



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Desarrollo de una aplicación
Bluetooth para un sistema de
control de accesos**

INFORME DE ACTIVIDADES PROFESIONALES

Que para obtener el título de
Ingeniero Eléctrico Electrónico

P R E S E N T A

Rodrigo Martín del Campo Alcocer

ASESOR DE INFORME

Dr. José Ismael Martínez López



Ciudad Universitaria, Cd. Mx., 2018

Tabla de contenidos

1. Introducción y objetivo	4
a. Introducción.....	4
b. Objetivo	5
2. Descripción de la empresa o medio en el que labora	5
3. Marco teórico	6
a. Microprocesador.....	6
b. Módulo microprocesador con Bluetooth	6
c. Protocolo Bluetooth Low Energy	8
i. Características Técnicas	8
ii. Introducción al Protocolo.....	9
iii. Roles de dispositivos BLE	10
iv. Protocolo de atributos	11
v. Perfil de Atributo Genérico	11
vi. Perfil de Acceso Genérico	11
vii. Operaciones de la base de datos	11
4. Antecedentes del proyecto, tema o problemática	12
a. Seguridad de un Sistema de Control de Acceso	12
b. Accesibilidad a la unidad operadora de un SAC	12
i. Conectividad Wi-Fi.....	12
ii. Conectividad Bluetooth	13
c. Programación por hardware	13
d. Programación por software	13
5. Definición del problema o contexto de la participación profesional.....	14
a. Definición del problema	14
b. Límites y condiciones necesarias	14
6. Metodología utilizada.....	14
a. Metodología Agile.....	14
b. Etapas del desarrollo.....	16
7. Participación profesional.....	17
a. Diseño del sistema	17

b.	Aplicación para sistema embebido	19
c.	Aplicación par sistema Android	20
d.	Adaptación del código propietario del SCA	22
8.	Resultados y aportaciones	22
9.	Conclusiones	23
10.	Bibliografía	24
11.	Glosario	25
12.	Anexos	26
A.	Pseudocódigo del control utilizado el protocolo BLE.....	26
B.	Pseudocódigo de la programación de los límites de una SCA residencial.....	26

1. Introducción y objetivo

a. Introducción

Hoy en día las compañías compiten ofreciendo más opciones a los consumidores, agregando características a sus productos como la incorporación de tecnologías inalámbricas y de conectividad que están integradas en el universo del internet de las cosas (*IoT* por sus siglas en inglés). Los sistemas conectados al IoT abarcan varias industrias y cubren muchos servicios, por ejemplo: casas inteligentes que controlan temperatura e iluminación dentro de ellas, monitoreo de parámetros como la calidad del agua y del suelo en la agricultura, administración de la energía consumida por un sistema, ciudades inteligentes que continuamente monitorean e identifican situaciones de riesgo, aplicaciones médicas que son capaces de avisar en caso de emergencia o incluso alertar al paciente o al profesional de la salud de un evento antes de que suceda.

Existen varios protocolos: *Wi-Fi, Zigbee, Thread, Bluetooth*; todos ellos tienen diferentes fortalezas y son adecuados para diferentes necesidades. El *hardware* necesario para implementar circuitos de radiofrecuencia, utilizados por estos protocolos, es cada vez más potente y la cantidad de soluciones disponibles aumenta año con año. Así, los diseñadores de sistemas embebidos pueden escoger el protocolo que cubre las necesidades de su producto: consumo de energía bajo, alto ancho de banda, mayor alcance de la señal, módulos que aceleran el tiempo en el que un producto llega al mercado, soluciones ya desarrolladas, documentación existente, adopción general, entre otras.

Uno de los protocolos que ha explotado desde su nacimiento, hace menos de una década, es *Bluetooth Low Energy* (BLE). Este protocolo fue concebido e inicialmente desarrollado por la compañía finlandesa Nokia en 2010. Después fue adoptado por el Grupo Especial de Intereses de Bluetooth (*SIG* por sus siglas en inglés) en 2014. La idea de quienes definieron el protocolo era crear un estándar de comunicación que operara en radiofrecuencia con el mínimo posible consumo de potencia, específicamente optimizado para bajos costos, bajo ancho de banda y baja complejidad. En la mente de los desarrolladores estaban los diseñadores de hardware y de sistemas embebidos que trabajarían con sistemas donde la capacidad de procesamiento, el área en los circuitos impresos y la energía disponible estarían limitados. Este protocolo es capaz de correr por periodos de tiempo extendido, sea un año o dos, utilizando una pila de botón (pila tipo G), algo que hasta ahora ningún otro protocolo ha logrado.

Este protocolo llega en el momento justo en el que se necesita pues las tecnologías actuales de almacenamiento de energía aún no permiten que muchos sistemas corran de manera independiente con una batería pequeña. La relación de peso a consumo en

watts de las baterías actuales limita mucho a quienes buscan implementar soluciones de tamaño reducido (Zu, C., & Li, H., 2011). Sistemas más robustos, como los teléfonos inteligentes, están constantemente buscando reducir el consumo de potencia para extender el tiempo entre recargas de energía y este protocolo satisface esta necesidad. No es buena solución para transmitir gran cantidad de datos en poco tiempo (música, por ejemplo), pero es muy útil para monitoreo constante. Por ejemplo, en tiendas departamentales puede detectar cuando un cliente se acerca y transmitir información sobre el producto en exhibición. Es el caso de *iBeacon* de Apple que está basado en esta tecnología y le da justamente este uso. Por ello, muchos accesorios para los teléfonos inteligentes fueron creados con BLE. La conectividad de muchos de estos accesorios los vuelve inteligentes: relojes inteligentes, ropa inteligente (llamados *wearables*) (Lomas, 2018), aparatos de monitoreo médico inteligente, control de temperatura de una casa inteligente, velocímetros de bicicleta inteligentes, sistemas de detección de proximidad, escalas de peso inteligentes, etc. Inclusive, estos sistemas son capaces de medir cuanta energía queda en la batería que los alimenta, alertando de la necesidad de un cambio antes de que esta se agote. La inteligencia en este contexto se refiere a que pertenecen a un sistema capaz de tomar decisiones donde el procesamiento está distribuido o es delegado a otra entidad más poderosa en el sistema, que alimenta con datos, para finalmente ejercer un control o simplemente notificar un estado.

Para este proyecto elegimos usar BLE por las características ya mencionadas y por la facilidad de desarrollo y amplia documentación que existe. Además, hay *kits* de desarrollo que permiten enfocarse en la solución y eliminan la necesidad de diseñar hardware, un circuito impreso y certificar los componentes RF adecuados ante la autoridad o ente regulatorio correspondiente.

b. Objetivo

Demostrar que es posible desarrollar un sistema de comunicación que use el protocolo Bluetooth Low Energy (BLE) para programar los límites del viaje de la puerta en un sistema de control de accesos residencial con una respuesta de tiempo razonable y que permita controlar la posición de dicha puerta con un grado de precisión razonable.

2. Descripción de la empresa o medio en el que labora

La empresa fabrica soluciones de acceso para garajes, locales comerciales, bodegas, conjuntos residenciales, estacionamientos y otros lugares donde las personas guardan lo que más valoran.

3. Marco teórico

a. Microprocesador

Un microprocesador es un circuito integrado que contiene un procesador digital completo, debe incluir como mínimo una unidad lógica aritmética y registros asociados (Davies, 2008). Esto no es suficiente para aplicaciones prácticas, el microprocesador necesita de una memoria para poder almacenar información y de periféricos para poder comunicarse con el mundo exterior. Los microcontroladores están integrados por un procesador, una memoria y puertos de entrada y salida en un circuito integrado monolítico (Valvano, 2013). Al ser dispositivos a los cuales se les puede programar una lógica, tener la capacidad de actuar como una máquina de estados, la miniaturización a la que durante décadas han sido sujetos y la dramática reducción de precio, los han colocado en casi todos los aparatos electrónicos que usamos hoy en día: computadoras personales, licuadoras, hornos de microondas, coches, tarjetas de identificación de acceso, el control remoto de la televisión, multímetros digitales, sistemas de riego caseros, teléfonos móviles, el sistema de piloto automático de un avión y sistemas de control de accesos.

b. Módulo microprocesador con Bluetooth

El microcontrolador usado para este proyecto fue el EFR32BG11 de la familia Blue Gecko del fabricante Silicon Labs. Para poder hacer usos de los varios periféricos de este y agilizar la programación del código, Silicon Labs diseñó un circuito impreso, *PCB* por sus siglas en inglés (*Printed Circuit Board*), con el hardware necesario para facilitar el uso. A esta integración se le conoce como módulo. En el módulo está incluida la antena de RF lista para ser usada, quitando la necesidad de diseñar un elemento de alta complejidad y vital para cualquier sistema inalámbrico. Además del desarrollo de este módulo, crearon un kit con elementos de entrada y salida como botones de contacto momentáneos, LEDs (*Light Emitting Diode* o Diodo Emisor de Luz), pantalla LCD (*Liquid Crystal Display* o Pantalla de Cristal Líquida), memoria Flash serial, sensor de humedad y temperatura, conector para *debugger*, conector de *ethernet* RJ-45

El diagrama de bloques del microcontrolador se presenta a continuación en la figura 3.1

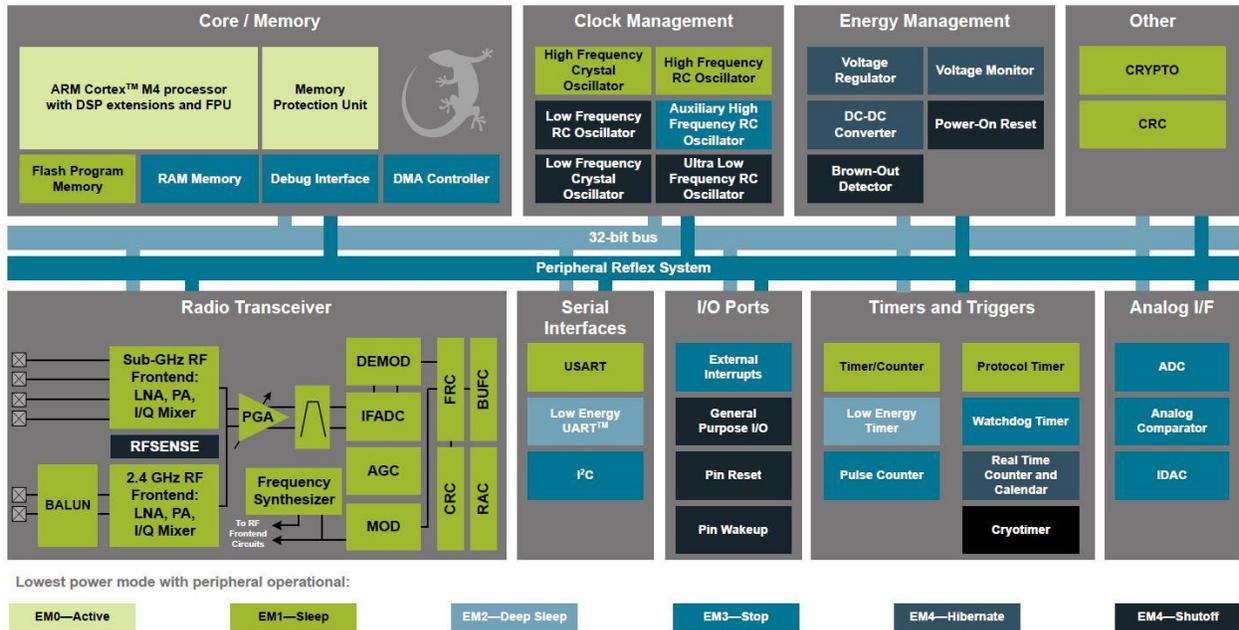


Fig 3.1

Silicon Labs, EFR32BG1 Diagrama de bloques. Austin, TX. EUA. Noviembre, 2017.

El diagrama de bloques de la tarjeta de desarrollo con la que se trabajó se muestra en la figura 3.2

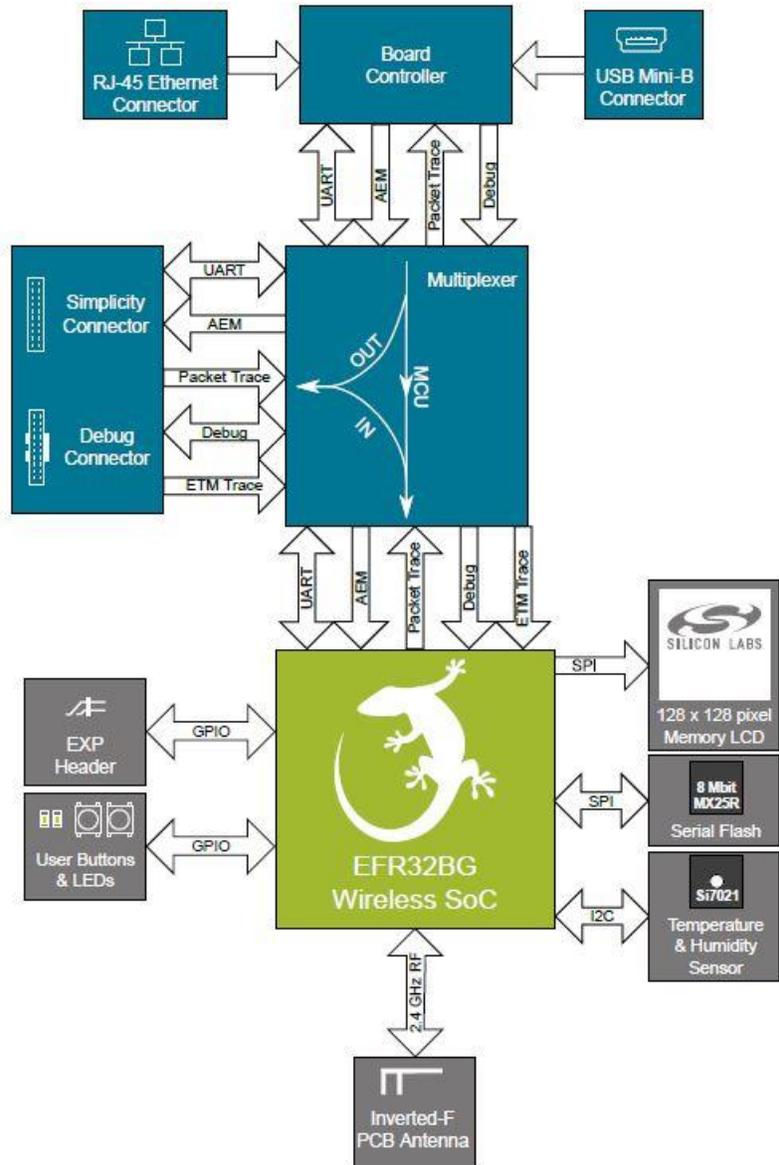


Fig 3.2

Silicon Labs, EFR32BG1 Diagrama de bloques del kit de desarrollo. Austin, TX. EUA. Noviembre, 2017.

Para la programación del módulo, Silicon Labs creó la interfaz de desarrollo gráfica Simplicity Studio, que permite programar y *debuggear* los proyectos.

c. Protocolo Bluetooth Low Energy

i. Características Técnicas

El protocolo usa la banda industrial, científica y médica para comunicarse y divide dicha banda en 40 canales de mismo ancho, desde 2.4GHz hasta 2.4835GHz. De estos canales 37 son usadas para la transferencia de datos y 3 de ellos son para notificar su presencia a otros dispositivos BLE para después establecer conexiones.

ii. Introducción al Protocolo

El *stack* del protocolo BLE es el conjunto de elementos lógicos, estructurado en capas con una jerarquía que hace posible la implementación de este. Consta de la aplicación, el huésped (*host*), y el controlador (Townsend, 2015).

La aplicación es la capa de mayor abstracción del sistema y con la cual el sistema interactúa con el usuario, obtiene información de su entorno, maneja interfaces y recibe la información con la cual trabajará el sistema. Cada proyecto BLE tendrá diferentes necesidades y por ello la implementación de la aplicación será particular para cada diseño.

El huésped (*host*) es la parte en la que se encuentra la información con la que trabaja el protocolo. También están la base de datos definida por el protocolo, los campos que estarán en esa base de datos, la seguridad junto con la encriptación y la configuración de la conexión. Otra de las funciones cruciales que está en esta capa es el control del estado de la antena.

Las partes del *host* son:

- Perfil de Acceso Genérico (Generic Access Profile GAP).
- Perfil de Atributos Generales (Generic Attribute Profile GATT).
- Control de Lógico de Vinculación y Adaptación de Protocolo (Logical Link Control and Adaptation Protocol L2CAP).
- Atributos del Protocolo (Attribute Protocol ATT).
- Manejador de Seguridad (Security Manager SM).
- Interfaz de Controlador de Huésped (Host Controller Interface HCI), parte del huésped.

Finalmente, el controlador se encarga de las tareas y de administrar eléctricamente la antena y todo el hardware de la aplicación. Muchas tareas que son demandantes en capacidad de procesamiento y fáciles de automatizar son implementadas en hardware, como son la encriptación, la generación aleatoria de números, filtrado de dispositivos que solicitan una conexión y códigos de detección de errores, entre otras funciones.

Sus partes son

- Interfaz de Controlador de Huésped (Host Controller Interface HCI), parte del controlador.

- Capa de Conexión (Link Layer LL)
- Capa Física (Physical Layer PHY)

La figura 3.3 muestra el arreglo de estas capas.

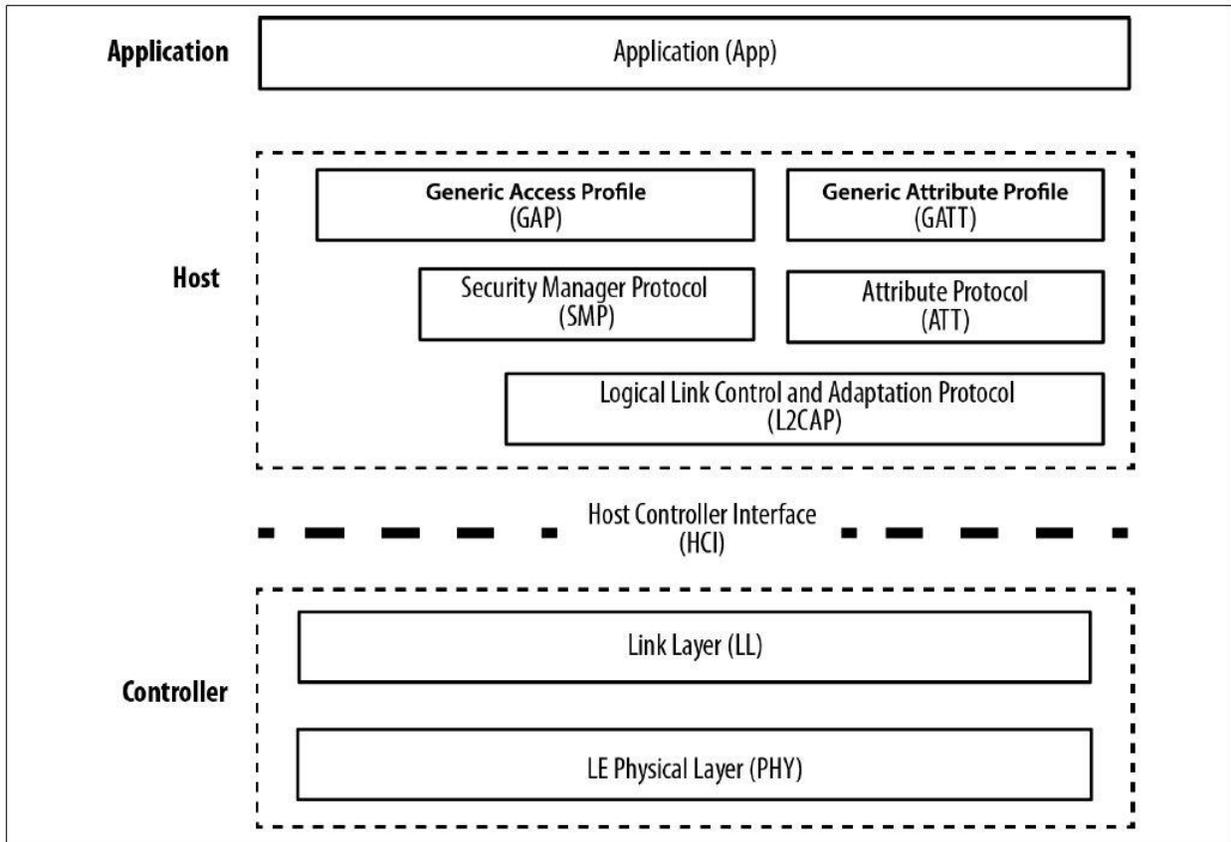


FIG 3.3

Townsend, Kevin. BLE protocol Stack. Getting started with Bluetooth Low Energy. p16. Beijing. 2015

iii. Roles de dispositivos BLE

Los diferentes roles que puede tener un dispositivo están definidos por la capa de conexión. Existen 4 roles que se agrupan en pares lógicos. Los roles son: anunciador (*advertiser*, envía paquetes anunciándose), buscador (*scanner*, busca paquetes de anunciantes), maestro (*master*, inicia y administra una conexión) y esclavo (*esclavo*, acepta una conexión y sigue el tiempo del maestro). Un dispositivo que inicia una conexión buscando a otros será maestro. Por otra parte, un dispositivo que anuncia su presencia y acepta una conexión será esclavo. Así las parejas quedan como maestro-buscador y esclavo-anunciador.

iv. Protocolo de atributos

Este es un simple protocolo cliente/servidor sin estados que trabajo sobre los atributos mostrados por un dispositivo. Un dispositivo puede ser cliente, servidor o ambos sin importar si es esclavo o maestro. La información en cada servidor está organizada como atributos y a cada uno de ellos tiene un identificador de 16 bits, un título, un identificador universal único (UUID), diferentes permisos y, más importante de todo, un valor. Cuando un cliente quiere leer o modificar un valor pide permiso al servidor usando el título. El cliente tiene la responsabilidad de hacer una interpretación correcta del valor o de proveer un valor que sea adecuado de acuerdo con el identificador único universal.

v. Perfil de Atributo Genérico

Usando el protocolo de atributos como base, añade una jerarquía y hace una abstracción. Es el elemento central para compartir información porque clasifica los datos y define como estos serán intercambiados por las aplicaciones.

Los datos ahora están encapsulados en un servicio, que consiste de una o más características. Lo que se hace es juntar el valor de un atributo con información acerca de él, como los permisos, las propiedades y el nombre. Los servicios tienen propiedades y la más importante de estas es el valor que se le asigna y si puede ser modificado.

vi. Perfil de Acceso Genérico

La capa de mayor abstracción en un dispositivo BLE es el perfil de acceso genérico (GAP) pues es aquel que dicta como el dispositivo hará las tareas de descubrimiento, establecerá una conexión, asegurará dicha conexión, todo ello para permitir que exista la comunicación entre los dispositivos. Se encuentra en este perfil el nombre que muestra a otros dispositivos e información básica del dispositivo donde está implementado como el nombre del que manufacturó el circuito integrado o el número serial que le fue asignado.

vii. Operaciones de la base de datos

El cliente y el servidor se comunican a través de la base datos de dos formas: lectura/escritura y notificaciones/indicaciones.

La lectura y escritura son operaciones donde el cliente pide acceso a los valores para leerlos o modificarlos. Si el cliente necesita enviar información al servidor, debe utilizar

la operación de escritura; si lo que necesita es información del servidor en un tiempo arbitrario debe usarse la operación de lectura. Tres ejemplos ilustran como usar estas operaciones: un sensor siempre está cambiando los valores en la base de datos que el cliente lee constantemente, un sistema de notificaciones a usuarios hará escrituras y un sistema que recibe órdenes y debe reportar los resultados tendrá las dos operaciones. En el caso de una SCA este debe reportar el estado en el que está y recibir órdenes, así que su base datos debe ser capaz de permitir lectura y escritura de datos por un cliente. El cliente es el usuario a través del sistema operativo móvil.

Las notificaciones y las indicaciones se utilizan cuando el servidor enviará información al cliente sin que el cliente lo solicite. La diferencia entre estas dos operaciones es la existencia de un acuse de recibo, conocido como *acknowledge* o *ACK*, por parte del cliente en una indicación y la falta de este en una notificación. La falta de acuse de recibo en la notificación hace al sistema más rápido, pero no hay una garantía de recibo. Un SCA hace uso de una indicación para dar aviso del cambio de estado.

El servidor no puede obtener información directamente del cliente. La forma en que se puede hacer es programar una indicación y que esta active en el cliente una operación de escritura en la base de datos

4. Antecedentes del proyecto, tema o problemática

a. Seguridad de un Sistema de Control de Acceso

La programación de los límites de un sistema de control de acceso es una parte crítica durante la instalación de este (y cuando es necesaria hacer la reprogramación de estos). En el momento de la programación se está definiendo en el sistema los límites físicos que alcanzará. Si no los cubre en su totalidad queda expuesto a riesgos o que el SCA no cumpla con la función de proteger lo que el usuario desea.

b. Accesibilidad a la unidad operadora de un SAC

i. Conectividad Wi-Fi

Actualmente los SCA fabricados por la compañía tienen integrado un módulo Wi-Fi que les permite conectarse al internet. Sólo se permite el envío de notificaciones sobre el estatus del SCA, si se ha dado algún comando a este o para dar un comando de forma remota. No existe la forma que permita controlar la programación de los límites de forma remota.

Inclusive no hay una conexión directa entre el SCA y el dispositivo que lo está controlando, todo se hace a través de un sistema en la nube propietario de la empresa llamado MyQ. Este sistema en la nube es quien envía las notificaciones tanto al SCA como al cliente que a través del portal web o por una aplicación en un dispositivo móvil hace la conexión.

ii. Conectividad Bluetooth

Actualmente, en los SCA no existe la posibilidad de conectarse por Bluetooth o hacerlo a través de un accesorio. Este proyecto busca demostrar que es posible hacer la programación de los límites usando Bluetooth con tiempos de respuesta razonables para permitir un buen control del mecanismo físico del que haga uso el SCA. La integración del hardware para Bluetooth o BLE puede existir en el circuito impreso del SCA o como un accesorio.

c. Programación por hardware

La programación de los límites en un sistema se hace en 4 partes (más un primer paso de seguridad):

0. Antes de realizar la programación de los límites, revisar que la puerta que controlará el sistema de control de accesos pueda operar en condiciones seguras.
1. Mantener el botón central presionado durante unos segundos hasta que el SCA entre en modo de programación de límites. Este avisará emitiendo dos pitidos cortos.
2. Usando los botones en forma triangular, que harán que la puerta se mueva hacia abajo o hacia arriba, colocar la puerta en la posición de apertura total y presionar el botón central para programar el límite superior.
3. Usando los botones que controlan la posición de la puerta, llevar esta al límite inferior. Una vez ahí presionar el botón central para programar el límite inferior.
4. Finalmente, para que el SCA pueda registrar la operación completa deberá hacer una acción de apertura total y cierre total.

d. Programación por software

En el momento en el que desarrollamos este proyecto no era posible programar los límites de una unidad usando una aplicación, ni en sistemas operativos móviles ni a

través de la plataforma conectada a la nube del SCA. La idea de incluir esta opción es dar al usuario mayor poder sobre su sistema y así controlar el acceso del que pretende hacer uso.

5. Definición del problema o contexto de la participación profesional

a. Definición del problema

Demostrar que es posible agregar funcionalidad a un SAC, la programación de los límites de una puerta, usando una aplicación en un sistema operativo móvil que use el protocolo Bluetooth Low Energy. Esto incluye conectar el hardware existente con lo que se desarrolle, además de escribir firmware para el microprocesador del módulo que controla el radio y comunicación Bluetooth.

b. Límites y condiciones necesarias

No podíamos quitar o reducir ninguna funcionalidad o característica ya existente al sistema de forma alguna. Esto es, se construirá sobre lo ya existe, haciendo revisión de las convenciones ya establecidas y respetando los documentos de especificaciones y requerimientos propietarios.

El resultado final también debe demostrar que no existe un retardo notable en el tiempo de respuesta entre el accionar de un comando y la ejecución de este. Esto es necesario para que se pueda realizar el procedimiento con seguridad y se puedan definir los límites con un buen grado de precisión usando una aproximación visual. Si el tiempo de retardo es grande puede ocasionar que el SAC intente desplazar la puerta contra límites físicos (por ejemplo, el suelo o el fin del riel de conducción) dañando la unidad al someterla a condiciones para las cuales no está diseñada.

6. Metodología utilizada

a. Metodología Agile

La forma de trabajo Agile fue pensada en 2001 por un grupo de desarrolladores buscando mejorar su productividad y aumentar la calidad de sus productos. Es un

proceso de administración de proyectos, es iterativo, exige colaboración entre los miembros del equipo, requiere de una comunicación constante y efectiva, además de estar constantemente evaluando los resultados. En proyectos tradicionales cada miembro del equipo desarrolla una parte y la entrega al siguiente, es revisado y liberado al final, tratando al problema por separado y sin una retroalimentación rápida que permita corregir errores que surjan en el camino o agregar nuevas funcionalidades. En cambio, cuando trabajas con Agile vas haciendo un seguimiento constante del trabajo y lo administras en pequeños incrementos. La planeación, los requerimientos y los resultados son flexibles y por ello es posible reaccionar a los cambios, sean los requerimientos, los tiempos o los recursos, con gran velocidad.

No sería correcto describir a Agile como una sola metodología, es mejor describirla como un conjunto de ellas enfocadas en la mejora a través de la retroalimentación y la mejora continua. Para comunicar efectivamente su mensaje el grupo de desarrolladores escribió un manifiesto, un párrafo breve que describe su filosofía de trabajo:

“Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:

Individuos e interacciones sobre procesos y herramientas

Software funcionando sobre documentación extensiva

Colaboración con el cliente sobre negociación contractual

Respuesta ante el cambio sobre seguir un plan

*Mientras que encontramos valor en los elementos de la derecha,
valoramos más los de la izquierda.*

Kent Beck, Mike Beedle, Arie van Bennekum,

Alistair Cockburn, Ward Cunningham, Martin Fowler,

James Grenning, Jim Highsmith, Andrew Hunt,

Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin

Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas

Una metodología que comparte los principios de retroalimentación y mejora continua es el uso de un Kanban (cartelera en japonés). Implementado por Toyota a final de la década de los años 40s, Kanban copia el modelo de los supermercados donde lo que hay en los estantes es justo la demanda calculada de los consumidores, optimizando la

cantidad de artículos disponibles y ventas, asegurando que lo que los clientes buscan siempre estará disponible. A esta forma de administración el inventario se le conoce como justo a tiempo (JIT por sus siglas en inglés *Just In Time*). Su uso en grupos de ingeniería aprovecha la idea central de la optimización de recursos y la comunicación entre las diferentes partes sobre las necesidades actuales o futuras del proyecto. El Kanban es la herramienta visual de esta metodología.

b. Etapas del desarrollo

Los grupos que trabajan de manera Agile no están restringidos a una serie de pasos o etapas, pueden decidir cuáles son las que mejor se adaptan a su ritmo y forma de trabajo. Mientras que un Kanban básico sólo tiene 4 etapas (pendientes, en progreso, en revisión y terminado) nuestro grupo identificó 5 etapas:

- **Creación:** Se definieron los miembros del equipo, sus roles (desarrollo de aplicación en sistema operativo móvil, desarrollo de firmware de microcontrolador Bluetooth, desarrollo de firmware para operador de un SCA), y los tiempos aproximados de entrega.
- **Definición:** Se establecieron claramente los objetivos del proyecto y se definieron que debe entregar cada uno. La meta fue puesta y se comenzó a discutir cual es la forma en la que las partes se comunicarían entre si. Los tiempos fueron puestos y la complejidad del proyecto fue analizada para ajustar los objetivos y metas a entregas realistas. Pensamos en casos especiales o situaciones en las que una reacción particular, es decir diferentes condiciones a las de operación normal, debía existir.
- **Desarrollo:** Empieza una etapa de aprendizaje sobre los protocolos de comunicación a usarse, cuanto tiempo es necesario para dominar estos y como será la implementación. Una visión general del sistema ya está definida y comienza a tomar forma. Los primeros prototipos son creados y probados para encontrar las oportunidades de mejora. Se empieza a trabajar iterativamente sobre los prototipos creados, puede ser que el resultado final no tenga elementos de esta primera forma, pero será la base sobre la cual se construyan las mejoras. Las partes han definido como se comunican y se retroalimentan entre ellas para formar el sistema.
- **Implementación:** Comienzan las pruebas funcionales del sistema y se prueba la comunicación entre ellas. Los objetivos son evaluados para ver si ya han sido cumplidos y se les hacen los ajustes necesarios para alcanzar las metas puestas. Se hacen pruebas funcionales completas. Se revisan que cumplan casos especiales y las excepciones que puedan surgir durante las operaciones.

- **Presentación:** Una presentación oral fue preparada con los detalles más relevantes del funcionamiento del sistema en un alto nivel de abstracción. Le dimos prioridad a la forma en la que el proyecto fue presentado y lo que era más importante destacar. Además, preparamos una demostración funcional del sistema.
- **Documentación:** Al haber trabajado de forma Agile hubo la necesidad de comunicar constantemente con los demás miembros del equipo los resultados que se estaban obteniendo. La comunicación se hizo a través de un Kanban digital donde iba quedando registro de todo lo que se había trabajado que era revisado en juntas semanales y diarias. Es así que la documentación fue un proceso presente durante todo el desarrollo del proyecto, que al final fue enriquecido con elementos (presentación de diapositivas, apuntes, códigos, motivo de decisiones) que no hubiesen sido guardados y comentarios que una vez finalizada la presentación se podían dar.

7. Participación profesional

a. Diseño del sistema

Empezamos por definir las limitaciones impuestas al diseño y modelando el comportamiento del sistema en un nivel alto de abstracción. Decidimos tomar un enfoque modular donde diferentes miembros de un equipo desarrollarían en paralelo las diferentes partes que compondrían al sistema y al final se comunicarían entre sí. Los módulos que se desarrollaron fueron: creación de una aplicación para sistema operativo móvil Android, desarrollo del firmware que controlara el hardware y la comunicación BLE y portación del firmware propietario existente para que el microcontrolador del dispositivo BLE sea capaz de enviar comandos al SCA.

Los límites impuestos al diseño fueron pensados para que el sistema se comportara de la forma más parecida a las unidades actuales en producción, permitiendo la validación de la idea propuesta. Por ello la principal restricción que tuvimos fue en la adaptación del firmware existente; debíamos agregar las capacidades de comunicación BLE sobre lo ya existente sin quitar otras funcionalidades o modificar el funcionamiento de otras. Esto dividió en dos las posibles formas de seguir con el diseño: replicar la lógica completa del sistema actual más las funcionalidades planeadas, esto es el microcontrolador Blue Gecko reemplazaría al actual del SCA, o crear una forma de que el Blue Gecko se comunique con el actual SCA.

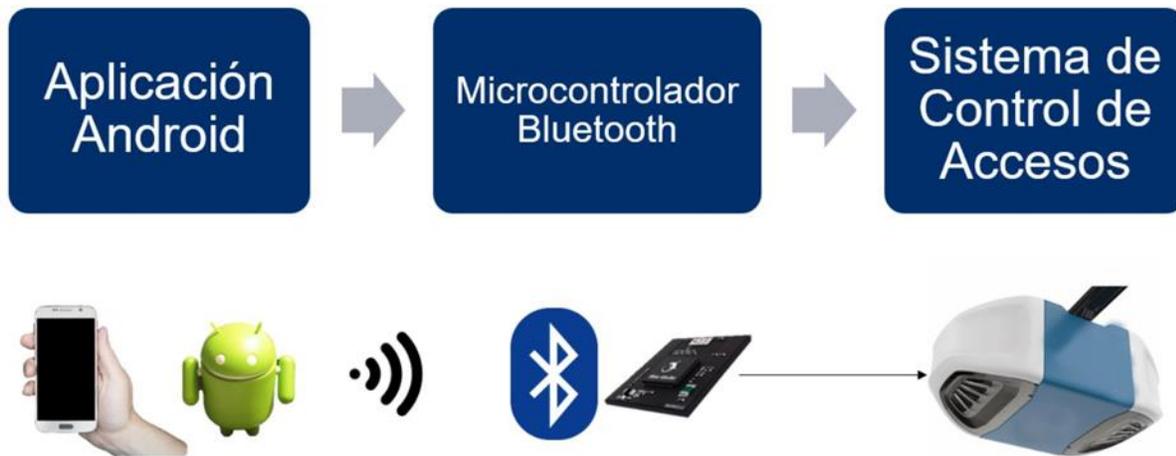


Fig 7.1 Conexión usando el BLE

Al ser los SCA sistemas que cuidan lo que más valoran las personas era indispensable que la comunicación que existía entre el teléfono móvil inteligente y el SCA fuera segura. Que, de ser interceptado (por ser ondas de radio podía ser captado por alguien con una antena y una forma de leer las señales) fuera muy difícil descifrar el mensaje. Ya existe un protocolo propietario desarrollado por la empresa para la comunicación entre los periféricos disponibles y la unidad principal. Por ello si la comunicación se estableciera directo desde la aplicación corriendo en un sistema operativo móvil al microcontrolador principal del SCA está debía ir ya encriptada y debería ser el equivalente a alguien presionando un radiotransmisor que está en producción actualmente. Ahora bien, esta forma de crear el sistema no es tan sencilla. Esta tarea equivaldría a consolidar los protocolos BLE y el propietario de la empresa en uno solo para la comunicación y transmisión. Se decidió entonces que la comunicación al sistema operativo móvil se haría a través de unos de sus accesorios: el control remoto de pared.

Usando el control remoto de pared nos daba la ventaja de trabajar con unidades que no tengan hardware para controlar dispositivos BLE, incluso unidades sin capacidad de conexión Wi-Fi podrían ser controladas usando este método. Otra ventaja de utilizar el control de pared es que este ya cuenta con el hardware para hacer los cambios de nivel del voltaje necesarios entre el microcontrolador y los que utiliza el protocolo de propietario. Únicamente fue necesario en este nuevo diseño remover el microcontrolador actual y conectar las salidas del Blue Gecko en los *pads* que donde irían las señales de comunicación.

Así hay dos etapas en de comunicación como es ilustrado en la figura 7.2

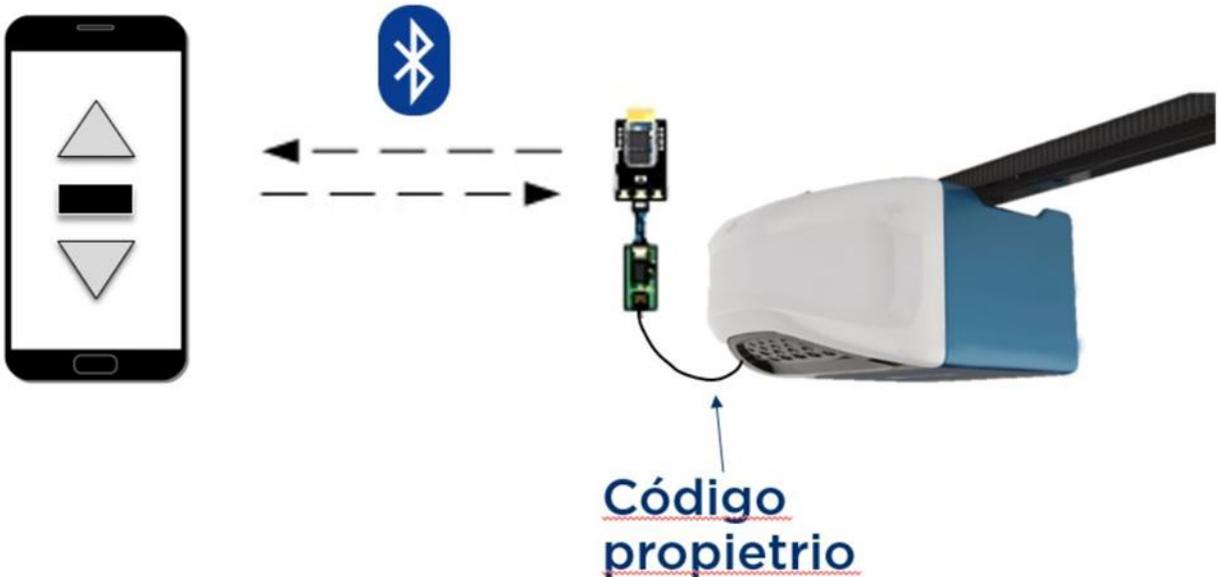


Fig 7.2 Diagrama de bloques con conexión de control remoto de pared para comunicación BLE y un actual SCA

b. Aplicación para sistema embebido

Hicimos la programación de la aplicación para sistemas embebidos utilizando la plataforma de desarrollo Simplicity Studio de Silicon Labs, el fabricante del circuito integrado que usamos.

Como desarrollamos usando el stack de BLE programamos un pequeño protocolo que serviría de interfaz para enviar notificaciones al celular y enviar comandos al SCA. Creamos un perfil que incluía lo básico que debe incluir un perfil: características del dispositivo como nombre del dispositivo visible a otros usuarios de BLE, y otras configuraciones necesarias para un dispositivo esclavo. Por estas configuraciones es que al iniciar el programa y hacer una revisión de los sistemas empezará a anunciar su presencia buscando alguien que empiece una conexión. En nuestro perfil creamos un servicio llamado "Limit Learn" que contenía dos características: modo de aprendizaje y control de dirección. Las dos son variables de dos bits, pudiendo tener 4 valores. De modo aprendizaje solo se utilizan 0 y 1 para saber si se está o no en modo aprendizaje y para control de dirección sabrá si debe empezar o detener el movimiento hacia arriba o hacia abajo. Cada una de ellas representaba a los diferentes botones que existían en la unidad operadora del SCA, el botón para entrar en el límite de aprendizaje y el botón que hace los desplazamientos de la puerta.

El pseudocódigo del algoritmo para hacer el control de la unidad se encuentra en el anexo A de este documento. Descrito a nivel alto, primero el microcontrolador se inicializa, establece sus parámetros y empieza a buscar una conexión. Ya que algún dispositivo se ha conectado, espera a que el maestro quiera cambiar alguno de los valores de las características del servicio Limit Learn. Cuando identifica que uno de los valores ha sido cambiado identifica cual y actuará dependiendo del valor que fue actualizado: entrar en modo aprendizaje, empezar el movimiento del SCA en dirección arriba, detener el movimiento del SCA en dirección arriba, empezar el movimiento del SCA en dirección abajo y detener el movimiento del SCA en dirección abajo. Para fijar los límites se debe seguir el siguiente procedimiento: mantener presionado el botón de aprendizaje de límites hasta que suene un pitido, primero establecer el superior, luego el inferior y completar una operación completa de apertura y cierre, se vuelve a presionar el botón de aprendizaje de límite y queda completa la programación. El algoritmo escrito en pseudocódigo se encuentra en el anexo B.

c. Aplicación par sistema Android

El objetivo de desarrollar la aplicación para sistema operativo móvil es dar al usuario el poder de controlar la unidad una vez que esta haya sido instalada sin la necesidad de tener que estar en proximidad a la unidad pues muchas veces esta se encuentra fuera del alcance, montada en el techo del lugar en el que está.

Para la aplicación hicimos dos pantallas: la principal y la de control del movimiento. La página principal muestra una pantalla de bienvenida y despliega los dispositivos Bluetooth activos. Una vez que encuentra uno que tenga el identificador de nuestra aplicación, lo mostrará en pantalla y al presionarlo iniciará la conexión y enviará a la pantalla de control de movimiento.

El objetivo de la pantalla de control de movimiento es replicar los botones que existen en una unidad SCA residencial. Por ello no existe en la aplicación la programación de algún protocolo propietario de la empresa, solo hará cambios en la base de datos GATT del dispositivo BLE que serán interpretados de acuerdo a las definiciones acordadas en el diseño de la comunicación entre la aplicación y el microcontrolador. Se diseñó con especial cuidado la parte gráfica de esta parte para que fuera una reproducción lo más cercana posible a los botones existentes y que un usuario ya familiarizado con la operación y programación de límites no tuviese que aprender una nueva secuencia de programación.

Cuando la aplicación pierde conexión con el dispositivo móvil hace regresar la pantalla principal para nuevamente empezar la búsqueda de dispositivos BLE

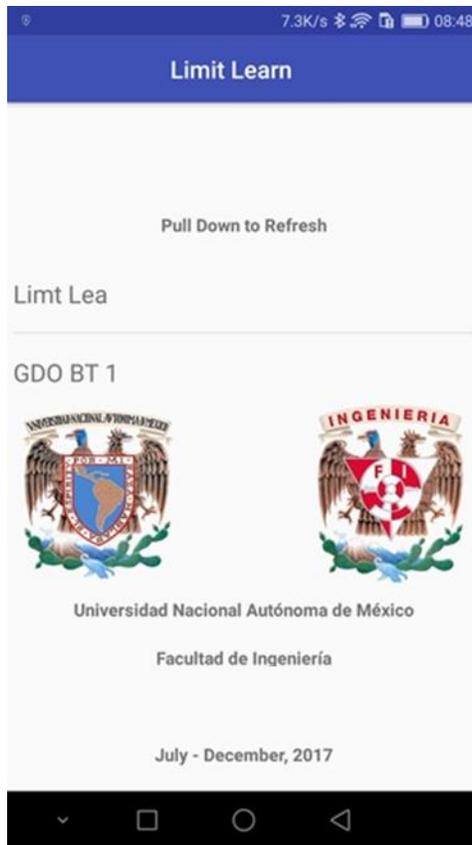


Fig. 7.3 Pantalla de bienvenida de la aplicación para Sistema Operativo Móvil Android 7.0



Fig 7.3 Fotografía que muestra los botones de una unidad SCA residencial

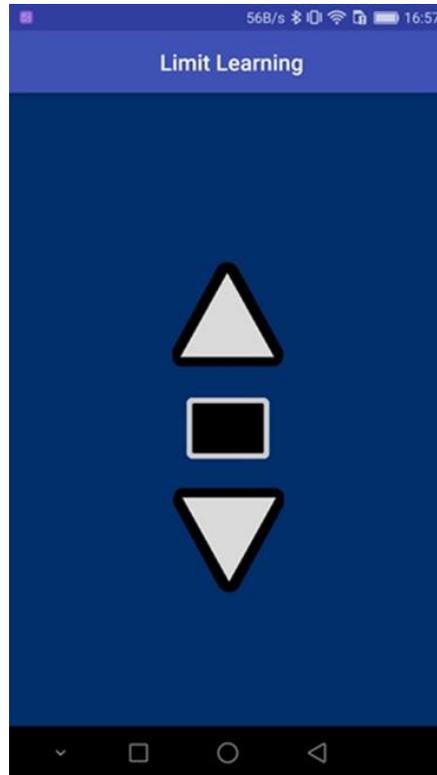


Fig. 7.4 Pantalla donde se muestran los controles que emulan los botones que estarían en la unidad de una SCA

d. Adaptación del código propietario del SCA

La adaptación del código propietario consistió en estudiar y entender la máquina de estados para poder enviar los comandos necesarios para la programación de límites. De nuestro análisis pudimos definir que necesitábamos poder enviar la secuencia de código definida como mover la puerta hacia arriba, hacia abajo, detener la puerta y enviar comando de aprendizaje de límite.

Para no tener que reescribir todo el código en el microcontrolador en el archivo principal *main* se hizo una integración de las librerías que ya contienen la secuencia de señales necesarias para enviar el comando y solo se crearon funciones que invocan estas señales.

8. Resultados y aportaciones

Logramos portar un firmare propiedad intelectual de la empresa que manufactura el SCA al microprocesador del módulo donde está implementado el protocolo Bluetooth y BLE y que controla el radio.

Logramos comunicar electrónicamente el módulo Bluetooth y el SCA haciendo modificaciones al hardware de uno de los accesorios de este. El accesorio hace el cambio en el nivel de voltaje para que los circuitos lógicos del módulo Bluetooth se comuniquen con los del SCA.

Desarrollamos una aplicación para sistema operativo móvil Android 7.1 capaz de emitir instrucciones para cambiar el estado de la puerta del SAC y recibir actualizaciones de los estados usando el protocolo BLE.

Logramos programar exitosamente los límites de un sistema de control de accesos usando la aplicación Android desarrollada.

Demostramos que es posible la programación de los límites de un SAC usando una aplicación corriendo en un dispositivo con sistema operativo móvil, en este caso Android 7.1, usando el protocolo BLE sin que existiera un retardo notable en el tiempo de respuesta entre la acción de dar un comando a través de la aplicación y que el SAC lo ejecute, por ejemplo, empezar o detener el movimiento de la puerta, asegurando una operación segura y precisa.

Queda pendiente el desarrollo de funciones en las que se consideren casos especiales que implementen medidas adicionales de seguridad, como por ejemplo detener la puerta si se sale de la aplicación o se apaga el teléfono. Código que sea redundante pero que añada una capa extra de protección durante la operación de cerrado, considerada la etapa de mayor riesgo para un usuario, es probablemente el siguiente paso para este desarrollo.

9. Conclusiones

Los conocimientos técnicos necesarios para desarrollar un sistema embebido exigen una capacidad de aprendizaje rápida además de una base sólida para entender los principios de las partes que integran el sistema. Sería imposible diseñar un sistema como el que se logró hacer sin la capacidad de comprender el sistema en un alto nivel de abstracción y entender las funciones de los circuitos. En lo que corresponde a la parte Bluetooth es necesario conocer el protocolo, la secuencia de instrucciones, las capacidades del microcontrolador y lo que es capaz de recibir de su contraparte en el sistema; para lograr la comunicación física entre el módulo Bluetooth y el SCA es necesario conocer el circuito.

Se logró demostrar que es posible hacer un control con un tiempo de respuesta razonable entre orden de comando y ejecución de este. El protocolo Bluetooth Low Energy es

adecuado para realizar este tipo de tareas por su bajo de potencia, el rápido procesamiento y sencillez en la implementación. La forma de trabajar Agile fue una forma que nos dio realimentación y nos permitió reaccionar rápidamente a los cambios en los límites del diseño y las nuevas ideas o mejoras que iban surgiendo durante el desarrollo del proyecto. El diseño e implementación de este sistema nos dio nuevas habilidades para trabajar en diferentes niveles de abstracción, nivel alto, implementación de aplicación BLE y aplicación para sistema operativo móvil, bajo, conexión de hardware entre los componentes y conexiones, y medio en el que relacionábamos las partes para que pudieran trabajar y poder hacer una demostración con el sistema.

10. Bibliografía

Davies, J. (2008). *MSP430 microcontroller basics*. Oxford: Newnes.

Townsend, K. (2015). *Getting started with Bluetooth low energy*. Beijing: O'Reilly.

Valvano, J. (2013). *Introduction to the Arm® Cortex(TM)-M3*. [s.l.]: Eigenverl. d. Verf.

Zu, C., & Li, H. (2011). Thermodynamic analysis on energy densities of batteries. *Energy & Environmental Science*, 4(8), 2614. <http://dx.doi.org/10.1039/c0ee00777c>

Agile design processes and guidelines | Atlassian. (2018). *Atlassian*. Retrieved 30 April 2018, from <https://www.atlassian.com/agile/design>

Getting started with iBeacon. (2018). *Developer.apple.com*. Retrieved 24 April 2018, from <https://developer.apple.com/ibeacon/Getting-Started-with-iBeacon.pdf>

Kanban - A brief introduction | Atlassian. (2018). *Atlassian*. Retrieved 30 April 2018, from <https://www.atlassian.com/agile/kanban>

Manifesto for Agile Software Development. (2018). *Agilemanifesto.org*. Retrieved 30 April 2018, from <http://agilemanifesto.org/>

Our History | Bluetooth Technology Website. (2018). *Bluetooth.com*. Retrieved 8 February 2018, from <https://www.bluetooth.com/about-us/our-history>

Who Needs the Internet of Things?. (2018). *Linux.com | The source for Linux information*. Retrieved 10 February 2018, from <https://www.linux.com/news/who-needs-internet-things>

Wireless and RF Solutions | Silicon Labs. (2018). *Silabs.com*. Retrieved 10 February 2018, from <https://www.silabs.com/products/wireless>

Lomas, N. (2018). *Global wearables market to grow 17% in 2017, 310M devices sold, \$30.5BN revenue: Gartner*. *TechCrunch*. Retrieved 16 February 2018, from

11. Glosario

- BLE: Bluetooth Low Energy, es un protocolo distinto a Bluetooth tradicional, enfocado a consumo mínimo de potencia optimizado para bajos costos, bajo ancho de banda y baja complejidad.
- Bluetooth: Protocolo de comunicación inalámbrica nacido a mediados de la década de los años noventa en la compañía Ericsson.
- Debugger: Conjunto de circuitos electrónicos y capacidad de ejecutar un programa una acción a la vez para poder detectar errores en la programación, conocidos como *bugs*, bichos en inglés. Se les llama así a los errores que no permiten una correcta ejecución o entorpecen un programa.
- Ethernet: Es el protocolo de red con la mayor adopción en para redes de área local. Este protocolo describe como será el envío y empaquetamiento de datos al ser enviados de una red a otra.
- Firmware: Es un código, que puede ser programado en un lenguaje de alto o bajo nivel, insertado en un microcontrolador con un bajo nivel de abstracción, esto es, trabaja con las señales que recibe de los periféricos directamente conectados a él. Un ejemplo de firmware es el control remoto de la televisión que emite diferentes códigos a través del LED infrarrojo que tiene dependiendo del botón que haya sido presionado.
- Hardware: Se refiere a los componentes electrónicos de un sistema
- IoT: Internet of Things (internet de las cosas), es un termino usado para describir todos aquellos productos que incorporen tecnologías de conectividad, principalmente al internet, y que pueden tomar datos y/o ser controlados.
- JIT: Metodología de manufactura y para prestadores de servicios, como por ejemplo comercios o soporte técnico, que busca optimizar la relación entre la demanda de un producto y la cantidad en inventario. La optimización busca que exista el mínimo inventario y que el cliente siempre encuentre lo que busca.
- SCA: Sistema de control de acceso, es el conjunto de elementos, puertas, mecanismos, operador electromecánico y accesorios, que se usan para controlar el acceso a un sitio.
- Stack: Es el conjunto de protocolos que hacen posible la comunicación de un dispositivo en una red. Generalmente, un dispositivo que implementa un *stack* necesita de tres partes: datos, transportación y aplicación. Esto describe la

información que se va transportar, como y en que medio se va a transportar esa información y el procesamiento de esa información.

12. Anexos

A. Pseudocódigo del control utilizado el protocolo BLE

```
1  INICIO
2  Cargar librerías, definir constantes, cargar valores de configuración
3  Inicializar microcontrolador, kit de prueba y aplicación
4  MIENTRAS
5      Ciclo esperar evento
6      CASO evento ES cargar sistema
7          Recibir parámetros de configuración como anunciante
8          Entrar en modo anunciador
9      CASO evento ES conexión terminada
10         Hacer RESET del sistema
11         Entrar en modo anunciante
12     CASO evento ES cambio de valor en la base GATT
13         SI evento ES cambio de valor de limit learn
14             Enviar mensaje entrar en modo de aprendizaje de límites
15         SI evento ES cambio de valor de control de dirección
16             SI control de dirección ES 0
17                 Empieza movimiento hacia arriba
18             SI control de dirección ES 1
19                 Detén movimiento hacia arriba
20             SI control de dirección ES 2
21                 Empieza movimiento hacia abajo
22             SI control de dirección ES 3
23                 Detén movimiento hacia abajo
24     FIN MIENTRAS
25     FIN
```

B. Pseudocódigo de la programación de los límites de una SCA residencial

- 1 INICIO
- 2 Mantener presionado el botón Limit Learn hasta escuchar un pitido
- 3 Programar el límite superior del SCA
- 4 Presionar para empezar el movimiento o soltar para detener en la dirección deseada el botón correspondiente
- 5 Una vez que la puerta del SCA se encuentre en la posición considerada como abierta oprimir el botón Limit Learn hasta escuchar un pitido
- 6 Programar el límite inferior del SCA
- 7 Presionar para empezar el movimiento o soltar para detener en la dirección deseada el botón correspondiente
- 8 Una vez que la puerta del SCA se encuentre en la posición considerada como abierta oprimir el botón Limit Learn hasta escuchar un pitido.
- 9 Oprimir el botón de dirección superior y esperar a que la puerta del SCA llegue al límite superior
- 10 Oprimir el botón de dirección inferior y esperar a que la puerta del SCA llegue al límite inferior
- 11 FIN