Actividad 1.1: Protocolo BlueTooth LE

Dispositivos y Sistemas Inalámbricos

Francisco M. Delicado Martínez*

Depto. Sistemas Informáticos

Escuela Superior de Ingeniería Informática de Albacete

Universidad de Castilla-La Mancha

Índice

1. Objetivo **2**

2. Requerimientos Hardware y Software **2**
2.1. Requisitos Hardware 2
2.2. Requisitos Software 3

3. Metodología a aplicar **3**

4. ¿Qué hay que hacer? **4**
4.1. Escaneo de dispositivos Bluetooth. 4
4.2. Conexión con un servidor. 4
4.3. Descubrimiento de servicios. 5
4.4. Descubrimiento de características. 5
4.5. Lectura de características. 6
4.6. Escritura de características. 7
4.7. Desconexión. 7

5. Evaluación de la actividad. **8**

*francisco.delicado@uclm.es

1. Objetivo

El objetivo general de esta actividad es que el alumno sea capaz de comprender y analizar el funcionamiento del protocolo *Bluetooth Low Energy* (BLE). Para ello el alumno deberá de alcanzar los siguientes objetivos paraciales:

1. Conocimiento y utilización del conjunto de herramientas incluidas en el paquete de implementación del protocolo Bluetooth en Linux: `BlueZ`.

2. Comprender las distintas capas del protocolo *Bluetooth Low Energy*.

3. Saber analizar, e identificar, el conjunto de tramas intercambiadas en una conexión *Bluetooth Low Energy*.

2. Requerimientos Hardware y Software

Para la realización de la actividad se requieren la utilización de dispositivos Bluetooth LE, así como herramientas software para el intercambio de información entre dichos dispositivos Bluetooth LE. A continuación se especifican los requerimientos tanto a nivel de hardware como de software que son necesarios para la realización de la actividad.

2.1. Requisitos Hardware

El hardware mínimo necesario será el que nos permita tener un cliente y un servidor Bluetooth LE.

Hardware para el cliente.
En el caso de no disponer de adaptador Bluetooth LE en el computador que hará las veces de cliente, deberemos de utilizar un adaptador USB Bluetooth 4.0 [3]. Este tipo de adaptador es compatible tanto con Bluetooth version 2.1 como en la versión 4.0 o LE.

Hardware para el servidor.
También necesitamos un dispositivo Bluetooth que realice las funciones de servidor. En nuestro caso hemos elegido el LigthBlue Bean© [12].

Este dispositivo es un microcontrolador compatible con Arduino, y programable, el cual puede comunicarse mediante *Bluetooth Low Energy* con

cualquier cliente que esté en su rango de cobertura. El dispositivo tiene incluidos tres sensores: temperatura, acelerómetro en tres dimensiones, y un LED RGB; además de poderle incorporar muchos más sensores utilizando los pines de extensión que tiene incorporados.

Cada uno de los dispositivos *LigthBlue Bean* dispone que cinco registros o *scratch*, que se pueden utilizar para el intercambio de información entre el cliente y el servidor. Además posee múltiples servicios que pueden ser accedidos por el cliente. En concreto, para esta práctica, cada *LigthBlue Bean* escribirá periódicamente en un *scratch* la temperatura captada por su sensor, e iluminará su LED RGB con el valor RGB situado en otro *scracth* (para un listado de representación de colores en RGB se puede consultar [2].

Como cualquier dispositivo Bluetooth, el acceso a sus *scratch* es mostrado por los dispositivos *LigthBlue Bean*, como acceso a un determinado servicio. En concreto, el UUID del servicio de acceso a los *scratch* es el `a495ff20c5b...`. Mientras, que cada uno de los *scratch* es mostrado como una característica dentro de dicho servicio. En el caso de la temperatura, se ha elegido el 2º *scratch* para su almacenamiento, el cual corresponde a la característica cuyo UUID es `a495ff22c5...`. Por otro lado, el color RGB a ser representado en el LED, se almacena en el 1er *scratch*, cuyo UUID de característica es `a495ff21c5...`.

2.2. Requisitos Software

Desde el punto de vista software, y dado que los dispositivos LigthBlue Bean, ya están programados, necesitaremos:

- Herramientas de comunicación para el protocolo Bluetooth LE, en la parte del cliente. En nuestro caso, al utilizar como cliente un PC con sistema operativo Linux Ubuntu, necesitaremos tener instalado el paquete `BlueZ` [10].

- Una herramienta para el análisis de los mensajes enviados por el protocolo Bluetooth LE. Para ello utilizaremos el conocido software de análisis de protocolos de comunicación `wireshark`[11].

3. Metodología a aplicar

Para realizar la actividad, se seguirán los siguientes pasos, en cada una de las fases de la misma:

1. Ejecución de la acción correspondiente: escaneo de dispositivos, conexión, descubrimiento de servicios y/o características, escritura y/o lectura de valores y desconexión. Mediante la utilización de alguna de las herramientas del paquete `BlueZ` [10].
2. Captura de los mensajes y comandos Bluettooth LE enviados por la acción del punto anterior. Para ello se utilizará `wireshark`.
3. Análisis de los mensajes capturados. Para ello se utilizará la documentación proporcionada en [4], [9], así como las páginas de Bluetooth SIG donde se describen los códigos para los servicios [8], las características [5], declaraciones y descriptores [6, 7]

4. ¿Qué hay que hacer?

Cada alumno dispondrá de un PC con un dispositivo de comunicación Bluetooth LE, y el packete `BlueZ` instalado. Además de una serie de dispositivos *LigthBlue Bean*.

Lo primero que deberá hacer el alumno es activar la captura de paquetes en la interfaz Bluetooth de su PC, mediante el uso de `wireshark`. Una vez hecho esto el alumno deberá de hacer cada una de las siguientes acciones:

4.1. Escaneo de dispositivos Bluetooth.

Una vez realizada esta parte de la actividad, el alumno deberá ser capaz de identificar los mensajes que están asociados a un escaneo de dispositivos, e interpretar el contenido de dichos mensajes.

Para ello, utilizando la herramienta `hcitool`, el alumno deberá escanear los dispositivos Bluetooth LE que están en el área de cobertura del PC.

Durante el escaneo de dispositivos se capturarán por parte de *wireshark* una serie de mensajes y paquetes Bluetooth, que el alumno con la ayuda de [4] y el estándar Bluetooth [9] deberá de interpretar. Comprobando que tanto el formato de mensajes, su contenido como la temporización cumplen con las especificaciones del estándar. Y de esta forma, deducir como se han generado las salidas obtenidas por la aplicación `hcitool`.

4.2. Conexión con un servidor.

El objeto de esta parte de la actividad es que el alumno identifique e interprete los mensajes que tienen lugar durante el establecimiento de una conexión Bluetooth LE.

Ahora el alumno seleccionará uno de los dispositivos anteriormente descubiertos durante el escaneado, y tomará su MAC. Ésta será utilizada como parámetro de entrada de la aplicación `gatttool`. La cual se ejecutará en modo interactivo, ver [10].

Una vez inicializada la aplicación `gatttool` se ejecutará el comando de conexión, y se estudiarán los mensajes capturados mediante `wireshark`; interpretándolos y comprobando que su contenido y secuenciación corresponde con lo que se indica en el estándar (utilizar [4] y [9].

4.3. Descubrimiento de servicios.

En esta parte de la actividad el alumno descubrirá los servicios que un dispositivo Bluetooth pone a disposición de los clientes que se conecten a él. Y aprenderá qué mensajes son utilizados para dicho descubrimiento, además de interpretar el contenido de los mismos.

Una vez conectado con un servidor (esto ha sido realizado en el apartado anterior), ahora utilizando el comando correspondiente de `gatttool`, descubriremos los servicios publicados por el dispositivo Bluetooth.

En esta parte, el alumno deberá de hacer dos cosas:

1. Utilizando la salida de la aplicación `gatttool`, [8] y/o [4], determinar en el caso de que sea un servicio estandarizado, de que servicio se trata, y para que se usa.

2. Analizar los mensajes capturados mediante `wireshark`, para identificar qué mensajes intervienen en el descubrimiento de mensajes, así como su contenido. Y comprobar que todos cumplen con las especificaciones del estándar [4] [9].

4.4. Descubrimiento de características.

Como en la sección anterior, ahora el alumno deberá de descubrir las características asociadas a un servicio. Para así, poder determinar y analizar que mensajes son los que intervienen en dicho descubrimiento y cual es su contenido y significado.

Para ello, y con la aplicación `gatttool` descubriremos las características asociadas a varios de los servicios anteriormente descubiertos. Como en el caso de los servicios, el alumno deberá realizar dos acciones diferentes:

1. Por un lado, analizar la salida de la aplicación `gatttool`, para con la ayuda de [4] y [5], determinar de que características se trata, y en el

caso de ser características estandarizadas, ver como las propiedades obtenidas de las mismas, corresponden con las estandarizadas.

2. Y en segundo lugar, analizando las capturas de `wireshark` asociadas al descubrimiento, ver que mensajes se intercambian cliente y servidor durante el descubrimiento, estudiando su contenido y viendo como tanto los mensajes como el contenido corresponde con el estándar, y con el resultado mostrado por la aplicación `gatttool`.

4.5. Lectura de características.

En esta sección el alumno deberá de leer características del dispositivo Bluetooth, siempre y cuando estas tengan la propiedad de accesibles para lectura. Además el alumno deberá interpretar los mensajes asociados con una operación de lectura.

Para realizar lo anterior, se seguirá utilizando la herramienta `gatttool`. El alumno deberá de seleccionar una característica cuyo valor leerá, de entre las que ha descubierto en el paso anterior. Una vez elegida, y comprobado que dicho valor puede ser accedido para lectura, ejecutará el comando de lectura. Una vez ejecutado dicho comando el alumno deberá:

1. Interpretar el valor devuelto por la aplicación `gatttool`. Hay que tener en cuenta que la aplicación devuelve siempre el dato en formato hexadecimal, siendo tarea del alumno determinar si dicho valor ha de interpretarse como un *string*, un entero, un número en punto flotante, o utilizando cualquier otro formato.

   Para elegir una correcta interpretación, se deberá de utilizar la descripción de la característica dada por el estándar [5], si la característica está estandarizada. En caso de ser una característica definida por el fabricante, es la especificación del mismo la que nos dará el método de interpretación. Toda característica con un UUID de 16-bits está estandarizada, mientras que las que tienen un UUID de 128-bits son definidas por el fabricante, [4].

   En el caso de las dos características definidas por el fabricante, y utilizadas en esta práctica, como son la Temperatura y el valor de LED RGB, el formato con que se guarda el valor es: para la Temperatura (`UUID = a495ff22c5b...`) es un entero de 8 bits. Mientras que para el LED RGB (`UUID = a495ff21c5...`) el valor se codifica como una tupla de enteros de 8its *red,green,blue*. Por ejemplo: el valor `0xFF0015` indicará un valor de *red*=255, *green*=0 y *blue*=21. Una lista de valores RGB, con los colores correspondientes, puede verse en [2].

2. Además, el alumno deberá analizar que mensajes se envían y el contenido de los mismos, para la lectura de una característica. Comprobando como el valor asociado a una característica es enviado al cliente. Para ello utilizará el analizador de tráfico `wireshark`.

Para la conversión de un valor hexadecimal a una cadena de caracteres y viceversa se puede utilizar el siguiente enlace [1].

4.6. Escritura de características.

El objetivo de esta sección es similar a la de lectura de características, solo que ahora se realizará la escritura. Lo primero que habrá que tener en cuenta es si el valor de la característica puede ser escrito por el cliente o no. Para ello habrá que analizar el contenido del campo propiedades de las características descubiertas.

Una vez obtenida una característica que se puede escribir, se realizará una escritura de su valor. Se debe de utilizar la codificación correcta del valor a escribir, ya que el servidor no realiza dicha comprobación, siendo esto responsabilidad del cliente. Para ello hay que ver o bien las especificaciones del fabricante, en el caso de características no estándares, o mirar en [5] para el caso de características estándares.

En el caso que nos ocupa, una sencilla manera de comprobar que se realizan las escrituras correctamente, es escribir directamente en el *scratch* asociado al color RGB del LED; y ver como este cambia su color según escribimos un valor distinto de código RGB (ver en [2] códigos RGB asociados a colores).

Con todo esto el alumno deberá:

1. Modificar el valor del color RGB a representar en el LED (UUID de la característica `a495ff21c5..`). La escritura se deberá de realizar con y sin confirmación. Y comprobar si el LED RGB cambia de color o no.

2. Interpretar mediante `wireshark` los mensajes que se intercambian el cliente y el servidor a la hora de realizar una escritura de una característica.

4.7. Desconexión.

Por último el alumno deberá de realizar la desconexión del cliente con el servidor mediante el comando correspondiente de `gatttool`. Analizando los mensajes que se envían en esta acción mediante `wireshark`.

5. Evaluación de la actividad.

Esta actividad no se evaluará como tal. Si no que, servirá como actividad de aprendizaje y entrenamiento para la realización de un posterior cuestionario.

En este cuestionario se le proporcionará al alumno una, o varias, trazas de tráfico Bluetooth LE, y se realizarán preguntas sobre el contenido y significado de uno o varios de los mensajes contenidos en dichas trazas. Este cuestionario sí será evaluable.

Referencias

[1] Convert String To Hexadecimal Online. http://string-functions.com/string-hex.aspx. Accedido: 21-03-2016.

[2] RapidTables.com - Engineering Resources. http://www.rapidtables.com/web/color/RGB_Color.htm. Accedido: 21-03-2016.

[3] Adafruit©. Bluetooth 4.0 USB Module. https://www.adafruit.com/products/1327. Accedido: 21-03-2016.

[4] Francisco M. Delicado Diego Hortelano. Descripción del protocolo BLE. Technical report, Dpto. Sistemas Informátivos. UCLM, 2016.

[5] Bluetooth SIG, Inc. Characteristics | Bluetooth Development Portal. https://developer.bluetooth.org/gatt/characteristics/Pages/CharacteristicsHome.aspx. Accedido: 2016-03-16.

[6] Bluetooth SIG, Inc. Declarations | Bluetooth Development Portal. https://developer.bluetooth.org/gatt/declarations/Pages/DeclarationsHome.aspx. Accedido: 2016-03-16.

[7] Bluetooth SIG, Inc. Descriptors | Bluetooth Development Portal. https://developer.bluetooth.org/gatt/descriptors/Pages/DescriptorsHomePage.aspx. Accedido: 2016-03-16.

[8] Bluetooth SIG, Inc. Services | Bluetooth Development Portal. https://developer.bluetooth.org/gatt/characteristics/Pages/ServicesHome.aspx. Accedido: 2016-03-16.

[9] Bluetooth SIG, Inc. Specification of the Bluetooth System. Covered Core Package version: 4.2. Technical report, Bluetooth SIG, Inc., 2010.

[10] Francisco M. Delicado. BlueZ Tools. Technical report, Dpto. Sistemas Informátivos. UCLM, 2016.

[11] Wireshark Fundation. Wireshark - Go Deep. `www.wireshark.org`. Accedido: 21-03-2016.

[12] Punch Through Design, LLC. The LightBlue Family - Bean. `https://punchthrough.com/bean`. Accedido: 21-03-2016.