

Telecontrol y Monitoreo de Sistemas Eléctricos a través de una Red de Área Local Inalámbrica

Amhed Ashid Ramos Díaz , Angel Benjamín López Martínez

Universidad Politécnica de Sinaloa.

Niños Héroes #1413, Centro Histórico

C.P. 82000 Mazatlán, Sinaloa

E-mail: [saphoenic@hotmail.com], E-mail: [angelben08@hotmail.com]

Abstract

This project tries to present an alternative way of home automation. By means of the implementation of different technologies, inside the areas of computer science, electronics, electricity and control. Developing a system that allows the automation, monitoring and electrical control of systems, in manual or automatic form. Besides the possibility to automate events; the user has the flexibility of programming tasks and these will be executed automatically by the system in the date and time previously designated.

1. Introducción

Se entiende por domótica al conjunto de sistemas capaces de automatizar una vivienda, aportando servicios de gestión energética, seguridad, bienestar y comunicación, y que pueden estar integrados por medio de redes interiores y exteriores de comunicación, cableadas o inalámbricas, y cuyo control goza de cierta ubicuidad, desde dentro y fuera del hogar.

Este documento consiste de la descripción de un sistema que permite la automatización, monitoreo y control de electrodomésticos dentro y fuera del hogar, desde una red local inalámbrica o bien desde Internet, en este caso específico de 4 luces y 4 conexiones. Es decir, se puede controlar a estos dispositivos de forma manual o utilizando una computadora con conexión alámbrica o inalámbrica, desde cualquier dispositivo móvil compatible con el estándar IEEE 802.11 (Wi-Fi) [1]. Además la posibilidad de automatizar eventos; el usuario tiene la flexibilidad de programar tareas y éstas pueden ser ejecutadas automáticamente por el sistema en la fecha y hora previamente designadas.

Cabe mencionar que el control puede hacerse en forma manual, como remota, de manera independiente

sin que interfieran entre ellas; por ejemplo existe la posibilidad de encender una luz de manera manual, presionando el botón de encendido, y posteriormente apagar la misma luz en forma remota, accediendo a ésta mediante una interfaz visual de desde una computadora.

2. Arquitectura

La conexión de control remota es lograda mediante una interfaz electrónica, conectada a una computadora central que funciona como servidor Web, y éste a su vez a una red de área local como se muestra en la figura 1.

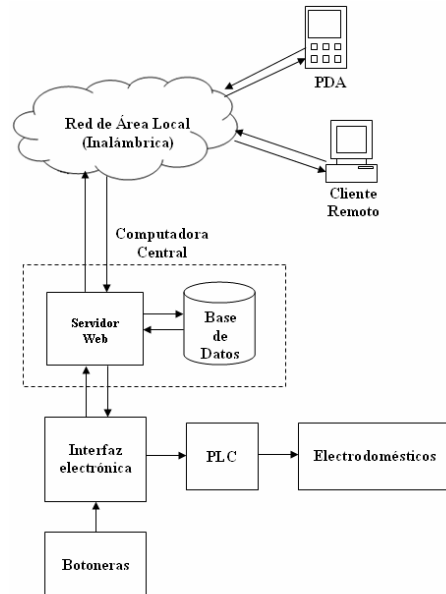


Figura 1. Arquitectura general del sistema.

Dentro de la interfaz hay un microcontrolador central que se encarga de procesar y almacenar la información (encendido o apagado de cada luz o conexión) proveniente de las botoneras o de la computadora central (servidor Web). Esta información

es mandada a un Controlador Lógico Programable (PLC), que se encarga de activar o desactivar las cargas de cada una de las luces y conexiones disponibles; al mismo tiempo, ésta misma es enviada hacia el servidor a través del puerto paralelo para poder monitorear el estado de cada uno de los dispositivos en tiempo real.

En el servidor Web se encuentra el software que realiza la comunicación con la interfaz electrónica, de igual forma éste se conecta a un enrutador inalámbrico, el cual permite acceder al sistema desde otra computadora o dispositivo móvil en forma de una página Web. La información acerca de las tareas programadas por el usuario es almacenada en una tabla dentro de una base de datos local.

3. Computadora central

En la computadora central se encuentran el servidor Web, el software controlador y la base de datos local.

3.1. Servidor Web

El servidor Web utilizó Apache Tomcat [2] versión 4.1.31 sobre Windows Xp Home Edition como servidor de páginas Web. El servidor es la parte principal de comunicación entre el sistema y los usuarios remotos, sus funciones son:

- Coordinar y permitir el acceso de las conexiones hechas por los clientes remotos.
- Ejecutar las clases que controlan la comunicación entre éste y la interfaz electrónica.
- Monitorear el estado de cada uno de los electrodomésticos.
- Conectar a la base de datos local para el registro y la consulta de eventos.

El monitoreo del estado de los dispositivos se logra, ejecutando una clase que siempre esta leyendo datos por el puerto paralelo (provenientes de la interfaz electrónica). La ejecución de los comandos seleccionados por los usuarios es realizada también por el puerto paralelo (escritura).

Para lograr que el sistema ejecute automáticamente una tarea previamente programada, el servidor Web realiza una conexión cada minuto a base de datos local, verifica en la tabla de tareas si existe alguna, de ser así, éste la ejecuta. Las tareas previamente descritas son todas ejecutadas en forma concurrente.

3.2 Software controlador

El software fue desarrollado en Java versión 1.4.2, utilizando las tecnologías Servlets, JavaServer Pages (JSP)[2], Java Database Connectivity (JDBC) [3] y métodos nativos[4].

3.2.1. Métodos nativos

Se requirió el uso de métodos nativos del lenguaje C, para tener control sobre el puerto paralelo. Mediante la implementación de la clase ParallelPort [5], la cual accesa al puerto paralelo (LPT1) directamente desde el lenguaje C.

3.2.2. Lectura y escritura de datos mediante el puerto paralelo

Como se mencionó anteriormente la lectura y escritura de datos por puerto paralelo se hizo mediante la clase ParallelPort. El puerto manda ocho bits (un byte) [6] hacia la interfaz electrónica por la dirección 0x378, de éste sólo se tomaron seis bits, con la intención de separar los 8 dispositivos en 2 registros diferentes, uno para las luces y otro más para las conexiones. Cuatro de los bits corresponden a los datos, uno para indicar la dirección del registro y otro más para señalar a modo de interrupción que se activara la escritura de datos, como se muestra en la figura 2.

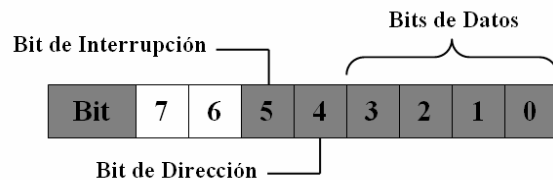


Figura 2. Distribución y asignación de los bits del byte de escritura.

Para la leer por el puerto paralelo se toma la dirección 0x379, esto da como resultado un registro de 5 bits [6], de igual manera esto permite separar los datos en 2 diferentes registros (Figura 3).Es decir, cuando en el bit 4 se lee un 1, significa que los datos (desde el bit 0 hasta la 3) corresponde al registro de las luces.

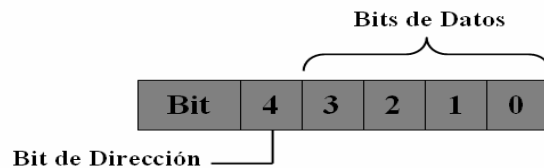


Figura 3. Distribución y asignación de los bits del registro de lectura.

3.2.3. Servlets y JSP

Las tecnologías Servlets y JSP permiten una implementación del modelo de petición-respuesta que se da entre el navegador Web del cliente, y el servidor Web (Apache Tomcat). Cuando un usuario selecciona una página Web del sistema a través de su navegador (la aplicación cliente), se envía una petición al servidor Web, luego, éste responde al cliente enviando una página Web de HTML [1].

Es decir, cuando se solicita una página al servidor Web, éste ejecuta un Servlet que devuelve como resultado una página JSP, con código HTML que el navegador del cliente puede entender y visualizar, como puede verse en la figura 4. Dentro del Servlet se encuentran encapsuladas las clases Java que realizan las tareas desde el servidor (conectar a la base datos, leer y escribir por el puerto paralelo, etc.).

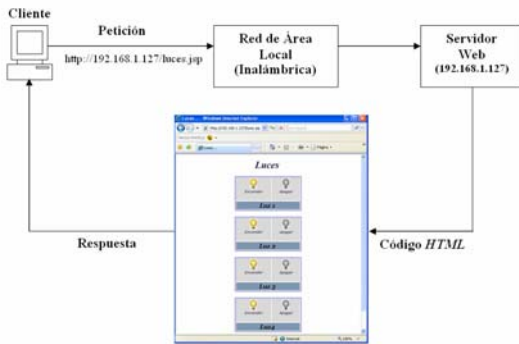


Figura 4. Esquema de la respuesta la petición de una página Web desde el servidor.

3.2.4. JDBC

Para poder manipular los datos almacenados en la base de datos local se utilizó la API JDBC [3], la cual permite la comunicación entre las clases Java y el servidor Web.

3.3. Base de datos

Se instaló MySQL Server versión 5.0 [7] dentro de la computadora central y en una tabla se almacenan los datos de las tareas programadas por los usuarios: fecha y hora del evento, estado del evento, etc.

4. Interfaz electrónica

Ésta consiste en un circuito electrónico con un microcontrolador central [8], el cual coordina las entradas del sistema, dando a la salida las señales necesarias para la activación o desactivación de cada uno de las cargas (luces y conexiones) como se muestra en la figura 5.

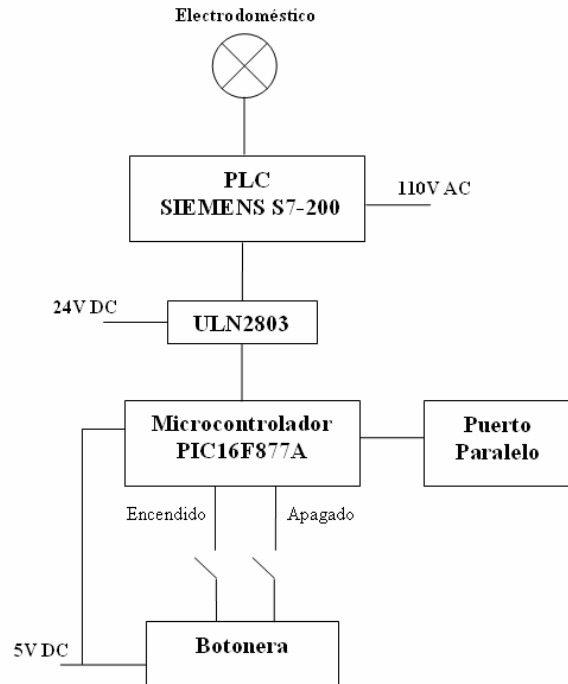


Figura 5. Diagrama a bloques que muestra las diferentes etapas de la interfaz electrónica.

4.1. Microcontrolador

Dentro del microcontrolador existen 4 registros; *luces* y *conexiones*, de 4 bits, *salida*, de 8 bits y *lectura* de 5 bits (figura 6). En los cuales un 1 significa encender, y un 0 apagar (cada dispositivo). El registro *salida* está conformado por la unión de luces y conexiones, con la finalidad de utilizar un solo registro a la salida del microcontrolador.

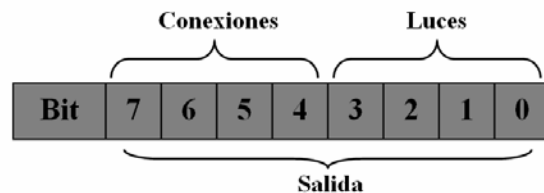


Figura 6. Distribución y asignación de los bits del registro salida.

Los bits de los registros luces y conexiones, son asignados de manera manual o automática. De forma manual cada luz y conexión está conectada a 2 pines del microcontrolador; uno para el encendido y el otro para el apagado. Estos pines reciben una señal de 5 V DC los cuales asignan un 1 o un 0 en bit que le corresponde a cada registro.

De forma automática, el microcontrolador recibe una interrupción externa [8], esto le indica que debe leer los datos de los pines los que esta conectado el puerto paralelo, verificando a que registro se refiere.

Luego, *luces* y *conexiones* se unen para formar *salida*, el cual es pasado al circuito integrado ULN2803 (controlador), éste aumenta la potencia de salida del microcontrolador, que está conectado a la salida 0 o 24 V, tomando como entrada las salidas [9] correspondientes al registro salida del microcontrolador. Finalmente, las salidas del mismo (manejador) se conectan a las entradas digitales de un PLC.

Para la etapa de potencia se utilizó el PLC Siemens S7-200 [10], éste a partir de las entradas activa o desactiva las cargas de cada dispositivo (luces y conexiones).

En la parte que corresponde al monitoreo del estado de los dispositivos, el microcontrolador al detectar un cambio en el estado de *luces* o *conexiones*, pasa el nuevo valor del registro correspondiente (*lectura*), añadiendo un quinto bit para indicar a que registro se refiere y así el servidor Web pueda detectar el cambio correspondiente.

5. Clientes

Los clientes tienen acceso al sistema a través de un enrutador, que tiene 4 puertos para conexión Ethernet [11], además de acceso inalámbrico. Cualquier dispositivo conectado alguno de estos puertos, o que sea compatible con el estándar IEEE 802.11 (Wi-Fi)[1] y que se encuentre dentro del alcance del enrutador tiene acceso al sistema mediante la dirección IP del servidor Web (192.168.1.127). Por ejemplo, si queremos asignar que el sistema ejecute una tarea específica en una fecha y hora determinada, bastaría con escribir dentro del navegador del dispositivo 192.168.1.127/agregarTarea.jsp. Donde

agregarTarea.jsp es la página Web que muestra la interfaz visual para programar una tarea, como se ve en la figura 7.



Figura 7. Agregar una tarea desde un navegador web

De manera similar, para monitorear el estado de las conexiones y las luces tendríamos 192.168.1.127/monitorearConexiones.jsp (figura 8) y 192.168.1.127/monitorearLuces.jsp (figura 9).

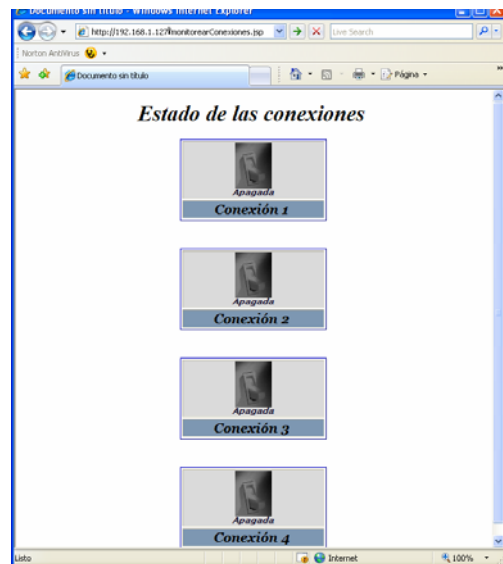


Figura 8. Monitorear el estado de las conexiones.



Figura 9. Monitorear el estado de las luces desde un navegador web.

Para encender o apagar las luces y conexiones tendríamos como dirección: 192.168.1.127/luces.jsp (figura 10) y 192.168.1.127/conexion.jsp (figura 11).

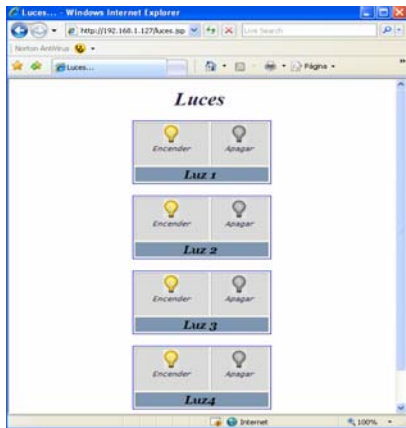


Figura 10. Controlar luces desde un navegador web.

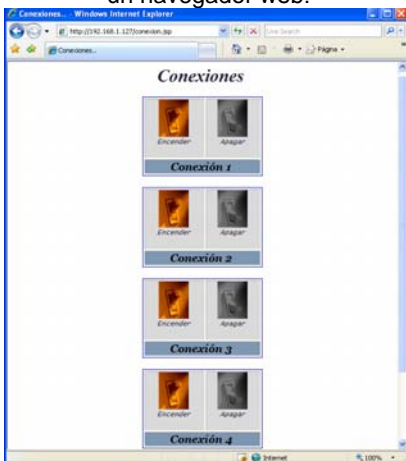


Figura 11. Controlar conexiones desde un navegador web.

6. Conclusiones

Mediante la combinación de las distintas tecnologías que se utilizaron en el diseño e implementación del sistema, tales como: electrónica, redes inalámbricas, software y dispositivos de control se logró automatizar un hogar. Este sistema pretende brindar a los usuarios un mayor confort, seguridad e incluso ahorros de energía. Otra característica es que a diferencia de otros sistemas ya conocidos el costo de éste es significativamente menor.

Una futura implementación puede incorporar un número mayor de dispositivos, así como la posibilidad de lograr el control y monitoreo a través una red de telefonía celular.

7. Referencias

- [1] WI-FI ALLIANCE CERTIFIED PRODUCTS. http://certifications.wi-fi.org/wbcs_certified_products.php CONSULTADO 2-03-2007
- [2] Deitel, JAVA cómo programar, Prentice Hall, México, 2003.
- [3] Java SE - Java Database Connectivity (JDBC) <http://java.sun.com/javase/technologies/database/> CONSULTADO 10-02-2006.
- [4] Schildt, Herbert, JAVA 2 Manual de Referencia, McGraw Hill, España, 2001.
- [5] CLASS PARALLELPORT. <http://www.geocities.com/Juanga69/parport/ParallelPort.html> CONSULTADO 20-10-2006.
- [6] INTERFACING THE PARALLEL PORT. <http://www.beyondlogic.org/epp/epp.htm#6> CONSULTADO 10-10-2005.
- [7] MySQL AB THE WORLD'S MOST POPULAR OPEN SOURCE DATABASE. <http://www.mysql.com/> CONSULTADO 15-11-2005.
- [8] Angulo, José María y Angulo Ignacio, Microcontroladores PIC Diseño práctico de aplicaciones (primera parte), McGraw Hill, México, 2003.
- [9] Tocchi, Ronald J. y Widmer, Neal S., Sistemas Digitales Principios y Aplicaciones, Pearson Prentice Hall, México, 2003.
- [10] S7-200 SIEMENS. http://www.automation.siemens.com/_en/s7-200/index.htm CONSULTADO 10-04-2007.
- [11] Corner, Douglas E., Redes de Computadoras Internet e Interredes, Pearson Prentice Hall, México, 1997.