

## DOMÓTICA: CONTROL DE ENCENDIDO DE DISPOSITIVOS POR MEDIO DE UNA APLICACIÓN MÓVIL Y UNA RASPBERRY PI

Ing. Rodrigo Vázquez López  
rodrigo\_em2@hotmail.com  
Ing. Esther Viridiana Vázquez Carmona  
ev.vazquezcc@gmail.com  
Dr. Juan Carlos Herrera Lozada  
jcrs.ipn@gmail.com  
Instituto Politécnico Nacional  
Centro de Innovación y Desarrollo Tecnológico en  
Cómputo (CIDETEC)  
Dr. Jacobo Sandoval Gutiérrez  
j.sandoval@correo.ler.uam.mx  
Universidad Autónoma Metropolitana  
Departamento de Procesos Productivos - Unidad  
Lerma

### Resumen

El objetivo del presente trabajo es presentar el desarrollo y funcionamiento de un prototipo enfocado a la domótica para el control del encendido y apagado de dispositivos por medio de una aplicación móvil y la microcomputadora Raspberry Pi. El sistema presentado utiliza dispositivos de bajo costo, arquitecturas abiertas y hace uso del modelo de programación concurrente cliente-servidor.

### Introducción

Conforme la tecnología avanza, conceptos y términos como big data, Internet de las Cosas y domótica se ven reflejados en los dispositivos con los que interactuamos día a día. Particularmente durante los últimos años, la domótica se ha popularizado con la aparición de los dispositivos inteligentes.

Se le llama domótica al conjunto de tecnologías aplicadas al control y la automatización inteligente de la vivienda [1]. Una de sus principales características es la interacción del usuario con dispositivos inteligentes. Dichos sistemas recolectan información proveniente de sensores para tomar decisiones que mejoren el confort, automaticen procesos del hogar y proporcionen seguridad y comodidad [2].

Una problemática particular es el costo de facturación de la energía eléctrica y su impacto al medio ambiente debido a su producción y distribución. La domótica ofrece sistemas inteligentes que permiten el uso eficiente de la misma, minimizan el gasto energético y además ayudan a prevenir riesgos o accidentes que pueden ocurrir al descuidar la operación de aparatos eléctricos al poder encenderlos o apagarlos desde ubicaciones remotas.

Con base en la mayoría de los sistemas de la domótica [3] se definen dos componentes principales: un teléfono inteligente y un microcontrolador, particularmente, en este último componente con la aparición en el mercado de tarjetas de desarrollo de bajo costo y arquitecturas abiertas como Raspberry Pi, se han desarrollado diversos prototipos de sistemas domóticos y de automatización para el hogar como los que se observan en [4, 5 y 6].

El objetivo de este trabajo es mostrar el desarrollo y funcionamiento de un prototipo de sistema domótico utilizando arquitecturas abiertas y dispositivos de bajo costo como la microcomputadora Raspberry Pi en conjunto con un sistema desarrollado que utiliza una aplicación móvil para el control de encendido y apagado de dispositivos conectados a la corriente eléctrica. Los resultados obtenidos han permitido validar el funcionamiento del sistema y muestran la facilidad de desarrollo al incorporar dichas tecnologías al diseño de prototipos.

### Componentes del sistema

La figura 1 muestra el diagrama de funcionamiento del sistema, el cual se compone de tres bloques: Aplicación cliente, Servidor y Receptor.

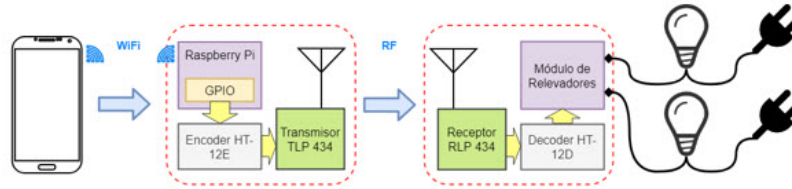


Figura 1: Diagrama general del sistema domótico realizado

El funcionamiento se puede describir de la siguiente forma: la aplicación cliente controla el encendido y apagado de los dispositivos conectados a los relevadores, para ello, envía al servidor la información del dispositivo a controlar utilizando la red Wifi. El servidor, compuesto de una microcomputadora Raspberry Pi recibe la información, la procesa y genera las salidas adecuadas en el puerto GPIO de la tarjeta, el cual se encuentra conectado al bloque de transmisión. Finalmente, el receptor decodifica la información y activa o desactiva los relevadores, encendiendo o apagando los dispositivos.

### Aplicación Cliente

Consiste en una aplicación diseñada para dispositivos móviles con sistema operativo Android (smartphones o tablets) y cuyo objetivo es funcionar como interfaz entre el usuario y el sistema. El desarrollo de la aplicación se realizó utilizando el Entorno de Desarrollo Android Studio y el lenguaje de programación Java [7].

La figura 2 muestra la pantalla principal y el prototipo de la aplicación en funcionamiento. La interfaz requiere como datos de entrada el valor de la dirección del transmisor, la dirección IP del servidor (Raspberry Pi) y el puerto de conexión.



Figura 2: Pantalla principal y prototipo de la aplicación

### Servidor

El bloque se compone de una microcomputadora Raspberry Pi en la cual se conectan por medio del puerto GPIO el transmisor TLP 343 y el circuito integrado HT-12E (figura 3). El circuito TLP 434 es un transmisor de radiofrecuencia de bajo costo ideal para proyectos de control remoto o de transmisión de datos. El transmisor opera en una frecuencia de 433.92 MHz, velocidad de transmisión de hasta 4.8

Kb/s y puede alcanzar distancias de hasta 100 metros en espacios abiertos. El voltaje de operación puede variar de 2-12V DC consumiendo una corriente de 20mA [8]. Por su parte, el circuito integrado HT-12E es un circuito codificador serial de datos para aplicaciones de control remoto que utilizan radiofrecuencia como medio de transmisión. El circuito puede codificar 8 bits de direcciones y 4 bits de datos, además, no requiere de un oscilador externo por lo que solo requiere de una resistencia para generar las señales [9].

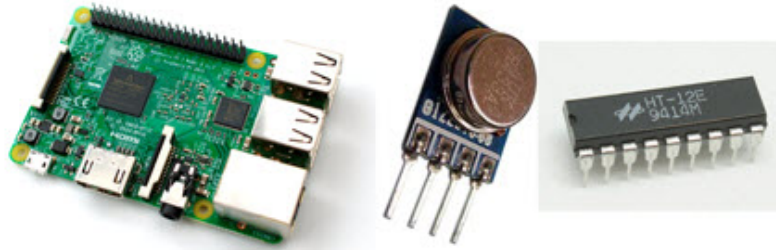


Figura 3: Componentes del Servidor: Raspberry Pi, transmisor TLP 434 y circuito HT-12E.

Raspberry Pi es una microcomputadora de bajo costo y tamaño reducido con velocidades de procesamiento desde 900 MHz hasta 1.4 GHz [10]. Por ser una microcomputadora, la tarjeta requiere la instalación de un sistema operativo para su funcionamiento. Uno de los sistemas operativos disponibles es Raspbian el cual deriva de Debian y utiliza el núcleo GNU/Linux por lo que cuenta un repositorio completo de herramientas de desarrollo de uso libre [11].

La figura 4 muestra el diagrama de comunicación entre procesos de la aplicación cliente y el programa que se ejecuta la Raspberry Pi, el cual gestiona las peticiones recibidas por una o varias instancias de la aplicación cliente y escribe los valores de salida en el puerto GPIO de la tarjeta. El programa se diseñó utilizando el modelo cliente-servidor y se codificó utilizando el lenguaje de programación Java. La elección de Java como lenguaje programación se debió a la facilidad de implementación del modelo previamente mencionado y del manejo de concurrencia utilizando sockets e hilos [12]. Es importante mencionar que para poder acceder el puerto GPIO utilizando Java es necesario utilizar la librería PI4J disponible en [13].

#### Receptor

La recepción de datos se realiza utilizando el circuito RLP 434, el decodificador HT12-D y el módulo de relevadores HL-525 (figura 5). El receptor RLP 434 es el complemento del transmisor TLP 434. Puede trabajar con frecuencias de 315, 418 o 433.92 MHz [14]. El circuito integrado HT-12D es complementario al circuito HT-12E utilizado en el servidor. El HT-12E cuenta con un sistema a prueba de errores que compara tres veces el valor de la dirección recibida en los paquetes de datos con la dirección colocada en la entrada del circuito, si la dirección coincide los datos previamente recibidos se decodifican y envían a las salidas del circuito [15]. Finalmente, el módulo HL-525 consta de dos relevadores optoacoplados que soportan cargas eléctricas de hasta 127 V y 10 A en AC o 28 V y 10 A en DC [16].

#### Pruebas y resultados

Se realizó el ensamble de la interface con la tarjeta Raspberry PI mediante una protoboard y unas lámparas de 40 watts en un espacio de 60 m<sup>2</sup>, En la Figura 6 podemos observar el dispositivo en pruebas.

La primer prueba realizada fue utilizando las condiciones ideales, es decir, sin interferencias entre la Raspberry PI y el router inalámbrico WIFI. En una segunda prueba se considero reducir la calidad de la señal en uno y dos tercios de la calidad de la señal.

Los resultados muestran que el sistema funciona correctamente cuando la calidad de la señal Wifi que recibe el dispositivo móvil es mayor a un tercio. Debido a que el dispositivo móvil espera a que la señal y la conexión sea estable para enviar paquetes de datos, cuando la conexión es inferior a un tercio, hubo una tasa de fallos en la información transmitida, Los errores provocan que la respuesta en las lámparas fuera diferente a la esperada.

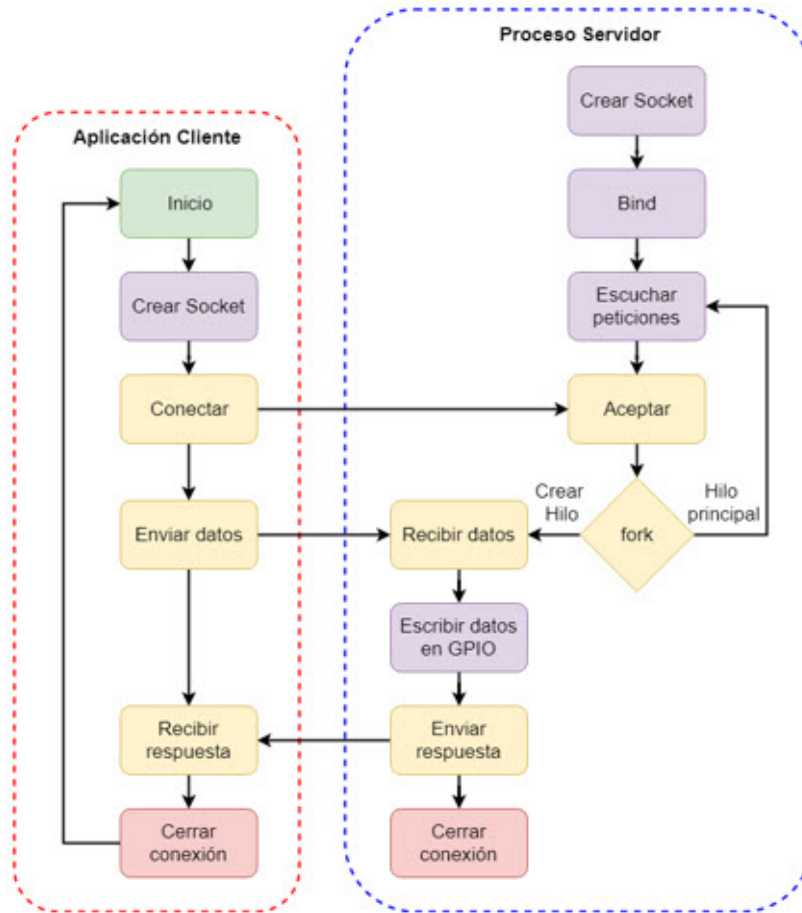


Figura 4: Diagrama de comunicación entre procesos

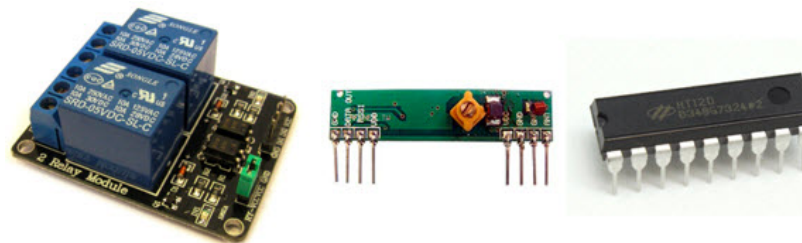


Figura 5: Componentes del Receptor: Módulo de relevadores HL-525, receptor RLP 434 y circuito HT-12D.

### Conclusiones

El uso de arquitecturas abiertas y dispositivos de bajo costo permite la rápida construcción y desarrollo de prototipos enfocados a la domótica. El prototipo presentado obtuvo resultados satisfactorios y demuestra la viabilidad del proyecto.

La elección de una plataforma como Raspberry Pi permitió la implementación del modelo cliente-servidor en el desarrollo del prototipo, el cual facilitó el diseño de la arquitectura del sistema, además, los recursos computacionales que ofrece la tarjeta se pueden aprovechar en la ejecución de otros algoritmos más sofisticados.



Figura 6: Pruebas de funcionamiento del prototipo.

Una de las ventajas de la elección de Wifi es que permite que el alcance de comunicación entre el servidor y la aplicación cliente sea mayor a las que ofrecen otras tecnologías como bluetooth. Por otra parte, Wifi no requiere códigos adicionales para iniciar la transmisión de datos. Las ventajas antes mencionadas son un punto fuerte en espacios que tengan una óptima recepción de señal.

Es importante mencionar que como trabajo futuro se busca ampliar las funciones que realiza la aplicación móvil, sobre todo en la seguridad de encendido o apagado, ya que en aparatos con potencial de riesgo o importancia en el hogar, el hecho que no se cumpla cabalmente con la funcionalidad puede ser fatal, por ello, este contexto nos lleva a aplicar conceptos de big data para el análisis de información con algoritmos inteligentes o de aprendizaje profundo para la toma de decisiones, y asociando finalmente con tecnologías como el Internet de las Cosas.

#### Referencias

1. [1] Asociación Española de Domótica e Inmótica CEDOM(junio 2019) <http://www.cedom.es/sobre-domotica/que-es-domotica>
2. [2] Huidobro, J. Manuel (2007) *La domótica como solución de futuro*, FENERCOM. Madrid.
3. [3] M. Asadullah and A. Raza (2016) *An overview of home automation systems*, "2016 2nd International Conference on Robotics and Artificial Intelligence (ICRAI), Rawalpindipp. 27-31.
4. [4] Jain, S., Vaibhav A., & Goyal L. Raspberry Pi (2014.) *based interactive home automation system through E-mail. Optimization, Reliability, and Information Technology (ICROIT), 2014 International Conference on. IEEE*
5. [5] Monteiro, P., Tomé P., & Albuquerque D. (2015) *Domotics control system architecture. Information Systems and Technologies (CISTI), 2015 10th Iberian Conference on. IEEE*

6. [6] Ramesh, R. & Lingaiah, J., Raspberry Pi (**octubre 2016**) *Based Interactive Home Automation System through E-Mail, International Journal of Innovate technologies (IJITECH), Vol.04, Issue.15*
7. [7] Google Foundation(**junio 2019**) <https://developer.android.com/studio> Conoce Android Studio.
8. [8] Laipac Technology Inc. LAIPAC (**junio 2019**) <http://www.laipac.com/rf-transmitters.html>
9. [9] Holtek Semiconductor (**Taiwan. 2002**) *HT12A/HT12E 212 Series of Encoders Datasheet. Science Park, Hsinchu*
10. [10] Raspberry Pi Foundation (**junio 2019**) , <https://www.raspberrypi.org>
11. [11] Raspbian Team, Raspbian FAQ (**junio 2019**) <https://www.raspbian.org>
12. [12] Coulouris, G., Dollimore, J., & Kindelberg T.(**2005**) *Distributed Systems: Concepts and Design. Pearson Education.*
13. [13] The PI4J Project (**junio 2019**) *Java I/O library for the Raspberry Pi. <http://pi4j.com>*
14. [14] Laipac Technology Inc.(**junio 2019.**) LAIPAC, <http://www.laipac.com/rf-receivers.html>
15. [15] Holtek Semiconductor (**Taiwan. 2002**) *HT12D/HT12F 212 Series of Decoders Datasheet. Science Park, Hsinchu*
16. [16] Modtronix Engineering (**junio 2019**) , <http://modtronix.com/mod-rly2-5v.html>