

Aplicaciones JAVA en Domótica

Ing. Colorado Cháirez Ángel Mario, Ing. Hernández Munguía Lucía,
Ing. Baruch Hernández Samantha, Ing. García Ramírez Iris Itzel,
M.C. Toledo Álvarez José Ángel.

División de Posgrado, Maestría en Ingeniería Electrónica, Instituto Tecnológico de Minatitlán
Blvd. Institutos Tecnológicos s/n Colonia Buena Vista Norte C.P. 96848
Minatitlán, Veracruz, México Tel. (922) 222 4345 Fax (922) 222 4346
thebetterchaires@hotmail.com, luhemu@hotmail.com
sam_bac_her@yahoo.com.mx, iris_itzel44@hotmail.com

Resumen

Se presenta el desarrollo de un sistema de control inteligente con aplicaciones en Domótica, basadas en la integración del lenguaje de alto nivel JAVA con el estándar USB (Bus Serie Universal). Se elabora un programa en JAVA capaz de utilizar los puertos USB de una PC para el control y manejo de dispositivos periféricos, en el caso de sus aplicaciones en Domótica, los dispositivos a controlar corresponderán a los presentes en una vivienda. El PIC18F4550 nos permite crear un puerto serie virtual, el cual se comunica físicamente con la computadora a través del puerto USB, y virtualmente con el programa de JAVA, añadiendo una etapa de potencia necesaria para algunos dispositivos utilizados. Los resultados obtenidos en nuestro sistema de control, son en base al desarrollo de un prototipo a escala de una casa-habitación.

Palabras clave: Domótica, automatización de viviendas inteligentes, API comunicación JAVA, USB, interfaz, compilador.

1. Introducción

Básicamente Domótica es la integración de las diferentes tecnologías presentes en el hogar, con el objetivo de lograr un control y automatización de ellas, mejorando aspectos de seguridad y confort. Desde sus inicios, la Domótica siempre ha contemplado la optimización de los recursos disponibles, principalmente el ahorro de energéticos.

En los países europeos se ha introducido de manera regular, debido a la asociación que se realiza con alta tecnología. Sin embargo, implementando éste prototipo podemos darnos cuenta de que no se

requiere de sistemas muy elaborados y con alto costo, sino lo contrario, utilizando el lenguaje de JAVA en conjunto con el estándar USB y el uso de sistemas digitales se logra un control integrado aplicado en un entorno Domótico.

En los sistemas donde están presentes aplicaciones Domóticas utilizando programación en JAVA, la interfaz electrónica utilizada para comunicarse con el circuito de control son los puertos serie RS-232 y paralelo IEEE-1284, debido a que la librería del API de comunicaciones (javax.comm) limita su uso a estos puertos. Sin embargo una de las tecnologías de mayor aplicación para la conexión de dispositivos periféricos en la actualidad es el protocolo USB, por lo que se decidió buscar la forma de integrarlo con JAVA.

Al no tener JAVA un soporte robusto para el protocolo USB se optó por crear un puerto serie virtual para que así el programa en JAVA perdiera complejidad en su programación. Dicho puerto serie virtual se crea al conectar el circuito microcontrolador a cualquier puerto USB de la PC, con ayuda de un firmware apropiado. Para la comunicación entre JAVA y el puerto serie virtual se utilizó la librería RxTx como alternativa a la librería oficial de Sun Microsystems.

2. Desarrollo

A través de la programación en JAVA, se desarrolla un sistema de control a ejecutarse en la PC se conecta al microcontrolador utilizando el protocolo de comunicación USB; el dato recibido por el microcontrolador es sacado de forma paralela a través de sus puertos, esta salida pasa por una etapa de potencia para finalmente aplicar el control sobre los dispositivos deseados. De manera general, nuestro

El sistema de control se muestra en el diagrama a bloques de la figura 1.

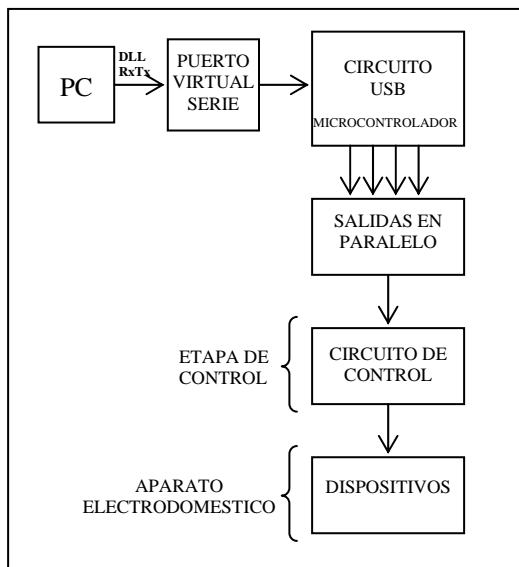


Figura 1. Esquema del Sistema de Control

2.1 Programación en JAVA

La librería que nos ofrece Java para lograr la comunicación del programa a ejecutar y el microcontrolador (parte física), se encuentra en el paquete "javax.comm", este proporciona el soporte para dispositivos serie y paralelo, sin embargo al no estar disponible su descarga en el sitio oficial de SUN Microsystems, se pueden utilizar como alternativa la librería genérica RxTx, permitiéndonos de la misma manera, la manipulación de los puertos.

El microcontrolador permite crear un puerto serie virtual, el cual se comunica físicamente con la computadora a través del puerto USB, y virtualmente con el programa de JAVA. Además este microcontrolador se utiliza para proporcionar las diferentes salidas de control, convirtiendo la señal serie en paralelo.

Generalmente los microcontroladores se programan en ensamblador (assembler), debido a la complejidad del protocolo USB hemos optado por realizar la programación del mismo en un compilador de C (PICCompiler) el cual incluye librerías con soporte USB, incluyendo así otra herramienta importante dentro del sistema como lo es el lenguaje C, del cual ocuparemos sus ventajas para establecer la comunicación y enlazarlo con el lenguaje de alto nivel que en éste caso será como ya se ha mencionado: JAVA.

Otras librerías utilizadas en el desarrollo del programa, también de gran importancia en la implementación son:

- Java.awt. Permite crear una interfaz gráfica con el usuario. Con este paquete podemos generar componentes gráficos como botones, menús, panel de tareas, áreas de escritura entre otras clases.
- Java.io. Este paquete contiene las clases de acceso a ficheros (entrada/salida): FileInputStream y FileOutputStream.
- Java.lang. Este paquete incluye las clases del lenguaje Java propiamente dicho: Object, Thread, Exception, System, Integer, Float, Math, String, etc.
- Java.util. Es una miscelánea de clases útiles para muchas cosas en programación. Se incluyen, entre otras, Date (fecha), Dictionary (diccionario), Random (números aleatorios) y Stack (pila FIFO).

2.2 Interfaz gráfica

La interfaz con el usuario para activar los diferentes eventos la proporciona una ventana en la cual se muestran diferentes botones, indicando cada uno el objeto a manipular y teniendo éste a su vez dos acciones dentro de su función que son las de encender y apagar. La figura 2 nos muestra la ventana de control a utilizar.

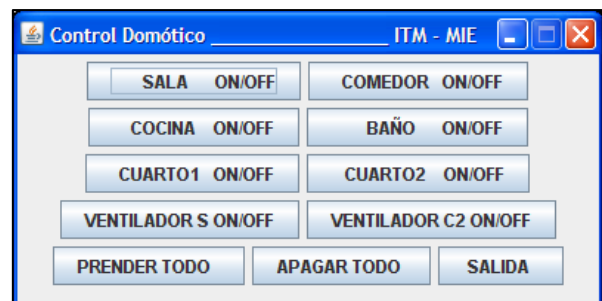


Fig. 2. Ventana de control

La acción del botón pulsado se envía al microcontrolador y éste a su vez al dispositivo indicado. Las diferentes propiedades de JAVA como lenguaje de alto nivel nos permiten crear un archivo ejecutable, así como un entorno más amigable con el usuario como lo es añadir gráficos o en todo caso el ambiente real de trabajo.

Utilizando las librerías de JAVA se podría incluir un servicio integrado a Internet con el fin de poder realizar un monitoreo en tiempo real a larga

distancia de las actividades que se realicen en nuestro entorno Domótico así como un control a distancia de éste mismo. Esta es una de las razones más importante para que el lenguaje de alto nivel utilizado en el diseño sea la programación en JAVA, su portabilidad hacia el Internet.

2.3 Uso del estándar USB

Como ya conocemos, el USB nace por la necesidad de crear conectores más sencillos y con mayores ventajas sobre los puertos serie y paralelo. Esta interfaz consta de 4 hilos, con una velocidad de 12 Mbps y además es "plug and play", lo cual nos ahorra tiempo en aplicación, distribuye 5V para alimentación, transmite datos y es en la actualidad la interfaz de comunicación con mayor alcance.

Hoy en día en una PC por lo general se pueden encontrar por lo menos tres conexiones USB disponibles, y algunas contarán con solo un puerto serial, otros modelos no muy recientes o demasiado completos contendrán por lo menos un puerto paralelo.

El estándar USB contempla varias clases de dispositivos para funcionalidades encontradas comúnmente en los dispositivos. Por ejemplo, existe una clase para las cámaras digitales, otra para los escáners, otra para las impresoras, etc. Las clases de dispositivos fueron inventadas para mejorar la interoperabilidad de los dispositivos. Así, cualquier sistema operativo que tenga un driver para trabajar con cámaras digitales puede leer fotos de la cámara digital que esté diseñada para cumplir las especificaciones de dicha clase de dispositivos.

En particular, para nuestro proyecto optamos por usar la clase de dispositivo CDC (Communication Device Class) que básicamente emula una conexión serie sobre el puerto USB. La razón por la cual optamos esta clase fue que el mecanismo de una conexión serie nos pareció un enfoque simple y efectivo para intercambiar simultáneamente información de control y datos. Además, al no haber ninguna clase prevista para control Domótico, una comunicación serie es el método más directo de implementar un driver propio puesto que solo basta con enviar y recibir cadenas de caracteres. En conclusión, escogimos la clase CDC por su sencillez y flexibilidad.

En comparación con la velocidad de transferencia de datos, el estándar USB 1.1 tenía dos velocidades de transferencia: 1.5 Mbit/s para teclados, mouse, joysticks, etc., y velocidad completa a 12 Mbit/s. La mayor ventaja del estándar USB 2.0

es añadir un modo de alta velocidad de 480 Mbit/s. En su velocidad más alta, el USB compete directamente con FireWire. Un puerto paralelo entre 600 Kbps a 1,5 Mbps. Un puerto serial puede llegar hasta 112 Kb/s.

2.4 Microcontrolador

Como ya se ha mencionado, el microcontrolador permite crear un puerto serie virtual; el cual se comunica físicamente con la computadora a través del puerto USB además virtualmente con el programa de JAVA. Además este microcontrolador se utiliza para proporcionar las diferentes salidas de control, convirtiendo la señal serie en paralelo.

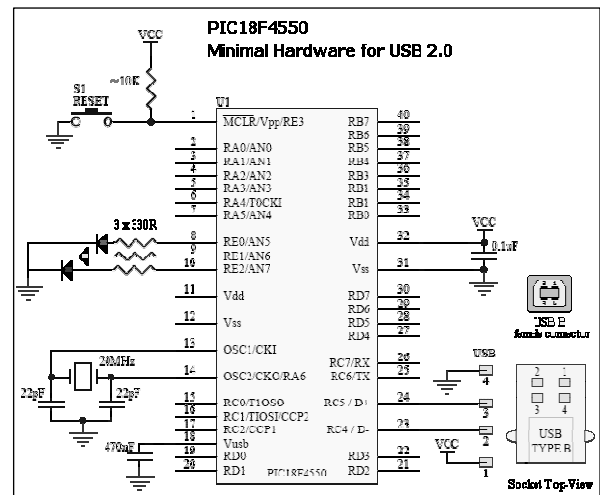


Figura 3. Diagrama del Hardware para USB 2.0

El diagrama de la figura 3, muestra el hardware mínimo necesario para hacer funcionar esta etapa del proyecto, la importancia se centra en el microcontrolador PIC18F4550 que cuenta con el manejo del puerto USB, del cual se muestran sus conexiones en la figura, para más detalle se consultó la hoja de datos del fabricante.

Generalmente los microcontroladores se programan en ensamblador (assembler), debido a la complejidad del protocolo USB hemos optado por realizar la programación del mismo en un compilador de C (PICCompiler) el cual incluye librerías con soporte USB, incluyendo así otra herramienta importante dentro del sistema como lo es el lenguaje C, del cual ocuparemos sus ventajas para establecer la comunicación y enlazarlo con el lenguaje de alto nivel.

3. Etapa de control

Los sistemas digitales son los encargados de controlar a otros sistemas, como en este caso, el microcontrolador realiza funciones de control que necesitan estar interconectadas a una etapa de manejo de potencia, para actuar sobre cargas resistivas o inductivas en sistemas de iluminación, ventilación, etc.

En esta última etapa, se realiza la adaptación del circuito a un dispositivo real; se realiza el diseño de un circuito que nos proporcione la suficiente corriente para poder encender cualquier equipo, a la vez nos brinde protección contra alguna descarga.

El driver utilizado es el TC4424, fue elegido debido a sus características con respecto al voltaje. Este chip tiene 8 pines. Dos de ellos son para la alimentación (5v), dos para las entradas de baja corriente y dos para las salidas de media/alta corriente. Es un dispositivo realmente sencillo de utilizar. Cada vez que apliquemos un 1 (alrededor de 5v) en las entradas de baja corriente, obtendremos un 1 (5v) en las de salida de media/alta corriente. Cada vez que apliquemos un 0 (alrededor de 0v) en las entradas de baja corriente, obtendremos un 0 (0v) en las de salida de media/alta corriente. Cada chip tiene dos entradas y dos salidas, así que podremos utilizar un chip para cada elemento.

En figura 4, se muestra la conexión eléctrica entre los elementos en primer lugar para el prototipo en fase uno, el cual consistió en controlar elementos de CD., como se muestra en la figura; implementado en una maqueta casa-habitación, utilizando un sistema a escala donde las lámparas incandescentes son representadas por LED's y equipos como ventiladores por dispositivos semejantes operados a corriente directa utilizando una interfaz apropiada; consiguiéndose así, una manera didáctica de comprender su aplicación en Domótica.

En una aplicación a dispositivos de CA, se utilizan los relevadores, la ventaja de utilizarlos es la completa separación eléctrica entre la corriente de accionamiento y los circuitos controlados por los contactos lo que facilita el uso de altos voltajes o elevadas potencias con pequeñas tensiones de control y la posibilidad de control de un dispositivo a distancia mediante el uso de pequeñas señales de control.

La descripción de su funcionamiento es el siguiente: cuando el relevador se excita, la laminilla cambia su posición y es entonces cuando el ventilador recibe la señal de CA, en caso contrario no trabaja el

ventilador. Esta implementación es útil para cualquier electrodoméstico que se quiera controlar.

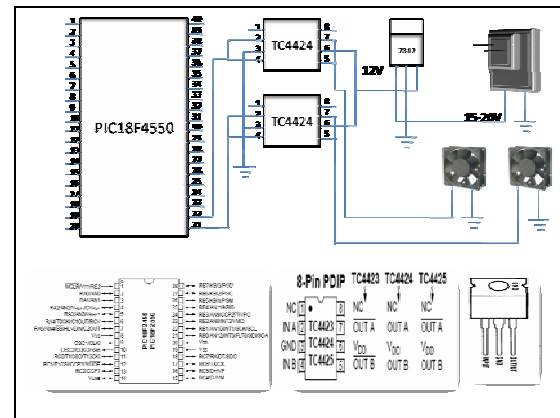


Figura 4. Circuito eléctrico para elementos en CD

4. Análisis de resultados

La razón de buscar integrar el lenguaje JAVA con el protocolo USB, es principalmente para trabajar con equipos electrónicos o mecánicos monitoreados desde un computador, y al no tener disponibles librerías para la utilización de este tipo de puertos, se crea un puerto serie virtual en el microcontrolador y así implementar nuestro sistema de control.

Los resultados obtenidos se basan en el prototipo a escala de una casa-habitación, en la cual se manipula la iluminación de la "casa" y se conectaron dos ventiladores operados a corriente directa. A través de nuestra interfaz gráfica, realizamos el encendido y apagado de los LED's en los diferentes espacios creados, los cuales constan de una sala, comedor, cocina y un baño. En la figura 2, previamente descrita se pueden observar cada uno de los botones que presenta nuestro prototipo, y al hacer la activación de los dispositivos mencionados esta es inmediata.

En la primera prueba hubo problemas con el cableado de la maqueta ya que no permitía el encendido de un LED, para la estructura se utilizó cable par trenzado, se optó por cambiar solo la falla y no modificar el cableado de todo la "casa". El control inteligente aplicado al prototipo ha sido implementado favorablemente, lográndose el objetivo planteado, consiguiendo así una manera didáctica de comprender su aplicación en Domótica.

La parte final que es la etapa de potencia, es relativamente sencilla en comparación con la programación que se requiere para la PC y el

microcontrolador, los resultados obtenidos son satisfactorios y visibles desde la salida paralela que el microcontrolador proporciona a los dispositivos eléctricos (sean lámparas o electrodomésticos).

5. Conclusiones

En el diseño de éste proyecto solo se evalúa un porcentaje del alto nivel del lenguaje de programación de JAVA, dejando abierta la opción de poder realizar más aplicaciones como así se requiera, especialmente como una introducción de la domótica y el desarrollo de sistemas inteligentes.

En la actualidad existen ambientes Domóticos que son manejados mediante la programación en JAVA, sin embargo en México solo se considera en lugares donde se trabaje con ramas profesionales de Mecatrónica o Robótica. Así que tomamos los lenguajes de programación actuales y a través del estándar USB buscamos introducirnos en el desarrollo de arquitecturas inteligentes.

Se dará seguimiento al proyecto buscando optimizar el manejo del USB. Una de las principales razones de utilizar este lenguaje fue su capacidad de emigrar hacia el Internet, además la flexibilidad de la programación de JAVA, que permite realizar una variedad de proyectos relacionados con el manejo de puertos, adaptándose a las necesidades del usuario, con la posibilidad de interactuar con ventanas de control. Incluso a partir de esto, otra manera de utilizar las aplicaciones de JAVA junto con el lenguaje C sería modificar nuestro sistema de ahorro de energía, en los cuales de acuerdo a la cantidad de luz solar presente o incluso con presencia humana realicemos el control de la iluminación en el

ambiente Domótico, siendo pauta para futuros proyectos de programación.

Referencias

- [1] Deitel, H y Deitel,P.J. “*Como programar el C/C++ y JAVA*”. Pearson Educación, México, 4ta. Edición, 2004.
- [2] Froufe, A. “*JAVA 2: Manual y tutorial de JAVA*”. Ra-Ma Editorial, España, 4ta. Edición, 2006.
- [3] Quinteiro José Ma., Lamas Javier, Sandoval Juan D. “*Sistemas de control para viviendas y edificios: domótica*”. Editorial Paraninfo. España, 2005.
- [4] Romero, Cristóbal, Vázquez, Francisco, De Castro, Carlos. “*Domótica e Inmótica. Viviendas y edificios inteligentes*”. Editorial Ra-Ma. España, 2005.
- [5] Maloney, J. Timothy. “*Electrónica industrial. Dispositivos y Sistemas*”. Editorial Prentice Hall Hispanoamericana. México, 2000.
- [6] Sun Microsystem, JAVA, **API specifications**. <http://java.sun.com/products/javacomm/referenc e/api/javax/comm/package-summary.html>. Ultima visita: 18:40 hrs. 03/Septiembre/07.
- [7] RXTX: The Prescription for Transmission. <http://users.frii.com/jarvi/rxtx/index.html>. Ultima visita: 16:36 hrs. 04/Septiembre/07.