Sistema a la Medida para el Control y Monitoreo de una Casa Inteligente por medio de Servidor Web

Greivin Fallas Sánchez
Escuela de Ingeniería en Computadores
Tecnológico de Costa Rica
Cartago, Costa Rica
Email: greivinfas15@gmail.com

Arturo Salas Delgado
Escuela de Ingeniería en Computadores
Tecnológico de Costa Rica
Cartago, Costa Rica
Email: artsaldel@gmail.com

César Solís Valverde
Escuela de Ingeniería en Computadores
Tecnológico de Costa Rica
Cartago, Costa Rica
Email: csolisv56@gmail.com

***Resumen*—En el siguiente documento se presentan los principales aspectos relacionados a la elaboración de un sistema empotrado, Raspberry pi 2, con procesador ARM y un sistema operativo a la medida para dicho fin, con las bibliotecas y dependencias necesarias. Esto con el fin de desarrollar una aplicación móvil para el control y monitoreo de luces, puertas y una cámara de seguridad en una casa inteligente o domótica la cual tiene conexión por medio de un servidor web. Colocando dicha implementación como parte de lo que hoy en día se conoce como el Internet de las Cosas (IoT).**

Palabras clave: Sistema empotrado, IoT, , Raspberry Pi, domótica, servidor web., sistema operativo.

1. Introducción

El campo tecnológico ha mostrado un crecimiento en el desarrollo de aplicaciones que faciliten el diario vivir de la humanidad en general, los sistemas embebidos han contribuido en gran parte este campo dando solución a problemas específicos y de interés para situaciones de necesidad y/o comodidad de las personas.

De la mano con los sistemas embebidos surge el Internet de las Cosas, el cual viene a "facilitarnos la vida"de todo punto de vista, en este caso la aplicación consiste en poder controlar las luces y puertas de nuestro hogar desde un dispositivo móvil, además de poder monitorear de manera gráfica para brindar una herramienta de seguridad.

En este documento se detallan los principales aspectos del desarrollo de la solución para el problema anteriormente planteado por medio del uso de una Raspberry Pi 2 y un sistema operativo a la medida con Yocto en el cual se realiza compilación cruzada de las bibliotecas y dependencias necesarias, en lenguaje de programación C.

Además se cuenta con una aplicación móvil para la plataforma Android, en la cual se controlan las señales de la casa, luces y puertas; así mismo la solicitud de fotografías. Para el diseño del prototipo se utilizaron LED's para la representación de las luces, botones para la simulación de las puertas y una cámara web de baja resolución para la toma de fotografías.

2. Sistema Desarrollado

El sistema desarrollado corresponde a un sistema a la medida para poder controlar y monitorear una casa inteligente por medio de un servidor web y un cliente en Android. A continuación se detallan sus componentes más importantes.

2.1. Servidor Web

El servidor web, escrito en lenguaje C, corre en la placa Raspberry Pi, el cual no requiere ninguna biblioteca adicional o dependencia.

El servidor corresponde a un fork(), el cual permite múltiples conexiones simultáneas, el servidor cuando se conecta un nuevo cliente se bifurca, de esta forma, el diálogo con el cliente se lleva a cabo desde un nuevo proceso mientras que el proceso padre (el primero) sigue escuchando en el puerto esperando conexiones. [1]

2.2. Aplicación Android

La aplicación móvil se desarrollo en Visual Studio 2015 con la plataforma Xamarin la cual es utilizada para crear aplicaciones nativas en Android, IOS y Windows Phone por medio de código en C# y XAML, en este caso se desarolló para el sistema Android, en versiones superiores a la 4.0 KitKat.

2.2.1. Base de Datos:. Además en la aplicación móvil se implementó una base de datos para almacenar la información sobre el estado de las luces y puertas de la

casa, dicha base fue creada mediante SQLite, un motor de base de datos relacional compatible con ACID, una biblioteca relativamente pequeña escrita en C y de uso común en aplicaciones móviles.

2.2.2. Encriptación y Cliente Web:. Así mismo, la aplicación funciona como un cliente web sobre el servidor que corre en la Raspberry Pi 2, de esta manera se pueden generar las consultas y modificaciones necesarias al servicio, con las solicitudes POST y GET; por otra parte se implementó un sistema de inició de sesión a la aplicación con el protocolo de seguridad SHA o Secure Hash Algorithm por sus siglas en inglés, para encriptar la información del usuario y brindar una mayor seguridad y confianza al cliente.

2.3. Biblioteca GPIO

Con el fin de desarrollar una interfaz de comunicación entre los componentes de hardware que simulan el comportamiento de luces y puertas se desarrolló una biblioteca en C y con un programa intermedio para realizar las llamadas correspondientes desde el servidor con dicha biblioteca mediante una llamada al sistema indicando la funcion que se requiere y los datos correspondientes de cada una de las funciones.

La biblioteca posee las siguientes funciones:

- init(): este método se encarga de preparar el pin para una conexión, o sea, para poder escribir o leer bytes en ese pin.
- finish(): se encarga de finalizar la conexión con un pin.
- pinMode(): este método se encarga de preparar al pin para lectura o para escritura.
- digitalRead(): este método se encarga de leer un 0 o un 1 desde un pin, una vez que se inicializó la conexión con el mismo.
- digitalWrite(): este método se encarga de escribir un 0 o un 1 a cierto pin, una vez que se inicializó la conexión.
- blink(): este método se encarga de prender y apagar un led en cierto pin a cierta frecuencia.
- lightOn(): este método se encarga de prender un led ubicada en cierto pin. En sí, utiliza al método digitalWrite.
- lightOff(): este método se encarga de apagar un led ubicada en cierto pin. En sí, utiliza al método digitalWrite.
- readDoor(): este metodo se encarga de leer el valor de un push button, que representa el estado de una puerta. En sí, utiliza el método digitalRead.
- initLights(): se encarga de inicializar todos los pins correspondientes a leds para poder escribir sobre ellos.
- initDoors(): se encarga de inicializar todos los pins correspondientes a push buttons para poder leer su estado el algún momento

2.4. Fswebcam

Para la toma de fotografías se utilizó Fswebcam, el cual ”es una aplicación que funciona bajo línea de comandos (CLI) y que permite capturar imágenes con dispositivos V4L1/V4L2 (Video for Linux versiones 1 y 2) compatibles (es decir, Webcams compatibles con Linux). Reduce el ruido de las imágenes capturadas, mejorando su calidad y permite guardar imágenes tanto en formato JPEG como PNG.” [2]

2.5. Sistema Físico

Se desarrollo un modelo a escala de una casa por medio de un diseño creado por el grupo de trabajo, con ayuda de Adobe Illustrator CC 2017 y las máquinas de corte láser que brinda la institución, en material MDF de 0.3 cm. Además el sistema eléctrico se realiza en la parte superior de la casa, denominada cielorraso, en el cual se colocaron las luces LED y la tarjeta de desarrollo Raspberry Pi de la manera más estética posible. En la Figura 1 se puede observar el diseño final de la maqueta.

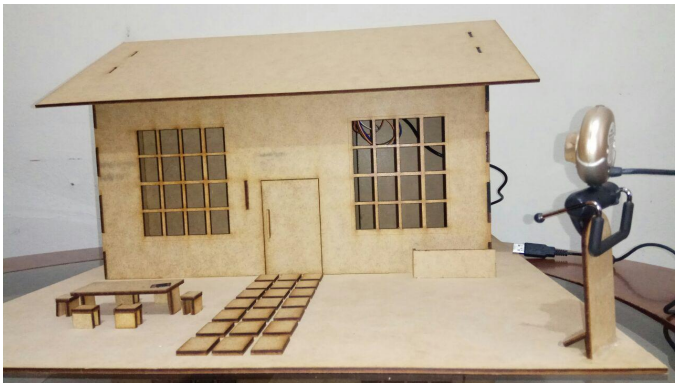

Figura 1. Vista de la Maqueta.

Además, en la Figura 2 se puede observar la distribución de las distintas luces Led utilizadas.

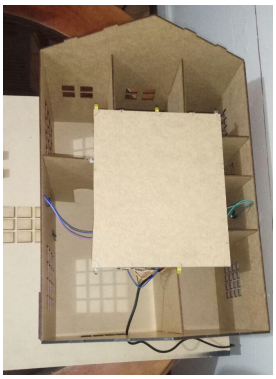

Figura 2. Vista Interna.

3. Resultados

Inicialmente se realizó la biblioteca para el manejo de los pines de entrada y salida (GPIO), tanto luces LED como botones pulsadores para la simulación de las puertas, en total se utilizaron 14 pines de la Raspberry Pi para la solución, de los cuales 13 son para GPIO y uno de tierra, la distribución de dichos pines se observa en la Figura 3.

ID	Componente	Número de Pin
Luz Cuarto 1	LED	7
Luz Cuarto 2	LED	11
Luz Sala	LED	15
Luz Comedor	LED	16
Luz Cocina	LED	13
Puerta Principal	Botón Pulsador	18 y 22
Puerta Trasera	Botón Pulsador	33 y 36
Puerta Cuarto 1	Botón Pulsador	35 y 38
Puerta Cuarto 2	Botón Pulsador	37 y 40

Figura 3. Distribución de Pines.

Por otra parte, el servidor fue exitosamente finalizado y cumple con las funciones requeridas de manera eficiente, incluso cuando varios usuarios manipulan la aplicación simultáneamente, siempre y cuando todos ellos se encuentren conectados a la misma red local.

Así mismo se cuenta con una aplicación móvil completamente funcional y amigable con el usuario la cual cumple todas las funcionalidades requeridas para el sistema. Cómo se observa en la Figura 4 la aplicación cuenta con un login, esto para tener un mayor nivel de seguridad en el sistema, además que las contraseñas están encriptadas.

Figura 4. Pantalla de Inicio.

Además en la aplicación se puede observar un mapa de la casa, en el cual se ve la distribución tanto de las puertas como de las luces que se controlan desde la aplicación, dicho mapa se puede observar en la Figura 5.

Figura 5. Mapa de la Casa.

Tanto en la Figura 6 como en la 7 se observa como se controlan la luces de manera gráfica y se puede observar el estado de las puertas, cabe destacar que las luces si se pueden encender/apagar desde la aplicación, en cambio las puertas solo muestran su estado, si están abiertas o cerradas.

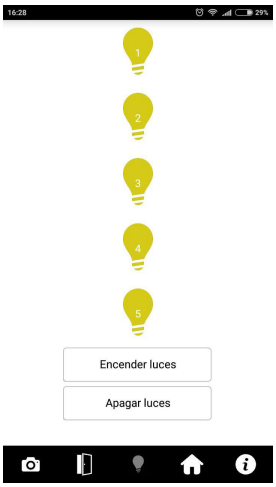

Figura 6. Manejo de Luces.

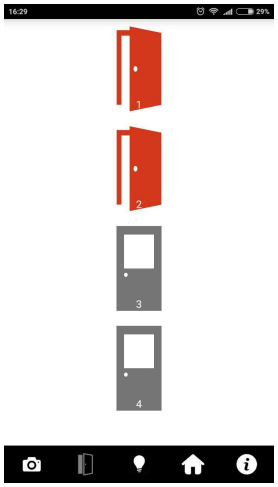

Figura 7. Estado de las Puertas.

Por último en la aplicación se cuenta con la posibilidad de solicitar una fotografía al servidor para monitorear nuestra casa, aunque la cámara utilizada no cuenta con una resolución alta se logra ver con claridad la entrada principal, en la Figura 8 se observa un ejemplo.

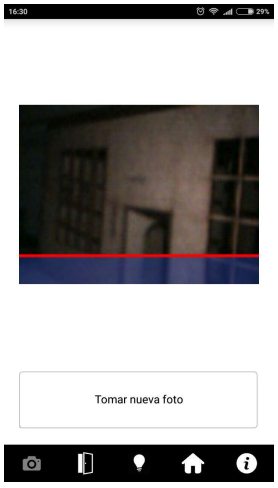

Figura 8. Fotografía de la casa desde la aplicación móvil.

4. Conclusiones

La creación de los sistemas embebidos a la medida para cualquier tipo de solución es de suma importancia, ya que en la mayoría de ocasiones se cuenta con limitaciones de memoria y/o procesamiento, por lo que crear bibliotecas que posean únicamente lo necesario y el correcto uso de los GPIO que brinde la tarjeta, en este caso la Raspberry.

Aparte de crear un sistema agradable para el usuario, relacionado con el Internet de las Cosas, involucrando diseño ingenieril en todos los puntos de vista, desde el diseño de la maqueta, implementación de un servidor web sencillo, sin dependencias externas, una aplicación móvil con su propia base de datos y seguridad se desarrolla un gran conocimiento en el área de sistemas embebidos.

Por último, la creación de sistemas operativos a la medida con Yocto, la compilación cruzada y el manejo de periféricos resulta enriquecedor en la formación de Ingenieros en Computadores.

5. Bibliografía

[1] Poesía Binaria. (2017). Creando un servidor que acepte múltiples clientes simultáneos en C - Poesía Binaria. [online] Available at: https://totaki.com/poesiabinaria/2011/02/creando-un-servidor-que-acepte-multiples-clientes-simultaneos-en-c/ [Accessed 10 Sep. 2017].

[2] Corvix, A. (2017). Timelapse 24 horas con Raspberry Pi + Webcam USB. [online] Geekytheory.com. Available at: https://geekytheory.com/timelapse-24-horas-con-raspberry-pi-webcam-usb [Accessed 10 Sep. 2017].