

Propuesta de un sistema de seguridad residencial inalámbrico basado en instrumentación virtual

Gereli de los Ángeles Gutiérrez Finol

Ing. Electricista Premio Cum Laude (Universidad Rafael Urdaneta).
Cursos en la Maestría de Computación Aplicada (Universidad del Zulia).
Componente Docente en Investigación. Ingeniero de Soporte (Fundación Petrociencia).
Área de Investigación: Instrumentación Virtual. *gereligf@gmail.com*

Recibido: 13-09-2013 Aceptado: 02-05-2014

Resumen

El objetivo general consistió en diseñar un sistema de seguridad residencial inalámbrico basado en instrumentación virtual. El trabajo se encuentra sustentado en la teoría de operación de los módulos de radiofrecuencia Xbee actuando en una red inalámbrica de topología tipo árbol, cuya simulación se basó en el algoritmo de Dijkstra. En el diseño se aprovecharon las funciones de modelado, control y simulación de Matlab y Prolog, el lenguaje de programación C y el simulador ISIS Proteus. Fue comprobada la integración del sistema mediante pruebas experimentales de reconocimiento de la red, pruebas de enrutamiento de las señales e interactividad con el usuario a través de internet, obteniéndose apropiado desempeño según la lógica operacional previamente diseñada, permitiendo concluir que el sistema propuesto es aplicable en ambientes asociados a viviendas unifamiliares.

Palabras Claves: Sistema de seguridad residencial, instrumentación virtual, red inalámbrica, dijkstra.

Proposal for a wireless home security system based on virtual instrumentation

Abstract

The main objective was to design a wireless home security system based on virtual instrumentation. The design is based on the Xbee RF modules's theory in a wireless network type topology tree, which was simulated using a variation of Dijkstra's algorithm. The design took advantage of modeling capabilities, control and simulation programming tools: Matlab and Prolog, the programming language C and ISIS Proteus simulator. Integration was verified by experimental test system for recognition of the network routing test signals and user's interactivity through the internet, was obtained appropriate operational performance according to the logic previously designed, allowing concluding the proposed system is applicable environments associated with houses.

Key words: Home security system, virtual instrumentation, wireless network, dijkstra.

Introducción

Actualmente la instalación de sistemas de seguridad en viviendas ha traído como consecuencia que los fabricantes de estos productos vean en el mercado de seguridad residencial un potencial cada vez mayor, por esta razón es necesario reducir los costos de inversión en su desarrollo, lo cual puede darse mediante la instrumentación virtual.

La instrumentación virtual es un sistema de medición, análisis y control de señales físicas utilizando un computador personal por medio de instrumentos virtuales^[1]. En el mercado se encuentran varios tipos de sistemas de seguridad cuyas funciones principales son las de controlar los accesos a la vivienda y monitorizar los dispositivos de detección de personas no autorizadas. En este sentido, se pretende combinar estas funciones y proponer como alternativa el diseño de un sistema de seguridad residencial mediante la identificación de los elementos que lo componen, la precisión de los criterios de diseño y la lógica operacional del mismo, el diseño de sus componentes y, por último, la propuesta de su integración.

Fundamentos teóricos

Para el diseño de este sistema se ha elegido el tipo de arquitectura de configuración centralizada, donde un controlador gestiona todas las funciones y acciones del sistema. A su vez, la comunicación inalámbrica establecida entre los elementos se encuentra bajo la especificación Zigbee^[2]. Existen tres tipos de dispositivos, también denominados nodos o módulos de radiofrecuencia, que construyen la red inalámbrica: coordinador, enrutador y nodo final. La organización de estos dispositivos se denomina topología. Zigbee permite tres topologías de red: estrella, árbol y malla. En el mercado existe una variedad de módulos de radiofrecuencia, los módulos Xbee son los primeros productos que se basan en una plataforma compatible con Zigbee^[3], por lo tanto, la prueba de integración del sistema de seguridad se apoyará en algunas de sus características de funcionamiento. En los sistemas de redes inalámbricas reales, los módulos de radiofrecuencia se encuentran conectados a los dispositivos de seguridad de manera que el control y la monitorización son admitidos gracias a las señales que se envían a los nodos, y éstos las envían a los dispositivos. En este caso se trabajará con dos tipos de dispositivos utilizados comúnmente en sistemas de seguridad: sensores y actuadores. Además, fue necesario contar con un algoritmo de enrutamiento que permitiera transmitir la información hasta su destino con el mínimo número de saltos, para lo cual se trabajó con una variación del algoritmo de Dijkstra^[4].

Las herramientas de programación utilizadas fueron: ISIS Proteus^[5], Matlab^[6], SWI Prolog^[7], HTML^[8] (lenguaje de marcado de hipertexto) y X-CTU^[9], este último se utilizó como interfaz de configuración y prueba para los módulos Xbee.

Parte Experimental

La investigación abarcó dos fases: en la primera se llevó a cabo el montaje virtual, donde se procedió a diseñar el Emulador de red inalámbrica en el software ISIS Proteus mediante la programación en C del PIC16F877, el Controlador y el Panel de Control local se elaboraron utilizando Matlab en conjunto con el recurso Guide, para el Panel de Control Remoto (Página Web) se utilizaron los lenguajes HTML y JavaScript y para el diseño del Servidor web se utilizó SWI Prolog. En la última fase se realizaron pruebas que permitieron reconocer la red inalámbrica emulada, conectar los módulos del sistema, activar/desactivar dispositivos y comprobar la interactividad con el usuario.

Resultados y discusión

Elementos del sistema de seguridad residencial inalámbrico

Los elementos que conforman el sistema son: el emulador de red inalámbrica, el cual imita una red real bajo el estándar Zigbee, el controlador que actúa sobre el sistema enviando órdenes a la red mediante un paquete de simulación de conexión serial virtual: VSPE (Virtual Serial Port Emulator) y el servidor web que genera una página web para el uso del sistema de seguridad desde cualquier equipo con acceso a internet.

Criterios de diseño del sistema de seguridad residencial inalámbrico

Para el diseño del emulador de la red, se ubicó en ISIS Proteus el circuito electrónico del PIC16F877 como elemento principal^[10]. La activación/desactivación de los actuadores y sensores se visualizó mediante el encendido y apagado de luces LED y la activación de los sensores se simuló utilizando un teclado numérico de 12 teclas. Posteriormente se procedió a diseñar el Controlador del sistema de manera que permitiera al usuario configurar la red mediante el uso del Panel de Control Local. Siguiendo el criterio de operatividad de manera remota, se diseñó el Servidor web para que el sistema pudiera ser controlado y monitorizado vía web.

Lógica operacional del sistema de seguridad residencial inalámbrico

El Emulador de la red inalámbrica está constituido por una aplicación en lenguaje C y compilado en código hexadecimal, el cual es llamado por ISIS Proteus durante la operación del PIC 16F877.

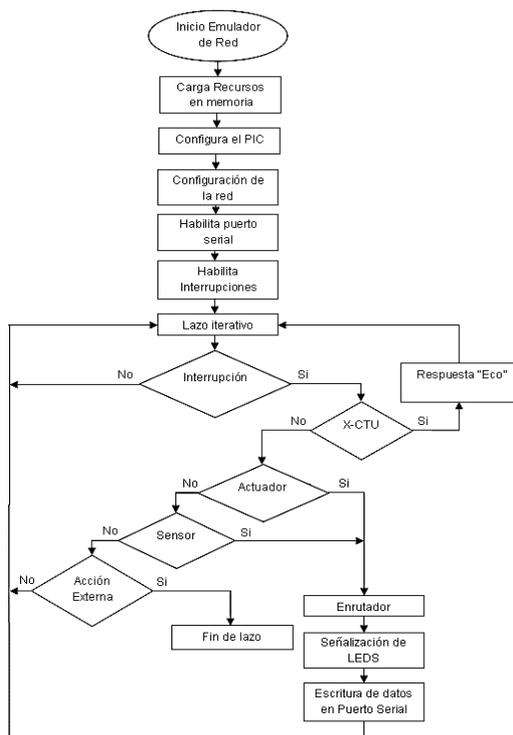


Figura 1. Diagrama de flujo del Emulador de red inalámbrica

Atendiendo la figura 1, se inicia el Emulador de red, se cargan las bibliotecas correspondientes al PIC y las funciones estándar de entrada y salida de datos, la configuración del PIC se basa en las siguien-

tes características: las bibliotecas para el manejo de cadenas de caracteres, la pantalla LCD, un teclado telefónico de tipo numérico y la conexión vía puerto serial con el Controlador utilizando el estándar RS-232. Además, posee internamente la configuración de la red, la cual fue compilada en un archivo hexadecimal idéntica a la guardada en el archivo *dist.txt*, en donde están distribuidos los nodos según su ubicación y la interrelación entre ellos. Al habilitarse el puerto serial, se permiten las interrupciones y el PIC recibe comandos del Controlador en forma de tramas. La estructura de datos del archivo *dist.txt* se muestra en la figura 2: comienza en la primera línea con el número de conexiones registradas entre los nodos (n), seguido de una matriz entera de $3 \times n$.

Nodo Enrutador	Nodo final	Distancia entre nodo enrutador y final (metros)
----------------	------------	---

Figura 2. Archivo de distribución (*dist.txt*)

El archivo de *configuracion.txt* de la figura 3, es grabado por el usuario utilizando el Panel Local del Controlador y presenta en cada línea del texto el identificador de la variable Prolog seguida de un conjunto de argumentos.

Red (número de nodo, 0, tipo de nodo, 'código', 'lugar').

Figura 3. Archivo de configuración (*configuración.txt*)

Luego de la configuración de la red virtual dentro del PIC se habilitan las interrupciones globales y locales que manejan el diálogo entre el microcontrolador y los dispositivos periféricos. En este momento comienza el lazo iterativo: si una interrupción proviene del programa X-CTU, el PIC devuelve un "eco" al puerto serial indicándole al programa X-CTU que la red existe. Una función de enrutamiento determina el camino más corto cuando se requiere una comunicación entre el nodo coordinador y el nodo destino. El estado de activación de un sensor o actuador es enviado al puerto serial para que sea leído por el Controlador, repitiéndose el lazo iterativo. Similarmente, si la interrupción es interna o proviene del teclado del ISIS Proteus se ejecuta la misma operación anterior. Se repite el lazo iterativo de forma indefinida hasta que una acción externa finalice el lazo inmediatamente.

Por otra parte, cuando el Controlador es activado desde el Panel Remoto, ver figura 4, Matlab carga los recursos del sistema consistentes en las bibliotecas de enlace dinámico. Se abre la interfaz del administrador y escribe "ok" en el buffer de comandos, indicando que el Controlador y su interfaz gráfica están en operación. Se lee el *buffer* de estados (archivo *estado.txt*), y escribe un "0" (desactivado) o un "1" (activado).

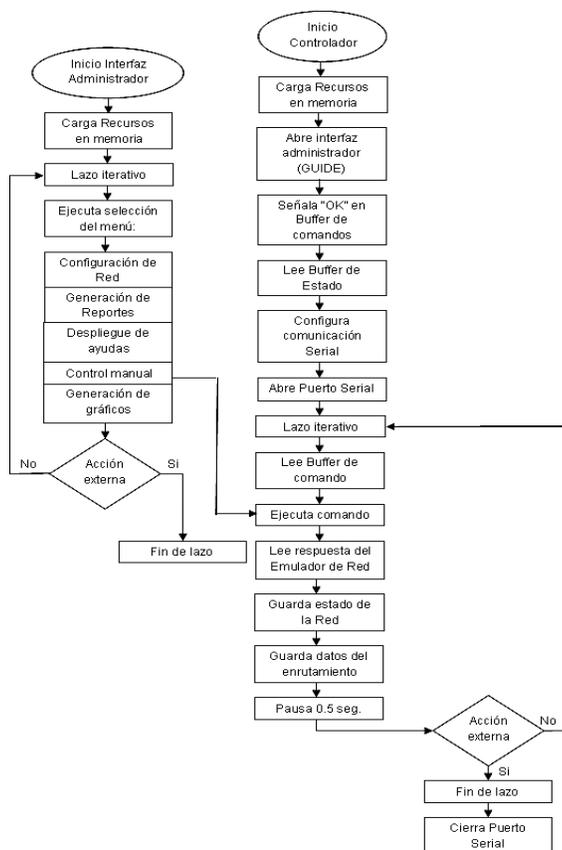


Figura 4. Diagrama de flujo del controlador

El Controlador es capaz de leer el archivo de comandos *buffer.txt*, esta información puede provenir del Panel Remoto o Local. Se ejecuta el comando enviando la información por el puerto serial hacia el Emulador de red a través del VSPE. Guarda el estado de la red y los datos de enrutamiento. Existe una pausa de 0.5 segundos que controla el tiempo en el que se ejecuta el lazo. La interfaz gráfica del Controlador, aparece en el momento en que se ejecuta el Controlador mediante el archivo *panel2.m*.

Ahora bien, atendiendo la figura 5, una vez que el usuario activa el Servidor web en la máquina local, el Servidor procede a cargar en la memoria RAM del computador los recursos del programa consistentes en la biblioteca de comandos gráficos para Windows 7, manejo y construcción de páginas HTML, reconocimiento de códigos de hipertexto y protocolos de internet, importación de recursos gráficos necesarios para la construcción del monitor del Panel Remoto. Posteriormente se habilita el puerto 8080, luego el Servidor realiza una limpieza de registros temporales y lee el archivo *estado.txt*. Para construir la respuesta de la interfaz del Panel Remoto, el Servidor lee la hora y fecha del sistema operativo, señala en el monitor de respuesta el estado de los nodos mediante: luz amarilla para la activación de los nodos actuadores, luz roja para los sensores y luz verde para los enrutadores (puede ser sensor o actuador).

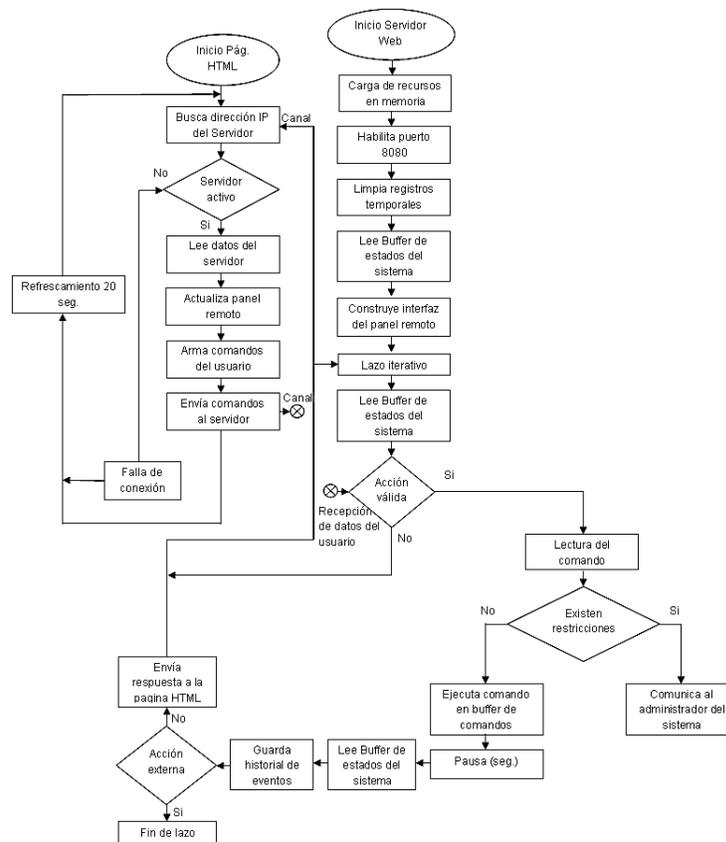


Figura 5. Diagrama de flujo del servidor web

Comienza un lazo iterativo en el cual se vuelve a leer el *buffer* de estados del sistema, se recibe el comando del Panel Remoto y se compara con las restricciones de la lógica de operación como lo son: la existencia del nodo, un cambio de estado de los nodos, otra operación en el proceso, o espera por la respuesta del Controlador. Si las restricciones están presentes se le comunica al usuario. En caso contrario, una vez que el comando es aceptado se escribe en el *buffer* de comandos, el cual es leído por el Controlador, se procede a una pausa de 5 segundos en espera de la respuesta ubicada en el *buffer* de estados del sistema. En caso de no existir acciones externas, la respuesta del Controlador se envía al Panel Remoto y se continúa con el lazo iterativo de forma indefinida hasta recibir interrupciones del usuario o de la máquina local.

Diseño del emulador de la red inalámbrica

En la figura 6 se muestra el emulador y sus componentes: el PIC16F877, el monitor de eventos del puerto serial, una pantalla LCD LM016L de 2x16 para la visualización de los comandos que provienen del puerto serial y la ruta calculada por algoritmo de enrutamiento del PIC, 16 luces LEDs, un conector DB9 y el teclado.

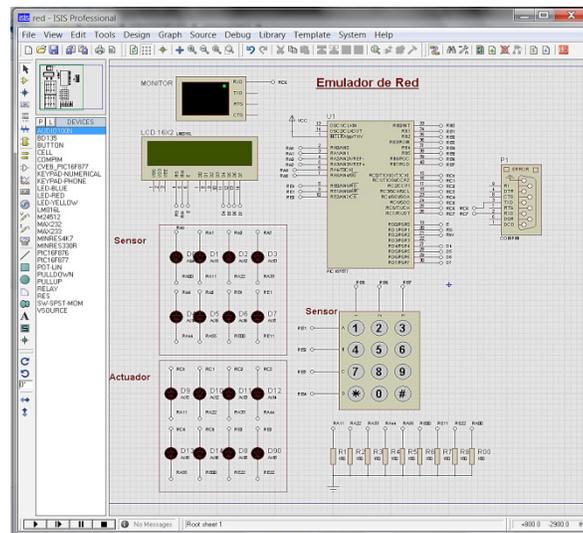


Figura 6. Emulador de la red inalámbrica (Panel ISIS Proteus)

Diseño del panel de control local

La interfaz gráfica es capaz de recibir las instrucciones a través de botones y modificar archivos de texto, ver figura 7. Luego de ingresar la clave de acceso, mediante los botones Activar, Desactivar y Enviar, junto con el teclado numérico, se construyen las órdenes que el usuario envía a los dispositivos de actuación. El teclado posee una ventana que muestra la tecla pulsada, mientras que los estados de los nodos de la red son observados en el monitor ubicado a la derecha del Panel mediante indicadores de forma circular. Se pueden verificar los archivos de estado, configuración de la red e historial de eventos, mediante botones ubicados en la parte inferior del Panel. Además, se puede realizar una prueba de enrutamiento que corrobora el estado de los nodos de la red mostrados en el monitor. Además, existe un botón para la consulta de la dirección IP de la máquina donde se encuentra el Controlador.



Figura 7. Panel de control local

Diseño del servidor web

La página web de la figura 8, comienza a operar cuando el usuario coloca la dirección IP del Servidor, y se actualiza cada 20 segundos. Permite la inclusión de la clave de acceso, el manejo de un menú

desplegable y herramientas adicionales, además posee un campo para la entrada de comandos en forma de cadena de caracteres.

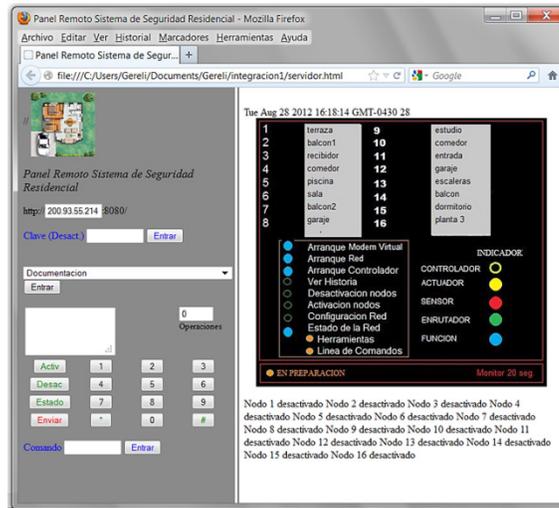


Figura 8. Panel de control remoto: página web

Integración del sistema de seguridad residencial inalámbrico

Comunicación emulador de la red inalámbrica - panel de control local: en la figura 9 se muestra cómo la comunicación entre el Emulador de red y el Controlador que se realiza a través de puerto serial, puede ser verificada utilizando el VSPE. Las tramas de datos observadas pueden ser: activación y desactivación de los actuadores y activación de los sensores; ambas consistentes en 13 caracteres: el primer carácter representa el comando (a = activación, d = desactivación y s = sensor); el segundo, un espacio en blanco; el tercero el número del nodo final (un valor máximo de 8 para los sensores y 8 para los actuadores), los diez caracteres restantes, permiten la posibilidad de incluir la identificación de hasta diez nodos en la ruta de activación del nodo final.

Comunicación controlador - servidor web: se utilizó un archivo de texto como *buffer* entre el Controlador y el Servidor web, ver figura 9, en donde se guardan de manera asíncrona la trama en modo de texto los comandos provenientes del usuario y aquellas tramas generadas por la lógica del Controlador. Desde el Servidor, la estructura de la trama es igual a la que se describió anteriormente.

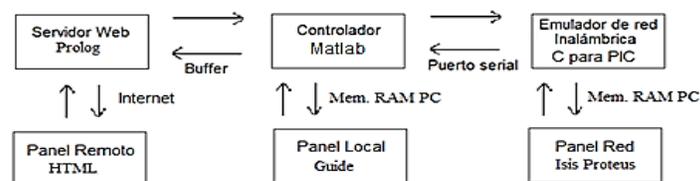


Figura 9. Sistema de Seguridad Residencial Inalámbrico

Una palabra clave es referida para detener el Controlador (*fin*), y otra, para indicar que el Servidor se encuentra listo para recibir respuesta del Controlador (*oks*).

Reconocimiento de la red inalámbrica: se realizó la conexión virtual entre el programa X-CTU y el Emulador de red utilizando los puertos de comunicación 2 y 6 configurados por el VSPE. El X-CTU envía una trama de prueba, la cual debe ser recibida en forma de “eco” desde la red ya configurada indi-

cando la existencia de una red aunque no se encuentre físicamente conectada al computador, ver figura 10.

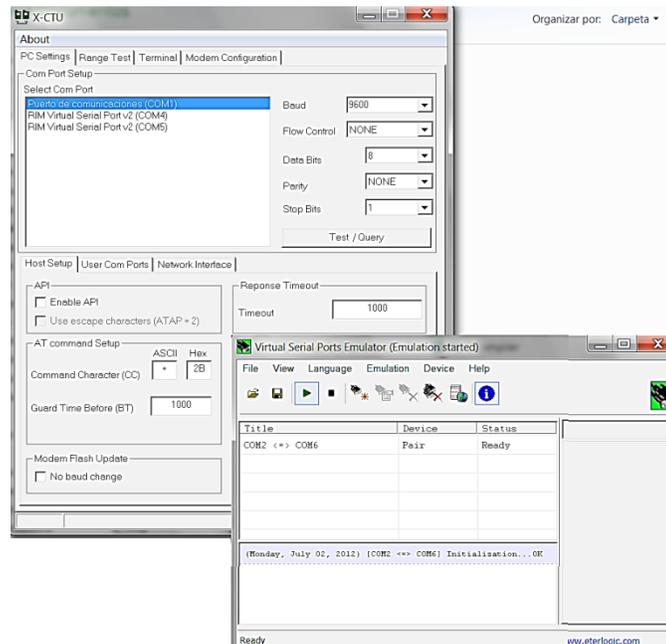


Figura 10. Reconocimiento de la red por parte del X-CTU

Ejecución de pruebas de enrutamiento: la prueba de configuración de la red virtual, utilizando la interfaz del Controlador, se basó en un plano de planta de una vivienda de tres pisos, en la cual se indicó la ubicación de los sensores, los actuadores, el coordinador de la red y las distancias entre cada uno de los nodos. Utilizando geometría básica, la red se aproximó a una red plana, ver figura 11, partiendo de lo siguiente: deben existir los tres tipos de nodos, la topología es tipo árbol, cada nodo de la red puede contar con solo un nodo ascendiente, cada nodo enrutador puede tener varios nodos descendientes y el número total de nodos es de 16. Del caso propuesto se plantearon tres configuraciones de red, ver figura 12, se trabajó con *dist.txt*, ver figura 13. En la tabla 1 se muestra la comprobación de ruta de los 16 nodos de las tres redes: en la primera columna se muestra el número del nodo activado, en la segunda, la tercera y la cuarta columna se presentan las rutas adoptadas, y en las últimas tres columnas, las distancias resultantes al activarse cada nodo. Según la tabla 1, la configuración Red 1 permite que la información que llega al nodo coordinador transite más rápido que en las otras dos. Se verificó el desempeño del algoritmo de Dijkstra de la red seleccionada con respecto al algoritmo de enrutamiento propuesto, en la tabla 2 se comprobó que no existían diferencias entre ambos.

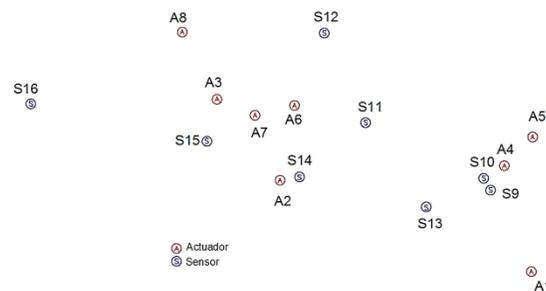


Figura 11. Distribución de nodos sensores y actuadores

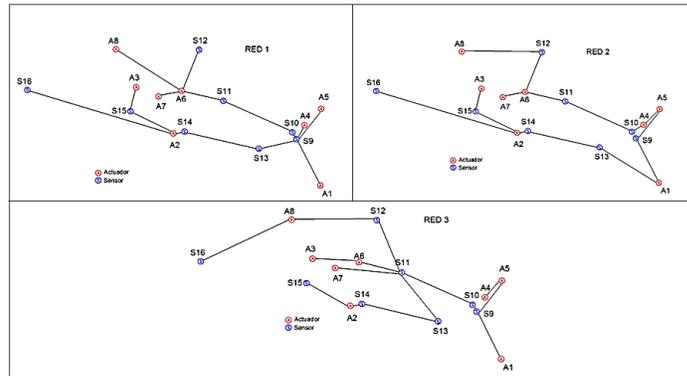


Figura 12. Red 1, Red 2 y Red 3

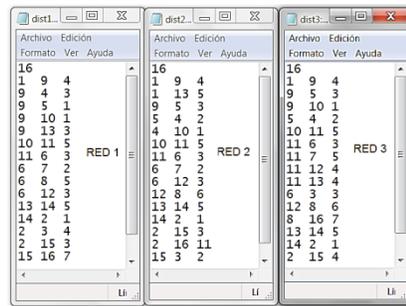


Figura 13. Archivos dist.txt: Red 1, Red 2 y Red 3

Tabla 1. Rutas de nodos de tres configuraciones de redes diferentes

Nodo	Ruta			Distancia (m)		
	Red 1	Red 2	Red 3	Red 1	Red 2	Red 3
1	1	1	1	0	0	0
2	1 9 13 14 2	1 13 14 2	1 9 10 11 13 14 2	13	11	20
3	1 9 13 14 2 3	1 13 14 2 15 3	1 9 10 11 6 3	17	16	16
4	1 9 4	1 9 5 4	1 9 5 4	7	9	9
5	1 9 5	1 9 5	1 9 5	5	7	7
6	1 9 10 11 6	1 9 5 4 10 11 6	1 9 10 11 6	13	18	13
7	1 9 10 11 6 7	1 9 5 4 10 11 6 7	1 9 10 11 7	15	20	15
8	1 9 10 11 6 8	1 9 5 4 10 11 6 12 8	1 9 10 11 12 8	18	27	20
9	1 9	1 9	1 9	4	4	4
10	1 9 10	1 9 5 4 10	1 9 10	5	10	5
11	1 9 10 11	1 9 5 4 10 11	1 9 10 11	10	15	10
12	1 9 10 11 6 12	1 9 5 4 10 11 6 12	1 9 10 11 12	16	21	14
13	1 9 13	1 13	1 9 10 11 13	7	5	14
14	1 9 13 14	1 13 14	1 9 10 11 13 14	12	10	19
15	1 9 13 14 2 15	1 13 14 2 15	1 9 10 11 13 14 2 15	16	14	24
16	1 9 13 14 2 15 16	1 13 14 2 16	1 9 10 11 12 8 16	23	22	27
Total				181	209	217
Promedio				11,31	13,06	13,56

Tabla 2. Comparación algoritmo de Dijkstra vs Algoritmo propuesto

Nodo	Ruta del algoritmo		Diferencias
	Dijkstra	Propuesto	
1	1	1	0
2	1 9 13 14 2	1 9 13 14 2	0
3	1 9 13 14 2 3	1 9 13 14 2 3	0
4	1 9 4	1 9 4	0
5	1 9 5	1 9 5	0
6	1 9 10 11 6	1 9 10 11 6	0
7	1 9 10 11 6 7	1 9 10 11 6 7	0
8	1 9 10 11 6 8	1 9 10 11 6 8	0
9	1 9	1 9	0
10	1 9 10	1 9 10	0
11	1 9 10 11	1 9 10 11	0
12	1 9 10 11 6 12	1 9 10 11 6 12	0
13	1 9 13	1 9 13	0
14	1 9 13 14	1 9 13 14	0
15	1 9 13 14 2 15	1 9 13 14 2 15	0
16	1 9 13 14 2 15 16	1 9 13 14 2 15 16	0

Pruebas de funcionamiento

Configuración de la red

Seleccionada la red, se procede a configurarla dependiendo de la ubicación de los sensores y actuadores en la vivienda. El usuario identifica los ambientes en los cuales se encuentra cada nodo y los escribe en el archivo *configuracion.txt*, ver figura 15, se introduce la Red 1 en el PIC y se compila el programa.

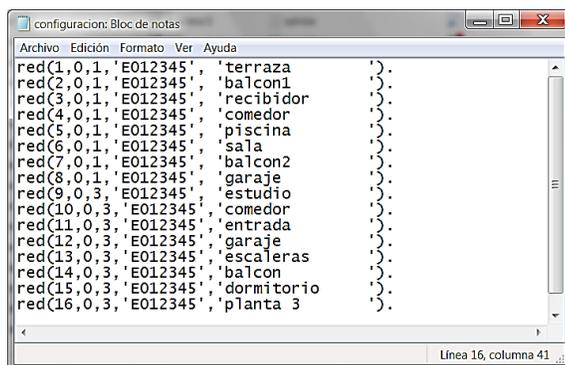


Figura 15. Archivo configuracion.txt Red 1

Secuencia de arranque del sistema

La secuencia de arranque puede ser de forma local o remota, en ambos casos se debe seguir la siguiente secuencia: 1° Arranque VSPE, 2° Arranque del Emulador, 3° Inicio de la simulación del Emulador, 4° Verificación del último estado de la red y 5° Envío y recepción de comandos del usuario.

Caso 1: Activación de sensor: activado el nodo sensor S11, ubicado en la entrada de la residencia, la ruta se comprueba mediante la tabla 2. El estado del sistema se muestra en las figuras 16 y 17.

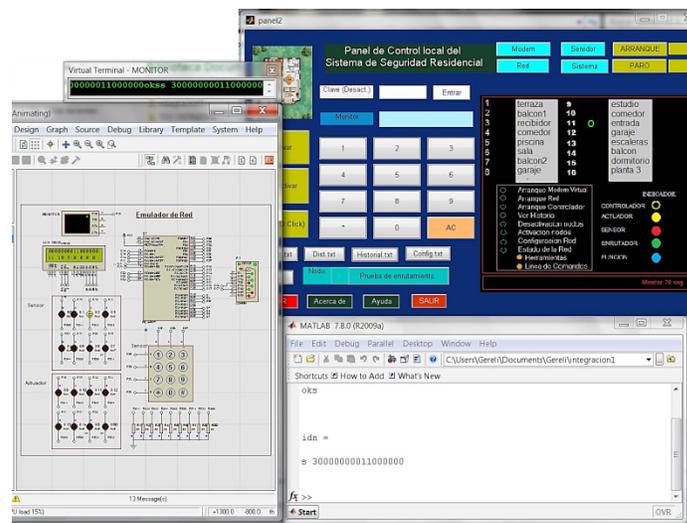


Figura 16. Respuesta local del Sistema ante activación de S11

Caso 2: Activación de actuador: se comprueba el flujo de información desde internet hasta la red, activándose el nodo A3 vía web, el resultado se muestra en la figura 17.

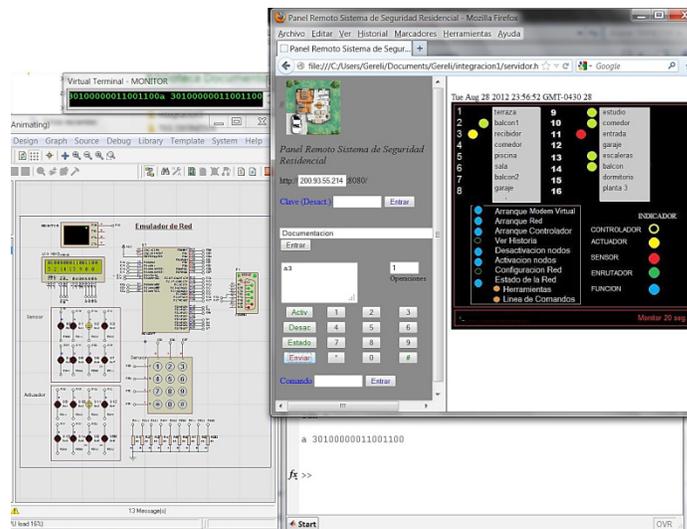


Figura 17. Activación remota del actuador A3 y respuesta del Sistema

En este caso se comprobó que los comandos que envía el usuario vía remota, son recibidos y asimilados por la red emulada en cualquier instante de tiempo, verificando de esta manera el carácter bidireccional y asíncrono que posee el control del sistema de seguridad residencial inalámbrico propuesto.

Conclusiones

Los módulos del sistema de seguridad residencial inalámbrico propuesto se interrelacionan entre sí y poseen características propias de la aplicación utilizada para su simulación. Los criterios de diseño y la lógica operacional permitieron especificar los dispositivos utilizados en la residencia, así como su ubicación y cantidad, dependiendo de las exigencias del usuario al momento de instalarlos y atendiendo a

las limitaciones del PIC16F877. El Emulador de red fue capaz de imitar una red real y trabajar con varias configuraciones de igual topología bajo el algoritmo propuesto, el cual permitió utilizar una mayor parte de la memoria RAM del PIC que no hubiese sido posible empleando el algoritmo de Dijkstra. El Panel de Control Local permitió visualizar las acciones emitidas localmente o señales externas provenientes de la red. El Controlador presentó la habilidad de interpretar las instrucciones recibidas del Emulador de red y enviarlas al Panel de Control Local y Remoto, así como también fue capaz de distribuir las órdenes provenientes del usuario y enviarlas como señales de control al Emulador de la red. El Servidor web estuvo calificado para reconocer las acciones de control provenientes del Panel de Control Remoto, de manera que fuese posible la ejecución de órdenes de activación y desactivación de los elementos del sistema vía web. La ejecución de pruebas de reconocimiento de la red, experimentos de enrutamiento y pruebas de funcionamiento global del sistema demostraron la apropiada integración del mismo, pues los resultados obtenidos evidenciaron su correcta operatividad cumpliendo tanto los criterios como la lógica operacional previamente establecida.

Referencias Bibliográficas

1. Antoni M. Instrumentación Virtual: adquisición, procesado y análisis de señales. Universidad Politécnica de Cataluña. Vol. 102, (2001) 15-30.
2. Farahani S. Zigbee Wireless Networks and Transceivers. USA (2008).
3. MaxStream. Xbee™/Xbee - Pro™ OEM RF Modules. Product Manual v1.xAx -802.15.4 Protocol (2007).
4. Joyanes L. Algoritmos y Estructura de Datos. Ed. McGrawHill. Madrid (2004).
5. García E. Compilador C CCS Y Simulador Proteus para Microcontroladores PIC. Ed. Alfaomega (2008).
6. Barragan D. Manual de interfaz gráfica de usuario en Matlab. (2008).
7. Wielemaker J. SWI-Prolog 5.6 Reference Manual. University of Amsterdam, Amsterdam (2008).
8. Duckett J. Beginning HTML, XHTML, CSS and JavaScript. USA (2010).
9. Oyarce A. Xbee Series 1. Guía del Usuario. (2001).
10. Microchip. PIC16F87X Data Sheet. (2001).