Tecnológico de Sonora. Dentro de las líneas de investigación se encuentran los sistemas empotrados, *soft computing* y redes inalámbricas de sensores y actores.

Erica Ruiz-Ibarra. Obtuvo el grado de M. en C. y el doctorado en ciencias, ambos en electrónica y telecomunicaciones por el Centro de Investigación Científica y Educación Superior de Ensenada, CICESE México, durante el 2002 y 2010, respectivamente. Actualmente es profesora de tiempo completo en el Instituto Tecnológico de Sonora. Sus áreas de interés incluyen redes inalámbricas de sensores y actores, comunicaciones digitales, sistemas MIMO y cómputo ubicuo.

Armando Garcia-Berumen. Recibió el título de ingeniero en electrónica en 1995 por el Instituto Tecnológico de Durango. En 1998, obtuvo el grado de maestría con acentuación en telecomunicaciones por el Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM). Es doctor por Telecom SudParis (antes Instituto Nacional de Telecomunicaciones) en Evry Francia. Actualmente se desempeña como profesor de tiempo completo en el Departamento de Ingeniería Eléctrica y Electrónica del Instituto Tecnológico de Sonora (ITSON). Cuenta con diversas publicaciones en congresos nacionales e internacionales. Sus áreas de interés son el modelado y evaluación de desempeño de redes.

Juan Antonio López-Chaidez. Ingeniero en electrónica por el Instituto Tecnológico de Sonora (ITSON) en 2012. Actualmente cursa la maestría en ciencias en electrónica y telecomunicaciones en CINVESTAV-Unidad Guadalajara. Sus áreas de interés son sistemas empotrados, redes inalámbricas de sensores y microcontroladores.

Joaquín Cortez González. Recibió el grado de M. en C. en ingeniería eléctrica por el CINVESTAV Unidad Guadalajara, en 2001, asimismo el grado de doctor en ciencias en ingeniería eléctrica por CINVESTAV-Guadalajara, en 2008. Actualmente es profesor en el Departamento de Ingeniería Eléctrica y Electrónica del Instituto Tecnológico de Sonora. Sus áreas de interés incluyen comunicaciones digitales, procesamiento digital de señales, redes inalámbricas, *soft computing* e interacción hombre-máquina.

Joel Ruiz-Ibarra. Ingeniero en electrónica por el Instituto Tecnológico de Sonora (ITSON). M. en C. en electrónica y telecomunicaciones por el Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE). Es doctor en ciencias por el CICESE (2011). Actualmente es profesor de tiempo completo en ITSON. Sus líneas de investigación son el diseño de protocolos de coordinación, acceso al medio, enrutamiento y localización para redes inalámbricas de sensores.